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

Olav Møyner and Knut-Andreas Lie

March 15, 2016

Abstract

Simulation problems encountered in reservoir management are often computationally expensive because of the complex fluid physics for multiphase flow and the large number of grid cells required to honor geological heterogeneity. Multiscale methods have been proposed as a computationally inexpensive alternative to traditional fine-scale solvers for computing conservative approximations of the pressure and velocity fields on high-resolution geo-cellular models. Although a wide variety of such multiscale methods have been discussed in the literature, these methods have not yet seen widespread use in industry. One reason may be that no method has been presented so far that handles the combination of realistic flow physics and industrystandard grid formats in their full complexity. Herein, we present a multiscale method that handles both the most wide-spread type of flow physics (black-oil type models) and standard grid formats like corner-point, stair-stepped, PEBI, as well as general unstructured, polyhedral grids. Our approach is based on a finite-volume formulation in which the basis functions are constructed using restricted smoothing to effectively capture the local features of the permeability. The method can also easily be formulated for other types of flow models, provided one has a reliable (iterative) solution strategy that computes flow and transport in separate steps.

The proposed method is implemented as open-source software and validated on a number of two and three-phase test cases with significant compressibility and gas dissolution. The test cases include both synthetic models and models of real fields with complex wells, faults, and inactive and degenerate cells. Through a prescribed tolerance, the solver can be set to either converge to a sequential or the fully implicit solution, in both cases with a significant speedup compared to a fine-scale multigrid solver. Altogether, this ensures that one can easily and systematically trade accuracy for efficiency, or vice versa.

1 Introduction

The subsurface flow of hydrocarbons during petroleum production is a complex multiscale process. Whereas the flow of liquid and gas phases occurs on a micrometer scale in the void between rock grains, buoyancy forces and pressure differentials induced by wells drilled into the rock will cause the hydrocarbon components to move hundreds or thousands of meters before they reach the point where the well tube perforates the rock. The porous rock's ability to store and transmit fluid is heterogeneous on all relevant length scales in between and typically has spatial correlations that vary from centimeter to kilometer scale. This heterogeneity has a profound impact on the movement of fluids. Obviously, one neither can nor should attempt to resolve all pertinent scales in a single model. Indeed, in the reservoir characterization process a combination of expert intuition, experience, and a multitude of data sources is used to build models on various scales. Various types of flow simulations are performed on many of these model scales. An important part of these simulation studies is to use some kind of upscaling or homogenization technique to derive and compute effective properties

that enable the simulator to account for sub-scale heterogeneities that are left unresolved in the model in an averaged sense. Model upscaling is unfortunately a time-consuming and error-prone process that often tends to introduce non-systematic bias in simulation results.

The reason why it has proved so difficult to develop robust and reliable upscaling methods is that flow in porous rock formations does not exhibit clear scale separation: the heterogeneous petrophysical parameters do not have a clearly defined small scale that one can safely average over. Around the turn of the century, a new and alternative approach to treating problems without scale separation started to appear in the literature, see e.g., [5, 7, 10, 20, 21]. These so-called *multiscale simulation methods* use a two-level approach to simulation, in which heterogeneity in petrophysical parameters and variations in phase compositions and saturations are represented on a fine-scale grid, whereas the mechanisms that affect the pressure depletion and drive global flow patterns are resolved on a coarser grid having fewer degrees of freedom. This setup is not very different from that of traditional upscaling, but whereas one in flow-based upscaling uses the solution of localized flow problems to determine effective parameters to be used in the coarse-scale model, multiscale methods keep the localized flow solutions and use them as numerical prolongation operators that can be used to systematically and correctly account for the effect of sub-scale variations when forming discretized flow equations on a coarser scale.

Initially, the development of multiscale methods held a lot of promise. However, despite many years of research, these methods have not yet found widespread use in industry. A reason may be that the methods proposed so far have either assumed simplified flow physics or idealized reservoir geometries, and no method has yet been able to incorporate the full model complexity needed for real petroleum assets. Among the many different multiscale methods found in the literature, the multiscale finite-volume (MsFV) method [21] is the one that has come the longest way in incorporating industry-relevant flow physics, such as black-oil models [13, 25, 29], fluid compressibility [16, 33, 50, 58], and (simplified) compositional formulations [17], all on rectilinear grids. The MsFV method, and its somewhat less accurate but more stable and versatile two-point counterpart [37], have been extended to stratigraphic and unstructured grids [38, 47], but these extensions and modifications have only been studied for incompressible flow models. On the other hand, the multiscale mixed finite-element (MsMFE) method [10] is the method that has come the longest way in incorporating realistic descriptions of reservoir geology, see [1–3, 42, 46]. However, except for the somewhat idealized black-oil and compressible flow cases studied in [26, 27, 31, 46], the MsMFE method has so far only been demonstrated to work well for incompressible flow.

Recently, we proposed a new multiscale finite-volume formulation, called the multiscale restrictedsmoothed basis (MsRSB) method [40] that resolves the combined challenge of complex grids and highly heterogeneous rock properties, as seen in models of real assets, in a simpler and more robust manner than previous methods. The method is formulated using the algebraic iterative framework that has been developed for the MsFV method [35, 56, 58, 59] and hence inherits this method's ability to treat complex flow physics, but uses a very different construction of numerical prolongation operators that mimics an approach that has proved to be quite successful in algebraic multigrid methods [9, 52–54]. Starting from a constant pressure value inside each coarse grid block, a localized iterative scheme is applied directly to the fine-scale discretization to compute a numerical prolongation operator that is consistent with the elliptic differential operator within a localized support region. The prolongation operator has good accuracy compared with alternative types of operators [36] and gives a very robust method that is not afflicted by the monotonicity issues that ail the MsFV method, see e.g., the discussion in [37, 38]. Equally important, the support regions are simple to construct for almost arbitrarily complex grids, and as a result, the MsRSB method is at least as flexible with respect to fine and coarse grid types as the MsMFE method. The method can, for instance, easily utilize various approaches for improving accuracy by adapting coarse grid

blocks to geological features, see e.g., [32].

Herein, we extend the MsRSB method to also account for realistic flow physics in the form of the compressible black-oil equations that describe the evolution of three components, water and two hydrocarbon pseudo components, referred to as oil and gas, respectively. The system of mass or volume conservation equations for these three components describes the interplay among buoyancy, capillary, and viscous forces and has an intricate mixture of elliptic and hyperbolic characteristics, which explains why these equations are generally difficult to solve efficiently and accurately. Whereas it is possible to utilize multiscale methods as part of a constrained-pressure-residual preconditioner to a fully implicit discretization [13], it is more efficient to use MsRSB to solve a separate pressure equation as part of sequential solution procedure that, if needed, can be used to formulate an iterated fully implicit discretization. To validate the MsRSB method for reservoir simulation, we present a number of test cases that contain examples of geological complexity and the type of flow physics that will be encountered in simulation of real assets. All numerical experiments are conducted using simulation tools from a new multiscale module in the open-source MRST software framework [28, 30, 31]; the module was first published as part of the 2015a MRST release and can be freely used under the GNU Public License version 3. The extensions to black-oil models discussed herein, will be published in a future release alongside of this article. In two forthcoming papers, we also describe the extension of the MsRSB method to polymer flooding including shear-thinning effects [19] and fractured media [49].

2 The black-oil model

The fluid model most commonly used for reservoir simulation is the so-called black-oil equations, in which water is modeled as a component and the various hydrocarbon chemical species are lumped together to form two components at surface conditions, a heavy hydrocarbon component called "oil" and a light hydrocarbon component called "gas". The chemical composition of each of these pseudo-components is assumed to be constant for all times. At reservoir conditions, the three components may form three phases, an aqueous phase consisting mainly of the water component, a liquid hydrocarbon phase consisting of mainly of oil and dissolved gas, and a hydrocarbon vapor phase consisting mainly of natural gas. In the general case, oil can also be dissolved in the gas phase, the hydrocarbon components are allowed to be dissolved in the aqueous water phase, and the water component may be dissolved in the two hydrocarbon phases. Herein, however, we assume that the hydrocarbon components do not dissolve in the water phase and that oil does not dissolve in gas. We will primarily look at two special models: (i) a two-phase, dead-oil model in which the oil does not contain any dissolved gas, and (ii) a three-phase, live-oil/dry-gas model in which the gas component may be partially or completely dissolved in the oil phase.

By convention, the black-oil equations are formulated as conservation of gas, oil, and water volumes at standard (surface) conditions rather than conservation of the corresponding component masses [51]. To this end, we will employ a simple PVT model that uses pressure-dependent functions to relate fluid volumes at reservoir and surface conditions. Specifically, we use the so-called inverse formation-volume factor, $b_{\alpha} = V_{\alpha}^{s}/V_{\alpha}$ (where V_{α} and V_{α}^{s} are volumes occupied by a bulk of component α at reservoir and surface conditions) and the gas solubility factor $r_{so} = V_{g}^{s}/V_{o}^{s}$, which is the volume of gas, measured at standard conditions, that at reservoir conditions is dissolved in a unit of stock-tank oil. The conservation equations then read

$$\partial_t(\phi b_o s_o) + \nabla \cdot (b_o \vec{v}_o) - b_o q_o = 0, \tag{1a}$$

$$\partial_t (\phi b_w s_w) + \nabla \cdot (b_w \vec{v}_w) - b_w q_w = 0, \tag{1b}$$

$$\partial_t [\phi(b_g s_g + b_o r_{so} s_o)] + \nabla \cdot (b_g \vec{v}_g + b_o r_{so} \vec{v}_o) - (b_g q_g + b_o r_{so} q_o) = 0.$$
(1c)

Here, ϕ is the pore volume and s_{α} denotes the fluid saturation (volume fraction) of phase $\alpha = w, o, g$. The phase velocities \vec{v}_{α} are given by a multiphase extension of Darcy's law,

$$\vec{v}_{\alpha} = -\frac{k_{r\alpha}\mathbf{K}}{\mu_{\alpha}} \big(\nabla p_{\alpha} - \rho_{\alpha}g\nabla z\big),\tag{2}$$

where p_{α} is the phase pressure, $k_{r\alpha}$ is the relative permeability, and μ_{α} the viscosity of phase α and g is the gravity acceleration. The gas is miscible in oil and the gas-oil ratio r_{so} for a saturated oil is a function of pressure. The formation volume factor and the viscosity of oil depend on both pressure and the gas-oil ratio. The model consisting of (1) and (2) has more unknowns than equations. We must therefore impose a closure relationship for the saturations, $\sum_{\alpha} s_{\alpha} = 1$, and two relationships for the phase pressures, $p_o - p_w = p_{cow}$ and $p_g - p_o = p_{cgo}$, where the capillary pressures p_{cow} and p_{cqo} are known functions of the phase saturations.

Commercial simulators predominantly rely on a fully implicit discretization to solve the nonlinear system (1). Following the discretization, the discrete nonlinear system is linearized and then solved using a direct solver like nested factorization. To obtain higher efficiency for large models, it is common to replace the direct solver by an iterative solver with a constrained-pressure residual (CPR) preconditioner, see e.g., [15, 55]. This preconditioner relies on a two-step procedure in which one first forms a near-elliptic pressure equation that can be efficiently solved, e.g., by an algebraic multigrid method, and then uses an incomplete LU-decomposition to perform a variable update for the full system. Whereas it is possible to form a multiscale method for the elliptic part of the CPR system [12], this approach is in our experience not very efficient. Instead, we will rely on a sequential solution procedure that separates the black-oil equations into a pressure equation describing fluid pressures, fluxes, and well rates and a system of transport equations that describe the evolution of saturations and/or dissolution factors.

3 Sequential splitting

Several sequential methods can be found in the literature and vary in their choice of primary unknowns and the manipulations, linearization, temporal and spatial discretization, and the order in which these operations are applied to derive a set of discrete equations. A classical approach is the implicit pressure, explicit saturation (IMPES) method, which starts by a temporal discretization of the balance equations (1) and then eliminates the saturations at the next timestep to derive a pressure equation. The pressure equation is solved implicitly to obtain pressure and fluxes, which are then subsequently employed to update the volumes (or saturations) in an explicit time step. Improved stability can be obtained by a sequential implicit method [57] that also treats the saturation equation implicitly.

The general strategy for formulating a sequential implicit method for the black-oil equations is well described in the literature. Nonetheless, our experience indicates that obtaining a robust and efficient implementation hinges on a number of inconspicuous details. In the following, we therefore describe in detail the discrete equations we have used to formulate our multiscale black-oil solver and implement it in the open-source software MRST [28].

3.1 The pressure equation

To derive a pressure equation from (1) we start by discretizing in time using the backward Euler method. This gives us the semi-discrete equations,

$$\mathcal{R}_w = \frac{1}{\Delta t} \left[(\phi \, b_w s_w)^{n+1} - (\phi \, b_w s_w)^n \right] + \nabla \cdot (b_w \vec{v}_w)^{n+1} - (b_w q_w)^{n+1} = 0, \tag{3a}$$

$$\mathcal{R}_{o} = \frac{1}{\Delta t} \left[(\phi \, b_{o} s_{o})^{n+1} - (\phi \, b_{o} s_{o})^{n} \right] + \nabla \cdot (b_{o} \vec{v}_{o})^{n+1} - (b_{o} q_{o})^{n+1} = 0, \tag{3b}$$

$$\mathcal{R}_{g} = \frac{1}{\Delta t} \Big[(\phi \, b_{g} s_{g} + \phi \, r_{so} b_{o} s_{o})^{n+1} - (\phi \, b_{g} s_{g} + \phi \, r_{so} b_{o} s_{o})^{n} \Big]$$

$$+ \nabla \cdot (b_{g} \vec{v}_{g} + b_{o} r_{so} \vec{v}_{o})^{n+1} - (b_{g} q_{g} + b_{o} r_{so} q_{o})^{n+1} = 0.$$
(3c)

Next, we assume that there exist factors β_{α} so that we can obtain an equation on the form,

$$\mathcal{R}_p = \beta_w \mathcal{R}_w + \beta_o \mathcal{R}_o + \beta_g \mathcal{R}_g = 0, \tag{4}$$

where \mathcal{R}_p does not depend on the saturations at the next time step in the accumulation term. For the black-oil model it is easy to show that these factors are given as

$$\beta_w = \frac{1}{b_w^{n+1}}, \qquad \beta_o = \frac{1}{b_o^{n+1}} - \frac{r_{so}^{n+1}}{b_g^{n+1}}, \qquad \beta_g = \frac{1}{b_g^{n+1}}, \tag{5}$$

so that the saturations at the next time step can be eliminated by the closure equation $\sum_{\alpha} s_{\alpha} = 1$.

To form a fully discrete system, we also need to introduce a spatial discretization. Here, we use a standard two-point finite-volume scheme extended with upstream weighting of all terms that depend on saturation, but we could equally well have used a multipoint flux-approximation for the spatial discretization. Assuming a general polyhedral grid that consists of n_c cells and n_f faces, we start by defining two mappings. The first, $\Gamma(c)$, maps a cell to the set of faces that delimit the cell. The second brings you from a given face f to the two cells $C_1(f)$ and $C_2(f)$ that lie on opposite sides of the face. (For exterior faces, either C_1 or C_2 is zero). We can now define discrete counterparts of the continuous divergence and gradient operators. The div operator is a linear mapping from faces to cells. Let v[f] denote a discrete flux over face f with orientation from cell $C_1(f)$ to cell $C_2(f)$. The divergence of this flux restricted to cell c is then given as

$$\operatorname{div}(\boldsymbol{v})[c] = \sum_{f \in f(c)} \operatorname{sgn}(f) \boldsymbol{v}[f], \qquad \operatorname{sgn}(f) = \begin{cases} 1, & \text{if } c = C_1(f), \\ -1, & \text{if } c = C_2(f). \end{cases}$$
(6)

Likewise, the grad operator maps from cell pairs to faces. Restricted to face f it is defined as

$$\operatorname{grad}(\boldsymbol{p})[f] = \boldsymbol{p}[C_2(f)] - \boldsymbol{p}[C_1(f)].$$
(7)

In addition, we need to define the *transmissibilities*. To this end, consider a face f between two cells $i = C_1(f)$ and $k = C_2(f)$ and let $A_{i,k}$ denote the area of the face, $\vec{n}_{i,k}$ the normal to this face, and $\vec{c}_{i,k}$ the vector from the centroid of cell i to the centroid of the face. Then, the transmissibility is defined as

$$T[f] = \left[T_{i,k}^{-1} + T_{k,i}^{-1}\right]^{-1}, \qquad T_{i,k} = A_{i,k} \mathbf{K}_i \frac{\dot{c}_{i,k} \cdot \dot{n}_{i,k}}{|\vec{c}_{i,k}|^2},\tag{8}$$

where K_i is the permeability tensor in cell *i*. Using these operators and dropping subscripts for brevity, the fully discrete version of the water equation reads

$$\frac{1}{\Delta t} \Big(\boldsymbol{\phi}[c] \, \boldsymbol{b}[c] \, \boldsymbol{s}[c] \Big)^{n+1} - \frac{1}{\Delta t} \Big(\boldsymbol{\phi}[c] \, \boldsymbol{b}[c] \, \boldsymbol{s}[c] \Big)^n + \operatorname{div}(\boldsymbol{v})[c]^{n+1} - \big(\boldsymbol{b}[c] \boldsymbol{q}[c] \big)^{n+1} = 0, \\
\boldsymbol{v}[f] = -\operatorname{upw}(\boldsymbol{b}\boldsymbol{\lambda})[f] \boldsymbol{T}[f] \Big(\operatorname{grad} \boldsymbol{p}[f] - g \operatorname{avg}(\boldsymbol{\rho})[f] \operatorname{grad}(\boldsymbol{z})[f] \Big).$$
(9)

Here, $\lambda = k_r/\mu$ denotes the phase mobility and we have introduced two additional operators that compute the arithmetic average and the phase-upwind values at a face

$$\operatorname{avg}(\boldsymbol{h})[f] = \frac{1}{2} \left(\boldsymbol{h}[C_1(f)] + \boldsymbol{h}[C_2(f)] \right)$$
(10a)

$$upw(\boldsymbol{h})[f] = \begin{cases} \boldsymbol{h}[C_1(f)], & \text{if } grad(\boldsymbol{p})[f] - g avg(\boldsymbol{\rho})[f]grad(\boldsymbol{z})[f] > 0, \\ \boldsymbol{h}[C_2(f)], & \text{otherwise.} \end{cases}$$
(10b)

The fully discrete residuals can now be added to eliminate the saturations in the same way as in the semi-continuous case giving an equation $\mathcal{R}_p = 0$ on the same form as in (4).

3.2 Well models

In the following, we will assume that the source terms q_{α} in (3) only represent injection or production wells. Each well w is modelled as a set of grid cells $c \in C_w$, called connections, along the well trajectory where fluids flow from reservoir to well or vice versa. This flow takes place on a subgrid scale and is modeled using a semi-analytical Peaceman model that expresses the relation between the reservoir pressure $\mathbf{p}[c]$ in a connection cell c, the total flux from well to reservoir inside this connection, and the pressure inside the wellbore $\mathbf{p}_{w}[c]$,

$$\boldsymbol{q}[c] = \boldsymbol{\lambda}[c] \, \boldsymbol{W}_{\boldsymbol{I}}[c] \big(\boldsymbol{p}_{w}[c] - \boldsymbol{p}[c] \big). \tag{11}$$

Here, λ is the total mobility in cell c, whereas the well index W_I accounts for rock properties and geometric factors affecting the flow. The flow inside the wellbore is modeled as instantaneous, so that mass entering a well can instantaneously be recovered at the surface. Moreover, the flow is assumed to be in hydrostatic equilibrium, so that the pressure at each completion can be computed as a hydrostatic pressure drop from a datum point called the bottom-hole, i.e., $p_w[c] = p_{bh} + \Delta p[c]$. The multiscale method is not affected by how these pressure drops are computed, and details are therefore omitted for brevity. The volumetric source term for each phase depends on whether the connection is injecting or producing, that is on the sign of $p_{bh} - p[c] + \Delta p[c]$. That is,

$$\boldsymbol{q}_{\alpha}[c] = \begin{cases} \boldsymbol{F}_{\alpha}[c] \, \boldsymbol{q}[c], & \text{if } c \text{ is producing,} \\ \boldsymbol{\nu}_{\alpha}[c] \, \boldsymbol{q}[c], & \text{if } c \text{ is injecting.} \end{cases}$$
(12)

If all connections of a well are injecting, ν_{α} is simply the phase-fraction of the injected fluid, whereas if a well contains cross-flow, ν_{α} depends on the source terms of all connections that are producing fluids. The examples discussed later in the paper only have injectors without cross-flow and hence we do not describe in detail how the general case is computed in our code.

Wells can either be controlled by their bottom-hole pressure or by their surface rate. In our code, these controls are supplied in terms of extra control equations, which on residual form read

$$(\mathcal{R}_{q,\alpha})_w = (q^s_\alpha)_w - \sum_{c \in C_w} \boldsymbol{b}_\alpha[c] \boldsymbol{q}_\alpha[c] = 0$$
(13a)

$$(\mathcal{R}_C)_w = p_{bh,w} - U_{bh,w}, \quad \text{or} \quad (\mathcal{R}_C)_w = q_w^s - U_{q,w}, \tag{13b}$$

where one constant value $U_{bh,w}$ or $U_{q,w}$ is prescribe per well. Altogether, the pressure equation on residual form now reads,

$$\mathcal{R}_{\pi} = [\mathcal{R}_p, \mathcal{R}_{q,w}, \mathcal{R}_{q,o}, \mathcal{R}_{q,g}, \mathcal{R}_C]^T = 0.$$
(14)

If we choose one of the phase pressures (say p_o) as our primary variable along with the surface rates q^s_{α} and the bottom-hole pressures p_{bh} , (14) can be solved using a Newton–Raphson method for the unknown pressure state $\boldsymbol{\pi} = [\boldsymbol{p}_o, \boldsymbol{q}^s_w, \boldsymbol{q}^s_o, \boldsymbol{q}^s_g, \boldsymbol{p}_{bh}]^T$.

3.3 The transport equations

To derive transport equations for updating the fluid saturations, we start by assuming that the pressure equation has converged to a sufficient tolerance. If we define the total flux $\vec{v}_T = \sum_{\alpha} \vec{v}_{\alpha}$ at reservoir conditions based on the pressure equation and insert the resulting expression for ∇p into each phase flux we obtain

$$\boldsymbol{v}_{\alpha}[f] = F_{\alpha} \Big[\boldsymbol{v}_T + \mathbf{K} \sum_{\nu} \boldsymbol{\lambda}_{\nu}[f] \Big(\operatorname{avg}(\boldsymbol{\rho}_{\alpha})[f] - \operatorname{avg}(\boldsymbol{\rho}_{\nu})[f] \Big) g \operatorname{grad}(\boldsymbol{z}) \Big], \qquad F_{\alpha} = \frac{\lambda_{\alpha}[f]}{\sum_{\nu} \lambda_{\nu}[f]}. \tag{15}$$

Here, the densities are defined so that the oil density also accounts for the dissolved gas,

$$\rho_w = b_w \rho_w^s, \qquad \rho_o = b_o (\rho_o^s + r_{so} \rho_g^s), \qquad \rho_g = b_g \rho_g^s. \tag{16}$$

For the mobilities on the faces $\lambda_{\alpha}[f]$ we use potential ordering to determine the upstream weighting [8], which for most situations coincides with the upstream weighting used in the pressure solver. For brevity, we have tacitly assumed that capillary forces are negligible so that there is only a single unique pressure. In general, these terms appear along with the gravity contributions in the potential differences, and are also included in the implementation.

Now that we have expressions for the fluxes, we can solve two of the three conservation equations. Typically, we solve the oil and gas equations to ensure that the hydrocarbon masses are conserved throughout the simulation and let the water phase fill up the remaining pore volume. In the classical IMPES method, the saturations are updated using a single step of an explicit scheme. Herein, we use an implicit solver to allow for longer time steps. That is, we use a Newton–Raphson method for (3b) and (3c) with fluxes provided by (15) to obtain saturations for the next time step. If the oil is undersaturated, there is no free gas and hence we solve for r_{so} rather than solving for s_g . Henceforth, we let $\boldsymbol{\sigma} = [\boldsymbol{s}_o, \boldsymbol{s}_g \vee \boldsymbol{r}_{so}]^T$, where the second half of the vector is interpreted cell-wise so that the state is $\boldsymbol{s}_g[c]$ if cell c is fully saturated and $\boldsymbol{r}_{so}[c]$ otherwise.

3.4 Solving the nonlinear problem

Both the pressure and transport equations are nonlinear equations and must be solved iteratively. In both cases, the nonlinear system of discrete equations are solved using a Newton–Raphson method. For the transport equations, we could equally well have used an explicit solver, but we choose to use an implicit discretization in time to avoid stability restrictions on the time steps. Assuming that we already know the current reservoir state (π^n, σ^n) , the solution procedure for a single time step can then be written algorithmically as:

- 1. Set iteration counters ℓ and m for the pressure and saturation equations to zero and use the current state as the initial guess, that is, $\pi_0 = \pi^n$ and $\sigma_0 = \sigma^n$, respectively.
- 2. Solve the linearized pressure equation and update the pressure,

$$\boldsymbol{\pi}_{\ell+1} = \boldsymbol{\pi}_{\ell} - J_{\pi}(\boldsymbol{\pi}_{\ell}, \boldsymbol{\sigma}_m, \boldsymbol{\pi}^n, \boldsymbol{\sigma}^n)^{-1} \mathcal{R}_{\pi}(\boldsymbol{\pi}_{\ell}, \boldsymbol{\sigma}_m, \boldsymbol{\pi}^n, \boldsymbol{\sigma}^n)$$
(17)

where J_{π} refers to the Jacobian $d\mathcal{R}_{\pi}/d\pi$ of the residual equation \mathcal{R}_{π} . Check if the pressure solution has converged using the following convergence criteria,

$$\|\boldsymbol{p}_{\ell+1} - \boldsymbol{p}_{\ell}\|_{\infty} \le \varepsilon_p(\max \boldsymbol{p}_{\ell} - \min \boldsymbol{p}_{\ell}) \qquad \|\mathcal{R}_q\|_{\infty} \le \varepsilon_q, \qquad \|\mathcal{R}_C\|_{\infty} \le \varepsilon_C.$$
(18)

Set $\ell \leftarrow \ell + 1$. If the residual has not yet converged, return to Step 2 and solve another linearized system. Otherwise, compute total flux $v_{T,\ell}$ by summing Darcy's law for each phase and proceed to the next step. 3. For each cell c, compute the maximum amount of solvable gas $r^{max}[c] = r_{so}(\mathbf{p}_{\ell}[c])$ and switch variable from r_{so} to s_g if $\mathbf{r}_{so,m}[c] > r^{\max}[c]$, or vice versa. Linearize and solve the transport system with the updated pressure,

$$\boldsymbol{\sigma}_{m+1} = \boldsymbol{\sigma}_m - J_{\sigma}(\boldsymbol{\pi}_{\ell}, \boldsymbol{v}_{T,\ell}, \boldsymbol{\sigma}_m, \boldsymbol{\pi}^n, \boldsymbol{\sigma}^n)^{-1} \mathcal{R}_{\sigma}(\boldsymbol{\pi}_{\ell}, \boldsymbol{v}_{T,\ell}, \boldsymbol{\sigma}_m, \boldsymbol{\pi}^n, \boldsymbol{\sigma}^n)$$
(19)

where we again let \mathcal{R}_{σ} and J_{σ} be the residual and Jacobians of the linearized transport equations respectively. We define the convergence using both a maximum error and a total error for the phases under consideration,

$$\left\|\frac{\Delta t\mathcal{R}_{\alpha}}{\phi}\right\|_{\infty} \leq \varepsilon_V b_{\alpha}^{\text{avg}}, \qquad \frac{\|\Delta t\mathcal{R}_{\alpha}\|_1}{\|\phi\|_1} \leq \varepsilon_M b_{\alpha}^{\text{avg}}, \quad \alpha \in \{o, w, g\}.$$
(20)

Set $m \leftarrow m + 1$ Reiterate Step 3 if we have not converged, otherwise proceed to next step.

4. Optionally, check the pressure residual against a tolerance for the outer iteration,

$$\|\mathcal{R}_{\pi}(\boldsymbol{\pi}_{\ell}, \boldsymbol{\sigma}_{m}, \boldsymbol{\pi}^{n}, \boldsymbol{\sigma}^{n})\|_{\infty} \leq \varepsilon_{it}.$$
(21)

If not converged, go back to Step 2. Otherwise, the time step is finished.

4 Multiscale formulation

To develop the multiscale formulation, we will need two grids. We have already introduced a finescale grid that holds petrophysical properties and on which we have developed the discrete flow equations. This grid is denoted Ω^c and will in the general case consist of a collection of *polyhedral cells* $\{\Omega_i^c\}_{i=1}^{n_c}$ of varying shapes whose connections form an unstructured topology. On top of this grid, we define a coarse grid Ω^b that is formed as a partition of Ω^c . In other words, this coarse grid will consist of a collection of *blocks* $\{\Omega_j^b\}_{j=1}^m$ that each is formed by amalgamating a connected set of cells from the fine grid. Usually, we represent the coarse grid in terms of a partition vector that takes the value *j* in element *i* if Ω_i^c belongs to Ω_j^b . This means that each cell is assigned to one and only one block.

In the following, we will focus on developing a multiscale formulation for the pressure equation only. To accelerate the transport step, one can for instance use a domain-decomposition method as discussed by Kozlova et al. [25].

4.1 Multiscale solver for pressure

As explained above, the pressure is computed by solving a nonlinear system of discrete equations using a Newton–Raphson method. The computational cost of this solution process is dominated by the cost of inverting the pressure Jacobian, see (17). To formulate the multiscale method we start by rewriting (17) as a discrete system for the increment in pressure and drop any subscripts since it is clear that we are working with a single increment in the pressure,

$$J\Delta p = r. \tag{22}$$

The result is a standard linear system of parabolic or elliptic type, depending on the amount of compressibility in the system. Since (22) potentially contains large variations in coefficients and a large degree of coupling between neighboring cells, we want to avoid solving it directly. Instead, we seek to approximate the pressure by a multiscale expansion that will enable us to formulate a

reduced system that is associated with the coarse grid Ω^b and hence can be inverted much more efficiently.

First, we assume that there exists a coarse-scale increment Δp_c and a prolongation matrix P the enables us to expand Δp_c to a vector that is defined over the fine grid and approximates the true fine-scale increment,

$$\Delta \boldsymbol{p} \approx \Delta \boldsymbol{p}_f = \boldsymbol{P} \Delta \boldsymbol{p}_c. \tag{23}$$

To find the coarse increment, we insert the multiscale pressure expansion (23) into (22) and multiply from the left with a restriction matrix \mathbf{R} ,

$$RJP\Delta p_c = J_c\Delta p_c = Rr = r_c.$$
⁽²⁴⁾

A simple choice is to use a finite-volume restriction

$$(\mathbf{R}_{cv})_{ji} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in \Omega_j^b, \\ 0, & \text{otherwise,} \end{cases}$$
(25)

which can be seen as a discrete analogue to applying the divergence theorem to the total velocity on the coarse scale. This restriction operator is also used in the classical multiscale finite-volume (MsFV) method [21] and ensures that the method is mass conservative. As is well known, the combination of the finite-volume restriction operator and the MsFV prolongation operator is unfortunately not always stable and can lead to strongly non-monotone solutions that upset the convergence of the multiscale solver [56]. To accelerate the convergence, Wang et al. [56] proposed to use the Galerkin restriction operator $\mathbf{R}_G = \mathbf{P}^T$ with \mathbf{R}_{cv} wrapped at the end to get mass-conservative solutions. The same strategy can be applied with MsRSB. However, as shown in [39, 40], the prolongation operators in our new multiscale method do not suffer from instabilities and have approximately the same convergence for \mathbf{R}_{cv} and \mathbf{R}_G , and herein we therefore only report results using \mathbf{R}_{cv} .

4.1.1 Restricted smoothed basis functions

The standard approach for constructing multiscale prolongation operators P is to concatenate the solution of a series of local flow problems that all are computed numerically. These flow problems can be set up in various ways, see e.g., [14, 21, 37]. For the classical MsFV method [21], for instance, the prolongation operator is constructed by solving flow problems defined over a dual grid subject to reduced-dimensional boundary conditions. In the MsTPFA method [37], the prolongation operator is constructed by multiplying elementary functions that correspond to flow solutions used in transmissibility upscaling with a set of carefully constructed partition-of-unity functions. Recently, we proposed an alternative approach in which the prolongation operator is constructed iteratively [39, 40], hence eschewing the need for complicated setup of flow problems localized by different types of boundary conditions. This approach gives high-quality basis functions that are straightforward to construct for grids with unstructured topologies, as found in real-life reservoir models.

For compressible problems, different pressure equations have been suggested to compute the basis functions [16, 33, 58]. The main choice is whether the compressibility should be included in the basis functions directly, or if the basis functions are defined as the solution of a steady-state / incompressible problem. Herein, we have chosen to use something roughly equivalent to steady-state basis functions that can be produced directly from the Jacobians of compressible flow problems,

$$\boldsymbol{A}_{b} = \frac{1}{2} \left(\boldsymbol{J} + \boldsymbol{J}^{T} - \operatorname{diag}((\boldsymbol{J} + \boldsymbol{J}^{T})\boldsymbol{1}) \right).$$
(26)

Here, **1** denotes a column vector of ones and diag is an operator that expands a vector to a diagonal matrix of the same size. Essentially, this regularization averages the ingoing and outgoing flux over

each interface and ensures that the sum of fluxes is equal to zero for each cell. This results in a matrix with the same properties as if the pressure equation was elliptic. By forming the matrix from the compressible system matrix, the multiscale solver does not need invasive knowledge of the Jacobian assembly when implemented.

The main reason for using a steady-state formulation is that the basis functions then become decoupled from the time-dependency in the problem, making them both reusable for multiple time steps and ensuring partition of unity. In [50], the authors reached the same conclusion when focussing on iterative performance for single-phase problems.

Given a suitable matrix A_b , we can construct the basis functions as follows. Let P be a sparse matrix of size $n \times m$, where each column corresponds to the basis function for a given coarse block. We then set the initial basis function for a given coarse block to a constant value in the interior of the block and zero elsewhere, i.e., define $P^0 = R^T$, and then apply the iterative process,

$$\boldsymbol{P}^{n+1} = \boldsymbol{P}^n - \mathcal{M}(\omega \boldsymbol{D}_b^{-1} \boldsymbol{A}_b \boldsymbol{P}^n), \tag{27}$$

where D_b is the diagonal of A, ω is a relaxation factor, and \mathcal{M} is a function that modifies the increment so that each basis function is limited to a predetermined region of support called the support region. Notice, in particular, that the iteration (27) does not need to converge to obtain a good prolongation operator. The iteration is only continued until the initial discontinuities in P have been sufficiently smoothed. Figure 1 illustrates this construction for a lognormal permeability field on a simple Cartesian grid. Specific details of how the increments are modified and when to terminate the iterations are given in [39, 40].



Figure 1: Iterative construction of a single basis function using restricted smoothing. The basis function is initially set to be a constant value in the interior of the grid block and zero elsewhere. During the iterative process, the basis function is never allowed to grow beyond the support region shown in red. As a result, the iterative process produces a set of localized, smooth interpolators that form a partition of unity and only overlap within predefined support regions. For this Cartesian grid, the support region coincides with the four dual coarse blocks that define the corresponding basis function for the MsFV method. Moreover, approximately 100 Jacobi iterations are required to obtain a reasonable converged basis function.

4.1.2 Iterative multiscale

The pressure computed by the multiscale method will generally only be an approximation of the fine-scale pressure. The basis functions do not account for changes in density and mobility caused by changes in pressure, and even for linear problems, the multiscale solution will typically have some relative error because the prolongation operator is unable to capture all fine-scale details. To enable a systematic reduction of the fine-scale residual towards zero in machine precision, we introduce an iterative procedure in which the multiscale pressure approximation is iteratively corrected using a local solver or *smoother*. Herein, we use ILU(0) as our smoother as suggested in [56] to account for

both nonlinear effects and approximation errors. If the pressure increment at iteration n is x^n , we write the defect as

$$\boldsymbol{d}^n = \boldsymbol{b} - \boldsymbol{J}\boldsymbol{x}^n. \tag{28}$$

Then, if we let $y^n = U^{-1}(L^{-1}d^n)$ denote the result of applying the ILU(0) smoother to the defect with initial guess zero, we can write the next update as the previous iterate with the smoothed update added in along with a *coarse correction* which ensures that we still have a solution at the coarse scale that is suitable for reconstruction

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^n + \boldsymbol{P} \big(\boldsymbol{J}_c^{-1} \boldsymbol{R} (\boldsymbol{d}^n - \boldsymbol{J} \boldsymbol{y}^n) \big) + \boldsymbol{y}^n.$$
⁽²⁹⁾

Unless explicitly specified otherwise, we will in the following use only a single smoothing step per iteration. Moreover, to make the multiscale method work like a noninvasive black-box solver, no adaptive updates are made to the Jacobi matrix, i.e. it is used as-is at every nonlinear iteration.

4.1.3 Multiscale strategy for nonlinear problems

Going back to our nonlinear solution method from Section 3.4, the iterative form of the multiscale method can be applied during each nonlinear step to systematically reduce the linearized residual below a prescribed tolerance,

$$\frac{\|\boldsymbol{J}\boldsymbol{x}^n\|_{\infty}}{\|\boldsymbol{r}\|_{\infty}} \le \varepsilon_{ms},\tag{30}$$

so that only a few multiscale iterations are applied within each nonlinear iteration step. This way, we never solve the linearized system exactly, but rather resolve the major features of the pressure using the multiscale solver. Once the changes in pressure are sufficiently small according to (21), we reconstruct a velocity field from the approximate pressure.

In a multiphase flow simulation, the basis functions depend on mobility and will thus be time dependent. Basis functions could have been recomputed adaptively as e.g., in [22, 23]. However, the MsRSB method does not need such 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 reduces the need for tolerance-based updates and means that the cost of updating the prolongation operators is proportional to the change in fluid mobility. In our prototype implementation we have chosen to update all basis functions for simplicity.

4.2 Flux reconstruction for compressible problems

When using a multiscale solver to simulate multiphase flow, we typically want to compute an approximate fine-scale flux field that can be used to evolve saturations and/or components. This means that the flux field must be mass conservative so that it does not introduce unphysical source terms in the transport equations. For an incompressible system, mass conservation means that $\operatorname{div}(v_T)[c] = q[c]$. The multiscale solution fulfills this property on the coarse grid by construction, but if one uses the prolongated pressure to compute a fine-scale flux from Darcy's law, the resulting flow field will generally not be mass conservative away from the faces of the coarse grid. To remedy this problem, one can solve a local flow problem on each coarse block with the coarse-grid fluxes as Neumann boundary conditions to reconstruct a pressure field that in turn can be used to compute mass-conservative fluxes inside each coarse block [21, 34]. In the incompressible case, the pressure equation is linear and hence it is straightforward to compute fluxes. For compressible flow, on the other hand, greater care must be taken to define the fluxes correctly. Because the pressure equation is generally nonlinear, the fine-scale fluxes will depend on three different sets of state variables: the

reservoir and well pressures π used to evaluate fluid and rock properties, the gradient of pressure ∇p that enters Darcy's law, and saturations σ that are used to evaluate relative mobilities and capillary pressures.

We start by observing that the multiscale solution π_{ℓ} in iteration ℓ is computed based on a linearized pressure equation that is assembled using pressures $\pi_{\ell-1}$ and saturations σ_m , see (17). The same linearized Jacobian matrix is then used to compute the *reconstructed pressure* \hat{p} . Let now B be the partition vector defined so that B(i) = j if cell i lies in block j. Then the reconstructed flux \hat{v}_T on face f is given as

$$\hat{\boldsymbol{v}}_{T}[f] = \begin{cases} \boldsymbol{v}_{T}(\boldsymbol{\pi}_{\ell-1}, \nabla \hat{\boldsymbol{p}}, \boldsymbol{\sigma}_{m}), & \text{if } B(C_{1}(f)) = B(C_{2}(f)) \\ \boldsymbol{v}_{T}(\boldsymbol{\pi}_{\ell-1}, \nabla \boldsymbol{p}_{\ell}, \boldsymbol{\sigma}_{m}), & \text{if } B(C_{1}(f)) \neq B(C_{2}(f)). \end{cases}$$
(31)

We note that the pressure used to evaluate properties will lag one iteration behind to ensure consistency for the reconstructed pressure. Moreover, we point out that the computation of the fine-scale reconstruction represents a *linear* update. Previous work [33] has considered nonlinear reconstruction, where properties are based on the reconstructed pressure. A nonlinear approach could equally well have been used for MsRSB, but we chose to implement a linearized approach as the iterative strategy with a local smoother generally removes local errors efficiently.

5 Numerical results

The multiscale restricted-smoothed basis method has been implemented as part of a free, opensource multiscale module in the Matlab Reservoir Simulation Toolbox (MRST) [28, 30, 31, 41]. MRST is primary a toolbox for rapid prototyping, but also contains sequential implicit and fully implicit simulators for black-oil models that have been extensively validated and verified on standard benchmarks and against commercial reservoir simulators. MRST will be used to provide reference simulations on fine grids in the following.

The results of a series of preliminary numerical experiments reported in [39] indicated that MsRSB is robust, accurate and efficient for a wide variety of cases, ranging from incompressible, single-phase flow on 2D Cartesian grids to multiphase simulations on geomodels of real petroleum assets. In [40], we report a more comprehensive study for incompressible that verifies that the method is particularly versatile with respect to grid complexity and is capable of handling the types of stratigraphic grids seen in industry today. Herein, we focus on demonstrating that the method is also highly versatile with respect to realistic flow physics seen in standard black-oil models. In two forthcoming papers, we will also show applications to EOR (polymer flooding) and simulation of fractured media [19, 49].

As part of our analysis, we also consider the gain in computational efficiency by replacing a finescale solver by an approximate multiscale solver. This analysis, should, however, only be considered as a preliminary and conservative indication for the following reasons: All solvers from MRST are implemented using automatic differentiation with operator overloading [28], and accordingly there will be a significant overhead in evaluation of fluid properties, implicit and dynamic assembly of Jacobians, and so on. These parts of the simulators are not written for speed, but to provide great flexibility in changing fluid models and (nonlinear) solution strategies. The corresponding computational cost is therefore singled out and reported as a separate category called 'other', as we expect that it can be significantly reduced if the simulators are reimplemented in a compiled language and carefully optimized to eliminate redundant property evaluations and unneeded matrix multiplications. Secondly, the linear solver used in the fine-scale simulation is an algebraic multigrid method written in Fortran [44] and is used with default settings. The multiscale solver is written in



Figure 2: Results for injection of a tracer into an ideal gas, reported at dimensionless times $\tau = 2.5 \cdot 10^{-3}, 2.5 \cdot 10^{-2}, 1.25 \cdot 10^{-1}$ and $2.5 \cdot 10^{-1}$, where $\tau = 1$ corresponds to a completely tracer-filled reservoir.

Matlab, with basis functions computed using a straightforward C-accelerated implementation. The sequential transport solver uses GMRES preconditioned with ILU(0). Time steps in all simulations are chosen automatically by the simulators according to industry-standard chopping procedures [4]. In the simulations presented in the following, MsRSB was used as a black-box solver and no attempt was made to tweak its settings (coarsening ratio, update of basis functions, combination of tolerances, etc) to optimize efficiency.

5.1 Injection of a ideal gas tracer

As a first validation, we employ a conceptual test case of compressible flow that was originally proposed in [33]. The setup considers the injection of a tracer into an ideal gas, which can be described as a two-phase problem in which both phases have the same properties as well as linear relative permeabilities. For an ideal gas, we will get a linear compressibility and consequently $b_g \propto p$. Gas is injected into the left end of a 1D homogeneous reservoir at a pressure of 1 MPa at x = 0. The initial pressure is set to 0.1 MPa and kept fixed at the right end of the domain, giving a density change of 10. As in [33], the reservoir is represented using five coarse blocks and one hundred fine cells. Figure 2 reports the pressure and tracer concentrations at the same dimensionless times considered in the original test case. Even if the linear tolerance is set to the relatively large value of 0.1, the multiscale solver shows an excellent match with the fine-scale solution.

5.2 SPE10: horizontal layers

Over the past years, use of petrophysical data sampled from Model 2 of the 10th SPE Comparative Solution Project [11] has emerged as a de facto test case for validating multiscale methods. The model consists of parts of a Brent sequence, whose dimensions were slightly rescaled to form a challenging benchmark for upscaling methods. The model contains two formations: The Tarbert formation at the top of the model represents a prograding near-shore formation, which even though it exhibits orders of magnitude variations in the permeabilities, is relatively smooth and tends to be well resolved by most multiscale methods. The Upper Ness formation in the lower part of the model is fluvial and has proved to be more challenging, in particular for finite-volume type multiscale methods that rely on some form of pressure extrapolation for localization, see e.g., [38, 48, 56] and references therein.

In [40], we verified that the MsRSB method can compute accurate pressure approximations for the SPE10 model for incompressible flow. Later in the paper, we will consider simulations of compressible flow for the full 3D model. Here, however, we start by considering a simple 2D test case sampled from Layer 15 to demonstrate the ability of MsRSB to solve compressible problems and assess the effects of the flux reconstruction. To this end, we set up a quarter five-spot problem and inject one pore volume over 25 years. The displaced phase has a constant compressibility of 0.001/bar, which causes the formation-volume factor to change from 0.93 to 1.52 over the injection period, thus making the problem highly compressible compared with conditions usually observed in (offshore) reservoirs. To isolate and only study the effect of pressure changes, we assume that the injected and the displaced fluids have the same viscosity and linear relative permeability curves so that the basis functions are independent of the saturation distribution.

We consider three different multiscale solution runs with increment tolerances ε_p set to 1, 0.1 and 0.001 for the nonlinear pressure updates, see (18). The tolerance on the linear updates is set to $\varepsilon_{ms} = 0.1$ in (30) and ILU(0) is used as the second-stage solver. The reference solution is computed using $\varepsilon_p = 0.001$ and a direct solver for the pressure updates. The well curves can be seen in Figure 3. We note that the solutions with reconstructed velocity fields agree very well with the reference regardless of the tolerance. However, using the same solvers without flux reconstruction results in large deviations from the reference solution, as observed previously in [18]. From the plot of the final residual after the transport solver has either converged or reached the maximum number of iterations, we can clearly see that without reconstruction, the multiscale solver struggles to converge to high accuracy. Plots of the saturation distribution at the end of simulation in Figure 4 show that with reconstruction, the multiscale solution is qualitatively correct even for the coarse tolerance of $\varepsilon_p = 1$. Conversely, if reconstruction is not used, unacceptable mass-balance errors are introduced in the form of increased saturation values ahead of the displacement front. These results suggest that flux reconstruction is required even for problems with significant compressibility.

5.3 SPE10: water-flooding of the full 3D model

Next, we simulate the full SPE10 benchmark case as described in [11]. The model is a two-phase oil/water model with small amounts of compressibility. Since the model contains very large permeability contrasts and the pressure updates are almost global in nature because of the low compressibility, the model is a challenging benchmark for a pressure solver.

Figure 5 reports well curves along with timing results and the model setup. The multiscale solver used a fixed tolerance of 0.01 for the linear solver and 0.005 as the pressure increment tolerance, while the fine-scale sequential solver used as reference had a tolerance of 10^{-6} for the multigrid solver and 0.001 for the pressure increments. Even though the multiscale solver has coarser tolerances than the fine-scale solver, the resulting well curves are in close agreement. We note that even with more relaxed tolerances, the multiscale solver is able to take the same time steps. The difference between the reference and the multiscale solver is in this case much smaller than the difference reported in [11] among various simulators using the same formulation.



Figure 3: Well curve and final residual comparison of the multiscale solvers with and without flux reconstruction.



Figure 4: Comparison of saturation fronts for a multiscale solver with tolerance of $\varepsilon_p = 1$. Without reconstruction, unacceptable errors in fluid distribution appear in the transport solver.



Figure 5: Simulation results for the full SPE10



Figure 6: Petrophysical data for the Norne field model, shown with the mockup well positions used for validating the MsRSB method.

5.4 Norne field case: water-flooding

Norne is an oil field located in the Norwegian Sea operated by Statoil. The reservoir consists of good quality Jurassic sandstones, where oil is mainly found in the Ile and Tofte Formations and gas in the Not Formation. Production started in 1997, with water injection as the main drive mechanism, but with gas injected until 2005. The Norne field case [43] is an open benchmark that offers seismic and production data of the full field up to 2006. This includes, in particular, a reservoir simulation model of the field history in Eclipse format until December 2006. To illuminate and illustrate salient features of our multiscale model, we will use the grid and petrophysical data from this model in combination with mockup wells and fluid models.

The reservoir is represented as a $46 \times 112 \times 22$ corner-point grid and consists of two disconnected rock formations that are separated by a full layer of inactive cells. The reservoir contains several faults and partially eroded layers. When the corner-point grid is turned into an unstructured matching polyhedral grid, 75% of the 44 420 active cells have six faces while the remaining 25% cells have from five to twenty-one faces. Cell volumes vary a factor 420, whereas face areas vary by more than six orders of magnitude. In the following, we will study multiscale simulation of an initially oil filled reservoir being displaced by water. To make sure that the displacement fronts interact with all the geological complexities that can be found in the model, we set up a completely artificial well pattern consisting of three producers and four injectors, see Figure 6. The injectors operate at a fixed pressure of 500 bar, while the producers operate at a constant rate which ensures that one pore volume will be drained over a production period of 100 years. All wells are vertical and completed in all layers and are the only means of communication between the top three layers and the rest of the model.

The rock is slightly compressible and the oil and water phase are assumed to have constant compressibility so that the inverse formation volume factors are on the form,

$$b_{\alpha} = b_{\alpha}(p_0)e^{(p-p_0)c_{\alpha}} \tag{32}$$

where p_0 is a reference pressure and c_{α} a non-negative compressibility factor for phase α . The injected water has a compressibility $c_w = 10^{-5} \text{ bar}^{-1}$, while the oil is assumed to either be weakly compressible with $c_o = 5 \cdot 10^{-4} \text{ bar}^{-1}$ or strongly compressible with $c_o = 5 \cdot 10^{-3} \text{ bar}^{-1}$. We simulate the displacement process for 100 years to ensure that a steady state is reached.

In our multiscale simulations, we use the same setup as in [40], with a coarse grid that is partitioned into 250 blocks by Metis [24] configured with the logarithm of the fine-scale transmissibilities as weights in the edge-cut minimization algorithm. This way, Metis will seek to generate coarse



Figure 7: Well responses for the synthetic Norne water-flooding scenario. Results from the sequential fine-scale solver are shown as thin solid lines, and results from the corresponding multiscale simulation are shown as thicker dashed lines.

blocks so that they avoid crossing large permeability contrasts. The resulting fully unstructured partition is used without modification in the MsRSB solver. Figure 7 reports well responses obtained by a sequential fine-scale solver and the multiscale solver with tolerance of $\varepsilon = 0.01$ for the nonlinear pressure equation, see (18). The match in production rates for oil and water is perfect in the visual norm both for the slightly and the highly compressible case.

In these sequential simulations, we used a tolerance $\varepsilon_{it} = \infty$ for the outer iterations, see (21). However, the multiscale solver can also be set to match a fully-implicit solution if we set ε_{it} to correspond to the tolerance used by the nonlinear solver in the fully-implicit fine-scale simulator. The resulting solvers will be referred to as MsRSB-Seq and MsRSB-FImp, respectively. Figure 8 confirms this behavior for the slightly compressible case: Without an outer iteration, the discrepancy between the multiscale simulation and a fully-implicit, industry-standard simulation is dominated by errors caused by the sequential simulation and these errors can be systematically reduced by increasing the number of outer iterations.

Having verified that the multiscale solver can be set up to be as accurate as the formulation it tries to mimic, the natural question is to as how efficient the solver is. Figure 9 reports a comparison of iteration count and computational cost for the four simulations compared in Figure 8. Starting



Figure 8: Discrepancy in water production for production well P3 between a fully-implicit reference solution and a sequential fine-scale solution, a multiscale sequential solution, and a multiscale iterative-implicit solution for the slightly compressible scenario.



Figure 9: Iteration count and computational cost for the weakly compressible Norne model.

with the two fine-scale solvers, we see that the sequential formulation uses approximately one third of the iterations required by the fully-implicit method. The reason is that whereas the sequential fine-scale simulator only solves the pressure and transport equations once per time step and hence has no guarantee on the size of the overall residual at the end of each time step, the fully-implicit simulator needs more iterations to reduce this residual below the prescribed tolerance of 10^{-6} . If we disregard evaluation of fluid properties and assembly of fine-scale Jacobians, which admittedly is inefficient in our Matlab implementation, the sequential solver is 5–6 times faster than the fullyimplicit solver. The two fine-scale simulators use the same algebraic multigrid solver (AGMG) [44] for linear algebra, but in the fully-implicit simulator AGMG is used in combination with a CPR preconditioner [15]. Without outer iterations, the multiscale solver uses almost exactly the same number of iterations as the fine-scale sequential solver but is 2–3 times faster than AGMG. With outer iterations, the multiscale method requires the same number of iterations as the fully-implicit solver and has a solver cost that is almost the same as for the sequential fine-scale simulation. If we also include the cost of property evaluations and fine-scale assembly, we see that the computational overhead is significantly higher for the multiscale methods. Altogether, the two sequential simulations come out with more or less the same computational cost, whereas the iterated-implicit multiscale simulation is *slower* than the fully-implicit simulation. The increased computational overhead is an artifact of our prototype implementation. To ensure flexibility when experimenting with various ways of reconstructing a conservative flux field, the fine-scale fluid properties are re-evaluated three times during reconstruction: once to extract the fine-scale Jacobians, once to compute fine-scale fluxes, and once to set correct coarse-scale fluxes. These redundant property evaluations should be relatively simple to eliminate if the multiscale method is reimplemented and optimized with respect to performance rather than prototyping flexibility. Likewise, a significant amount of the overhead seen in the fine-scale simulations can be eliminated if one implements hand-coded Jacobians and/or optimizes the automatic differentiation library in MRST to reduce the number of costly sparse-matrix products.

5.5 Modified SPE1: three phases with solution gas

The first SPE Comparative Solution Project was initiated by Aziz S. Odeh [45] in 1981 to provide an independent comparison of methods and simulators for a three-layered reservoir that is initially filled with an undersaturated oil, with gas injection into the upper layer. All pertinent data were supplied for a black-oil model with nonlinear relative permeabilities, pressure-dependent viscosity, and gas dissolution. Except for the layering, the original setup does not contain any heterogeneities or structural complexity. To create a more challenging test case, we replace the original reservoir geometry with a synthetic corner-point model that is part of an anticline structure and contains four intersecting and partially sealing faults with modest throws. The reservoir covers an 3×3 km² area, is approximately 30 meters thick and is represented on a $50 \times 50 \times 10$ corner-point grid with just over 20,000 active cells. The cells have a regular hexahedral shape. Approximately 8% of the cells lie next to a fault and have more than six faces, and the faces on the faults have six orders of variation in geometric area. The permeability is lognormal (see Figure 10) and slightly anisotropic with the vertical component equal one half of the lateral components.

We place an injector to the far west and a producer at the far east. The injector injects gas for five years into the undersaturated and mostly oil filled reservoir with a fixed rate corresponding to 2% of the pore volume per day at surface conditions, which is approximately 4000 m^3 per day at reservoir conditions. The producer operates at a fixed bottom-hole pressure. Our primary concern is to test the multiscale solver on a model with realistic flow physics, including significant compressibility, gravity, and gas dissolution. We therefore simulate the production up to a short period after gas breaks through in the producer. To verify our method, we compare the multiscale simulations with a fine-scale simulation based on a sequential splitting, which is set up so that it produces solutions that are close to those obtained by a fully-implicit method. Figure 10 shows the pressure, gas saturation, and solution gas ratio at the final step of the sequential simulation. Notice in particular, the increased gas solution near the faults, where the gas that moves through the upper layer will accumulate before it is forced down and can continue to migrating in the topmost layer on the opposite side of the fault.

For the multiscale solver, we set up a simple partition consisting of $10 \times 10 \times 5$ fine cells per coarse block. This gives a coarse grid with regular geometry and a relatively simple topology in which 60% of the blocks have six neighbors, 35% have seven neighbors, 4.2% have eight neighbors, and 0.8% have nine neighbors. Because the areal circumference of the reservoir is curved, the block volumes vary a factor 27, compared to a factor 1.5 variation for the cell volumes in the fine grid. The variation in face areas, on the other hand, has been reduced to a factor 500. We then run the same





(a) The horizontal permeability for the faulted gas injection case. The vertical permeability is reduced by a factor 10.

(b) Pressure in the final timestep after gas breakthrough.



less than 0.1 gas saturation are not plotted.

(d) R_s after final timestep

Figure 10: Simulation of the modified SPE1

simulation three times with increasingly stricter tolerances. Plots of the resulting well responses are shown in Figure 11. Because the bottom-hole pressure is held fixed in the producer as the gas breaks through, there is a significant spike in the production that might not have appeared in a physical system having more realistic well controls. However, this spike serves us well to illuminate differences in the multiscale approximations. With a tolerance of 0.1, which results in only a twothree iterations on average per linear solve and two-three nonlinear iterations per time step, we see a delay of approximately one month in the pressure peak. Because both the compressibility and the dissolution model have significant pressure dependencies, the gas front is sensitive to the accuracy of the pressure approximation. For instance, if the pressure is higher than the reference value, some of the injected gas will remain in the oil phase, resulting in differences in mobility. By setting a stricter tolerance of 10^{-3} , the multiscale method is able to correctly predict the pressure spike, although the actual pressure values are still slightly off during the spike. Notice that whereas the number of linear iterations increases, the average number of nonlinear iterations decreases because of the improved quality of each linear solve. With a tolerance of 10^{-6} we obtain that is more or less identical throughout the whole simulation history.

5.6 Watt field: two-phase water flooding

The Watt field case published by the Heriot-Watt University is based on a mixture of synthetic and real data from a North Sea oil field to "describe a realistic field example seen through appraisal



Figure 11: Well responses, iteration count, and computational cost for simulations of the modified SPE1 case.



Figure 12: Porosity and well locations (producers in red, injectors in blue) for the Watt field. the fine-scale model has $256 \times 59 \times 40$ corner-point cells out of which 415711 are active.



Figure 13: Relative permeability curves and distribution of rock types for the Watt field.

into the early development life stage" [6]. The benchmark was formulated to study how uncertainty in interpretation is integrated through a reservoir modeling workflow. As part of the benchmark, a number of different model realizations were generated to represent the typical uncertainties that you encounter in a geomodelling workflow (e.g., where to place faults, cut offs for porosities and permeabilities, topology of the top and base of the reservoir surface, etc.). In this example, we consider a simulation model developed from one of these realizations. The reservoir spans a 12.5×2.5 km² surface area and has a thickness of approximately 190 m, much of which is below the water contact. The depositional environment for the field is a braided river system, with common facies types including fluvial channel sands, overbank fine sands, and background shales, see Figures 12 and 13. The reservoir is operated with 15 horizontal production wells located across the central parts of the reservoir and 5 horizontal and 2 vertical injectors around the edges. Altogether, this forms a simulation model that is representative of the complexities seen in models of real assets.

Figure 14 reports well responses predicted by the multiscale solver on a coarse grid with 800 blocks and solver tolerances $\varepsilon_p = 0.05$ and $\varepsilon_{it} = \infty$. The partition is generated by Metis with the logarithm of the fine-scale transmissibilities as weights in the edge-cut minimization algorithm and is used without any modifications. Compared with the sequential fine-scale solver, the multiscale solution is spot on and is, in particular, able to correctly predict the behavior when producers and injectors change control from bottom-hole pressure to fluid rates. Figure 15 reports the total number of iterations and the computational cost for the whole simulation. As we observed for the Norne model above, the multiscale pressure solver requires 10-20% more Newton iterations than



Figure 14: Well responses for the simulation of the Watt model. Results from the fine-scale simulation are shown as thin solid lines, and results from the multiscale simulation are shown as thicker dashed lines.



Figure 15: Iteration counts and computational cost for the Watt model.

the fine-scale solver to converge to the prescribed tolerance. Because we only solve to a tolerance of 0.05, the mass-conservative fluxes reconstructed on the fine grid will contain minor inaccuracies that causes a minor increase in the number of nonlinear iterations in the transport solver. However, this does not seem to adversely affect the general stability of the sequential solution procedure; only in two out of the 145 report steps, the transport solver required more than the prescribed limit of two sub-steps so that the pressure solver had to chop the control step in two. As explained for the Norne case above, the 30% increase in computational overhead (category 'other') for the multiscale solver is an artifact resulting from the way we have implemented the multiscale method to maximize prototyping flexibility. If we disregard this overhead and the cost of the transport solver, the multiscale pressure solver is 8–9 times faster than the fine-scale multigrid solver.

6 Conclusion

We have presented a novel multiscale method that can be used to accelerate the simulation of three-phase compressible flow in petroleum reservoirs. The method relies on an iterative process for constructing prolongation operators, which ensures great flexibility with respect to flow physics and grid formats. Indeed, the method is, to the best of our knowledge, the first method that can handle the complexity of geological description and fluid physics seen in contemporary reservoir simulation models in a robust manner.

Through a series of test cases, we have demonstrated that the MsRSB method can easily adapt to changing well controls and complex fluid behavior such as strong compressibility, gravity, and gas dissolution. The method is applicable to general polyhedral grids, both on the fine and the coarse scale, and can in particular be set up to use automatically generated coarse partition that adapt to specific features in the geological description; see [40] for a more thorough discussion. By setting an appropriate tolerance on the outer iterations, the method can be set to mimic a sequential or a fully-implicit simulator. For the sequential formulation, our numerical experiments confirm that the computational efficiency of the method compares favorably to an algebraic multigrid solver, mainly because accurate fine-scale fluxes can be computed regardless of whether the pressure solutions have converged to machine precision or not. For the fully-implicit case, certain artifacts in our prototype implementation prevent us from making a real efficiency assessment and the technology should therefore be implemented in an industry-standard simulator before one can draw the final conclusion. To this end, an important component will be to reap the obvious potential for concurrent computing.

Finally, we point out that our prototype MsRSB simulators have been implemented in the open-source software MRST and will be published in a forthcoming release.

7 Acknowledgments

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

- 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. 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. doi:10.1007/s10596-007-9072-8.
- [3] 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.
- [4] 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.
- T. Arbogast. Implementation of a locally conservative numerical subgrid upscaling scheme for two-phase Darcy flow. Comput. Geosci., 6(3-4):453-481, 2002. ISSN 1420-0597. doi:10.1023/A:1021295215383.
- [6] 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.
- [7] I. Babuka, G. Caloz, and J. Osborn. Special finite element methods for a class of second order elliptic problems with rough coefficients. SIAM J. Numer. Anal., 31(4):945–981, 1994. doi:10.1137/0731051.
- [8] 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.
- [9] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge. Adaptive smoothed aggregation (αSA) multigrid. SIAM Review, 47(2):317–346, 2005. doi:10.1137/S1064827502418598.
- [10] Z. Chen and T. Y. Hou. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Math. Comp.*, 72:541–576, 2003. doi:10.1090/S0025-5718-02-01441-2.
- [11] 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/.
- [12] M. Cusini, A. A. Lukyanov, J. Natvig, and H. Hajibeygi. A constrained pressure residual multiscale (cpr-ms) compositional solver. In ECMOR XIV-14th European Conference on the Mathematics of Oil Recovery, Catania, Italy, 8–11 September 2014. EAGE. doi:10.3997/2214-4609.20141778.
- [13] M. Cusini, A. A. Lukyanov, J. R. Natvig, and H. Hajibeygi. Constrained pressure residual multiscale (CPR-MS) method for fully implicit simulation of multiphase flow in porous media. J. Comput. Phys., 299:472–486, 2015. doi:10.1016/j.jcp.2015.07.019.
- [14] M. M. Dehkordi and M. T. Manzari. Effects of using altered coarse grids on the implementation and computational cost of the multiscale finite volume method. Adv. Water Resour., 59(0):221–237, 2013. doi:10.1016/j.advwatres.2013.07.003.
- [15] S. Gries, K. Stüben, G. L. Brown, D. Chen, and D. A. Collins. Preconditioning for efficiently applying algebraic multigrid in fully implicit reservoir simulations. SPE J., 19(04):726–736, 2014. doi:10.2118/163608-PA.
- [16] H. Hajibeygi and P. Jenny. Multiscale finite-volume method for parabolic problems arising from compressible multiphase flow in porous media. J. Comput. Phys, 228(14):5129–5147, 2009. doi:10.1016/j.jcp.2009.04.017.

- [17] H. Hajibeygi and H. A. Tchelepi. Compositional multiscale finite-volume formulation. SPE J., 19(2): 316–326, 2014. doi:10.2118/163664-PA.
- [18] H. Hajibeygi, S. H. Lee, and 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(4):1071–1083, 2012. doi:10.2118/141954-PA.
- [19] S. T. Hilden, O. Møyner, K.-A. Lie, and K. Bao. Multiscale simulation of polymer flooding with shear effects. *Transp. Porous Media*, 2016. in press.
- [20] T. Hou and X.-H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. J. Comput. Phys., 134:169–189, 1997. doi:10.1006/jcph.1997.5682.
- [21] 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.
- [22] P. Jenny, S. H. Lee, and H. A. Tchelepi. Adaptive multiscale finite-volume method for multiphase flow and transport in porous media. *Multiscale Model. Simul.*, 3(1):50–64, 2004. doi:10.1137/030600795.
- [23] 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.
- [24] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comp., 20(1):359–392, 1998. doi:10.1137/S1064827595287997.
- [25] A. Kozlova, Z. Li, J. R. Natvig, S. Watanabe, Y. Zhou, K. Bratvedt, and S. H. Lee. A real-field multiscale black-oil reservoir simulator. In SPE Reservoir Simulation Symposium, 23-25 February, Houston, Texas, USA, 2015. doi:10.2118/173226-MS. SPE 173226-MS.
- [26] S. Krogstad, K.-A. Lie, H. M. Nilsen, J. R. Natvig, B. Skaflestad, and 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. doi:10.2118/118993-MS.
- [27] S. Krogstad, K.-A. Lie, and B. Skaflestad. Mixed multiscale methods for compressible flow. In Proceedings of ECMOR XIII–13th European Conference on the Mathematics of Oil Recovery, Biarritz, France, 10–13 September 2012. EAGE. doi:10.3997/2214-4609.20143240.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] K.-A. 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. doi:10.1007/s10596-011-9244-4.
- [32] K.-A. Lie, J. R. Natvig, S. Krogstad, Y. Yang, and X.-H. Wu. Grid adaptation for the Dirichlet– Neumann representation method and the multiscale mixed finite-element method. *Comput. Geosci.*, 18 (3):357–372, 2014. doi:10.1007/s10596-013-9397-4.

- [33] 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.
- [34] 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.
- [35] 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.
- [36] 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. doi:10.3997/2214-4609.20141775.
- [37] O. Møyner and K.-A. Lie. A multiscale two-point flux-approximation method. J. Comput. Phys., 275: 273–293, 2014. doi:10.1016/j.jcp.2014.07.003.
- [38] O. Møyner and K.-A. Lie. The multiscale finite-volume method on stratigraphic grids. SPE J., 19(5): 816–831, 2014. doi:10.2118/163649-PA.
- [39] 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.
- [40] O. Møyner and K.-A. Lie. A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids. J. Comput. Phys., 304:46–71, 2016. doi:10.1016/j.jcp.2015.10.010.
- [41] MRST. The MATLAB Reservoir Simulation Toolbox, 2015a, May 2015. http://www.sintef.no/MRST/.
- [42] J. R. Natvig, B. Skaflestad, F. Bratvedt, K. Bratvedt, K.-A. Lie, V. Laptev, and S. K. Khataniar. Multiscale mimetic solvers for efficient streamline simulation of fractured reservoirs. SPE J., 16(4): 880–888, 2011. doi:10.2018/119132-PA.
- [43] Norne. IO Center Norne Benchmark Case, June 2012. URL http://www.ipt.ntnu.no/~norne/wiki/ doku.php. http://www.ipt.ntnu.no/~norne.
- [44] Y. Notay. An aggregation-based algebraic multigrid method. *Electron. Trans. Numer. Anal.*, 37:123–140, 2010.
- [45] A. S. Odeh. Comparison of solutions to a three-dimensional black-oil reservoir simulation problem. J. Petrol. Techn., 33(1):13–25, 1981. doi:10.2118/9723-PA.
- [46] M. Pal, S. Lamine, K.-A. Lie, and S. Krogstad. Validation of the multiscale mixed finite-element method. Int. J. Numer. Meth. Fluids, 77(4):206–223, 2015. doi:10.1002/fld.3978.
- [47] 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 11-13*, Stanford, California, USA, 2013.
- [48] A. Sandvin, E. Keilegavlen, and J. M. Nordbotten. Auxiliary variables for 3d multiscale simulations in heterogeneous porous media. J. Comput. Phys, 238:141–153, 2013. doi:10.1016/j.jcp.2012.12.016.
- [49] S. Shah, O. Møyner, M. Tene, K.-A. Lie, and H. Hajibeygi. The multiscale restriction smoothed basis method for fractured porous media. J. Comput. Phys., 2016. in review.
- [50] M. Tene, Y. Wang, and H. Hajibeygi. Adaptive algebraic multiscale solver for compressible flow in heterogeneous porous media. J. Comput. Phys., 300:679–694, 2015. doi:10.1016/j.jcp.2015.08.009.

- [51] 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.
- [52] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid on unstructured meshes. Technical Report 34, University of Colorado at Denver, Denver, CO, USA, 1994.
- [53] 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.
- [54] 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.
- [55] 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.
- [56] Y. Wang, H. Hajibeygi, and H. A. Tchelepi. Algebraic multiscale solver for flow in heterogeneous porous media. J. Comput. Phys., 259:284–303, 2014. doi:10.1016/j.jcp.2013.11.024.
- [57] J. W. Watts. A compositional formulation of the pressure and saturation equations. SPE J., 1(3): 243–252, 1986.
- [58] 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.
- [59] H. Zhou and H. A. Tchelepi. Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models. SPE J., 17(2):523–539, 2012. doi:10.2118/141473-PA.