

Fast sequential implicit porous media flow simulations using multiscale finite elements and reordering of cells for solution of nonlinear transport equation

Jørg E. Aarnes, Stein Krogstad, Knut–Andreas Lie, and Jostein R. Natvig

SINTEF ICT, Dept. Applied Math., P.O. Box 124 Blindern, N-0314 Oslo, Norway

Abstract

It is demonstrated previously in the literature that *multiscale methods* can be used to provide accurate high-resolution velocity fields at a low computational cost. However, to achieve enhanced accuracy in flow simulations compared with a standard approach, the multiscale method must be accompanied by a transport solver that can account for the fine-scale structures of the velocity fields. In this paper, we use the standard implicit single-point upwind (SPU) finite-volume method for computing transport. This method requires that a nonlinear system is solved at each time-step. However, if we assume (as in streamline methods) that capillary forces can be disregarded and that gravity can be treated by operator splitting, and reordering the cells in an optimal way, the nonlinear systems in each implicit advective step can be solved on a cell-by-cell (or block-by-block) basis. This approach makes the standard SPU method at least as fast as a streamline method, even on geo-cellular models with multimillion cells, and alleviates many limitations that streamline methods have. In particular, the method is mass conservative, compressibility can be handled in a straightforward manner, and pressure can be updated frequently without severely influencing the computational efficiency. By combining this transport solver with a multiscale pressure solver, we obtain a very efficient solution method capable of direct simulation of geo-cellular models with multimillion cells within an acceptable time-frame on a single desktop computer.

1. Introduction

Reliable modeling of flow in porous media is challenging, and techniques that are available today do not always produce accurate results. Researchers are therefore continually trying to develop better and more reliable simulation methods. In this paper we will focus on how to incorporate fine-scale features from a detailed geo-cellular model into flow simulations on a reservoir scale. Rather than performing flow simulations on coarsened geological models obtained by an upscaling procedure, we will do flow simulations directly on the geo-cellular model. To this end we will use so-called *multiscale methods*, which is a recent and promising technology that can be used to provide high-resolution velocity fields at a low computational cost.

To achieve enhanced accuracy in flow simulations compared with a standard approach, the multiscale method must be accompanied by a transport solver that can account for the fine-scale structures of the velocity fields. In previous studies (Aarnes et al., 2005; Gautier et al., 1999), *streamline methods* have been used in conjunction with multiscale methods to perform simulations directly on geo-cellular models. Although streamline methods are capable of including quite complex physics and have shown to be useful in many applications (see e.g., Thiele, 2001), the overall usefulness of the methodology is (unfortunately) still disputed.

In this paper, we will instead base our transport solver on the standard implicit single-point upwind (SPU) finite-volume method. In each time-step of this method one must generally solve a nonlinear $n \times n$ system, where n denotes the number of grid cells (with nonzero volume). The method is therefore generally considered to be unsuited for simulation studies of large geo-cellular models. To speed up the transport solver, we assume (as in streamline methods) that capillary forces can be disregarded and that gravity can be treated by operator splitting. The remaining advective transport is then unidirectional along streamlines. By using an optimal reordering of the cells, the nonlinear systems in each implicit advective step can then be solved in $\mathcal{O}(n)$ operations, thereby making the standard SPU method at least as fast as a streamline method even on geo-cellular models with multimillion cells. The new solver alleviates many limitations that streamline methods have. In particular, the method is mass conservative, compressibility can be handled in a straightforward manner, and pressure can be updated frequently without severely influencing the computational efficiency. Moreover, the current methodology can easily be embedded in a sequential fully-implicit formulation. By combining this transport solver with a multiscale pressure solver, we obtain a very efficient solution method capable of direct simulation of geo-cellular models with multimillion cells within an acceptable time-frame on a single desktop computer.

Next we introduce the equations for immiscible and incompressible two-phase flow. In Section 3 we present the multiscale mixed finite-element method (Aarnes et al., 2006a) that is used to compute flow velocities. In Section 4 we describe the discretization of the transport equation and how to solve the corresponding nonlinear systems in $\mathcal{O}(n)$ operations. Numerical results are presented in Section 5 to demonstrate the performance of the methodology, and, finally, we make some concluding remarks in Section 6.

2. The Two-Phase Model

We consider immiscible and incompressible flow of two phases (oil and water) in the absence of capillary pressure. To simplify the presentation, we neglect gravity and assume no-flow boundary conditions. The flow equations can be formulated as an elliptic equation for the pressure p and the Darcy velocity v ,

$$v = -\lambda(S)K\nabla p, \quad \nabla \cdot v = q. \quad (1)$$

Here q is a source term representing injection and production wells, K is the rock permeability (i.e., the ability to transmit fluids), and $\lambda = \lambda_o + \lambda_w$ denotes the total mobility. The mobility of phase α is given by $\lambda_\alpha = k_\alpha^r / \mu_\alpha$, where μ_α is viscosity of phase α and $k_\alpha^r = k_\alpha^r(S)$ is the relative permeability, i.e., the reduced ability of the rock to transmit fluids due to the presence of other phases. The second primary

unknown is the water saturation S , which denotes the volume fraction of water and is described by the transport equation

$$\phi \partial_t S + v \cdot \nabla f(S) = q_s, \quad (2)$$

where ϕ is the rock porosity and $f = \lambda_w / (\lambda_o + \lambda_w)$ the fractional flow function. The permeability tensor K and the porosity ϕ are given by the geo-cellular model and typically have a (strong) heterogeneous spatial structure with several orders of magnitude variations.

The system of equations (1)–(2) will be solved using a sequential splitting, i.e., the pressure equation is solved at the current time-step using saturation values from the previous time-step.

3. The Multiscale Pressure Solver

Let $\Omega \subset \mathbb{R}^d$ and define n as the outward-pointing unit normal on $\partial\Omega$. Define $H_0^{1,\text{div}}(\Omega)$ as the space of all functions $v \in (L^2(\Omega))^d$ such that $\nabla \cdot v \in L^2(\Omega)$ and $v \cdot n = 0$ on $\partial\Omega$. In a mixed formulation of (1), we seek a pair $(v, p) \in U \times V$, where U and V are finite-dimensional subspaces of $H_0^{1,\text{div}}(\Omega)$ and $L^2(\Omega)$, respectively, such that

$$\int_{\Omega} v \cdot (\lambda K)^{-1} u \, dx - \int_{\Omega} p \nabla \cdot u \, dx = 0 \quad \text{for all } u \in U, \quad (3)$$

$$\int_{\Omega} l \nabla \cdot v \, dx = \int_{\Omega} q l \, dx \quad \text{for all } l \in V. \quad (4)$$

To determine p , one must add an additional constraint such as $\int_{\Omega} p = 0$.

In a standard discretization, U and V typically consist of low-order piecewise polynomials. As a rule, solving (3)–(4) directly on the geo-cellular grid is prohibitively expensive. The spaces U and V are therefore typically defined on a coarser grid, onto which the permeability K must be upscaled (averaged).

Multiscale Basis Functions In the multiscale mixed finite-element method (MsMFEM) introduced by Chen and Hou (2003), the basis functions of V are given as solutions to local flow problems inside the coarse grid blocks. This way V embodies the impact of subgrid variations in K and gives a mass-conservative discretization of the velocity on a coarse grid. Below we outline a variant of MsMFEM due to Aarnes et al. (2006a), which in addition provides mass-conservative velocity fields on the underlying subgrid.

Let $\mathcal{T} = \{T_i\}$ be a grid in which each grid block T_i is a connected union of grid cells in an underlying subgrid $\mathcal{K} = \{K_j\}$ (typically the geo-cellular grid on which K and ϕ are defined). Henceforth, we will call \mathcal{T} the *coarse grid* and the subgrid \mathcal{K} will be called the *fine grid*. The approximation space for pressure consists of piecewise constant functions on \mathcal{T} , i.e.,

$$U = \text{span}\{u \in L^2(\Omega) : u|_T \text{ is constant for all } T \in \mathcal{T}\}.$$

In the approximation space V for velocity v , a basis function ψ_{ij} is assigned to each non-degenerate interface $\Gamma_{ij} = \partial T_i \cap \partial T_j$ in the coarse grid. This basis function is supported in $\Omega_{ij} = T_i \cup \Gamma_{ij} \cup T_j$, and is related to a function ϕ_{ij} through Darcy's law: $\psi_{ij} = -\lambda K \nabla \phi_{ij}$. The functions ϕ_{ij} and ψ_{ij} are obtained by solving (numerically with a mass-conservative method) the following local elliptic problem:

$$\psi_{ij} \cdot n_{ij} = 0 \text{ on } \partial\Omega_{ij}, \quad \nabla \cdot \psi_{ij} = \begin{cases} f_i(x) / \int_{T_i} f_i(x) dx & \text{for } x \in T_i, \\ -f_j(x) / \int_{T_j} f_j(x) dx & \text{for } x \in T_j. \end{cases} \quad (5)$$

Here n_{ij} is the outward pointing unit normal on $\partial\Omega_{ij}$, and

$$f_i = \begin{cases} f & \text{if } \int_{T_i} f \, dx \neq 0, \\ \lambda \text{ trace}(K) & \text{else.} \end{cases} \quad (6)$$

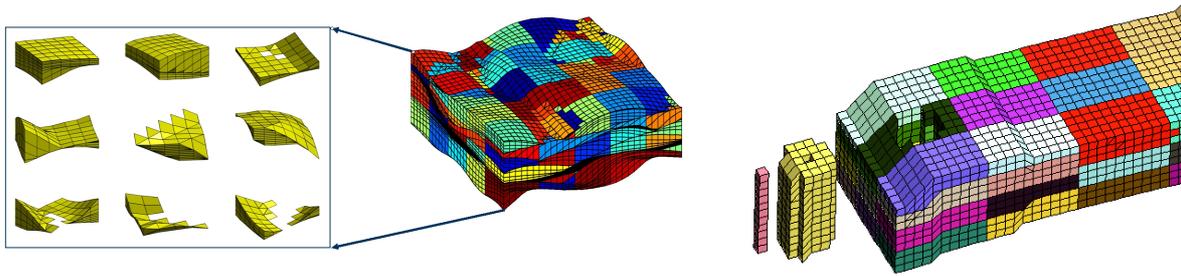


Figure 1: Examples of coarse grids and a few corresponding coarse blocks that can arise when partitioning a corner-point grid (or pillar grid) uniformly in index space. Left: a depositional bed with eroded layers. Right: a sector model with basis functions adapted to well paths.

The MsMFE approximation space V for velocity is the span of the basis functions $\{\psi_{ij}\}$.

Observe that the MsMFE basis functions force, by definition, unit flux across the associated coarse-grid interface. This implies that the MsMFE solution $\{v_{ij}\}$ for velocity gives the fluxes across the respective coarse-grid interfaces. Moreover, note that by using special source terms in blocks with a source allows the method to model radial flow around point or line sources, such as wells in oil-reservoirs, on the subgrid scale. Finally, by letting f_i scale according to the permeability as in (6), one can to some extent avoid unnaturally high flow-velocities in low-permeable fine-grid cells (Aarnes et al., 2006a).

Computational Efficiency If the pressure equation (1) is to be solved only once, the computational cost of MsMFEM is not significantly lower than for solving the full problem on the fine grid with a very efficient linear solver. However, for two-phase flow simulations, (1) needs to be solved multiple times due to dynamic changes in the mobility λ . In this setting, MsMFEM can give substantial computational savings since the basis functions do not need to be recomputed at each time-step. Thus, the computation of basis functions, which is the most expensive task in MsMFEM, becomes part of a preprocessing step. Hence, MsMFEM becomes analogous to single-phase upscaling methods, see e.g., (Christie, 1996) and references therein. The cost of computing a full set of basis functions is also comparable to a standard single-phase flow upscaling procedure. We should add, however, that numerical experiments show that if saturation profiles exhibit sharp fronts, a slight improvement in accuracy can be obtained by regenerating basis functions in regions where the saturation has changed substantially since the previous time-step (Aarnes, 2004). Similar observations were made by Jenny et al. (2003) for a finite-volume version of the multiscale method.

Flexibility with Respect to Grids Computational efficiency and the prospect of obtaining flow fields with fine-scale details have so far been the driving forces in the development of multiscale methods for porous media flow. In our opinion, the enhanced flexibility with respect to choice of simulation grids may be at least equally important. Indeed, since MsMFEM provides mass-conservative velocity field, users may choose grids for flow transport simulations in a nearly seamless fashion; grid cells in the simulation grid need only consist of a connected collection of cells in the fine grid \mathcal{K} . This appealing feature applies, in principle, also for MsMFEM. A bit simplified, this can be stated as follows: given an appropriate pressure solver for the local problems (on the geo-cellular grid or refinements thereof), the multiscale method can be formulated and basis functions can be computed on (almost) any coarse grid where each grid block consists of an arbitrary connected collection of fine-grid cells. In particular, MsMFEM allows users to tune the coarse grid to well trajectories, and heterogeneous structures such as fractures and faults without having to resort to re-sampling procedures. For standard pressure solvers, creating coarse grids for real-field geological models can be a non-trivial task. By using MsMFEM to model pressure, this task is significantly simplified, as is demonstrated in (Aarnes et al., 2006a,b). Figure 1 shows two examples

of typical coarse grids defined on top of an underlying corner-point grid.

4. The Fast Transport Solver

For the transport solver, we introduce a new grid $\mathcal{G} = \{G_i\}$, where we once again assume that each grid cell consists of a connected collection of cells in \mathcal{K} . Furthermore, let $\{\gamma_{ij} = \partial G_i \cap \partial G_j\}$ denote the non-degenerate faces in \mathcal{G} and let v_{ij} be the corresponding flux that we obtain when solving (1) with MsMFEM. If $\mathcal{G} \neq \mathcal{T}$, subgrid fluxes are sampled from the faces of \mathcal{K} using the subresolution in MsMFEM and averaged over γ_{ij} . Then the implicit upstream method for (2) reads (for $i = 1, \dots, n$)

$$S_i^{k+1} = S_i^k + \frac{\Delta t}{\int_{G_i} \phi dx} \left[\int_{G_i} q_s(S_i^{k+1}) dx - \sum_{\gamma_{ij} \subset \partial G_i} F_{ij}(S^{k+1}) \right]. \quad (7)$$

Here Δt denotes the time-step, S_i^k the net saturation in G_i at time $t = t_k$, and

$$F_{ij}(S) = f(S_i) \max\{v_{ij}, 0\} + f(S_j) \min\{v_{ij}, 0\}.$$

Since q_s and f are (in general) non-linear functions of saturation, we employ a Newton–Raphson method to solve (7). However, instead of iterating on the global $n \times n$ linear system obtained by assembling (7) for all i , we make a clever enumeration of the cells that allows us to only assemble (7) locally and perform a Newton iteration directly on a cell-by-cell (or block-by-block) basis. This will ensure that the nonlinear iteration has an optimal $\mathcal{O}(n)$ computational complexity. In linear algebra terms, the reordering corresponds to finding a permutation P for the Jacobi matrix J of the global system such that $L = PJP^T$ is a block-triangular matrix.

Reordering of Cells The key to obtaining an optimal reordering comes from the observation that the flow described by (2) is unidirectional along streamlines. When using an upwind discretization as in (7), the flow (of information) will similarly be unidirectional in the directed graph spanned by the discrete fluxes v_{ij} . To see this, we rewrite (7) as

$$S_i^{k+1} = S_i^k + Q_i(S_i^{k+1}) - F_i^+(S_i^{k+1}) + F_i^-(S_j^{k+1}; j \in U(i)),$$

where F_i^+ and F_i^- are terms representing flow out of and into cell G_i and $U(i) = \{j \mid v_{ij} < 0\}$ denotes the indexes of the neighboring cells on the upstream side of G_i . In a cell G_i penetrated by an injection well, $U(i) = \emptyset$ and the saturation can thus be computed by solving (7) locally using a Newton method. Having found S_i^{k+1} , we can immediately update the saturation in all cells having G_i as their only “inflow neighbor” (i.e., in all G_j for which $U(j) \equiv \{i\}$) and so on. This process can be continued if we are able to find a sequence of cells $p = (p_1, \dots, p_n)$ such that if G_{p_i} is on the upstream side of G_{p_j} , then $p_i < p_j$.

Finding such a reordering is known as a *topological sort* and is a standard operation in graph theory. We therefore view the grid as a directed graph in which each vertex corresponds to a cell and each edge corresponds to a (directed) flux. Then a reordering can be found by using a backward depth-first traversal of the grid-graph, in which each cell is visited only once (see Natvig and Lie, 2006). If sequence p cannot be found by a single depth-first traversal, then the graph has at least one cycle of strongly connected elements; that is, a set of vertexes where each vertex can reach any other vertex in the same cycle. In other words, all cells in a cycle of strongly connected elements are mutually dependent and must be solved for simultaneously. Fortunately, such cycles can be detected automatically by a forward depth-first search of the grid-graph. By lumping all elements in a cycle into a single vertex, we obtain a directed *acyclic* graph, in which we are guaranteed to find an optimal reordering. Altogether this gives an optimal sequential solution procedure for the global nonlinear system, where we in each step either solve (7) by a Newton method in a single cell or by a Newton–Raphson method in a usually small collection of cells.

The reordering of cells gives us local control over the nonlinear iteration. This will not only speed up the nonlinear iterations, but we also avoid assembling global discretization matrices, which means that we can significantly increase the size of the problems we are able to solve. However, the computational savings of the reordering scheme depends on the number of cycles of connected elements. If there are large cycles in the velocity field, we may achieve moderate computational savings. The reordering scheme is, however, no worse than a direct solution of the full nonlinear system since the reordering of grid cells is relatively inexpensive.

The number and size of the cycles in the velocity field depend partly on the heterogeneity and partly on the numerical method that is used to discretize the pressure equation. For instance, if a two-point flux-approximation scheme is used, then no cycles of connected elements occur in the velocity field because the fluxes are directly related to the gradient of the pressure. However, if a mixed finite-element method is used, then streamlines may pass through and reenter cells, which creates loops of connected elements. When coupling the multiscale pressure solver with the reordering scheme, the multiscale solution can create loops of connected coarse-grid elements. As a consequence, the total number of elements that belong to a cycle is potentially much larger for a for the multiscale pressure solver than for a mixed finite-element method. Numerical experience shows that (large) cycles in mixed and multiscale pressure solvers occurs mainly for permeability fields with large scale structures that dominate flow, such as fluvial and layered permeability fields with sharp permeability contrasts. Furthermore, the size of the cycles in the multiscale pressure solution depends on the coarse grid, and may be reduced by choosing a not too coarse grid. In any case, the total number of cells in connected cycles is usually very small compared to the total number of cells in the domain.

5. Numerical Results

In this section we assume that the simulation grid coincides with the fine grid for MsMFEM, i.e., $\mathcal{G} = \mathcal{K}$. Porosity ϕ is set to be constant, and the water and oil mobilities are defined by

$$\lambda_w(S) = \frac{S^2}{\mu_w} \quad \text{and} \quad \lambda_o(S) = \frac{(1-S)^2}{\mu_o}, \quad (8)$$

where we assume $\mu_w = 5\mu_o$.

5.1. Multiscale Mixed Finite Element Method

Although more extensive numerical experiments have been performed to assess the robustness and accuracy of MsMFEM elsewhere (e.g., Aarnes et al. (2006a,b)), we present here, for completeness, a single example. The model, which is depicted in Figure 2 along with the coarse grid, is a layered model where the permeability in each layered is sampled from a spatially correlated log-normal probability distribution. The fine grid \mathcal{K} consists of 540.000 active cells, the coarse grid for MsMFEM consists of 5.400 cells, and the permeability spans across more than 7 orders of magnitude.

In our experience, it is difficult to assess the quality of the multiscale pressure and velocity solution directly, for instance by measuring how much the velocity field deviates from a reference solution on the fine grid. This is primarily due to the fact that the velocity oscillates rapidly so that even if the overall quality of the solution is good, the difference measured in a suitable norm may be large. Hence, to assess accuracy of the multiscale pressure solution, we compare instead saturation fields obtained using MsMFEM and the upstream method with saturation fields obtained by solving the pressure equation directly on the fine grid. Figure 3 displays the relative error (the L^2 -norm of the error relative to the L^2 -norm of the reference solution) and water-cut curves (fraction of water in the produced fluid) for the two producers as a function of time in PVI (fraction of total accessible pore-volume that has been injected into the domain). The plots in Figure 3 demonstrate that the overall error is small, and that MsMFEM produces water-cut curves that match closely the corresponding water-cut curves obtained by solving the pressure equation directly on the fine grid.

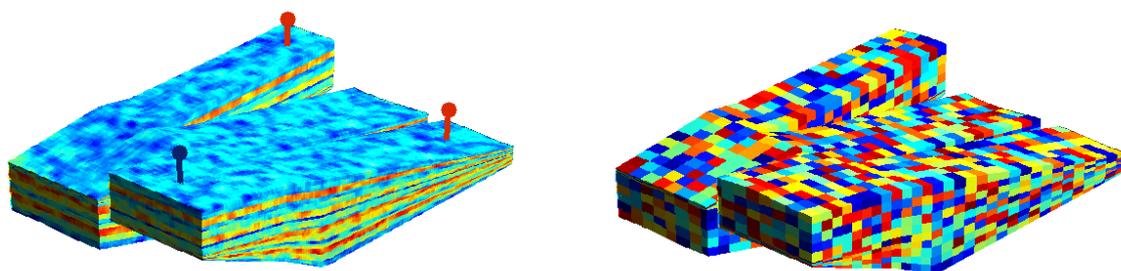


Figure 2: Right: The permeability model on the fine grid, and well locations. If we view the model as a capital M, two producers are located at the base of the M, and an injector is placed at the right top of the M. Left: Coarse grid that stems from a uniform partitioning in index space.

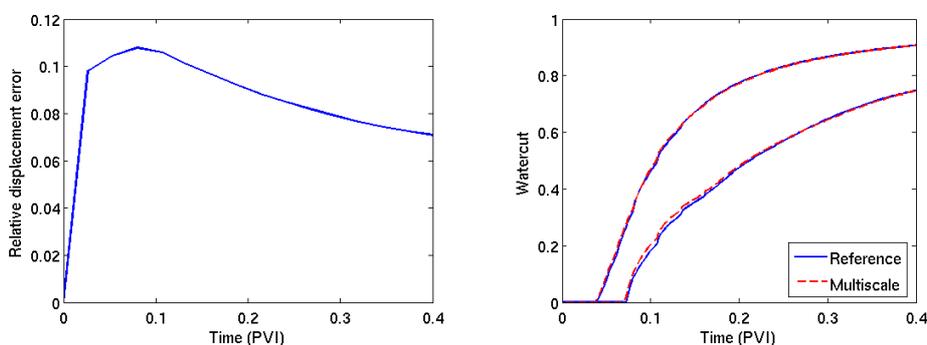


Figure 3: Right: The L^2 -norm of the saturation field error relative to the L^2 -norm of the reference solution. Left: Water-cut curves obtained by solving the pressure equation directly on the fine grid and with MsMFEM respectively.

5.2. Implicit Upstream Method with Reordering of Cells

By ordering cells in an optimal way, one can solve the nonlinear systems that arise from a discretization of (2) using the implicit upstream method in a cell-by-cell or block-by-block fashion. The efficiency of the method depends primarily on the number of cells that belong to cycles of strongly connected elements. The main purpose of this section is to demonstrate that the fraction of elements that belong to cycles of strongly connected elements is typically relatively small, and hence does not have a significant impact on the CPU-time. Moreover, we present an example of an application of the same reordering principle to compartmentalize reservoir domains into regions of flow for each injector-producer pair.

To demonstrate that cycles of strongly connected elements generally does not have a strong impact on the CPU-time, we have computed the solution of a water-flooding scenario in a sequence of permeability fields in the unit cube. The permeability is layered, with a spatially correlated log-normal distribution in each layer (see e.g., Figure 2). The simulations are performed on tetrahedral grids that are generated by splitting each hexahedral cell in a uniform N^3 Cartesian mesh into six tetrahedrons. The lowest-order Raviart-Thomas mixed finite element method is used to generate velocity fields. Moreover, for the larger grids, we employ also MsMFEM on a $10 \times 10 \times 10$ coarse grid to generate velocity fields.

On the left in Figure 5 we have plotted the average CPU-time per grid-cell per time-step for each grid. As a reference, the CPU-time per time-step for a case with unit permeability has been plotted. In the homogeneous case, no loops occur in the velocity field. For the heterogeneous cases, the CPU-time increase with the grid-size. This is partly due to increasing fraction of cells that belong to loops of strongly connected elements, and partly that the largest loops are larger, and thus more time-consuming to solve.

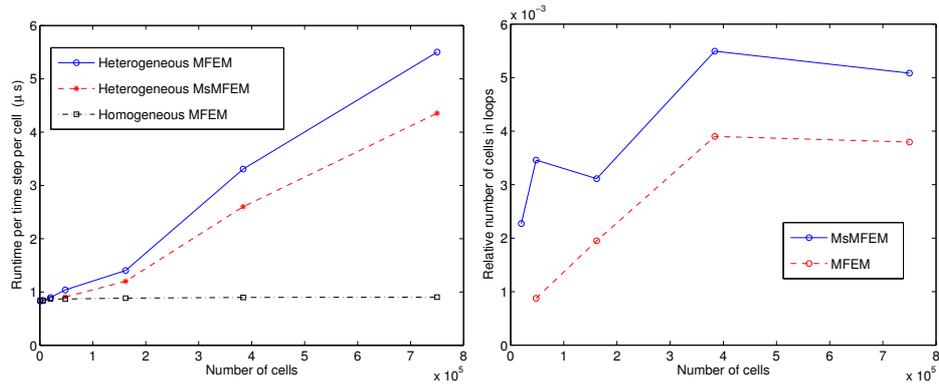


Figure 4: Left: CPU-time per time-step per grid cell for a sequence of uniform tetrahedral grids in the unit cube. Right: Fraction of cells that belong to loops of strongly connected elements in the heterogeneous cases.

However, the increase in CPU time is moderate. Indeed, the CPU-time for the homogeneous case corresponds to linear complexity, i.e., no cycles occur. For the largest case, the computation time is roughly six times larger than for the homogeneous case. We should also note that we have not made any effort to optimize solution of the small linear systems corresponding to each cycle of strongly connected elements. The fraction of grid cells that are part of a cycle of strongly connected elements for the heterogeneous case is shown on the right in Figure 4.

5.2.1. Using cell-reordering for visualization of regions of flow for injector-producer pairs

As an illustration of possible applications where reordering grid cells improve efficiency, we include an example where a stationary solution of simple tracer flow (i.e., (2) with $\partial S/\partial t = 0$ and $f(S) = S$) has been computed. The model is taken from model 2 of the tenth SPE comparative solution project (Christie and Blunt, 2001). This is a Cartesian model with an injector at the center of the domain, and a producer at each corner. To compartmentalize the domain, we reverse the flow field, i.e., replace q_s with $-q_s$ so that the injectors become producers and vice-versa. For each injector, a stationary tracer distribution has been computed with unit tracer saturation in a single injector and zero injection in the other three. In Figure 5, each of the four colored domains have been delineated by coloring the domain where $S_i \geq 0.5$, for i =red, green, blue, orange. To obtain better spatial resolution in this computation, we have used a second-order discontinuous Galerkin scheme, with the reordering scheme applied at the element level.

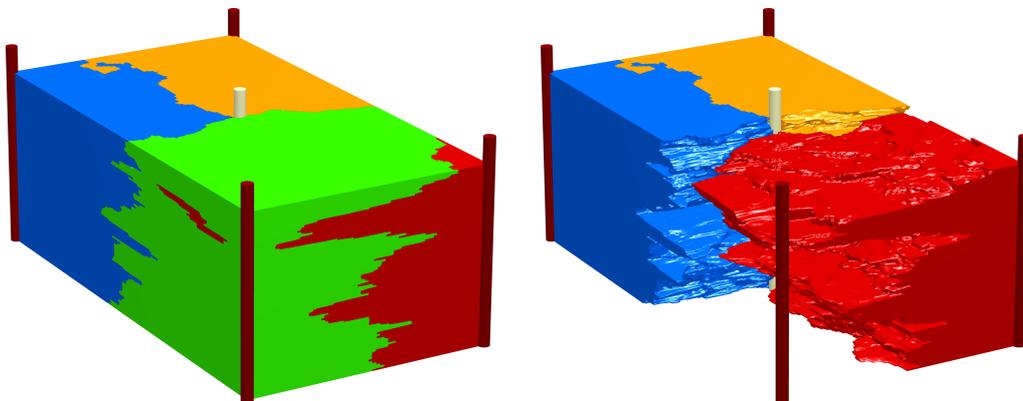


Figure 5: Delineation of the domain in model 2 from the tenth SPE comparative solution project (Christie and Blunt, 2001) into flow regions for each of four injection wells (Left) with all tracers shown and (Right) with one tracer removed to show the intricate surface separating the regions.

6. Concluding Remarks

In this paper, we consider convection dominated incompressible and immiscible two-phase flow. The multiscale mixed finite element method is used to create mass conservative velocity fields on fine grids, and an implicit upstream method is used to solve the transport equation. Although the implicit upstream method allows large time-steps, it is generally considered to be prohibitively expensive for conducting simulations on models with very large number of cells. However, here the unknowns are reordered so that the whole nonlinear system can be solved in a sequential fashion. The method has similarities with the streamline based methods, but the speed advantage of streamline methods is closely tied to the assumption that one can use an IMPES (Implicit Pressure Explicit Saturation) formulation with large time-intervals between each pressure update. This assumption may not always be justified for systems with a strong dynamic coupling between the pressure equation and the transport equations, e.g., for flow scenarios that are strongly influenced by gas. The efficiency of the alternative solver that we have presented here does not deteriorate significantly when frequent pressure updates are required, and, unlike streamline based methods, handles compressibility in a straightforward manner. By coupling this method with a multiscale transport solver, we obtain a methodology that can be used to perform simulations directly on high-resolution geological models. Moreover, although the algorithms are easy to parallelize, they also have very low memory requirements (one does not need to assemble the whole system on the fine grid), and one can therefore perform large simulations on standard single-processor desktop computers.

References

- Aarnes, J. E. (2004). 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):421–439.
- Aarnes, J. E., Kippe, V., and Lie, K. A. (2005). Mixed multiscale finite elements and streamline methods for reservoir simulation of large geomodels. *Adv. Water Resources.*, (28):257–271.
- Aarnes, J. E., Krogstad, S., and Lie, K. A. (2006a). A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform grids. *Multiscale Model. Simul.*, 5(2):337–363.
- Aarnes, J. E., Krogstad, S., Lie, K.-A., and Laptev, V. (2006b). Multiscale mixed finite-element methods on corner-point grids. *Comput. Geosci.*, *submitted*.
- Chen, Z. and Hou, T. Y. (2003). A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Math. Comp.*, 72:541–576.
- Christie, M. A. (1996). Upscaling for reservoir simulation. *JPT J. Pet. Tech.*, (48):1004–1010.
- Christie, M. A. and Blunt, M. J. (2001). Tenth SPE comparative solution project: A comparison of upscaling techniques. SPE 72469, url: www.spe.org/csp.
- Gautier, Y., Blunt, M. J., and Christie, M. A. (1999). Nested gridding and streamline-based simulation for fast reservoir performance prediction. *Comput. Geosci.*, 3:295–320.
- Jenny, P., Lee, S. H., and Tchelepi, H. A. (2003). Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *J. Comput. Phys.*, 187:47–67.
- Natvig, J. R. and Lie, K.-A. (2006). Fast computation of multiphase flow in porous media by implicit discontinuous galerkin schemes with optimal ordering of elements. <http://folk.uio.no/kalie>.
- Thiele, M. R. (2001). Streamline simulation. In 6th *International Forum on Reservoir Simulation*. Schloss Fuschl, Austria.