Vera Louise Hauge

Multiscale Methods and Flow-based Gridding for Flow and Transport in Porous Media

Thesis for the degree of Philosophiae Doctor

Trondheim, September 2010

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

NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

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

ISBN 978-82-471-2336-2 (printed ver.)
ISBN 978-82-471-2337-9 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2010:181

Printed by NTNU-trykk

Abstract

The topic of this thesis is fast and accurate simulation techniques used for simulations of flow and transport in porous media, in particular petroleum reservoirs. Fast and accurate simulation techniques are becoming increasingly important for reservoir management and development, as the geological models increase in size and level of detail and require more computational resources to be utilized.

The multiscale framework is a promising approach to facilitate simulation of detailed geological models. In contrast to traditional upscaling approaches, the multiscale methods have the detailed geological models present at all times.

The work in this thesis includes development of a multiscale-multiphysics method for naturally fractured reservoirs and a new coarsening strategy for geological models to facilitate fast and accurate transport simulations in a multiscale framework. In addition, the work comprises an application of the multiscale framework for flow and transport simulation for rate optimization loops. The coarsening strategy generates flow-based transport grids and is based on amalgamating cells from a fine model, typically the geological model, according to an indicator function. The research indicates a great potential for flexibility and scalability suitable for multi-fidelity simulators.

Contents

Part I

Part II

Preface

The following thesis is submitted for the degree of philosophiae doctor (PhD) at the Norwegian University of Science and Technology (NTNU), Trondheim, Norway. The work has been part of the project "Multiscale Simulation of Highly Heterogeneous and Fractured Reservoirs" at SINTEF ICT and Department of Mathematical Sciences at NTNU.

The thesis is organized in two parts: Part I gives a brief overview of the main concepts of flow in porous media and a short introduction to upscaling and multiscale methods. Part II includes submitted and published research papers.

Acknowledgments

First of all, I want to thank my advisor Prof. Knut–Andreas Lie for all helpful discussions, all good ideas, and for his motivating enthusiasm. His extensive insight into the world of research has been a great source of inspiration. Next, I want to thank Prof. Helge Holden, for his guidance, support and helpfulness in the course of the PhD, and to Dr. Jørg E. Aarnes for introducing me to the field of reservoir simulation and for giving me a good starting point on the path of research. I would also like to thank my co-authors, for all interesting and helpful discussions. I also owe Dr. Bård Skaflestad a great thank for his detailed answers to all kinds of questions regarding Matlab and other topics. Finally, I am grateful to the whole reservoir simulation group at SINTEF ICT, Applied Mathematics, for including me in the working group and being a source of inspiration.

My research has been funded by A/S Norske Shell and has been conducted as part of the project "Multiscale Simulation of Highly Heterogeneous and Fractured Reservoirs". The project is funded in part by A/S Norske Shell and the Research Council of Norway (Petromaks), grant number 175962/S30.

Vera Louise Hauge
Oslo, September 2010

List of Papers

Paper I: **A Multiscale Mixed Finite-Element Method for Vuggy and Naturally-Fractured Reservoirs**
Astrid Fossum Gulbransen, Vera Louise Hauge, and Knut–Andreas Lie
SPE Journal, Volume 15, Number 2, June 2010, pp. 395–403

Paper II: **Coarsening of Three-Dimensional Structured and Unstructured Grids for Subsurface Flow**
Jørg Espen Aarnes, Vera Louise Hauge, and Yalchin Efendiev
Advances in Water Resources, Volume 30, Issue 11,
November 2007, pp. 2177–2193

Paper III: **Flow-Based Coarsening for Multiscale Simulation of Transport in Porous Media**
Vera Louise Hauge, Knut–Andreas Lie, and Jostein R. Natvig
Submitted. A short version is in the Proceedings of ECMOR XII – 12th European Conference on the Mathematics of Oil Recovery, Oxford, UK, 6–9 September 2010

Paper IV: **Grid Coarsening Based on Amalgamation for Multi-Fidelity Transport Solvers**
Vera Louise Hauge, Knut–Andreas Lie, and Jostein R. Natvig
Preprint

Paper V: **Modeling of Two-Phase Flow in Fractured Porous Media on Unstructured Non-Uniformly Coarsened Grids**
Vera Louise Hauge and Jørg Espen Aarnes
Transport in Porous Media, Volume 77, Number 3,
April 2009, pp. 373–398

Paper VI: **Adjoint Multiscale Mixed Finite Elements**
Stein Krogstad, Vera Louise Hauge, and Astrid Fossum Gulbransen
Accepted for publication in SPE journal. SPE-119112-PA

Part I

Chapter 1

Introduction

Flow through porous media occurs in many different physical, biological, and man-made systems. Porous material is composed of solid matter and void pores. A simple example of a porous medium is a sponge with the solid and void continuum well perceptible for the human eye as both parts are equally dominating. In the other end of the scale, we find rocks in which the void pores are small, and in the case of dense rocks, not detectable by the human eye. More examples of porous media include biological tissues, unconsolidated soils, and natural and man-made filters. Herein, we will focus on fluid flow in subsurface rocks, which is of great interest for groundwater management, petroleum production, and subsurface storage of CO_2.

Fluid flow in porous media may be considered at different physical scales. Dominant processes and governing equations may vary with these scales. At the pore-scale (microscopic level), the fluid flow can be described by percolation-invasions theory, network models, and the Navier–Stokes equations. Capillary forces and interfacial effects may be studied at this pore-scale level. At the macroscopic level, the pores and solid material are considered as a single continuum with effective properties like porosity (percentage of void space), hydraulic and electric conductivity. Here, the fluid flow obeys Darcy's law and effects of fluid phases and heterogeneities of the continuum may be studied. Considering fluid flow in for instance subsurface reservoirs, introduces an additional field-scale. Here, the continuum level is brought up to a coarser level, describing a whole field of porous media.

In this thesis, we will mostly consider fluid flow in sedimentary rocks at the reservoir scale. More specifically, we will consider flow of oil and water in reservoirs described by Darcy's law. In addition, we consider one case of porous media with large void spaces, where the flow is best described both as Darcy flow in the porous regions and as free flow in the void spaces. In this case the flow will be modeled by the Stokes–Brinkman equations.

Field-scale geological models used in reservoir simulations are large and detailed. It is not uncommon that geological models range in size from 10 to 100 million cells. Most reservoir simulators do not have the computational capacity to process such large models, and the gap between the available level of detail in geological models and the computational capacity of commercial simulators is steadily increasing. To close this gap, two main approaches are used: Firstly, the size of geological models are reduced by upscaling and upgridding to get simulation models. Secondly, efficient flow simulators are developed by using fast and accurate numerical methods for flow in porous media.

Generally, reservoir engineers and geoscientists tend to work with different representations of the reservoir fields. The reservoir engineers work with full-field descriptions at relatively low resolution or with small local descriptions (sector models) at higher resolution. Geoscientists work with larger and more detailed descriptions. Use of a shared-earth model (a common model used throughout all steps of the modeling workflow) is desirable as this would simplify communication between experts from the different subdisciplines involved in the modeling and increase consistency of updates to the model, for instance from history matching. However, the use of a shared-earth model requires efficient and automated methods for up- and downscaling, particularly for reservoir simulation, for which automated upscaling processes would give the possibility of selecting resolution according to the need for detail and to fit computational capacity. Multi-fidelity modeling at multiple scales will thus be more feasible when automated and robust scaling methods are available.

The push for faster numerical methods is driven by the increasing model sizes and complexity, but also from the desire to perform a large number of forward simulations. Large uncertainties in reservoir modeling mean that the models are usually created using geostatistical methods and fast simulation methods are essential to evaluate the large number of equally probable geostatistical realizations or to consolidate simulation models with observed dynamic data from well tests and production history. Likewise, computerized methods for planning and production optimization require multiple forward simulations. With faster methods, results ready for evaluation are achieved faster. This in turn, results in reduced costs, more time for possible modification, and more time for evaluation of models and results.

The main focus of this thesis has been multiscale models for efficient computation of fluid transport. The motivation for this topic, is that multiscale methods for flow simulations have been proposed as a good alternative to traditional upscaling of geological models. However, equivalent multiscale methods for fluid transport have not been offered the same amount of attention.

Multiscale methods for flow simulations solve the velocity and pressure on coarse models, but at the same time incorporate fine-scale variations. They have been shown to be fast and accurate, with applicability directly on the geological models with acceptable computation time. To complement the simulation loop by using traditional upscaling for the transport equation, reduces potentially the accuracy of the total simulation results. On the other hand, continuing the transport simulation on the fine-scale will create a computational bottleneck in the simulation process. Thus, a multiscale counterpart for the transport is desirable. We have in mind a method solving the transport at a coarse scale, at the same time taking into account fine-scale information. In this thesis, we have studied a grid coarsening strategy to upgrid geological models. The strategy aims at generating coarse simulation grids that can utilize fine-scale information from multiscale flow solvers.

Next, the thesis shortly addresses an improved description of porous media with free flow in vugs, caves, faults, and fractures. A multiscale-multiphysics method is developed that solves flow problems at different scales, using different physics at the different scales. This is an example of how an efficient and accurate solver can be extended to include more physics compared to the most simplified models.

Finally, we seek to apply the combination of multiscale methods and the suggested coarsening strategy for transport to achieve fast approximate simulations in production optimization loops using adjoint based gradients. Such an application demonstrates the strength and efficiency of the suggested multiscale combination.

Chapter 2

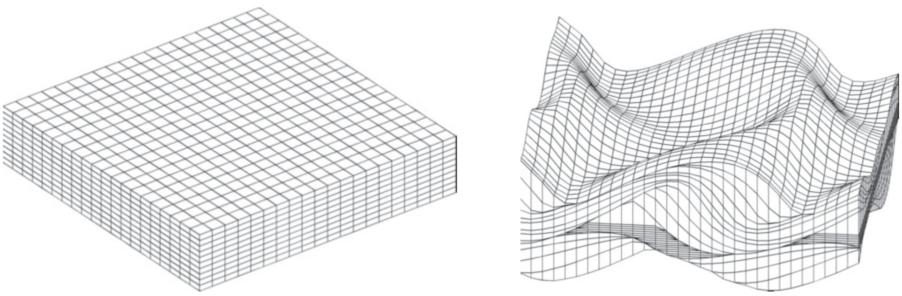
Reservoir Description

Reservoirs are composed of geological layers of sedimentary rock. Sedimentary rocks are formed through erosion, transportation, and deposition of sediments. Typical layered sedimentary rocks are sandstone and limestone. Layers may be (partially) eroded, leading to a more complex geometric description. Complex physical geometries are also introduced by faults and fractures.

Reservoirs are characterized by their geometries and petrophysical properties. The spatial structure of the geology in the reservoirs makes the geometry of reservoirs often complex and inconveniently, even inaccurately, modeled by Cartesian coordinates. Stratigraphic grids, which model the geological layered structures, will give a more accurate geometric description of the reservoir. The petrophysical properties we will cover here, include porosity and permeability. More on reservoir geology and geometry can be found in [39, 54]. General background material for reservoir simulations can be found in [5].

The porosity of a rock, in the following denoted by ϕ, is the void volume fraction of the material, with $0 \leq \phi < 1$. The porosity is an important quantity when attempting to evaluate the potential total volume of fluids that a reservoir may contain.

The permeability, usually denoted by $\mathbf{K}$, is a measure of the ability of a porous medium to transmit fluids. Absolute permeability is the ability to transmit fluid when there is only one single fluid or phase present in the porous medium. Effective permeability is the ability to transmit fluid when other immiscible fluids or phases are present in the porous medium. Relative permeability, usually denoted k_r, is then the ratio of effective permeability of a fluid to the absolute permeability.



Structured Cartesian grid

Corner-point grid

Figure 2.1: Examples of grid structures.

The permeability is strongly correlated to the porosity ϕ, since the orientation and interconnections of the pores are essential for flow. But there is generally no proportional relationship between the permeability and porosity. Permeability is described by a scalar value if the porous medium is homogeneous and isotropic, that is, if the permeability is the same in all spatial directions. In oil reservoirs, the permeability is usually both heterogeneous and anisotropic, that is, varying with each spatial position and in all spatial directions. Then permeability is described by a tensor for each reference volume. The SI unit for permeability is m^2. However, the common unit for permeability is the darcy (D) or millidarcy (mD) ($1 \text{ D} \approx 10^{-12} \text{ m}^2$).

Petrophysical properties are usually modeled using a discrete representation in terms of a volumetric grid. Shape and topology of cells represent geometry and connectivity in the reservoir. Petrophysical properties are assigned to each grid cell and are considered constant within each cell. These cells represent a representative elementary volume (REV). The simplest grid type is a structured grid with regular cells, a typical Cartesian grid or a mapping thereof. The left plot in Figure 2.1 is an example of a structured Cartesian grid. However, complex reservoir geometries might be better described by unstructured grids. Unstructured grids consist of a set of simple shapes laid out in an irregular pattern with any number of cells meeting a simple vertex.

Most common grid types in the industry are stratigraphic grids, that is, corner-point grids and 2.5D PEBI grids. A corner-point grid consists of a set of hexahedral cells that are topologically aligned in a Cartesian fashion. Cells are numbered using logical indexing and are described with eight logical corner points that are restricted by four pillars. Cells may be inactive and corners along the pillars may collapse into a single point. Thus, simple fault systems and eroded layers may be adequately represented. The right plot in Figure 2.1 shows an example of such a corner-point grid, in which layers are eroded and result in pinch-outs. Grid formats of 2.5D PEBI (perpendicular-bisector) type, are Voronoi grids constructed by extruding a 2D Voronoi grid in the vertical direction.

In reservoir simulations, the need for simple grid descriptions and the desire for grid descriptions representing complex grid features must be balanced. Simple structured grids are inflexible with respect to modeling geological features, but at the same time they are not too difficult to construct and can readily be partitioning into structured coarse grids if needed. In addition, simple numerical discretization schemes can be applied, for instance TPFA if the grid is **K**-orthogonal, that in turn produces simpler systems to solve. Fully unstructured grids are flexible with respect to modeling geological features, but may be difficult to construct and to upgrid. Such grids generally require more complex discretization schemes and are thus potentially more computationally complex and expensive. Unstructured grids may not have any simple upscaling alternatives as structured grids, and more sophisticated upgridding with subsequent upscaling of reservoir properties has to be applied. More on reservoir grids and the use in reservoir simulators can be found in [31, 53]. A later chapter will give an introduction to upscaling and upgridding.

Chapter 3

Fluid Flow

A fluid flowing in porous media is described by its density, viscosity, compressibility, pressure, and velocity. In general, the density, denoted ρ, varies with pressure and temperature. Fluid viscosity, denoted μ, is a measure of the resistance or tendency of the fluid to flow. Compressibility is the measure of how much the volume changes when a fluid is subject to changes in pressure. The simplest fluid models assume incompressibility, i.e., that the density does not change with pressure. We will in the following only consider incompressible flow in isothermal reservoirs, i.e., we neglect variations in density and temperature.

Fluids are composed of one or more components. The term component refers to a unique chemical species. A component may be present in different phases. The term phase refers to the physical state of a component. Three phases can be distinguished: solid, liquid, and gaseous phases. Several liquid phases can be present in a porous media; however, only one gaseous phase can exist. In miscible flow, the phases are free to mix and mass can be transferred from one phase to another.

We start out by describing the equations governing the simplest model of fluid flow in porous media given by the single-phase flow model.

3.1 Single-Phase Flow

The basic governing equation for the flow of a single phase in a porous medium is the conservation of mass, stated as

$$\int_\Omega \frac{\partial(\phi\rho)}{\partial t}\, dV + \int_{\partial\Omega} (\rho\vec{v}) \cdot \vec{n}\, dS = \int_\Omega q\, dV, \tag{3.1}$$

where $\vec{v}$ is the fluid velocity, q is a possible sink or source term, and Ω is a control volume in which the mass is conserved. This equation states that the change in mass of the fluid within the control volume Ω is equal to the mass flowing in and out of the control volumes as well as the accumulation or reduction caused by the source or sink term q, respectively. Applying the divergence theorem, we can restate the conservation of mass as

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla \cdot (\rho\vec{v}) = q. \tag{3.2}$$

Fluids flow quite slowly in porous media and the fluid velocity is modeled by Darcy's law:

$$\vec{v} = -\frac{\mathbf{K}}{\mu}\left(\nabla p + \rho g \nabla z\right). \tag{3.3}$$

Here, the fluid velocity $\vec{v}$ is described by the permeability (tensor) $\mathbf{K}$ of the porous medium, the pressure p, the gravitational constant g, and the spatial coordinate z in the upward vertical direction. In words, Darcy's law states that the flow velocity is proportional to a gradient of the fluid potential.

Assuming that the porosity is constant in time and that the flow is incompressible, we can simplify the mass-conservation equation (3.2) to

$$\nabla \cdot \vec{v} = \frac{q}{\rho}. \tag{3.4}$$

Combined with Darcy's law (3.3), this gives an elliptic equation

$$-\nabla \cdot \left[\frac{K}{\mu}\left(\nabla p + \rho g \nabla z\right)\right] = \frac{q}{\rho}. \tag{3.5}$$

Appropriate boundary conditions are imposed to close the model. The simplest type of boundary conditions is no-flow on the boundary. This means that we impose $\vec{v} \cdot \vec{n} = 0$, where $\vec{n}$ is the normal vector pointing outwards on the boundary $\delta\Omega$. This makes the reservoir an isolated flow system where no fluid enters or leaves. Neglecting the gravity effects, the equation is further simplified to a Poisson equation

$$-\nabla \cdot \left(\frac{K}{\mu}\nabla p\right) = \frac{q}{\rho}. \tag{3.6}$$

This fluid description is sufficient in reservoirs consisting only of porous media. However, fractures and other void spaces can introduce regions of free flow, where Darcy's law is no longer applicable. Fluid description for combined porous media and free flow will be given next.

3.2 Flow in Naturally-Fractured Reservoirs

Most natural porous media contain fractures at various scales. Small-scale fractures strongly influence the preferential flow directions and this effect can to some extent be upscaled and embedded into the permeability tensor of the model. Fractures with lengths comparable to field scale may dominate the large-scale flow pattern and should ideally be resolved in the simulation grid. However, complex geometries often make this difficult.

The traditional approach has been to consider fractured reservoirs as so-called dual-porosity reservoirs, [6, 61, 43]. The rock, often called matrix, has one porosity, while the fracture network has another higher porosity. The dual-porosity model is a conceptual model of the two overlapping reservoirs, the matrix and the fractures, with interaction. In a dual-porosity/dual-permeability model, also the permeability is specified in the two continua. Other multi-continuum models describe reservoirs where the rock formation can be separated into a set of coupled media, each representing one hydraulic component.

Some reservoir rocks can also contain relatively large void volumes or volumes that contain fluid suspensions. Karst reservoirs are characterized by vugs, caves, and fractures. Vugs are relatively large void spaces in the form of fractures, small cavities, and caves. In these void regions, fluid flow is described as free flow and is no longer dominated by Darcy's law; Figure 3.1 shows an illustration of a vuggy reservoir.

Incompressible flow in vuggy reservoirs can be described by Darcy's law in the porous regions and by Stokes equations in the free-flow regions. The Stokes equations read

$$\begin{aligned} -\mu\Delta\vec{v}_S + \nabla p_S &= f, \\ \nabla \cdot \vec{v}_S &= q. \end{aligned} \tag{3.7}$$

Here the subscript S is introduced to denote Stokes quantities. In reservoirs with both porous parts with Darcy flow and free-flow regions with Stokes flow, the model is closed with conditions on the interfaces between the Darcy and Stokes subdomains. However, the porous and free-flow regions are usually not well separated; precise geometric information of the interfaces is not easily available. Moreover, the free-flow regions appear at multiple scales. Resolving all regions on a centimeter scale would make the flow problem computationally intractable.

Stokes–Brinkman equations are a combination of Darcy's law (3.3) and the mass conservation (3.4) and Stokes' equation (3.7). Before stating the

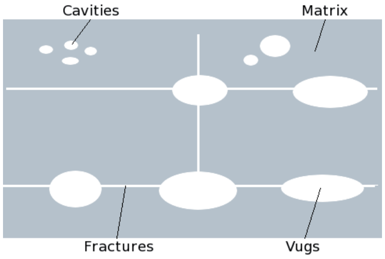


Figure 3.1: Illustration of a vuggy reservoir.

equations, we repeat the Darcy's flow model. Neglecting gravity, Darcy's flow model can be expressed as

$$\begin{aligned} \mu \mathbf{K}^{-1}\vec{v}_D + \nabla p_D &= 0, \\ \nabla \cdot \vec{v}_D &= q. \end{aligned} \tag{3.8}$$

Here, the subscript D denotes Darcy quantities. The Stokes–Brinkman model is given as

$$\begin{aligned} \mu \mathbf{K}^{-1}\vec{v} + \nabla p - \tilde{\mu}\Delta\vec{v} &= 0, \\ \nabla \cdot \vec{v} &= q. \end{aligned} \tag{3.9}$$

Here, p is the pressure, $\vec{v}$ the velocity of the fluid, $\mathbf{K}$ is the permeability equal to the Darcy permeability in the porous subdomain, μ is the viscosity of the fluid, $\tilde{\mu}$ is the effective viscosity, and q is a source term. This model can be reduced to either the Stokes or the Darcy equations with appropriate choices of parameters and gives a somewhat coarser model that does not require a precise description of the interface between free-flow and porous domains. In the free-flow domain, we may let $\mathbf{K}$ tend to infinity and set the effective viscosity equal to the fluid viscosity, $\tilde{\mu} = \mu$, such that the Stokes–Brinkman equations (3.9) simplify to the Stokes equations. If $\tilde{\mu}$ is set to zero in the porous domains, then the Stokes–Brinkman equations simplify to the coupled Darcy–Stokes equations and the need for resolving the interface conditions reappears. Setting the effective viscosity equal to the fluid viscosity, $\tilde{\mu} = \mu$, we rewrite the Stokes–Brinkman equation as

$$\nabla p = -\mu \mathbf{K}^{-1}\vec{v} + \mu\Delta\vec{v} \approx -\mu \mathbf{K}^{-1}\vec{v}. \tag{3.10}$$

Comparing the magnitude of the two velocity terms in (3.10) shows that the first term dominates the second by several orders of magnitude for typical reservoirs. Thus, this equation may be considered as Darcy's equation with a small viscosity perturbation.

Hence, using the Stokes–Brinkman model with a high permeability in the free-flow regions and using $\tilde{\mu} = \mu$ in the porous regions, gives us a unified model in vuggy reservoirs without explicit modeling of interfaces between the two types of flow domains.

3.3 Two-Phase Flow

A typical example of two-phase flow is when a fluid (usually water) is injected into wells to displace the oil. For two-phase flow, we introduce the concept of saturation. The saturation of phase i, denoted by S_i, gives the volume fraction occupied by phase i. In two-phase flow, we assume that the two phases completely fill the reservoir, that is, $S_1 + S_2 = 1$. Furthermore, the simplest two-phase flow model assumes that the two phases are immiscible, i.e., no mass transfer between the phases. Each phase is further described by its phase pressure p_i, phase velocity $\vec{v}_i$, viscosity μ_i, and density ρ_i. The simultaneous flow of the two phases causes each phase to interfere with the flow of the other; more precisely, each phase reduces the ability of the other phase to flow. This effect is described in the *effective* permeability, which is usually expressed by the product of the rock-dependent absolute permeability $\mathbf{K}$ and the saturation-dependent *relative* permeabilities, $k_{ri}(S_i)$, which are between 0 and 1.

Darcy's law extended to two phases relates the phase velocity $\vec{v}_i$ to the phase pressure p_i and phase potential by

$$\vec{v}_i = -\frac{\mathbf{K}k_{ri}}{\mu_i}(\nabla p_i - \rho_i g \nabla z). \tag{3.11}$$

The relation $k_{ri}(S_i)/\mu_i$ is denoted the phase relative mobility λ_i. The mass of both fluid phases is conserved through the mass conservation equation for each phase:

$$\frac{\partial\left(\phi\rho_i S_i\right)}{\partial t} + \nabla\cdot\left(\rho_i\vec{v}_i\right) = q_i. \tag{3.12}$$

When two or more immiscible phases are present in the reservoir, one phase wets the porous medium more than the other and is referred to as the wetting phase (subscript w). By wetting the medium we mean that the phase has the ability to maintain contact with the solid surface in the porous medium. The other phase is the non-wetting phase (subscript n). In a water-oil system, water is most often the wetting phase.

In general, there will be a difference in fluid pressures between the phases. This pressure difference is given by the capillary pressure

$$p_{cnw} = p_n - p_w. \tag{3.13}$$

Using a simple model, it is usual to either neglect the capillary effects or introduce a global pressure to reduce the coupling. A global pressure can be defined as

$$p = p_n - p_c, \tag{3.14}$$

where $\nabla p_c = \frac{\lambda_w}{\lambda_n+\lambda_w}\nabla p_{cnw}$.

We assume the two phases to be oil and water ($i = o, w$). For the simplest model, we neglect capillary effects so that $\nabla p_o = \nabla p_w$, and we further assume $p_o = p_w = p$. Neglecting the gravity effects, and combining Darcy's law (3.11) with the conservation of mass (3.12), we obtain the pressure equation:

$$\begin{aligned} \vec{v} &= -\lambda(S_w)\mathbf{K}\nabla p, \\ \nabla \cdot \vec{v} &= q, \end{aligned} \tag{3.15}$$

where $\vec{v} = \vec{v}_w + \vec{v}_o$ is the total velocity, $\lambda = \lambda_w + \lambda_o$ is the total mobility, and $q = q_w/\rho_w + q_o/\rho_o$ the total source term.

The conservation equation for water, referred to as the saturation equation, now reads

$$\phi\frac{\partial S_w}{\partial t} + \nabla \cdot (f_w\vec{v}) = \frac{q_w}{\rho_w}, \tag{3.16}$$

with water fractional flow $f_w(S_w) = \lambda_w(S_w)/\lambda(S_w)$.

The system of equations (3.15)–(3.16) is closed by applying appropriate boundary conditions, either imposing pressure or flow conditions along the boundaries.

For further details on fluid flow we refer the reader to [58, 18, 8, 2].

3.4 Numerical Discretization

The flow models presented above have to be discretized to be solved on the discrete grid representing the reservoir. To exemplify the discretization methods, we will use the single-phase flow model. It is common to discretize the pressure equation with a finite-volume method. We consider a grid cell Ω_i in the reservoir domain Ω and the integral of the pressure equation (3.4) over Ω_i:

$$\int_{\Omega_i}\left(\frac{q}{\rho} - \nabla \cdot \vec{v}\right)\, dx = 0. \tag{3.17}$$

Applying the divergence theorem and assuming that $\vec{v}$ is sufficiently smooth, transforms the integral to

$$\int_{\partial\Omega_i} \vec{v}\cdot\vec{n}\, ds = -\int_{\partial\Omega_i}\frac{\boldsymbol{K}}{\mu}\nabla p\cdot\vec{n}\, ds = \int_{\Omega_i}\frac{q}{\rho}\, dx, \tag{3.18}$$

where $\vec{n}$ in the outward pointing unit normal vector on the boundary $\partial\Omega_i$. This expression imposes mass conservation locally for each grid cell.

The simplest finite-volume method is the two-point flux approximation (TPFA) scheme. This scheme is a cell-centered finite-volume method, expressed by differences in the pressure variable for each grid cell Ω_i:

$$\sum_j T_{ij}(p_i - p_j) = \int_{\Omega_i} \frac{q}{\rho}\, dx, \tag{3.19}$$

where j loops over all neighboring cells to cell i. The transmissibilities T_{ij} are associated with cell interfaces and the expression $T_{ij}(p_i - p_j)$ is the discrete form of

$$-\int_{\partial\Omega_i \cap \partial\Omega_j} \frac{K}{\mu} \nabla p \cdot \vec{n}_{ij}\, ds. \tag{3.20}$$

The TPFA method gives correct results only on **K**-orthogonal grids. **K**-orthogonal grids are orthogonal with respect to the diagonal permeability tensors, but not with respect to the full tensor permeabilities. Thus corner-point grids for instance, cannot be used for this method. The TPFA method has still been widely implemented in commercial reservoir simulators due to its simplicity and computational speed. More on finite-volume methods can be found in [18, 1].

An alternative to finite-volume methods is the mixed finite-element method. Finite-element methods are in general accurate and robust for solving equations on complicated geometric domains, but implementation on complex grids can become quite difficult.

In a mixed finite-element method, one solves for two unknowns using two equations. We consider again the single-phase pressure equation (3.4) and the Darcy's law (3.3) (disregarding the gravity effects). This time we do not combine them into one equation, but keep them separate. We restate the system of equations in the simplified form:

$$\begin{aligned} \lambda^{-1}\vec{v} + \nabla p &= 0, \\ \nabla \cdot \vec{v} &= q, \end{aligned} \tag{3.21}$$

where $\lambda = \mathbf{K}/\mu$. We impose no-flow boundary conditions, that is $\vec{v} \cdot \vec{n} = 0$ on $\partial\Omega$. We want to find approximate solutions of the velocity $\vec{v}$ and pressure p in finite-dimensional subspaces $\mathbf{V} \subset H_0^{\mathrm{div}}(\Omega)$ and $W \subset L^2(\Omega)$. The mixed formulation thus reads: Find $(\vec{v}, p) \in \mathbf{V} \times W$ such that

$$\begin{array}{rcll} \int_\Omega \vec{v} \cdot \lambda^{-1}\vec{u}\, dx - \int_\Omega p \nabla \cdot \vec{u}\, dx &=& 0, & \forall \vec{u} \in \mathbf{V}, \\ \int_\Omega l \nabla \cdot \vec{v}\, dx &=& \int_\Omega ql\, dx, & \forall l \in W. \end{array} \tag{3.22}$$

If the subspaces $\boldsymbol{V}$ and W are properly balanced, then p and $\vec{v}$ are defined (up to an additive constant for p). A stable element pair for the mixed formulation for (single-phase) Darcy flow is the lowest-order Raviart–Thomas element. Then W equals the space of piecewise constants over the given mesh, while $\mathbf{V}$ is the space restricting the normal components of the velocity on each grid interface to be constants. General presentations of finite-element and mixed finite-element methods can be found in [10, 12, 13].

Simple finite-difference methods are direct and intuitive methods, but they are not flexible with respect to geometrically complex grids. However, a new type of high-quality finite-difference methods for pressure solvers has emerged lately. In the new methods, the discrete operators preserve or "mimic" certain important properties of the original operators, and they are therefore called mimetic finite-difference methods. Furthermore, mimetic finite-difference methods can be seen as finite-difference counterparts of mixed finite-element methods. The discretization in the mixed method is introduced in the discrete function spaces and by using numerical integration to evaluate the integrals over cell volumes and cell faces. In mimetic methods, the discretization is already introduced in the variational formulation by introducing a discrete inner product that mimics the innerproduct of the mixed method. Mimetic methods are very flexible with respect to cell geometries and easy to formulate on different grid types. For more on the mimetic finite-difference methods, the reader is referred to [15, 14].

The pressure equation for two-phase flow can be discretized with the same method as for the single-phase pressure equation. Since the pressure in two-phase flow is dependent on saturation, both the pressure and saturation equations have to be solved repeatedly in the course of simulation.

The transport equation (3.16) is usually discretized with a finite-volume method in the spatial direction, while using either explicit or implicit finite-difference discretization for the time derivative. Considering a cell Ω_i, we define S_i^n as the cell-average of the water saturation at time step n:

$$S_i^n = \frac{1}{|\Omega_i|}\int_{\Omega_i} S(x,t^n)\ dx. \tag{3.23}$$

Then the finite-volume scheme takes the form

$$\phi_i \frac{S_i^{n+1} - S_i^n}{\Delta t} + \frac{1}{|\Omega_i|}\int_{\partial\Omega_i} (f_w\vec{v})\cdot\vec{n}\ d\nu = \frac{1}{|\Omega_i|}\int_{\Omega} \frac{q_i}{\rho_i}\ dx, \tag{3.24}$$

where ϕ_i is the porosity in cell Ω_i, Δt is the time step, and q_i is the source term in cell Ω_i. Changing the boundary integral to a sum over the boundaries of the cells, gives us

$$\phi_i \frac{S_i^{n+1} - S_i^n}{\Delta t} + \frac{1}{|\Omega_i|} \sum_{j \neq i} F_{ij}(S^k) = \frac{1}{|\Omega_i|} \int_{\Omega} \frac{q_i}{\rho_i} \, dx. \tag{3.25}$$

Here, $F_{ij}(S)$ is a numerical approximation of the flux over the edge γ_{ij}:

$$F_{ij}(S) \approx \int_{\gamma_{ij}} f_w(S)_{ij} \vec{v}_{ij} \cdot \vec{n}_{ij} \, d\nu, \tag{3.26}$$

where $f_w(S)_{ij}$ denotes the fractional flow function associated with edge γ_{ij}, $\vec{v}_{ij}$ is the Darcy flux on the edge γ_{ij}, and $\vec{n}_{ij}$ is the normal vector pointing out of the cell Ω_i. For an upstream-weighted scheme for the fractional flow, the saturation for the fractional flow function is chosen as the saturation in the cell from which the fluid flows:

$$f_w(S)_{ij} = \begin{cases} f_w(S_i) & \text{if } \vec{v} \cdot \vec{n}_{ij} \geq 0, \\ f_w(S_j) & \text{if } \vec{v} \cdot \vec{n}_{ij} < 0. \end{cases} \tag{3.27}$$

Choosing $k = n$ in (3.25) gives an explicit time difference, while $k = n + 1$ gives an implicit scheme.

To solve the system of flow (3.4) and transport (3.16) equations, different numerical time-stepping strategies may be applied. A fully implicit method will solve the system simultaneously and implicitly at each time step. This is an unconditionally stable, but computationally very expensive method. The IMPES method, implicit pressure explicit saturation, solves for each time step first the pressure equation implicitly, then uses updated values of total velocity for the saturation equation, which is solved explicitly. This approach is less computationally expensive compared to fully implicit, but may require small time steps for stability of solution. Finally, the sequential method solves the set of equations sequentially in each time step using any optimal method for each of the equations. Considering the different nature of the two equations, the pressure equation being elliptic, while transport hyperbolic, optimal methods will possibly exploit this fact.

Throughout this thesis, sequential splitting will in general be applied. More specific, a mixed finite-element method will be applied for the pressure together with the implicit finite-volume discretization of the saturation.

Chapter 4

Upscaling and Upgridding

As mentioned in the introduction, fluid flow in reservoir modeling can be considered at multiple scales, from pore scale to field scale. For most accurate modeling, important features from the fine scale should be taken into consideration in the overall behavior at the coarser field scale. To some degree, fine-scale details are incorporated in the permeability tensors assigned to each representative elementary volume. However, modeling fluid flow at field scale requires in general further upscaling of the geological models due to the large gap between the size of the detailed geological models and the capacities of commercial simulators. Here we will consider the upscaling of these geological models to compatible simulation models.

Typically, upscaling of reservoir properties is carried out in combination with upgridding of the geological grid. Upscaling in this context, is the process of averaging reservoir parameters from the fine grid to the upgridded coarse model. This reduces the level of detail that the model can represent. Upgridding is the process of generating coarse grids for the upscaled models. In general, the complete flow and transport simulation is carried out on the upscaled model, that is, both the pressure and saturation equations are solved on this model.

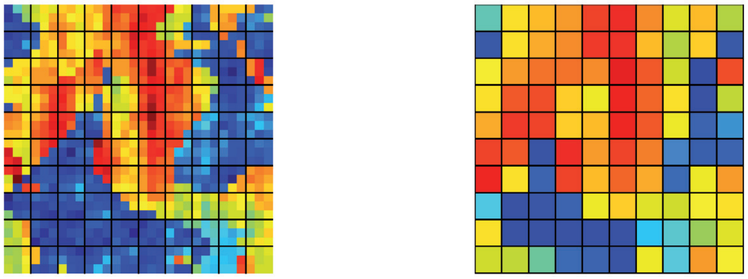
Recently, multiscale methods have been presented as an alternative to the traditional upscaling for flow simulations. These methods incorporate geological subscale details into a coarse-scale model in an accurate and efficient way. However, the multiscale framework has been provided only for the flow equations and a counterpart for the transport simulation has not yet been given equivalent amount of attention. This thesis presents a flow-based coarsening strategy for transport simulations, which aims to utilize the fine-scale details provided by the multiscale flow solvers and solve the transport in an accurate and efficient manner suitable to complement the multiscale flow solvers. The papers of Part II presents my research on this coarsening strategy. In the following, we will introduce the main traditional upscaling techniques and the different upgridding strategies. To provide a complete background for the research papers of Part II, we also describe the multiscale alternative.

4.1 Upscaling

The simplest method for upscaling any reservoir parameter from a fine model to a coarse model, is using standard arithmetic mean. Each coarse grid block is then assigned a mean value based on the fine grid cells that constitute the volume of the coarse grid block. Figure 4.1 shows a simple example. The permeability of a fine model of 30×30 cells is upscaled to a 10×10 grid by using arithmetic averaging. The figure shows the logarithm of the permeabilities both on the fine and upscaled models. The coarse grid is outlined on top of the fine model to illustrate the cells which are averaged. This figure gives an indication of the errors introduced by upscaling.

For single-phase flow, the only reservoir parameters to upscale are the porosity and absolute permeability (or transmissibility in discretized models). For two-phase flow, the relative permeability and capillary pressure come in addition to the two single-phase parameters. In many cases, it is possible to achieve reasonably accurate coarse-scale models for two-phase flow without upscaling of the relative permeability, in particular when used in connection with flow-based upgridding or grid generation. In such cases, the fine-scale relative permeabilities are used directly on the coarse scale. Thus single-phase upscaling can also be applied for two-phase flow problems, [22].

In single-phase upscaling, one of the simplest methods for computing coarse permeabilities is power averaging methods, [21]. This term covers typical averaging techniques such as arithmetic, geometric, and harmonic averaging, and the different techniques are obtained by varying a power-averaging exponent. Next, pressure-solver upscaling methods, using flow simulations to upscale parameters, are more robust and accurate for computing upscaled permeabilities. One approach is suggested by [9] where a pressure solver with specific boundary conditions is used to compute the effective permeability that generates the same flow rate as the fine-scale computation. Extensions to use of periodic boundary conditions to obtain a full effective permeability tensor is employed in [63]. Other, more rigorous methods include the renormalization approach by [45], which computes better estimates of the permeability than simple averaging, and homogeniza-



Fine-scale model

Upscaled model

Figure 4.1: Example of upscaling of permeability using arithmetic averaging. The fine-scale model in the left plot consists of 30 × 30 cells, while the upscaled model to the right has 10 × 10 coarse grid cells. The coarse grid is outlined on top of the fine model to illustrate which cells are averaged.

tion of the coefficients, [37, 41]. For two-phase upscaling mainly pseudorelative permeabilities have been employed, [48]. For more detailed reviews of single-phase and two-phase upscaling techniques, we refer the reader to [23, 24, 30, 19, 7]. A comparison of upscaling methods to multiscale methods is provided in [46].

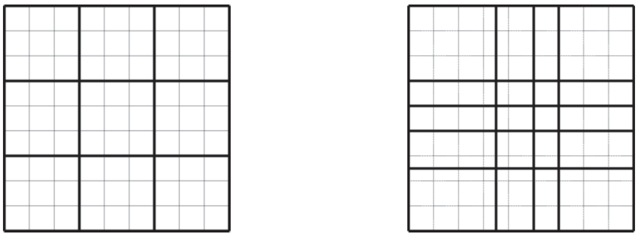
Another classification of upscaling methods, is by considering the domain used to convert fine-scale quantities to coarse grid cells. Using a *local* upscaling technique, as most of the above-mentioned techniques are, coarse-scale properties are computed by only considering the fine-scale region corresponding to the target coarse region, or more specifically the target coarse block in the coarser grid model. In *global* upscaling, the entire fine-scale model is considered in the computation of coarse-scale properties. For instance in [36] the global flow pattern is taken into account to compute the effective permeability on coarse scale. Moreover, there are extended local methods that consider an extended region in the fine-scale model bordering the target coarse block in addition to the coarse block itself. Finally, quasi-global methods attempt to estimate the effects of the global flow without actually solving the global fine-scale problem by for instance use of a global coarse-scale simulation and then in turn use local computations (or extended local). The computational cost is typically greater for global or quasi-global methods, while accuracy might be better than for local methods.

4.2 Upgridding

Upgridding techniques determine the spatial resolution of the coarse grid models, onto which the different properties are upscaled. The simplest type of a coarse grid is a uniform partitioning of a structured grid, with coarse grid interfaces aligned with the interfaces of the fine underlying grid. The left plot in Figure 4.2 shows an example of a 3 × 3 upgridded coarse grid with an underlying fine grid of 9 × 9 grid cells. The coarse grid cells are uniform and the coarse grid lines are aligned with the underlying grid. The right plot in Figure 4.2 shows an example of a nonuniform, nonaligned 4×4 coarse grid, where the spacing of the coarse grid lines is nonuniform (i.e., varying spacing between coarse grid lines) and the grid lines are no longer aligned with grid lines of the underlying fine grid.

Advantageous upscaled grids are assumed to be grids with high resolution where necessary and a simplified description where possible. Similarly as the fine-grid representation of reservoirs should capture the correct geometries of the underlying geological structures, good upscaled models should also take important geological features into consideration by imposing geometric constraints. This enforces constraints of the grids. General reviews of upgridding can be found in [23, 24].

Most upgridding methods are formulated using some background quantity that determines the spatial resolution of the grid. Permeability, fluid velocity, fluid vorticity, and *a priori* local error analysis have been investigated for use in upgridding in the literature. In [32], the permeability is used as the background indicator for upgridding, and [29] investigates Delaunay tessellation based on different grid-point density indicators. Use of local *a priori* error analyses for upgridding has been carried out in [44]. Incorporating both grid-resolution targets and geometrical constraints is described in [59]. Constrained gridding with respect to underlying reservoir geometry is covered in [11], where both constrained two-dimensional Delaunay and Voronoi grids are considered. In [25], a nonuniform approach that is constrained to be aligned with the underlying fine grid is presented. Moreover, flow-based upgridding has in particular been shown to be a powerful approach in combination with upscaling. This type of upgridding introduces higher resolution in regions of higher velocity, while allowing coarser resolution in regions of lower velocity. Flow-based gridding techniques have been developed for Cartesian and structured frameworks, as well as for unstructured and triangular grids, [16, 62, 35]. Recent research investigates vorticity of the velocity field to determine spatial resolution, [52, 28].



Uniform, aligned coarse grid Nonuniform, nonaligned coarse grid

Figure 4.2: Illustratory example of upgridding. To the left a 9×9 grid is upgridded to a uniform, aligned coarse grid of 3 × 3 blocks. To the right the fine grid is upgridded to a 4 × 4 grid with coarse grid lines both aligned and nonaligned with the underlying grid lines. The spacing of grid lines is nonuniform.

The gridding strategies presented above are all meant for upscaling of reservoir parameters and subsequent solution of both flow and transport. In Part II, we present a strategy that is developed especially to generate flow-based, nonuniform grids that are tailor-made for use in transport simulations only. The corresponding coarse-grid transport solvers are well suited to be applied in combination with multiscale pressure solvers.

4.3 Multiscale methods

Multiscale methods have been presented as a robust alternative to upscaling of the pressure equation and have proved to be efficient on large reservoir models. In flow simulations, multiscale methods use the geological fine-scale models to solve the problem on a coarse model. This means for instance that rock properties are not upscaled onto the coarse level. Instead, fine-scale rock information is incorporated into the global solution through local subproblems. Although solutions of the flow equations are at the coarse level, the fine-scale velocity field can be reconstructed on the fine-scale as well. The underlying idea of the multiscale methods is thus to capture subgrid effects on coarse grids and to allow reconstruction of velocity or pressure fields on underlying fine grids. Figure 4.3 illustrates the concepts of multiscale methods: A coarse grid is defined over the geological model; local subproblems are defined over pairs of blocks that share an interface; and multiscale basis functions that incorporates fine-scale flow variations are computed. The coarse-scale flow problem is solved and a fine-scale flow field can be reconstructed using the multiscale basis functions.

Several multiscale methods have been presented in the literature. The concept of heterogeneous multiscale methods and the general methodology

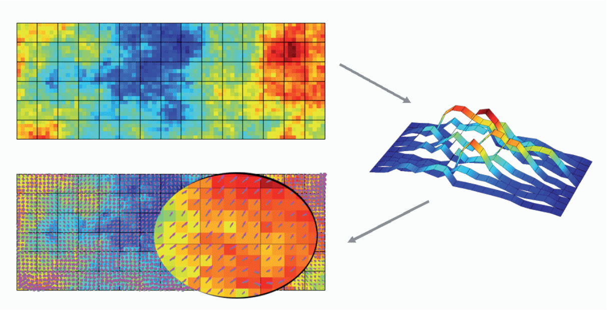


Figure 4.3: Illustration of the key features of multiscale methods. A coarse grid is defined over the geological model; local subproblems, defined over pairs of blocks that share an interface, compute multiscale basis functions that incorporate fine-scale flow variations; and the fine-scale flow field can be reconstructed using the computed multiscale basis functions.

are described in [26]. Here the idea of solving a macroscopic state of a system, by using the microscopic model for the microscale state is introduced. The multiscale finite-element method (MsFEM) is first introduced in [38]. Next, the multiscale *mixed* finite element method (MsMFEM) is presented in [17]. A version of the multiscale mixed finite-element method that incorporates mimetic finite-difference methods as sub-grid solvers is suggested in [4]. This version of the multiscale methods has been employed in some of the papers of this thesis. A multiscale finite-volume method is suggested in [40]. The main distinction between the different multiscale methods is how the local flow problem for computing the velocity basis functions are constructed. A general presentation of multiscale methods can be found in [27] and a comparison of different methods is provided in [46].

To describe the multiscale mixed finite-element method in more detail, we define a coarse grid, where the coarse flow problem will be solved. The fine-scale information is incorporated in the approximation spaces in which the solutions are sought. This means that special multiscale basis functions are computed. In general, these multiscale basis function are computed only for the velocity $\vec{v}$. An approximation space for the pressure p that reflects subgrid information, can also be defined. However, solving only for the velocity is usually satisfactory since for incompressible flow the pressure is never used explicitly (except possibly to determine well-rates through use of appropriate well-models). With the multiscale method, the two variables, pressure p and velocity $\vec{v}$, can thus have different resolution and computational capacity can be used where most needed. However, the two approximation spaces must still satisfy the Ladyzhenskaya–Babuška–Brezzi (inf-sup) condition.

We now define the approximation space for the velocity $\vec{v}$. Consider a coarse grid that consists of blocks K_i containing a connected set of cells from an underlying fine grid. We associate one vector of basis functions $\vec{\psi}_{ij}$ with each non-degenerate coarse interface γ_{ij} between two neighboring coarse grid blocks K_i and K_j

$$\vec{\psi}_{ij} = -\mathbf{K}\lambda\nabla\phi_{ij} \quad \text{in } K_i \cup K_j, \tag{4.1}$$

where ϕ_{ij} is determined by

$$\begin{aligned} (\nabla \cdot \vec{\psi}_{ij})|_{K_i} &= l(x)/\int_{K_i} l(x)\; dx, && (4.2) \\ (\nabla \cdot \vec{\psi}_{ij})|_{K_j} &= -l(x)/\int_{K_j} l(x)\; dx, && (4.3) \end{aligned}$$

with no-flow boundary conditions along the edges $\partial K_i \cup \partial K_j \backslash \gamma_{ij}$. The function $l(x)$ can be defined in several ways. Solving for the velocity field using an incompressible flow model, a natural choice would be use of the Raviart–Thomas mixed finite-element method on the coarse scale. This means we want to solve the local subproblems such that they reconstruct the Raviart–Thomas basis functions on the coarse scale. For this choice, it is common to use $l(x) = \text{trace}(\mathbf{K})$ away from possible wells and $l(x) = q(x)$ in grid blocks penetrated by wells. For more details, the reader is referred to [3, 2].

In general, using upgridding and upscaling of the reservoir model means that both the flow and transport is solved on the same upscaled model. For a multiscale approach to reservoir simulation, the first component is a multiscale flow solver, as described above. For the second component, the transport, we can either evolve the saturation on a coarse grid (for instance the grid from the multiscale flow solver), a fine grid, or some intermediate grid. Solving the transport on a coarse grid will in general be too inaccurate, while solving the transport on a fine grid will in most cases be computationally too expensive. A multiscale approach to the transport solver has been proposed in [49, 65]. Here an adaptive multiscale finite-volume method is presented. This method uses three prolongation operators with different computational complexity to construct a multiscale transport solver.

Another multiscale approach, using an intermediate grid, has been one of the topics in the research of Part II. Here a grid coarsening strategy is developed and investigated with the purpose of serving as an intermediate saturation grid for multiscale transport simulation. The combination of the coarse grid together with a multiscale flow solver seems promising. The next chapter provides a summary of this research.

Chapter 5

Summary of Papers

In this chapter we summarize the research papers of Part II and comment on how they are connected, on new results and insight that have been obtained after the papers were written, and on ideas that need to be further investigated. The papers are structured according to their position in the sequence of fluid flow description in porous media. This means that we start by considering work done on the fluid model itself in Paper I, developing a multiscale method for the Stokes–Brinkman flow model. Next, we consider upgridding for simulating transport. Upgridding here is flow-based coarsening by amalgamating fine grid cells. This coarsening in combination with a multiscale pressure solver, is presented as an alternative to traditional upscaling/upgridding techniques. Papers II to IV cover different aspects of this coarsening method: Paper II introduces the initial coarsening algorithm, Paper III describes improvement of the algorithm and discusses additional features possible to combine with the algorithm, and Paper IV provides a general framework of coarsening principles and draws connections to existing methods in the literature. Next, Paper V applies the coarsening strategy on fractured reservoirs. Finally, in Paper VI we extend the multiscale concept to optimization of wells using the adjoint formulation and we exemplify the use of flow-based coarsening in combination with a pressure multiscale method for fast and accurate simulations. This shows an actual application of the flow-based coarsening where it is incorporated in a multiscale framework.

Numerical examples and simulations in the papers are produced in Matlab, mostly using the framework of the Matlab Reservoir Simulation Toolbox [55, 50] or earlier versions of the code.

Several of the numerical simulations in the papers are performed on Model 2 of the Tenth SPE Comparative Solution Project [20]. This is a

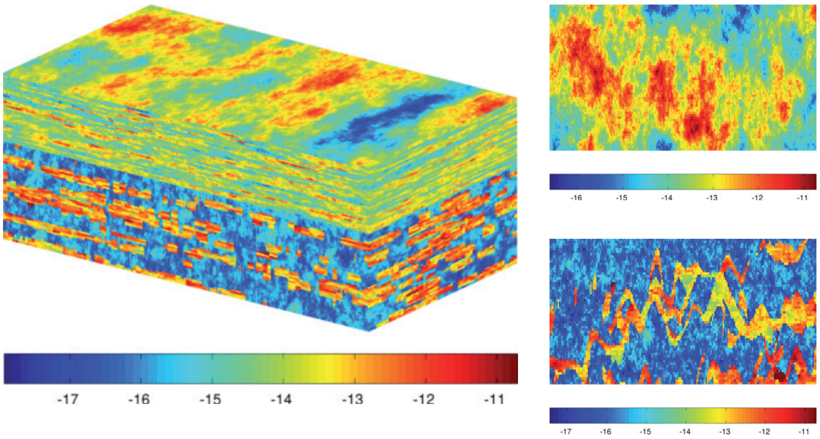


Figure 5.1: Logarithm of horizontal permeability of Model 2 of the Tenth SPE Comparative Solution Project. The left plot shows the complete model, while the top left plot shows layer 15 and bottom left plot shows layer 37.

large geostatistical model used for comparison of upscaling and upgridding techniques for simulation of waterflooding. The model is three-dimensional, described on a regular Cartesian grid of size $60 \times 220 \times 85$ cells, with dimensions $1200 \times 2200 \times 170$ ft, see left plot in Figure 5.1. The top 35 layers represent the smooth Tarbert formation, with log-normal permeability distribution. The lower 50 layers represent the fluival Upper Ness formation. This formation is highly channelized with intertwined high-permeable channels on low-permeable background. Top right plot in Figure 5.1 shows an example of the permeability distribution from the Tarbert formation (layer 15), while the lower right plot shows an example from the Upper Ness formation (layer 37). In the following, the model will be referred to as the SPE 10 model. The reason for the wide use of this model, has been the possibility of evaluating the performance of grids on several similar cases, both log-normal permeability distributions and highly heterogeneous fields, such that we can obtain a more general opinion of the typical performance of the grids. In addition, the model is widely used by other researchers and is well-know in the field of reservoir simulation.

Paper I

A Multiscale Mixed Finite-Element Method for Vuggy and Naturally-Fractured Reservoirs

Astrid Fossum Gulbransen, Vera Louise Hauge, and Knut–Andreas Lie

SPE Journal, Volume 15, Number 2, June 2010, pp. 395–403
DOI: 10.2118/119104-PA

We develop a multiscale mixed finite-element (MsMFE) method for modeling of vuggy and naturally-fractured reservoirs. In this setting, multiscale methods have a natural and intuitive application since vugs, caves, and fractures are present at multiple scales. We model the free-flow and porous regions using the unified approach by the Stokes–Brinkman equations and avoid the necessity of resolving specific boundary conditions between the regions. The MsMFE method uses a standard Darcy model to approximate the solution at the coarse level, whereas fine-scale effects are captured through the multiscale basis functions computed numerically by solving local Stokes–Brinkman flow problems on the underlying fine grid. Both fine-scale discretization of the local problems as well as the coarse flow problem is described. Taylor–Hood elements are used for the local Stokes–Brinkman flow problems.

The paper includes some illustrative numerical examples. First, numerical simulations demonstrate that the effect of the second-order Stokes term of the Stokes–Brinkman model is indeed negligible when considering pure Darcy flow, and that the discrepancy between using Taylor–Hood elements and Raviart–Thomas elements for pure Darcy flow is of the same order of magnitude as the discrepancy introduced when using the multiscale method. Next, numerical simulations demonstrate how well the fine-scale Stokes–Brinkman flow in vugs and fracture networks is modeled within a coarse-scale Darcy flow model using multiscale elements. For models with short correlation lengths, for instance models where each free-flow vug is confined to one single coarse block, the multiscale method reproduces the flow with good accuracy. For models with moderately longer correlation length (fractures penetrating a few consecutive coarse blocks), the multiscale method still reproduces the flow with reasonable accuracy, although some qualitative differences in the resulting velocity field appear. In models with long correlation lengths, represented by models with vugs connected by fractures, the qualitative differences are more obvious and reveal large local errors, even though the velocity discrepancy has not significantly increased compared to the two previous models. Figure 5.2 shows both the fine-scale and multi-

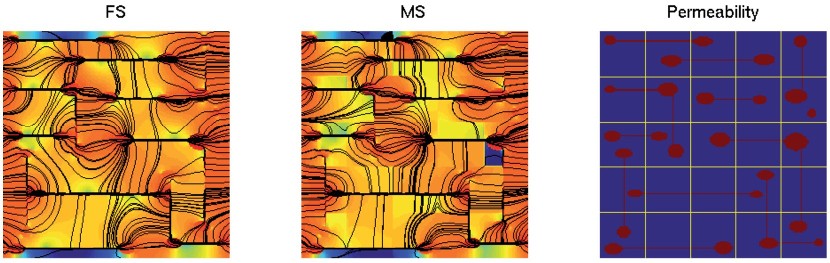


Figure 5.2: Simulation results of a vuggy reservoir with a homogeneous matrix and high-permeable fractures. The rightmost plot shows the permeability with vugs and fractures. The left and middle plots show the logarithm of velocity for the fine-scale and multiscale solutions, respectively, overlaid by streamlines.

scale solutions of the latter case, and illustrates the described results. The paper provides an explanation to this behavior, as well as suggestions of improvements.

Comments: Paper I demonstrates the use of a multiscale multiphysics method for simulations in reservoirs with different physics at different scales.

Further work on the multiscale method for Stokes–Brinkman has been continued by researchers at SINTEF and University of Oslo, Physics of Geological Processes. This work includes extensions of the method to three dimensions and simulations on real data sets. In particular, comparisons of the relative importance of the Stokes and Darcy terms for a wide variety of parameter ranges have been done. Moreover, the resolution of the MsMFE method has been compared to fine-scale discretizations on structured and unstructured grids and to flow-based upscaling, [47, 51].

An inevitable challenge one meets when discretizing with Taylor–Hood elements, is the high number of degrees of freedom. Increasing model size, especially in three dimensions, soon makes the flow problem computationally intractable. However, the computation of basis functions has the potential of both parallelization and reduced memory requirements. On the other hand, the linear system for the Stokes–Brinkman equations results in badly conditioned systems, due to the large permeability contrasts in the model. Better solvers, both direct and iterative, should be investigated to overcome the bad conditioning.

A subgrid approach for upscaling the Stokes–Brinkman equations is also developed in [64]. Here a two-scale finite-element method is derived and the Brezzi–Douglas–Marini mixed finite-element space of order 1 is employed.

Paper II

Coarsening of Three-Dimensional Structured and Unstructured Grids for Subsurface Flow

Jørg Espen Aarnes, Vera Louise Hauge, and Yalchin Efendiev

Advances in Water Resources, Volume 30, Issue 11,
November 2007, pp. 2177–2193
DOI: 10.1016/j.advwatres.2007.04.007

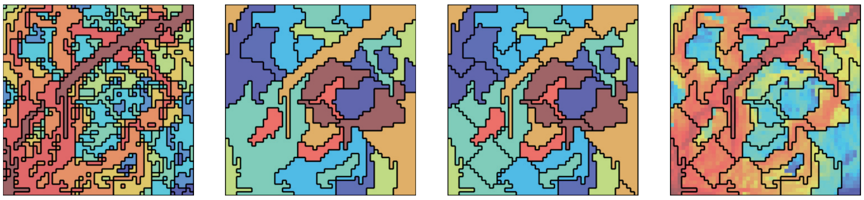
Paper II is the initial paper on a flow-based nonuniform grid coarsening algorithm. The algorithm is intended to generate coarse grids suitable for transport simulations, used in combination with multiscale pressure solvers. The algorithm assumes a velocity field on a fine grid. Shortly summarized, the coarsening algorithm consists of four steps:

1. An initial segmentation of the fine-grid cells according to logarithm of velocity into ten bins. The collections of blocks have to be connected.
2. Merging of too small blocks according to volume. Too small blocks are merged to the neighboring block of closest flow magnitude.
3. Refining too large blocks according to total flow magnitude through the blocks.
4. A final merging (Step 2) before terminating.

Two coarsening parameters determine the level of coarsening: A lower bound on the volume of blocks, N_L, prevents the algorithm from generating too small blocks, and an upper bound on the total amount of flow through each grid block, N_U, prevents the algorithm from generating too large blocks. Figure 5.3 illustrates the four steps. Here the partitioning after Step 1 is outlined on top of the ten bins. In Steps 2 and 3, the blocks are colored randomly, while in Step 4 the final grid is outlined on top of the logarithm of the underlying fine-scale velocity field.

The paper provides discussions of applicability and limitations, as well as an analysis of the algorithm. A discretization of the saturation equation intended to be used in conjunction with the coarse grid is presented. In this discretization, fine-scale velocity components across the fine-scale interfaces that constitute the coarse interfaces are utilized. This means that fine-scale resolution is brought into the discretization and bi-directional flow across coarse interfaces is allowed.

Numerical experiments are performed using the SPE 10 model and a corner-point model with 30 layers of log-normal permeability distribution.



Step 1: 304 blocks Step 2: 29 blocks Step 3: 47 blocks Step 4: 39 blocks

Figure 5.3: Illustration of Steps 1 to 4 of the coarsening algorithm, using a quarter five-spot simulation of a 50 × 50 excerpt from Layer 68 from the SPE 10 model. The grid in Step 4 is outlined on top of the logarithm of the underlying fine-scale velocity field.

Saturation solutions using the non-uniform grids are compared with saturation solutions on uniform coarse grids. Both two- and three-dimensional examples are included. In particular, water-cut errors and averaged saturation errors are compared. The numerical simulations show that the flow-based grids give good accuracy, especially for highly heterogeneous reservoirs. The main difference in accuracy of flow-based and uniform coarse grids is on highly heterogeneous reservoirs, where the flow-based grid blocks better match the flow patterns, while uniform grids tend to excessively smear out the flow. Even high upscaling factors produce grids with satisfactory accuracy. Robustness with respect to the degree of coarsening, well-placement, and changing flow conditions are evaluated. The flow-based grids are found to be robust to all these changing conditions, indicating that it is not necessary to regenerate the grids when flow conditions change.

Comments: In retrospect, we see that the paper is somewhat too optimistic with regard to the efficiency and general applicability of the method. A prerequisite for successful generation of effective coarse grids is that the flow is dictated by permeability. This is especially the case in highly heterogeneous reservoirs, for instance the fluvial layers of the SPE 10 model, with high-flow channels on a low-permeable background. The numerical results reflect this fact.

Moreover, in investigating the robustness of the algorithm with respect to well-placement and changing flow conditions, one can argue that the variation in well conditions studied in the paper is modest in the sense that high- and low-flow regions are more or less preserved for all the well configurations considered. The generated coarse grids will likely be less robust with respect to changing well conditions when these changes constitute a significant change in flow pattern.

However, the idea of several different well-configurations with different flow patterns sharing *one* realization of a flow-based nonuniform coarse grid, is still of interest. Combinations or intersections of different flow conditions may be used to produce more general coarse grids. More studies regarding coarse grids for varying well conditions is also interesting in several application in which wells might be opened or closed, or in which one wants to have a general grid for several possible alternative cases. Research in this area is suggested as future work.

Later work has also shown that bi-directional flow across coarse interfaces is not significant for the accuracy of solution. Indeed, preliminary studies indicate improved accuracy by using net flow across coarse interfaces. This may be explained by less numerical diffusion introduced when using net flow. In addition, using net flow gives a simpler matrix structure in the linear system, thus giving improved computational efficiency if exploited. Paper III covers more on this topic.

The idea of combining the coarse grids for the transport solver with a multiscale pressure solver, is strongly emphasized in this introduction. However, all numerical testing of the quality and accuracy of the coarse grids is done by computing pressure updates on the *fine* grid. This is done to avoid introducing errors from other sources than the coarse grid or coarse grid discretization. Simulation results will therefore not necessarily reflect the overall accuracy of a fully coarse or multiscale simulation. Actual combination with a multiscale pressure has been carried out, however, without any study or evaluation of the total error. To fully complement the study of coarse grids, actual testing and evaluation within a fully multiscale setting needs to be done.

Construction, testing, and improving the algorithm of this paper has been a major part of the work of this thesis. Papers III and IV summarize improvements, variations, and an abstraction into a general framework of this initial algorithm. Paper VI is an example of actual and efficient use of the coarsening algorithm.

In particular, Paper IV places the algorithm in a more general coarsening framework. This framework defines the algorithm as one, indeed *ad hoc*, implementation of a set of more general coarsening principles. In our effort to improve the original algorithm of Paper II, we came up with a large number of changes to various parts of the algorithm. In particular, imposing *a priori* shapes in the coarse grids, considering the coarse discretization scheme, and dynamically adaptive grids, are topics described in Paper III. During the work, we realized that these different choices could be combined in various ways, thereby leading to different algorithms that would generate different types of flow-based coarsening. The result of these ideas is summarized in Paper IV.

Paper III

Flow-Based Coarsening for Multiscale Simulation of Transport in Porous Media

Vera Louise Hauge, Knut–Andreas Lie, and Jostein R. Natvig

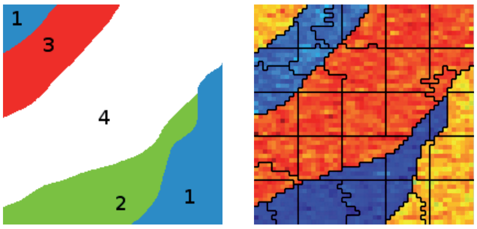
Submitted
Short version: In Proceedings of ECMOR XII – 12th European Conference on the Mathematics of Oil Recovery, Oxford, UK, 6–9 September 2010.

This paper suggests algorithmic improvements of the flow-based nonuniform grid coarsening algorithm from Paper II. The improvements aim at introducing more regularity into the grids and reducing the emphasis of flow in reservoirs with small or modest heterogeneity. More precisely, the improvements of the algorithm consist of including an *a priori* partitioning in the first coarsening step; the initial segmentation according to velocity is intersected with, in the case of a structured Cartesian grid, a uniform partitioning of the grid. The next improvement, to reduce the effect of underlying heterogeneity, we reduce the number of bins of colors into which the fine-grid velocity is segmented. In particular, we propose to set the number of bins of colors equal the span in magnitude of the logarithm of the underlying velocity.

Introducing *a priori* shapes into the grids, is also applied in the case of adapting coarse grids to special geological features. We use an example with facies distribution as the *a priori* partitioning, see Figure 5.4. The coarsening process preserves the facies by preventing coarse grid blocks to cross the different facies boundaries.

Next, we propose to simplify the coarse-scale discretization by using net fluxes across coarse interfaces instead of bi-directional fluxes as used in Paper II. Firstly, this simplifies the matrix structure of the discretized system as there are fewer entries. The simplified matrix structure is easily reordered according to flow. Such reordering generates favorable block structures of the matrix which can be exploited in efficient linear solvers. Secondly, simulation results show that net fluxes give even more accurate transport solutions. This is because there is less numerical diffusion introduced by the net fluxes into the numerical scheme. Figure 5.5 illustrates the difference in saturation distributions using net fluxes and bi-direction fluxes in the coarse discretization. We clearly see that the bi-directional fluxes result in more diffusion.

Lastly, the paper demonstrates use of dynamic adaptivity. We consider the case of a sharp saturation front and show how local refinement along



Facies distribution Facies and coarse grid

Figure 5.4: Coarsening using *a priori* information of facies distribution.

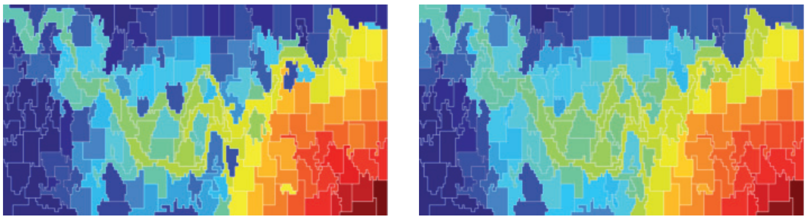
the moving front increases accuracy significantly. The coarse grids are represented as partition vectors, and dynamically adding and removing local resolution is therefore implemented as straightforward manipulation of the partition vector. We include an illustration of the local refinement in Figure 5.6.

Comments: This paper presents specific alternatives and variations of the initial algorithm of Paper II. Work has also been carried out in a more generalized path regarding the coarsening algorithm. Paper IV summarizes this work.

The ideas presented are conceptually simple. However, they seem to be very effective and point out promising alternatives in flow-based coarsening. The aim of the research on flow-based coarsening is not to find the *one* optimal coarse grid suitable for all purposes. Rather, the aim is to find suitable variations and alternatives for different purposes. Hence, the focus of the work has not been to undertake rigorous and extensive numerical simulations to assess accuracy of the grids, nor the robustness with respect to changing conditions, properties and simulation cases. However, to specialize and fine-tune the alternatives and variations to be able to provide guidelines for use, such rigorous numerical studies will be necessary.

The results of the paper are mainly presented visually. Parts of the development of ideas has also been done by visual inspection and assessment of plots of grids, considering the various properties and features. Although all presented concepts are applicable to three dimensions, inspecting grids visually in three dimensions quickly becomes a challenging task. Developing regularity measures and other measures of the quality of grids with respect to certain properties will enable a more powerful development for three-dimensional gridding strategies.

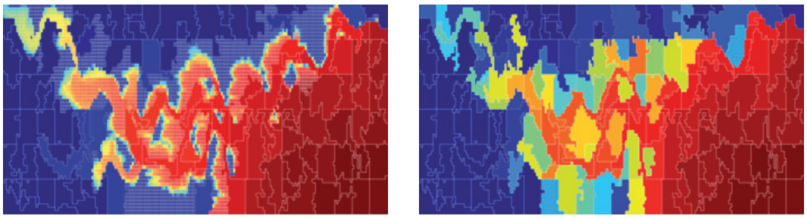
The results regarding use of net fluxes, rather than bi-directional fluxes in the coarse discretization, are very promising. The increased accuracy using



Net fluxes

Bi-directional fluxes

Figure 5.5: Comparison of saturations at dimensionless time 0.8 PVI, Layer 37 of the SPE 10 model, using net fluxes (left) and bi-directional fluxes (right).



Locally adapted grid at 0.5 PVI

Coarse grid at 0.5 PVI

Figure 5.6: Example of locally adapted grids for Layer 37 of the SPE 10 model. Left plot is the dynamically adapted grid, while the right plot is the coarse grid used for estimation.

net fluxes is of course interesting in itself. However, the potential of applying more efficient linear solvers by exploiting the causality of the discretized system, is a quite uncovered topic in research regarding gridding and upscaling and further research should be encouraged. This is suggested as future work. Also a closer look into the actual mechanisms of the discretizations is suggested as future work.

The trends of regularity and simpler matrix structures are very promising also with respect to multiscale speedup and mappings between coarse multiscale grids and coarse transport grids. This is further discussed in the comments to Paper VI.

Initially, the work on using net fluxes in the discretized scheme was motivated by the idea of reordering of unknowns to be able to apply efficient linear solvers exploiting the cascading order, [57, 56]. Initial work consisted of combining flow-based grids with reordered grids. By reordered grids we here mean grids partitioned into dependence areas, such that the solution in one block is dependent of only some neighboring blocks. An example of a reordered grid is given in Figure 5.7. Although the ideas seemed promising, numerical testing revealed that such a partitioning is highly sensitive to the sign of fluxes. After a time step with a pressure update, the dependence

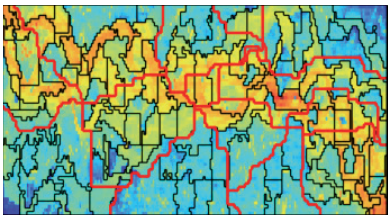


Figure 5.7: Reordered coarse grid. The red lines outline the reordered partitioning which is combined with flow-based nonuniform coarsening.

areas will not be valid any longer. Instead, the desired effect of obtaining a matrix structure suitable for efficient linear solver, could be obtained by using net fluxes and reorganization of the entries in the matrix.

Paper IV

Grid Coarsening Based on Amalgamation for Multi-Fidelity Transport Solvers

Vera Louise Hauge, Knut–Andreas Lie, and Jostein R. Natvig

Preprint

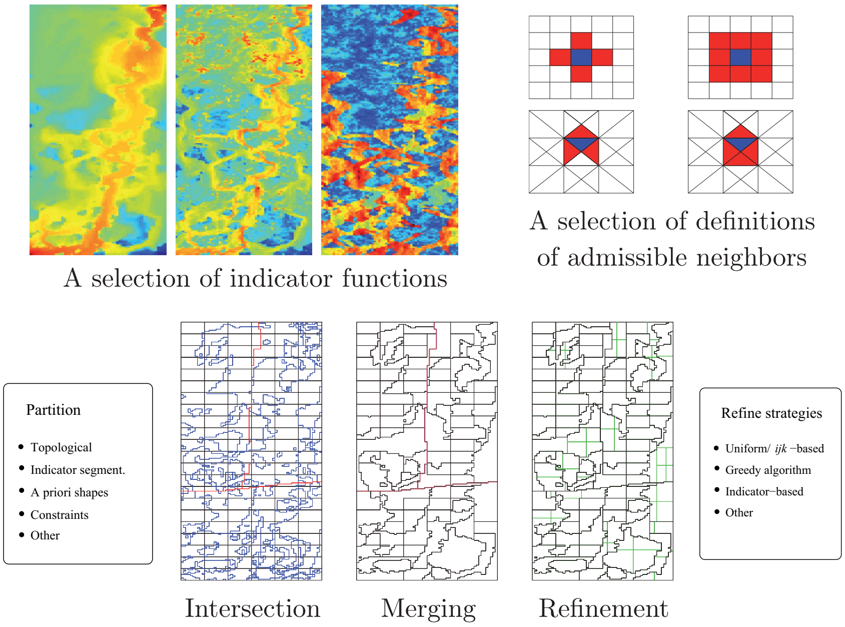
Paper IV presents a general framework for grid coarsening to be applied for transport solvers in combination with multiscale flow solvers. The framework consists of a set of coarsening principles and heuristic algorithms that can be combined in various ways to generate multi-fidelity, fit-for-purpose grids of sufficient quality. The main difference of this coarsening framework from other gridding strategies, is the use of amalgamation of fine grid cells to form coarse grid blocks. The coarse grids are represented as partition vectors and the only assumption made on the fine grid, is that there exist a mapping between each cells and its neighbors. One or more indicator functions, defined cell-wise, are used as a guide for amalgamation directions and deciding the grid resolution. The grid coarsening principles can be summarized as:

1. Minimizing the heterogeneity of the flow field within each coarse block.
2. Equilibrating the error contribution over the coarse blocks.
3. The size of each coarse block should be within prescribed bounds.

Instead of formulating these principles as minimization of functions in an optimization problem, we provide a framework of heuristic algorithms to produce grids with acceptable quality. This framework includes four types of algorithmic components:

1. Partition according to an indicator function, either topologically, based on a set of *a priori* shapes, or by segmentation of fine-grid cells into bins according to the indicator.
2. Intersection of one or more partition vectors, representing different partitioning criteria.
3. Merging grid blocks according to a prescribed indicator function.
4. Refining grid blocks according to a prescribed indicator function.

Combination of these components can be done in various ways. Together with several choices of indicator functions, neighborhood definitions, and different algorithms, the coarsening framework provides for great flexibility and generality of generating coarse grids for transport solvers. Figure 5.8 illustrates the main components of the coarsening framework described in the paper.



A selection of indicator functions

A selection of definitions of admissible neighbors

Intersection

Merging

Refinement

Figure 5.8: The different components in the coarsening framework: The upper left plots illustrate a selection of different indicator functions described in the paper. The upper right plots illustrate different neighborhood definitions for structured grids. The lower plots illustrate the algorithmic components: Partition and its different choices; intersection of indicator-based segmentation (blue), topological partition (black), and *a priori* regions (red) to be preserved; merging without crossing the regions marked with red; and refinement using uniform subdivision marked with green.

As an initial example, the paper formulates the coarsening algorithm of Paper II in the described framework. In the rest of the paper, various heuristic algorithms are outlined and different algorithmic choices are examined. The focus is on qualitative behavior of the different realizations and choices. Firstly, the effect of different definitions of admissible neighbors, in particular topological definitions, is discussed. Especially for the algorithmic component performing refinement, we see a strong impact of neighbor definition on the shapes of coarse grid blocks and coarse interfaces. In addition we show that cell- and face-based constraints imposed on the coarse grids may be preserved throughout the coarsening process by redefining neighborhood definitions.

Next, we give examples of several flow-based indicator functions. Time-of-flight, velocity, vorticity, and the Courant number are all illustrated and discussed in examples. The focus is on indicators that are well-suited to form good transport grids. Moreover, by using the Courant number as the indicator, we demonstrate local coarsening where only parts of the components in the algorithmic framework are employed. This exemplifies the flexibility within the framework to produce specific, fit-for-purpose grids.

The merging component in the algorithmic framework is generally easy to define. However, for the refinement components, there are various possible approaches. The paper discusses a simple uniform subdivision for structured grids, and the *ad hoc* greedy algorithm as implemented in the original algorithm of Paper II. Next, the flexibility of combining the algorithmic components is illustrated through regional, successive, adaptive, and supervised coarsening.

Finally, the paper presents results of some numerical experiments. In particular time-of-flight-based grids and velocity-based grids are compared with respect to accuracy and regularity. The results reveal that time-of-flight is a good indicator of saturation transport. In addition, time-of-flight-based grids have less complex matrix structures in the discretized system, indicating that they will be more computationally efficient. A study of robustness of the original nonuniform coarsening algorithm of Paper II, with respect to the coarsening bounds is included. The results point out trends in saturation errors and water-cut errors, and relate them to the coarsening parameters and the general features of the coarse grids.

The paper concludes by suggesting more numerical investigation of the presented concepts, as well as development of postprocessing of the grids for optimal results. In addition, we point out that the aim is not to create *the* best grid, since such a grid strongly depends on geology, flow processes, requirements of accuracy and runtime, and the purpose for use. However, the emphasis is on the flexibility and generality of the presented modular algorithmic framework.

Comments: This paper describes the state-of-art of the coarsening strategy based on amalgamation of fine grid cells. The paper provides a summary of the current ideas and concepts, while at the same time opens up for further extensions and alternatives within the framework.

There are several natural continuations to the work. As mentioned in the paper, extensive and rigorous numerical studies of the grids are necessary for further development. Such studies will be important to quantify the properties of grids with respect to accuracy, computational complexity and robustness. Next, additional postprocessing will be needed in practical applications to smoothen unconstrained faces, removed different artifacts, as blocks confined within other coarse blocks, and to fine-tune the grids to fit specific purposes. Research to develop guidelines for use is needed. This will also be important in the process of automating the coarsening processes.

In addition, there are openings for further development of heuristic algorithms for the algorithmic components, in particular the refinement strategies, and further investigation of use of other indicator functions. More testing within a fully multiscale framework is also an interesting continuation of the work.

Paper V

Modeling of Two-Phase Flow in Fractured Porous Media on Unstructured Non-Uniformly Coarsened Grids

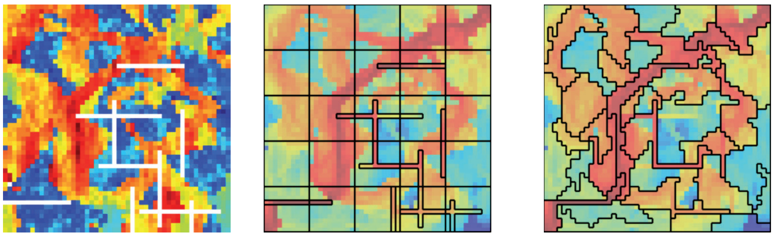
Vera Louise Hauge and Jørg Espen Aarnes

Transport in Porous Media, Volume 77, Number 3,
April 2009, pp. 373–398
DOI: 10.1007/s11242-008-9284-y

Here the nonuniform flow-based coarsening of Paper II is applied on fractured reservoirs. An important assumption for the work is that the fractures are represented as volumetric elements and that the flow in the (conducting) fractures obeys Darcy's law. Thus, the fractured reservoirs considered here are different from the fractured reservoirs in Paper I, where fractures conduct Stokes flow. The flow-based grid coarsening is compared with a uniform coarsening with explicit separation of the fractures into separate grid blocks, referred to as explicit fracture-matrix separation. Figure 5.9 gives an illustration of the uniform coarsening of fractured reservoirs that is compared to the flow-based nonuniform coarsening strategy of Paper II. The left plot shows the permeability field with ten fractures, and the middle plot and the right plot show the uniform and nonuniform coarse grids, respectively. The discretization of the transport equation from Paper II is used for both grids. Both conducting and sealing fracture networks are considered, but the main focus is on conducting fractures.

Numerical examples are run on models with varying number of fractures, with both homogeneous and heterogeneous background permeability fields, as well as with varying degrees of coarsening. Robustness with respect to altering well location and with respect to viscosity ratio is investigated. The general result is that the nonuniform flow-based coarsening reproduces the fluid transport in a more accurate manner. On heterogeneous background permeability fields, the numerical results reflect to a large degree the differences between uniform coarse grids and nonuniform flow-based coarse grids. Already Figure 5.9 supports the numerical results by indicating how the uniform coarse grids with fracture separation may smear out flow patterns other than in the fractures. Indeed, the explicit fracture-matrix separation smears out underlying flow patterns other than in the fractures and this introduces inaccuracy.

The method is also extended to flow models with gravity and capillary pressure, under the assumption that the viscous forces are dominant. The discretization uses operator splitting of the viscous and diffusive flow. The



Permeability field Uniform coarse grid Flow-based coarse grid

Figure 5.9: The logarithm of the permeability field with ten fractures is shown in the left plot. Uniform partitioning with explicit separation of fractures is shown in the middle plot, and the flow-based nonuniform coarsening is shown in the right plot. Both coarse grids are outlined on top of the logarithm of fine-scale velocity.

viscous part has separate upstream weighting of the fluxes and the gravity term. The fluxes are upstream weighted with respect to the total flow, while the gravity term is upstream weighted with respect to purely gravity driven flow. A straight-forward projection of the diffusion from fine to coarse scale together with a scaling or damping, brings the discretized diffusion equation to the coarse scale. The numerical simulations model cases with relatively strong capillary diffusion. Without scaling or damping of the diffusion on coarse grids, the coarse-grid operator overestimates the effect of diffusion. Plots of saturation profiles as well as water cut curves illustrate this. The numerical results also indicate that scaling or damping the diffusion has a favorable effect.

Comments: In the paper, we comment that the fracture models used are not realistic, but rather serve the purpose of assessing the performance and applicability of the proposed methods. One possible extension of the models, is to model fractures as lower-dimensional entities, as used in [60, 42]. Applying the nonuniform coarsening algorithm to such models, involves introduction of constraints in the grid in the form of fine-scale faces that should not be crossed in a coarse grid. Paper IV demonstrates this concepts in general terms.

The alternative approach of explicit *separation* of fractures from a uniform coarse partitioning, would in the case of lower-dimensional fractures not be applicable. However, geometric constraints can be included in uniform partitioning by enforcing coarse interfaces along the constraints. This concept has already been described in Paper III as how to incorporate *a priori* features in coarse grids.

Another alternative to the explicit fracture-matrix separation would be to separate the (volumetric) fracture cells from the matrix by a thresholding on the permeability. This should remove the necessity of having the fracture cells explicitly prescribed. On the other hand, the limit for the threshold could cause high-permeability regions to be identified. A sophisticated use of thresholding would potentially also include the flow effects from these regions. Such separation of some grid cells from another set of grid cells is in fact in loose terms grouping or amalgamation of cells according to the static property permeability, as described in Paper IV. Combining thresholding and uniform partitioning can thus also be expressed in terms of the general framework of Paper IV.

Another interesting case to study, is fracture models with barriers that have unknown properties. For instance, the barriers may be fully sealing or have highly conductive properties. Flow-based coarse grids incorporating both possibilities would have to be based on two flow patterns. A coarse grid could be generated from a combination or intersection of flow patterns using sealing and fully open barriers or even more patterns. The underlying idea is also similar to grids meeting different flow patterns of different well configurations, as described in the comment to Paper II.

Discretization of the gravity term by weighting the mobilities according to the gravity flow should rather be changed to be weighted according to phase flow. Such that both the flux and gravity is upstream weighted in the same manner and with respect to the actual flow. However, it is particularly in stationary cases that the implemented discretization possibly will give wrong results. In the simulated cases in this paper, the difference is most likely very small.

The coarse-scale diffusion operator also needs more investigation. Preliminary studies have been done in [34]. The effect observed in this paper may not purely be the diffusive effect, but might also include physical dispersion due to heterogeneity. Paper III also indicates numerical diffusion introduced through the use of bi-directional fluxes used in Paper II. This has to be included in further considerations and investigations.

Paper VI

Adjoint Multiscale Mixed Finite Elements

Stein Krogstad, Vera Louise Hauge, and Astrid Fossum Gulbransen

Accepted for publication in SPE journal. SPE-119112-PA
DOI: 10.2118/1119112-PA

The paper develops an adjoint multiscale model for rate optimization, where the simulator consists of a multiscale pressure solver and a nonuniform coarse-grid saturation solver. The multiscale solver employs a recent version of the multiscale mixed finite-element method from [4]. The saturation solver works on flow-based nonuniform coarse grids, as developed in Paper II. The adjoint model for rate optimizations is a gradient-based method and computes efficiently optimized rates for a given objective function.

The paper presents the discretization of the forward simulation model, both on the fine scale and the coarse scale. The optimization problem is formulated, and the adjoint model is developed on both scales and we show that the models have basically the same structure as the forward models.

An important contribution of the paper, is the coarse-grid mapping for rapid updating from the pressure solver to the saturation solver. This mapping accelerates the computations involved such that no fine-grid quantities are required during simulations. All fine-grid computations are performed as a preprocessing step; this includes computation of multiscale basis functions, generation of the coarse transport grid, and the coarse mapping of fine-grid quantities between the two coarse grids (multiscale and transport grids). We assume no significant changes in well configurations.

Two numerical examples are presented. The first example consists of rate optimization of a series of two-dimensional Cartesian models. For each model, two coarse models are considered and optimized recoveries are computed and compared with fine-scale model optimization. The results show that both coarse models succeed in giving close to optimal rates compared to the fine model.

In the second example, we optimize the net present value on a real-field model. Figure 5.10 shows the model and the pie charts with blue color indicate injection wells, while pie charts with red color indicate production wells. The colored regions of the pie charts represent the fraction of the total injection/production rates. As an example, injection well number 3 injects at about half the total rate, while injection well number 1 injects at zero rate. Likewise, production wells 1 and 2 produce most of the oil. Six

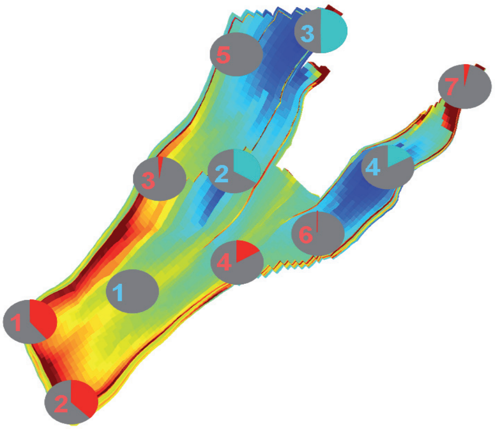


Figure 5.10: Optimized rates for the real-field model. Blue-colored pie charts represent injection wells, while red-colored charts represent production wells. The pie charts show the fraction of the total injection/production rates.

coarse models are considered, and all models compute rates that in turn give net present values quite close to that of the fine-scale optimization model. However, some discrepancies are observed in the two coarsest models. We also observe that optimization converges faster for the coarse models than for the fine model, but we have not managed to find a good explanation for the phenomenon. This field-scale example demonstrates the strength of the multiscale simulator with the mapping between pressure solver and saturation solver. The suggested preprocessing step results in a speedup factor of 10 or higher compared to running the simulation without preprocessing and mapping. We point out that the speedup factor when compared to running the optimization on the fine scale will be significantly larger.

Comments: Paper VI demonstrates the strength and potential of the combination of multiscale methods and nonuniform coarsening in the case of well rate optimizations. Both methods have shown to produce accurate solutions of the pressure and transport equation, respectively. In addition, the fact that grid generation and mapping of information from the coarse pressure solver to the coarse transport solver is performed as a preprocessing step, giving a significant speedup, suggests a large potential for fast and accurate simulations.

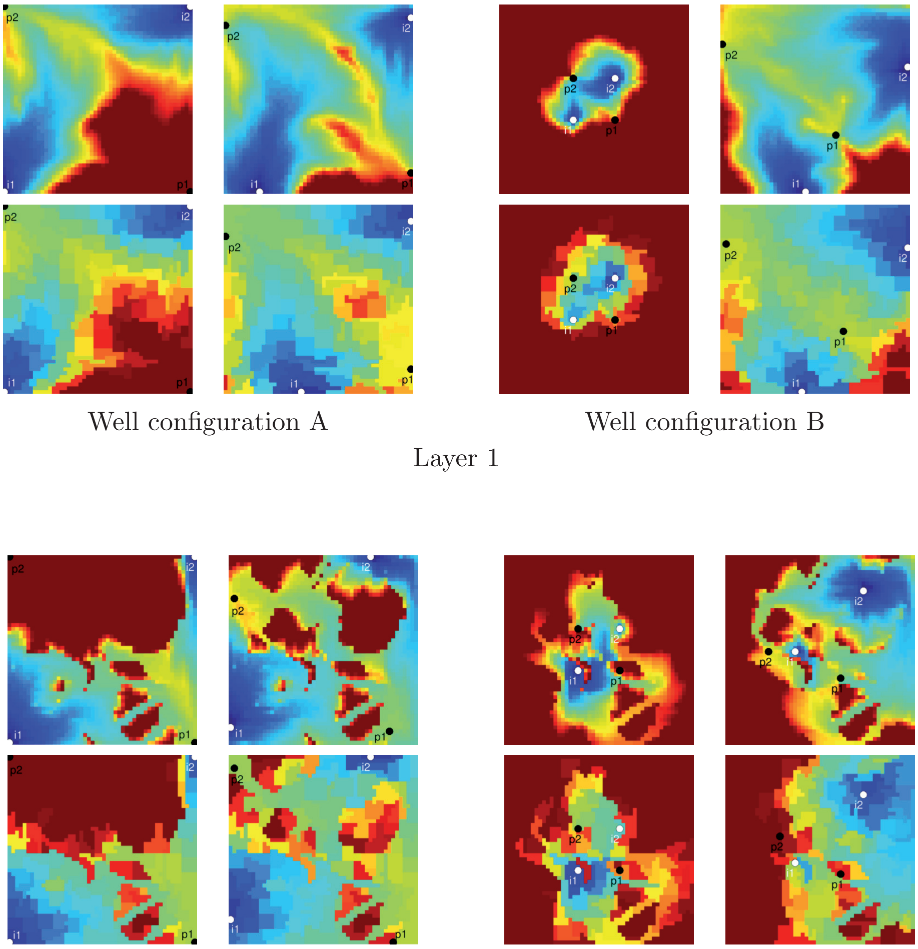
Using flow-based transport grids with more regularity, as introduced in Paper III, might also open up for a possibly simpler mapping between the coarse pressure grid and the transport grid. Investigating possible simplifications in the preprocessing step should be included in future work. In addition, use of net fluxes in the coarse discretization and exploiting matrix structures of the discretized transport system in linear solvers, as described in Paper III, will contribute to the total speedup. Investigating this contribution is also encouraged in the setting of optimization.

For the numerical examples, a prototype MATLAB implementation has been used. For even greater computational savings, a more sophisticated implementation is suggested. Here we have in mind further speedups due to optimization possibilities within the programming language as well as tailored solutions in both the pressure and transport implementation. Faster numerical implementations will in turn allow optimizations of larger and more complex models.

As written in the paper, in the real-field example we observe that the optimization converges faster for the coarser models than for the fine model. An explanation for the phenomenon is missing. Thus, more investigation of the optimization process with respect to convergence is suggested as further work. A possible starting point for further investigation is to look into how the model complexity (fine models assumed more complex than coarse) affects the convergence.

Extensions of the adjoint multiscale method to include well placement optimization has been carried out. For this purpose, we have employed the approach of [33] using so-called pseudo-wells in the adjoint method. The idea is to surround each well to be optimized with respect to location, by pseudo-wells in the neighboring grid cells. These pseudo-wells are set to produce or inject at very low or zero rates, so that they have negligible influence on the overall flow in the reservoir. The adjoint method computes the gradients of these pseudo-wells and these gradients indicate improved locations for the wells.

Here we include two examples of well placement optimization. The first example is a qualitative investigation to show that the coarse adjoint model approximates the fine model reasonably well and how the initial well configuration affects the local optima found by the adjoint method. We consider two models of size 50 × 50 grid cells with different permeability fields. The permeability fields are taken from Layer 1 and 46 from the SPE 10 model, and represents a log-normal permeability distribution and a fluvial structure, respectively. On each of the two models, we start out with two sets of initial well configuration. Configuration A has one well in each corner of the model, while configuration B has four wells placed in a square in the middle of the domain. All four wells are subject to repositioning in the



Well configuration A

Well configuration B

Layer 1

Well configuration A

Well configuration B

Layer 46

Figure 5.11: End-time saturation distributions on Layer 1 (top) and 46 (bottom) after optimization: Leftmost plots in each group are after rate optimization and rightmost plots are after well placement optimization. Top row in each group shows the results on fine scale, while the bottom row corresponds to coarse-scale model.

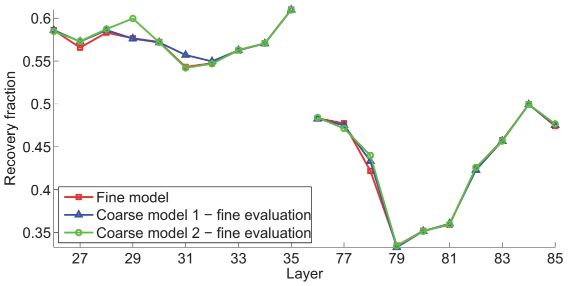
optimization. An initial rate optimization is carried out, before optimizing the positions. Between each repositioning of the movable wells, the rates are again optimized through three optimization iterations. A simple net present value objective function is used, and the optimization is carried out for a simulation time period of 200 days. Figure 5.11 shows end-time saturation distributions of the different cases, for both the fine-scale model and the coarse-scale model, both after the rate optimization and well placement optimization. We have defined convergence of the well placement optimization when the well positions reach a position that has been visited before.

From Figure 5.11 we first observe a reasonably good approximate solution of the coarse well placement optimization, both with respect to the actual positions of the wells and with respect to the general sweep of the reservoir. The end-time fraction of recovered oil to the total volume of oil is for layer 1 well configuration A 0.5512 for the fine model and 0.5546 for the coarse model. Corresponding values for layer 1 well configuration B is 0.5344 and 0.5365, and for layer 46 well configuration A 0.4393 and 0.4022, and well configuration B 0.3427 and 0.3565. Secondly, we observe in the fluvial case (lower plots of Figure 5.11), that different initial well configurations result in different final (local) optima. This is also reflected by the reported recovery fractions above: Starting with well configuration A, results in a well configuration that gives a larger sweep of the reservoir than when starting with well configuration B.

We point out that the good results in these examples, are also due to the quite low upscaling factors for the coarse models. The upscaling factor of the coarse pressure model is 25, while the upscaling factor varies between 13 and 15 for the transport grid. On the other hand, the size of the models is artificially small and the wells move relatively unrealistic long distances, compared to more realistic cases. However, for the purpose of illustration we have kept the model size small. For larger, more realistic examples, the upscaling factors could be adjusted higher, while keeping the necessary resolution in areas where wells are expected to move.

Flow-based transport grids need in general to be regenerated when well conditions and flow patterns change considerably. Regeneration of the coarse saturation grids has been carried out in these illustratory examples. However, no further gain was observed due to the small size of the model problem and the small upscaling factor. For larger models, with a higher upscaling factor of the flow-based transport grid, we expect a regeneration of the flow-based grid to have some impact. More research in this direction is necessary.

The second example runs a series of optimizations on a set of two-dimensional models of 60×220 cells with both log-normal permeability fields



Relative recovery of reservoir

Figure 5.12: Simulations on two sets of layers from the SPE 10 model: Layer 26–35 and 76–85. Optimization consists of rate and well placement optimizing of the four producing wells in the corners of an initially five-spot configuration. The reported optimized values for the coarse models are evaluated using the fine-scale model (optimized coarse values applied on fine-scale model).

and highly channelized fields. The models are again taken from the SPE 10 model. We model a five-spot well configuration: We place an injecting well in the middle of the reservoirs and optimize both the rates and placement of the four producing wells in the corners. Two sets of coarse models are considered. Coarse model 1 has an upscaling factor of 60 in the pressure grid and an upscaling factor between 18–20 in the transport grid. Coarse model 2 has correspondingly upscaling factors of 400 and between 120–132. Comparisons of the final relative recoveries of the reservoirs (fraction recovered of total oil in reservoir) after a simulation time of 200 days, are reported in Figure 5.12. This plot shows that the general performance of the coarse adjoint well placement optimization. We observe good correspondence in values computed by coarse and fine models.

In field-scale optimization of well placement, the reduction in computational complexity through the suggested preprocessing and mapping is expected to be significant. At the time of writing, implementational issues prevent us from running such field-scale examples, investigating the actual speedup and possible further reductions. This is suggested as future work.

In field-scale models it will also be possible to study use of more sophisticated transport grids. We have in mind grids with in general a high upscaling factor, but in regions where well repositioning is expected, we provide a more detailed gridding. We expect the setting of well placement optimization and possible changes of flow conditions due to the optimization process, to reveal further ideas and new directions regarding development of coarse grids.

Chapter 6

Future work

In the comments to the research papers in the previous chapter, several directions for further research have already been suggested. Here we will repeat some of the interesting areas, suggested in more general terms.

Firstly, future work on multiscale-multiphysics methods using the Stokes–Brinkman model, includes a more sophisticated and computationally less expensive implementation. The current prototype implementation has proven to be insufficient for larger problems. Next, expansions to both triangular grids and unstructured grids, and use of other finite elements in the discretization of the Stokes–Brinkman equation will be of interest. Use of (limited) global information in computation of the multiscale basis functions will also potentially improve the method.

Regarding the multiscale adjoint model, further application to real-field models and more complex well placement problems are possible continuations. This will involve both investigation of further speedups in the preprocessing step, as well as investigation of alternative, possibly more rigorous, optimization heuristics. As mentioned in the previous chapter, further combinations with more sophisticated transport grids may provide both more accurate simulations and better search regions for repositioning of the wells.

Finally, as indicated in the previous chapter, there are several openings for further research regarding the grid coarsening strategy using amalgamation of fine grid cells. One direction is to further develop heuristic algorithms, and algorithmic improvements and options. Here we think of better, perhaps more specialized algorithms for the coarsening steps suited for particular purposes. Next, extensions to more sophisticated grids, as hybrid and hierarchical grids, grids with specialized resolution for specialized purposes, and grids meeting special requirements, may be investigated. Finally, more research is suggested to develop guidelines for selecting type of coarsening and settings of the coarsening parameters.

A comparison of our suggested coarsening strategy to different regridding and upscaling strategies is a highly relevant topic for future work. Up to this point, no direct comparison is available.

In addition, more rigorous numerical evaluation of the existing ideas in a fully multiscale setting is of interest. Suggested follow-up work here includes actual application of the coarsening strategy for transport solvers on geological models, used in combination with a multiscale pressure solver. Work in this direction would reveal if the anticipated potential of the proposed combination fulfills the expectations.

Bibliography

[1] J.E. Aarnes, T. Gimse, and K.A. Lie. An introduction to the numerics of flow in porous media using Matlab. In G. Hasle, K.A. Lie, and E. Quak, editors, *Geometrical Modeling, Numerical Simulation and Optimisation: Industrial Mathematics at SINTEF*, pages 265–306. Springer Verlag, Berling Heidelberg New York, 2007.

[2] J.E. Aarnes, V. Kippe, K.A. Lie, and A.B. Rustad. Modelling of multiscale structures in flow simulations for petroleum reservoirs. In *Geometric modelling, numerical simulation, and optimization: applied mathematics at SINTEF*, pages 307–360. Springer, Berlin, 2007.

[3] 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 Modeling & Simulation*, 5(2):337–363, 2006.

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

[5] K. Aziz and A. Settari. *Petroleum Reservoir Simulation.* Elsevier Applied Science Publishers, London and New York, 1979.

[6] G.I. Barenblatt, I.P. Zheltov, and I.N. Kochina. Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks. *Journal of Applied Mathematics and Mechanics USSR*, 24(5):1286–1303, 1960.

[7] J.W. Barker and S. Thibeau. A critical review of the use of pseudorelative permeabilities for upscaling. *SPE Reservoir Engineering*, 12(2):138–143, 1997.

[8] J. Bear. *Dynamics of fluids in porous media.* American Elsevier, New York, 1972.

[9] S.H. Begg, R.R. Carter, and P. Dranfield. Assigning effective values to simulator gridblock parameters for heterogeneous reservoirs. *SPE Reservoir Engineering*, 4(4):455–463, 1989.

[10] D. Braess. *Finite Elements: Theory, fast solvers and applications in solid mechanics.* Cambridge University Press, 1997.

[11] L.V. Branets, S.S. Ghai, S.L. Lyons, and X.-H. Wu. Review article: Challenges and technologies in reservoir modeling. *Communications in Computational Physics*, 6(1):1–23, 2009.

[12] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods.* Number 15 in Texts in Applied Mathematics. Springer-Verlag, 1994.

[13] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*, volume 15 of *Springer Series in Computational Mathematics.* Springer-Verlag, New York, 1991.

[14] F. Brezzi, K. Lipnikov, and M. Shashkov. Convergence of mimetic finite difference methods for diffusion problems on polyhedral meshes. *SIAM Journal on Numerical Analysis*, 43:1872–1895, 2005.

[15] F. Brezzi, K. Lipnikov, and V. Simoncini. A family of mimetic finite difference methods on polygonial and polyhedral meshes. *Mathematical Models and Methods in Applied Sciences*, 15:1533–1553, 2005.

[16] A. Castellini, M.G. Edwards, and L.J. Durlofsky. Flow based modules for grid generation in two and three dimensions. In *Proceedings of 7th European Conference on the Mathematics of Oil Recovery, Baveno, Italy, Sept. 5–8*, 2000.

[17] Z. Chen and T.Y. Hou. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Mathematics of Computation*, 72(242):541–576, 2003.

[18] Z. Chen, G. Huan, and Y. Ma. *Computational Methods for Multiphase Flows in Porous Media (Computational Science and Engineering 2).* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.

[19] M.A. Christie. Upscaling for reservoir simulation. *Journal of Petroleum Technology*, 48(11):1004–1010, 1996.

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

[21] C. Deutch. Calculating effective absolute permeability in sandstone/shale sequences. *SPE Formation Evaluation*, 4(3):343–348, 1998.

[22] L. J. Durlofsky. Coarse scale models of two phase flow in heterogeneous reservoirs: volume averaged equations and their relationship to existing upscaling techniques. *Computational Geosciences*, 2(2):73–92, 1998.

[23] L. J. Durlofsky. Upscaling of geocellular models for reservoir flow simulation: A review of recent progress, 2003. Presented at 7th International Forum on Reservoir Simulation Bühl/Baden-Baden, Germany, June 23–27, 2003.

[24] L. J. Durlofsky. Upscaling and gridding of fine scale geological models for flow simulation, 2005. Presented at 8th International Forum on Reservoir Simulation Iles Borromees, Stresa, Italy, June 20–24.

[25] L. J. Durlofsky, R. C. Jones, and W.J. Milliken. A nonuniform coarsening approach for the scale-up of displacement processes in heterogeneous porous media. *Advances in Water Resources*, 20(5–6):335–347, 1997.

[26] W. E and B. Engquist. The heterogeneous multiscale methods. *Communications in Mathematical Sciences*, 1(1):87–132, 2003.

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

[28] M. Evazi and H. Mahani. Generation of Voronoi grid based on vorticity for coarse-scale modeling of flow in heterogeneous formations. *Transport in Porous Media*, 83(3):541–572, 2010.

[29] M. Evazi and H. Mahani. Unstructured-coarse-grid generation using background-grid approach. *SPE Journal*, 15(2):326–340, 2010.

[30] C. L. Farmer. Upscaling: a review. *International Journal for Numerical Methods in Fluids*, 40(1–2):63–78, 2002.

[31] C.L. Farmer. *Mathematical Methods and Modelling in Hydrocarbon Exploration and Production*, volume 7 of *Mathematics in Industry*, chapter Geological Modelling and Reservoir Simulation, pages 119–212. Springer Berlin Heidelberg, 2005.

[32] M.H. Garcia, A.G. Journel, and K. Aziz. Automatic grid generation for modeling reservoir heterogeneities. *SPE Reservoir Engineering*, 7(6):278–284, 1992.

[33] M. Handels, M.J. Zandvliet, D.R. Brouwer, and J.D. Jansen. Adjoint-based well-placement optimization under production constraints. *SPE Journal*, 13(4):392–399, 2008.

[34] V.L. Hauge, J.E. Aarnes, and K.A. Lie. Operator splitting of advection and diffusion on non-uniformly coarsened grids. In *Proceedings of ECMOR XI – 11th European Conference on the Mathematics of Oil Recovery*, Bergen, Norway, 8–11 September 2008.

[35] C. He and L.J. Durlofsky. Structured flow-based gridding and upscaling for modeling subsurface flow. *Advances in Water Resources*, 29(12):1876–1892, 2006.

[36] L. Holden and B.F. Nielsen. Global upscaling of permeability in heterogeneous reservoirs; the output least squares (OLS) method. *Transport in Porous Media*, 40(2):115–143, 2000.

[37] U. Hornung. *Homogenization and Porous Media*. Springer-Verlag, New York, 1997.

[38] T.Y. Hou and X-H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of Computational Physics*, 134:169–189, 1997.

[39] F. Jahn, M. Cook, and M. Graham. *Hydrocarbon Exploration and Production*, volume 46 of *Developments in Petroleum Science*. Elsevier Publishers, 2008.

[40] P. Jenny, S.H. Lee, and H.A. Tchelepi. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *Journal of Computational Physics*, 187:47–67, 2003.

[41] V.V. Jikov, S.M. Kozlov, and O.A. Oleinik. *Homogenization of differential operators and integral functionals*. Springer-Verlag, New York, 1994.

[42] M. Karimi-Fard and A. Firoozabadi. Numerical simulation of water injection in 2d fractured media using discrete-fracture model. In *SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, 30 September–3 October*, 2001. Paper 71615.

[43] H. Kazemi. Pressure transient analysis of naturally fractured reservoirs with uniform fracture distribution. *SPE Journal*, 9(4):451–462, 1969.

[44] M.J. King. Recent advances in upgridding. *Oil & Gas Science and Technology - Rev. IFP*, 62(2):195–205, 2007.

[45] P.R. King. The use of renormalization for calculating effective permeability. *Transport in Porous Media*, 4(1):37–58, 1998.

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

[47] M. Krotkiewski, I.S. Ligaarden, K.A. Lie, and D.W. Schmid. On the importance of the Stokes–Brinkman equations for computing effective permeability in carbonate karst reservoirs. *Communications in Computational Physics*, 2010. Submitted.

[48] L.W. Lake, E. Kasap, and M. Shook. *Pseudofunctions – The key to practical use of reservoir description.* North Sea Oil and Gas Reservoirs – II. Graham and Trotman, London, 1999.

[49] S.H. Lee, H. Zhou, and H.A. Tchelepi. Adaptive multiscale finite-volume method for nonlinear multiphase transport in heterogeneous formations. *Journal of Computational Physics*, 228(24):9036–9058, 2009.

[50] K.A. Lie, S. Krogstad, I.S. Ligaarden, J.R. Natvig, H.M. Nilsen, and B. Skaflestad. Discretisation on complex grids – open source MATLAB implementation. In *Proceedings of ECMOR XII – 12th European Conference on the Mathematics of Oil Recovery*, Oxford, UK, 6–9 September 2010.

[51] I.S. Ligaarden, M. Krotkiewski, K.A. Lie, M. Pal, and D.W. Schmid. On the Stokes–Brinkman equations for modeling flow in carbonate reservoirs. In *Proceedings of ECMOR XII – 12th European Conference on the Mathematics of Oil Recovery*, Oxford, UK, 6–9 September 2010.

[52] H. Mahani and A.H. Muggeridge. Improved coarse grid generation using vorticity. In *14th Europec Biennial Conference, Madrid, Spain, 13–16 June*, 2005.

[53] J.L. Mallet. *Geomodeling.* Oxford University Press, Inc., New York, NY, USA, 2002.

[54] J.L. Mallet. *Numerical Earth Models*, volume III of *EAGE Education Tour*. EAGE Publications BV, 2008.

[55] MATLAB reservoir simulation toolbox, 2009. `http://www.sintef.no/MRST/`.

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

[57] J.R. Natvig, K.A. Lie, B. Eikemo, and I. Berre. A discontinuous Galerkin method for computing single-phase flow in porous media. *Advances in Water Resources*, 30(12):2424–2438, 2007.

[58] D.W. Peaceman. *Fundamentals of Numerical Reservoir Simulation.* Elsevier Science Inc., New York, NY, USA, 1991.

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

[60] V. Reichenberger, H. Jakobs, P. Bastian, and R. Helmig. A mixed-dimensional finite volume method for two-phase flow in fractured porous media. *Advances in Water Resources*, 29(7):1020–1036, 2006.

[61] J.E. Warren and P.J. Root. The behavior of naturally fractured reservoirs. *SPE Journal*, 3(3):245–255, 1963.

[62] X.H. Wen, L.J. Durlofsky, and M.G. Edwards. Upscaling of channel systems in two dimensions using flow-based grids. *Transport in Porous Media*, 51(3):343–366, 2003.

[63] C.D. White and R.N Horne. Computing absolute transmissibility in the presence of fine-scale heterogeneity. In *SPE Symposium on Reservoir Simulation, 1–4 February, San Antonio, Texas*, 1987.

[64] J. Willems. *Numerical Upscaling for Multiscale Flow Problems.* PhD thesis, Technische Universität Kaiserslautern, 2009.

[65] H. Zhou, S.H. Lee, and H.A. Tchelepi. Multiscale finite volume formulation for the saturation equations. In *SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 2–4 February*, 2009.

Part II

Paper I

A Multiscale Mixed Finite-Element Method for Vuggy and Naturally-Fractured Reservoirs

Astrid Fossum Gulbransen, Vera Louise Hauge, and Knut–Andreas Lie

SPE Journal, Volume 15, Number 2, June 2010, pp. 395–403

Paper II

Coarsening of Three-Dimensional Structured and Unstructured Grids for Subsurface Flow

Jørg Espen Aarnes, Vera Louise Hauge, and Yalchin Efendiev

Advances in Water Resources, Volume 30, Issue 11,
November 2007, pp. 2177–2193

Available online at www.sciencedirect.com

ScienceDirect

Advances in Water Resources 30 (2007) 2177–2193

Advances in Water Resources

www.elsevier.com/locate/advwatres

Coarsening of three-dimensional structured and unstructured grids for subsurface flow ☆

Jørg Espen Aarnes [a,*], Vera Louise Hauge [a], Yalchin Efendiev [b]

[a] *SINTEF ICT, Department of Applied Mathematics, P.O. Box 124 Blindern, N-0314 Oslo, Norway*
[b] *Department of Mathematics, Texas A&M University, College Station, TX 77843, United States*

Received 7 December 2006; received in revised form 16 April 2007; accepted 18 April 2007
Available online 16 May 2007

Abstract

We present a generic, semi-automated algorithm for generating non-uniform coarse grids for modeling subsurface flow. The method is applicable to arbitrary grids and does not impose smoothness constraints on the coarse grid. One therefore avoids conventional smoothing procedures that are commonly used to ensure that the grids obtained with standard coarsening procedures are not too rough. The coarsening algorithm is very simple and essentially involves only two parameters that specify the level of coarsening. Consequently the algorithm allows the user to specify the simulation grid dynamically to fit available computer resources, and, e.g., use the original geomodel as input for flow simulations. This is of great importance since coarse grid-generation is normally the most time-consuming part of an upscaling phase, and therefore the main obstacle that has prevented simulation workflows with user-defined resolution. We apply the coarsening algorithm to a series of two-phase flow problems on both structured (Cartesian) and unstructured grids. The numerical results demonstrate that one consistently obtains significantly more accurate results using the proposed non-uniform coarsening strategy than with corresponding uniform coarse grids with roughly the same number of cells.

Keywords: Two-phase flow; Grid generation; Porous media; Upscaling

1. Introduction

To visualize and quantify fluid flow in heterogeneous subsurface reservoirs (e.g., groundwater and petroleum reservoirs), it is common to perform numerical flow simulations. To this end, statistical descriptions of the subsurface formations are built using geomodeling software tools. Unfortunately the number of cells in the geological grid-model (geomodel) that estimate the spatial distribution of reservoir parameters very often exceed the capabilities of flow simulators. Hence, rather than using the original geomodel as input, current simulators normally take as input coarsened and simplified models derived through an upscaling process. These upscaled models consist of a coarsened grid accompanied by a corresponding set of reservoir parameters. The simulation model then consists of the upscaled geomodel combined with an elliptic (or parabolic) equation modeling pressure and flow velocity, and a set of mass balance equations modeling the transport. In this paper we focus on immiscible flow so that by transport we refer to movement of phases – aqueous, liquid, or vapor – and not dispersion of different components within each phase.

Coarsened grids obtained by upscaling are usually constrained to be on a specific grid format, e.g., corner-point grid format (“logically hexahedral grids”) or PEBI grid format (“orthogonal Voronoi grids”). This is partly due to grid-constraints associated with the numerical methods

☆ The research at SINTEF was partially supported by Shell and the Research Council of Norway under grants 158908/420 and 175962/S30. The research of Y. Efendiev was partially supported by NSF grants DMS-062113 and EIA-0540138 and DOE grant DE-FG02-05ER25669.

* Corresponding author. Tel.: +47 22 06 77 95.
E-mail addresses: Jorg.Aarnes@sintef.no (J.E. Aarnes), Vera.Louise.Hauge@sintef.no (V.L. Hauge), Efendiev@math.tamu.edu (Y. Efendiev).

0309-1708/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advwatres.2007.04.007

employed to discretize the governing system of partial differential equations, in particular the pressure equation, and partly because most techniques for upscaling permeability are designed for grids with hexahedral (shoe-box shaped) grid blocks. The widely used two-point finite volume method, for instance, is designed for so-called *K*-orthogonal grids, meaning that the connections between cell centers are *K*-perpendicular to the cell faces. In general, grid constraints make it very difficult to build grids that capture the important features in the underlying geomodels in an appropriate way. Indeed, upscaled subsurface flow models often fail to capture important small scale structures that have profound impact on the resulting flow regime, such as narrow high-flow channels and shale barriers (low-permeable obstacles).

Grid generation procedures that tune the coarse grid to dominant geological features have been proposed by several authors, see e.g., [13,15,19] and the references therein. The basic goal in these methods is often to generate grids that are more finely gridded in regions of particular importance, e.g., around wells and in high-flow regions. However, although these techniques do offer better resolution in regions of interest, and in general provide more robust upscaled models, they tend to suffer from lack of grid flexibility. Flow-based grid generation approaches, for instance, are often limited to two-dimensional applications. Moreover, because flow paths in heterogeneous formations may be highly irregular, flow-based grid generation approaches generally require a grid-smoothing procedure. Finally, because grid lines in the coarsened grid usually do not align with grid lines in the underlying geomodel grid, it is often necessary to perform resampling of geological data.

In this paper, we propose a semi-automated grid-coarsening strategy for subsurface flow applications that can be applied to both structured and unstructured geomodel grids. The basic idea is to develop a generic all-applicable version of the non-uniform coarsening approach introduced by Durlofsky et al. [10]. Here the coarse grid is generated by selectively grouping cells in the geomodel. We require only that the cells in the coarse grid are connected, have at least some minimum volume, and that the total flow through each coarse cell is bounded above. Because only the transport is computed on this grid, we avoid conventional constraints (e.g., whereas the approach of Durlofsky et al. was based on simply removing grid lines in two-dimensional Cartesian grids, thus obtaining a non-uniform Cartesian grid, we here allow coarse cells with arbitrary shape). The transport is modeled using a two-scale version of a first-order upstream weighted finite volume method. This scheme requires, in principle, only that the grid cells are connected and that the velocity field is mass conservative on this grid.

To model pressure and velocity we employ here a mimetic finite difference method on the geomodel. Hence, we propose that the pressure equation is solved on the geomodel grid, whereas the phase-transport equation is solved on a coarsened grid. This can be justified for oil–water two-phase flow scenarios because the pressure generally changes at a moderate pace so that one can use much larger time-steps for the pressure equation than for the phase-transport equations [8]. However, for flows with strong dynamics, e.g., three phase flow with a separate gas-phase, and for very large geomodels, it may be too computationally expensive to solve the pressure equation directly on the geomodel. For these cases, a more efficient alternative is to use a multiscale method capable of providing mass-conservative velocity fields on the geomodel, e.g., [1,5,14]. The key observation is that a velocity field that is mass conservative on the geomodel is also mass conservative on *any* grid with cells that consist of a connected collection of cells in the geomodel.

We will start by introducing the model problem, which is a model for incompressible and immiscible two-phase flow. Next, in Section 3 we present the non-uniform coarsening algorithm and discuss applicability and limitations and possible implications. We also provide some analysis to give an insight into why the coarsening algorithm provides more accurate simulation results. In Sections 4 and 5 we describe the discretization of the model equations and report the results of the numerical experiments, respectively. The examples range from relatively simple two-dimensional Cartesian models to models with complex channelized heterogeneous structures and unstructured corner-point grid models. Finally, we review the main observations and make some concluding remarks in Section 6.

2. Mathematical model

We consider immiscible and incompressible two-phase flow without gravity and capillary pressure effects. The equations are derived from conservation of mass for each phase:

$$\phi\frac{\partial S_j}{\partial t} + \nabla\cdot v_j = q_j, \tag{1}$$

where the phase velocities v_j are given by Darcy's law:

$$v_j = -\lambda_j(S_j)K\nabla p_j. \tag{2}$$

Here ϕ is the porosity, S_j is the *j*-phase saturation (fraction of the void occupied by phase *j*) and q_j is a source (or sink) term. In Darcy's law, *K* is the permeability tensor, p_j is the phase pressure, and $\lambda_j(S_j) = k_{rj}(S_j)/\mu_j$, where k_{rj} and μ_j are the relative permeability and viscosity of phase *j*, respectively. The relative permeability models the reduced conductivity of a phase due to the presence of other phases and is assumed to be function of the saturations only. The porosity is taken to be constant, i.e., the rock is assumed to be rigid and non-deforming.

Let the two phases be oil and water ($j = \mathrm{o}, \mathrm{w}$). Since we neglect capillary pressure effects so that $\nabla p_\mathrm{o} = \nabla p_\mathrm{w}$, we assume $p_\mathrm{o} = p_\mathrm{w} = p$. Then the Darcy equations without gravity effects combined with conservation of mass yields the *pressure equation*:

$$\begin{aligned} v &= -\lambda(S_{\rm w}, S_{\rm o})K\nabla p, \\ \nabla \cdot v &= q, \end{aligned} \tag{3}$$

where $v = v_{\rm w} + v_{\rm o}$, $\lambda = \lambda_{\rm w} + \lambda_{\rm o}$, and $q = q_{\rm w} + q_{\rm o}$.

Assume now that the two phases fill the void space completely, i.e., that $S_{\rm w} + S_{\rm o} = 1$, and introduce the water *fractional flow* $f_{\rm w}(S_{\rm w}) = \lambda_{\rm w}(S_{\rm w})/\lambda(S_{\rm w},\ 1 - S_{\rm w})$. Then the conservation equation for water, henceforth called the *saturation equation*, reads as follows:

$$\phi\frac{\partial S_{\rm w}}{\partial t} + \nabla \cdot (f_{\rm w} v) = q_{\rm w}. \tag{4}$$

The system of Eqs. (3), (4) 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. Moreover, for ease of notation, we will henceforth drop the w-subscript of $S_{\rm w}$.

3. Grid coarsening strategy

In the current section we seek to develop a grid coarsening strategy based on the following principles (B refers to a cell in the coarsened grid, henceforth called a block):

- The grid should separate high-flow regions from low-flow regions.
- The volume of a cell in the coarse grid should not be too small.
- The total amount of flow through a single cell in the coarse grid should not be too large.

Hence, after separating high-flow regions from low-flow regions, we will refine blocks that are too large, and merge blocks that are too small with a neighboring block. The refinement strategy is based on imposing an upper bound on the total flow through each block rather than a bound on the total volume. Thus, the coarsening algorithm involves two parameters that should be tuned to give the intended degree of coarsening: $N_{\rm L}$ (lower bound on volume of blocks), and $N_{\rm U}$ (upper bound on total amount of flow in blocks). To quantify the amount of flow through each block we introduce the following monitoring function:

$$g(v) = [\log|v| - \min(\log|v|) + 1].$$

The various steps in the coarsening algorithm now reads as follows:

Algorithm 1 (*Non-uniform coarsening of heterogeneous geomodels*).

1: Introduce coloring of cells based on the logarithm of the velocity:

$$\begin{aligned} C_i = &\ \{\text{cells with color } i\} \\ = &\ \{c : m_v + (i-1)D_v < \log|v(c)| < m_v + iD_v\}. \end{aligned}$$

Here $m_v = \min(\log|v|)$ and $D_v = (\max(\log|v|) - m_v)/10$. Create an initial grid where each block B is a connected collection of cells in the fine grid with the same color.

2: If $|B| < N_{\rm L}|\Omega|/N$, then merge B with a neighboring block B' for which

$$\begin{aligned} &\left|\frac{1}{|B|}\int_B g(v){\rm d}x - \frac{1}{|B'|}\int_{B'} g(v){\rm d}x\right| \\ &\quad \leqslant \left|\frac{1}{|B|}\int_B g(v){\rm d}x - \frac{1}{|B''|}\int_{B'} g(v){\rm d}x\right| \end{aligned}$$

for all other neighboring blocks B''. Here Ω is the computational domain (the reservoir), N is the total number of cells in the original grid, and $N_{\rm L}$ is a lower bound on the number of average-sized cells that a block B in the coarse grid may consist of. Hence, B is adjoined with a neighboring block subject to flow of similar magnitude as that which B is subject to.

3: Refine a block B if

$$\int_B g(v){\rm d}x > \frac{N_{\rm U}}{N}\int_\Omega g(v){\rm d}x. \tag{5}$$

Here the right hand side provides an upper bound on the amount of flow allowed to pass through a single block. The refinement strategy consists of the following parts:

(a) Pick an arbitrary cell T_0 in the fine grid that is contained in B.

(b) Find the cell $T_i \subset B$ for which the center of T_i is furthest away from the center of T_0 (T_i will be located along the boundary B).

(c) Define $B' = T_i$ and progressively enlarge B' by successively adding cells surrounding T_i until the upper bound in (5) is reached.

(d) Define $B = B\backslash B'$ and refine B further if (5) still holds.

4: Repeat Step 2 and terminate.

Step 4 does not have significant impact on the accuracy of obtained solutions, and may therefore be skipped, but it tends to reduce the number of blocks by 30–50%. This is because the refinement strategy in Step 3 produces cells with volume less than $N_{\rm L}|\Omega|/N$. Although the final cells may satisfy (5) we do *not* repeat Step 3.

The numerical results in Section 5 indicate that the algorithm is quite robust with respect to the choice of $N_{\rm L}$ and $N_{\rm U}$. Tuning these parameters is quite easy. For instance, as a rule of thumb, to generate a grid with a scale-up factor of N, choose $N_{\rm U} \sim 5N/4$ and $N_{\rm L} \sim N/4$. Moreover, the algorithm is not sensitive to the cells T_0 used in the refinement step because it is only used as a starting point for locating the cell T_i along ∂B. Hence, in our implementation, we simply take T_0 to be the first cell in the list of cells that belong to B.

To illustrate how the algorithm works we consider a two-dimensional test case representing one of the layers (layer 37) in Model 2 used in the 10th SPE Comparative Solution Project [9], a project used to test and validate upscaling methods. The logarithm of the horizontal permeability is shown in the upper left hand plot of Fig. 1. Here we see the trace of high-permeability channels on a low-permeable background. Fig. 1 also shows plots of the logarithm of the velocity depicted in the upper right plot mapped onto a non-uniform coarse grid with 208 cells and a uniform Cartesian grid with 220 cells.

Fig. 2 shows the grids obtained in each step of the non-uniform coarsening algorithm with $N_{\mathrm{L}} = 15$ and $N_{\mathrm{U}} = 80$, giving an upscaling factor of about 65. In each plot each block is assigned a random color. After the initial step we see a myriad of small cells due to oscillations in the velocity, which, in turn, are caused by oscillations in the permeability. These bits and pieces are generally too small to have significant impact on the flow regime. Hence, they are merged with a neighboring block to obtain the next grid depicted in the upper-right plot. Here we see that some of the blocks stretch across large portions of the domain, hence clearly being too large to model an advancing saturation front. Upon completion of the refinement step we obtain the grid depicted in the lower-left plot. In this grid we may spot some very small blocks consisting of only a few cells. After the blocks consisting of less than 15 cells are merged with a neighboring block, we obtain the final grid depicted in the lower-right plot. Note that the algorithm delivers an unstructured grid even if the fine grid is structured, and that the blocks have very irregular shapes.

The fact that we use flow information to generate the coarse grid may give the impression that the grid is case-specific, and therefore needs to be recomputed when flow conditions change, e.g., when well-rates, boundary conditions, or well-configurations change. But because high-flow regions represent high-permeability zones with good large scale connectivity, it is generally *not* necessary to generate a new coarse grid, even if flow conditions change significantly. This will be demonstrated in Section 5.4. Note also that since the coarsened grid is employed only to compute the flow transport, one does not need to solve an additional single-phase flow problem. The grid is generated after the first pressure step in the sequential time-stepping loop, i.e., before the first saturation step.

An alternative coarsening strategy that does not require flow information is to base the initial coloring on the logarithm of permeability. This option has been tested, and found to be less viable. The reason for this is that low-permeability cells that occur in high-permeability regions may be subject to large amount of flow, and should therefore not be separated from the high-permeability region. Similarly, high-permeability patches may be isolated inside low-permeable regions so that they are not subject to significant flow.

It should also be mentioned that a potential drawback with flow based coarsening strategies is that the grid will

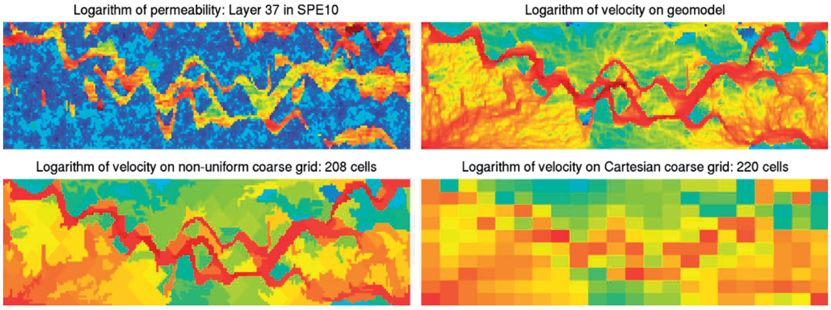


Fig. 1. Top-left: Logarithm of permeability (60 × 220 Cartesian grid). Top-right: Logarithm of $|v|$ obtained for a five-spot. Bottom-left: Logarithm of $|v|$ mapped onto a non-uniform coarse grid with 208 cells. Bottom-right: Logarithm of $|v|$ mapped onto a Cartesian grid with 220 cells.

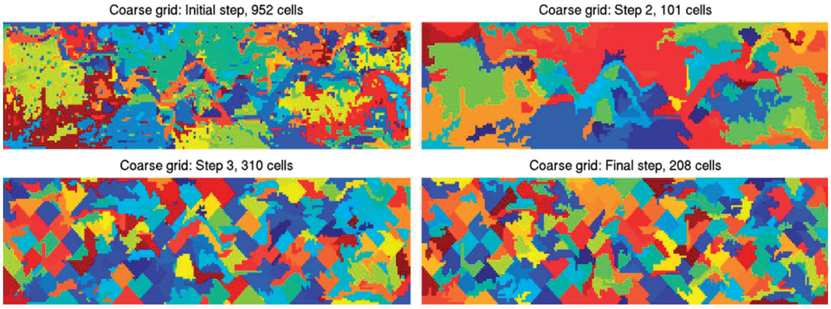


Fig. 2. Grid after each step when the non-uniform coarsening algorithm is applied to layer 37 in the SPE10 model with $N_{\mathrm{L}} = 15$ and $N_{\mathrm{U}} = 80$.

in general have to be regenerated for every permeability realization. However, the current algorithm is very fast – the time it takes to generate a new coarse grid is negligible relative to the time it takes to perform a single simulation. We therefore believe that this type of upscaling strategy should be a valuable tool, also for an analyst trying to complete an uncertainty study.

3.1. Discussion

The grid-coarsening approach described above is proposed as a remedy for situations where conventional upscaling is inaccurate or it is prohibitively computationally expensive to perform the entire simulations directly on the geomodel grid. The main objective is to exploit information in high-resolution geological models in an optimal way. Here we discuss the applicability and limitations of the proposed approach as well as possible implications and relations to other methods. Since a key to our approach is the ability to compute mass conservative velocity fields with high-resolution, e.g., directly on the geomodel, we discuss first the validity of this assumption.

As an intermediate solution between upscaling and direct simulation on geomodels, the possibility of using multiscale methods [1,5,14] to provide high-resolution velocity fields has been proposed. These methods may be used to generate accurate mass conservative velocity fields on high-resolution geomodel grids at the cost of a conventional flow-based upscaling method. They may therefore be regarded as a more robust and flexible alternative to upscaling for the pressure equation (see, e.g., [3,16] for a discussion). In particular it has been shown that the multiscale mixed finite element method [7,1] is very flexible with respect to handling very complex grids [3]. This method is currently being extended to the three-phase black-oil model, which is the industry standard for reservoir simulation. Thus, with ongoing efforts, we believe that multiscale methods will, in time, provide a viable tool for computing mass conservative velocity fields directly on real-field geomodels.

Multiscale methods for subsurface flow simulation have for the most part been used to solve the pressure equation and combined with a traditional method for solving the saturation equation(s) on the underlying fine grid. Unfortunately, even with streamline methods, computing the phase transport on the geomodel may create a bottle-neck in the simulations that prevents high-speed-up factors relative to sequential time-stepping approaches where the pressure equation is solved with a standard numerical method directly on the geomodel. Thus, whether we solve the pressure equation on the fine grid, or employ a multiscale method to incorporate subgrid effects into a set of coarse scale equations, there is a need to address how the phase transport can be computed in an optimal way with respect to solution accuracy and computational efficiency.

Here we have proposed creating an upscaled model only for the saturation equation, i.e., to generate a coarse grid that resolves more accurately underlying flow patterns than traditional coarse grids used in reservoir simulation, see Figs. 1 and 4 and the analysis in Section 3.2 below. Although many authors have proposed coarsening strategies for subsurface flow applications, see, e.g., [13,15,19] and the references therein, the current approach is primarily motivated by [10]. In particular, it was observed in [10] that introducing a grid which is more finely gridded in high-flow regions allows capturing more accurately flow quantities of interest, such as production characteristics, without resorting to multiphase upscaling. The main significance of our approach is that it is all-applicable. Hence, whereas the non-uniform coarsening strategy in [10] applies primarily to Cartesian-like geomodels with high-flow channels that are aligned with the grid, our approach applies to all types of grids and essentially all types of heterogeneous structures.

In the current paper, we have chosen to consider a simplified model that does not include effects from gravity and capillary forces. In general, flow-based grid coarsening approaches based on separating high- and low-flow regions, or on tuning the grid to resolve high-flow regions, are designed for modeling flows where heterogeneity, rather than gravity or capillary forces, dominates the flow patterns. This does not mean that the current approach is not applicable to problems with non-zero gravity or capillary forces. Indeed, most flow scenarios on the scale of an oil reservoir are affected by both gravity and capillary forces, but the flow is usually primarily driven by pressure (viscous forces), meaning that flow patterns are generally dominated by heterogeneity. The current algorithm is therefore expected to work well for this type of flow scenario.

If flow patterns are dominated by gravity, then we do not expect that solutions obtained using flow-based grids will be much more accurate than solutions obtained using standard coarse grids. Flow-based gridding implicitly assumes that the high-flow regions remain the same throughout a simulation. For gravity dominated flows the flow patterns may change significantly during simulation so that regions initially subject to significant flow may be subject to little flow at later times, and vice-versa. Thus, tuning the grid to high-flow regions at initial time may be of little value. On the other hand, the coarsening strategy proposed in this paper should not be regarded only as a tool to achieve higher accuracy, it may also be used as a generic tool to coarsen complex unstructured geomodels. Moreover, although the algorithm has not been applied to gravity dominated flows, we believe that it will not, on average, produce less accurate results than uniform coarsening strategies.

Upscaling of capillary dominated flows is a problem of a different nature. Indeed, capillary dominated flows are more strongly coupled, and the sequential splitting where one solves the pressure equation and saturation equation sequentially is no longer justified. However, the proposed coarsening algorithm is intended for field scale (10 m–10 km) simulations, whereas flows that are dominated by capillary forces occur on a much finer scale (dm–m).

Finally we would like to remark that combining the proposed coarsening approach with a multiscale method for the pressure equation brings us closer to having an Earth Model shared between reservoir engineers and reservoir geologists [15]. This means that the geological model provided by geologists is used as input for the reservoir simulator. At present geoscientists and reservoir engineers are forced to work with different descriptions of the subsurface. To bridge the gap it is necessary that the reservoir engineer can select semi-automatically numerical technology and grid resolution for the simulations at run-time to fit available computer resources and project requirements. Having a "Shared Earth Model" will, apart from giving the reservoir engineer a better description of the subsurface, offer the geoscientist the ability to validate the geomodel before it is sent to the reservoir engineer. This can simplify and accelerate reservoir simulation workflows considerably, and thereby allow, e.g., oil-companies to save many man-hours. Indeed, subsurface geological models are not static, they are analyzed, iterated and rebuilt as project requirements and business decisions change, or when uncertainty studies indicate that the model does not give a representative picture of the reservoir flow regime.

3.2. Analysis

Recall that Algorithm 1 groups high and low flow regions separately. The velocity field inside a single grid block will therefore be of nearly the same magnitude, i.e., the velocity will have small variations in each coarse block. We will now show that this implies also that the saturation variation is small within each coarse block in the non-uniform coarse grid, i.e., that the saturation is well resolved on the coarse grid.

Consider the equation for the time-of-flight function

$$v \cdot \nabla \tau = \phi.$$

The time-of-flight functions describe travel times along flow trajectories (streamlines). The streamlines $\psi = x(\tau)$ are trajectories along which v is a tangential vector (see e.g., [2] for details). Each streamline eminates from an injection well and terminate at a production well. Along streamlines, the corresponding time-of-flight functions are defined by

$$\frac{\mathrm{d}\tau}{\mathrm{d}s} = \frac{\phi}{|v|},$$

where s is arclength distance. Moreover, between wells we have

$$\frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial \tau} = 0. \tag{6}$$

Observe now that since we employ a velocity computed on the fine grid for the coarse grid simulations, we do not really alter the streamlines when coarsening the grid. Rather we introduce a coarse grid approximation to the time-of-flight functions. Thus, conceptually we may think of our approach as solving (4) by solving (6) along one-dimensional streamlines where the time-of-flight function τ is replaced with a coarse grid approximation τ_0. Hence, one should obtain an accurate coarse grid saturation solution S_0 away from shocks if the projection τ_0 of τ onto the coarse grid is close to τ. Moreover, if the time-of-flight functions are close to each other, then the shock locations also remain close to each other. Thus, although the L_∞ norm of $S - S_0$ can be of order O(1) in vicinity of shock regions, the L^p norm of $S - S_0$ is small because of the small volume of this region.

To prove that τ_0 approximates τ, we define $\delta(x) = \tau(x) - \tau_0(x)$, i.e., $\delta(x)$ is the time lag of reaching the point x with coarse-scale velocity. We will show that the path traversed during this time lag is small due to small variations of the velocity field within the coarse block. Thus, if the velocity does not degenerate, the time lag $\delta(x)$ will remain small.

Denote v_0 to be the coarse-scale velocity that determines τ_0. Or more precise, v_0 is the projection of v onto the coarse grid and τ_0 is the corresponding time-of-flight function. Next, introduce two streamlines ψ and ψ_0 that emanate from the same point, but with velocity v and v_0, respectively. The corresponding time-of-flight functions satisfy

$$x(\tau) = x, \quad x_0(\tau_0) = x, \tag{7}$$

where $x(t)$ and $x_0(t)$ are functions that trace the streamline trajectories, defined by $\mathrm{d}x/\mathrm{d}t = v$ and $\mathrm{d}x_0/\mathrm{d}t = v_0$, respectively. It now follows from (7) that

$$\int_0^\tau v(x(t))\mathrm{d}t = \int_0^{\tau_0} v_0(x_0(t))\mathrm{d}t,$$

and hence that

$$\int_\tau^{\tau_0} v_0(x_0(t))\mathrm{d}t = \int_0^\tau (v(x(t)) - v_0(x_0(t)))\mathrm{d}t. \tag{8}$$

To see that the right hand side of (8) is small, note that

$$\left|\int_0^\tau (v(x(t)) - v_0(x_0(t)))\mathrm{d}t\right| \leq \int_0^\tau |v(x(t)) - v(x_0(t))|\mathrm{d}t + \int_0^\tau |v(x_0(t)) - v_0(x_0(t))|\mathrm{d}t. \tag{9}$$

If $|v - v_0|$ is uniformly small, then the last integral on the right hand side is small. Moreover, one can also easily show that if $|v - v_0|$ is uniformly small, then flow trajectories that emanate from the same point remain close to each other. Consequently, we have that when $|v - v_0|$ is uniformly small, then also the first term on the right hand side (9) is small, and hence the whole right hand side of (9) is small. On the other hand, the left hand side of (8) is equal to $x_0(\tau_0) - x_0(\tau)$, or the distance traversed from time τ_0 to time τ with the velocity v_0. Thus, (8) implies that this distance is small. Consequently, if v_0 is non-degenerate, then the time lag $\delta(x) = \tau(x) - \tau_0(x)$ will also remain small.

4. Numerical discretization

4.1. The pressure equation

To discretize (3) we first need to prescribe boundary conditions. In this paper we impose no-flow boundary conditions ($v \cdot n = 0$ on $\partial\Omega$) for simplicity. Now denote the fine grid (the geomodel) by $\mathscr{T} = \{T\}$, and define the following function spaces:

$$H^{\text{div}}(T) = \left\{ v \in L^2(T)^d : \nabla \cdot v \in L^2(T) \right\},$$
$$H_0^{\text{div}}(\mathscr{T}) = \{ v \in H^{\text{div}}(\cup_{T\in\mathscr{T}} T) : v \cdot n = 0 \quad \text{on } \partial\Omega \}.$$

Next we introduce the following bilinear forms:

$$b(\cdot,\cdot) : H_0^{\text{div}}(\mathscr{T}) \times H_0^{\text{div}}(\mathscr{T}) \to \mathscr{R}$$
$$b(u,v) = \sum_{T\in\mathscr{T}} \int_T u \cdot K^{-1} v \mathrm{d}x$$
$$c(\cdot,\cdot) : H_0^{\text{div}}(\mathscr{T}) \times L^2(\Omega) \to \mathscr{R}$$
$$c(v,p) = \sum_{T\in\mathscr{T}} \int_T p \nabla \cdot v \mathrm{d}x$$
$$d(\cdot,\cdot) : H_0^{\text{div}}(\mathscr{T}) \times L^2(\partial\mathscr{T}) \to \mathscr{R}$$
$$d(v,\pi) = \sum_{T\in\mathscr{T}} \int_{\partial T} \pi v \cdot n_T \mathrm{d}s.$$

Here n_T is the unit normal on ∂T pointing outward.

In a so-called hybrid formulation [6] of (3) with prescribed no-flow boundary conditions one finds a unique triplet of functions $(v,p,\pi) \in H_0^{\text{div}}(\mathscr{T}) \times L^2(\Omega) \times L^2(\partial\mathscr{T} \setminus \partial\Omega)$ such that

$$\begin{aligned} &b(u,v) - c(u,p) + d(u,\pi) = 0, \quad \forall u \in H_0^{\text{div}}(\mathscr{T}), \\ &c(v,q) = (f,q), \quad \forall q \in L^2(\Omega), \\ &d(v,\mu) = 0, \quad \forall \mu \in L^2(\partial\mathscr{T} \setminus \partial\Omega). \end{aligned} \tag{10}$$

Here p represents pressure, v represents velocity, and π is a Lagrange multiplier used to enforce mass conservation. Moreover, $\partial\mathscr{T} = \cup_{T\in\mathscr{T}} \partial T$ and $(\cdot,\cdot)$ is the L^2 inner product.

4.1.1. Mimetic finite-difference method (FDM)

Mimetic FDMs [12,11] may be regarded as finite-difference or finite-volume versions of mixed finite element methods (FEMs). Since mimetic FDMs are quite new, we describe first the hybrid formulation of corresponding mixed FEMs. In the hybrid formulation one selects finite-dimensional subspaces $V \subset H_0^{\text{div}}(\mathscr{T})$, $U \subset L^2(\Omega)$, and $\Pi \subset L^2(\partial\mathscr{T} \setminus \partial\Omega)$, and seeks $(v,p,\pi) \in V \times U \times \Pi$ such that (10) holds for all $(u,q,\mu) \in V \times U \times \Pi$. Here each approximation space is spanned by a particular set of basis functions. For instance, V is spanned by a set of basis functions $\{\psi_i^m \in H^{\text{div}}(T_m) : T_m \in \mathscr{T}, i = 1,2,\ldots,N_m\}$, where ψ_i^m is supported in T_m only. Thus, since $b(\psi_i^m, \psi_j^n)$ is nonzero only if $n = m$, we have

$$\begin{aligned} b\left(\sum_{i,m} u_i^m \psi_i^m, \sum_{i,m} v_i^m \psi_i^m\right) &= \sum_{m,i,j} u_i^m v_j^m b(\psi_i^m, \psi_j^m) \\ &= \sum_m \mathbf{u_m^T B_m v_m}, \quad \mathbf{u_m}, \mathbf{v_m} \in \mathscr{R}^{\mathrm{N_m}}, \end{aligned} \tag{11}$$

where $\mathbf{u}_m = [u_i^m]$, $\mathbf{v}_m = [v_i^m]$, and $\mathbf{B}_m$ is a local matrix associated with T_m. It is important to observe that by enumerating the velocity unknowns (basis-functions) in a cell-wise manner, the mass-matrix that stems from $b(u,v)$ in (10) has a block-diagonal structure, where the local matrices $\mathbf{B}_m$ are the block-diagonals. This allows the hybrid system to be reduced to a symmetric and positive definite linear system, which is in general easier to solve than the original system which is indefinite.

The mimetic FDM formulation is equivalent, except that the subspace V in $H_0^{\text{div}}(\mathscr{T})$ is replaced by a discrete subspace $M \subset L^2(\partial\mathscr{T})$, and the associated bilinear form $b(\cdot,\cdot)$ is replaced by a bilinear form $m(\cdot,\cdot)$ that acts on $L^2(\partial\mathscr{T}) \times L^2(\partial\mathscr{T})$. The basic idea is to introduce means for evaluating $b(\cdot,\cdot)$ in an approximate sense without having explicit representations of the velocity in each cell. In particular, instead of seeking a velocity field defined over each element T, one seeks a set of fluxes defined over the cell faces ∂T.

The bilinear form $m(\cdot,\cdot)$ employed here is defined so that the resulting method reduces to a first order Raviart–Thomas mixed FEM (RT0) [18] when the grid is Cartesian and the permeability is scalar and cell-wise constant. In particular, we associate here (as in RT0) a basis function ψ_i^m with each face F_i^m of every grid cell T_m. The velocity unknown v_i^m corresponding to F_i^m will represent net velocity across F_i^m in the direction of the unit normal n_i^m to F_i^m pointing out of T_m. If we now expand u and v in the basis $\{\psi_i^m\}$:

$$u = \sum_{i,m} u_i^m \psi_i^m \quad \text{and} \quad v = \sum_{i,m} v_i^m \psi_i^m$$

and assume that $m(\psi_i^m, \psi_j^n)$ is nonzero only if $n = m$, then

$$m(u,v) = \sum_{m,i,j} u_i^m v_j^m m(\psi_i^m, \psi_j^m) = \sum_m \mathbf{u}_m^T \mathbf{M}_m \mathbf{v}_m, \quad \mathbf{u}_m, \mathbf{v}_m \in \mathscr{R}^{N_m}, \tag{12}$$

for a given matrix $\mathbf{M}_m$ associated with T_m. Hence, contrary to (11) where $\mathbf{B}_m$ was defined by the bilinear form $b(\cdot,\cdot)$, the bilinear form $m(\cdot,\cdot)$ is defined by the local matrices $\mathbf{M}_m$.

In the implementation we employ only the inverse of $\mathbf{M}_m$. We therefore provide only a formula for computing a matrix $\mathbf{W}_m$ that represents the inverse of $\mathbf{M}_m$ [12]. To this end, we define the following auxiliary matrices:

$\mathbf{N}_m$ – matrix whose ith row is defined by

$$\mathbf{n}_{m,i} = \frac{1}{|F_i^m|} \int_{F_i^m} (n_i^m)^T \mathrm{d}s,$$

$\mathbf{C}_m$ – matrix whose ith row is defined by (x_m is the center of T_m)

$$\mathbf{c}_{m,i} = \int_{F_i^m} (x - x_m)^T \mathrm{d}s,$$

$\mathbf{Z}_m$ – matrix whose columns form an orthonormal basis for the column space of $\mathbf{C}_m$.

Then the inverse matrix $\mathbf{W}_m$ is defined by

$$\mathbf{W}_m = \frac{1}{|T_m|}\mathbf{N}_m K \mathbf{N}_m^T + \frac{2\mathrm{trace}(K)}{|T_m|}\left(\mathbf{I} - \mathbf{Z}_m\mathbf{Z}_m^T\right). \tag{13}$$

This matrix is symmetric and positive definite, and hence ensures that $m(\cdot,\cdot)$ defines an inner-product on $M \times M$.

We would like to note that the current approach to modeling subsurface flow does not require using a mimetic FDM to solve the pressure equation. Any method that provides a mass-conservative velocity field on the geomodel, or at least on some given fine grid, may be used. The primary reason why we have chosen to employ a mimetic FDM is the natural ability to handle complex grid geometries, as well as full-tensor permeabilities [4]. Indeed, geomodels that model real oil or groundwater reservoirs are often significantly more complex than the corresponding upscaled simulation models. Hence, when discretizing the pressure equation directly on a geomodel it is particularly important that a flexible numerical method is employed. For a discussion of the pros and cons of various discretization techniques for (3) on geological models, we refer the reader to [3], in which a multiscale method for corner-point grids, the industry-standard in reservoir simulation, is proposed.

4.2. The saturation equation

To discretize (4) on grids where each block consists of a connected collection of cells in the original fine grid we employ an upstream weighted finite volume method with respect to fluxes obtained on the fine grid. That is, instead of using only the net flux for each interface in the coarse grid and a standard upstream weighted finite volume method with respect to the coarse grid fluxes, we utilize the subgrid resolution in the velocity.

Hence, let each grid block B_m consist of a connected collection of cells in the fine grid and denote non-degenerate interfaces in the fine grid by $\gamma_{ij} = \partial T_i \cap \partial T_j$. The discrete system of equations for the saturation equation now reads:

$$S_m^{n+1} = S_m^{n+1} + \frac{\Delta t}{\int_{B_m}\phi \mathrm{d}x}\left[\int_{B_m} q_w(S^{n+1})\mathrm{d}x - \sum_{\gamma_{ij}\subset\partial B_m} V_{ij}(S^{n+1})\right]. \tag{14}$$

Here S_m is the net saturation in B_m and

$$V_{ij}(S) = \max\{v_{ij}f_w(S|_{T_i}), -v_{ij}f_w(S|_{T_j})\}, \tag{15}$$

where v_{ij} is the Darcy flux across γ_{ij}, i.e., from T_i to T_j. Notice that if there is bi-directional flow across $\Gamma_{kl} = \partial B_k \cap \partial B_l$, i.e., if $\int_{\Gamma_{kl}} \max\{v \cdot n_{kl}, 0\}\mathrm{d}s > 0$ and $\int_{\Gamma_{kl}} \min\{v \cdot n_{kl}, 0\}\mathrm{d}s < 0$ where n_{kl} is the unit normal to Γ_{kl} pointing from B_k to B_l, then the phase-flux across Γ_{kl} is approximated using both S_k and S_l. Thus, although (14) is based on a one-sided upstream scheme, we may obtain a two-sided upstream scheme on the coarse grid.

5. Numerical results

In this section we seek to demonstrate that by using Algorithm 1 for grid coarsening one consistently obtains more accurate saturation solutions than if uniform coarsening is employed. To this end, solutions obtained using non-uniform coarse grids are compared with corresponding solutions obtained using uniform coarse grids, as well as with a reference solution computed using the original grid to perform the transport simulations.

We first apply the non-uniform coarse gridding strategy to a sequence of Cartesian grid models with permeability data from Model 2 of the Tenth SPE Comparative Solution Project [9], henceforth called the SPE10 model. Next we consider a corner-point model with 30 layers. On this model we generate 20 different permeability fields by populating each layer with values drawn from a spatially correlated log-normal distribution. In terms of grid, this model is more complex to coarsen with traditional coarsening strategies because many layers are partially eroded away, giving rise to degenerate cells, and pinch-outs causing so-called non-neighboring connections. However, the grid is given on a logically Cartesian format, and may therefore be partitioned uniformly in index space. We apply this strategy as an alternative to the method proposed in this paper. Finally, we employ a model consisting of four layers from the SPE10 model to assess robustness with respect to degree of coarsening, different well-configurations, and varying flow conditions.

5.1. Experimental setup

The numerical experiments below model incompressible and immiscible two-phase flow without gravity and capillary pressure effects. For the time-stepping, we apply a non-iterated sequential splitting. This means that the pressure equation is solved at the current time-step with total mobility computed using saturations from the previous time-step. Next, the saturations are convected forward in time using the current velocities, and the new saturation values are used to compute the pressure at the next time-step, and so on. The total simulation time for all simulations below is one PVI (pore volume injected). This means that at the end of a simulation the total volume of water that has been injected is equal to the total accessible pore volume in the model.

The dynamic nature of an incompressible two-phase flow system is often quantified by the ratio of the end-point values of the total mobility. Here we define the phase mobilities by

$$\lambda_w = \frac{S^2}{\mu_w}, \quad \lambda_o = \frac{(1-S)^2}{\mu_o}, \quad 0 \leqslant S \leqslant 1. \tag{16}$$

Without gravity, the simulation results depend on the viscosities only as a function of the ratio μ_w/μ_o, henceforth referred to as the viscosity ratio. Thus, for simplicity we assume that $\mu_w = 1$. If we now choose $\mu_o > 1$ then we will get so-called unstable displacement flows for which small scale "fingers" develop and move rapidly at the saturation front. In contrast, choosing $\mu_o < 1$ gives stable displacement flows for which one typically gets sharp saturation fronts because the total mobility is larger behind the front than ahead of the front. In this paper we use $\mu_o = 0.2$, $\mu_o = 1$ and $\mu_o = 10$. These values give rise to both stable and unstable displacement flows so that the simulation results should give an indication of the versatility of doing simulations using the proposed coarsening strategy.

For two-phase flow with dynamic total mobility, different saturation solutions will give different velocity fields. However, since we here want to assess how much we can improve saturation solutions by using a non-uniformly coarsened grid, as opposed to uniform coarsening, we want to eliminate differences caused by having different velocity fields. We therefore use the same (fine grid) velocity field to perform simulations on both grids. That is, for each time step, the velocity solution used to perform the coarse grid simulations is obtained by solving the pressure equation on the fine grid using a reference saturation solution to update the total mobility. The reference solution S_{ref} is obtained by solving the saturation equation on the fine grid.

We assess the accuracy of a saturation solution S using a measure for the overall accuracy, and also a measure for assessing the accuracy of the predicted oil-production at the producers. To measure the overall accuracy, we compute how much the solution deviates from the reference solution in the L^1-norm, divide by the L^1-norm of the reference solution, and integrate in time, i.e.,

$$e(S) = \int_0^1 e(S,t)\mathrm{d}t$$

where

$$e(S,t) = \frac{\|S(\cdot,t) - S_{\mathrm{ref}}(\cdot,t)\|_{L^1(\Omega)}}{\|S_{\mathrm{ref}}(\cdot,t)\|_{L^1(\Omega)}},$$

with time measured in PVI.

Similarly, to quantify the accuracy of the predicted oil-production, we measure the accuracy of the predicted water-cut curve w, showing the fraction of water in the produced fluid. To this end, we compute how much the water-cut curve deviates from the reference water-cut w_{ref} curve in the L^2-norm and divide by the L^2-norm of w_{ref}, i.e.,

$$e(w) = \frac{\|w - w_{\mathrm{ref}}\|_{L^2([0,1])}}{\|w_{\mathrm{ref}}\|_{L^2([0,1])}},$$

again with time measured in PVI.

Finally, because well-models are used only to model well-rates and do not appear explicitly in the saturation equation, we will throughout this section assume that the well rates are fixed. Hence, we assume that the source term q in (3) is given explicitly. Moreover, the source term in (4) will be defined by $q_w(S) = \max(q,0) + f_w(S)\min(q,0)$. In the figures and tables presented in this section we will frequently use UC as an abbreviation for results obtained with uniformly coarsened grids and NUC as an abbreviation for results obtained with non-uniformly coarsened grids.

5.2. Cartesian grids

In this section we consider a series of two- and three-dimensional Cartesian grid models. Each model represents either a single layer in the SPE10 model [9], or a stack of five consecutive layers. The SPE10 model itself consists of a total of 85 layers, where the top 35 layers model a Tarbert formation representing a prograding near-shore environment, and the bottom 50 layers model a fluvial Upper Ness formation with a spaghetti of narrow high-flow channels. The entire model consists of $60 \times 220 \times 85$ cells, each of size 20 ft × 10 ft × 2 ft.

In order to be able to interpret the results in this section, we need to explain some basic characteristics of the heterogeneous structures in the SPE10 model. Both the Tarbert formation and the Upper Ness formation are highly heterogeneous (in each formation the permeability is anisotropic and spans more than 10 orders of magnitude), but the heterogeneous permeability structures are qualitatively different.

- In the Tarbert formation the permeability in each layer is generated stochastically using a spatially correlated log-normal distribution. However, because the mean permeability in the layers varies significantly, one obtains segregated flow scenarios with sharp saturation contrasts from one layer to the next.
- The Upper Ness formation consists of multiple intertwined high-permeable flow-channels through a low-permeable background. Hence, in this model the flow channels will carry the majority of the flow, and therefore cause sharp saturation contrasts between the background structure and the channels.

Fluvial formations, such as the Upper Ness model, are generally very hard to upscale. Indeed, whereas one may obtain good results for the Tarbert formation by using grid blocks that do not cut across the layers (i.e., are one layer thick), upscaling the Upper Ness formation adequately requires that the channels are resolved by the coarse grid. This is very difficult, if not impossible, to accomplish with conventional grid-constrained upscaling strategies.

For the flow simulations that we perform here we employ $\mu_w = 1$ and $\mu_o = 10$, i.e., we consider an unstable displacement process with the same viscosity ratio as in [9]. We also use the same well-configuration as in [9], i.e., a five-spot with a vertical injection well in the middle and vertical production wells at each of the four corners. The total production rate is the same for all producers, and q is assumed to be constant along each well trajectory. Finally the simulations are performed with 20 pressure steps and 10 saturation steps between each pressure step.

5.2.1. *Two-dimensional simulations*

Fig. 3 shows the saturation and water-cut errors for flow simulations on each of the 85 layers in the SPE10 model. Table 1 displays the average saturation and water-cut errors. We see that the quality of the solutions obtained with the non-uniformly coarsened grids is nearly independent of the heterogeneous structures in the model. In contrast, when using uniform coarse grids both the saturation error and water-cut error are much larger for the fluvial Upper Ness formation than for the Tarbert formation.

The varying performance of uniform coarse grids can be explained by the nature of the flow. Because the permeability in each layer of the Tarbert formation is relatively smooth, one obtains saturation fields that are well resolved on the uniform coarse grids. In contrast, the heterogeneity in the layers from the Upper Ness formation creates flow scenarios where a majority of the flow occur in narrow flow channels. This gives sharp saturation differences between the channels and the low-permeable background. Because the uniform coarse grids do not resolve the channels properly (see, e.g., Fig. 4), one will generally smooth the saturation profile, and therefore obtain inaccurate results. For the non-uniform coarse grids, on the other hand, significant smoothing of the saturation profile is avoided because the cells are grouped according to magnitude of flow. However, we note that for the layers from the Tarbert formation the errors for the non-uniform grids are slightly larger. This is not really surprising since both methods perform well and the shape of the grid blocks in the non-uniform grids are more irregular in size and shape.

Table 1
Mean of the errors plotted in Fig. 3

Geomodel	Tarbert formation		Upper ness formation	
Coarsening strategy	NUC	UC	NUC	UC
Mean saturation error	0.1737	0.1357	0.1730	0.3031
Mean water-cut error	0.0246	0.0258	0.0263	0.0875

5.2.2. *Three-dimensional simulations*

Fig. 5 shows the saturation and water-cut errors for flow simulations on 17 models consisting of a stack of five consecutive layers from the SPE10 model. Two sets of simulations are done. In the first set the non-uniform grid is generated by the parameters $N_{\mathrm{L}} = 6$ and $N_{\mathrm{U}} = 20$ and the uniform grid is a Cartesian grid of $15 \times 44 \times 5$ cells, thus resolving the layers also in the vertical direction. The second set consists of a non-uniform grid generated by the parameters $N_{\mathrm{L}} = 25$ and $N_{\mathrm{U}} = 100$ and the uniform grid is a Cartesian grid of $15 \times 44 \times 1$ cells, i.e., the layers are not resolved in the vertical direction.

We notice first that with the non-uniform coarse grids we consistently obtain accurate results. In particular, even though the grids used to obtain the results in Fig. 5b are four to five times coarser than the grids used to obtain the results in Fig. 5a, we notice that the errors depicted in Fig. 5b are only slightly larger than the corresponding errors depicted in Fig. 5b (see also Table 2).

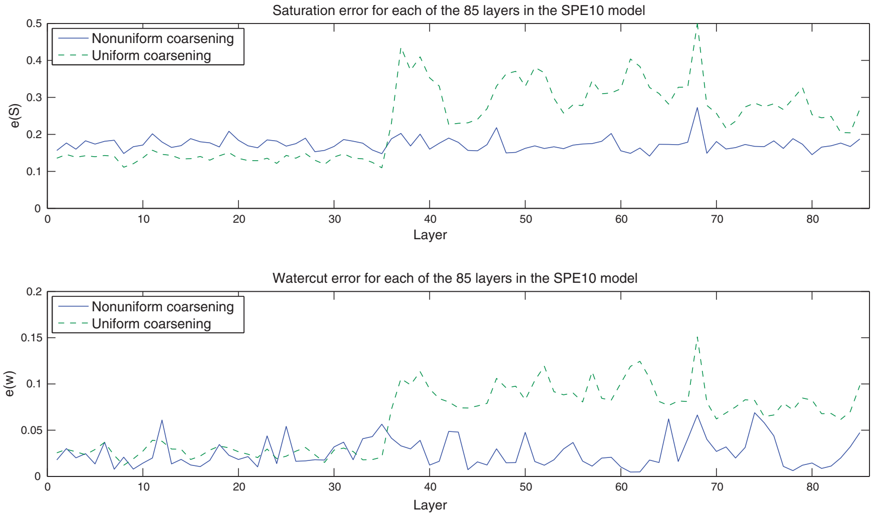


Fig. 3. Saturation and water-cut errors for flow simulations on each of the 85 layers in the SPE10 model. The uniformly coarsened grid is a uniform 15×44 Cartesian grid. The non-uniform grids contain 619–734 cells ($N_{\mathrm{L}} = 6$, $N_{\mathrm{U}} = 20$).

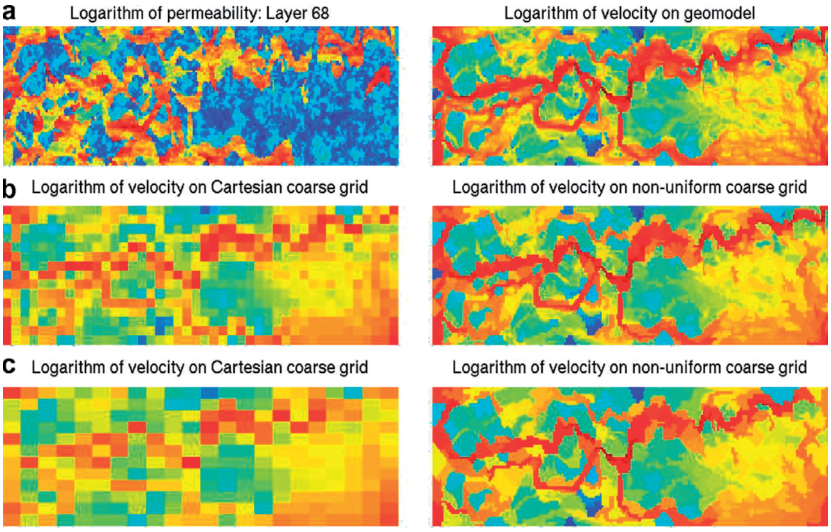


Fig. 4. Logarithm of velocity for a five spot approximated on a sequence of different grids. Notice that the channelized flow pattern is lost in the Cartesian coarse grids, whereas the channels are well resolved on the non-uniform coarse grids, even with only 257 blocks. (a) Geomodel (13 200 cells) and velocity on geomodel grid. (b) Coarse grid: 660 blocks; Coarse grid: 649 blocks. (c) Coarse grid: 264 blocks; Coarse grid: 257 blocks.

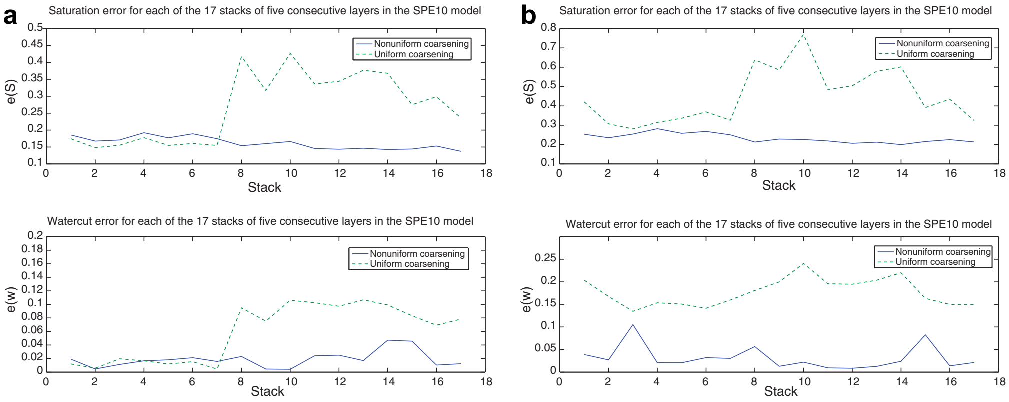


Fig. 5. Saturation and water-cut errors for flow simulations on 17 models consisting of a stack of five consecutive layers from the SPE10 model. (a) UC: $15 \times 44 \times 5$/NUC: 2900–3141 cells. (b) UC: $15 \times 44 \times 1$/NUC: 655–714 cells.

Table 2
Mean of the errors plotted in Fig. 5

Uniform grid	$15 \times 44 \times 5$ Cartesian grid		$15 \times 44 \times 1$ Cartesian grid	
Non-uniform grid	$N_L = 6$ and $N_U = 20$		$N_L = 25$ and $N_U = 100$	
Tarbert formation				
Mean saturation error	NUC: 0.1796	UC: 0.1608	NUC: 0.2575	UC: 0.3367
Mean water-cut error	NUC: 0.0152	UC: 0.0123	NUC: 0.0393	UC: 0.1588
Upper Ness formation				
Mean saturation error	NUC: 0.1493	UC: 0.3397	NUC: 0.2162	UC: 0.5315
Mean water-cut error	NUC: 0.0213	UC: 0.0913	NUC: 0.0263	UC: 0.1899

With uniform coarsening we obtain accurate results only when using the $15 \times 44 \times 5$ Cartesian grid on stacks from the Tarbert formation. The reason why accurate results are obtained for these cases is that each grid block in the uniform Cartesian grid is only one layer thick so that we do not have sharp permeability contrasts within each grid block. Consequently we expect large saturation differences inside individual grid blocks only near the saturation front. For the stacks from the Upper Ness formation, on the other hand, one expects large saturation differences within individual grid blocks in large regions. When using the $15 \times 44 \times 1$ Cartesian grid one obtains poor results for all stacks. This is due to the fact that in the Tarbert formation there are large permeability differences between the respective layers, giving rise to segregated flow scenarios. By approximating the saturation in all five layers with a constant value, as is done in the uniform coarse grid simulations, one obtains large errors. In particular we see that this has great impact on the water-cut errors. Indeed, the errors obtained with the $15 \times 44 \times 1$ Cartesian grid are significantly larger than the errors obtained with the $15 \times 44 \times 5$ Cartesian grid.

5.3. Corner-point grid

The corner-point grid (or pillar grid) format [17] is a very flexible grid format that is used in many commercial geomodeling softwares. Essentially a corner-point grid consists of a set of hexahedral cells that are aligned in a logical Cartesian fashion where one horizontal layer in the logical grid is assigned to each sedimentary bed to be modeled. In its simplest form, a corner-point grid is specified in terms of a set of vertical or inclined pillars defined over an areal Cartesian 2D mesh in the lateral direction. Each cell in the volumetric corner-point grid is restricted by four pillars and is defined by specifying the eight corner points of the cell, two on each pillar.

Fig. 6 shows the corner-point grid that we employ in this section populated with two different permeability fields. This grid is given on a $30 \times 30 \times 30$ logical Cartesian format. However, several layers are partially eroded away so that many cells disappear and introduce connections between cells that are not neighbors in the logical grid. Whereas the logical grid has 27000 cells, the physical grid contains only 15206 cells.

We consider here a quarter-of-a-five-spot case, i.e., we inject at constant rate along a vertical column located at one of the corners and produce at constant rate along a vertical well at the opposite corner. The corner-point grid has been populated with 20 different permeability fields. Each field is a layered scalar permeability field where the permeability in each layer is spatially correlated and drawn from a log-normal distribution. For each realization the permeability distribution of layer z in the logical grid is generated using the following Matlab code:

```
k = exp(10 * rand(1) + smooth3(randn(30,30,
3),  'gaussian',[3,3,3],0.55));  K(:,:,z) =
k(:,:,2);
```

In the simulations we have taken 40 pressure steps and 10 saturation steps between each pressure step. The non-uniform coarse grids are generated using $N_{\mathrm{L}} = 6$ and $N_{\mathrm{U}} = 20$. These parameters resulted in 20 coarse grids with 647–704 cells. For the uniform coarsening approach we subdivide the $30 \times 30 \times 30$ logical Cartesian grid uniformly into a $10 \times 10 \times 10$ grid. However, because some cells disappear, the resulting coarse grid consists of a total of 838 cells. Hence, the non-uniformly coarsened grids contain 20–30% less cells than the uniformly coarsened grid that we use for comparison.

Fig. 7 shows saturation errors $e(S)$ and the water-cut errors $e(w)$ for all permeability fields and three viscosity ratios. Table 3 shows the mean error over the 20 realizations. The results clearly demonstrate that the solutions obtained with non-uniform coarse grids are, on average, significantly more accurate than the solutions obtained for the uniformly coarsened grid. For instance, the saturation errors obtained on the non-uniform coarse grids are in all cases, except for realization 19 with $\mu_{\mathrm{o}} = 10\,\mu_{\mathrm{w}}$, about 0.25, whereas the saturation errors obtained using the uniformly coarsened grid is often much larger.

The water-cut errors increase for both methods when μ_{o} decreases relative to μ_{w}. This is primarily due to the fact that high-viscosity ratios have a sharpening effect on the saturation front. In particular, for $\mu_{\mathrm{o}} = 0.2\,\mu_{\mathrm{w}}$ we obtain shock-type saturation fronts. Simply due to less spatial resolution, shock-type saturation fronts will be smoothed on coarsened grids. Hence, because the reference water-cut curve is a function of fine-grid saturations, and coarse grid water-cut curves are functions of coarse grid saturations, we obtain larger water-cut errors for high viscosity ratio flows than for low viscosity ratio flows. We notice also that for a couple of realizations for the case $\mu_{\mathrm{o}} = 0.2\,\mu_{\mathrm{w}}$ the water-cut error obtained with the non-uniform grid is larger than for the uniform grid. A reason for this may be that the algorithm sometimes groups many of the cells along the production well in one block, and therefore do not account for different breakthrough time in the different layers. This

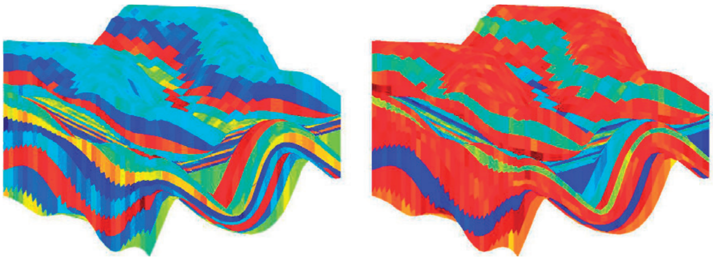


Fig. 6. Permeability field realizations 1 (left) and 2 (right). Realization 1 is particularly difficult to upscale because there are very few high-permeability layers. Consequently a lot of flow is forced into only a few of the layers. Realization 2 gives flow scenarios that are much easier to upscale. This model has thick high-permeable zones that transmit majority of the flow.

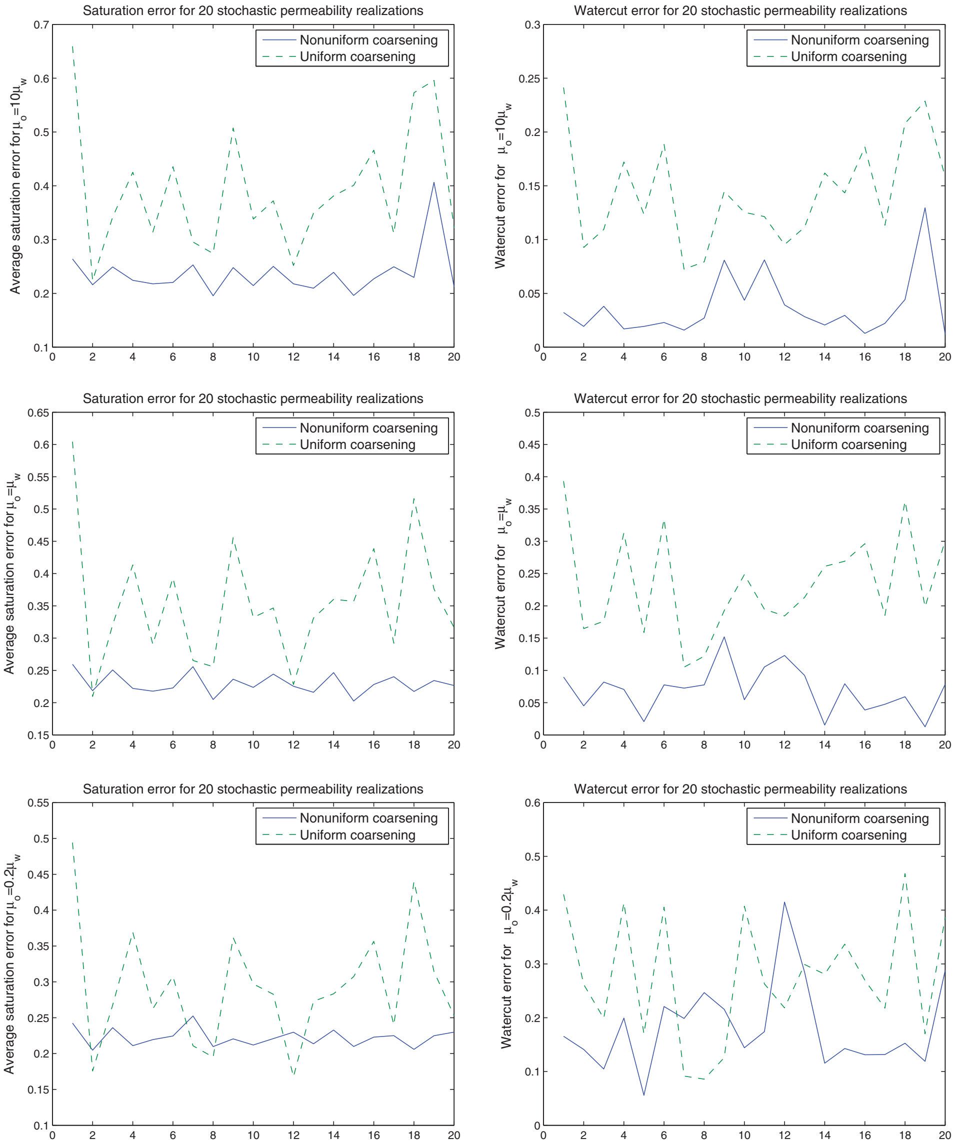


Fig. 7. Saturation and water-cut errors for 20 different permeability field realizations and viscosity ratio 0.1 (top), 1 (middle), and 5 (bottom).

Table 3
Mean of the errors plotted in Fig. 7

Viscosity ratio	$\mu_o = 10\mu_w$		$\mu_o = \mu_w$		$\mu_o = 0.2\mu_w$	
Coarsening strategy	NUC	UC	NUC	UC	NUC	UC
Mean saturation error	0.2370	0.3919	0.2296	0.3551	0.2224	0.2930
Mean water-cut error	0.0368	0.1439	0.0697	0.2337	0.1823	0.2750

could of coarse be avoided by adding an extra feature in the coarsening algorithm that ensures that blocks do not contain too many cells with a source.

5.4. Robustness

In the current section we seek to demonstrate the robustness of the proposed grid coarsening strategy. In particular we make an effort to show that the method provides accurate results for various degrees of coarsening and for various flow scenarios. To demonstrate the latter, we first consider flows imposed by different well-configurations. Next we pick one of the well-configurations and show that one obtains more accurate results with the non-uniform coarsening approach than with uniform coarsening, also when well-rates change during simulation. Finally we explore a flow scenario where the well-configuration changes rapidly during simulations. This case corresponds to a situation where new wells are drilled and old wells are shut down or reopened during simulation.

We would like to emphasize that we never regenerate the non-uniform coarse grid even though flow conditions change significantly so that the simulation velocity differs a lot from the velocity that is used to generate the coarse grid (cf. Algorithm 1).

In this section we revisit the SPE10 model. In particular, we consider the four bottom layers (from the fluvial Upper Ness formation) and use the same definition of the phase mobilities, i.e., λ_w and λ_o are defined by (16) with $\mu_o = 3$ cp and $\mu_w = 0.3$ cp. In each simulation we take 20 pressure steps and 200 saturation steps.

5.4.1. Robustness with respect to degree of coarsening

To assess robustness with respect to the degree of coarsening, we have selected five different Cartesian coarse grids, and chosen the parameters N_L and N_U correspondingly to create grids with comparable resolution. Table 4 shows the Cartesian grid dimensions along with the number of cells in the Cartesian grid and in the non-uniformly coarsened grid.

Table 4
Number of cells in each of the uniformly coarsened (Cartesian) and non-uniformly coarsened grids, respectively

Cartesian grid	$30 \times 110 \times 4$	$20 \times 55 \times 4$	$15 \times 44 \times 2$	$10 \times 22 \times 2$	$6 \times 22 \times 1$
Number of cells	13200	4400	1320	440	132
NUC: N_L/N_U	2/6	4/16	10/40	25/150	50/500
Number of cells	7516	3251	1333	419	150

Here we use the same well-configuration as in [9]; one injector in the middle and a producer at each corner (well-configuration A in Fig. 9). The corresponding saturation and water-cut errors are shown in Fig. 8. The results show that the non-uniform coarse grids consistently allow more accurate results than one obtains with the corresponding uniformly coarsened Cartesian grids. In particular it is worth noticing that for the four finest grids we obtain water-cut curves that nearly match perfectly the ref-

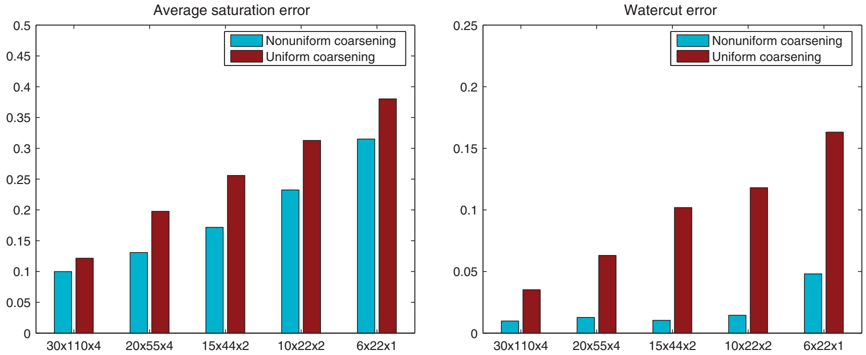


Fig. 8. Saturation and water-cut errors for five degrees of coarsening. For each case, the grid dimensions for the uniformly coarsened Cartesian grid is given.

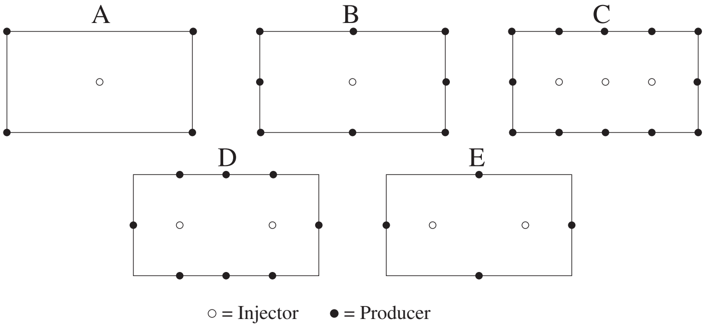


Fig. 9. Five selected well-configurations.

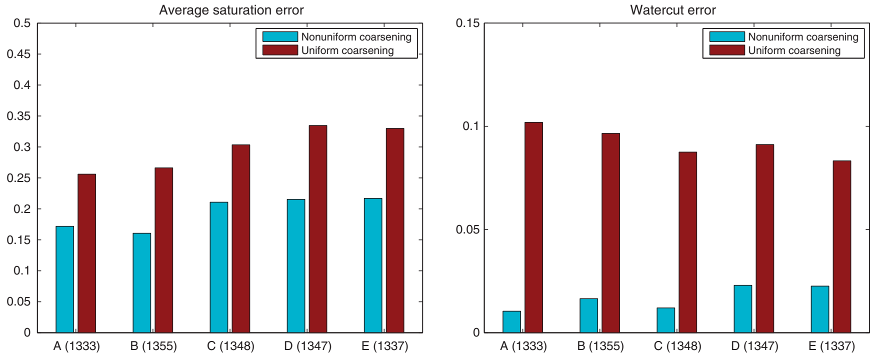


Fig. 10. Saturation and water-cut errors for each of the five well-configurations.

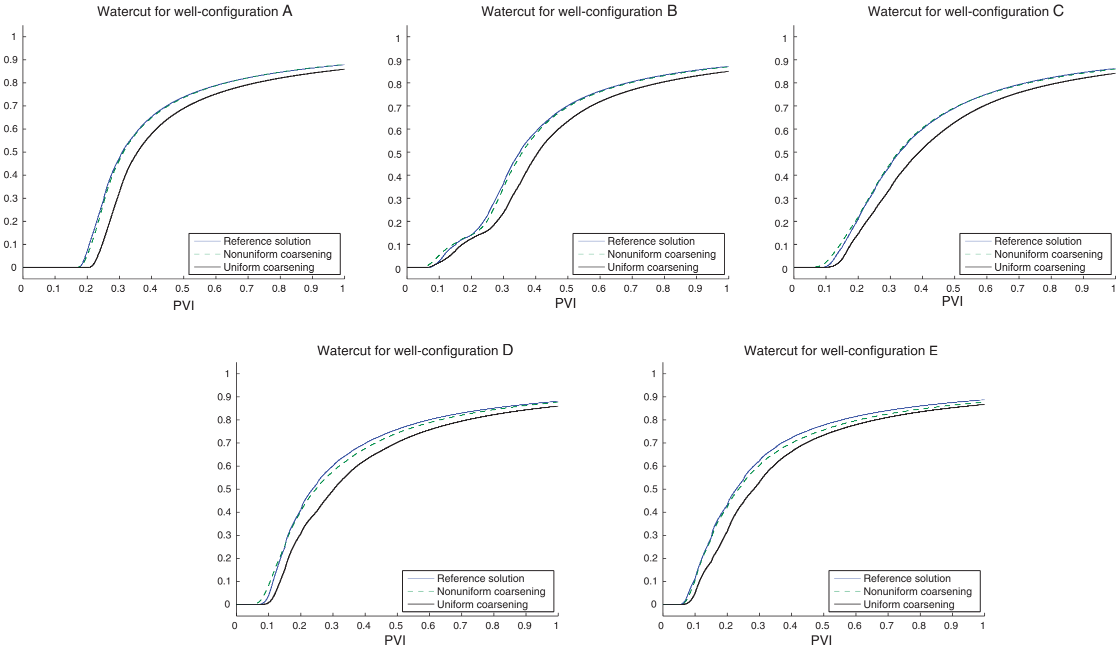


Fig. 11. Water-cut curves for each of the five well-configurations.

erence curve (see also the water-cut curves corresponding to well-configuration A in Fig. 11).

The saturation errors shown in Fig. 8 are computed on the fine grid. Hence since the coarse grid saturation solutions provide estimates for the average saturation in each block, we should indeed expect that the saturation error increases when the blocks become larger. However, the fact that we continue to produce accurate water-cut curves demonstrates that we are able to capture the qualitative behavior of the flow accurately, also on highly coarsened grids. The main reason for this is that the non-uniform grids resolve high flow channels quite well, also on very coarse grids, as is illustrated in Fig. 4.

5.4.2. Robustness with respect to well-placement

In this section we investigate the robustness with respect to well-locations. To this end we consider five test-cases with the five well-configurations shown in Fig. 9. To create the coarse grid we use $N_{\mathrm{L}} = 10$ and $N_{\mathrm{U}} = 40$, i.e., the same parameters that were used to generate the third non-uniform coarse grid above. Fig. 10 compares the saturation and water-cut errors obtained for each well-configuration with the corresponding errors obtained using a uniform $15 \times 44 \times 2$ Cartesian grid. The number of cells in the non-uniform coarse grids are shown in parenthesis along the lower edge of the plots. These plots show the same trend as Fig. 8; with the non-uniform coarsening approach we consistently get significantly less saturation and water-cut errors than we obtain using the corresponding Cartesian coarse grid. This is further illustrated in Fig. 11 which shows the respective water-cut curves for each well-configuration.

5.4.3. Robustness with respect to changing flow conditions

Finally we demonstrate that the proposed gridding strategy is applicable also when flow conditions change significantly, i.e., by generating the non-uniformly coarsened grids at initial time only. To this end we consider first a case with well-configuration A presented above where only the well rates change. For each pressure step we select the total well rates for each producer from a random distribution, ensuring only that they sum up to the injection rate for compatibility. We employ $N_{\mathrm{L}} = 10$, and $N_{\mathrm{U}} = 40$ and compute the initial velocity field for the case where all producers have the same total well rate. Fig. 12 shows how the saturation error $e(S,t)$ evolves along with the associated water-cuts. For both the non-uniform grid simulation and the uniform grid simulation we see that the error is about the same as for the case with fixed well-rates (Fig. 11: well-configuration A). In particular, the water-cut for the non-uniformly coarsened grid still matches very closely the water cut for the reference solution.

Next use the grid generated with well pattern C to run a simulation where we rotate among the well-configurations depicted in Fig. 9. That is, for each pressure step we choose a new well-configuration so that at the end of a simulation we have employed each well-configuration four times. Qualitatively the results depicted in Fig. 13 are very similar to the results depicted in Fig. 12, and thus also to the results for fixed well-rates and fixed well-configurations seen in Figs. 10 and 11.

These examples indicate that it is not necessary to regenerate the non-uniform coarse grid when flow conditions change, i.e., when flow patterns during simulation are dominated by heterogeneity, but possibly differ substantially from the pattern used to generate the grid. More detailed

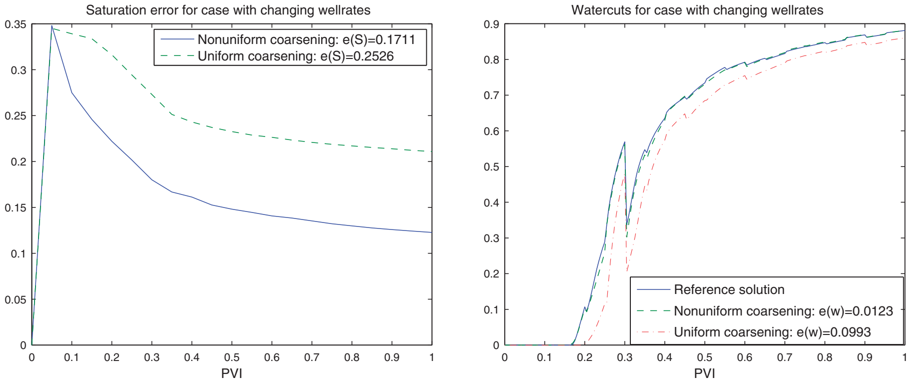


Fig. 12. Saturation and water-cut errors for a case where the well-rates change during simulation.

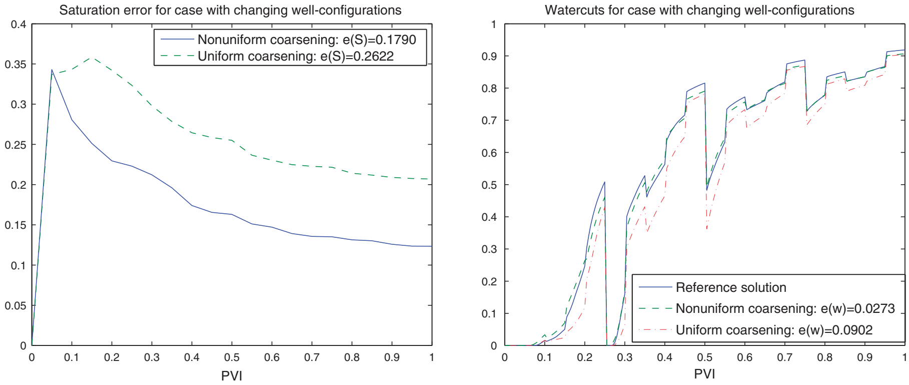


Fig. 13. Saturation and water-cut errors for a simulation where we rotate between the five well-configurations depicted in Fig. 9.

analysis of the effects of changing flow directions on the coarsening algorithm (e.g., for gravity dominated flows) is currently under investigation.

6. Summary and concluding remarks

In this paper we have presented a generic non-uniform coarsening strategy for modeling subsurface flow applications. The main objective has been to exploit information in high-resolution geological models in an optimal way. The proposed grid coarsening algorithm is applicable to both structured and unstructured grids and gives consistently more accurate results compared to results obtained with uniformly coarsened grids with roughly the same number of cells. The key to the enhanced accuracy is that the flow velocity is more accurately resolved on the non-uniform coarse grid.

The current work is motivated in particular by previous work of Durlofsky et al. [10] who proposed a strategy for generating coarse grids that are more finely gridded in high-flow regions than in low-flow regions. The main significance of our approach is that it is generic. Whereas the coarsening strategy in [10] applies primarily to Cartesian-like geomodels with high-flow channels that are aligned with the grid, our approach applies to arbitrary grids and arbitrary heterogeneous structures. The algorithm is also conceptually simple and easy to implement. We therefore believe that the proposed approach should be valuable in an industrial setting, e.g., by allowing users to specify grid-resolution at run-time to fit available computer resources and project requirements.

The ability to generate grids with user-defined resolution at run-time brings us closer to having an Earth Model shared between reservoir engineers and reservoir geologists [15]. This means that the geological model provided by geologists is used as input for the reservoir simulator. Apart from giving the reservoir engineer a better description of the subsurface, this will also offer the geoscientist the ability to validate the geomodel before it is sent to the reservoir engineer. This can simplify and accelerate flow simulation workflows considerably, and thereby allow e.g., oil-companies to save many man-hours.

Acknowledgements

We would like to thank Alf B. Rustad at STATOIL for providing the corner-point grid model used for the flow simulations in Section 5.3.

References

[1] Aarnes JE. 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. 2004;2(3):421–39.

[2] Aarnes JE, Kippe V, Lie K-A. Mixed multiscale finite elements and streamline methods for reservoir simulation of large geomodels. Adv Water Resour 2005;28(3):257–71.

[3] Aarnes JE, Krogstad S, Lie K-A. Multiscale mixed/mimetic methods on corner-point grids. Comput Geosci, in press.

[4] Aarnes JE, Krogstad S, Lie K-A, Laptev V. Multiscale mixed methods on corner-point grids: mimetic versus mixed subgrid solvers. Technical report, SINTEF, 2006.

[5] Arbogast T. Numerical subgrid upscaling of two-phase flow in porous media. In: Chen Z, Ewing R, Shi Z-C, editors. Lecture Notes in Physics, Berlin: Springer-Verlag; 2000. pp. 35–49.

[6] Arnold DN, Brezzi F. Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates. RAIRO Modél Math Anal Numér 1985; 19(1): 7–32. MR 813687 (87g:65126).

[7] Chen Z, Hou TY. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. Math Comp 2003;72: 541–76.

[8] Chen Z, Huan G, Li B. An improved impes method for two-phase flow in porous media. Trans Porous Media 2004;54:361–76.

[9] Christie MA, Blunt MJ. Tenth SPE comparative solution project: A comparison of upscaling techniques. SPE Reservoir Eval Eng. 2001;4:308–17.

[10] Durlofsky LJ, Jones RC, Milliken WJ. A nonuniform coarsening approach for the scale-up of displacement processes in heterogeneous porous media. Adv Water Resour 1997;20:335–47.

[11] Lipnikov K, Brezzi F, Shashkov M, Simoncini V. A new dicretization methodology for diffusion problems on generalized polyhedral meshes. LAUR-05-8717, Report, 2005.

[12] Lipnikov K, Brezzi F, Simoncini V. A family of mimetic finite difference methods on polygonal and polyhedral meshes. Math Models Methods Appl Sci 2005;15:1533–53.

[13] He C, Durlofksy LJ. Structured flow-based gridding and upscaling for modeling subsurface flow. Adv Water Resour 2006;29(12): 1876–92.

[14] Jenny P, Lee SH, Tchelepi HA. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. J Comput Phys 2003;187:47–67.

[15] King MJ, Burn KS, Wang P, Muralidharan V, Alverado F, Ma X, et al. Optimal coarsening of 3d reservoir models for flow simulation, SPE 95759 (2005).

[16] Kippe V, Aarnes JE, Lie K-A. A comparison of multiscale methods for elliptic problems in porous media flow. Comp Geosci, in press.

[17] Ponting DK. Corner point geometry in reservoir simulation. In: King PR, editor. Proceedings of the first European Conference on Mathematics of Oil Recovery, Cambridge, 1989 (Oxford) , Clarendon Press; July 25–27 1989. pp. 45–65.

[18] Raviart PA, Thomas JM. A mixed finite element method for second order elliptic equations. In: Galligani I, Magenes E, editors. Mathematical Aspects of Finite Element Methods. Berlin, Heidelberg, New York: Springer–Verlag; 1977. p. 292–315.

[19] Wen X-H, Gómez-Hernández JJ. Upscaling hydraulic conductivities in heterogeneous media: an overview. J Hydrol 1996(183):ix–xxxii.

Paper III

Flow-Based Coarsening for Multiscale Simulation of Transport in Porous Media

Vera Louise Hauge, Knut–Andreas Lie, and Jostein R. Natvig

Submitted
A short version is in the Proceedings of ECMOR XII – 12th European
Conference on the Mathematics of Oil Recovery, Oxford, UK,
6–9 September 2010

FLOW-BASED COARSENING FOR MULTISCALE SIMULATION OF TRANSPORT IN POROUS MEDIA

VERA LOUISE HAUGE, KNUT–ANDREAS LIE, AND JOSTEIN R. NATVIG

Abstract. Geological models are becoming increasingly large and detailed to account for heterogeneous structures on different spatial scales. To obtain simulation models that are computationally tractable, it is common to remove spatial detail from the geological description by upscaling. Pressure and transport equations are different in nature and generally require different strategies for optimal upgridding. To optimize the accuracy of a transport calculation, the coarsened grid should generally be constructed based on *a posteriori error* estimates and adapt to the flow patterns predicted by the pressure equation. However, sharp and rigorous estimates are generally hard to obtain, and herein we therefore consider various ad-hoc methods for generating flow-adapted grids. Common for all, is that they start by solving a single-phase flow problem once and then continue to form a coarsened grid by amalgamating cells from an underlying fine-scale grid. We present several variations of the original method. First, we discuss how to include *a priori* information in the coarsening process, e.g., to adapt to special geological features or to obtain less irregular grids in regions where flow-adaption is not crucial. Second, we discuss the use of bi-directional versus net fluxes over the coarse blocks, and show how the latter gives systems that better represent the causality in the flow equations, which can be exploited to develop very efficient nonlinear solvers. Finally, we demonstrate how to improve simulation accuracy by dynamically adding local resolution near strong saturation fronts.

1. Introduction

The main purpose of reservoir simulation is to provide predictions of the movement of hydrocarbon phases and water that will help oil and gas companies make better decisions on how to develop and produce their assets. The complexity of the workflows that lead to decisions is ever increasing, and advances in reservoir characterization, production optimization, and real-time reservoir management is leading to continued demand for faster and more advanced flow simulation tools. In particular, optimizing the recovery from mature and brown field assets will require multi-fidelity simulators that have a lot of flexibility and scalability to enable reservoir engineers to evaluate many (different) scenarios.

Coming up with a satisfactory solution is a challenging and daunting task and in this paper, we will only address a small part of the problem: development of multi-fidelity transport solvers to overcome the gap in resolution between geological and simulation models. Geo-cellular models resulting from structural and petrophysical modelling typically contain significantly more detail than what the reservoir engineer can afford if the simulation is to fit in memory and finish within a reasonable time frame of what he/she feels is necessary to capture the flow dynamics of a particular scenario with sufficient detail.

The traditional approach has been to use upgridding to create a new grid model with reduced spatial resolution and upscaling to bring petrophysical properties from the high-resolution geological description down to the new grid. A large number of different strategies have been developed to minimize the errors introduced in this model-reduction process, see e.g., [4, 6, 7, 10]. Upgridding and upscaling is generally a manual process and choosing the 'right' method and model resolution can be highly problem dependent and very time-consuming. The problem becomes more complicated as changes are introduced in the reservoir description to match observed (dynamical) data. Ideally, all changes should be made to the fine-scale geological model. However, because the turnaround time of traditional up- and downscaling processes is typically much larger than the man-hours allocated to the modelling project, one ends up with incompatible models at different spatial resolutions.

Herein, we will consider a multiscale approach to geological modelling. This approach differs from the traditional upgridding/upscaling approach in the sense that a fine-grid model is present at all times. Then it is up to the multiscale simulator to (automatically) coarsen the grid to reduce the number of degrees of freedom to a level that is sufficient to resolve flow physics and satisfy requirements on computational costs. The first component in such a simulator is a multiscale flow solver [9, 12, 13] that captures the fluid flow as a linear combination of a set of numerically computed basis functions. As such, the multiscale flow solver can be considered either as a robust upscaling method, or as a single-step upscaling-downscaling method that delivers approximate fine-scale fluxes. The basis functions are computed by solving localized flow problems, and the main distinction between different multiscale methods is how the local flow problems are constructed. The methods presented in the following are developed to accompany a particular method [3], but all ideas presented can readily be combined with any multiscale flow solver that produces *conservative* fluxes on coarse, fine, and intermediate grids. Previous research [3] has shown that our particular multiscale flow solver gives the best performance when the associated coarse grid follows geological structures. For corner-point grids, this is typically achieved through a regular partitioning in index space (using the underlying logical ijk numbering). Figure 1 gives a visual illustration of a multiscale flow solver.

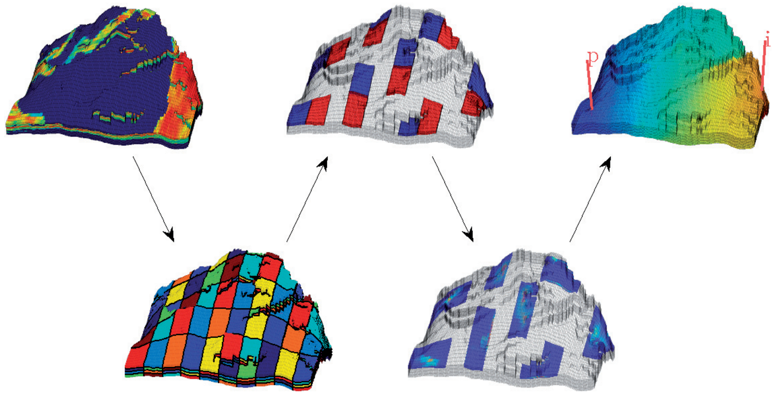


Figure 1. Illustration of a multiscale flow solver applied to a model from the SAIGUP study [17]. The coarse grid is generated as a regular partitioning of the underlying fine grid. Localized flow problems are set up (here, between each pair of blocks that share a common face) and then solved numerically. Finally, the basis functions are used to construct a global flow solution.

The second component of a multiscale simulator is an efficient transport solver that uses the flow field (pressure and total velocity) from the multiscale flow solver to evolve fluid saturations and compositions. To this end, there are several alternatives, depending upon which resolution one chooses to use. If the transport is to be computed on the same coarse grid as used in the multiscale flow solver, the best choice is to use a standard implicit finite-volume method with the coarse-scale fluxes computed by the multiscale solver. To solve the transport without upscaling—using the fine-scale, approximate fluxes—the best choice is probably a streamline method, e.g., as described in [1, 20], or one can use similar operator-splitting techniques to develop highly-efficient finite-volume solvers [18, 19] that use flow information to obtain an optimal ordering of the nonlinear discrete transport equations so that these can be solved in a cell-by-cell or block-by-block fashion. This gives local control over the (computationally expensive) nonlinear iterations and can significantly reduce the computational cost compared with standard (implicit) finite-volume methods.

In many cases solving saturation equations on the coarse scale may be too inaccurate and solving it on the fine grid may be too costly, and one would therefore look for a compromise between accuracy and computational speed. The adaptive multiscale finite-volume method of Lee et al. [16] and Zhou et al. [21] is one approach in this direction, in which three prolongation operators with different computational complexity were used to construct a multiscale transport solver. Alternatively, to optimize the accuracy of a transport calculation, the coarsened grid should generally adapt to the flow patterns predicted by the flow solver. Ideally, the flow-adapted grid should be constructed using a posteriori error estimates. Obtaining sharp and rigorous estimates are generally hard and good ad-hoc methods have been shown to capture flow and transport in highly heterogeneous reservoirs with good accuracy [2, 8]. In a flow-based method the grid is aligned to capture high-flow regions and (clearly) distinguish between regions of high and low flow. By capturing these important characteristics of the flow, one is able to generate coarse grids with a high upscaling factor that still deliver good accuracy.

In this paper, we will combine several of these ideas for speeding up the transport solves in a multiscale simulator setting. That is, we will make a few algorithmic improvements for the nonuniform coarsening method of Aarnes et al. [2] and borrow ideas from the reordering methods of Natvig and Lie [18] and the adaptive multiscale method of Lee et al. [16] and Zhou et al. [21].

2. Mathematical Model and Numerical Discretization

In general, multi-fidelity simulators need to cover a wide range of enhanced recovery processes including biological, chemical, electrical, gas-based, thermal, and water-based methods. However, in the following we will only consider the simplified setting of an incompressible, immiscible two-phase flow, described by a set of flow equations for the global pressure p and the total velocity $\vec{v}$

$$\nabla \cdot \vec{v} = h_p, \qquad \vec{v} = -\mathbf{K}\lambda(S)\nabla p, \tag{1}$$

and a transport equation for the saturation of one of the phases,

$$\phi \frac{\partial S}{\partial t} + \vec{v} \cdot \nabla f(S) = h_S. \tag{2}$$

These equations are defined over a singly-connected domain, represented by a grid that consists of a set of grid cells c_i, $i = 1, \ldots, n$. No further assumptions are made on the geometry and topology of the grid, apart from the requirement of an explicit mapping $\mathcal{N}(c)$ between cell c and its nearest neighbours. Most of the ideas we present in the following will therefore be applicable to any matching, unstructured, polyhedral grid. To keep the presentation as simple as possible, our examples will, with two exceptions, focus on 2D Cartesian grids, taken from individual layers of the widely used SPE10 benchmark [5]. Likewise, as our interest is primarily in the transport solver, we make no assumptions about the flow solver except that it produces mass-conservative fluxes on each cell. In a multiscale setting, our primary example of such a solver would be the multiscale mixed-finite method of Aarnes et al. [3]. In the following, v_{ij} will denote the flux over fine-cell interface γ_{ij} between cells c_i and c_j.

The remains of the paper will focus on transport solvers defined over a coarse grid that is constructed by grouping sets of cells into blocks B_ℓ, $\ell = 1, \ldots, N$. The simplest way of representing such a coarse grid is by a partition vector p with n elements, for which element p_i assumes the value ℓ if cell c_i is member of block B_ℓ. Representing the coarse grid by a partition vector gives us great flexibility in the shape of individual grid blocks and also opens up for a simple (interactive) manual editing, if deemed necessary. Herein, however, we will *not* use any manual editing to improve grid quality. More details of the grid generation will be given from the next section and onward.

Having created a coarse grid, the next step is to construct a coarse-grid transport solver. To this end, we assume that the saturations are constant over each grid block, i.e., $S_\ell = |B_\ell|^{-1} \int_{B_\ell} S(x)\,dx$. Then, a conservative coarse-grid discretization is obtained by summing a standard single-point upwind discretization for all cells in a block:

$$(3)\quad S_\ell^{n+1} = S_\ell^n + \frac{\Delta t}{\int_{B_\ell}\phi\,dx}\int_{B_\ell} h_S(S^{n+1})\,dx \\ - \frac{\Delta t}{\int_{B_\ell}\phi\,dx}\Big[f(S_\ell^{n+1})\sum_{\gamma_{ij}\subset\partial B_\ell}\max(v_{ij},0) - \sum_{k\neq\ell}\Big(f(S_k^{n+1})\sum_{\gamma_{ij}\subset\Gamma_{k\ell}}\min(v_{ij},0)\Big)\Big].$$

Notice that if there is bi-directional flow across $\Gamma_{k\ell} = \partial B_k \cap \partial B_\ell$, then the phase-flux across $\Gamma_{k\ell}$ is approximated using both S_ℓ and S_k. Thus, although (3) stems from a single-point upwind scheme on the fine grid, we may obtain a two-sided upwind scheme on the coarse grid. Because the method uses fluxes computed on a finer scale to correctly propagate information over the faces of the coarse grid, it can be viewed as multiscale solver.

The observant reader will notice that the evaluation of the fractional flow function has been moved outside the sum over fine-grid faces in (3). Ideally, the saturation should have been reconstructed in the cells along the block faces $\Gamma_{k\ell}$ to account for subscale variations in the fractional flow. Whereas this is straightforward for an explicit scheme on rectangular blocks, we are not aware of any good method to do so for implicit schemes on arbitrarily shaped blocks. In the following, we will therefore only compute block-averaged saturations.

The multiscale transport solver in (3) has an upscaling counterpart, which only uses one-way fluxes over each coarse interface. These net coarse-scale fluxes are derived by integrating the fine-scale fluxes, giving the following scheme

$$(4)\quad S_\ell^{n+1} = S_\ell^n + \frac{\Delta t}{\int_{B_\ell}\phi\,dx}\Big[\int_{B_\ell} h_S(S^{n+1})\,dx - \sum_{k\neq\ell}\max\Big(f(S_\ell^{n+1})\sum_{\gamma_{ij}\subset\Gamma_{k\ell}} v_{ij}, -f(S_k^{n+1})\sum_{\gamma_{ij}\subset\Gamma_{k\ell}} v_{ij}\Big)\Big].$$

Like the fine-scale discretization, this is a single-point upwind scheme, but now on the coarse scale. Using net fluxes will simplify the coupling in the resulting nonlinear discrete system. Notice also that the net fluxes used in (4) only coincide with the coarse-grid fluxes computed by the multiscale flow solver when the two solvers are defined over the same coarse grid.

Based on (3) and (4), one can easily develop an adaptive scheme that uses net fluxes across all grid faces where the flux is predominantly unidirectional and fine-scale fluxes across the other faces.

3. Flow-Based Nonuniform Coarsening

We will use the nonuniform coarsening method by Aarnes et al. [2] as our starting point for developing efficient transport solvers on flow-adapted coarse grids. This method partitions the fine grid into coarse blocks according to flow magnitude by separating regions of high and low flow.

The algorithm follows four steps:

(1) *Compute an initial partitioning.* To this end, we will use the logarithm of the flow as our indicator function, $g(c_i) \propto \log|\vec{v}(c_i)|$, which is segmented into ten uniform bins: $c_i \subset \tilde{B}_\ell$ if $g(c_i) \in [g_\ell, g_{\ell+1})$. Each bin $\tilde{B}_\ell$ may consist of a multiply connected set of cells and must be postprocessed and split into singly-connected blocks.

(2) *Merge small blocks.* If a block B' has too small volume, $|B'| < \frac{N_L}{n}|\Omega|$, it is merged with the neighbouring block B that has the closest g-value defined as $g(B)|B| = \int_B g(c)\,dx$.

(3) *Refine blocks with too much flow.* If $\int_B g(c)\,dx > \frac{N_U}{n}\int_\Omega g(c)\,dx$, then

   (a) Pick an arbitrary cell c_0 belonging to ∂B.

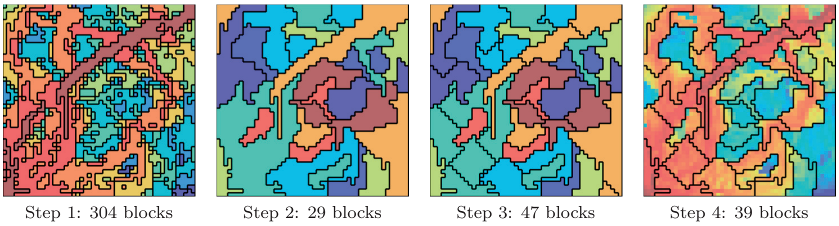


FIGURE 2. Illustration of Steps 1 to 4 for a quarter five-spot simulation of a 50×50 excerpt from Layer 68 from Model 2 of the SPE10 benchmark [5]. The coarsening parameters are $N_L = 20$ and $N_U = 100$. In the rightmost plot, the colours represent logarithm of the underlying fine-scale velocity field.

(b) Find the cell $c_1 \subset B$ that is furthest away from c_0, using, e.g., the distance between the cell centroids as a metric, and define $B' = c_1$.
(c) Progressively enlarge B' by adding cells surrounding B'; that is, add $c \subset \mathcal{N}(B')$ if $c \not\subset B'$, as long as $\int_{B'} g(c)\,dx \leq \frac{N_U}{n} \int_\Omega g(c)\,dx$.
(d) Define $B = B \setminus B'$ and continue to refine B if the upper bounds are still violated.

(4) *Repeat Step 2.*

The four steps are illustrated in Figure 2. Two coarsening parameters determine the coarsening level: N_L gives a lower bound on the volume of blocks and prevents the algorithm from generating too small blocks, and N_U gives an upper bound on total amount of flow through each grid block and prevents the algorithm from generating too large blocks. In the original algorithm, $\mathcal{N}(c)$ is defined as the face neighbours: that is, cells c and $\tilde{c}$ are neighbours if they share a common face. For a Cartesian grid, this corresponds to the usual 5-point neighbour relation, and gives rise to the characteristic diamond-shaped cells seen in the two rightmost plots. To get blocks with a more regular shape, we will in the following use a 9-point neighbour relation. We refer the reader to [11] for a more thorough discussion of neighbour relations in coarsening algorithms.

4. Imposing *A Priori* Shapes

The algorithm presented in the previous section is fully automated in the sense that the degree of coarsening is determined by two user-supplied parameters (N_L and N_U) and the shape of the blocks is determined by the neighbourhood definition, so that the algorithm creates coarse grids with block boundaries that are aligned with distinct features in the permeability field as reflected in the computed velocity field. For highly heterogeneous reservoirs, and in particular for strongly channelized reservoirs, this results in grids that capture the dominating flow patterns very accurately, even for high upscaling factors. This was demonstrated in [2], for both structured and unstructured grids, and in particular for Model 2 from the 10th SPE Comparative Solution Project [5], for which flow-based coarsening significantly reduces saturation errors and errors in water cuts compared with a regular Cartesian coarsening for the fluvial part of the model (the 50 layers of the Upper Ness formation). Moreover, Aarnes et al. [2] demonstrated that the method is robust with respect to coarsening degree and varying flow patterns, as long as the flow field is dominated by the underlying heterogeneity.

Generally, the nonuniform coarsening algorithm is not as robust and efficient as it was somewhat optimistically reported in [2]. On the upper layers of the SPE10 model (the somewhat smoother Tarbert formation), the flow-based grids do not give better accuracy than a straightforward Cartesian coarsening. In fact, we have run numerous experiments with varying coarsening factors for the SPE10 and other models that all indicate that Cartesian coarsening predominantly gives slightly better accuracy in saturation fields and water cuts than the flow-based grid for cases with small or moderate heterogeneity. Altogether, our experiments suggest that the original algorithm, as presented in the previous section, has a tendency of exaggerating the effect of the underlying velocity pattern and thus creates grids that are more irregular than what is needed.

The flow-based grids typically have more irregular blocks with more neighbours and coarse-block interfaces and this tends to increase the coupling in the discrete nonlinear system. The number of couplings in the nonlinear system will typically affect how costly it is to solve, and hence it is desirable to increase the regularity of the blocks if it does not significantly affect their ability to resolve flow patterns; we will come back to this discussion in the next section. In many cases, the user will have expert knowledge of what has the most influence on accuracy and may want, e.g., to impose *a priori* information on the (local) shape of the coarse blocks. Likewise, there can be other geological features that the user may want to use to create grids that better adapt to the underlying geology. In the rest of this section, we will demonstrate how to use such *a priori* information to create grids that give improved accuracy.

First of all, we propose an additional step in the original algorithm that consists of intersecting the initial flow-based colouring of cells in the first step with an *a priori* partitioning. This intersection will then be the basis for the rest of the steps, which remain unchanged from the original algorithm. We point out that this additional step is applicable to *any* grid, for which the user is able to specify an *a priori* partition vector p_a. Second, to reduce the influence of the underlying heterogeneity on the coarsening, we look at the initial colouring of cells. Reducing the number of bins means that we increase the size of the blocks resulting from the flow-based colouring, and hence to a large extent preserve the *a priori* partition. If the sizes of the *a priori* blocks are within the bounds specified by the N_L and N_U parameters, a large number of these blocks will be left intact by Steps 2–4.

To illustrate the effect of the extra step and the adaptive number of initial bins, we will consider an example in which we seek to impose a regular Cartesian partitioning on the flow-based gridding process.

Example 1 (Layer 1 of SPE10)**.** *The leftmost plot in Figure 3 shows the intersection of a regular* 6×22 *Cartesian coarsening with the initial flow-based colouring. Altogether, this generates a finer partitioning as the starting point of the merging and refinement steps. The added interfaces are straight lines that will typically be preserved in low-flow regions, as seen by comparing Grid 1 with the grid generated by the original algorithm. In the original algorithm and for Grid 1, we have used 10 bins in the initial colouring. This amounts to approximately one bin per order of magnitude in the underlying velocity field for the fluvial parts of the SPE10 model. For the layers in the Tarbert formation, as considered here, the logarithmic span in the velocities is significantly smaller. For Grid 2, we therefore have chosen the number of initial bins equal to the logarithmic span in the underlying fine-scale velocity field. As a result, more blocks from the* a priori *partitioning remain unchanged throughout Steps 2–4, and the resulting grid has a much more regular structure than Grid 1. On the other hand,*

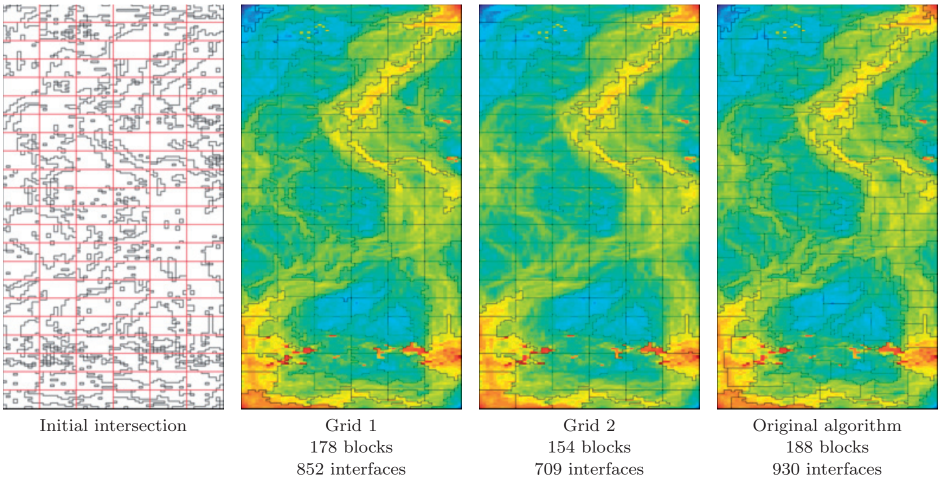
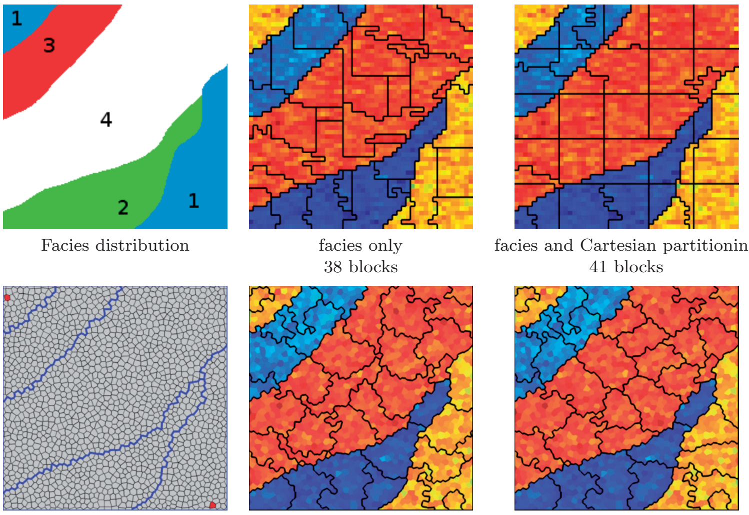


FIGURE 3. Nonuniformly coarsened grids for Layer 1 of Model 2 from SPE10. The leftmost plot outlines the intersection of the initial partitioning, where red lines represent the *a priori* partitioning and black lines the partitioning into bins. Grid 1 is generated with 10 initial bins and Grid 2 with the number of bins adapted to the flow, and for both we used coarsening parameters N_L=25, N_U=125 and a uniform 6×22 *a priori* partitioning. The rightmost plot shows the grid generated by the original algorithm with parameters $N_L = 15$, $N_U = 75$ and a 9-point neighbourhood relation. All three grids are outlined on top of the logarithm of the fine-scale velocity field, computed by solving a quarter five-spot problem.

for both Grid 1 and Grid 2, we observe that the high-flow channels are distinctly outlined and corresponds to the ones detected by the original algorithm. In regions of low flow, we also observe that the original grid has a much more complex grid structure, with a larger number of neighbouring connections.

We have run a large number of different studies using the original and our improved algorithm (a few quantitative results will be reported in the next section). Choosing the number of bins according to the logarithm span in velocity (or possibly permeability) seems to be a good choice. The other parameters, however, must be chosen with some care and possibly be fine-tuned to give optimal results. For small and moderate heterogeneities, it seems particularly important to balance the choice of the two partitioning mechanisms. Choosing a coarse *a priori* partitioning and small partitioning parameters (N_L, N_U), means that the flow-based partitioning will dominate and any advantages from the *a priori* partitioning disappears. Likewise, choosing a fine *a priori* partitioning and large values for N_L and N_U, implies that most cells are merged into large blocks in Step 2 and then refined in Step 3. The resulting grid will hence have the characteristics of the original algorithm, and the effects of the *a priori* partitioning disappears.

A priori information can also be used to distinguish different geological features that need to be taken into account and/or preserved during the coarsening. In the next example, we demonstrate how one can use facies numbers as the initial partitioning to ensure that the



Facies distribution

facies only
38 blocks

facies and Cartesian partitioning
41 blocks

PEBI grid and wells

facies only
39 blocks

facies and METIS partitioning
40 blocks

FIGURE 4. Coarsening using *a priori* information of facies distribution. In the upper row, the underling fine grid is a regular 50 × 50 Cartesian grid. In the lower row, the fine grid is a fully unstructured PEBI grid with 1697 cells.

coarse blocks do not cross facies boundaries. This can be very useful if the facies have different relative permeabilities or capillary pressure curves. By making sure that each coarse block consists of only one facies, one can avoid cumbersome upscaling of relative permeabilities and capillary pressure functions to the coarse grid.

Example 2 (Facies model)**.** *We consider a rectangular domain with a facies distribution as shown in the upper-left plot of Figure 4. The permeability distribution follows a lognormal distribution inside each facies with mean values of* 400*,* 20*,* 35*, and* 800 *mD, respectively. The permeability distribution is sampled on a regular* 50 × 50 *Cartesian grid. The reservoir is produced by an injector-producer pair located near the upper-left and lower-right corners of the domain, respectively; the wells are shown as red dots in the lower-left plot in Figure 4. For illustration purposes, we generate two coarse grids: for the first one, we only use the facies distribution as our* a priori *partition vector, and for the second one, we impose a regular Cartesian partitioning, in addition, as discussed for Grid 2 in Example 1. The two grids are shown in the middle and left plot in the upper row of Figure 4. From the plots, we clearly see that the coarse blocks are confined inside a single facies. By additionally imposing an* a priori *Cartesian partitioning, we increase the number of blocks, but also get more regularity in our coarsening.*

To demonstrate that our method is not restricted to Cartesian grids, we consider the same facies and permeability distribution (re)sampled on the PEBI grid (unstructured Voronoi grid) with 1697 cells shown in the lower-left plot of Figure 4. The middle and left plots in the lower row of Figure 4 show two coarse grids by our algorithm. The difference in the two grids lie in how we perform the refinement in Step 3 of the algorithm. In the middle plot, we have used *the refinement method outlined above with the unstructured equivalent of the 9-point neighbour definition. Also in this case, the blocks are clearly confined to a single facies, as desired. However, because of the unstructured connections in the fine-scale grid, the coarse blocks are significantly more irregular than in the Cartesian case. In the left plot, we have instead used METIS to perform the refinement step, this in an attempt to improve the regularity of the blocks.*

Although the results presented in the previous example are encouraging for PEBI grids, we believe that more research is needed to come up with a better algorithm for refining large blocks in Step 3, particularly in 3D, in a way that imposes more regularity on the resulting blocks. The same problem arises when generating coarse grids to be used in multiscale flow solvers. Because of the complex connection pattern of the underlying unstructured grid, it is generally difficult to come up with coarse blocks that do not have irregular boundaries. One possibility would be to include a smoothing step, but this would need to be carefully designed to distinguish between irregular faces caused by flow adaption and irregular faces induced by the unstructured connections.

In the next example, we will study a realistic 3D model represented in the industry-standard corner-point format, i.e., as a grid that consists of a set of hexahedral cells that are topologically aligned in a Cartesian fashion so that the cells can be numbered using a logical ijk index. From a coarsening perspective, the underlying ijk index is very useful and can e.g., be utilized to impose a regular *a priori* partitioning as in Example 1. Here, however, we will use saturation regions as our *a priori* partitioning.

Example 3 (SAIGUP)**.** *We consider one of the faulted models from the SAIGUP study [17], which we have already used in Figure 1 to illustrate the key steps of a multiscale flow solver. The petrophysical parameters for the model were generated on a regular* $40\times120\times20$ *Cartesian grid and then mapped onto a structural model described using the corner-point grid formate. The left plot in Figure 5 shows the structural model. The colors represent the six different saturation regions (Eclipse keyword SATNUM), which may or may not correspond to different facies or rock types. Because the main purpose of the example is to illustrate the gridding capabilities on a model with realistic geometry and petrophysical properties, we use a simple injector–producer pair (see Figure 1) and create a relatively coarse flow-based grid.*

The coarse grid was created by imposing the six saturation regions as an a priori *partitioning. Moreover, in Steps 2 and 4, we restricted the neighbourhood definition to only include cells that were part of the same saturation region. As we see from Figure 5, the coarse blocks have complicated shapes but seem to follow the saturation regions; this is particularly evident in region six. The plot may be slightly deceiving with respect to the connection between blocks: blocks that appear to be multiply connected are, in fact, singly connected through cells in deeper layers that are not visible in the plot.*

Another example of flow-based gridding on corner-point grids was presented by Krogstad et al. [15], who used such grids to accelerate forward simulations in a production optimization workflow. In the next section, we will give a more quantitative study of the gridding methods introduced in the previous section when applied together with the multiscale transport solver (3) and the coarse-scale solver (4).

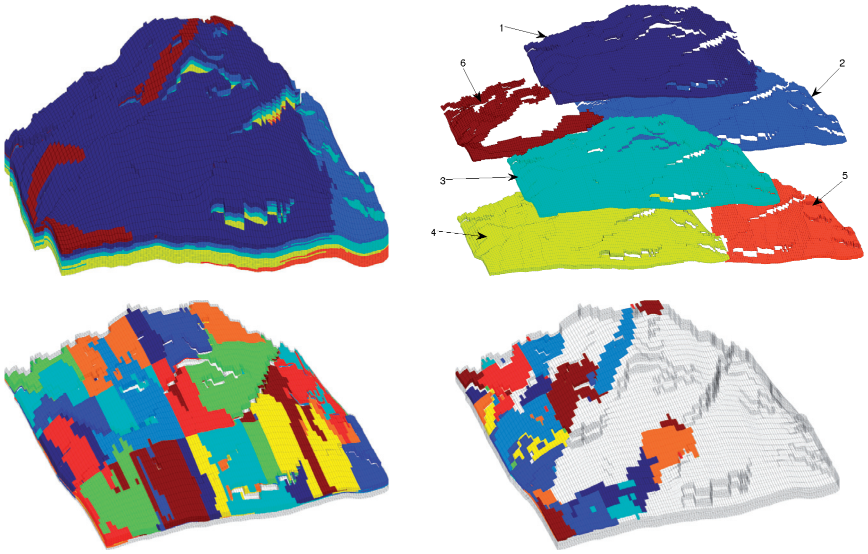


Figure 5. Flow-based coarsening of a faulted SAIGUP model. The upper-left plot shows the structural model. The upper-right plot shows the six saturation regions. The lower plots show the coarse-grid blocks in saturation regions three and six, respectively.

5. Multiscale versus Coarse-Scale Solver

First, we start by discussing the computational efficiency. Both discretizations, (3) and (4), lead to a system of nonlinear equations that will typically be solved by a Newton–Raphson type method. The computational efficiency of the nonlinear solver will to a large extent depend on the structure and the condition number of the system. The best we can hope for, is that the discrete system has an upper (or lower) triangular form, because then we can use a nonlinear Gauss–Seidel solution procedure and compute the unknown block saturations by backward (or forward) back-substitution. This is clearly possible for a 1D problem. Likewise, in the absence of gravity and capillary forces, (2) has an inherent causality principle that is utilized in streamline methods to transform the multi-dimensional transport equation into a family of 1D problems along streamlines.

Natvig and Lie [18] recently demonstrated how this causality principle can be used to compute an optimal flow-based ordering that renders the system in a block-triangular form. If the flow solver is monotone, each diagonal matrix block will correspond to only one grid block and the solution can be computed block-by-block, moving gradually downstream from wells or other fluid sources. For non-monotone flow solvers, there will be some circulation in the discrete fluxes, which will lead to larger matrix blocks that contain grid blocks that are circularly dependent. Still, the system can be solved by an efficient block-wise back-substitution procedure, in which the circularly dependent grid blocks (henceforth called connected components) are solved for simultaneously. Furthermore, if the linearization is performed *locally* on each matrix block, we gain local control over the nonlinear (Newton–Raphson) iterations

Table 1. Comparison of the matrix structure on two layers of SPE10. For each column, the first number refers to the multiscale solver (3) and the second number refers to the coarse-scale solver (4).

	Layer 1						Layer 37					
	Grid 2		Original		Cartesian		Grid 2		Original		Cartesian	
# grid blocks	154	154	188	188	220	220	169	169	205	205	220	220
# scalar components	1	127	0	138	75	220	1	65	0	100	16	208
# connected components	4	7	1	7	50	0	1	13	2	9	24	3
largest component	89	6	188	26	10	–	168	33	203	26	48	4
# off-diagonal elements	507	342	798	495	503	408	756	439	911	557	574	408

and thereby obtain highly efficient and (near) optimal nonlinear solvers [18, 19]. Finding the connected components and the optimal ordering are standard and efficient $\mathcal{O}(N)$ operations from graph theory that are easy to implement.

Rather than considering a specific nonlinear solver, we will in the following use the degree to which the discrete system can be reordered into a triangular form as a measure of the efficiency of the grid. Our idea is that if such a structure exists, any efficient nonlinear solver should ideally be able to exploit it.

Example 4 (Layers 1 and 37 of SPE10)**.** *Continuing from Example 1, we consider a quarter five-spot problem on two layers from the SPE10 model: Layer 1 from the Tarbert formation and Layer 37 from the fluvial Upper Ness formation. For each layer, we will use three different grids: a 10×22 Cartesian grid, a grid generated by the original algorithm from [2], and Grid 2 from Example 1 and its equivalent on Layer 37. Table 1 reports the corresponding number of scalar components, number of connected components, number of blocks in the largest connected component, and number of off-diagonal entries.*

Let us first look in detail on a few of the grids. We start by the Cartesian grid for Layer 1. Using bi-directional fluxes, there are 50 connected components that contain at most ten grid blocks and 75 scalar components. If we instead use net fluxes, there are only scalar components in the system, which can therefore be solved one grid block at the time. Next we consider Grid 2 on Layer 37. Figure 6 shows the concept of the matrix reordering for the case with net fluxes: Out of the 169 blocks in the grid, 104 blocks have some circular dependence and are part of thirteen connected components: two large, one intermediate, and ten small. The largest component contains 33 blocks, in which the saturations must be computed simultaneously by solving a 33×33 nonlinear system. Similar block systems must be solved for the other twelve connected components. The remaining 65 scalar components are only connected to their upwind neighbours and here the saturation can be computed by solving a scalar nonlinear problem once the upwind neighours have been computed.

Overall, we see that the use of net fluxes, as in (4)*, rather than bi-directional fluxes across the block interfaces, as in* (3)*, reduces the number of off-diagonal elements, the number of connected components, and the size of these components on all six grids and hence leads to a nonlinear system that generally will be less expensive to solve. This conclusion should also be true in the general case: replacing bi-directional fluxes with net fluxes will decrease the number of couplings in the nonlinear discrete system and hence decrease the computational cost.*

Let us now briefly look at the accuracy of the two schemes, (3) *and* (4)*. In Figure 7, we compare the corresponding coarse-scale saturations with the fine-scale reference saturation for*

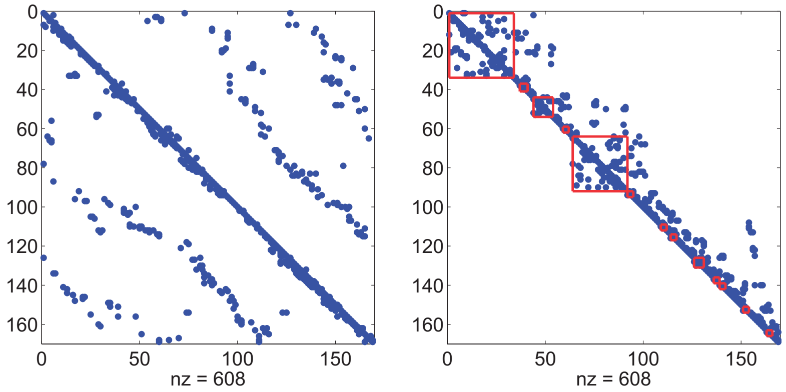


Figure 6. Matrix structure of the flux vector for Grid 2 on Layer 37 with the coarse-grid solver (4). The left plot shows the matrix with the ordering coming from the coarsening algorithm, whereas the right plot shows the matrix structure after we have performed a flow-based reordering. Altogether, 65 of the saturation values depend only on their upwind neighbours, whereas the remaining 104 have some circular dependence and are part of one of the thirteen connected components. The corresponding matrix blocks are marked in red: two large, one intermediate, and ten small.

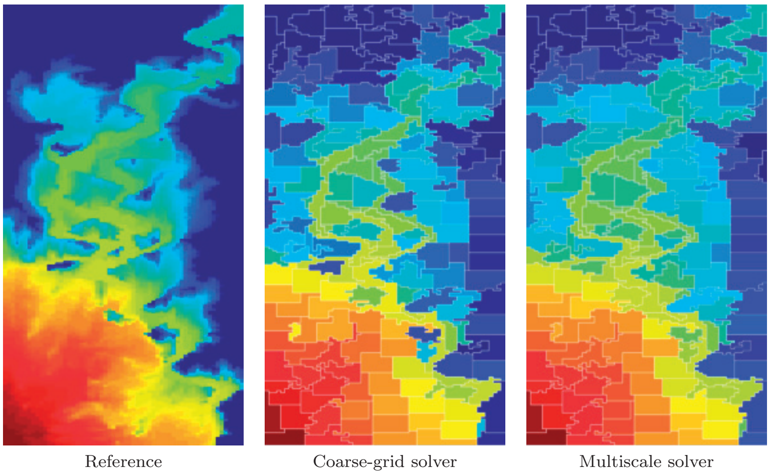


Figure 7. Comparison of saturations at dimensionless time 0.8 PVI for the coarse grid generated by the original algorithm on Layer 37.

the original flow-based grid on Layer 37. In the visual norm, the coarse-grid solver appears to resolve the saturation field more accurately than the multiscale solver: the solver exhibits less numerical diffusion in low-flow regions and has been able to capture pockets of bypassed oil. We believe that the difference in the two solvers can be explained as follows: the multiscale solver effectively uses central differences and hence spreads small saturations induced by numerical diffusion into large(er) areas. The coarse-scale solver is effectively an upwind solver and hence has less coarse-scale numerical diffusion.

In a multiscale setting, the results above are interesting for the following reason. Krogstad et al. [15] have previously demonstrated that the combination of a multiscale flow solver and a flow-adapted grid can be very efficient if one can avoid communication through the underlying fine grid and only store fine-scale fluxes at the coarse interfaces and use precomputed mappings to move saturations from the transport grid to the multiscale flow solver. Replacing fine-scale fluxes on each coarse-grid interface by net fluxes will decrease the communication need even further and hence increase the efficiency of the overall solver.

In the next example, we will study the accuracy of the multiscale and the coarse-scale solvers on the SPE10 model. To this end, we will need some notation. Let S_f and S_c denote the saturation field computed on the fine and coarse grids, respectively, and let w_f and w_c denote the respective water cuts. Moreover, we define $\mathcal{R}$ to be the restriction from the fine to the coarse grid and $\mathcal{P}$ to be the prolongation from the coarse to the fine grid. Finally, we define two different error norms

$$E_s(q,p) = \frac{1}{T}\int_0^T \frac{\|[q(\cdot,t)-p(\cdot,t)]\phi(\cdot)\|_1}{\|p(\cdot,t)\phi(\cdot)\|_1}\,dt, \qquad E_w(w_1,w_2) = \frac{\|w_1(\cdot)-w_2(\cdot)\|_2}{\|w_2(\cdot)\|_2}.$$

Example 5 (All layers of SPE10)**.** *For each layer of the SPE10 model, we conduct a quarter five-spot simulation using a uniform* 12×22 *Cartesian coarsening as well as the three coarsening choices discussed in Example 1. The Cartesian grid corresponds to an upscaling factor of 50 and the parameters in the flow-based algorithm were hence chosen to produce a similar (or slightly larger) upscaling factor. Table 2 shows a comparison of water-cut and saturation errors, where the saturation errors have been split in three parts: error measured on the fine grid, error measured on the coarse grid, and projection error. All simulations used 20 equally-spaced pressure steps with 15 equally-spaced substeps in the transport solvers. To minimize the errors introduced, the pressure updates were performed on the underlying fine-grid.*

We start by considering the original multiscale transport solver (3) *from [2]. Here, the results in the upper part of the table clearly show that the water-cut and projection errors are significantly* reduced *on the fluvial layers by using the flow-based coarsening methods, whereas there is no significant change in the water-cut error and slightly larger projection errors on the smooth Tarbert layers. On the other hand, comparing with the Cartesian grid, we see that the coarse-scale evolution error is significantly* increased *on the Tarbert layers, and slightly reduced on the fluvial layers. Finally, we notice that both Grid 1 and Grid 2 consistently produce lower errors and fewer blocks and faces than the original coarsening algorithm.*

Looking at the coarse-scale scheme (4)*, we see that this solver consistently gives lower errors than the multiscale solver for all four grids, in particular for the water-cut error. As discussed in Example 4, we think this can be attributed to a significantly lower coarse-scale diffusion. Moreover, on the Tarbert layers, the errors for Grid 1 and Grid 2 are lower than for the regular Cartesian grid.*

Table 2. Comparison of saturation and water-cut errors up to dimensionless time $T = 1.0$ PVI for quarter five-spot simulations on the individual layers of the SPE10 model. The upper part of the table shows errors for the multiscale solver (3), the middle part shows errors for the coarse-scale solver (4), and the lower part gives statistics for the grids.

	Tarbert formation				Upper Ness formation			
	Grid 1	Grid 2	Original	Cartesian	Grid 1	Grid 2	Original	Cartesian
$E_s(\mathcal{PR}S_f, S_f)$	0.0920	0.0941	0.1042	0.0911	0.1394	0.1371	0.1355	0.1772
$E_s(\mathcal{P}S_c, S_f)$	0.2071	0.1910	0.2426	0.1687	0.2180	0.2124	0.2243	0.2305
$E_s(S_c, \mathcal{R}S_f)$	0.1784	0.1599	0.2100	0.1381	0.1572	0.1522	0.1683	0.1604
$E_w(w_c, w_f)$	0.0649	0.0695	0.0773	0.0701	0.0613	0.0609	0.0668	0.0982
$E_s(\mathcal{P}S_c, S_f)$	0.1591	0.1607	0.1875	0.1619	0.1827	0.1795	0.1862	0.2191
$E_s(S_c, \mathcal{R}S_f)$	0.1220	0.1237	0.1459	0.1302	0.1155	0.1135	0.1225	0.1486
$E_w(w_c, w_f)$	0.0349	0.0473	0.0444	0.0647	0.0232	0.0237	0.0325	0.0844
# blocks: span	232–268	217–261	233–312	264	202–234	205–241	220–303	264
# blocks: mean	249	236	275	264	216	222	264	264
# faces: mean	1175	1069	1363	1090	1049	1070	1309	1090

Having presented the results in the previous example, we must concede that the results are slightly volatile. Based on a large number of experiments using the multiscale transport solver, we see that using a different upscaling factor can in many cases produce more favourable results for the flow-based coarsening methods (on the Tarbert layers), but may also in certain cases produce slightly worse results. The coarse-scale solver is more recent, and we have not yet conducted an equally extensive study. Still, we believe that this solver will prove to be more accurate because the single-point flux approximation generally has less coarse-scale diffusion than the bi-directional flux.

6. Dynamically Adaptive Grid

For displacements with strong displacement fronts, the majority of the projection error, which was briefly discussed in Example 5, is associated with inaccurate representation of the fluid front. Think of a typical Buckley–Leverett profile: in the unswept area ahead of the displacement front, the solution is constant and can be accurately represented on a relatively coarse grid. Likewise, behind the displacement front, the solution is smooth and slowly varying and can hence be evolved on a coarse grid. In the absence of capillary forces, or other second-order terms in the transport equation (2), the displacement front is a discontinuity that needs high grid resolution to be accurately approximated. Motivated by these observations, we will in this section demonstrate how the simulation accuracy can be significantly improved by dynamically adding local resolution near strong saturation fronts. Somewhat similar ideas were used by Lee et al. [16] and Zhou et al. [21] in their adaptive multiscale finite-volume method.

Because all grids considered herein are obtained by coarsening an underlying fine grid, it is relatively straightforward to add local refinement by manipulating the partition vector p, giving a local resolution that may be less or equal that of the fine grid. Moreover, this refinement can be added or removed dynamically provided we have good indicators of when to do so. Herein, we rely on the simplest approach possible, namely to compute each saturation step twice: once with a coarse resolution to estimate the movement of the front and once (locally) with higher resolution to resolve the movement more accurately. After the first step, we mark all blocks in which the saturation change from the previous time step exceeds a prescribe tolerance. These blocks are then refined. Likewise, after the second saturation step, we go through all refined blocks and mark those where the total fine-scale saturation changes are below another prescribed tolerance. In each marked block, the saturations are averaged back onto the original coarse block, and the refinement is removed by manipulating the partition vector.

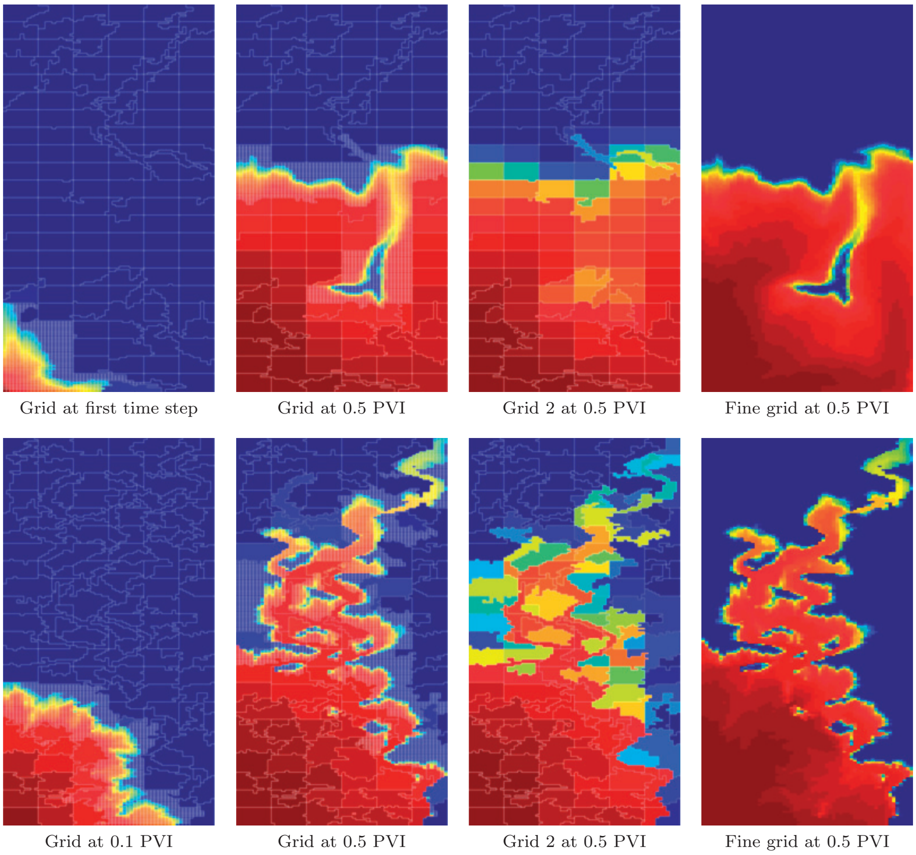
Example 6 (Layers 1 and 37 from SPE10)**.** *We revisit the two models studied in Example 4. In the fluid model we use quadratic relative permeabilities with a viscosity of 1 cP for the displacing fluid (water) and a viscosity of 0.2 cP for the displaced fluid (oil). This favourable displacement ratio will lead to a sharp front, for which local refinement of the grid is needed to avoid excessive numerical smearing when using a relatively coarse grid.*

Figure 8 shows the solution at dimensionless time 0.5 PVI computed on a coarse grid (Grid 2 from Figure 3), a coarse grid with local adaptive refinement, and on the original 60×220 *Cartesian grid. For Layer 1, we notice the excessive smearing at the fluid front, midway through the reservoir, but also that the coarse grid completely fails to capture the pocket of bypassed oil. The adaptive grid, on the other hand, is in good correspondence with the fine-scale reference solution. For Layer 37, most of the flow is confined to an intertwined pattern of narrow high-flow channels. Outside these channels, the static grid has relatively coarse blocks and once fluid enters these large low-flow blocks, the saturations get spread over a large area, causing excessive numerical diffusion.*

Figure 9 shows the corresponding errors as a function of time. The coarse-scale saturation error is largest initially and decays toward water breakthrough; the qualitative behaviour of the fine-scale error is almost identical and the corresponding curves are therefore not reported. It may come as a surprise how well the static flow-based grid captures the water-cut curve for Layer 1, given the large initial error, but this result is in correspondence with previous observations [2, 15] both for Cartesian and corner-point models. The most interesting result, however, is how much both the coarse-scale and projection errors are reduced by adding dynamical refinement.

The previous example used a simple refinement approach in which all blocks marked for refinement were replaced with the underlying fine grid. We have also experimented with more advanced options, like adding an intermediate resolution and using flow-based coarsening with finer thresholds N_L and N_U in the refinement areas. Two examples of such grids are shown in Figure 10, where we (for illustration purposes) have used a uniform Cartesian coarse grid and added flow-based refinement dynamically along the sharp displacement fronts. Likewise, one can relatively easily implement multilevel approaches. However, for the relatively simple water-flooding scenarios we have considered, these ideas have so far not been worth the (slight) increase in algorithmic complexity.

In the previous section, we briefly discussed the need for efficient communication between a multiscale flow solver and a transport solver working on a flow-adapted grid. Introducing local refinements would potentially reintroduce the need to communicate through the fine-scale grid to dynamically provide fine-scale fluxes in refined blocks. However, it has previously been demonstrated by Kippe et al. [14] that to accurately capture the dynamic changes in the flow field, it is sufficient to update the multiscale basis functions only when a strong saturation front passes through a coarse block. Basis functions will therefore typically be updated in



Grid at first time step

Grid at 0.5 PVI

Grid 2 at 0.5 PVI

Fine grid at 0.5 PVI

Grid at 0.1 PVI

Grid at 0.5 PVI

Grid 2 at 0.5 PVI

Fine grid at 0.5 PVI

Figure 8. Examples of locally adapted grids for Layer 1 (top row) and Layer 37 (bottom row) of the SPE10 model.

the regions where the transport grid is refined, and hence all the necessary fine-scale fluxes will be available. After some time steps, the strong fronts will have left the region, the coarse blocks have been reintroduced, and we can go back to use precomputed saturation mappings and sparse representation of fine-scale fluxes. For adverse mobility ratios, weak saturation fronts and smooth changes in the saturation implies that neither a dynamic refinement nor dynamic updates of basis functions are necessary.

7. Concluding Remarks

In this paper, we have shown that flow-based coarsening is a versatile method to develop efficient transport solvers that can be used in combination with multiscale flow solvers. In particular, we have started with a method proposed by Aarnes et al. [2] and shown how the partitioning computed by this method can be improved by including *a priori* information

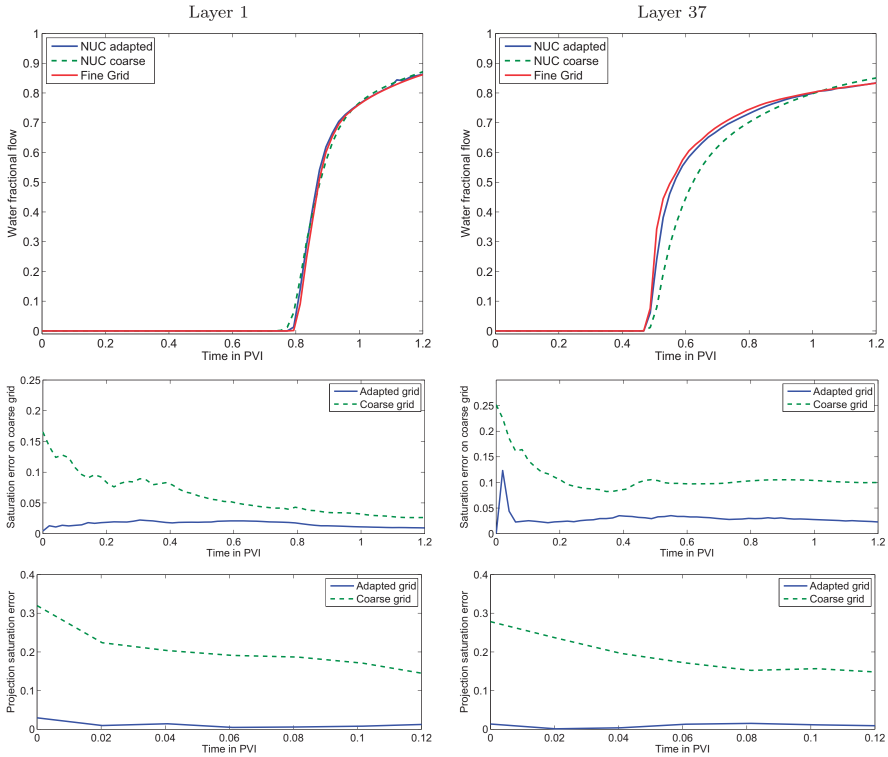
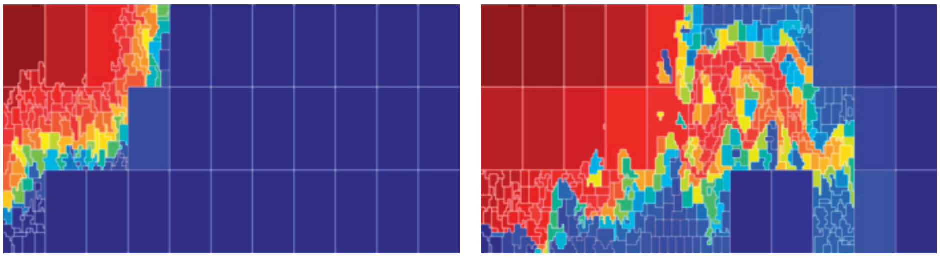


FIGURE 9. Water-cut curves and coarse-scale and projection errors as function of time for Layers 1 and 37 of the SPE10 model.



Grid at 0.1 PVI

Grid at 0.5 PVI

FIGURE 10. Uniform Cartesian coarsening with adaptive flow-based refinement for Layer 37 of SPE10. The figures show the dynamic grid at dimensionless time 0.1 and 0.5 PVI, respectively.

about block shapes or geology. Moreover, we have shown that using net fluxes rather than bi-directional fluxes over the coarse-block interfaces leads to both improved accuracy and computational efficiency. Finally, we have demonstrated how dynamical adaptivity easily can be included in the method to improve both the evolution and representation errors of strong saturation fronts. To simplify the presentation, most of the examples have focused on the widely-used SPE10 model. However, we have also included two examples that demonstrate that the ideas also apply to corner-point and PEBI grids.

8. Acknowledgments

The research was funded in part by Shell Norge AS and the Research Council of Norway through grants no. 175962 and 186935. The authors thank Arne Skorstad (Norwegian Computing Center) for providing access to the SAIGUP model.

References

[1] J. E. Aarnes, V. Kippe, and K.-A. Lie. Mixed multiscale finite elements and streamline methods for reservoir simulation of large geomodels. *Adv. Water Resour.*, 28(3):257–271, 2005.

[2] J. E. Aarnes, V. L. Hauge, and Y. Efendiev. Coarsening of three-dimensional structured and unstructured grids for subsurface flow. *Adv. Water Resour.*, 30(11):2177–2193, 2007.

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

[4] M. A. Christie. Upscaling for reservoir simulation. *J. Pet. Tech.*, 48(11):1004–1010, 1996. doi: 10.2118/37324-MS.

[5] 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. Url: http://www.spe.org/csp/.

[6] L. J. Durlofsky. Upscaling of geocellular models for reservoir flow simulation: A review of recent progress, 2003. Presented at 7th International Forum on Reservoir Simulation Bühl/Baden-Baden, Germany, June 23–27, 2003.

[7] L. J. Durlofsky. Upscaling and gridding of fine scale geological models for flow simulation, 2005. Presented at 8th International Forum on Reservoir Simulation Iles Borromees, Stresa, Italy, June 20–24, 2005.

[8] L. J. Durlofsky, R. C. Jones, and W. J. Milliken. A nonuniform coarsening approach for the scale-up of displacement processes in heterogeneous porous media. *Adv. Water Resour.*, 20:335–347, October 1997. doi: 10.1016/S0309-1708(96)00053-X.

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

[10] C. L. Farmer. Upscaling: a review. *Int. J. Numer. Meth. Fluids*, 40(1–2):63–78, 2002. doi: 10.1002/fld.267.

[11] V. L. Hauge, K.-A. Lie, and J. R. Natvig. Grid coarsening based on amalgamation for multi-fidelity transport solvers. *submitted*, 2010.

[12] T. Y. 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.

[13] 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.

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

[15] S. Krogstad, V. L. Hauge, and A. F. Gulbransen. Adjoint multiscale mixed finite elements. In *SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 2–4 February 2009*, 2009. doi: 10.2118/119112-MS.

[16] S. Lee, H. Zhou, and H. Tchelepi. Adaptive multiscale finite-volume method for nonlinear multiphase transport in heterogeneous formations. *J. Comput. Phys*, 228(24):9036 – 9058, 2009. ISSN 0021-9991. doi: 10.1016/j.jcp.2009.09.009.

[17] T. Manzocchi, J. N. Carter, A. Skorstad, B. Fjellvoll, K. D. Stephen, J. A. Howell, J. D. Matthews, J. J. Walsh, M. Nepveu, C. Bos, J. Cole, P. Egberts, S. Flint, C. Hern, L. Holden, H. Hovland, H. Jackson, O. Kolbjørnsen, A. MacDonald, P. A. R. Nell, K. Onyeagoro, J. Strand, A. R. Syversveen, A. Tchistiakov, C. Yang, G. Yielding, , and R. W. Zimmerman. Sensitivity of the impact of geological uncertainty on production from faulted and unfaulted shallow-marine oil reservoirs: objectives and methods. *Petrol. Geosci.*, 14(1):3–15, 2008.

[18] J. R. Natvig and K.-A. Lie. Fast computation of multiphase flow in porous media by implicit discontinuous Galerkin schemes with optimal ordering of elements. *J. Comput. Phys.*, 227(24):10108–10124, 2008. ISSN 0021-9991. doi: 10.1016/j.jcp.2008.08.024.

[19] J. R. Natvig and K.-A. Lie. On efficient implicit upwind schemes. In *Proceedings of ECMOR XI, Bergen, Norway, 8–11 September*. EAGE, 2008.

[20] 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. In *SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 2–4 February 2009*, 2009. doi: 10.2118/119132-MS.

[21] H. Zhou, S. H. Lee, and H. A. Tchelepi. Multiscale finite volume formulation for the saturation equations. In *SPE Reservoir Simulation Symposium, The Woodlands, TX, USA, 2–4 February 2009*, 2009. doi: 10.2118/119183-MS.

Grid Coarsening Based on Amalgamation for Multi-Fidelity Transport Solvers

Vera Louise Hauge, Knut–Andreas Lie, and Jostein R. Natvig

Preprint

Paper IV

GRID COARSENING BASED ON AMALGAMATION FOR MULTI-FIDELITY TRANSPORT SOLVERS

VERA LOUISE HAUGE, KNUT–ANDREAS LIE, AND JOSTEIN R. NATVIG

Abstract

The paper considers grid coarsening for use in multi-fidelity transport solvers in combination with multiscale flow solvers. We propose three coarsening principles, formulated as minimization problems and suggest a general framework of heuristic (minimization) algorithms. The algorithmic framework consists of a set of modular components that can be combined in different ways to create fit-for-purpose grids. The fundamental characteristic of all algorithms, is that coarse blocks are generated by amalgamating cells from the original fine grid, with a cell-wise indicator function guiding the amalgamation directions and the new grid resolution. We discuss different choices of indicator functions, how to extend or constrain amalgamation directions, and various strategies for controlling the grid resolution. Our emphasis is on qualitative examples that demonstrate the coarsening concepts and the flexibility of our framework. However, a few quantitative comparisons are also included.

1. Introduction

To optimize recovery of oil and gas fields, engineers need to consider many different scenarios and evaluate different options for enhanced recovery, in particular for mature and brown fields. In doing so, the engineers must utilize static and dynamical data that have widely different spatial (and temporal) resolution and hence need multi-fidelity modelling capabilities and simulators that offer a lot of flexibility and scalability. For assets with a production history, an important step is to assimilate dynamical data from multiple sources into preexisting earth models, which may contain millions of grid cells. Likewise, to develop new earth models, it is often necessary to perform flow simulations on multiple realizations and at different spatial resolutions to determine the correct model fidelity and spatial resolution in the geological model. To this end, there is a growing need for more accurate and robust computational methods that can seamlessly move between different spatial and temporal resolutions and be used to reduce model fidelity (e.g., the number of degrees of freedom) to a level that is sufficient to resolve flow physics and satisfy requirements on computational costs, while preserving the important characteristics of the underlying static and dynamic data.

To reduce the turnaround time of modelling, the model rescaling should be as automated as possible. Today, model rescaling are typically manual (and work-intensive) workflows, based on a combination of upgridding and upscaling. Upgridding[1] is the process of generating a model with *increased* spatial resolution. Upscaling is the process of creating a grid model with reduced spatial resolution (grid coarsening) and then bringing reservoir properties from a

Date: September 16, 2010.

[1] The word 'upgridding' is also sometimes used for the opposite process of creating a grid with *decreased* spatial resolution, i.e., what we in the following will refer to as '(grid) coarsening'.

model with higher resolution down to the new coarse grid using an appropriate approximation method, see the reviews in, e.g., [8, 11, 12, 16].

In most coarsening methods, the size of each block in the new grid is determined from a background quantity; a common approach is to define a density measure and generate a new grid with blocks that equilibrate this measure. Alternatively, one can use an indicator and define coarse blocks so that they minimize the indicator variance within the blocks and maximize the variance between different blocks. Likewise, it is common to impose geometrical constraints that determine the shape of the new grid blocks and the degree to which they align with the cells in the original grid. Density measures and indicators can be defined using *a priori* quantities like petrophysical parameters; flow-based quantities like fluid velocity, vorticity, or streamlines; or statistical or *a posteriori* goal-oriented error indicators that measure how a particular point influences the error in production responses or other predefined quantities. In particular, flow-based coarsening has been shown to be a powerful approach in combination with upscaling. The basic goal of flow-based gridding is to introduce higher resolution in regions of high flow while allowing coarser resolution in regions of lower flow. Flow-based gridding techniques have been developed for structured grids and for unstructured and triangular grids.

In the following we will shortly review some of the relevant literature on grid coarsening. Garcia et al. [17] use permeability as the background indicator used to coarsen the grid. King [22] reviews advances in grid coarsening based on static *a priori* properties, whereas King et al. [23] explore *a priori* local error analysis to generate upscaled models by vertical combination of layers. Evazi and Mahani [14] investigate Delaunay tessellation based on different grid-point density indicators, such as permeability variations, fluid velocity, and vorticity. Prevost et al. [29] describe how to incorporate both grid-resolution targets and geometrical constraints. Constrained gridding with respect to the underlying reservoir geometry is explored by Branets et al. [6], who consider constrained two-dimensional Delaunay and Voronoi grids. Durlofsky et al. [13] present a nonuniform approach that is constrained to be aligned with the underlying fine grid. Techniques based on streamlines are investigated by Castellini et al. [7], Cirpka et al. [10], He and Durlofsky [20], Verma and Aziz [30], Wen et al. [31]. Later research considers the use of vorticity to determine spatial resolution of coarse grids: Mahani and Muggeridge [25] construct coarse grids that seek to preserve high vorticity regions; Ashjari et al. [4] present a more rigorous approach for optimal coarse grids; and Evazi and Mahani [15] demonstrate a technique for generating unstructured coarse Voronoi grids.

In all the methods reviewed above, grid coarsening is used in combination with an upscaling procedure that transfers reservoir properties to the new grid. Herein, we will focus on coarsening within a multiscale setting. We propose a generic framework in which coarse grids are formed by amalgamating cells from an underlying fine grid into coarse block according to cell-based indicator functions. This gives grids with blocks having complex polyhedral forms that follow the geometries of the original fine grid. In particular, we focus on indicator functions that reflect features in the fine-scale flow field and hence generate flow-adapted grids that can be used to construct multi-fidelity transport solvers that use conservative fine-scale fluxes to accurately transport coarse-scale saturation values, when combined with robust multiscale flow solvers [2] that compute conservative fluxes that are available on coarse, intermediate, and fine grids. As such, our framework extends and generalizes the nonuniform coarsening method proposed by [1], as well as more recent developments reported in [19]. A different multiscale approach is described by Lee et al. [24], Zhou et al. [32], who utilize an adaptive multiscale finite-volume flow solver together with dynamic grid coarsening and approximate prolongation operators for the transport equation.

Our new framework is very general and is presented as a set of coarsening principles that each can be formulated as the minimization of a functional of any suitable cell-wise indicator function. The resulting optimization problem is not well posed and must generally be accompanied by a set of geometrical constraints to allow a unique solution. However, rather than solving the optimization problem, we propose to use various heuristic algorithms that will semi-automatically produce multi-fidelity, fit-for-purpose grids of sufficient quality. The specific algorithmic choices in the different coarsening methods can be considered to be part of a more general heuristic approach. The main purpose of the paper is to demonstrate the flexibility of the framework and relate it to existing techniques. Hence, most examples focus on qualitative, rather than quantitative, comparisons. A few quantitative comparisons are included in the last section, but extensive numerical testing of the relative accuracy, robustness, and efficiency of the various algorithmic choices is left for later.

2. Coarsening Principles

In the following, we consider coarsening of a fine grid defined over a singly-connected domain Ω. The grid is assumed to consist of a set of grid cells, $\{c_i\}_{i=1}^n$. No assumptions are made on the geometry of the grid, whereas for the topology we assume that there is an explicit mapping $\mathcal{N}(c_i)$ between cell c_i and its nearest neighbours c_ℓ. On top of this fine grid, we will construct a coarse grid by amalgamating (grouping) sets of cells into blocks B_j, $j = 1, \ldots, N$. The simplest way of representing such a coarse grid is by a partition vector p with n elements, for which element p_i assumes the value j if cell c_i is member of block B_j. This definition of a coarse grid is very general and allows for blocks that potentially may be multiply connected. However, unless stated otherwise, we will henceforth tacitly assume that the blocks B_j are singly connected at all times.

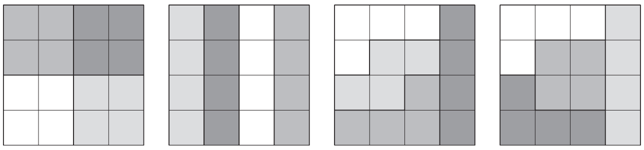
Our main interest is to develop semi-automated algorithms that produce good coarse grids for transport problems on the (simplified) form:

$$\phi\frac{\partial S}{\partial t} + \nabla\cdot\big(\vec{v}f(S)\big) = q. \tag{1}$$

To form a good coarse grid for fluid transport, one would ideally seek the particular grid that minimizes the error in the fluid saturations and/or the error in selected production responses for a given computational cost. The saturation error has two important contributions: The projection error is the difference between a fine-grid saturation field and the same field prolongated onto the coarse grid. The evolution error measures the difference between how an initial saturation field evolves on the fine and the coarse grid. We will come back to these errors later in the paper.

Finding rigorous error estimates is difficult both from a theoretical and computational point of view. Herein, however, we tacitly assume the existence of suitable *indicator functions* that can be used as good guidelines for choosing the grid resolution. Furthermore, we assume that each indicator function takes the value $I(c_i)$ in cell c_i and that these values can be interpreted as densities, i.e., that they are positive, additive, and normalized by the cell volume $|c_i|$. Using the indicator function, we will amalgamate or group fine grid cells into blocks. As for the cells, we associate indicator values $I(B_j)$ to each block, defined as the volumetric average of the cell indicator values within the block (or as the arithmetic average if I is not a density).

We propose that the amalgamation of cells into blocks should follow three main principles:



	Grid 1	Grid 2	Grid 3	Grid 4
Face/volume	2	2.5	2.5	2.5
Centroids	11.3	16	14.2	13.8
Normals	3	3	5	4

FIGURE 1. Four different ways of coarsening a 4×4 grid that all satisfy the coarsening principles for $I \equiv 1$. For each grid we report three regularity measures: (1) the mean ratio between face length and block area, (2) the total sum of distances between cell and block centroids, and (3) the average number of changes in the normal vector along each block circumference.

CP-1: *The heterogeneity of the flow field should be minimized inside each block.* Mathematically, this can be formulated as follows

$$\min_{B_j} \Big(\sum_{p_i = j} |I_1(c_i) - I_1(B_j)|^p \, |c_i| \Big)^{\frac{1}{p}}, \qquad 1 \le p \le \infty, \tag{2}$$

where I_1 is a measure of the heterogeneity in the flow field.

CP-2: *The error contribution should be equilibrated over the grid blocks.* Mathematically, this can be formulated as follows

$$\min \Big(\sum_{j=1}^{N} |I_2(B_j) - \bar{I}_2(\Omega)|^p \, |B_j| \Big)^{\frac{1}{p}}, \qquad 1 \le p \le \infty, \tag{3}$$

where $I_2(B_j)$ is a measure of the average error contribution from block B_j and $\bar{I}_2(\Omega)$ is the volumetric average of this indicator over the whole grid.

CP-3: *The size of each block should be within prescribed lower and upper bounds.*

The motivation for CP-1 is to force the grid to adapt to the heterogeneity in the flow, assuming that isocontours of I_1 must be a good measure of flow patterns. (In many cases, $I_1 = I_2$). Previous research has shown that good *ad hoc* indicator functions can be computed from, e.g., *a priori* quantities like permeability or from flow-based quantities like fluid velocity and vorticity obtained from a single-phase or multiphase flow computation. However, the ideas presented in the following may also be applicable to sensitivities or error measures obtained from adjoint computations or *a posteriori* error analysis.

Minimization of the two functionals (2) and (3) will generally not lead to a uniquely defined coarse grid. To illustrate the lack of uniqueness, we consider a square 4×4 grid with a constant indicator function $I \equiv 1$ and prescribe the number of coarse blocks to be equal four. Figure 1 shows four possible coarse grids that all meet the coarsening principles. This demonstrates that further restrictions or guidelines on shapes, aspect ratios, simplicity, and regularity of blocks need to be imposed to uniquely define a coarse grid. In this particular example, we can immediately tell by visual inspection that Grid 1 is the one that most resembles the underlying fine grid. This grid also minimizes the three regularity measures given in the tabular.

In general, it may prove quite difficult to formulate a practical and well-posed minimization problem for optimal coarsening. Not only is it difficult to formulate good coarsening criteria, but even if such criteria can be formulated, it can be difficult to determine whether they should be imposed as part of the objective functional or as hard or soft constraints. Rather than trying to solve a minimization problem to optimality, we will in the following stick with our insufficient coarsening principles introduced above and instead focus on developing a general framework of efficient heuristic algorithms that produce grids of acceptable quality for particular purposes.

3. Heuristic Minimization Algorithms

This section outlines a general framework of heuristic algorithms that produce flow-based coarse grids by amalgamation of cells from an underlying fine grid. The framework is quite general and enables us to choose from a large range of principles to develop specific fit-for-purpose heuristic algorithms. The framework is composed of four types of configurable algorithmic components: partition, intersection, merging, and refinement. Algorithmically, the partition is a source primitive, intersection and merging are filter primitives, while the refinement can either be another source primitive or consist of a combination of source and filter primitives.

Partition. The primitive takes the grid and an indicator function, or some other auxiliary information and computes a partition vector. This partition vector will either be constructed based on a prescribed coarse-grid topology or a set of predefined shapes for the coarse blocks, or be based on an initial segmentation of the cells into bins according to the value of an indicator function. In the first case, the partition will typically focus on principle CP-3, imposing, e.g., a regular partition in index space for grids with logical numbering like logically Cartesian corner-point grids. For the segmentation, we assume that the indicator function I is scaled to the interval $[1, M+1]$. Then segmentation bins $\tilde{B}_\ell$ are defined as follows:

$$c_i \subset \tilde{B}_\ell \text{ if } I(c_i) \in [\ell, \ell+1).$$

Each bin $\tilde{B}_\ell$ may consist of a multiply connected set of cells.

Intersection. The primitive takes one or more partition vectors as input—the vectors will typically represent different partition criteria—and produces a new admissible partition vector as output. The input vectors, or their intersection, may potentially have blocks that are multiply connected. The primitive splits multiply connected blocks into sets of singly connected cells to produce a new set of admissible blocks $\{B_j\}$. If necessary, the splitting can be implemented to only allow a restricted set of connections; a detailed discussion is given in Section 4.

Merging. The primitive takes a partition vector and one or more indicator functions as input, merges blocks that are too small, and produces a new partition vector as output. That is, if block B violates the condition

$$I(B)\,|B| \geq \frac{N_L}{n}\,\bar{I}(\Omega)\,|\Omega|, \tag{4}$$

for a prescribed constant N_L, the block is merged with the neighbouring block B' that has the closest indicator value, i.e.,

$$B' = \operatorname{argmin}_{B'' \subset \mathcal{N}(B)} |I(B) - I(B'')|. \tag{5}$$

Lower bounds on the block volumes can be enforced by setting $I \equiv 1$. More than one condition can be imposed, and one can also use different indicator functions in (4) and (5).

Refinement. This component of the algorithm should take a partition vector and one or more indicator functions as input, refine blocks B violate the condition

$$I(B)\,|B| \le \frac{N_U}{n}\,\bar{I}(\Omega)\,|\Omega|, \tag{6}$$

for a prescribed constant N_U, and produce a new partition vector as output. To this end, there are many different methods, as will be discussed in Section 6. Like in the merging primitive, an upper bound on block volumes is enforced by setting $I \equiv 1$, and more than one indicator condition can be imposed.

The simplest approach is to apply these four primitives in sequence, but they can also be combined in other ways and be applied to subregions with different configuration in each subregion, or be invoked in a hierarchical or recursive manner.

When the four primitives are applied in sequence to the whole grid at once, the resulting algorithm can be seen as a generalization of the nonuniform coarsening method proposed by Aarnes et al. [1]. Depending on how the refinement is performed, one may have to iterate over the merging and refinement steps (and perhaps progressively narrow the $[N_L, N_U]$ intervals) until some sort of optimality is achieved. However, it is our experience that one extra iteration of the merging primitive generally is sufficient.

Going back to the three coarsening principles introduced in the previous section, we see that partition primitive based upon segmentation minimizes the indicator variation inside each coarse block and hence enforces principle CP-1. Likewise, one can foresee other *a priori* partitions that try to fulfil CP-2 through the use of expert knowledge. However, our backup mechanisms to enforce CP-2 is through the merging and refinement steps, which seek to satisfy a relaxed version of CP-2 by restricting the variance in the indicator values to a prescribed interval. The bound on the block volumes (CP-3) can be introduced likewise by using $I \equiv 1$. A particular feature of the algorithmic framework is that it is generally difficult to strictly control the number of coarse blocks; setting N_L and N_U close to each other will give more control on the number of blocks, but will also restrict the ability to effectively adapt the grid to flow patterns.

To make our discussion more concrete, we consider the nonuniform coarsening (NUC) algorithm proposed by Aarnes et al. [1]. Parts of this algorithm will be important ingredients in other examples presented later in the paper.

Example 1. *The key idea of the NUC algorithm is that a good transport grid should separate high-flow regions from low-flow regions. In addition, the grid should have a lower bound on the block volumes and an upper bound on the flow through each block. This can be obtained from our general framework by the following algorithmic choices: The initial partition is generated by segmenting the cells into ten bins. As indicator function, the algorithm uses the logarithm of the fluid velocity as indicator function, except in (4), where $I \equiv 1$ is used to set a lower bound on block volumes. The refinement is implemented as a greedy four-step algorithm:*

(1) *Pick an arbitrary cell c_0 belonging to ∂B.*
(2) *Find the cell $c_1 \subset B$ that is furthest away from c_0 (e.g., using the distance between the cell centroids as a metric).*

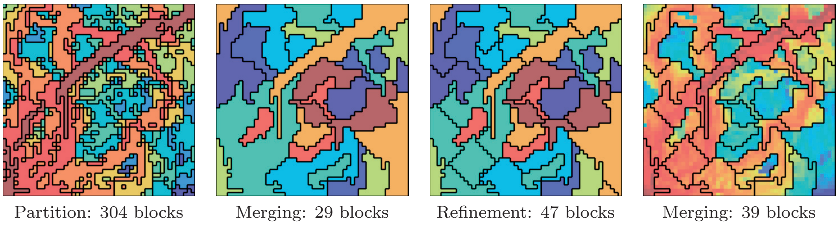

Partition: 304 blocks Merging: 29 blocks Refinement: 47 blocks Merging: 39 blocks

Figure 2. Illustration of the NUC algorithm for a quarter five-spot simulation of a 50 × 50 excerpt from Layer 68 from Model 2 of the SPE10 benchmark [9]. The coarsening parameters are $N_L = 20$ and $N_U = 100$. In the right-most plot, the colours represent the logarithm of the underlying fine-scale velocity field.

(3) *Define* $B' = c_1$ *and progressively enlarge* B' *by adding cells surrounding* B' *until the upper bound* (6) *is reached. Neighbours to* B' *are determined using face connections (i.e., the 5-point connection for 2D Cartesian grids).*

(4) *Define* $B = B \setminus B'$ *and continue to refine* B *if the upper bounds are still violated.*

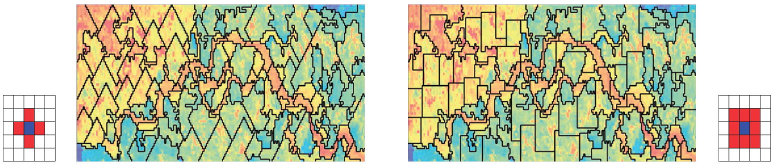
Because of the way the splitting is performed, we will typically have to reiterate the merging primitive to get rid of small blocks.

Figure 2 shows the coarse grid after each of the four stages in the NUC algorithm. In the initial partition, the coarse blocks are coloured with the ten colours representing the ten segmentation bins. For the merging and refinement, the coarse blocks are coloured with random colours to better show the sizes and shapes of the coarse blocks, while the coarse grid obtained after a second merging step is outlined on top of the logarithm of the fine-scale velocity field. The segmentation generates ten multiply-connected bins that are split into 304 small and large blocks. The merging primitive reduces the number of blocks to 29. Several of these blocks carry too much flow and are hence split into smaller blocks in the refinement step. This splitting results in some blocks that have too small volume and these are merged with one of their neighbours by a second call to the merging primitive, creating a grid with 39 blocks in total.

In the next four section, we will discuss how to develop specific algorithms that fall into the framework outlined above. Our discussion will focus on qualitative behaviour of the various algorithmic realizations; quantitative comparisons are given in Section 8. We start by examining two algorithmic choices that can be used to configure the primitives: Section 4 discusses how the definition of admissible neighbours affects the two filter primitives (intersection and merging) and the refinement step, whereas Section 5 gives several examples of flow-based indicators. Then, Section 6 goes into more details about the refinement step, before we end the algorithmic overview by giving examples of regional, hierarchical, and recursive combinations of the four primitives in Section 7.

4. Definition of Admissible Neighbours

A main characteristic of our amalgamation approach is that coarse blocks are formed by collecting neighbouring cells from a fine-scale grid. Feasible directions for amalgamation are defined by the indicator function, whereas the neighbour definition defines the admissible directions. Information about admissible amalgamation directions is used in all three types of filter primitives and has a strong impact on the shape (and regularity) of the resulting



5-point neighbours 9-point neighbours

Figure 3. Use of the original nonuniform coarsening algorithm [1] with two different definitions of admissible neighbours to coarsen a quarter five-spot simulation on Layer 37 of SPE10 [9].

coarse blocks. In the intersection primitive, one needs to know the admissible neighbours of each cell in the fine grid when postprocessing partition vectors to ensure that all blocks are singly connected. In the merging primitive, the algorithm searches through the admissible neighbours to find blocks that can be merged. Likewise, the refinement step will typically use neighbourship among individual cells to split a block into multiple parts. The NUC algorithm (Example 1), for instance, needs to know all the neighbours of a given cell on the boundary of B' to determine which cells it should add when growing B'. How to define the neighbourhood of a cell, is therefore one of the important algorithmic choices in our framework.

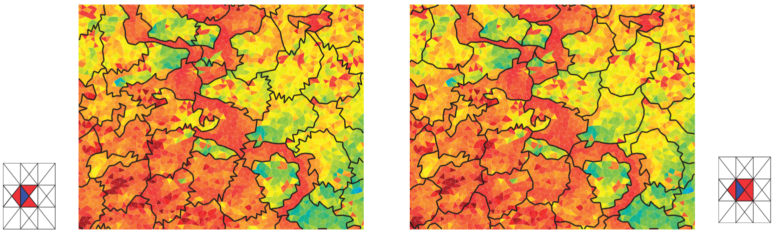
4.1. **Topology.** The basic neighbour definition is that introduced by the topology of the grid. In 2D, level-one neighbours share a common edge and level-two neighbours share a common point. For a logical Cartesian grid, this corresponds to the standard 5-point and 9-point connections, respectively. However, as we will see later, concrete algorithms can also use other definitions of admissible neighbours (e.g., only vertical or horizontal neighbours, etc). In 3D, level-one neighbours share a common face, level-two neighbours share a common edge, and level-three neighbours share a common point.

To illustrate the effect the definition of permissible neighbours can have on a specific algorithm, we consider the NUC algorithm from Example 1 applied to a 2D quarter five-spot simulation. Figure 3 shows the difference between using a 5-point and 9-point stencil when growing new blocks during Step 3 of the refinement algorithm. With a 5-point stencil, the algorithm will create diamond-shaped blocks, whereas rectangular blocks are created using 9-point neighbours. A large number of numerical experiments show that the latter grids typically give slightly smaller errors in saturation and production profiles.

For triangular and tetrahedral grids, the use of level-one neighbours will typically lead to quite irregular interfaces between coarse blocks. Figure 4 shows the corresponding coarse grid generated using: (i) face neighbours only, and (ii) face neighbourhood extended with cells that share faces with two face neighbours. The latter definition clearly leads to more regular block interfaces, and can be used as an alternative to level-two and level-three neighbours. Alternatively, one could also imagine adding an extra primitive that performs edge minimization after the refinement.

Finally, we present an example in which we go the other way and demonstrate that one can also restrict the set of admissible neighbours, and only coarsen the grid in one of the spatial directions. The example algorithms are motivated by recent work by King [22].

Example 2. *We consider a 2D vertical slice of a low-permeable reservoir that contains a few high-permeable channels. Because of the large variance in media properties, we use a*



face neighbours

extended neighbourship

FIGURE 4. Coarsening of an unstructured triangular grid with permeability sampled from Layer 37 of SPE10. The plots show parts of the coarse grids generated with the original nonuniform coarsening algorithm using two different definitions of admissible neighbours.

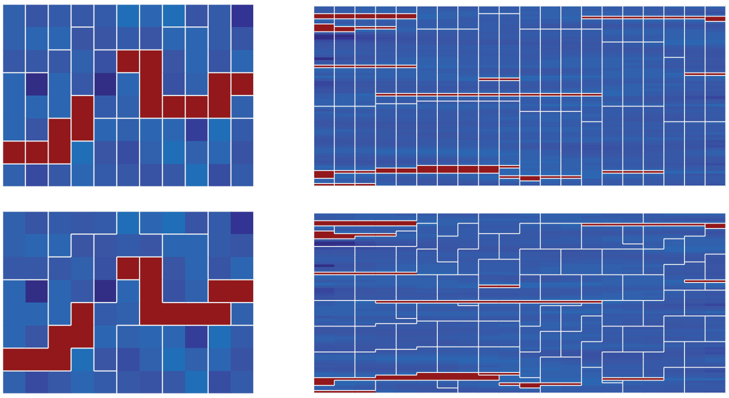


FIGURE 5. Coarsening of a system consisting of high-permeable channels on a very low-permeable background; the left column shows a 11×8 model and the right column plots a 20×70 model. The plots in the upper row are constructed using only vertical neighbours, whereas the standard 9-point neighbourship is used in the lower row.

step function that classifies permeability values as either foreground and background as our indicator function; this is an example of an a priori *indicator. In the postprocessing of the initial segmentation (into two bins: low and high permeabilities) and in the refinement step, we use the following neighbour definitions: (i) only vertical connections, that is, cell c_{kl} is connected to cells $c_{k,l\pm1}$, but not to cells $c_{k\pm1,l}$; and (ii) connections in the vertical, horizontal, and diagonal directions (9-point stencil). For the merging and refinement, we set $I \equiv 1$ and $N_L = 1$ to control the block volumes. Figure 5 shows how the resulting two algorithms coarsen a small 11×8 grid containing two high-permeable channels and a more detailed 20×70 model. In all four plots, it is clear that the grid adapts to the high-permeable regions.*

4.2. **Geometry.** Whereas postprocessing the segmentation in the intersection primitive must be based on topological neighbour definition, the refinement step can, at least in principle, use geometry to define admissible neighbours. A neighbourhood can, for instance, be defined

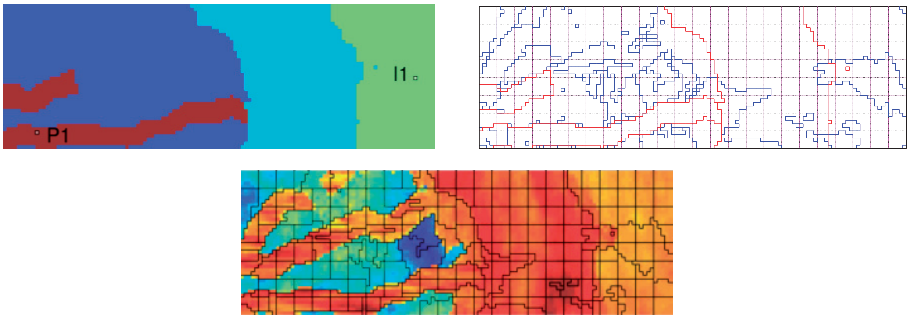


FIGURE 6. Coarsening using saturation regions as *a priori* partitioning. The upper-left plot shows the saturation regions and wells, the upper-right plot shows the three partition vectors that are combined by the intersection primitive, whereas the lower plot shows the 251 coarse blocks in the final grid plotted on top of the logarithm of the permeability field.

as all cells whose centroids are within a certain distance from the given cell. However, our experience is that uncritical use of geometrical neighbourhoods may lead to undesirable coarse blocks having highly irregular sawtooth faces, blocks not respecting geological layers, etc. A more thorough discussion is given in [2].

4.3. **Cell or face constraints.** In certain cases, it is desirable to introduce hard constraints into the coarsening algorithm to force the blocks to respect features in the underlying geology. To this end, one can use both cell-based and face-based constraints. Cell-based constraints will typically be added as extra *a priori* partition vectors that are intersected with partition vectors based on topology, geometry, and/or indicator functions by the intersection primitive. To preserve the cell-based constraints throughout merging and refinement, the set of admissible neighbours should be redefined to exclude cells on opposite sides of the artificial internal boundaries imposed by the cell-based constraints.

Hauge et al. [19] recently presented a modified NUC-type algorithm in which auxiliary grid properties were used as a hard cell-based constraint to forced each coarse block to be confined to a given facies or rock type. This way, one can avoid having to upscale relative permeabilities or capillary pressure curves, which one would have to do for blocks containing more than one saturation region. Next, we show an example of this algorithm, using saturation regions to create a grid in which each block corresponds to one rock type only.

Example 3. *We consider a* 40×120 *horizontal slice of a 3D petrophysical realization from the SAIGUP study [26]; saturation regions (rock types) and permeability are shown in Figure 6. A flow field is created by placing a fluid source (I1) to the right and a fluid sink (P1) to the left. The algorithmic choices are the following. The intersection primitive combines three different partition vectors, as shown in the upper-right plot in Figure 6: (i) a uniform* 8×24 *Cartesian partition, shown as dashed black lines; (ii) a vector representing the four saturation regions, shown as red lines; and (iii) a segmentation using the logarithm of the fluid velocity as indicator function, shown as blue lines. Here, the number of bins is equal to the span in logarithm of velocity. During merging and refinement, the saturation regions are preserved by removing the cell connections across the boundaries of saturation regions. The resulting*

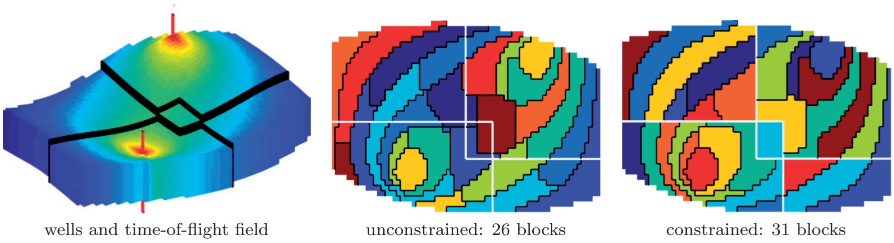


Figure 7. Unconstrained versus constrained coarsening of a homogeneous model with four intersecting faults. Logarithm of time-of-flight is used as indicator value.

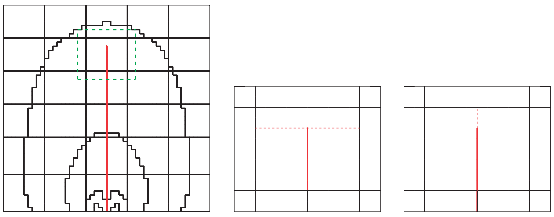


Figure 8. A constrained coarse grids, with two simple solutions defined through neighbourhood relations.

coarse grid is shown in the lower plot of Figure 6. The coarse blocks are rectangular, except at the boundaries of the saturation regions and in regions with (very) high flow. (The observant reader will observe an isolated single cell contained within one of the coarse blocks. Such artifacts can, for practical applications, be removed by a simple postprocessing).

Face-based constraints are introduced to specify faces in the original grid that the coarse blocks should not cross—typical examples are discontinuities like faults, throws, etc. Face-based constraints do not correspond naturally to any partition vector and must therefore be introduced during the postprocessing phase of the intersection primitive by removing cell connections over the constrained faces. To preserve the constraints, the set of admissible neighbours should be manipulated by removing the same cell connections also during merging and refinement. Figure 7 shows an example of a homogeneous 3D model with four intersecting faults. To generate a flow-based grid, we have employ time-of-flight as indicator function; use of this indicator will be discussed more in Section 5.1. With the standard topological neighbourship, we get a grid with coarse blocks that cross the faults. By eliminating cell connections across the fault faces, we get a constrained grid, in which no blocks cross the faults.

For the model in Figure 7, the constrained faces connect to the outer boundaries of the grid, thereby splitting the physical domain into a set of disconnected subdomains. This effectively introduces a natural partition in the same way as a cell-based constraint would do. However, this is not always the case: a constrained face may potentially be confined to the interior of the grid, leaving the partitioning of the region surrounding its edges undefined for further partitioning. In the undefined regions, one will typically get coarse blocks that encapsulate the edges of the constrained faces, as illustrated in the left plot of Figure 8. Here, the constrained face has a hanging edge inside a coarse block. By defining the cells above the tip of the constraint to not be neighbour to the cells below this point, the block can be partitioned as shown in the middle plot of Figure 8. Likewise, extending the constrained face to the top of the block results in the local partitioning shown in the rightmost plot of Figure 8. Similar *ad hoc* approaches can easily be implemented for more general cases, but may lead to undesired side-effects. A better solution might be to alert the user and let him/her make the appropriate choice.

5. Flow-Based Indicator Functions

Indicator functions have two main uses in our algorithmic framework: (i) to identify feasible amalgamation directions in the partition primitives and the refinement step, and (ii) to determine locally the correct size of the blocks during merging and refinement. In the previous sections, we have seen a few examples of indicator functions. In this section, indicator functions will be discussed in more detail, focusing on functions that are well suited to form good grids for the transport equation (1).

An important feature of our coarsening strategy based on amalgamation, using a flow-based quantity as indicator, is to match borders of important flow patterns accurately and adapt the coarse grid blocks to sharp saturation fronts. This is in contrast to density-based coarsening algorithms that generate grids with increased resolution in important regions and decreased resolution in less important regions.

5.1. **Time-of-flight.** For a stationary velocity field, the time-of-flight $\tau(\vec{x})$ is the time it takes a passive fluid particle to reach a given point $\vec{x}$ in space, starting from a fluid source or inflow boundary. The trajectory Ψ along which the particle travels is called a streamline and is an integral curve of the velocity. In mathematical terms, the time-of-flight is defined as

$$\tau(\vec{x}) = \int_{\Psi} \frac{\phi}{|\vec{v}|}\, ds, \qquad \frac{d\vec{x}}{d\tau} = \frac{\vec{v}}{\phi}, \tag{7}$$

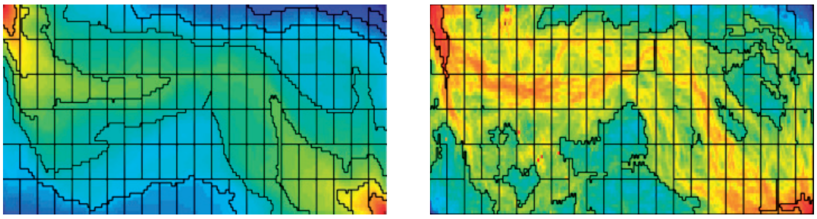
i.e., using the interstitial velocity $\vec{u} = \vec{v}/\phi$ rather than the Darcy velocity $\vec{v}$. Alternatively, τ can be expressed by the differential equation

$$\vec{v} \cdot \nabla\tau = \phi. \tag{8}$$

Isocontours of τ define natural time-lines in the reservoir and time-of-flight is therefore a natural ingredient when forming flow-based indicator functions. To see this, we can introduce the bistreamfunctions ψ and χ [5], for which $\vec{v} = \nabla\psi \times \nabla\chi$, and apply a coordinate mapping from physical space to streamline coordinates, $(x, y, z) \to (\tau, \psi, \chi)$. For incompressible flow, the transport equation (1) then simplifies to:

$$\frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial \tau} = 0 \tag{9}$$

because of (8) and the fact that $\vec{v}$ is orthogonal to $\nabla\psi$ and $\nabla\chi$. Forming a new grid using τ as a spatial coordinate means that we replace the fine mesh $\tau(c)$ in the τ-direction by a coarse mesh $\tau(B)$. Loosely speaking, we can then think of solving (1) on the coarse grid as solving (9) using the coarsely sampled $\tau(B)$ rather than $\tau(c)$, thereby introducing a natural flow adaption. In the vicinity of strong gradients, $S(\tau(c)) - S(\tau(B))$ will be of order $\mathcal{O}(1)$ in the L^∞ norm, but since this occurs in regions of small volume, the contribution to the L^p error remains small for $1 \leq p < \infty$.



Time-of-flight (189 blocks)

Logarithm of velocity (161 blocks)

FIGURE 9. Leftmost plot shows an example of time-of-flight grid using Layer 22 of the SPE10 model, simulating a quarter five-spot. The coarse grid is outlined on the logarithm of the time-of-flight measure. For comparison, the rightmost plot shows an example of the equivalent grid using logarithm of velocity.

To construct the flow-adapted grid in physical space, we introduce the *reversed* time-of-flight τ_r, that can be computed by tracing streamlines backward from fluid sinks or outflow boundaries. Then a suitable indicator function is defined as

$$I_\tau = \log\left(\frac{1}{\tau\tau_r}\right). \tag{10}$$

The inverse relationship ensures that regions in which the fluid is highly present most of the time, are assigned higher values, while smaller values are assigned in regions where it takes a long time before the fluid arrives. The left plot in Figure 9 shows an example of a coarse grid generated using the time-of-flight indicator function (10). The grid was generated by intersecting a time-of-flight segmentation with a regular Cartesian partition vector.

5.2. **Velocity.** The time-of-flight indicator introduced above generally tends to give good flow-adapted grids. On the other hand, it requires cell-based time-of-flight values, which are derived quantities that are not available in most flow simulators and must be computed by tracing streamlines or by solving (8) using an appropriate finite-volume or finite-element method, see e.g., [28]. Aarnes et al. [1] therefore suggested to use the logarithm of the velocity (i.e., either $\log(|\vec{v}|)$ or $\log(|\vec{u}|)$) to get an indicator that is based on a quantity that is available in most flow solvers. The right plot in Figure 9 shows a coarse grid generated in the same way as for the time-of-flight grid, but with $\log(|\vec{v}|)$ as indicator function.

To see why the logarithm of the velocity may give a reasonable flow indicator, we use a similar hand-waving argument as above: By introducing a new grid we effectively replace $\vec{u}$ by its projection $\vec{w}$ onto the coarse grid. Given a fixed point $\vec{x}$, this means that the corresponding time-of-flight value τ is replaced by a value $\tilde{\tau}$ and the streamline Ψ that goes through x is replaced by another streamline $\tilde{\Psi}$. From (7) and the fact that $\Psi(\tau) = \vec{x} = \tilde{\Psi}(\tilde{\tau})$, it follows that

$$\int_0^\tau \vec{u}\big(\Psi(r)\big)dr = \int_0^{\tilde{\tau}} \vec{w}\big(\tilde{\Psi}(r)\big)dr.$$

Hence,

$$\begin{aligned}|\tilde{\Psi}(\tilde{\tau}) - \tilde{\Psi}(\tau)| =& \Big|\int_\tau^{\tilde{\tau}} \vec{w}\big(\tilde{\Psi}(\tilde{\tau})\big)dr\Big| = \Big|\int_0^\tau \big[\vec{u}\big(\Psi(r)\big) - \vec{w}\big(\tilde{\Psi}(r)\big)\big]dr\Big| \\ &\le \int_0^\tau \big|\vec{u}\big(\Psi(r)\big) - \vec{u}\big(\tilde{\Psi}(r)\big)\big|dr + \int_0^\tau \big|\vec{u}\big(\tilde{\Psi}(r)\big) - \vec{w}\big(\tilde{\Psi}(r)\big)\big|dr\end{aligned}$$

Making a grid that minimizes the variation of $\log(|\vec{u}|)$ inside each block, means that we also minimize $|\vec{u}-\vec{w}|$. Hence, $|\tilde{\Psi}-\Psi|$ is small, which again implies that both the terms on the left-hand side are small. The right-hand side is the distance traversed from time τ to $\tilde{\tau}$ along $\tilde{\Psi}$. If this distance is small, then $|\tau-\tilde{\tau}|$ is also small as long as $\vec{w}$ is non-degenerate. Consequently, $|S(\tau(c))-S(\tilde{\tau}(B))|$ is small away from strong gradients in the saturation.

Numerical experiments indicate that using $\log(|\vec{v}|)$ gives a somewhat smoother indicator field than $\log(|\vec{u}|)$ and is hence preferable. For relatively homogeneous porosity there is no practical difference between using Darcy and interstitial velocities. However, for larger variations in porosity, the $\log(|\vec{u}|)$ indicator field will often exhibit speckled patterns that seem to exaggerate the impact of flow heterogeneity and generate grids with too much irregularity. Preliminary studies indicate that smoother indicator field improves the accuracy; further research is required to explain why.

Finally, we point out two important algorithmic details: First, in their original algorithm, Aarnes et al. [1] suggested that ten segmentation bins was a good choice for the velocity indicator. A large number of experiments run by the authors indicate that this choice tends to overestimate the influence of the flow heterogeneity, and in [19] we therefore proposed to set the number of segmentation bins equal $\log(\max|\vec{v}|/\min|\vec{v}|)$. Second, experience has shown that because of the large magnitude differences in flow velocities, the velocity indicator should not be used for the lower bound (4) in the merging primitive; a plain volume condition is better.

5.3. **Vorticity.** In mathematical terms, the vorticity is defined as the curl of the velocity field $\vec{v}$

$$\vec{\omega}=\vec{\nabla}\times\vec{v}=\vec{i}\left(\frac{\partial v_z}{\partial y}-\frac{\partial v_y}{\partial z}\right)+\vec{j}\left(\frac{\partial v_x}{\partial z}-\frac{\partial v_z}{\partial x}\right)+\vec{k}\left(\frac{\partial v_y}{\partial x}-\frac{\partial v_x}{\partial y}\right), \tag{11}$$

and measures the rate and direction of rotation in fluid flow. A finite-difference counterpart of the vorticity has been suggested as a good estimator for grid density in various coarsening methods [3, 4, 15, 25]. For our purposes, we use the logarithm of the vorticity, $\log(|\vec{\omega}|)$, as indicator.

Coarsening methods based on regridding typically require a smooth density estimator to avoid too abrupt changes in grid density, and because the vorticity tends to smooth velocity variations slightly, it is typically a better density estimator than velocity for regridding purposes and for smoothly varying heterogeneity. For strongly heterogeneous cases, however, one is typically interested in preserving abrupt changes in the flow field caused by large media contrasts and here smoothing is not desirable. Figure 10 shows a comparison of a velocity and a vorticity grid for a 2D fluvial reservoir with distinct flow channels that need to be reproduced accurately to guarantee low errors in saturation and reservoir-response curves. The left plots show the grids on top of the indicator fields, whereas the middle plots contrast the two grids plotted on top of a saturation field computed on the original fine grid. A zoom of the grids show that the vorticity indicator fails to capture the borders of the high-permeable channels, whereas they are accurately captured by the velocity indicator. In summary, our experience is that vorticity is a good measure to determine grid resolution during merging and refinement. Vorticity can also be used in the partition primitive to provide an initial segmentation, but here a velocity indicator will preserve flow contrasts sharper and hence give better grids, at least for strongly heterogeneous and channelized media.

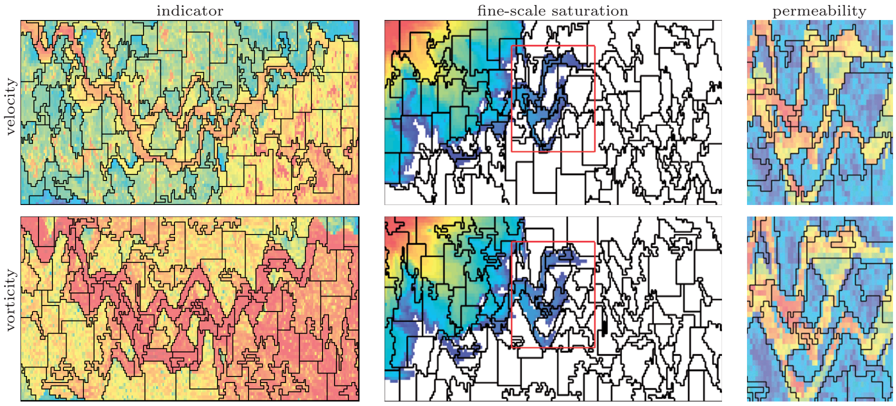


FIGURE 10. Comparison of nonuniform coarsening based on velocity and vorticity for a quarter five-spot simulation on Layer 37 of SPE10. The plots show the coarse grids plotted on top of the indicator field (left), the fine-scale saturation (middle), and the permeability field (right).

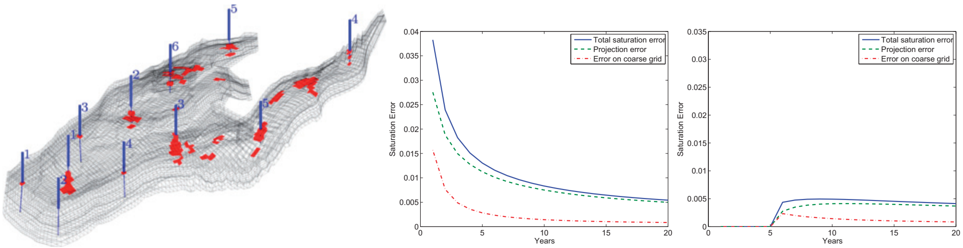
5.4. **Courant number.** The Courant number ν measures how fast information (disturbances, discontinuities) is advected by the hyperbolic part of the transport equation. Mathematically, ν is defined as the ratio between a time increment Δt and the characteristic advection time, which is given as $\Delta x/a$, where Δx is a small distance and a is the local advection speed at which information is propagated—for the transport equation (1) in 1D, the advection speed is $\frac{v}{\phi}f'(S)$. This means that for a fixed Δt, the Courant number will be large in cells where the fluids flow quickly and small in cells where the fluids flow slowly.

To develop an indicator function based on the Courant number, we drop the derivative of the fractional flow function, which is saturation dependent, and define the indicator function to be

$$I_\nu(c_i) = \frac{|\vec{q}|}{\phi|c_i|}, \tag{12}$$

where $\vec{q}$ is the flux through the cell. Although I_ν could be used in NUC-type algorithms, we will in the following employ it in a different type of algorithm.

The main use of the Courant number is to impose restrictions on the time step. For explicit methods, the time step is restricted by the CFL condition which requires that $\nu < C$, where typically $0 < C \leq 1$, depending on the specific scheme. Similarly, for implicit schemes it may be desirable to limit the Courant number to improve convergence in the nonlinear solver. Considered in terms of the coarsening principles, we have an indicator function I_ν, inversely proportional to the admissible time step, which we want to limit by an upper threshold to relax the strict time-step restriction imposed by cells with high flow. In many cases, the global time step of a model is dictated by a few cells around wells or by cells having a (very) low porosity. Hence, significantly larger time steps can be used if we perform a local coarsening to decrease the Courant number in these regions of the grid. That is, the resulting grid is coarse in the sense that some cells are amalgamated into coarse blocks, while the larger part of the grid preserves its original resolution.



Coarse grid saturation errors, coarse grid saturation errors, improved run

Figure 11. Coarsening of a real-field model to improve time-step restrictions; coarse blocks shown in red colour. Saturation errors measured for two simulation strategies: using the coarse grid for all 20 years (middle), and using the original grid for the first five years and then the coarse grid for the next 15 years (right).

The desired effects are produced by the following algorithm, which uses only one of the four primitives from the algorithmic framework[2] to obtain a specific fit-for-purpose grid.

Example 4. *The initial partition is set to be the original grid itself. Then, the merging primitive is iterated until there are no blocks where I_ν exceeds the upper threshold N_ν. The merging primitive chooses the neighbour that contributes to the largest decrease in I_ν. No upper bounds are imposed on the indicator function or on the volumes of the blocks.*

To illustrate the algorithm, we consider a real-field model consisting of 44 915 cells. Simulating the fluid transport for 20 years requires 111 560 time steps. The maximum indicator value in the grid was 281.4. To reduce the number of time steps, we impose a threshold value $N_\nu = 13.93$ *and apply the algorithm to generate a locally coarsened grid, in which 243 cells have been merged into 52 blocks. The top left plot in Figure 11 shows the modified grid with the coarse blocks marked in red colour. Although the algorithm has reduced the number of cells by less than 0.4 % to 44 724 blocks, the new grid only requires* 14 660 *time steps, i.e., 13 % of the original number of steps. This indicates that only minor changes in the grid might lead to important improvements in computational time.*

Coarsening the cells around the injection wells has a small effect on the production curves, but will, of course, affect the local accuracy of the saturation field. The middle plot in Figure 11 shows saturation errors (these errors are defined in Section 8). We observe that after five years, when the water front has moved through the coarsened blocks, the relative error of the solution is less than 1 % compared with the solution on the original grid. To improve the accuracy of the simulation, we retain the original grid for the first five years and switch to the coarsened grid afterwards. With this improved strategy, the discrepancies in saturation are kept below 0.5 %, as shown in the right plot in Figure 11. The number of time steps used for this simulation is 38 885, distributed as 5 578 time steps for each of the first five years of simulation, then 733 time steps for each year for the rest of the simulation.

[2] The observant reader will notice that we here deviate from our general framework since blocks are merged if the indicator *exceeds* an upper bound and that the merging gives *reduced* indicator values. However, what appears to be contradiction, is not—we get an algorithm following the usual setup if we instead define an indicator $\tilde{I} = I_\nu^{-1}$. Indeed, the opposite definition was chosen deliberately to illustrate this equivalence, and make the indicator correspond to the Courant number.

5.5. **Permeability.** The indicators discussed so far have assumed that an initial flow solution has been computed, either a single-phase solution or a full multiphase flow solution. However, in certain cases, one must generate a coarse transport grid prior to the flow solution. Then, $\log K$ can be used as a simplified *a priori* flow indicator. The quality of this indicator will depend on the extent to which the pressure gradient can be considered to be a constant.

Permeability is also a natural indicator that can be used to separate high-permeable or impermeable regions from a background field. In Example 2, for instance, we used a simple step function to segment the grid cells into two categories—foreground cells with high permeability and background cells with low permeability—and then generate coarse blocks consisting only of foreground or background cells. A similar technique was used in the explicit matrix-fracture separation method presented in [18].

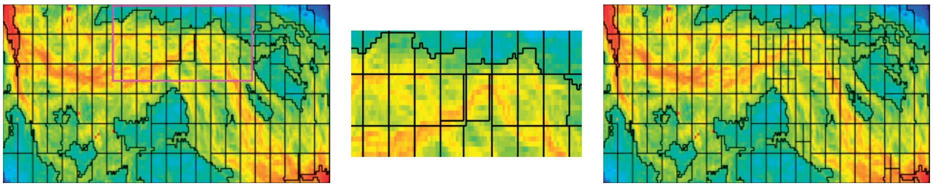
5.6. **Volume.** In some of the specific algorithmic realizations presented so far, we have used a uniformly constant indicator in combination with other indicator functions; the original NUC algorithm [1], for instance, uses $I \equiv 1$ in (4) and $I = \log(|\vec{v}|)$ elsewhere. On the other hand, using a constant as the only indicator function means that no flow information is used to determine feasible amalgamation directions and that constraints on block volumes (CP-3) are used to decide which blocks will be merged and refined. Furthermore, the shapes of the blocks will be determined by a combination of *a priori* partition vectors, the block ordering in the merging primitive, and the algorithm used to refine blocks. Using porosity rather than a constant indicator means that the merging and refinement of blocks is based on pore volume rather than geometrical (rock) volume.

6. The Refinement Step

Criteria for what is a good transport grid should, to the extent possible, be implemented using partition primitives. However, in many cases a single partition primitive may not be sufficient to give a good grid—e.g., if the flow-based indicator contains large contiguous regions of approximately constant values—or multiple primitives may interact in an undesired way, e.g., because the criteria they implement are conflicting. Either way, the result is an infeasible grid that is too fine in some regions and too coarse in other. Sometimes, an improved grid can easily be obtained by going back and adjusting the parameter values of the partition primitives, or by adding extra *a priori* expert information. However, this is not possible in general, and as a backup, our framework therefore includes auxiliary steps that merge blocks that are too small and refine blocks that are too large. Devising a good method for merging blocks is quite easy. In the following we will outline and discuss some of the alternatives for the refinement.

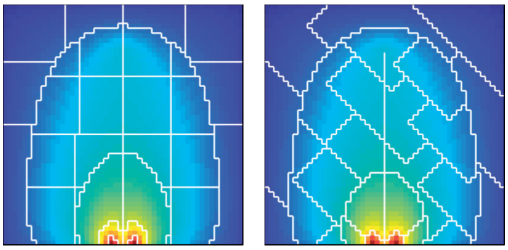
For structured grids, a simple approach is to split each single block, or a set of blocks forming a contiguous region, into multiple parts by imposing a uniform subdivision inside the bounding box in index space. For unstructured grids, an equally obvious choice would be to use a graph-based partition algorithm, e.g., as implemented in the software package METIS [21]—one such example is given in Figure 18 later in the paper. Particularly for unstructured grids, such partitioning algorithms may be a good alternative. In general, the flexibility of the framework opens up for combination with any external partitioning algorithm in the refinement step. This way, one can design algorithms that combine the strengths of flow-based and topological partitions.

Likewise, a coarsening algorithm formulated using the primitives described above can be invoked on all cells within each block that needs to be refined or on the entire region formed



greedy (161 blocks) zoom topological (189 blocks)

FIGURE 12. Illustration of a suboptimal refinement. The original grid from Figure 9 in the left plot uses a greedy refinement, whereas the grid in the right plot uses a topological 2×2 refinement.



32 blocks 26 blocks

FIGURE 13. Illustration of how identical indicator function, given two different refinement realizations produce highly different grids, in particular with respect to symmetry.

by all blocks that need to be refined. Such refinement procedures may be implemented as recursive subdivision into two or more parts according to volume, indicator function, topological and geometric information, cell constraints, etc. We will come back to a few such examples in the next section.

The original NUC algorithm [1], on the other hand, employed an *ad hoc* greedy algorithm, in which blocks were split by progressively growing new blocks inward from the boundary of the yet unrefined part of the block; the precise algorithm is given in Example 1. This algorithm works well in many cases, but may also produce quite unaesthetic blocks, in particular when combined with an *a priori* topological partitioning. To illustrate this, we consider the grid in the right plot in Figure 9, which was generated by intersecting a velocity indicator with a uniform Cartesian partition. Figure 12 shows a zoom of a subregion in the grid, where two of the blocks happen to exceed the upper bound by a very small amount. The refinement algorithm subdivides each block into a block satisfying the upper bound, leaving a thin sheet of remaining cells to form a second block. An improved grid can be obtained by replacing the greedy algorithm by, e.g., a topological 2×2 refinement as shown in the right plot of Figure 12. Figure 13 shows the greedy and the topological refinement applied to the case with a sealing fault from Section 4.3. Whereas the topological refinement basically results in a symmetric grid, except for one small mismatch, the grid created by the greedy algorithm does not. A somewhat improved and less greedy algorithm can be obtained if one stops growing when the size of the new block and the unrefined block are almost the same. Similarly, one can easily foresee improved greedy methods that incorporate feasible amalgamation directions, topological shapes, etc.

7. Combining the Algorithmic Primitives

In previous sections we have discussed how different indicator functions and definitions of admissible neighbours may be used to configure the algorithmic primitives to affect the shape of coarse grid blocks and how the coarse grid adapts to the flow pattern. The most decisive algorithmic choice, however, is how the algorithmic primitives are combined and applied to the grid. To this end, we will use a top-down approach, assuming that all algorithms start with a coarse grid that is gradually refined towards the desired resolution. This approach gives the user great flexibility to vary the coarsening degree throughout the grid and using amalgamation of fine cells to form the coarse blocks enables the coarse grid to adapt exactly to important geometrical and geological features in the original grid.

In this section we illustrate different types of regional, hierarchical, and recursive coarsening approaches. We start out by a simple example in which we use facies distribution and *a priori* information about the flow pattern to steer the resolution in different regions of the grid. Next, we take a closer look into successive refinement of a domain and discuss various aspects. Finally, examples of adaptive and supervised nonuniform coarsening are presented.

Example 5 (Regional refinement)**.** *We consider a square reservoir that consist of four facies. The permeability distribution follows a lognormal distribution inside each facies with mean values of 400, 20, 800, and 35 mD, respectively. The reservoir is sampled on a PEBI-grid using 1 697 cells as shown in the leftmost plot in Figure 14. The first facies forms two disconnected regions that have been labeled 1 and 5 in the figure, whereas the next three facies form contigous regions (labeled 2 to 4 in the figure). An injector is placed in the upper-right corner and producers are placed in the lower-right corner of facies 1 and in the lower-right corner of facies. We expect that the injected water will mainly flow through the two high-permeable facies. Hence, the five regions are gridded using different strategies: Region 1 is left with the original resolution and in Region 3 we use the original NUC algorithm (admissible neighbours = face neighbours). Region 2 lies in between the two high-flow regions, so here we use a medium coarse partitioning generated by METIS. Regions 4 is partitioned into three blocks and Region 5 is represented as one block since very limited flow is expected in these regions. The leftmost plot in Figure 14 shows the outline of the coarse grid. Simulation of a water flooding scenario reveals a good match of water breakthrough for the well in the lower-right corner. In the second well, too early water breakthrough is predicted because of the coarser grid resolution in Region 3. The leftmost plots in Figure 14 show the saturation distributions on the fine grid and on the coarse grid after breakthrough in the first well but before breakthrough in the second well.*

We observe that leaving the two regions in the upper-left corner very coarse, one not even refined at all, does not seem to affect the simulation results, since no flow actually occurs there. Likewise, leaving Facies 2 quite coarse does not significantly affect the solution as long as the flow has not arrived into this region. After reaching both wells, the flow will slowly diffuse into this region. The coarse blocks here will thus not necessarily resolve this flow very accurately. This example invites to investigate regions of importance, i.e., identify decisive connections and regions where refinement matters the most.

Example 6 (Successive refinement)**.** *We reuse the model from the example above. This time we start out with an initial coarse partitioning of the domain into 26 coarse blocks generated by METIS. The coarse blocks are refined to the original grid resolution, one by one, until the complete original grid is obtained. To determine the order in which the blocks are refined, we*

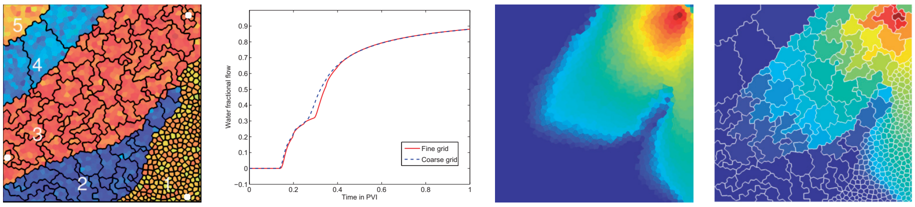
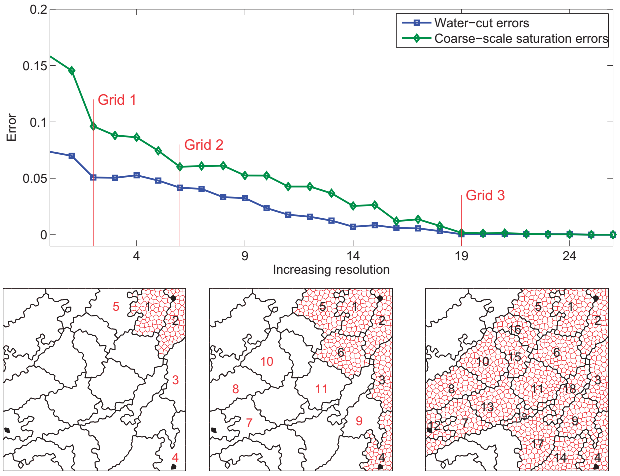


FIGURE 14. Example of regional refinement on a PEBI grid. The left plot shows the five regions (four facies), with wells shown as white dots. The second plot shows the water-cut for one pore volume of water injected. The two rightmost plots show the saturation distribution of the fine grid and coarse grid, respectively, at a stage in the simulation after water breakthrough in the first well, but before water breakthrough in the second well.



Grid 1 *Grid 2* *Grid 3*

FIGURE 15. Successive refinement using the sum of the cell-wise velocity indicator in the blocks to determine the order of refinement.

use the sum of the total velocity within each block. After each refinement, we simulate the injection of one pore volume of water and measure the discrepancy in saturation profiles and water-cuts compared with the solution computed on the original grid. Figure 15 shows the error curves and three specific coarse grids.

The fluid velocities are largest in Regions 1 and 3, and increasing the resolution in these blocks contributes most to decrease the saturation and water-cut errors (see Grids 1 and 2). As expected, we also observe (see Grid 3) that refining the grid in Regions 4 and 5, and half of Region 2, does not contribute to any error reduction because the water does not enter these regions. Qualitatively similar results were obtained by using the time-of-flight indicator (but here the block containing the second producer appears earlier in the refinement sequence).

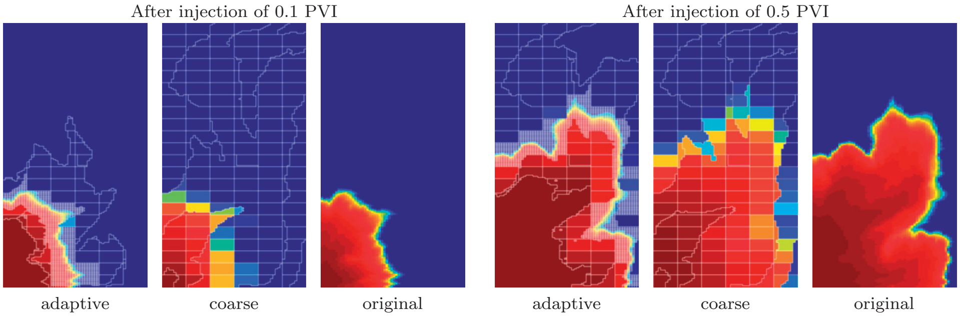


FIGURE 16. Examples of locally adapted grids for Layer 22 of the SPE10 model.

Another example of a regional and successive coarsening/refinement cycle was recently presented by Hauge et al. [19]. For scenarios with strong displacement fronts, the majority of the errors introduced by the coarsening will appear near the saturation fronts. High grid resolution is therefore needed in these regions. In the unswept area ahead of the front the solution is constant, and behind the displacement front the solution is slowly varying, and in both these regions the saturation can be represented on a coarser grid. In other words, one uses a refined region that follows the evolving displacement front. Creating such a dynamic partition is relatively simple as long as it can be represented using a partition vector. In [19], the extent of the refinement region was estimated by evolving the saturation on a coarse grid. Alternatively, the refined region can be defined inside a band of values in the time-of-flight for simple displacements—quarter five-spots, line drives, etc—or by combining time-of-flight with *a prior* regional indicators for more complex scenarios.

Example 7 (Adaptive grids)**.** *We generate a coarse grid based on the time-of-flight indicator intersected with a uniform topological partitioning as explained in Section 5.1. Local refinement, down to the resolution of the original grid, is added in regions near the displacement front (basically identified by checking where saturation changes significantly from one step to the next, as explained in [19]). Cells that have time-of-flight values a factor $C > 1 + \epsilon$ larger than those near the displacement front will likely belong to the unswept zone and are therefore merged into a single coarse block. Moreover, a band of blocks with intermediate resolution are kept ahead of the displacement front (measured in time-of-flight) to localize the search for regions that need to be refined in the next step and as a precaution when multiple time steps are computed without updating the grid. Figure 16 shows examples the adaptive grids for a layer from the SPE10 model. For comparison, we also show the saturation profiles on the original and on the coarse grid.*

In the final example, we consider supervised use of the coarsening primitives. That is, based on an initial coarse grid, the user utilizes his/her expert knowledge to pick the correct refinement strategy in different parts of the grid to create a composite, medium coarse grid.

Example 8 (Supervised coarsening)**.** *A large number of experiments run by the authors show that the original NUC algorithm is best when applied to strongly channelized media. On smoother heterogeneity, the algorithm tends to overestimate the importance of flow adaption, and more accurate results can be obtained using a topologically refined grid; see [19] for a more detailed discussion. Developing robust criteria for automatically choosing the right coarsening*

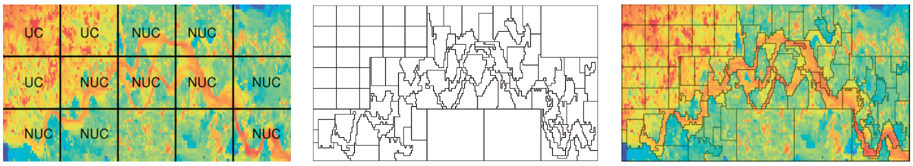


FIGURE 17. Supervised hierarchical coarsening applied to Layer 37 of SPE10. The left plot shows the initial partition and the refinement choices plotted on top of the velocity indicator. The middle and the rightmost plot show the resulting grid.

approach is difficult, and in this example we therefore consider a supervised approach. First, a very coarse topological partition is created. Then, in each block the user makes one out of three choices: (i) to keep the block as it is if the total flow through it is small, (ii) to perform a further topological subpartition if there is high flow throughout most of the block, or (iii) to perform a flow-based subpartition (using the NUC algorithm from Example 1) if the block contains high-flow channels on a low-flow background. Figure 17 shows an illustration of the supervised gridding on Layer 37 from SPE10.

8. Numerical Experiments and Discussion

In this section we will make more quantitative comparisons between several of the specific algorithmic realizations discussed in the previous sections. To keep the presentation as simple as possible, most of the examples use petrophysical data from Model 2 of SPE10 [9]. This data set has two formations: The Tarbert formation is a shallow-marine deposit with relatively smooth permeability (and porosity) variations. The Upper Ness formation is fluvial and has a set of distinct, but strongly intertwined, high-permeable channels on a low-permeable background.

To simulate transport on the coarse grids, we will in the following use two different schemes that both assume that the saturations are constant on the coarse grid, i.e., $S_\ell = |B_\ell|^{-1}\int_{B_\ell} S(x)\,dx$. The first scheme is derived by summing a standard upwind discretization for all cells in a block

$$\begin{aligned}(13)\quad S_\ell^{n+1} = S_\ell^n &+ \frac{\Delta t}{\int_{B_\ell}\phi\,dx}\int_{B_\ell} h_S(S^{n+1})\,dx \\ &- \frac{\Delta t}{\int_{B_\ell}\phi\,dx}\Big[f(S_\ell^{n+1})\sum_{\gamma_{ij}\subset\partial B_\ell}\max(v_{ij},0) - \sum_{k\neq\ell}\Big(f(S_k^{n+1})\sum_{\gamma_{ij}\subset\Gamma_{k\ell}}\min(v_{ij},0)\Big)\Big].\end{aligned}$$

Unlike the underlying fine-scale discretization, this scheme uses a bi-directional approximation for all faces across which there are both positive and negative fine-scale fluxes. In the second scheme, we first sum the fine-scale fluxes to obtain net fluxes across the coarse faces and then use a standard single-point upwind discretization to derive,

$$\begin{aligned}(14)\quad S_\ell^{n+1} = S_\ell^n &+ \frac{\Delta t}{\int_{B_\ell}\phi\,dx}\int_{B_\ell} h_S(S^{n+1})\,dx \\ &- \frac{\Delta t}{\int_{B_\ell}\phi\,dx}\sum_{k\neq\ell}\max\Big(f(S_\ell^{n+1})\sum_{\gamma_{ij}\subset\Gamma_{k\ell}} v_{ij},\ \ -f(S_k^{n+1})\sum_{\gamma_{ij}\subset\Gamma_{k\ell}} v_{ij}\Big).\end{aligned}$$

Table 1. Comparison of two different indicator functions in the original NUC algorithm on all layers in each of the two formations of SPE10. (Coarsening parameters: $N_L = 6$ and $N_U = 32$ for velocity and $N_L = 6$ and $N_U = 28$ for vorticity).

	Tarbert		Upper Ness	
	$\log(\vert\vec{\omega}\vert)$	$\log(\vert\vec{v}\vert)$	$\log(\vert\vec{\omega}\vert)$	$\log(\vert\vec{v}\vert)$
$E_s(\mathcal{P}S_c, S_f)$	0.1529	0.1538	0.1622	0.1506
$E_s(S_c, \mathcal{R}S_f)$	0.1264	0.1235	0.1100	0.1026
$E_s(\mathcal{P}\mathcal{R}S_f, S_f)$	0.0758	0.0811	0.1239	0.1010
$E_w(w_c, w_f)$	0.0585	0.0467	0.0559	0.0192
blocks: span	514–591	368–431	508–582	435–507
blocks: mean	546	405	541	467
faces: mean	2671	1995	2649	2291

To study the accuracy of these schemes, we need some further notation. Let S_f and S_c denote the saturation field computed on the fine and coarse grids, respectively, and let w_f and w_c denote the respective water-cuts. Moreover, we define $\mathcal{R}$ to be the restriction from the fine to the coarse grid and $\mathcal{P}$ to be the prolongation from the coarse to the fine grid. Finally, we define two different error norms

$$E_s(q,p) = \frac{1}{T}\int_0^T \frac{\|[q(\cdot,t) - p(\cdot,t)]\phi(\cdot)\|_1}{\|p(\cdot,t)\phi(\cdot)\|_1}\,dt, \qquad E_w(w_1,w_2) = \frac{\|w_1(\cdot) - w_2(\cdot)\|_2}{\|w_2(\cdot)\|_2}.$$

A large number of numerical experiments show that using net fluxes rather than bi-directional fluxes typically reduces saturation and water-cut errors by 20–40 % because the former scheme has lower numerical dissipation. For the following examples, simulations were performed with both schemes, but results are only reported for the upwind scheme (14), unless stated otherwise.

In the first example, we revisit the original nonuniform coarsening algorithm [1] as presented in Example 1 and compare the accuracy for grids obtained with the velocity indicator $\log(|\vec{v}|)$ and the vorticity indicator $\log(|\vec{\omega}|)$.

Example 9 (Vorticity versus velocity)**.** *Consider a quarter five-spot simulation on a* 220×60 *grid with petrophysical data taken from individual layers from Model 2 of SPE10. Table 1 shows saturation and water-cut errors averaged over all layers within each of the two geological formations simulated using the upwind scheme* (14)*. (Plots of the two grid types are shown in Figure 10 for Layer 37 from the Upper Ness formation). For the smooth Tarbert layers, the improved smoothness of the vorticity indicator has a slightly positive effect on the accuracy. On the fluvial layers of the Upper Ness formation, however, the vorticity indicator exaggerates the extent of the high-flow channels and hence produces grids having higher errors, in particular for the water-cut.*

Hauge et al. [19] recently proposed an improved version of the NUC algorithm, in which the initial partitioning consists of the intersection of an *a priori* uniform partitioning and a segmentation of the cells according to a flow-based indicator with the number of bins proportional to the span of the indicator function. The merging and refinement steps are carried out as in the original NUC algorithm, except that we use 9-neighbours when growing new blocks. We have previously seen this algorithm in Example 3.

Table 2. Comparison of saturation and water-cut errors up to dimensionless time 1.0 PVI for quarter five-spot simulations on the individual layers of the SPE10 model using the scheme (14) based on net fluxes. The upper part of the table shows errors, and the lower part gives statistics for the grids.

	Tarbert formation			Upper Ness formation		
	$-\log(\tau\tau_r)$	$\log(\lvert\vec{v}\rvert)$	Cartesian	$-\log(\tau\tau_r)$	$\log(\lvert\vec{v}\rvert)$	Cartesian
$E_s(\mathcal{P}S_c, S_f)$	0.1398	0.1607	0.1619	0.1556	0.1795	0.2191
$E_s(S_c, \mathcal{R}S_f)$	0.1030	0.1237	0.1302	0.0964	0.1135	0.1486
$E_s(\mathcal{P}\mathcal{R}S_f, S_f)$	0.0887	0.0941	0.0911	0.1291	0.1371	0.1772
$E_w(w_c, w_f)$	0.0369	0.0473	0.0647	0.0345	0.0237	0.0844
blocks: span	219–261	217–261	264	213–252	205–241	264
blocks: mean	238	236	264	229	222	264
faces: mean	1054	1069	1090	1068	1070	1090

In [19], it was shown that the hybrid grids generated by the new algorithm have more regularity compared with the original NUC-algorithm and lead to improved accuracy of the transport simulations—using either of the schemes (13) or (14)—as well as simpler structures in the discretized systems. In the next example, we will compare the accuracy of this algorithm using the time-of-flight and the velocity indicator; that is, we compare the accuracy of the two grid types shown in Figures 9.

Example 10 (Time-of-flight versus velocity)**.** *In this example we repeat the simulation set-up of Example 9 and run transport simulation for all layers of the SPE10 model. In addition, we include simulations on a uniform* 22×12 *Cartesian grid as a reference. Table 2 summarizes errors and some statistics about the coarsened grids.*

Our first observation is that the flow-based grids have improved accuracy compared with the Cartesian grid. Moreover, we see that the time-of-flight indicator gives significantly lower errors than the velocity indicator, except for the water-cut on the Upper Ness formation. Refering back to our motivation for the two indicators in Sections 5.1 and 5.2, it seems reasonable that time-of-flight will be a better indicator of saturation transport and hence give grids with improved accuracy.

Secondly, the results show that the regularity of both the flow-based grids are comparable to that of the Cartesian coarse grid in the sense that the numbers of faces, representing connections in the grids, are approximately the same. Regularity of the grids also affects the matrix structure of the discretized system and may be exploited in efficient linear solvers. As seen in Section 5.1, the transport equation has a certain causality in the sense that it can be decomposed into a family of one-dimensional transport equations along streamlines. This is reflected as follows in the discretized equations: If the flow field on the original grid is computed by a (strictly) monotone scheme, the fluxes will form a directed acyclic graph. If the transport equation is discretized by an upwind scheme, the corresponding nonlinear system of equations can be reordered into a lower triangular form by performing a topological sort to flatten the directed flux graph. This means that the cell values can be updated using a highly efficient, sequential, nonlinear Gauss–Seidel method, see [27]. On the coarse grid, the flux graph used in (13) *and* (14) *will generally not be acyclic because of bi-directional fluxes over coarse faces and cyclic dependence among coarse blocks. The discrete system of transport*

Table 3. Different measures of the matrix structure on two layers of SPE10. For each grid, the first number refers to the bi-directional scheme (13), while the second number refers to the scheme using net fluxes (14).

	Layer 22						Layer 62					
	$-\log(\tau\tau_r)$		$\log(\|\vec{v}\|)$		Cartesian		$-\log(\tau\tau_r)$		$\log(\|\vec{v}\|)$		Cartesian	
grid blocks	189	189	161	161	220	220	166	166	171	171	220	220
single entries	6	163	7	135	74	220	0	91	0	43	8	200
block entries	11	4	13	5	46	0	1	11	1	10	14	4
largest block	74	15	58	13	7	—	166	18	171	33	60	8
off-diagonals	583	420	507	355	508	408	663	406	768	443	598	408

equations can still be reordered, but now into a block-triangular form, where each matrix block corresponds to coarse grid-blocks having circular dependence.

Table 3 reports properties of the reordered matrices for the schemes (13) *and* (14) *for one layer from each formation, Layers 22 and 62, respectively, now compared with a uniform* 22×10 *Cartesian grid. For each system, we report the number of grid blocks, the number of single entries on the diagonal (corresponding to grid blocks with no circular dependence), the number of block entries on the diagonal (corresponding to grid blocks with circular dependence), the largest block (number of grid blocks in the largest cycle), and the number of off-diagonal elements. Several observations can be made. First of all, flow-based grids have more cyclic dependencies that the Cartesian grids. Second, the number of cyclic dependencies is significantly reduced using net fluxes rather than bi-directional fluxes. Third, the time-of-flight grids give less cyclic dependencies than the velocity grids, indicating that they will be more computationally efficient.*

The large degree of regularity of the grids in the above example comes from the regularity of the *a priori* topological partitions. In all examples considered so far, these partitions have been uniform Cartesian. To not delude the reader to assume that this type of algorithm is restricted to structured grids where one can find a simple topological partitioning, we include an example of the same type of time-of-flight coarsening of a 2.5D PEBI grid. Here, the initial segmentation into bins is intersected with a partition generated using the software package METIS [21]. The motivation for intersecting with a METIS grid is to avoid that the effects of the refinement step dominate in regions where the indicator does not separate the grid blocks well. Hence, using a METIS partitioning will cause the final coarse grids to inherit, to some degree, the regularity of this partitioning.

Example 11 (A 2.5D PEBI grid)**.** *We consider an extruded PEBI model consisting of 11 864 cells. Petrophysical parameters have been sampled from Layers 50–60 of SPE10. One injection and one production well are placed to create a diagonal displacement. Figure 18 shows the saturation distribution after one pore-volume of water has been injected, simulated on the original fine grid (left), on a coarse time-of-flight grid consisting of 127 blocks (middle), and on the coarse grid, partitioned topologically by METIS into 175 blocks (right). We observe that the METIS grid has too large blocks in the flow channel to capture the flow well, while the time-of-flight grid to a large degree matches the flow pattern from the original fine grid.*

Because the flow-based indicator functions discussed in Section 5 only measure magnitude of flow and do not distinguish between opposite flow direction, they will in certain cases not

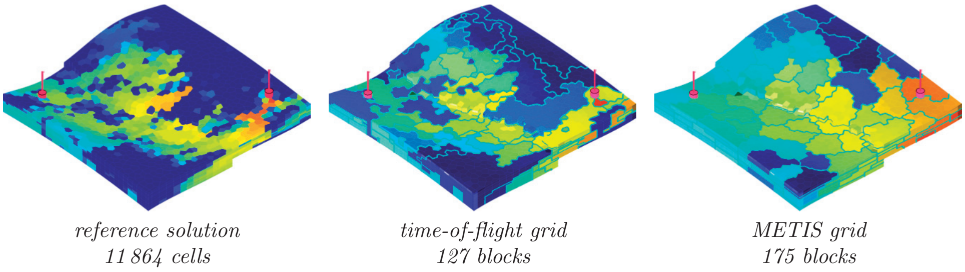


FIGURE 18. Saturation profiles after simulation of 1 PVI on the fine-scale reference grid (left), time-of-flight-based grid (middle), and METIS grid (right). Both coarse grid solutions have been solved using net fluxes.

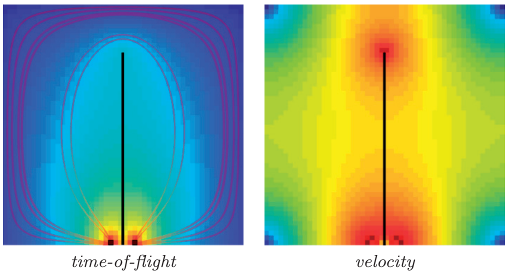


FIGURE 19. Indicator fields for a case with flow around a sealing fault.

provide sufficient information to partition grid cells correctly. To illustrate this, we consider a highly idealized example.

Example 12 (Pathological barrier case)**.** *Consider a* 50×50 *Cartesian model with a source and a sink placed on opposite sides of a sealing fault, so that the flow is forced to go around the sealing barrier, creating a symmetric flow pattern. Figure 19 shows a plot of the time-of-flight and the velocity indicators. We observe that both fields are symmetric and not distinguishing between direction of flow. Hence, flow out from the source is associated with the flow into the sink, as the magnitude of the flow is the same.*

Two different coarse grids were generated by intersecting the segmented time-of-flight indicator values with two Cartesian partitions with 5×5 *and* 6×5 *cells, respectively. No constraints were imposed along the barrier. The grids and the corresponding water-cut curves during the injection of one pore volume are shown in the upper row of Figure 20. Both grids fail to capture the flow, as seen in the water-cut curves. The grid based on the* 5×5 *initial partition, in particular, contains a coarse block that completely encapsulates the two blocks that contain the source and sink, which leads to an almost instant water breakthrough.*

Including a constraint for all faces along the barrier[3] *and repeating the simulation gives the results shown in the bottom row of Figure 20. Here, the restriction of admissible neighbours*

[3] To avoid the questions of hanging edges, as explained in Section 4.3, both partitions have been chosen so that the end of the constraint coincides with a horizontal line in the coarse grid.

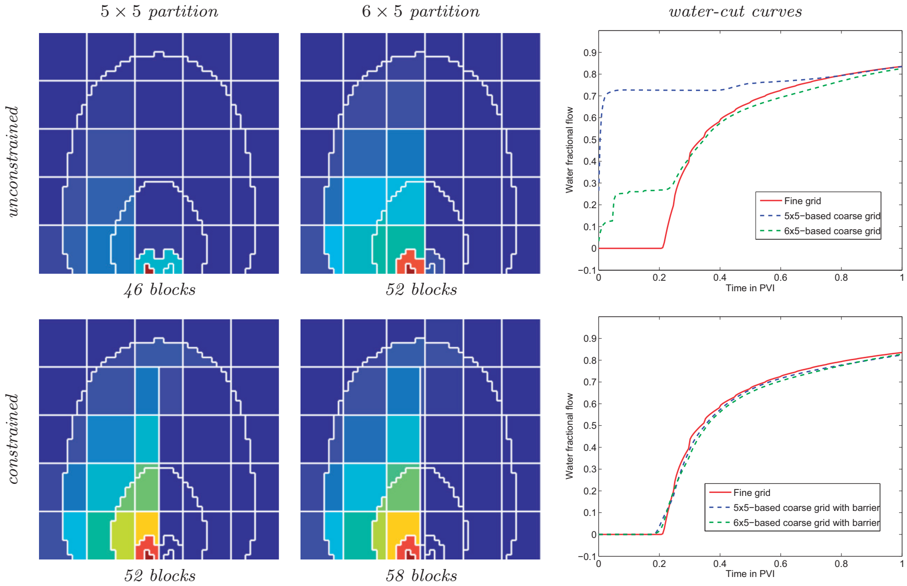


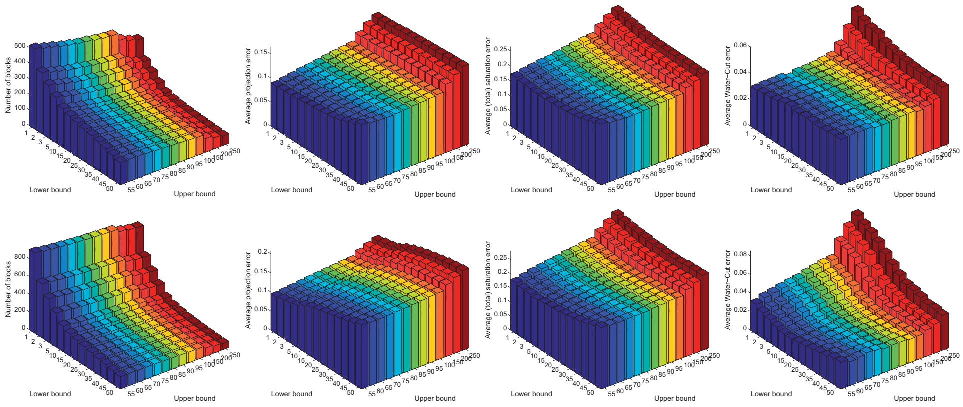
Figure 20. Coarse grids generated using the time-of-flight indicator in combination with a Cartesian partition outlined on top of the saturation distribution at 0.1 PVI. The rightmost plots show the water-cut curves for a simulation up to 1.0 PVI.

along the barrier enables the algorithm to distinguish the flow out of the sink and into the source, producing grids that accurately reproduce the water-cut curve from the original grid.

The example illustrates two interesting features of our algorithmic framework. First of all, it shows the importance of making intelligent choices when setting up a specific algorithm. The coarsening algorithms discussed herein are not fully automated, but should be used in a supervised manner, so that the user can exploit his/her knowledge and the flexibility of the framework to develop fit-for-purpose coarsening. For the example above, the coarsening fails quite spectacularly without inclusion of face constraints that reduce the set of admissible neighbours along the barrier, because the indicators only measure flow magnitude and do not distinguish flow directions. Alternatively, the same effect, and good coarse grids, could be obtained if the cell connections were postprocessed to exclude all faces that have zero or very small fluxes. However, this approach may potentially also have undesired side-effects in certain cases.

Secondly, the example demonstrates a shortcoming of NUC-type algorithms. Although they are generally quite robust with respect to different parameter choices, as we will see in the next example, accidental circumstances may in some cases lead them to produce infeasible grids. Conversely, fortunate coincidences may in other cases lead to disproportionate improvements in accuracy for small changes in the coarsening parameters.

In the next example we investigate the robustness of the flow-based coarsening with respect to different parameter choices. To reduce the number of parameters involved, we consider the



number of blocks

projection error $E_s(\mathcal{PR}S_f, S_f)$

total saturation error $E_s(\mathcal{P}S_c, S_f)$

water-cut error, $E_w(w_c, w_f)$

FIGURE 21. The original NUC algorithm applied with varying parameters on all layers of the Tarbert formation (top) and all layers (except Layers 74–76) of the Upper Ness formation (bottom). In each plot, the histogram values are averaged over all layers in the formation.

original NUC algorithm Aarnes et al. [1], which has only two parameters: the constants N_L and N_U in (4) and (6), respectively.

Example 13 (Robustness)**.** *We continue with the computational setup from Examples 9 and 10: quarter five-spot well pattern on a 220×60 grid with petrophysical data taken from individual layers from Model 2 of SPE10. Figure 21 shows the number of blocks, projection error, total saturation error, and water-cut error for all layers in the Tarbert formation and all layers in the Upper Ness formation (except for Layers 74–76, which were excluded because of long simulation times caused by small porosity values). For both formations, the number of coarse blocks is mainly determined by N_L, in particular for the fluvial layers. For the smooth permeability in the Tarbert formation, the projection error is dominated by the size of the coarse blocks. Indeed, the histogram shows that the projection error increases with increasing values of N_U, but is almost constant with respect to N_L. Interestingly, the lowest saturation errors are observed for* intermediate *values of N_L and small values of N_U. Moreover, for large values of N_U, the water-cut errors decrease with increasing values of N_L. Conversely, for fixed N_L, the water-cut errors increase with N_U, because larger upper bounds allow larger blocks, which in turn causes smearing of the saturation fronts and typically leads to earlier water breakthrough.*

For the channelized permeability in the Upper Ness formation, the saturations will follow narrow flow channels and the projection error will be dominated by how well the grid resolves these channels, for which N_L is the decisive parameter. This is confirmed by the histogram, in which the largest increase in projection error is observed in the direction of increasing N_L values. The water-cut error, on the other hand, decreases with increasing N_L values. A possible explanation is that the smallest blocks occur in and near the high-permeable channels and as the sizes of these blocks increase, the blocks adapt to the high-flow paths and thereby improve the grids' ability to correctly evolve saturations.

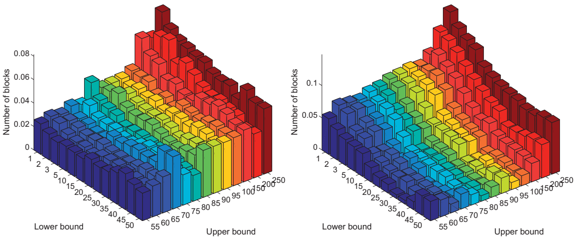


Figure 22. Water-cut errors for the original NUC algorithm applied with varying parameters to Layer 28 (left) and Layer 48 (right).

Finally, we notice the existence of relatively flat regions in the middle of the parameter domain, where reasonable accuracy is achieved in a robust manner. This is not an averaging effect—similar regions are observed when considering a single layer. However, for a single layer, small changes in parameters can lead to disproportionate changes in accuracy as shown in Figure 22 for the water-cut errors in Layers 28 and 48.

9. Concluding remarks

In this paper, we have discussed how to generate good flow-based grids to be used in multi-fidelity transport solvers that can be combined with multiscale flow solvers to give a complete framework for fast and accurate multiscale flow modelling. Our point has not been to come up with a single algorithm that creates *the* best grid—in our experience, such a grid would vary from case to case, depending strongly on the geology and the flow processes to be modelled, and would also generally be different for flow and transport solvers. Instead, we have presented a modular algorithmic framework that consists of a set of components that can be combined in various ways to fit different flow scenarios, requirements with respect to accuracy and runtime, as well as other topological and geological constraints. A particular feature is our use of amalgamation of cells from the original grid to form coarse blocks, which gives the user high flexibility in adapting the coarse grid to details in the flow field, as well as to soft and hard geological constraints (layers, pinchouts, fault faces, etc).

Our framework is very general and leaves the user with a large number of choices. The paper has mainly focused on presenting concepts and outlining possibilities for flow adaption; more extensive and rigorous numerical studies are left for later. Likewise, we have not discussed the additional postprocessing that will be needed in practical applications to smoothen unconstrained faces, remove blocks that are confined within other blocks, etc. Moreover, there is a need to develop guidelines for use in supervised approaches, criteria for use in automated approaches, and ways of setting soft and hard constraints to reflect the uncertainty in the underlying geological modelling. These are all topics for further research.

References

[1] J. E. Aarnes, V. L. Hauge, and Y. Efendiev. Coarsening of three-dimensional structured and unstructured grids for subsurface flow. *Adv. Water Resour.*, 30(11):2177–2193, 2007. doi: 10.1016/j.advwatres.2007.04.007.

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

[3] M. A. Ashjari, B. Firoozabadi, H. Mahani, and D. Khoozan. Vorticity-based coarse grid generation for upscaling two-phase displacements in porous media. *J. Petrol. Sci. Eng.*, 59(3-4):271 – 288, 2007. ISSN 0920-4105. doi: 10.1016/j.petrol.2007.04.006.

[4] M. A. Ashjari, B. Firoozabadi, and H. Mahani. Using vorticity as an indicator for the generation of optimal coarse grid distribution. *Transp. Porous Media*, 75(2):167 – 201, 2008. doi: 10.1007/s11242-008-9217-9.

[5] J. Bear. *Dynamics of Fluids in Porous Media*. Dover, 1988. ISBN 0-486-45355-3.

[6] L. V. Branets, S. S. Ghai, S. L. Lyons, and X.-H. Wu. Review article: Challenges and technologies in reservoir modeling. *Commun. Comput. Phys.*, 6(1):1–23, 2009.

[7] A. Castellini, M. G. Edwards, and L. J. Durlofsky. Flow based modules for grid generation in two and three dimensions. In *Proceedings of 7th European Conference on the Mathematics of Oil Recovery, Baveno, Italy, Sept. 5-8*, 2000.

[8] M. A. Christie. Upscaling for reservoir simulation. *J. Petrol. Technol.*, 48(11):1004–1010, 1996. doi: 10.2118/37324-MS.

[9] 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. Url: `http://www.spe.org/csp/`.

[10] O. A. Cirpka, E. O. Frind, and R. Helmig. Streamline-oriented grid generation for transport modelling in two-dimensional domains including wells. *Adv. Water Resour.*, 22(7):697 – 710, 1999. ISSN 0309-1708. doi: 10.1016/S0309-1708(98)00050-5.

[11] L. J. Durlofsky. Upscaling of geocellular models for reservoir flow simulation: A review of recent progress, 2003. Presented at 7th International Forum on Reservoir Simulation Bühl/Baden-Baden, Germany, June 23 – 27, 2003.

[12] L. J. Durlofsky. Upscaling and gridding of fine scale geological models for flow simulation, 2005. Presented at 8th International Forum on Reservoir Simulation Iles Borromees, Stresa, Italy, June 20 –24, 2005.

[13] L. J. Durlofsky, R. C. Jones, and W. J. Milliken. A nonuniform coarsening approach for the scale-up of displacement processes in heterogeneous porous media. *Adv. Water Resour.*, 20(5-6):335 – 347, 1997. ISSN 0309-1708. doi: 10.1016/S0309-1708(96)00053-X.

[14] M. Evazi and H. Mahani. Unstructured-coarse-grid generation using background-grid approach. *SPE J.*, 15(2):326–340, 2010. SPE-120170-PA.

[15] M. Evazi and H. Mahani. Generation of Voronoi grid based on vorticity for coarse-scale modeling of flow in heterogeneous formations. *Transp. Porous Media*, 2010. doi: 10.1007/s11242-009-9458-2.

[16] C. L. Farmer. Upscaling: a review. *Int. J. Numer. Meth. Fl.*, 40(1–2):63 –78, 2002. doi: 10.1002/fld.267.

[17] M. H. Garcia, A. G. Journel, and K. Aziz. Automatic grid generation for modeling reservoir heterogeneities. *SPE Reservoir Eng.*, 7(6):278–284, 1992.

[18] V. L. Hauge and J. E. Aarnes. Modeling of fractured two-phase flow in fractured porous media on unstructured non-uniform coarse grids. *Transp. Porous Media*, 77(3):373–398, 2009. doi: 10.1007/s11242-008-9284-y.

[19] V. L. Hauge, K.-A. Lie, and J. R. Natvig. Flow-based grid coarsening for transport simulations. In *Proceedings of ECMOR XII–12th European Conference on the Mathematics of Oil Recovery*, Oxford, UK, 6–9 September 2010. EAGE.

[20] C. He and L. J. Durlofsky. Structured flow-based gridding and upscaling for modeling subsurface flow. *Adv. Water Resour.*, 29(12):1876 – 1892, 2006. ISSN 0309-1708. doi: 10.1016/j.advwatres.2005.12.012.

[21] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. In *International Conference on Parallel Processing*, pages 113–122, 1995.

[22] M. J. King. Recent advances in upgridding. *Oil & Gas Science and Technology - Rev. IFP*, 62(2):195 – 205, 2007. doi: 10.2516/ogst:2007017.

[23] M. J. King, K. S. Burn, P. W. V. Muralidharan, F. Alvarado, X. Ma, and A. Datta-Gupta. Optimal coarsening of 3D reservoir models for flow simulation. *SPE Reserv. Eval. Eng.*, 9(4):317 – 334, 2006. doi: 10.2118/95759-PA.

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

[25] H. Mahani and A. H. Muggeridge. Improved coarse grid generation using vorticity. In *14th Europec Biennial Conference, Madrid, Spain, 13-16 June*, number SPE 94319, 2005.

[26] T. Manzocchi et al. Sensitivity of the impact of geological uncertainty on production from faulted and unfaulted shallow-marine oil reservoirs: objectives and methods. *Petrol. Geosci.*, 14(1):3–15, 2008.

[27] J. R. Natvig and K.-A. Lie. Fast computation of multiphase flow in porous media by implicit discontinuous Galerkin schemes with optimal ordering of elements. *J. Comput. Phys.*, 227(24):10108–10124, 2008. ISSN 0021-9991. doi: 10.1016/j.jcp.2008.08.024. URL `http://dx.doi.org/10.1016/j.jcp.2008.08.024`.

[28] J. R. Natvig, K.-A. Lie, B. Eikemo, and I. Berre. An efficient discontinuous Galerkin method for advective transport in porous media. *Adv. Water Resour*, 30(12):2424–2438, 2007. doi: 10.1016/j.advwatres.2007.05.015.

[29] M. Prevost, F. Lepage, L. J. Durlofsky, and J.-L. Mallet. Unstructured 3D gridding and upscaling for coarse modelling of geometrically complex reservoirs. *Petrol. Geosci.*, 11 (4):339–345, 2005. doi: 10.1144/1354-079304-657.

[30] S. Verma and K. Aziz. A control volume scheme for flexible grids in reservoir simulation. In *SPE Reservoir Simulation Symposium, Dallas, Texas, 8-11 June*, number SPE37999, 1997. doi: 10.2118/37999-MS.

[31] X. H. Wen, L. J. Durlofsky, and M. G. Edwards. Upscaling of channel systems in two dimensions using flow-based grids. *Transp. Porous Media*, 51(3):343 – 366, 2003. doi: 10.1023/A:1022318926559.

[32] H. Zhou, S. H. Lee, and H. A. Tchelepi. Multiscale finite volume formulation for the saturation equations. In *SPE Reservoir Simulation Symposium, 2-4 February, The Woodlands, Texas*, number SPE119183, 2009. doi: 10.2118/119183-MS.

Modeling of Two-Phase Flow in Fractured Porous Media on Unstructured Non-Uniformly Coarsened Grids

Vera Louise Hauge and Jørg Espen Aarnes

Transport in Porous Media, Volume 77, Number 3,
April 2009, pp. 373–398

Paper V

Is not included due to copyright

Adjoint Multiscale Mixed Finite Elements

Stein Krogstad, Vera Louise Hauge, and Astrid Fossum Gulbransen

Accepted for publication in SPE journal. SPE-119112-PA

Paper VI

Is not included due to copyright