Multiscale Methods for Subsurface Flow

Jørg E. Aarnes, Knut–Andreas Lie, Vegard Kippe, and Stein Krogstad

SINTEF ICT, Dept. of Applied Mathematics, Oslo Norway

Modelling of flow processes in the subsurface is important for many applications. In fact, subsurface flow phenomena cover some of the most important technological challenges of our time. To illustrate, we quote the UN's Human Development Report 2006:

> "There is a growing recognition that the world faces a water crisis that, left unchecked, will derail the progress towards the Millennium Development Goals and hold back human development. Some 1.4 billion people live in river basins in which water use exceeds recharge rates. The symptoms of overuse are disturbingly clear: rivers are drying up, groundwater tables are falling and water-based ecosystems are being rapidly degraded. Put bluntly, the world is running down one of its most precious natural resources and running up an unsustainable ecological debt that will be inherited by future generations."

The road toward sustainable use and management of the earth's groundwater reserves necessarily involves modelling of groundwater hydrological systems. In particular, modelling is used to acquire general knowledge of groundwater basins, quantify limits of sustainable use, and to monitor transport of pollutants in the subsurface.

A perhaps equally important problem is how to reduce emission of greenhouse gases, such as CO_2, into the atmosphere. Indeed, the recent report from the UN Intergovernmental Panel on Climate Change (see e.g., `www.ipcc.ch`) draws a frightening scenario of possible implications of human-induced emissions of greenhouse gases. Carbon sequestration in porous media has been suggested as a possible means. Schrag [46] claims that

> "Carbon sequestration (...) is an essential component of any serious plan to avoid catastrophic impacts of human-induced climate change. Scientific and economical challenges still exist, but none are serious enough to suggest that carbon capture and storage (in underground repositories) will not work at the scale required to offset trillions of tons of CO_2 emissions over the next century."

The primary concern related to storage of CO_2 in subsurface repositories is related to how fast the CO_2 will escape. Repositories do not need to store CO_2 forever, just long enough to allow the natural carbon cycle to reduce the atmospheric CO_2 to near pre-industrial level. Nevertheless, making a qualified estimate of the leakage rates from potential CO_2 storage facilities is a non-trivial task, and demands interdisciplinary research and software based on state-of-the art numerical methods for modelling subsurface flow.

These examples illustrate that the demand for software modelling subsurface flow will not diminish with the decline of the oil and gas era. In fact, the need for tools that help us understand flow processes in the subsurface is probably greater than ever, and increasing. Nevertheless, more than 50 years of prior research in this area has led to some degree of agreement in terms of how subsurface flow processes can be modelled adequately with numerical simulation technology. Because most of the prior research in this area targets reservoir simulation, i.e., modelling flow in oil and gas reservoirs, we will focus on this application in the remainder of this paper. However, the general modelling framework, and the numerical methods that are discussed, apply also to modelling flow in groundwater reservoirs and CO_2 storage facilities.

To describe the subsurface flow processes mathematically, two types of models are needed. First, one needs a mathematical model that describes how fluids flow in a porous medium. These models are typically given as a set of partial differential equations describing the mass-conservation of fluid phases. In addition, one needs a geological model that describes the given porous rock formation (the reservoir). The geological model is used as input to the flow model, and together they make up the reservoir simulation model.

Unfortunately, geological models are generally too large for flow simulation, meaning that the number of grid cells exceed the capabilities of current flow simulators (usually by orders of magnitude) due to limitations in memory and processing power. The traditional, and still default, way to build a reservoir simulation model therefore starts by converting the initial geomodel (a conceptual model of the reservoir rock with a plausible distribution of geological parameters) to a model with a resolution that is suitable for simulation. This process is called upscaling. Upscaling methods aim to preserve the small-scale effects in the large-scale computations (as well as possible), but because small-scale features often have a profound impact on flow occurring on much larger scales, devising robust upscaling techniques is a non-trivial task.

Multiscale methods are a new and promising alternative to traditional upscaling. Whereas upscaling techniques are used to derive coarse-scale equations with a reduced set of parameters, multiscale methods attempt to incorporate fine-scale information directly into the coarse-scale equations. Multiscale methods are rapidly growing in popularity, and have started to gain recognition as a viable alternative to upscaling, also by industry. The primary purpose of this paper is to provide an easily accessible introduction to multiscale methods for subsurface flow, and to clarify how these methods relate to some standard, but widely used, upscaling methods.

We start by giving a crash course in reservoir simulation. Next, we describe briefly some basic discretisation techniques for computing reservoir pressure and velocity fields. We then provide a brief introduction to upscaling, and present some of the most commonly used methods for upscaling the pressure equation. The final part of the paper is devoted to multiscale methods for computing pressure and velocity fields for subsurface flow applications.

1 Introduction to Reservoir Simulation

Reservoir simulation is the means by which we use a numerical model of the petrophysical characteristics of a hydrocarbon reservoir to analyse and predict fluid behaviour in the reservoir over time. For nearly half a century, reservoir simulation has been an integrated part of oil-reservoir management. Today, simulations are used to estimate production characteristics, calibrate reservoir parameters, visualise reservoir flow patterns, etc. The main purpose is to provide an information database that can help the oil companies to position and manage wells and well trajectories in order to maximize the oil and gas recovery. Unfortunately, obtaining an accurate prediction of reservoir flow scenarios is a difficult task. One of the reasons is that we can never get a complete and accurate characterisation of the rock parameters that influence the flow pattern. And even if we did, we would not be able to run simulations that exploit all available information, since this would require a tremendous amount of computer resources that exceed by far the capabilities of modern multi-processor computers. On the other hand, we do not need, nor do we seek a simultaneous description of the flow scenario on all scales down to the pore scale. For reservoir management it is usually sufficient to describe the general trends in the reservoir flow pattern.

In this section we attempt only to briefly summarise some aspects of the art of modelling porous media flow and motivate a more detailed study of some of the related topics. More details can be found in one of the general textbooks describing modelling of flow in porous media, e.g., [10, 21, 26, 30, 41, 43, 23].

1.1 The Reservoir Description

Natural petroleum reservoirs typically consist of a subsurface body of sedimentary rock having sufficient porosity and permeability to store and transmit fluids. Sedimentary rocks are formed through deposition of sediments and typically have a layered structure with different mixtures of rock types. In its simplest form, a sedimentary rock consists of a stack of sedimentary beds that extend in the lateral direction. Due to differences in deposition and compaction, the thickness and inclination of each bed will vary in the lateral directions. In fact, during the deposition, parts of the beds may have been weathered down or completely eroded away. In addition, the layered structure of the beds may have been disrupted due to geological activity, introducing

fractures and faults. Fractures are cracks or breakage in the rock, across which there has been no movement. Faults are fractures across which the layers in the rock have been displaced.

Oil and gas in the subsurface stem from layers of compressed organic material that was deposited millions of years ago, and, with time, eventually turned into water and different hydrocarbon components. Normally the lightest hydrocarbons (methane, ethane, etc.) escaped quickly, whilst the heavier oils moved slowly towards the surface, but at certain sites geological activity had created and bent layers of low-permeable (or non-permeable) rock, so that the migrating hydrocarbons were trapped. It is these quantities of trapped hydrocarbons that form today's oil and gas reservoirs.

Rock formations found in natural petroleum reservoirs are typically heterogeneous at all length scales, from the micrometre scale of pore channels between sand grains to the kilometre scale of the full reservoir. To obtain a geological description of these reservoirs, one builds models that attempt to reproduce the true geological heterogeneity in the reservoir rock. However, it is generally not possible to account for all pertinent scales that impact the flow. Instead one has to create models for studying phenomena occurring at a reduced span of scales. In reservoir engineering, the reservoir is modelled in terms of a three-dimensional grid, in which the layered structure of sedimentary beds (a small unit of rock distinguishable from adjacent rock units) in the reservoir is reflected in the geometry of the grid cells. The physical properties of the rock (porosity and permeability) are represented as constant values inside each grid cell. The size of a grid block in a typical geological grid-model is in the range 10–50 m in the horizontal direction and 0.1–1 m in the vertical direction. Thus, a geological model is clearly too coarse to resolve small-scale features such as the micro-structure of the pores.

Rock Parameters

The rock *porosity*, usually denoted by ϕ, is the void volume fraction of the medium; i.e., $0 \leq \phi < 1$. The porosity usually depends on the pressure; the rock is *compressible*, and the rock compressibility is defined by:

$$c_r = \frac{1}{\phi}\frac{d\phi}{dp},$$

where p is the reservoir pressure. For simplified models it is customary to neglect the rock compressibility. If compressibility cannot be neglected, it is common to use a linearisation so that:

$$\phi = \phi_0\big(1 + c_r(p - p_0)\big),$$

where p_0 is a specified reference pressure and $\phi_0 = \phi(p_0)$.

The (absolute) *permeability*, denoted by K, is a measure of the rock's ability to transmit a single fluid at certain conditions. Since the orientation
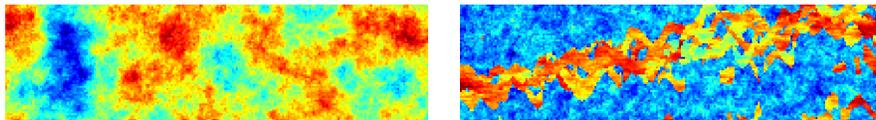
Fig. 1. Examples of two permeability fields: a shallow-marine Tarbert formation (left) and a fluvial Upper Ness formation (right).

and interconnection of the pores are essential for flow, the permeability is not necessarily proportional to the porosity, but K is normally strongly correlated to ϕ. Rock formations like sandstones tend to have many large or well-connected pores and therefore transmit fluids readily. They are therefore described as permeable. Other formations, like shales, may have smaller, fewer or less interconnected pores, e.g., due to a high content of clay. Such formations are described as impermeable. Although the SI-unit for permeability is m^2, it is commonly represented in Darcy (D), or milli-Darcy (mD). The precise definition of 1D ($\approx 0.987 \cdot 10^{-12}\, m^2$) involves transmission of a 1cp fluid (see below) through a homogeneous rock at a speed of 1cm/s due to a pressure gradient of 1atm/cm. Translated to reservoir conditions, 1D is a relatively high permeability.

In general, K is a tensor, which means that the permeability in the different directions depends on the permeability in the other directions. We say that the medium is isotropic (as opposed to anisotropic) if K can be represented as a scalar function, e.g., if the horizontal permeability is equal to the vertical permeability. Moreover, due to transitions between different rock types, the permeability may vary rapidly over several orders of magnitude, local variations in the range 1 mD to 10 D are not unusual in a typical field.

The heterogeneous structure of a porous rock formation is a result of the deposition history and will therefore vary strongly from one formation to another. In Figure 1 we show two permeability realisations sampled from two different formations in the Brent sequence from the North Sea. Both formations are characterised by large permeability variations, 8–12 orders of magnitude, but are qualitatively different. The Tarbert formation is the result of a shallow-marine deposition and has relatively smooth permeability variations. The Upper Ness formation is fluvial and has been deposited by rivers or running water, leading to a spaghetti of well-sorted high-permeable channels of long correlation length imposed on low-permeable background.

Grids

As described above, the rock parameters ϕ and K are usually given on a grid that also gives the geometrical description of the underlying rock formations. The most widespread way to model the geometry of rock layers is by so-called *corner-point grids*. A corner-point grid consists of a set of hexahedral cells that are aligned in a logical Cartesian fashion. One horizontal layer in the grid is
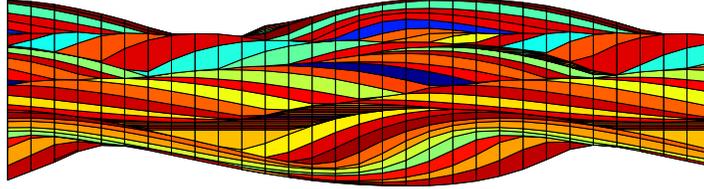
Fig. 2. Side view in the xz-plane of corner-point grid with vertical pillars modelling a stack of sedimentary beds (each layer indicated by a different colour).
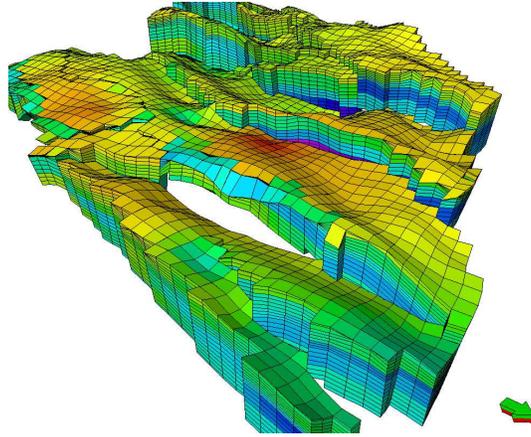
Fig. 3. Example of a geological grid model.

then assigned to each sedimentary bed to be modelled. 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. Figure 2 shows a side-view of such a corner-point grid. Notice the occurrence of degenerate cells with less than eight non-identical corners where the beds are partially eroded away. Some cells also disappear completely and hence introduce new connections between cells that are not neighbours in the underlying logical Cartesian grid.

The corner-point format easily allows for degeneracies in the cells and discontinuities (fractures/faults) across faces. Hence, using the corner-point format it is possible to construct very complex geological models that match the geologist's perception of the underlying rock formations, e.g., as seen in Figure 3. Due to their many appealing features, corner-point grids are now an industry standard and the format is supported in most commercial software for reservoir modelling and simulation.

1.2 Flow Parameters

The void in the porous medium is assumed to be filled with different phases. The volume fraction s occupied by each phase is the *saturation* of that phase. Thus,

$$\sum_{\text{all phases}} s_i = 1. \tag{1}$$

Here only three phases are considered; aqueous (a), liquid (l), and vapour (v). Each phase contains one or more *components*. A hydrocarbon component is a unique chemical species (methane, ethane, propane, etc). Since the number of hydrocarbon components can be quite large, it is common to group components into pseudo-components, e.g., water (w), oil (o), and gas (g).

Due to the varying conditions in a reservoir, the hydrocarbon composition of the different phases may change throughout a simulation. The mass fraction of component α in phase j is denoted by $m_{\alpha,j}$. In each of the phases, the mass fractions should add up to unity, so that for N different components, we have:

$$\sum_{\alpha=1}^{N} m_{\alpha,j} = 1.$$

The *density* ρ and *viscosity* μ of each phase are functions of *phase pressure* p_i $(i = a, l, v)$ and the component composition. That is, for vapour

$$\rho_v = \rho_v(p_v, \{m_{\alpha,v}\}), \qquad \mu_v = \mu_v(p_v, \{m_{\alpha,v}\}),$$

and similarly for the other phases. These dependencies are most important for the vapour phase, and are usually ignored for the aqueous phase.

The compressibility of the phase is defined as for rock compressibility:

$$c_i = \frac{1}{\rho_i} \frac{d\rho_i}{dp_i}, \quad i = a, l, v.$$

Compressibility effects are more important for gas than for fluids. In simplified models, the compressibility of the aqueous phase is usually neglected.

Due to interfacial tensions, the phase pressures are different, defining the *capillary pressure*,

$$p_{ij}^c = p_i - p_j,$$

for $i, j = a, l, v$. Although other dependencies are reported, it is usually assumed that the capillary pressure is a function of the saturations only.

Even though phases do not really mix, we assume that all phases may be present at the same location. The ability of one phase to move will then depend on the environment at the actual location. That is, the permeability experienced by one phase depends on the saturation of the other phases at that specific location, as well as the phases' interaction with the pore walls. Thus, we introduce a property called *relative permeability*, denoted by $k_{ri}, i = a, l, v,$

which describes how one phase flows in the presence of the two others. Thus, in general, and by the closure relation (1), we may assume that

$$k_{ri} = k_{ri}(s_a, s_v),$$

where subscript r stands for *relative* and i denotes one of the phases a, l, or v. Thus, the (effective) permeability experienced by phase i is $K_i = Kk_{ri}$. It is important to note that the relative permeabilities are nonlinear functions of the saturations, so that the sum of the relative permeabilities at a specific location (with a specific composition) is not necessarily equal to one. In general, relative permeabilities may depend on the pore-size distribution, the fluid viscosity, and the interfacial forces between the fluids. These features, which are carefully reviewed by Demond and Roberts [27], are usually ignored. Of greater importance to oil recovery is probably the temperature dependency [42], which may be significant, but very case-related.

Other parameters of importance are the bubble-point pressures for the various components. At given temperature, the bubble-point pressures signify the pressures where the respective phases start to boil. Below the bubble-point pressures, gas is released and we get transition of the components between the phases. For most realistic models, even if we do not distinguish between all the components, one allows gas to be dissolved in oil. For such models, an important pressure-dependent parameter is the solution gas-oil ratio r_l for the gas dissolved in oil at reservoir conditions. It is also common to introduce so-called formation volume factors that model the pressure dependent ratio of bulk volumes at reservoir and surface conditions. We will introduce these parameters later when presenting the three-phase black-oil model.

1.3 Production Processes

Initially, a hydrocarbon reservoir is at equilibrium, and contains gas, oil, and water, separated by gravity. This equilibrium has been established over millions of years with gravitational separation and geological and geothermal processes. When a well is drilled through the upper non-permeable layer and penetrates the upper hydrocarbon cap, this equilibrium is immediately disturbed. The reservoir is usually connected to the well and surface production facilities by a set of valves. If there were no production valves to stop the flow, we would have a "blow out" since the reservoir is usually under a high pressure. As the well is ready to produce, the valves are opened slightly, and hydrocarbons flow out of the reservoir due to over-pressure. This in turn, sets up a flow inside the reservoir and hydrocarbons flow towards the well, which in turn may induce gravitational instabilities. Capillary pressures will also act as a (minor) driving mechanism, resulting in local perturbations of the situation. During this stage, perhaps 20 percent of the hydrocarbons present are produced until a new equilibrium is achieved. We call this *primary production* by natural drives. One should note that a sudden drop in pressure also may

have numerous other intrinsic effects. Particularly in complex, composite systems this may be the case, as pressure-dependent parameters experience such drops. This may give non-convective transport and phase transfers, as vapour and gaseous hydrocarbons may suddenly condensate.

As pressure drops, less oil and gas is flowing, and eventually the production is no longer economically sustainable. Then the operating company may start *secondary production*, by engineered drives. These are processes based on injecting water or gas into the reservoir. The reason for doing this is twofold; some of the pressure is rebuilt or even increased, and secondly one tries to push out more profitable hydrocarbons with the injected substance. One may perhaps produce another 20 percent of the oil by such processes, and engineered drives are standard procedure at most locations in the North Sea today.

In order to produce even more oil, *Enhanced Oil Recovery* (EOR, or tertiary recovery) techniques may be employed. Among these are heating the reservoir or injection of sophisticated substances like foam, polymers or solvents. Polymers are supposed to change the flow properties of water, and thereby to more efficiently push out oil. Similarly, solvents change the flow properties of the hydrocarbons, for instance by developing miscibility with an injected gas. In some sense, one tries to wash the pore walls for most of the remaining hydrocarbons. The other technique is based on injecting steam, which will heat the rock matrix, and thereby, hopefully, change the flow properties of the hydrocarbons. At present, such EOR techniques are considered too expensive for large-scale commercial use, but several studies have been conducted and the mathematical foundations are being carefully investigated, and at smaller scales EOR is being performed.

One should note that the terms primary, secondary, and tertiary are ambiguous. EOR techniques may be applied during primary production, and secondary production may be performed from the first day of production.

2 Mathematical Models

In this section we will present two mathematical models, first a simple single-phase model that incorporates much of the complexities that arise due to heterogeneities in the porous rock formations. Then we present the classical black-oil model, which incorporates more complex flow physics.

2.1 Incompressible Single-Phase Flow

The simplest possible way to describe the displacement of fluids in a reservoir is by a single-phase model. This model gives an equation for the pressure distribution in the reservoir and is used for many early-stage and simplified flow studies. Single-phase models are used to identify flow directions; identify connections between producers and injectors; in flow-based upscaling; in history matching; and in preliminary model studies.

Assume that we want to model the filtration of a fluid through a porous medium of some kind. The basic equation describing this process is the continuity equation which states that mass is conserved

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla \cdot (\rho v) = q. \tag{2}$$

Here the source term q models sources and sinks, that is, outflow and inflow per volume at designated well locations.

For low velocities v, filtration through porous media is modelled with an empirical relation called Darcy's law after the French engineer Henri Darcy. Darcy discovered in 1856, through a series of experiments, that the filtration velocity is proportional to a combination of the gradient of the fluid pressure and pull-down effects due to gravity. More precisely, the volumetric flow density v (which we henceforth will refer to as flow velocity) is related to pressure p and gravity forces through the following gradient law:

$$v = -\frac{K}{\mu}(\nabla p + \rho g \nabla z). \tag{3}$$

Here g is the magnitude of the gravitational acceleration and z is the spatial coordinate in the upward vertical direction. For brevity we write $G = -g\nabla z$ for the gravitational pull-down force. We note that Darcy's law is analogous to Fourier's law of heat conduction (in which K is replaced with the heat conductivity tensor) and Ohm's law of electrical conduction (in which K is the inverse of the electrical resistance). However, whereas there is only one driving force in thermal and electrical conduction, there are two driving forces in porous media flow: gravity and the pressure gradient.

As an illustrative example, we will now present an equation that models flow of an incompressible fluid, say, water, through a rigid and incompressible porous medium characterised by a permeability field K and a corresponding porosity distribution ϕ. For an incompressible medium, the temporal derivative term in (2) vanishes and we obtain the following elliptic equation for the water pressure:

$$\nabla \cdot v = \nabla \cdot \left[-\frac{K}{\mu}(\nabla p - \rho G) \right] = \frac{q}{\rho}. \tag{4}$$

To close the model, we must specify boundary conditions. Unless stated otherwise we shall follow common practice and use no-flow boundary conditions. Hence, on the reservoir boundary $\partial\Omega$ we impose $v \cdot n = 0$, where n is the normal vector pointing out of the boundary $\partial\Omega$. This gives an isolated flow system where no water can enter or exit the reservoir.

2.2 Three-Phase Black-Oil Model

The most commonly used model in reservoir simulation is the so-called black oil model. Here we present the three-phase black-oil model, in which there are

three components; water (w), oil (o), and gas (g), and three phases; aqueous (a), liquid (l), and vapour (v). The aqueous phase contains only water, but oil and gas may exist in both the liquid phase and the vapour phase. The three-phase black-oil model is governed by mass-balance equations for each component

$$\sum_{j=a,l,v} \left\{ \frac{\partial}{dt}(\phi m_{\alpha,j}\rho_j s_j) + \nabla \cdot (m_{\alpha,j}\rho_j v_j) \right\} = q_\alpha, \qquad \alpha = w, o, g, \qquad (5)$$

where the Darcy velocities v_j are given by

$$v_j = -\frac{K k_{rj}}{\mu_j}(\nabla p_j - \rho_j G), \qquad j = a, l, v. \qquad (6)$$

Here q_α is a source term and p_j denotes the phase pressure.

We now introduce the volume formation factors $b_\alpha = V_{\alpha s}/V_\alpha$, where $V_{\alpha s}$ and V_α are volumes occupied by a bulk of component α at surface and reservoir conditions, respectively; the phase densities at surface conditions ρ_{js}; $r_l = V_{gs}/V_{os}$, the ratio of the volumes of gas and oil in the liquid phase at surface conditions; and $r_v = V_{os}/V_{gs}$, the ratio of the volumes of oil and gas in the vapour phase at surface conditions. Recalling that water does not mix into the liquid and vapour phases, we derive

$$
\begin{array}{lll}
m_{w,a}\rho_a = b_w\rho_{ws}, & m_{o,a} = 0, & m_{g,a} = 0, \\
m_{w,l} = 0, & m_{o,l}\rho_l = b_o\rho_{os}, & m_{g,l}\rho_l = r_l b_o\rho_{gs}, \\
m_{w,v} = 0, & m_{o,v}\rho_v = r_v b_g\rho_{os}, & m_{g,v}\rho_v = b_g\rho_{gs}.
\end{array}
$$

Inserting these expressions into (5) gives

$$\frac{\partial}{dt}(\phi A[s_j]) + \nabla \cdot (A[v_j]) = [q_\alpha], \qquad (7)$$

where $[\xi_j] = (\xi_a, \xi_l, \xi_v)^t$, $[\xi_\alpha] = (\xi_w, \xi_o, \xi_g)^t$, and

$$A = \begin{bmatrix} b_w\rho_{ws} & 0 & 0 \\ 0 & b_o\rho_{os} & r_v b_g\rho_{os} \\ 0 & r_l b_o\rho_{gs} & b_g\rho_{gs} \end{bmatrix} = \begin{bmatrix} \rho_{ws} & 0 & 0 \\ 0 & \rho_{os} & 0 \\ 0 & 0 & \rho_{gs} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & r_v \\ 0 & r_l & 1 \end{bmatrix} \begin{bmatrix} b_w & 0 & 0 \\ 0 & b_o & 0 \\ 0 & 0 & b_g \end{bmatrix}.$$

Premultiplying (7) with $\mathbf{1}^t A^{-1}$, expanding $\partial/\partial\xi = (\partial/\partial p_l)(\partial p_l/\partial\xi)$, and assuming $\mathbf{1}^t[s_j] = 1$, i.e., that the three phases occupy the void space completely, gives an equation of the following form:

$$\frac{\partial\phi}{\partial p_l} + \phi \sum_j c_j s_j \frac{\partial p_l}{\partial t} + \nabla \cdot \left(\sum_j v_j \right) + \sum_j c_j v_j \cdot \nabla p_l = q. \qquad (8)$$

Exercise: *Derive (8) from (7) and show that q and the phase compressibilities c_j are defined by*

$$q = \mathbf{1}^t A^{-1}[q_\alpha] = \frac{q_w}{b_w \rho_{ws}} + \frac{1}{1 - r_v r_l} \left(\left(\frac{1}{b_o} - \frac{r_l}{b_g} \right) \frac{q_o}{\rho_{os}} + \left(\frac{1}{b_g} - \frac{r_v}{b_o} \right) \frac{q_g}{\rho_{gs}} \right).$$

and

$$c_a = \frac{\partial \ln b_w}{\partial p_l}, \quad c_l = \frac{\partial \ln b_o}{\partial p_l} + \frac{1}{b_g} \frac{b_o - r_v b_g}{1 - r_v r_l} \frac{\partial r_l}{\partial p_l},$$

$$c_v = \frac{\partial \ln b_g}{\partial p_l} + \frac{1}{b_o} \frac{b_g - r_l b_o}{1 - r_v r_l} \frac{\partial r_v}{\partial p_l}.$$

3 Discretisation of Elliptic Pressure Equations

In this section we present four different numerical methods for solving elliptic pressure equations on the form (4). We only consider mass-conservative methods, meaning that each method provides velocity fields that satisfy the following mass-balance equation:

$$\int_{\Omega_i} \nabla \cdot v \, dx = \int_{\partial \Omega_i} v \cdot n \, ds = \int_{\Omega_i} \frac{q}{\rho} \, dx \tag{9}$$

for each grid cell Ω_i in Ω (the reservoir). Here n denotes the outward-pointing unit normal on $\partial \Omega_i$ and ds is the surface area measure. We first present the two-point flux-approximation (TPFA) scheme, a very simple discretisation technique that is widely used in the oil-industry.

3.1 The Two-Point Flux-Approximation (TPFA) Scheme

In classical finite-difference methods, partial differential equations (PDEs) are approximated by replacing the partial derivatives with appropriate divided differences between point-values on a discrete set of points in the domain. Finite-volume methods, on the other hand, have a more physical motivation and are derived from conservation of (physical) quantities over cell volumes. Thus, in a finite-volume method the unknown functions are represented in terms of average values over a set of finite volumes, over which the integrated PDE model is required to hold in an averaged sense.

   Although finite-difference and finite-volume methods have fundamentally different interpretation and derivation, the two labels are used interchangeably in the scientific literature. We therefore choose to not make a clear distinction between the two discretisation techniques here. Instead we ask the reader to think of a finite-volume method as a conservative finite-difference scheme that treats the grid cells as control volumes. In fact, there exist several finite-volume and finite-difference schemes of low order, for which the cell-centred values obtained with a finite-difference scheme coincide with cell averages obtained with the corresponding finite-volume scheme.

To derive a set of finite-volume mass-balance equations for (4), consider Equation (9). Finite-volume methods are obtained by approximating the pressure p with a cell-wise constant function $\{p_{w,i}\}$ and estimating the normal velocity $v \cdot n$ across cell interfaces $\gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$ from a set of neighbouring cell pressures. To formulate the TPFA scheme it is convenient to reformulate equation (4) slightly, so that we get an equation of the following form:

$$-\nabla \cdot \lambda\nabla u = f, \tag{10}$$

where $\lambda = K/\mu$. To this end, we have two options: we can either introduce a flow potential $u = p + \rho g z$ and express our model as an equation for u

$$-\nabla \cdot \lambda\nabla u = \frac{q}{\rho},$$

or we can move the gravity term $\nabla \cdot (\lambda\rho G)$ to the right-hand side. Hence, we might as well assume that we want to solve (10) for u.

As the name suggests, the TPFA scheme uses two points, the cell-averages u_i and u_j, to approximate the flux $F_{ij} = -\int_{\gamma_{ij}}(\lambda\nabla u) \cdot n\, ds$. To be more specific, let us consider a regular hexahedral grid with gridlines aligned with the principal coordinate axes. Moreover, assume that γ_{ij} is an interface between adjacent cells in the x–coordinate direction so that the interface normal n_{ij} equals $(1,0,0)^T$. The gradient ∇u on γ_{ij} in the TPFA method is now replaced with

$$(\nabla u \cdot n)|_{\gamma_{ij}} \approx \frac{2(u_j - u_i)}{\Delta x_i + \Delta x_j}, \tag{11}$$

where Δx_i and Δx_j denote the respective cell dimensions in the x-coordinate direction. Thus, we obtain the following expression for F_{ij}:

$$F_{ij} = -\frac{2(u_j - u_i)}{\Delta x_i + \Delta x_j} \int_{\gamma_{ij}} \lambda ds.$$

However, in most reservoir simulation models, the permeability K is cell-wise constant, and hence not well-defined at the interfaces. This means that we also have to approximate λ on γ_{ij}. In the TPFA method this is done by taking a distance-weighted harmonic average of the respective directional cell permeabilities, $\lambda_{i,ij} = n_{ij} \cdot \lambda_i n_{ij}$ and $\lambda_{j,ij} = n_{ij} \cdot \lambda_j n_{ij}$. To be precise, the n_{ij}–directional permeability λ_{ij} on γ_{ij} is computed as follows:

$$\lambda_{ij} = (\Delta x_i + \Delta x_j)\left(\frac{\Delta x_i}{\lambda_{i,ij}} + \frac{\Delta x_j}{\lambda_{j,ij}}\right)^{-1}.$$

Hence, for orthogonal grids with gridlines aligned with the coordinate axes, one approximates the flux F_{ij} in the TPFA method in the following way:

$$F_{ij} = -|\gamma_{ij}|\lambda_{ij}(\nabla u \cdot n)|_{\gamma_{ij}} = 2|\gamma_{ij}|\left(\frac{\Delta x_i}{\lambda_{i,ij}} + \frac{\Delta x_j}{\lambda_{j,ij}}\right)^{-1}(u_i - u_j). \tag{12}$$

Finally, summing over all interfaces, we get an approximation to $\int_{\partial\Omega_i} v \cdot n \, ds$, and the associated TPFA method is obtained by requiring the mass-balance equation (9) to be fulfilled for each grid cell $\Omega_i \in \Omega$.

In the literature on finite-volume methods it is common to express the flux F_{ij} in a more compact form than we have done in (12). Terms that do not involve the cell potentials u_i are usually gathered into an interface transmissibility t_{ij}. For the current TPFA method the transmissibilities are defined by:

$$t_{ij} = 2|\gamma_{ij}| \left(\frac{\Delta x_i}{\lambda_{i,ij}} + \frac{\Delta x_j}{\lambda_{j,ij}} \right)^{-1}.$$

Thus by inserting the expression for t_{ij} into (12), we see that the TPFA scheme for equation (10), in compact form, seeks a cell-wise constant function $\mathbf{u} = \{u_i\}$ that satisfies the following system of equations:

$$\sum_j t_{ij}(u_i - u_j) = \int_{\Omega_i} f \, dx, \qquad \forall \Omega_i \subset \Omega. \tag{13}$$

We have now derived a system of linear equations $\mathbf{Au} = \mathbf{f}$, where the matrix $\mathbf{A} = [a_{ik}]$ is given by

$$a_{ik} = \begin{cases} \sum_j t_{ij} & \text{if } k = i, \\ -t_{ik} & \text{if } k \neq i. \end{cases}$$

This system is symmetric, and a solution is, as for the continuous problem, defined up to an arbitrary constant. The system is made positive definite, and symmetry is preserved, by forcing $u_1 = 0$, for instance. That is, by adding a positive constant to the first diagonal of the matrix. In [2] we present a simple, but yet efficient, Matlab implementation of the TPFA scheme, which we have used in the following example:

Example 1. Our first example is the so-called quarter five-spot *test case, which is the most widespread test case within reservoir simulation. The reservoir is the unit square with an injector at $(0,0)$, a producer at $(1,1)$, and no-flow boundary conditions. Figure 4 shows pressure contours and streamlines for two different isotropic 32×32 permeability fields. The first field is homogeneous, whereas the other is sampled from a log-normal distribution. The pressure and velocity field are symmetric about both diagonals for the homogeneous field. For the heterogeneous field, the flow field is no longer symmetric since the fluids will seek to flow in the most high-permeable regions.*

3.2 Multipoint Flux-Approximation (MPFA) Schemes

The TPFA finite-volume scheme presented above is convergent only if each grid cell is a parallelepiped and

$$n_{ij} \cdot K n_{ik} = 0, \qquad \forall \Omega_i \subset \Omega, \qquad n_{ij} \neq \pm n_{ik}, \tag{14}$$
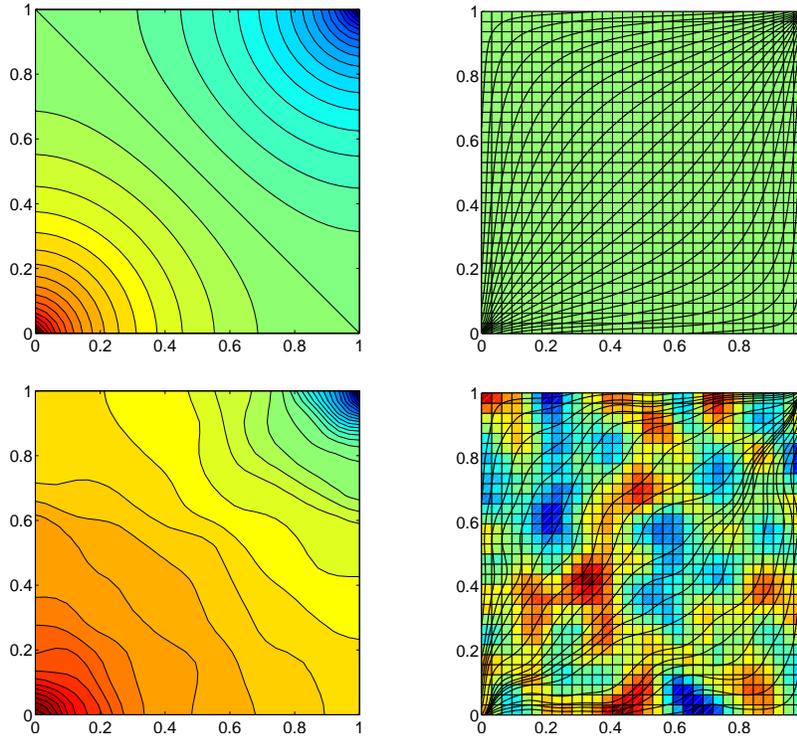
Fig. 4. Pressure contours and streamlines for the classical quarter five-spot test case with a homogeneous and a log-normal permeability field (top and bottom row, respectively).

where n_{ij} and n_{ik} denote normal vectors into two neighbouring grid cells. A grid consisting of parallelepipeds satisfying (14) is said to be K-orthogonal. Orthogonal grids are, for example, K-orthogonal with respect to diagonal permeability tensors, but not with respect to full tensor permeabilities. Figure 5 shows a schematic of an orthogonal grid and a K-orthogonal grid.

If the TPFA method is used to discretise (10) on grids that are not K-orthogonal, the scheme will produce different results depending on the orientation of the grid (so-called grid-orientation effects) and will generally converge to a wrong solution. Despite this shortcoming of the TPFA method, it is still the dominant (and default) method for practical reservoir simulation, owing to its simplicity and computational speed. We now present a class of so-called *multi-point flux-approximation (MPFA) schemes* that aim to amend the shortcomings of the TPFA scheme.

Consider an orthogonal grid and assume that $K = [K^{\xi,\zeta}]_{\xi,\zeta=x,y,z}$, is a constant tensor with nonzero off-diagonal terms and let γ_{ij} be an interface between two adjacent grid cells in the x–coordinate direction. Then for a
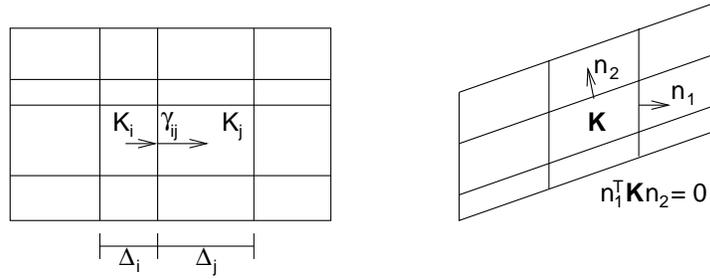
Fig. 5. The grid in the left plot is orthogonal with gridlines aligned with the principal coordinate axes. The grid in the right plot is a K-orthogonal grid.

given function u, the corresponding flux across γ_{ij} is given by:

$$\int_{\gamma_{ij}} v \cdot n_{ij} \, ds = -\int_{\gamma_{ij}} \frac{1}{\mu} \left(K^{x,x}\partial_x u + K^{x,y}\partial_y u + K^{x,z}\partial_z u \right) ds.$$

This expression involves derivatives in three orthogonal coordinate directions. Evidently, two point values can only be used to estimate a derivative in one direction. In particular, the two cell averages u_i and u_j can not be used to estimate the derivative of u in the y and z-directions. Hence, the TPFA scheme neglects the flux contribution from $K^{x,y}\partial_y u$ and $K^{x,z}\partial_z u$.

To obtain consistent interfacial fluxes for grids that are not **K**-orthogonal, one must also estimate partial derivatives in coordinate directions parallel to the interfaces. For this purpose, more than two point values, or cell averages, are needed. This leads to schemes that approximate F_{ij} using multiple cell averages, that is, with a linear expression on the form:

$$F_{ij} = \sum_k t_{ij}^k g_{ij}^k(\mathbf{u}).$$

Here $\{t_{ij}^k\}_k$ are the transmissibilities associated with γ_{ij} and $\{g_{ij}^k(\mathbf{u})\}_k$ are the corresponding multi-point pressure or flow potential dependencies. Thus, we see that MPFA schemes for (10) can be written on the form:

$$\sum_{j,k} t_{ij}^k g_{ij}^k(\mathbf{u}) = \int_{\Omega_i} f \, dx, \qquad \forall \Omega_i \subset \Omega. \tag{15}$$

MPFA schemes can, for instance, be designed by simply estimating each of the partial derivatives $\partial_\xi u$ from neighbouring cell averages. However, most MPFA schemes have a more physical motivation and are derived by imposing certain continuity requirements. We will now outline very briefly one such method, called the O-method [6, 7], for irregular, quadrilateral, matching grids in two spatial dimensions.

The O-method is constructed by defining an interaction region around each corner point in the grid. For a two-dimensional quadrilateral grid, this
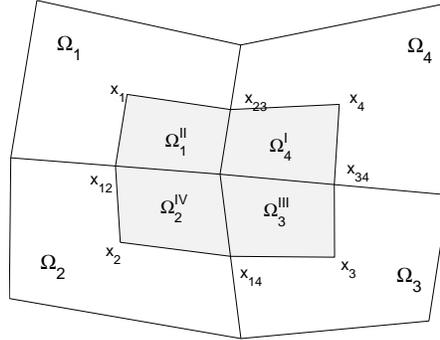
Fig. 6. The shaded region represents the interaction region for the O-method on a two-dimensional quadrilateral grid associated with cells Ω_1, Ω_2, Ω_3, and Ω_4.

interaction region is the area bounded by the lines that connect the cell-centres with the midpoints on the cell interfaces, see Figure 6. Thus, the interaction region consists of four sub-quadrilaterals ($\Omega_1^{\mathrm{II}}, \Omega_2^{\mathrm{IV}}, \Omega_3^{\mathrm{III}}$, and Ω_4^{I}) from four neighbouring cells (Ω_1, Ω_2, Ω_3, and Ω_4) that share a common corner point. For each interaction region, define

$$U_{\mathrm{IR}} = \mathrm{span}\{U_i^{\mathrm{J}} : i = 1, \ldots, 4, \quad \mathrm{J=I}, \ldots, \mathrm{IV}\},$$

where $\{U_i^{\mathrm{J}}\}$ are linear functions on the respective four sub-quadrilaterals. With this definition, U_{IR} has twelve degrees of freedom. Indeed, note that each U_i^{J} can be expressed in the following non-dimensional form

$$U_i^{\mathrm{J}}(x) = u_i + \nabla U_i^{\mathrm{J}} \cdot (x - x_i),$$

where x_i is the cell centre in Ω_i. The cell-centre values u_i thus account for four degrees of freedom and the (constant) gradients ∇U_i^{J} for additional eight.

Next we require that functions in U_{IR} are: (i) continuous at the midpoints of the cell interfaces, and (ii) flux-continuous across the interface segments that lie inside the interaction region. To obtain a globally coupled system, we first use (i) and (ii) to express the gradients ∇U_i^{J}, and hence also the corresponding fluxes across the interface segments of the interaction region, in terms of the unknown cell-centre potentials u_i. This requires solution of a local system of equations. Finally, the cell-centre potentials are determined (up to an arbitrary constant for no-flow boundary conditions) by summing the fluxes across all interface segments of the interaction region and requiring that the mass-balance equations (9) hold. In this process, transmissibilities are assembled to obtain a globally coupled system for the unknown pressures over the whole domain.

We note that this construction leads to an MPFA scheme where the flux across an interface γ_{ij} depends on the potentials u_j in a total of six neighbouring cells (eighteen in three dimensions). Notice also that the transmissibilities

$\{t_{ij}^k\}$ that we obtain when eliminating the gradients of the interaction region now account for grid-cell geometries in addition to full-tensor permeabilities.

3.3 A Mixed Finite-Element Method (FEM)

Whereas finite-volume methods treat velocities as functions of the unknown discrete pressures, mixed FEMs [18] obtain the velocity directly. The underlying idea is to consider both the pressure and the velocity as unknowns and express them in terms of basis functions. To this end, we return to the original formulation and describe how to discretise the following system of differential equations with mixed FEMs:

$$v = -\lambda(\nabla p - \rho G), \qquad \nabla \cdot v = q. \tag{16}$$

As before we impose no-flow boundary conditions on $\partial\Omega$. To derive the mixed formulation, we first define the following Sobolev space

$$H_0^{\mathrm{div}}(\Omega) = \{v \in (L^2(\Omega))^d : \nabla \cdot v \in L^2(\Omega) \text{ and } v \cdot n = 0 \text{ on } \partial\Omega\}.$$

The mixed formulation of (16) with no-flow boundary conditions now reads: find $(p, v) \in L^2(\Omega) \times H_0^{\mathrm{div}}(\Omega)$ such that

$$\int_\Omega v \cdot \lambda^{-1} u \, dx - \int_\Omega p \, \nabla \cdot u \, dx = \int_\Omega \rho G \cdot u \, dx, \tag{17}$$

$$\int_\Omega l \, \nabla \cdot v \, dx = \int_\Omega q l \, dx, \tag{18}$$

for all $u \in H_0^{\mathrm{div}}(\Omega)$ and $l \in L^2(\Omega)$. We observe again that, since no-flow boundary conditions are imposed, an extra constraint must be added to make (17)–(18) well-posed. A common choice is to use $\int_\Omega p \, dx = 0$.

In mixed FEMs, (17)–(18) are discretised by replacing $L^2(\Omega)$ and $H_0^{\mathrm{div}}(\Omega)$ with finite-dimensional subspaces U and V, respectively. For instance, in the Raviart–Thomas mixed FEM [44] of lowest order (for triangular, tetrahedral, or regular parallelepiped grids), $L^2(\Omega)$ is replaced by

$$U = \{p \in L^2(\Omega) : p|_{\Omega_i} \text{ is constant } \forall \Omega_i \in \Omega\}$$

and $H_0^{\mathrm{div}}(\Omega)$ is replaced by

$$V = \{v \in H_0^{\mathrm{div}}(\Omega) : v|_{\Omega_i} \text{ has linear components } \forall \Omega_i \in \Omega,$$
$$(v \cdot n_{ij})|_{\gamma_{ij}} \text{ is constant } \forall \gamma_{ij} \in \Omega, \text{ and } v \cdot n_{ij} \text{ is continuous across } \gamma_{ij}\}.$$

Here n_{ij} is the unit normal to γ_{ij} pointing from Ω_i to Ω_j. The corresponding Raviart–Thomas mixed FEM thus seeks

$$(p, v) \in U \times V \text{ such that (17)–(18) hold for all } u \in V \text{ and } q \in U. \tag{19}$$

To express (19) as a linear system, observe first that functions in V are, for admissible grids, spanned by base functions $\{\psi_{ij}\}$ that are defined by

$$\psi_{ij} \in \mathcal{P}_1(\Omega_i)^d \cup \mathcal{P}_1(\Omega_j)^d \quad \text{and} \quad (\psi_{ij} \cdot n_{kl})|_{\gamma_{kl}} = \begin{cases} 1, & \text{if} \quad \gamma_{kl} = \gamma_{ij}, \\ 0, & \text{else}, \end{cases}$$

where $\mathcal{P}_1(B)$ is the set of linear functions on B. Similarly,

$$U = \operatorname{span}\{\chi_m\} \quad \text{where} \quad \chi_m = \begin{cases} 1, & \text{if} \quad x \in \Omega_m, \\ 0, & \text{else}. \end{cases}$$

Thus, writing $p = \sum_{\Omega_m} p_m \chi_m$ and $v = \sum_{\gamma_{ij}} v_{ij} \psi_{ij}$, allows us to write (19) as a linear system in $\mathbf{p} = \{p_m\}$ and $\mathbf{v} = \{v_{ij}\}$. This system takes the form

$$\begin{bmatrix} \mathbf{B} & -\mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix}. \tag{20}$$

Here $\mathbf{f} = [f_m]$, $\mathbf{g} = [g_{kl}]$, $\mathbf{B} = [b_{ij,kl}]$ and $\mathbf{C} = [c_{m,kl}]$, where:

$$g_{kl} = \left[\int_\Omega \rho G \cdot \psi_{kl} \, dx \right], \qquad f_m = \left[\int_{\Omega_m} f \, dx \right],$$

$$b_{ij,kl} = \left[\int_\Omega \psi_{ij} \cdot \lambda^{-1} \psi_{kl} \, dx \right], \qquad c_{m,kl} = \left[\int_{\Omega_m} \nabla \cdot \psi_{kl} \, dx \right].$$

A drawback with the mixed FEM is that it produces an indefinite linear system. These systems are in general harder to solve than the positive definite systems that arise, e.g., from the TPFA and MPFA schemes described in Sections 3.1 and 3.2. However, for second-order elliptic equations of the form (4) it is common to use a so-called hybrid formulation. This method leads to a positive definite system where the unknowns correspond to pressures at grid-cell interfaces. The solution to the linear system arising from the mixed FEM can now easily be obtained from the solution to the hybrid system by performing only local algebraic calculations.

3.4 A Mimetic Finite Difference Method (FDM)

The current mimetic FDM [19, 20] is based on the same principles as the above mixed FEM, but the approximation space $V \subset H^{\operatorname{div}}(\Omega)$ is replaced with a space $M \subset L^2(\cup_i \partial \Omega_i)$, and the L^2 inner product on $H^{\operatorname{div}}(\Omega)$ is replaced with an approximative form $m(\cdot, \cdot)$ that acts on $L^2(\cup_i \partial \Omega_i)$. Moreover, whereas functions in V represent velocities, functions in M represent fluxes across grid cell boundaries. Thus, for the current mimetic FDM

$$M = \operatorname{span}\{\psi_{ij}\}, \qquad \psi_{ij} = \begin{cases} 1, & \text{on } \gamma_{ij}, \\ 0, & \text{on } \gamma_{kl}, \ kl \neq ij, \end{cases}$$

where one interprets ψ_{ij} to be a basis function that represents a quantity of flow with unit velocity across γ_{ij} in the direction of the unit normal n_{ij}, and zero flow across all other interfaces. Hence, conceptually, the only difference between these basis functions and the Raviart–Thomas basis functions is that we here do not associate a corresponding velocity field in Ω_i and Ω_j.

Next, we present an inner-product $m(u, v)$ on M that *mimics* or "approximates" the L^2 inner-product $(u, \lambda^{-1}v)$ on $H^{\mathrm{div}}(\Omega)$. That is, if $u, v \in H^{\mathrm{div}}(\Omega)$, then we want to derive an inner-product $m(\cdot, \cdot)$ so that

$$(u, \lambda^{-1}v) \approx m(u, v) = \sum_k \sum_{i,j} u_{ki} v_{kj} m(\psi_{ki}, \psi_{kj}) = \sum_k \mathbf{u}_k^t \mathbf{M}_k \mathbf{v}_k, \qquad (21)$$

where u_{ki} and v_{ki} are the average velocities across γ_{ki} corresponding to u and v, respectively, and $\mathbf{u}_k = [u_{ki}]_i$, $\mathbf{v}_k = [v_{ki}]_i$. Furthermore, $\mathbf{M}_k$ is defined by

$$\mathbf{M}_k = \frac{1}{|\Omega_k|} \mathbf{C}_k \lambda^{-1} \mathbf{C}_k^t + \frac{|\Omega_k|}{2\mathrm{trace}(\lambda)} (\mathbf{I} - \mathbf{Q}_k \mathbf{Q}_k^t), \qquad (22)$$

where the matrices $\mathbf{C}_k$, and $\mathbf{Q}_k$ are defined as follows:

$\mathbf{N}_k$: row i is defined by

$$\mathbf{n}_{k,i} = \frac{1}{|\gamma_{ki}|} \int_{\gamma_{ki}} (n_{ki})^t \, ds,$$

$\mathbf{C}_k$: row i is defined by

$$\mathbf{c}_{k,i} = \int_{\gamma_{ki}} (x - x_k)^t \, ds,$$

where x_k is the mass centre of Ω_k,

$\mathbf{Q}_k$: columns form an orthonormal basis for the column space of $\mathbf{N}_k$.

The discrete system that arises from this mimetic FDM is of the same form as (20). The only difference at the discrete level is that the entries in $\mathbf{B}$ and $\mathbf{g}$ are computed using the $m(\cdot, \cdot)$ inner-product instead of the L^2 inner-product $(u, \lambda^{-1}v)$ on $H^{\mathrm{div}}(\Omega)$. Thus, for the mimetic FDM we have

$$g_{kl} = \left[m(\rho \Xi, \psi_{kl}) \right], \qquad b_{ij,kl} = \left[m(\psi_{ij}, \psi_{kl}) \right],$$

where $\Xi = \sum_{ij} \xi_{ij} \psi_{ij}$ and $\xi_{ij} = \frac{1}{|\gamma_{ij}|} \int_{\gamma_{ij}} G \cdot n_{ij} \, ds$.

3.5 General Remarks

Using geological models as input to flow simulation introduces several numerical difficulties. First of all, typical reservoirs extend several hundred or thousand metres in the lateral direction, but the zones carrying hydrocarbon may be just a few tens of metres in the vertical direction and consist of several layers with different rock properties. Geological models therefore have grid-cells with very high aspect ratios and often the majority of the flow in
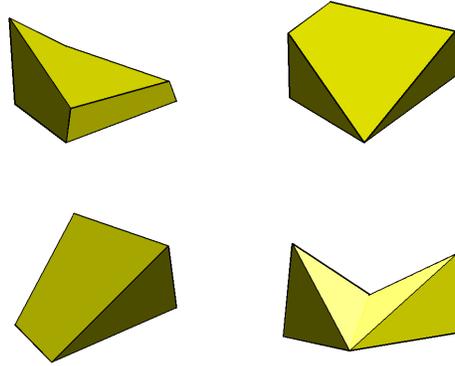
Fig. 7. Examples of deformed and degenerate hexahedral cells arising in corner-point grid models.

and out of a cell occurs across the faces with the smallest area. Similarly, the possible presence of strong heterogeneities and anisotropies in the permeability fields typically introduces large conditions numbers in the discretised flow equations. These difficulties are observed even for grid models consisting of regular hexahedral cells.

The flexibility in cell geometry of the industry-standard corner-point format introduces additional difficulties. First of all, since each face of a grid cell is specified by four (arbitrary) points, the cell interfaces in the grid will generally be bilinear surfaces and possibly be strongly curved. Secondly, corner-point cells may have zero volume, which introduces coupling between non-neighbouring cells and gives rise to discretisation matrices with complex sparsity patterns. Moreover, the presence of degenerate cells, in which the corner-points collapse in pairs, means that the cells will generally be polyhedral and possibly contain both triangular and quadrilateral faces (see Figure 7). Finally, non-conforming grids arise, using the corner-point format, in fault zones where a displacement along a hyperplane has occurred, see Figure 8. Altogether, this calls for a very flexible discretisation that is not sensitive to the geometry of each cell or the number of faces and corner points.

Having said this, it is appropriate with some brief remarks on the applicability of the methods presented above.

TPFA: Most commercial reservoir simulators use traditional finite-difference methods like the TPFA scheme. These methods were not designed to cope with the type of grid models that are built today using modern geomodelling tools. Hence, if one is interested in *accurate* solutions, two-point schemes should be avoided.

MPFA methods amend shortcomings of two-point scheme, but are unfortunately hard to implement for general grids, especially if the grid is non-conforming with non-matching faces.
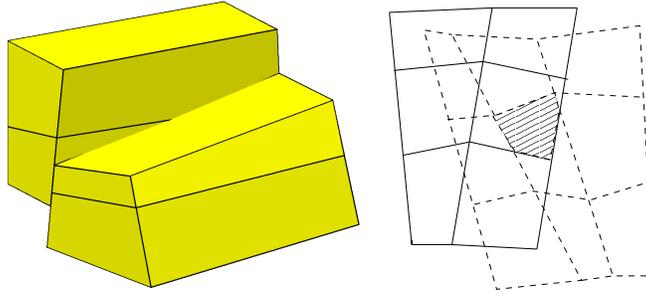
Fig. 8. Two examples of fault surface in a three-dimensional model with non-matching interfaces across the faults. (Left) Three-dimensional view. (Right) Two-dimensional view, where the shaded patch illustrates a "sub-interface" on the fault surface.

Mixed FEMs are more accurate than two-point schemes and generally quite robust. However, the different cells in geological models are generally not diffeomorphic. One therefore needs to introduce a reference element and a corresponding Piola transform for each topological case. This complicates the implementation of a mixed FEM considerably. Moreover, mixed FEMs gives rise to larger linear systems than TPFA and MPFA.

Mimetic FDMs have similar accuracy to MPFA methods and low-order mixed FEMs. But unlike MPFA methods and mixed FEMs, mimetic FDMs are quite easy to formulate and implement for grids with general polyhedral cells. In particular, it is relatively straightforward to handle grids with irregular cell geometries and non-matching faces.

4 Upscaling for Reservoir Simulation

The basic motivation behind upscaling is to create simulation models that produce flow scenarios that are in close correspondence with the flow scenarios that one would obtain by running simulations directly on the geomodels. The literature on upscaling techniques is extensive, ranging from simple averaging techniques, e.g., [37], via local simulation techniques [14, 28], to multiscale methods [1, 8, 9, 22, 33, 34] and homogenisation techniques for periodic structures [15, 32, 36]. It is not within our scope to give a complete overview over the many upscaling techniques that have been applied in reservoir simulation. Instead, we refer the reader to the many review papers that have been devoted to this topic, e.g., [13, 24, 45, 48]. Here we give only a brief introduction to upscaling rock permeability for the pressure equation.

The process of upscaling permeability for the pressure equation (4) or (8) is often termed single-phase upscaling. Most single-phase upscaling techniques seek homogeneous block permeabilities that reproduce the same total flow through each coarse grid-block as one would get if the pressure equation

was solved on the underlying fine grid with the correct fine-scale heterogeneous structures. However, designing upscaling techniques that preserve averaged fine-scale flow-rates is in general nontrivial because the heterogeneity at all scales have a significant effect on the large-scale flow pattern. A proper coarse-scale reservoir model must therefore capture the impact of heterogeneous structures at all scales that are not resolved by the coarse grid.

To illustrate the concept behind single-phase upscaling, let p be the solution that we obtain by solving

$$-\nabla \cdot K \nabla p = q, \qquad \text{in } \Omega \tag{23}$$

on a fine grid with a suitable numerical method, e.g., a TPFA scheme of the form (13). To reproduce the same total flow through a grid-block V we have to find a homogenised tensor K_V^* such that

$$\int_V K \nabla p \, dx = K_V^* \int_V \nabla p \, dx. \tag{24}$$

This equation states that the net flow-rate $\bar{v}$ through V is related to the average pressure gradient $\overline{\nabla p}$ in V through the upscaled Darcy law $\bar{v} = -K^* \overline{\nabla p}$.

Note that for a given pressure field p, the upscaled permeability tensor K_V^* is not uniquely defined by (24). Conversely, there does not exist a K_V^* such that (24) holds for any pressure field. This reflects that K_V^* depends on the flow through V. Of course, one does not know a priori what flow scenario V will be subject to. However, the aim is not to replicate a particular flow regime, but to compute coarse-scale permeability tensors that give reasonably accurate results for a wide range of flow scenarios. We now review some of the most commonly used single-phase upscaling methods.

Averaging Methods

The simplest method to upscale permeability is to compute the average of the permeabilities inside the coarse block. To this end, power averaging is a popular technique

$$K_V^{*,p} = \left(\frac{1}{|V|} \int_V K(x)^p \, dx \right)^{1/p}, \quad -1 \leq p \leq 1.$$

Special cases include the arithmetic average ($p = 1$), the harmonic average ($p = -1$), and the geometric average ($p \to 0$).

The use of power averaging can be motivated by the so-called Wiener-bounds [49], which state that for a statistically homogeneous medium, the correct upscaled permeability will be bounded above and below by the arithmetic and harmonic mean, respectively. This result has a more intuitive explanation. To see this, consider the one-dimensional pressure equation:

$$-\partial_x (K(x) p'(x)) = 0 \quad \text{in } (0,1), \quad p(0) = p_0, \, p(1) = p_1.$$

Integrating once, we see that the corresponding Darcy velocity is constant. This implies that $p'(x)$ must scale proportional to the inverse of $K(x)$. Hence, we derive

$$p'(x) = \frac{p_1 - p_0}{K(x)} \left[\int_0^1 \frac{dx}{K(x)} \right]^{-1} = \frac{p_1 - p_0}{K(x)} K_V^{*,-1}.$$

If we insert this expression into (24) we find that the correct upscaled permeability K_V^* is identical to the harmonic mean $K_V^{*,-1}$.

The same argument applies to the special case of a perfectly stratified isotropic medium; for instance, with layers perpendicular to the x–axis so that $K(x, \cdot, \cdot)$ is constant for each x. Now, consider a uniform flow in the x–direction:

$$\begin{aligned} -\nabla \cdot K \nabla p &= 0 \quad \text{in } V = (0,1)^3, \\ p(0, y, z) &= p_0, \quad p(1, y, z) = p_1, \\ (-K\nabla p) \cdot n &= 0 \quad \text{for } y, z \in \{0, 1\}, \end{aligned} \tag{25}$$

where n is the outward unit normal on ∂V. This means that for each pair $(y, z) \in (0, 1)^2$ the one-dimensional function $p_{y,z} = p(\cdot, y, z)$ satisfies

$$-\partial_x \big(K p'_{y,z}(x) \big) = 0 \text{ in } (0, 1), \quad p_{y,z}(0) = p_0, \ p_{y,z}(1) = p_1,$$

from which it follows that

$$-K(x)\nabla p = -(K(x)p'_{y,z}(x), 0, 0)^T = -K_V^{*,-1}(p_1 - p_0, 0, 0)^T.$$

Hence, the correct upscaled permeability is equal to the harmonic mean.

Exercise: *Show that if K instead models a stratified isotropic medium with layers perpendicular to the y or z–axis, then the correct upscaled permeability for uniform flow in the x–direction would be equal to the arithmetic mean.*

The discussion above shows that averaging techniques can be appropriate in special cases. However, if we consider the model problem (25) with a less idealised heterogeneous structures, or with the same heterogeneous structures but with other boundary conditions, then both the arithmetic and harmonic average will generally give wrong net flow-rates. Indeed, these averages give correct upscaled permeability only for cases with essentially one-dimensional flow. To try to model flow in more than one direction, one could generate a diagonal permeability tensor with the following diagonal components:

$$K^{x,x} = \mu_a^z(\mu_a^y(\mu_h^x)), \quad K^{y,y} = \mu_a^z(\mu_a^x(\mu_h^y)), \quad K^{z,z} = \mu_a^x(\mu_a^y(\mu_h^z)).$$

Here μ_a^ξ and μ_h^ξ represent the arithmetic and harmonic means, respectively, in the ξ-coordinate direction. Thus, in this method one starts by taking a harmonic average along grid cells that are aligned in one coordinate-direction. One then computes the corresponding diagonal by taking the arithmetic mean of all "one dimensional" harmonic means. This average is sometimes called
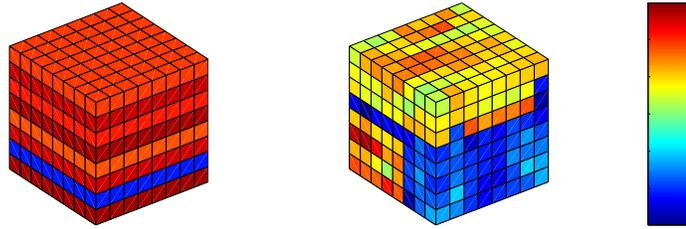
Fig. 9. Logarithm of permeability: the left cube is a layered medium, whereas the right cube is extracted from the lower part of the fluvial Upper Ness formation from Model 2 of the 10th SPE Comparative Solution Project [25].

the harmonic-arithmetic average and may give good results if, for instance, the reservoir is layered and the primary direction of flow is along the layers.

Despite the fact that averaging techniques can give correct upscaling in special cases, they tend to perform poorly in practice since the averages do not reflect the structure or orientation of the heterogeneous structures. It is also difficult to decide which averaging technique to use since the best average depends both on the heterogeneity of the media and on the flow process we want to model (flow direction, boundary conditions, etc). To illustrate the dependence on the flow process we consider an example.

Example 2 (from [2]). Consider a reservoir in the unit cube $[0,1]^3$ with two different geomodels that each consist of a $8 \times 8 \times 8$ uniform grid blocks and permeability distribution as depicted in Figure 9. We consider three different upscaling methods: harmonic average (H), arithmetic average (A), and harmonic-arithmetic average (HA). The geomodels are upscaled to a single grid-block, which is then subjected to three different boundary conditions:

BC1: $p = 1$ at $(x, y, 0)$, $p = 0$ at $(x, y, 1)$, no-flow elsewhere.
BC2: $p = 1$ at $(0, 0, z)$, $p = 0$ at $(1, 1, z)$, no-flow elsewhere.
BC3: $p = 1$ at $(0, 0, 0)$, $p = 0$ at $(1, 1, 1)$, no-flow elsewhere.

Table 1 compares the observed coarse-block rates with the flow-rate obtained by direct simulation on the $8 \times 8 \times 8$ grid. For the layered model, harmonic and harmonic-arithmetic averaging correctly reproduce the vertical flow normal to the layers for BC1. Arithmetic and harmonic-arithmetic averaging correctly reproduce the flow along the layers for BC2. Harmonic-arithmetic averaging also performs well for corner-to-corner flow (BC3). For model two, however, all methods produce significant errors, and none of the methods are able to produce an accurate flow-rate for boundary conditions BC1 and BC3.

Flow-Based Upscaling

A popular class of methods are so-called flow-based upscaling methods as first suggested by Begg et al. [14]. In this approach one solves a set of homogeneous

Table 1. Flow-rates relative to the reference rate Q_R on the fine grid.

	Model 1			Model 2		
	BC1	BC2	BC3	BC1	BC2	BC3
Q_H/Q_R	1	2.31e−04	5.52e−02	1.10e−02	3.82e−06	9.94e−04
Q_A/Q_R	4.33e+03	1	2.39e+02	2.33e+04	8.22	2.13e+03
Q_{HA}/Q_R	1	1	1.14	8.14e−02	1.00	1.55e−01

pressure equations on the form

$$-\nabla \cdot K \nabla p = 0 \quad \text{in } V,$$

for each grid block V with prescribed boundary conditions that induce a desired flow pattern. Each member of this class of methods differ in the way boundary conditions are prescribed.

A simple and popular choice is to impose a pressure drop in one of the coordinate directions and no-flow conditions along the other faces, as in (25) for flow in the x–direction. This gives us a set of three flow-rates for each grid block that can be used to compute an effective diagonal permeability tensor with components

$$K^{x,x} = -Q_x L_x / \Delta P_x, \qquad K^{y,y} = -Q_y L_y / \Delta P_y, \qquad K^{z,z} = -Q_z L_z / \Delta P_z.$$

Here Q_ξ, L_ξ and ΔP_ξ are the net flow, the length between opposite sides, and the pressure drop in the ξ-direction inside V, respectively.

Another popular option is to choose periodic boundary conditions. That is, one assumes that each grid block is a unit cell in a periodic medium and imposes full correspondence between the pressures and velocities at opposite sides of the block; that is, to compute $K^{x,x}$, $K^{x,y}$, and $K^{x,z}$ we impose the following boundary conditions:

$$p(1,y,z) = p(0,y,z) - \Delta p, \quad p(x,1,z) = p(x,0,z), \quad p(x,y,1) = p(x,y,0),$$
$$v(1,y,z) = v(0,y,z), \quad v(x,1,z) = v(x,0,z), \quad v(x,y,1) = v(x,y,0),$$

and define $K^{x,\xi} = -Q_\xi L_\xi / \Delta p$. This approach yields a symmetric and positive definite tensor [28], and is usually more robust than the directional flow boundary conditions.

Example 3 (from [2]). We revisit the test-cases considered in Example 2, but now we compare harmonic-arithmetic averaging (HA) with the flow-based techniques using directional (D) and periodic (P) boundary conditions. The latter method gives rise to full permeability tensors, but for the cases considered here the off-diagonal terms in the upscaled permeability tensors are small, and are therefore neglected for simplicity.

Table 2 compares the observed coarse-block rates with the flow-rate obtained by direct simulation on the $8 \times 8 \times 8$ grid. For the layered model, all

Table 2. Flow-rates relative to the reference rate Q_R on the fine grid.

| | Model 1 | | | Model 2 | | |
	BC1	BC2	BC3	BC1	BC2	BC3
Q_{HA}/Q_R	1	1	1.143	0.081	1.003	0.155
Q_D/Q_R	1	1	1.143	1	1.375	1.893
Q_P/Q_R	1	1	1.143	0.986	1.321	1.867

methods give the same diagonal permeability tensor, and hence give exactly the same results. For Model 2 we see that the numerical pressure computation methods give significantly better results than the harmonic-arithmetic average. Indeed, the worst results for the pressure computation method, which were obtained for corner-to-corner flow, is within a factor two, whereas the harmonic-arithmetic average underestimates the flow rates for BC1 and BC3 by almost an order of magnitude.

It should be noted that in the discrete case, the appropriate upscaling technique depends on the underlying numerical method. For instance, if the pressure equation is discretised by a TPFA scheme of the form (13), then grid-block permeabilities are used only to compute interface transmissibilities at the coarse scale. Upscaling methods for this method may therefore instead be targeted at computing coarse-scale transmissibilities (that reproduce a fine-scale flow field in an averaged sense) directly. Procedures for computing coarse-scale transmissibilities similar to the averaging and numerical pressure computation techniques have been proposed in [38] and, e.g., [31], respectively.

5 Multiscale Methods the Pressure Equation

Subsurface flow problems represent an important application that calls for a more mathematically rigorous treatment of the way the large span of permeability values and correlation lengths impact the solution. Conventional methods are inadequate for this problem because the heterogeneity in natural porous media does not have clearly separated scales of variation, and because permeability variations occurring at small length scales (e.g., smaller scale than the grid resolution) may have strong impact on the flow at much larger scales. This makes subsurface flow problems a natural target for a new class of methods called multiscale methods – methods that attempt to model physical phenomena on coarse grids while honouring small-scale features that impact the coarse grid solution in an appropriate way, e.g., by incorporating subgrid information into numerical schemes for partial differential equations in a way that is consistent with the local property of the differential operator.

A large number of multiscale methods have appeared in the literature on computational science and engineering. Among these, there are a variety of

methods (e.g., [1, 8, 9, 22, 33, 34]) that target solving elliptic equations of the same form as the pressure equation for incompressible subsurface flow. Upscaling methods that derive coarse-grid properties from numerical subgrid calculations may also in a certain sense be viewed as multiscale methods, but the way the upscaled properties are incorporated into the coarse-scale systems is not necessarily consistent with the properties of the differential operator.

In this section we present three selected multiscale methods. The main idea is to show how multiscale methods are built, and how subgrid information is embedded into the coarse-scale system. For presentational brevity and enhanced readability we consider only elliptic (incompressible flow) equations, and disregard capillary forces so that $\nabla p_j = \nabla p$ for all phases j.

Let Ω denote our reservoir. Furthermore, let $\mathcal{B} = \{B_i\}$ be a partitioning of Ω into polyhedral grid-blocks and let $\{\Gamma_{ij} = \partial B_i \cap \partial B_j\}$ be the corresponding set of non-degenerate interfaces. Throughout we implicitly assume that all grid-blocks B_i are divided into smaller grid cells that form a sub-partitioning of Ω. Without compressibility and capillary forces, the pressure equation for the three-phase black-oil model now reads:

$$v = -K(\lambda \nabla p - \lambda_G G), \qquad \nabla \cdot v = q \qquad \text{in } \Omega. \tag{26}$$

where we have inserted $v = \sum_j v_j$, $\lambda = \sum_j \frac{k_{rj}}{\mu_j}$, and $\lambda_G = \sum_j \rho_j \frac{k_{rj}}{\mu_j}$ for brevity. We assume that no-flow boundary conditions $v \cdot n = 0$ are imposed on $\partial \Omega$, and that p is uniquely determined by adding the constraint $\int_\Omega p \, dx = 0$.

5.1 The Multiscale Finite-Element Method (MsFEM) in 1D

Before we introduce multiscale methods for solving (26) in three-dimensional domains, we start with an instrumental example in one spatial dimension. To this end, we consider the following elliptic problem:

$$\partial_x \left(K(x) p'(x) \right) = f, \text{ in } \Omega = (0,1), \qquad p(0) = p(1) = 0, \tag{27}$$

where $f, K \in L^2(\Omega)$ and K is bounded above and below by positive constants.

The MsFEM was first introduced by Hou and Wu [33], but the basic idea goes back to earlier work by Babuška and Osborn [12] for 1D problems and Babuška, Caloz, and Osborn [11] for special 2D problems. The method is, like standard FEMs, based on a variational formulation. In the variational formulation of (27) we seek $p \in H_0^1(\Omega)$ such that

$$a(p, v) = (f, v) \quad \text{for all } v \in H_0^1(\Omega), \tag{28}$$

where $(\cdot, \cdot)$ is the L^2 inner-product and

$$a(p, v) = \int_\Omega K(x) u'(x) v'(x) \, dx.$$

Now, let $\mathcal{N}_\mathcal{B} = \{0 = x_0 < x_1 < \ldots < x_{n-1} < x_n = 1\}$ be a set of nodal points and define $B_i = (x_{i-1}, x_i)$. For each x_i, $i = 1, \ldots, n-1$ we associate a corresponding basis function $\phi^i \in H_0^1(\Omega)$ defined by

$$a(\phi^i, v) = 0 \quad \text{for all } v \in H_0^1(B_i \cup B_{i+1}), \qquad \phi_i(x_j) = \delta_{ij}, \qquad (29)$$

where δ_{ij} is the Kronecker delta. The multiscale finite-element method seeks the unique function p_0 in

$$V^{\mathrm{ms}} = \mathrm{span}\{\phi_i\} = \{u \in H_0^1(\Omega) : a(u, v) = 0 \text{ for all } v \in H_0^1(\cup_i B_i)\} \qquad (30)$$

satisfying

$$a(p_0, v) = (f, v) \quad \text{for all } v \in V^{\mathrm{ms}}. \qquad (31)$$

We now show that the solution p of (28) can be written as a sum of p_0 and a family of solutions to independent local subgrid problems. To this end, we first show that $p_0 = p_I$, where p_I is the unique function in V^{ms} with $p_I(x) = p(x)$, $x \in \mathcal{N}_\mathcal{B}$. Indeed, since $p - p_I$ vanishes on $\mathcal{N}_\mathcal{B}$, we have $p - p_I \in H_0^1(\cup_i B_i)$. Hence, it follows from (28) and the mutual orthogonality of V^{ms} and $H_0^1(\cup_i B_i)$ with respect to $a(\cdot, \cdot)$ that

$$a(p_I, v) = a(p, v) = (f, v) \quad \text{for all } v \in V^{\mathrm{ms}}.$$

Thus, in particular, by (31) and choosing $v = p_I - p_0$ we obtain

$$a(p_I - p_0, p_I - p_0) = 0,$$

which implies $p_0 = p_I$. Thus, $p = p_0 + \sum_{i>0} p_i$ where $p_i \in H_0^1(B_i)$ is defined by

$$a(p_i, v) = (f, v) \quad \text{for all } v \in H_0^1(B_i).$$

Hence, as promised, the solution of (28) is a sum of p_0 and solutions to independent local subgrid problems. This result can also be seen directly by noting that p_0 is, by definition, the orthogonal projection onto V^{ms} with respect to the inner-product $a(\cdot, \cdot)$ and noting that $H_0^1(\Omega) = V^{\mathrm{ms}} \oplus H_0^1(\cup_i B_i)$.

Exercise: *Show that*

$$a(\phi_i, \phi_j) = \begin{cases} K_i^{*,-1}/(x_i - x_{i-1}) + K_{i+1}^{*,-1}/(x_{i+1} - x_i), & \text{if } i = j, \\ -K_{\max(i,j)}^{*,-1}/|x_i - x_j|, & \text{if } |i - j| = 1, \\ 0, & \text{if } |i - j| > 1, \end{cases} \qquad (32)$$

where $K_i^{*,-1}$ is the harmonic mean of K over the interval $[x_{i-1}, x_i]$, i.e.,

$$K_i^{*,-1} = \frac{x_i - x_{i-1}}{\int_{x_{i-1}}^{x_i} K(x)^{-1}\, dx}.$$

Consider next the standard nodal basis functions used in the linear FEM. Here the basis functions ϕ_i are linear on each interval and satisfy $\phi_i(x_j) = \delta_{ij}$. Show

that the corresponding coefficients for this method is obtained by replacing the harmonic means in (32) with the associated arithmetic means.

The multiscale finite-element method can also be extended to higher dimensions, but does not give locally mass-conservative velocity fields. Next we present a multiscale finite-volume method that is essentially a control-volume finite-element version of the MsFEM. Control-volume finite-element methods seek solutions in designated finite-element approximation spaces (on a dual-grid), but rather than formulating the global problem in a variational framework, they employ a finite-volume formulation (on a primal grid) that gives mass-conservative velocity fields.

5.2 The Multiscale Finite-Volume Method (MsFVM)

The multiscale finite-volume method [34] employs numerical subgrid calculations (analogous to those in [33]) to derive a multi-point stencil for solving (26) on a coarse grid. The method then proceeds and reconstructs a mass-conservative velocity field on a fine grid as a superposition of local subgrid solutions, where the weights are obtained from the coarse-grid solution.

The derivation of the coarse-scale equations in the MsFVM is essentially an upscaling procedure for generating coarse-scale transmissibilities. The first step is to solve a set of homogeneous boundary-value problems of the form

$$-\nabla \cdot K\lambda \nabla \phi_i^k = 0, \quad \text{in } R, \qquad \phi_i^k = \nu_i^k, \quad \text{on } \partial R, \tag{33}$$

where R are so-called interaction regions as illustrated in Figure 10 and ν_i^k are boundary conditions to be specified below. Subscript i in ϕ_i^k denotes a corner-point in the coarse grid (x_i in the figure) and the superscript k runs over all corner points of the interaction region (x^k in the figure). Thus, for each interaction region associated with e.g., a hexahedral grid in three dimensions we have to solve a total of eight local boundary-value problems of the form (33). The idea behind the MsFVM is to express the global pressure as a superposition of these local pressure solutions ϕ_i^k. Thus, inside each interaction region R one assumes that the pressure is a superposition of the local subgrid solutions $\{\phi_i^k\}$, where k ranges over all corner-points in the interaction region (i.e., over the cell-centres of the coarse-grid blocks).

First, we define the boundary conditions ν_i^k in (33). These are defined by solving a reduced-dimensional flow problem on each face F of the interaction region

$$-\nabla \cdot K\lambda \nabla \nu_i^k = 0 \quad \text{in } F, \tag{34}$$

with boundary conditions given by $\nu_i^k(x^l) = \delta_{kl}$ at the corner points of the interaction region. (In 3D, the corner-point values are first extended to the edges of F by linear interpolation). Once ν_i^k are computed, the local pressure solutions ϕ_i^k can be computed from (33).

The next step is to identify basis functions for the multiscale method. To this end, we observe that the cell centers x^k constitute a corner point for
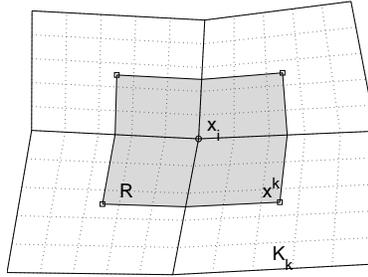
Fig. 10. The shaded region represents the interaction region R for the MsFVM, where x_i denotes corner-points and x^k the midpoints of the coarse grid-blocks. The midpoints x^k are the corner-points of the interaction region.
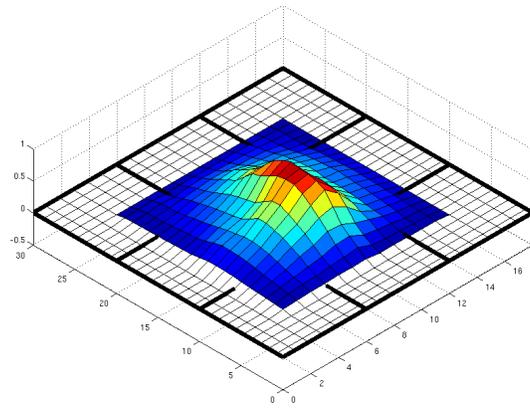
Fig. 11. Pressure basis function ϕ^k for the MsFVM in two-dimensional space.

four interaction regions in 2D and for eight interaction regions in 3D (for a regular hexahedral grid). Moreover, for all corner-points x_i of the coarse grid, the corresponding boundary conditions ν_i^k for the different pressure equations coincide on the respective faces of the interaction regions that share the corner point x^k. This implies that the basis function

$$\phi^k = \sum_i \phi_i^k \tag{35}$$

is continuous (in a discrete sense), see Figure 11. In the following construction, the base functions defined in (35) will serve as building blocks that are used to construct a global "continuous" pressure solution.

   Thus, define now the approximation space $U^{\mathrm{ms}} = \mathrm{span}\{\phi^k\}$ and observe that all basis functions vanish at all but one of the grid block centres x^k. This implies that, given a set of pressure values $\{p^k\}$, there exists a unique extension $\{p^k\} \to p \in U^{\mathrm{ms}}$ with $p(x^k) = p^k$. This extension is defined by

$$p = \sum_k p^k \phi^k = \sum_{i,k} p^k \phi_i^k. \tag{36}$$

A multi-point stencil can now be defined by assembling the flux contribution across the grid-block boundaries from each basis function. Thus, let

$$f_{k,l} = -\int_{\partial B_l} n \cdot K\lambda\nabla\phi^k \, ds$$

be the local flux out of grid-block B_l induced by ϕ^k. The MsFVM for solving (26) then seeks constant grid-block pressures $\{p^k\}$ satisfying

$$\sum_k p^k f_{k,l} = \int_{B_l} \left(q - \nabla \cdot K\lambda_G G \right) dx \qquad \forall l.$$

To reconstruct a mass-conservative velocity field on a fine scale, notice first that the expansion (36) produces a mass-conservative velocity field on the coarse grid. Unfortunately, this velocity field will not preserve mass across the boundaries of the interaction regions. Thus, to obtain a velocity field that is also mass conservative on the fine grid we will use the subgrid fluxes obtained from p as boundary conditions for solving a local flow problem inside each coarse block B_l to *reconstruct* a fine-scale velocity v_l. That is, solve

$$v_l = -K(\lambda\nabla p_l - \lambda_G G), \quad \nabla \cdot v_l = \frac{1}{|B_l|}\int_{B_l} q \, dx \quad \text{in } B_l, \tag{37}$$

with boundary conditions obtained from (36), i.e.,

$$v_l = -K\lambda\nabla p \quad \text{on } \partial B_l, \tag{38}$$

where p is the expanded pressure defined by (36). If these subgrid problems are solved with a conservative scheme, then the global velocity field $v = \sum_{B_l} v_l$ will be mass conservative. Note, however, that since the subgrid problems (37)–(38) are solved independently we loose continuity of the global pressure solution, which is now defined by $p = \sum_{B_l} p_l$.

Remark 1. The present form of the MsFVM, which was developed by Jenny et al. [34], does not model sources at the subgrid scale. Indeed, the source term in (37) is equally distributed within the grid-block. Thus, to use the induced velocity field to simulate the phase transport one has to treat the wells as a uniform source within the entire well block. However, a more detailed representation of flow around wells can be obtained by replacing (37) by

$$v_l = -K(\lambda\nabla p_l - \lambda_G G), \qquad \nabla \cdot v_l = q \quad \text{in } B_l \tag{39}$$

in grid blocks containing a well, i.e., for all B_l in which q is nonzero.

5.3 A Multiscale Mixed Finite-Element Method (MsMFEM)

Recall that mixed finite-element discretisations of elliptic equations on the form (26) seek a solution (p, v) to the mixed equations

$$\int_\Omega u \cdot (K\lambda)^{-1} v \, dx - \int_\Omega p \, \nabla \cdot u \, dx = \int_\Omega \lambda_G G \cdot u \, dx, \qquad (40)$$

$$\int_\Omega l \, \nabla \cdot v \, dx = \int_\Omega q l \, dx, \qquad (41)$$

in a finite-dimensional product space $U \times V \subset L^2(\Omega) \times H_0^{1,\mathrm{div}}(\Omega)$. If the subspaces $U \subset L^2(\Omega)$ and $V \subset H_0^{1,\mathrm{div}}(\Omega)$ are properly balanced (see, e.g., [16, 17, 18]), then p and v are defined (up to an additive constant for p) by requiring that (40)–(41) holds for all $(l, u) \in U \times V$.

In MsMFEMs one constructs a special approximation space for the velocity v that reflects the important subgrid information. For instance, instead of seeking velocities in a simple approximation space spanned by basis functions with polynomial components, one computes special multiscale basis functions Ψ in a manner analogous to the MsFVM, and defines a corresponding multiscale approximation space by $V^{\mathrm{ms}} = \mathrm{span}\{\Psi\}$. The pressure approximation space consists simply of piecewise constant functions on the coarse grid, i.e.,

$$U = \{p \in L^2(\Omega) : p|_B \text{ is constant for all } B \in \mathcal{B}\}.$$

Hence, in the MsMFEM we seek

$$p \in U, \ v \in V^{\mathrm{ms}} \quad \text{such that (40)–(41) holds for} \quad \forall l \in U, \ \forall u \in V^{\mathrm{ms}}. \quad (42)$$

The MsMFEM thus resolves subgrid-scales locally through the construction of special multiscale basis functions, whereas the large scales are resolved by solving the discretised equations on a coarse-grid level.

An approximation space for the pressure p that reflects subgrid structures can be defined in a similar manner. However, whereas velocity fields for flow in porous media may fluctuate rapidly, the pressure is usually relatively smooth. It is therefore often sufficient to model pressure with low resolution as long as it does not significantly degrade the accuracy of the velocity solution. Thus, because the MsMFEM treats the pressure and velocities as separate decoupled variables, it is natural to use a high-resolution space for velocity and a low-resolution space for pressure. In other words, the computational effort can be spent where it is most needed. Moreover, the approximation spaces can not be chosen arbitrarily. Indeed, the convergence theory for mixed finite element methods, the so-called *Ladyshenskaja–Babuška–Brezzi* theory (see [16, 17, 18]) states that the approximation spaces must satisfy a relation called the *inf-sup* condition, or the LBB (*Ladyshenskaja–Babuška–Brezzi*) condition. Using a multiscale approximation space, also for the pressure variable, can cause the LBB condition to be violated.

Exercise: *Show that if the velocity solution v of (17)–(18) is contained in V^{ms}, then the velocity solution of (42) coincides with v.*
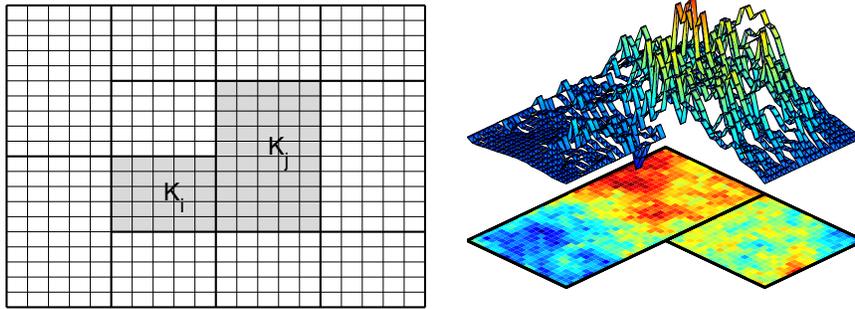
Fig. 12. Left: Schematic of the coarse and fine grid for the MsMFEM. The shaded region denotes the support of the velocity basis function associated with the edge between the two grid-blocks B_i and B_j. Right: x-component of a MsMFEM basis function associated with an interface between two rectangular (two-dimensional) grid-blocks.

Approximation Space for the Darcy Velocity

Consider a coarse grid that overlays a fine (sub)grid, for instance as illustrated in Figure 12. For the velocity we associate one vector of basis functions with each non-degenerate interface Γ_{ij} between two neighbouring grid-blocks B_i and B_j. To be precise, for each interface Γ_{ij} we define a basis function Ψ_{ij} by

$$\Psi_{ij} = -K\nabla\phi_{ij}, \qquad \text{in } B_i \cup B_j, \tag{43}$$

where ϕ_{ij} is determined by

$$(\nabla \cdot \Psi_{ij})|_{B_i} = \ell(x)/\int_{B_i} \ell(x)\,dx, \tag{44}$$

$$(\nabla \cdot \Psi_{ij})|_{B_j} = -\ell(x)/\int_{B_j} \ell(x)\,dx. \tag{45}$$

with no-flow boundary conditions along the edges $\partial B_i \cup \partial B_j \backslash \Gamma_{ij}$.

The function ℓ in (44)–(45) is a positive function that can be defined in various ways. Chen and Hou [22] simply used $\ell(x) = 1$, which produces mass-conservative velocity fields at the coarse-scale level and on the fine scale for all blocks where the source term q is zero. For blocks with nonzero source term q, the fine-scale velocity is not conservative unless q is treated as a constant within each grid block (analogous to the way sources are modelled in the original MsFVM [34]). In reservoir simulation, however, this way of treating sources is inadequate. Indeed, here the source term q represents wells that are point- or line-sources, and modelling flow correctly in the near-well region is considered to be very important. However, since this issue is linked specifically to the reservoir simulation application, we will discuss how ℓ can be defined to handle wells along with other implementational issues in Section 6.

To obtain a mass-conservative velocity field on a subgrid scale we need to solve the subgrid problems (43)–(45) with a mass conservative scheme. Figure 12 displays the x-component of a velocity basis function for the case with $\ell(x) = 1$ computed using the lowest order Raviart–Thomas mixed FEM. We clearly see strong fluctuations in the velocity that reflect the fine-scale heterogeneity. Note also that the basis functions Ψ_{ij} are defined to be time-independent. This implies that the computation of the multiscale basis functions may be made part of a preprocessing step, also for flows with large variations in the total mobility λ. In other words, a single set of basis functions may be used throughout the entire simulation. The reason why it is not necessary to include the total mobility in (43) is that mobility variations within a single block are usually small relative to the jumps in the permeability. Therefore, by including only K we account for the dominant part of the fine-grid variability in the coefficients $K\lambda$. The coarse grid variability of the total mobility is taken into account by reassembling the coarse grid system at each time step.

Remark 2. For the MsFVM one can also use a single set of basis functions throughout entire simulations. However, to account for coarse-grid variability of the total mobility one needs to update the upscaled MsFVM transmissibilities, e.g., by multiplying the initial transmissibilities with a factor that reflects the change in total mobility. This implies that one can not escape from solving the local subproblems (37) or (39) in order to obtain a mass conservative velocity field on the fine grid. This feature generally makes the MsFVM more computationally expensive for multi-phase flows than the MsMFEM.

5.4 Numerical Examples

Both MsMFEM and MsFVM solve a coarse-scale equation globally while trying to resolve fine-scale variations by using special multiscale basis functions. Next, we demonstrate that the accuracy of the generated velocity solutions is not very sensitive to the dimension of the coarse grid.

Example 4 (from [3]). Consider a horizontal, two-dimensional reservoir with 60×220 grid cells with permeability from the bottom layer of Model 2 in the 10th SPE Comparative Solution Project [25]. We inject water in the centre of the domain and produce oil and water at each of the four corners. The pressure equation is solved using the MsFVM and the MsMFEM with various coarse-grid dimensions. For comparison, we also compute two reference solutions using the TPFA scheme, one on the original 60×220 grid, and one on a grid that is refined four times in each direction. Employing the corresponding velocity fields, we solve an equation modelling transport of an incompressible fluid using an upstream finite-volume method on the underlying fine grid.

Figure 13 shows the resulting saturation fields when the total volume of the water that has been injected is equal to 30% of the total accessible pore

Table 3. Relative saturation error $\delta(S)$ for a five-spot simulation in Layer 85 of Model 2 of the 10th SPE Comparative Solution Project for various coarse grids.

	30×110	15×55	10×44	5×11
MsMFEM	1.0916	1.2957	1.6415	1.9177
MsFVM	1.0287	1.6176	2.4224	3.0583

volume. We observe that all saturation plots are quite similar to the saturation plots obtained using the reference velocity fields. We therefore also quantify the errors in the respective saturation fields by

$$\delta(S) = \frac{\epsilon(S)}{\epsilon(S_{ref})}, \qquad \epsilon(S) = \frac{\|S - \mathcal{I}(S_{ref}^{4\times})\|_{L^1}}{\|\mathcal{I}(S_{ref}^{4\times})\|_{L^1}},$$

where $\mathcal{I}$ is an operator that maps the saturation solution on the refined 240×880 grid onto the original 60×220 grid. The results displayed in Table 3 show that there is some degradation of solution quality when the grid is coarsened, but the errors are not very sensitive to coarse-grid size.

When the pressure equation (26) needs to be solved once, the multiscale methods described above can only offer limited speed-up relative to the time spent on solving the full problem on the fine grid using state-of-the-art linear solvers, e.g., algebraic multigrid methods [47]. However, for two-phase flow simulations, where the pressure equation needs to be solved repeatedly, it has been demonstrated that the basis functions need to be computed only once, or updated infrequently [1, 35, 39]. This means that the main computational task is related to solving the global coarse-grid system, which is significantly less expensive than solving the full fine-grid system. This is illustrated by the following example.

Example 5 (from [40]). Consider now the full SPE 10 model, which consists of $60 \times 220 \times 85$ uniform cells. The top 35 layers are from a smooth Tarbert formation, whereas the bottom 50 layers are from a fluvial Upper Ness formation, see Figure 1. The reservoir is produced using a five-spot pattern of vertical wells with an injector in the middle; see [25] for more details.

To simulate the production process we use a streamline simulator with two different pressure solvers: (i) TPFA with an algebraic multigrid linear solver [47], and (ii) MsMFEM on a $5 \times 11 \times 17$ coarse grid. Streamline solvers are known to be very efficient compared to conventional (finite-difference) reservoir simulators, for which computing the full 3D SPE10 model is out of bounds using a single processor and takes several hours on a parallel processor. The key to the high efficiency of streamline solvers is underlying operator splitting used to separate the solution of pressure/velocity from the solution of the fluid transport, which here is solved along 1D streamlines (i.e., in Lagrangian coordinates) and mapped back to the Eulerian grid used to compute pressure and velocities.
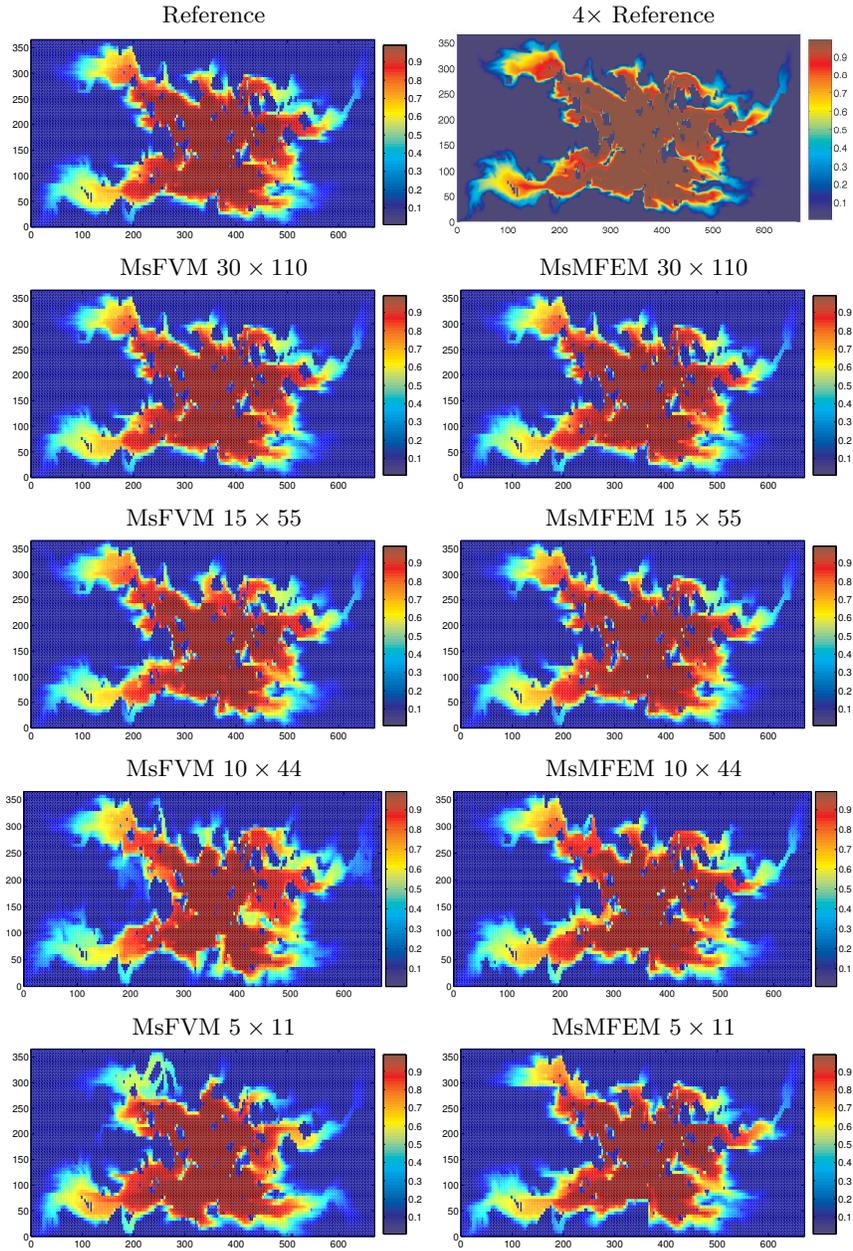
Fig. 13. Saturation solutions computed using velocity fields obtained with MsM-FEM and MsFVM on various coarse grids (c–j), TPFA on the original fine grid (a), and TPFA on the grid that is refined four times in each direction (b).

Table 4. Runtimes for Model 2 of the 10th SPE Comparative Solution Project using a streamline simulator with TPFA or MsMFEM pressure solver measured on a workstation PC with a 2.4 GHz Intel Core 2 Duo processor with 4 Mb cache and 3 Gb memory.

	Pressure	Streamline	Total
TPFA	465 sec	51 sec	516 sec
MsMFEM	91 sec	51 sec	142 sec

Table 4 reports runtimes for two simulations of 2 000 days of production for the whole model. In both runs the simulator used 5 000 streamlines and 25 times steps. The time spent on the transport step includes tracing of stream-lines, solving 1D transport equations, and mapping solutions back and forth between the pressure and the streamline grid. The time spent in the multiscale pressure solver includes initial computation of basis functions and assembly and solution of coarse-grid system for each time step. Using the MsMFEM pressure solver gives a speedup of 5.1 for the pressure solution and 3.6 for the overall computation. Moreover, with a total runtime of 2 minutes and 22 seconds, simulating a million-cell reservoir model has become an (almost) interactive task using the the multiscale–streamline solver.

Remark 3. Note that the basis function can be computed independently, which means that the computation of basis functions is a so-called embarrassingly parallel task. Even further speedup should therefore be expected for parallel implementations, using e.g., the multi-core processors that are becoming available in modern PCs.

6 Implementational Issues for MsMFEM

In this section we discuss some of the implementational issues that need to be addressed when implementing the MsMFEM. We start by discussing what considerations one should take into account when generating the coarse grid. Next we explain how the coarse-grid system can be assembled efficiently, and the implications that this has on the choice of numerical method used for computing the multiscale velocity basis functions. We then discuss the role of the function ℓ in the definition of the basis functions, and how it impacts the MsMFEM solution. Finally, we describe briefly how to build global information into the basis functions to more accurately resolve flow near large-scale heterogeneous structures that have a strong impact on the flow regime.

6.1 Generation of Coarse Grids

It has been demonstrated in [4, 5] that MsMFEM is *very* flexible with respect to the geometry and topology of the coarse grid. A bit simplified, the grid
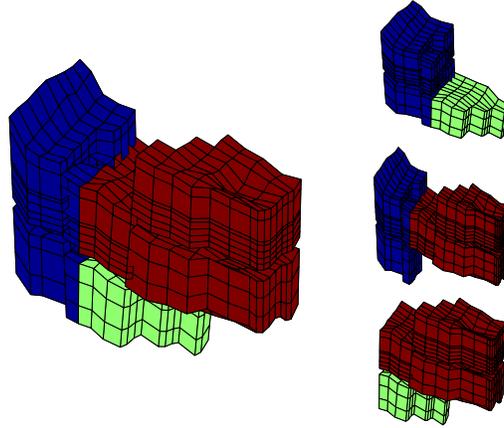
Fig. 14. A three-block domain and the corresponding subdomains constituting the support of the resulting MsMFEM basis functions.

flexibility can be stated as follows: given an appropriate solver for the local flow problems on a particular type of fine grids, the MsMFEM can be formulated on any coarse grid where each grid block consists of an arbitrary collection of connected fine-grid cells. To illustrate, consider a small model where Ω is defined as the union of the three blocks depicted in Figure 14. In the multiscale formulation we construct three basis functions for this set of blocks, one for each pair depicted in Figure 14.

Extensive tests, some of which are reported in [4, 5], show that the accuracy of the MsMFEM is generally not very sensitive to the shape of the blocks. In fact, accurate results are obtained for grids containing blocks with rather 'exotic' shapes, see e.g., [4, 5]. In the next three examples we will show some examples of coarse grids to substantiate this claim. The reader is referred to [4, 5] for a more thorough discussion of the numerical accuracy obtained using this kind of coarse grids.

Example 6 (Near-well grid). Figure 15 shows a vertical well penetrating a structured corner-point grid with eroded layers. On the coarse grid, the well is confined to a single cell consisting of all cells in the fine grid penetrated by the well. Moreover, notice the single neighbouring block shaped like a 'cylinder' with a hole.

Example 7 (Barriers). Figure 16 shows a subsection of the SPE10 model, in which we have inserted a few flow barriers with very low permeability. In [4] it was shown that MsMFEM becomes inaccurate if coarse grid-cells are cut into two (or more) non-communicating parts by a flow barrier. Fortunately, this can be automatically detected when generating basis functions, and the resolution can be improved by using some form of grid refinement. The figure
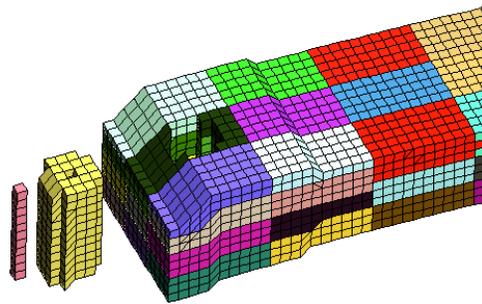
Fig. 15. A coarse grid defined on top of a structured corner-point fine grid. The cells in the coarse grid are given by different colours.
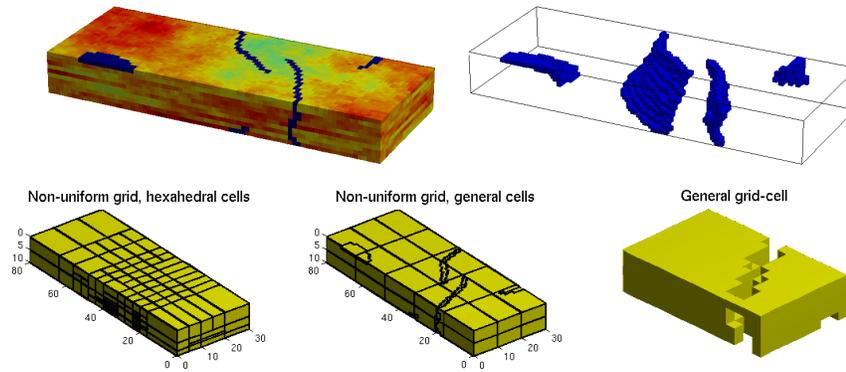
Fig. 16. The upper row shows the permeability field (right), and the interior barriers (left). The lower row shows a hierarchically refined grid (left), the barrier grid (middle), and a coarse grid-block in the barrier grid (right).

shows two different approaches: (i) structured, hierarchical refinement, and (ii) direct incorporation of the flow barriers as extra coarse grid-blocks intersecting a uniform $3 \times 5 \times 2$ grid. This results in rather exotic coarse cells, e.g., as shown in the figure, where the original rectangular cell consisting of $10 \times 16 \times 5$ fine cells is almost split in two by the barrier, and the resulting coarse cell is only connected through a single cell in the fine grid. Although the number of grid cells in the barrier grid is five times less than for the hierarchically refined grid, the errors in the production curves are comparable, indicating that MsMFEM is robust with respect to the shape of the coarse cells.

Example 8 (Eroded layers). Figure 17 shows a uniform partitioning in index space of a corner-point grid modelling a wavy depositional bed on a meter-scale. The corner-point grid is described by vertical pillars that form a uniform 30×30 in the horizontal plane and 100 very thin layers, out of which many
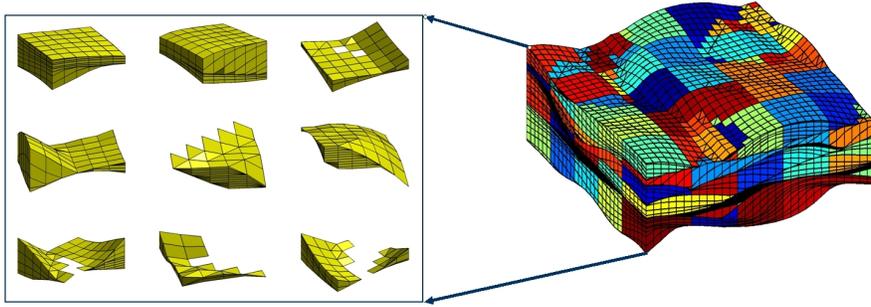
Fig. 17. Uniform partitioning in index space of a corner-point model containing a large number of eroded layers.

collapse to a hyper-plane in some regions. The figure also shows the shape in physical space of some of the coarse blocks resulting from the uniform partitioning in index space. All blocks are used directly in the simulation, except for the block in the lower-right corner, which has two disconnected parts and thus can be split in two automatically.

The complex coarse blocks arising from the straightforward partitioning in index space will in fact give more accurate results than what is obtained from more sophisticated up-gridding schemes trying e.g., to make each cell be as close to a regular hexahedral box as possible. The reason is that the flow will follow the layered structure of the medium and therefore is resolved most accurately by coarse grids that reflect the layering.

The fact that MsMFEM is rather insensitive to the number and the shape of the blocks in the coarse grid means that the process of generating a coarse simulation grid from a complex geological model can be greatly simplified, especially when the fine grid is fully unstructured or has geometrical complications due to faults, throws, and eroded cells; e.g., as seen in Figures 3 and 8. However, MsMFEM does have some limitations, as identified in [4]. Here it was observed that barriers (low-permeable obstacles) may cause inaccurate results unless the coarse grid adapts to the barrier structures. In addition it was demonstrated that MsMFEM in its present form has limited ability to model bidirectional flow across coarse-grid interfaces; fine-grid fluxes at coarse-grid interfaces in the reconstructed flow field will usually go in the same direction.

As a remedy for the limitations identified in [4], it is possible to exploit global information (e.g., from an initial fine-scale pressure solve) when constructing the basis functions [1], see also Section 6.4. However, our experience indicates that accurate results are also obtained if the coarse grid obeys certain guidelines; see the left plot in Figure 18 for illustrations:
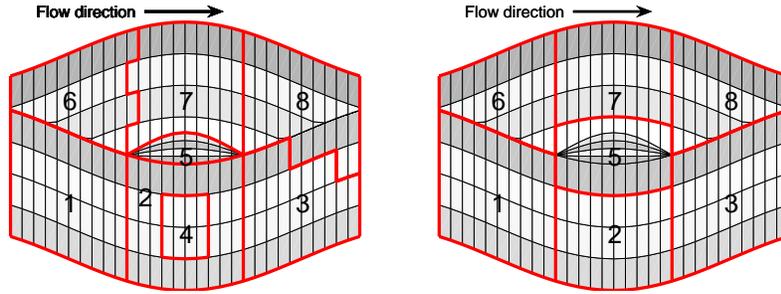
Fig. 18. Illustration of some of the guidelines for choosing a good coarse grid. In the left plot, all blocks except for Block 1 violate at least one of the guidelines each. In the right plot, the blocks have been improved at the expense of more couplings in the coarse-grid system.

1. The coarse grid should preferably minimise the occurrence of bidirectional flow across coarse-grid interfaces. Examples of grid structures that increase the likelihood for bidirectional flow are:
   - Coarse-grid faces with (highly) irregular shapes, like the 'saw-tooth' faces between Blocks 6 and 7 and Blocks 3 and 8.
   - Blocks that do not contain source terms and have only one neighbour, like Block 4. (A simple remedy for this is to split the interface into at least two sub-faces, and define a basis function for each sub-face.)
   - Blocks having interfaces only along and not transverse to the major flow directions, like Block 5. (To represent flow in a certain direction, there must be at least one non-tangential face that defines a basis function in the given flow direction.)
2. Blocks and faces in the coarse grid should follow geological layers whenever possible. This is not fulfilled for Blocks 3 and 8.
3. Blocks in the coarse-grid should adapt to flow obstacles (shale barriers, etc.) whenever possible; see [4].
4. For parabolic (compressible flow) problems, e.g., three-phase black-oil models, one should model point-sources (and line-sources) at the subgrid level. For instance, for reservoir simulation one should assign a separate grid block to each cell in the original grid with an open well perforation[1].

---

[1]For reservoir simulation there is also another reason, apart from compressibility, to why it is preferable to assign separate blocks to each cell with an open well perforation. Indeed, the source q in reservoir simulation models is generally not known a priori, but determined by so-called well-models that relate the well-rates to the pressure in the associated well-block. To compute the rates "correctly" one needs to get the pressure in the well-block correct. The MsMFEM provides a pressure value for each coarse grid-block. Thus, by assigning a block to each cell with an open well perforation, we extract values that represent the actual pressure in these cells. In other words, the pressure at the wells is modelled with subgrid resolution.

In addition, to enhance the efficiency of the method, one should try to keep the number of connections between coarse-grid blocks as low as possible to minimise the bandwidth of the coarse-scale system, and avoid having too many small blocks as this increases the dimension of the coarse-scale system, but does not necessarily improve accuracy significantly.

In the right plot of Figure 18, we have used the guidelines above to improve the coarse grid from the left plot. In particular, we joined Blocks 2 and 4 and have have increased the size of Block 5 to homogenise the block volumes and introduce basis functions in the major flow direction for this block. In doing so, we increase the number of couplings from nine to twelve (by removing the coupling between Blocks 2 and 4 and introducing extra coupling among Blocks 1, 3, 5, 6, and 8). In general it may be difficult to obtain an 'optimal' coarse grid, since guidelines may be in conflict with each other. On the other hand, this is seldom necessary, since the MsMFEM is relatively robust with respect to the choice of coarse grid.

6.2 Computing Basis Functions and Assembling the Linear System

In principle, any conservative numerical method may be used to construct the basis functions, e.g., any of the four methods discussed in Section 2.1. However, computing the entries in the coarse-grid linear system requires evaluating the following inner-products between the multiscale basis functions:

$$\int_{\Omega} \Psi_{ij} \cdot (K\lambda)^{-1} \Psi_{kl} \, dx. \tag{46}$$

Alternatively, one can use an approximate inner product like the one used in the mimetic formulation discussed in Section 3.4.

If a finite-volume method is used, a computational routine for computing these inner-products, either exactly or approximately, is generally not available. Thus, to implement the MsMFEM one needs to add an extra feature in the numerical implementation. When a mixed FEM or mimetic FDM is used, on the other hand, a routine for calculating the inner-product (46) is part of the implementation of the subgrid solver. In fact, in this case the integral (46) can be expressed as a vector-matrix-vector product.

Let $\mathbf{R}$ be the matrix formed with columns $\mathbf{r}_{ij}$ holding the coefficients r_{kl}^{ij} in the following expansion:

$$\Psi_{ij} = \sum_{\gamma_{kl}} r_{kl}^{ij} \psi_{kl}.$$

Furthermore, let $\mathbf{B}$ be the $\mathbf{B}$-matrix in a system of the form (20) that stems from a Raviart–Thomas mixed FEM or a mimetic FDM on a fine grid. Then

$$\int_{\Omega} \Psi_{ij} \cdot (K\lambda)^{-1} \Psi_{kl} \, dx = \mathbf{r}_{ij}^{t} \mathbf{B} \mathbf{r}_{ij}. \tag{47}$$

Thus, the coarse-grid system for the MsMFEM may be expressed as follows:

$$\mathbf{B}^{\mathrm{ms}} = \mathbf{R}^t \mathbf{B} \mathbf{R}, \qquad \mathbf{g}^{\mathrm{ms}} = \mathbf{R}^t \mathbf{g}.$$

The right hand side $\mathbf{q}^{\mathrm{ms}}$ in the multiscale system is formed by integrating q over each grid block, and the matrix $\mathbf{C}^{\mathrm{ms}} = [c_{m,kl}]$ is given by

$$c_{m,kl} = \int_{B_m} \nabla \cdot \Psi_{kl} \, dx = \begin{cases} 1, & \text{if } k = m, \\ -1, & \text{if } l = m, \\ 0, & \text{otherwise.} \end{cases}$$

6.3 Role of the Weighting Function

The weighting function ℓ in (44)–(45) has been defined in different ways

- $\ell = 1$ in [22];
- $\ell = q$ if $\int_{B_m} q \neq 0$ and $\ell = 1$ elsewhere in [1]; and
- $\ell = q$ if $\int_{B_m} q \neq 0$ and $\ell = \mathrm{trace}(K)$ elsewhere in [4, 5].

To understand how these definitions have come into play, recall first that the MsMFEM velocity solution is a linear superposition of the velocity basis functions. Hence,

$$(\nabla \cdot v)|_{B_i} = \sum_j v_{ij} \nabla \cdot \Psi_{ij} = \frac{\ell}{\int_{B_i} \ell \, dx} \sum_j v_{ij}$$

$$= \frac{\ell}{\int_{B_i} \ell \, dx} \int_{\partial B_i} v \cdot n \, ds = \frac{\ell}{\int_{B_i} \ell \, dx} \int_{B_i} \nabla \cdot v \, dx.$$

One can therefore say that the primary role of ℓ is to distribute the divergence of the velocity field onto the fine grid in an appropriate way.

For incompressible flow problems $\mathrm{div}(v)$ is non-zero only in blocks with a source. For blocks where $\int_{B_i} q \neq 0$, the choice $\ell = q$ stems from the fact that it gives mass conservative velocity fields on the subgrid. For blocks without a source (where the velocity is divergence free) ℓ can be chosen nearly arbitrarily. The idea of letting the weight function scale with the trace of the mobility was introduced in [4] as a way of avoiding unnaturally large amount of flow through low-permeable zones and in particular through flow barriers. In general, however, using $\ell = 1$ gives (almost) equally accurate results.

For compressible flow (e.g., (8)) we may no longer choose ℓ arbitrarily. For instance, defining base functions using $\ell = q$ would concentrate all compressibility effects where q is nonzero. To avoid this, one has to separate the contribution to the divergence field stemming from sources and from compressibility. This can be achieved, as we have proposed in Section 6.1, by assigning one "coarse" grid block to each cell in the fine grid with a source or sink. By doing so, we may, in principle, choose $\ell = 1$ everywhere. But, for the three-phase black-oil model (cf. Section 2.2), we have

$$\nabla \cdot v = q - c_t \frac{\partial p}{dt} - \sum_j c_j v_j \cdot \nabla p_l. \qquad (48)$$

Hence, ℓ should ideally be proportional to the right hand side of (48). Although the right hand side of (48) can be estimated from local computations, we do not propose using this strategy to define ℓ. Indeed, the multiscale concept is not to try to replicate fine-scale solutions by trying to account for all subgrid information. The important thing is to account for the subgrid effects that strongly influence flow on the coarse-grid level, and subgrid variability in the velocity divergence field is generally not among these effects.

Our own numerical experience so far indicates that good accuracy is obtained by taking ℓ to be the porosity ϕ. To motivate this choice, we note that c_t is proportional to ϕ when the saturations are smooth. Moreover, using $\ell = \phi$ is in accordance with the idea behind using $\ell = \text{trace}(\lambda)$. Indeed, regions with very low permeability also tend to have low porosity, so by choosing $\ell = \phi$ one should (to some extent) avoid forcing too much flow through low-permeable barriers, [4]. Using $\ell = \text{trace}(K)$, on the other hand, will generally give velocity solutions for which $\text{div}(v)$ oscillates too much, i.e., is underestimated in low-permeable regions and overestimated in high-permeable regions.

6.4 Incorporating Global Information

All multiscale methods essentially attempt to decouple the global problem into a coarse-grid system and a set of independent local problems. In Section 5.1 it was shown that in the one-dimensional case there is an exact splitting. That is, the global solution (of the variational formulation) can be expressed as the sum of the MsFEM solution and solutions of independent local problems. In higher dimensions, however, decoupling the system into a low-dimensional coarse-grid system and independent local subproblems is not possible in general. But it is possible to invoke global information, e.g., from a single-phase flow solution computed at initial time, to specify better boundary conditions for the local flow problems and thereby improve the multiscale solutions, as was shown in [1] for MsMFEM and in [29] for MsFVM.

For many problems, invoking global information may have little effect, and will, for multi-phase flow problems, only give an incremental improvement in accuracy. But for certain problems, such as for models with large scale near-impermeable shale barriers that force the flow to take a detour around the barrier, invoking global information can improve accuracy quite significantly, and should be viewed as an alternative to grid refinement.

Since MsMFEM allows running entire simulations with a single set of basis functions, solving the pressure equation once on a fine grid in order to improve the accuracy of the multiscale solution is easily justified. To this end, one needs to split each of the subgrid problems (43)–(45) into two independent problems in B_i and B_j, respectively, with a common Neumann boundary condition on

the interface Γ_{ij}. In particular, if v is the initial fine-scale velocity solution, the following boundary condition should be imposed on Γ_{ij}:

$$\Psi_{ij} \cdot n_{ij} = \frac{v \cdot n_{ij}}{\int_{\Gamma_{ij}} v \cdot n_{ij}\, ds}. \tag{49}$$

The method that stems from defining the multiscale basis functions with this formulation is usually referred to as the global, as opposed to local, MsMFEM.

Exercise: *Assign one grid block to each cell with a source and let $\ell = 1$. Alternatively let $\ell = q$ if $\int_{B_i} q \neq 0$ and $\ell = 1$ elsewhere. Show that if the multiscale basis functions are defined by (43)–(45) and (49), then $v \in \mathrm{span}\{\Psi_{ij}\}$.*

References

1. J. E. Aarnes. On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation. *Multiscale Model. Simul.*, 2(3):421–439, 2004.
2. 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 Optimization: Industrial Mathematics at SINTEF*, pages 265–306. Springer Verlag, 2007.
3. J. E. Aarnes, V. Kippe, K.-A. Lie, and A. Rustad. Modelling of multiscale structures in flow simulations for petroleum reservoirs. In G. Hasle, K.-A. Lie, and E. Quak, editors, *Geometrical Modeling, Numerical Simulation, and Optimization: Industrial Mathematics at SINTEF*, pages 307–360. Springer Verlag, 2007.
4. J. E. Aarnes, S. Krogstad, and K.-A. Lie. A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids. *Multiscale Model. Simul.*, 5(2):337–363, 2006.
5. J. E. Aarnes, S. Krogstad, and K.-A. Lie. Multiscale mixed/mimetic methods on corner-point grids. *Comput. Geosci*, 12(3):297–315, 2008.
6. I. Aavatsmark, T. Barkve, Ø. Bøe, and T. Mannseth. Discretization on unstructured grids for inhomogeneous, anisotropic media. Part I: Derivation of the methods. *SIAM J. Sci. Comp.*, 19(5):1700–1716, 1998.
7. I. Aavatsmark, T. Barkve, Ø. Bøe, and T. Mannseth. Discretization on unstructured grids for inhomogeneous, anisotropic media. Part II: Discussion and numerical results. *SIAM J. Sci. Comp.*, 19(5):1717–1736, 1998.
8. T. Arbogast. Numerical subgrid upscaling of two-phase flow in porous media. In Z. Chen, R. E. Ewing, and Z.-C. Shi, editors, *Numerical Treatment of Multiphase Flows in Porous Media (Beijing, 1999)*, Lecture Notes in Phys., pages 35–49. Springer-Verlag, Berlin, 2000.
9. T. Arbogast. Analysis of a two-scale, locally conservative subgrid upscaling for elliptic problems. *SIAM J. Numer. Anal.*, 42(2):576–598, 2004.
10. K. Aziz and A. Settari. *Petroleum reservoir simulation.* Elsevier, London and New York, 1979.
11. I. Babuška, G. Caloz, and E. Osborn. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM J. Numer. Anal.*, 31:945–981, 1994.

12. I. Babuška and E. Osborn. Generalized finite element methods: Their performance and their relation to mixed methods. *SIAM J. Numer. Anal.*, 20:510–536, 1983.
13. J. W. Barker and S. Thibeau. A critical review of the use of pseudorelative permeabilities for upscaling. *SPE Reservoir Eng.*, 12:138–143, 1997.
14. S. H. Begg, R. R. Carter, and P. Dranfield. Assigning effective values to simulator gridblock parameters for heterogeneous reservoirs. *SPE Reservoir Eng.*, pages 455–463, 1989.
15. A. Benesoussan, J.-L. Lions, and G. Papanicolaou. *Asymptotic Analysis for Periodic Structures.* Elsevier Science Publishers, Amsterdam, 1978.
16. D. Braess. *Finite elements: Theory fast solvers and applications in solid mechanics.* Cambridge University Press, Cambridge, 1997.
17. S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods.* Springer–Verlag, New York, 1994.
18. F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods.* Computational Mathematics. Springer–Verlag, New York, 1991.
19. F. Brezzi, K. Lipnikov, M. Shashkov, and V. Simoncini. A new discretization methodology for diffusion problems on generalized polyhedral meshes. *Comput. Methods Appl. Mech. Engrg.*, 196(37-40):3682–3692, 2007.
20. F. Brezzi, K. Lipnikov, and V. Simoncini. A family of mimetic finite difference methods on polygonial and polyhedral meshes. *Math. Models Methods Appl. Sci.*, 15:1533–1553, 2005.
21. G. Chavent and J. Jaffre. *Mathematical models and finite elements for reservoir simulation.* North Holland, 1982.
22. Z. Chen and T. Y. Hou. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Math. Comp.*, 72:541–576, 2003.
23. Z. Chen, G. Huan, and Y. Ma. *Computational methods for multiphase flows in porous media.* Computational Science & Engineering. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.
24. M. A. Christie. Upscaling for reservoir simulation. *J. Pet. Tech.*, 48:1004–1010, 1996.
25. M. A. Christie and M. J. Blunt. Tenth SPE comparative solution project: A comparison of upscaling techniques. *SPE Reserv. Eval. Eng*, 4(4):308–317, 2001. url: `http://www.spe.org/csp`.
26. L. P. Dake. *Fundamentals of reservoir engineering.* Elsevier, Amsterdam, 1978.
27. A. H. Demond and P. V. Roberts. An examination of relative permeability relations for two-phase flow in porous media. *Water Res. Bull.*, 23:617–628, 1987.
28. L. J. Durlofsky. Numerical calculations of equivalent gridblock permeability tensors for heterogeneous porous media. *Water Resour. Res.*, 27(5):699–708, 1991.
29. Y. Efendiev, V. Ginting, T. Hou, and R. Ewing. Accurate multiscale finite element methods for two-phase flow simulations. *J. Comput. Phys.*, 220(1):155–174, 2006.
30. R. E. Ewing. *The mathematics of reservoir simulation.* SIAM, 1983.
31. L. Holden and B. Nielsen. Global upscaling of permeability in heterogeneous reservoirs; the output least squares (ols) method. *Transp. Porous Media*, 40(2):115–143, 2000.
32. U. Hornung. *Homogenization and porous media.* Springer Verlag, New York, 1997.

33. 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.
34. 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.
35. P. Jenny, S. H. Lee, and H. A. Tchelepi. Adaptive multiscale finite-volume method for multiphase flow and transport in porous media. *Multiscale Model. Simul.*, 3(1):50–64, 2004/05.
36. V. V. Jikov, S. M. Kozlov, and O. A. Oleinik. *Homogenization of differential operators and integral functionals*. Springer–Verlag, New York, 1994.
37. A. G. Journel, C. V. Deutsch, and A. J. Desbarats. Power averaging for block effective permeability. In *SPE California Regional Meeting*, Oakland, California, 2-4 April 1986. SPE 15128.
38. M. J. King, D. G. MacDonald, S. P. Todd, and H. Leung. Application of novel upscaling approaches to the Magnus and Andrew reservoirs. In *SPE European Petroleum Conference*, The Hague, Netherlands, 20-22 October 1998. SPE 50463.
39. 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.
40. V. Kippe, H. Hægland, and K.-A. Lie. A method to improve the mass-balance in streamline methods. In *SPE Reservoir Simulation Symposium*, Houston, Texas U.S.A., February 26-28 2007. SPE 106250.
41. L. Lake. *Enhanced oil recovery*. Prentice Hall, Inglewood Cliffs, NJ, 1989.
42. B. B. Maini and T. Okazawa. Effects of temperature on heavy oil-water relative permeability. *J. Can. Petr. Tech*, 26:33–41, 1987.
43. D. W. Peaceman. *Fundamentals of numerical reservoir simulation*. Elsevier, Amsterdam, 1977.
44. P. A. Raviart and J. M. Thomas. A mixed finite element method for second order elliptic equations. In I. Galligani and E. Magenes, editors, *Mathematical Aspects of Finite Element Methods*, pages 292–315. Springer–Verlag, Berlin – Heidelberg – New York, 1977.
45. P. Renard and G. de Marsily. Calculating equivalent permeability. *Adv. Water Resour.*, 20:253–278, 1997.
46. D. P. Schrag. Preparing to capture carbon. *Science*, 315:812–813, 2007.
47. K. Stüben. *Multigrid*, chapter Algebraic Multigrid (AMG): An Introduction with Applications. Academic Press, 2000.
48. X.-H. Wen and J. J. Gómez-Hernández. Upscaling hydraulic conductivities in heterogeneous media: An overview. *J. Hydrol.*, 183:ix–xxxii, 1996.
49. O. Wiener. *Abhandlungen der Matematisch*. PhD thesis, Physischen Klasse der Königlichen Sächsischen Gesellscaft der Wissenschaften, 1912.