Robust Multiscale Control-volume Methods for Reservoir Simulation

PhD Thesis

Andreas Sandvin

Department of Mathematics
University of Bergen

January, 2012

Preface

This dissertation is submitted as a partial fulfilment of the requirements for the degree Doctor of Philosophy (PhD) at the University of Bergen. The work has been conducted at the Centre for Integrated Petroleum Research (CIPR) in collaboration with the Department of Mathematics at the University of Bergen.

The dissertation is organised in three parts. In Part I we present the general and theoretical background for the scientific studies that we have performed. The results of these studies have been formalised in three scientific papers. These are included in Part II. Finally, in Part III we provide supplementary material to the scientific papers.

Acknowledgements

The project has been funded by Statoil. I am grateful for their financial support, which has given me the opportunity of writing this thesis. My main supervisors have been professor Ivar Aavatsmark and professor Jan Martin Nordbotten. I would like to thank them both for valuable help and guidance throughout this project. My co-supervisors has been Edel Reiso and later Eirik Keilegavlen. I would specially like to thank Eirik, as my friend during my studies, a colleague at CIPR and finally my co-supervisor at this project. Your support has been extremely appreciated, in particular at the final stages of the work on this thesis.

I would also like to thank all my colleagues and friends during my research at CIPR. I have been most fortunate to conduct my work at CIPR. I want to thank my family, which has been very supportive and encouraged me during the whole PhD period.

There has also been many frustrating hours and days, when there has been little or no progression in the thesis, but luckily life does not only consist of research. I thank my lovely wife for lightning up these days. Your love and patience has been absolutely vital for the realization of this thesis.

Andreas Sandvin,
Bergen, January 2012

Contents

Part I

Background Material

Chapter 1

Introduction

Understanding the scaling mechanisms on how to map information between scales, is important in many areas of research. In climate research, scientists try to answer questions like: What is the mean temperature in the world? To what extent will a possible change in global temperature affect the climate locally? In economy, understanding how a small perturbation in the economy somewhere in the world inflict the economy at other places in the world, might help to avoid future financial crises. As for reservoir engineering, understanding the fundamental scaling mechanisms within a porous media, is crucial in order to develop accurate reservoir models and efficient and robust reservoir simulators.

The purpose of this chapter is to provide the reader with some basic understanding of petroleum reservoir engineering, and the scales that are involved. We start by giving a brief introduction to what a petroleum reservoir is, how to characterise it, and the main production techniques for extracting hydrocarbons from the subsurface. The study of this thesis is devoted to multiscale techniques for improved reservoir simulations. We will highlight some of the current challenges within reservoir simulations, and discuss what we see as the challenges for the future. Finally, we will give an outline to the rest of the manuscript.

1.1 Reservoir engineering

A petroleum reservoir is a porous medium where hydrocarbons, such as oil and gas, has accumulated and is being trapped. Such geological formations have been formed over millions of years, as compressed layers of sediments. Over time, the reservoir may have experienced erosion, tilting, bending, cracking and even breaking. As a result, the geometry and geology

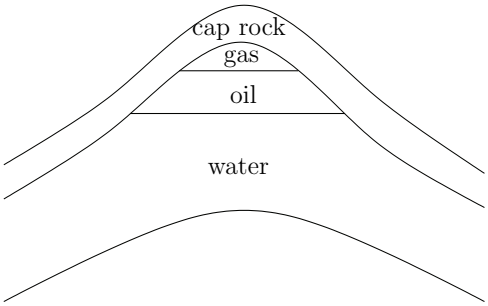


Figure 1.1: Petroleum reservoir

of petroleum reservoirs can be highly complex and highly heterogeneous. An important property of the petroleum reservoir is the trapping mechanism, which prevent the lighter fluids, like oil and gas, from migrating further upwards. Typically, the top layer of the reservoir forms a caprock, which act as a seal. The simplified sketch in Figure 1.1 shows a typical distribution of fluids within a petroleum reservoir, where the lighter gaseous phase is formed on top of the oil-phase, which again is above the water-phase.

1.1.1 Reservoir characterisation and scales

In general, many rock types have the potential of becoming a petroleum reservoir. The only requirement, in addition to the trap, is that the rock must contain a network of interconnected pores, where fluids can flow. In practise, there are two rock types in which the major hydrocarbon reserves have been found, those are sandstone and carbonate rock formations. The different rock types are characterised in terms of their petrophysical rock properties. In the following we will discuss two of the main rock properties, the porosity and permeability.

Porosity

The porosity of the rock is defined as the fraction of pore volume per bulk volume,

$$\text{Porosity} = \frac{\text{Pore volume}}{\text{Bulk volume}}, \tag{1.1}$$

and is a measure of the potential storage capacity of the reservoir. We can use the porosity to estimate the amount of hydrocarbons in the reservoir.

In general, the porosity can be determined by the use of either visual measurements, e.g NMR and micro-scan or by the use of lab-experiments. By injecting oil into a porous core sample, we can experimentally determine the effective porosity, i.e., the amount of pore space accessible for the oil. This is the most widely used technique for determining the porosity. The porosity is an averaged quantity, and will in general vary with the size of bulk volume (see Equation (1.1)). If the entire bulk volume is inside a single pore, the porosity is 1. On the pore scale, ($\sim 10^{-6}m$), we will in general experience large spatial variations in the measurements of the porosity. As we increase the bulk volume, the spatial variations of our measurements will decrease. At the scale, where all local measurements give more or less the same porosity, we say that we have a representative elementary volume (REV) for measuring the porosity. This is illustrated in Figure 1.2. If the reservoir is perfectly homogeneous, there will be only one REV for the entire reservoir. Unfortunately, such reservoirs do not exist in nature.

Permeability

The permeability of the reservoir is a measure for the fluid conductivity of the reservoir rock. This parameter is more complex and involves both the size, geometry and interconnectivity of the porous network. Much research has been devoted to finding correlations between the permeability and the porosity, the pore-sizes or the pore-geometry. In general, the permeability will increase with increasing porosity and pore-size of the reservoir. Similar to porosity, the permeability can be measured experimentally on the lab, however, the notion of an REV for the permeability is less understood. The

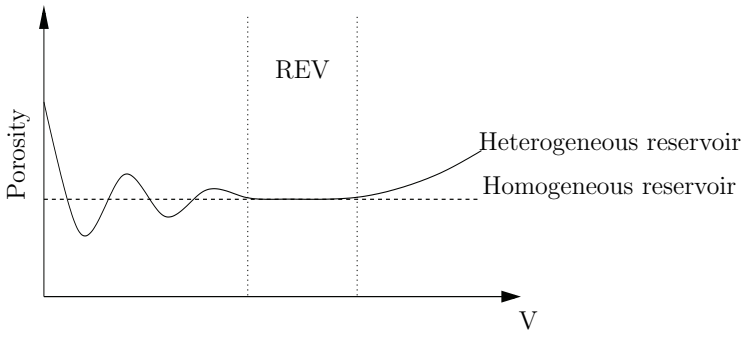


Figure 1.2: Representative elementary volume for the porosity

proper definition and quantification of the permeability on different scales will be discussed in more detail in the next two chapters.

In general, all reservoirs are heterogeneous with a range of scales, both for the porosity and the permeability. However, some general trends can be identified. Sandstone reservoirs tend to have a more narrow pore-size distribution, and a characteristic pore size and porosity can be detected. Carbonate reservoirs are usually more difficult to characterise, with several characteristic pore-size distributions on different scales. The pore size distribution for a sandstone reservoir is typically measured on the scale of micro (10^{-6}) meters. For carbonates, it is not uncommon to additionally find pores on the size of centi (10^{-2}) meters (vugs), or even meters (cavernous pores). Such large-scale pores can have a major impact on the production of the hydrocarbons. Fractures on the scale of centimetres and faults on the scale of meters are other large scale heterogeneities which can be found in any type of natural reservoirs. They have a strong effect on the permeability of the reservoir.

1.1.2 Geological model

Many of today's oil and gas reserves are located deep in the subsurface, not easily accessible. To develop such petroleum reservoirs for production is expensive, thus, it is important to get a good overview of the economical feasibility and risks involved. An important tool in that process, is the construction of a geological model, describing the static state of the reservoir.

Building a high-resolution geological model, consists of several steps. In the first step we want to honour the geometry of the main geological features, like the different layers, zones and faults of the reservoir. This preliminary model is often called the structural model. The fluid flow will mainly follow the geological layering of the reservoir, which can be detected by seismic imaging of the subsurface. The oil and gas can further be situated in different zones of the reservoir divided by impermeable layers of e.g. clay. Each zone can again consist of several geological layers. The distribution of the oil and gas zones, together with the flow properties within the zones and connection between the zones, will to a large extent determine the proper choice of production strategy. Finally, the faults represent a great uncertainty, in which it is difficult to predict their impact on fluid flow. A fault may act as both a trapping mechanism and a conductor of fluid flow. Because of their complex geometry, the faults are often grossly simplified in the geological model.

A (stratigraphical) flow grid is constructed to represent the principle directions of the fluid flow. The grid is usually optimized in terms of smooth-

ness, however, the complex geological features may put constraints on the development of the grid.

Finally, a full high-resolution geological model contains a description of all the petrophysical rock properties within each of the fine-grid cells. Because of the complex geometry and high spatial variability of the rock properties, this geological description can be rather detailed, and the flow grid might have $(10^7 - 10^8)$ grid cells [39].

1.1.3 Petroleum recovery

The recovery process consists of drilling several production and/or injection wells into the subsurface. Based on the pressure and temperature in the reservoir, together with the reservoir rock and fluid properties, several production techniques are possible. The objective is to recover as much of the hydrocarbons as possible, at as low cost as possible, i.e. minimizing the number of wells. The recovery techniques are usually divided into three main stages; primary recovery, secondary recovery and tertiary or enhanced recovery.

The pressure within the reservoir can be several hundred times larger then the pressure on the surface. Thus, the release of potential energy, related to the compression of hydrocarbons are often large enough to extract up to 25-30 percent of the oil an gas reserves. The actual recovery rate will depend on the size of the pressure compartment and the placement of the fluids. As an example, a gas cap above the oil cap, in the near-well region, will expand due to drop in pressure, and can be used as a pressure support to increase the recovery of oil. This early stage of production is called *primary* recovery, and is the simplest, most used and most inexpensive production type.

To further increase the recovery of oil and gas and maintain a stable production rate, it is necessary to provide an external pressure support. This stage is called the *secondary* recovery. The most common approach is to inject water, which displaces the hydrocarbons in the direction of the production well. An important issue is that the injected fluid should not be miscible with the displaced fluid. Moreover, the difference in fluid properties of the two fluids is important. If the pressure decreases to a value below the bubble point for the oil, gas is released from the oil-phase. In some situations it is practical to first recover the oil-phase, before producing the gas. Thus, we can re-inject the gas, temporarily storing it and at the same time use it as a pressure support. In the later years, the problem of increased emissions of CO_2 to the atmosphere has received much attention. Some of today's production fields also involves capturing and storage of CO_2. The CO_2 can either be used as a pressure support, or injected into a depleted reservoir or deep-water aquifer.

After the secondary recovery stage, as much as 40-50 percent of the oil and gas reserves is usually left behind. The reason for this can be many. Basically, the fluids form preferential flow paths, and the displacement of hydrocarbons only covers parts of the reservoir. In order to recover more of the residual and immobile oil, enhanced recovery techniques need to be applied. All techniques aimed at improving the sweep efficiency and recovery of hydrocarbons, beyond the traditional displacement of oil and gas belongs to the *tertiary* or *enhanced* recovery stage. These techniques include injection of chemical and biological species, to change the viscosity or surface tension of the oil or the injected water. Many of the applied techniques are costly, and the effects are not well documented. However, a successful technique may increase the recovery substantially.

1.2 Multiscale modelling and simulation

Over the last couple of decades much of the easy accessible oil and gas have been produced, many existing oil fields are mature and secondary and enhanced production techniques are applied to extend the production-life of these reservoirs. Complex and highly heterogeneous reservoirs, which earlier were not developed, are now being opened for production. An increased understanding of the flow in such formations (e.g. faulted and fractured reservoirs) is necessary in order to make the production from these reservoirs economically feasible.

Much work has focused on including more physics into the reservoir model, e.g. dual-permeability models for fractured reservoirs [77] and the unified earth model for including complex fault structures [67]. Monitoring techniques have been developed or improved, i.e. gravimetric monitoring of water displacing gas [33] and 4D time-lapse seismic for measuring the oil-water contact [18]. Together, the improvements in scientific research and technology give us a more detailed picture of the geological structures of the reservoir and the fluid contact movements during production. Moreover, the computers are getting faster; in which case we can more often than before simulate on the full geological model. As a result, we can put more information of the reservoir into our reservoir models and run full-field simulations.

In order to take advantage of the detailed description of the reservoir, it is crucial that we have robust computational methods, which can handle difficult geometries and heterogeneities, and do not add more uncertainty to the already uncertain process of flow and transport in porous media. Traditionally, the fine-scale properties of the reservoir are upscaled to a suitable resolution for applying conventional reservoir simulators. However, over the

last 10-15 years, there has been a growing interest in multiscale methods. These are computational methods for flow in porous media, which aim at capturing heterogeneities on several scales into the simulations of the geological model at comparable computer efficiency. Upscaling is now an integrated part of the direct simulation on the geological model. This gives us the possibility of controlling the uncertainties of our coarse scale simulations.

The scope of this thesis is to study the numerical framework of the multiscale methods for applications towards reservoir simulation. We want to analyse the mathematical properties of the numerical multiscale algorithms and develop robust numerical techniques which can be applied to problems involving realistic porous media.

1.3 Outline of the thesis

In the following chapters we will describe various strategies for building the simulation model and conducting robust and efficient reservoir simulations. The mathematical and numerical models of the reservoir simulator will be described in Chapter 2. Traditional methods for solving the governing equations will also be explained, and numerical challenges are addressed. In Chapter 3 we describe conventional techniques for upscaling petrophysical properties of the reservoir from the geological model to be included in the simulation model. Here, we will mainly focus on the upscaling of single-phase parameters. Chapter 4 describes the class of domain decomposition techniques, which is commonly used to solve the large system of linear equations arising from the discretisation of the governing flow equations. The multiscale methods can be viewed as a combination of upscaling and domain decomposition, and is described in Chapter 5. Here we will show how the upscaling step can be integrated, so as to efficiently simulate on the fine-scale geological model. We summarize the papers in Chapter 6, before concluding this thesis.

Chapter 2

Reservoir Simulation

Reservoir simulation is an important tool in order to understand the flow and transport of fluids in a porous medium, and in order to predict future developments of the reservoir. The simulations can be used to optimise the placement of wells, quantify uncertainties and foresee future production scenarios of the reservoir.

In this chapter we will describe the reservoir simulator; its mathematical models for flow and transport in porous media and the various computational techniques to solve the equations. Our main focus will be on the numerical aspects of the reservoir simulator, like the various discretisation techniques to represent the mathematical equations and the different numerical solvers to efficiently carry out the simulations. For more information about reservoir modelling and simulation, see e.g. [12, 13, 22, 76].

2.1 Mathematical models for flow in porous media

In the previous chapter we briefly introduced the physical setting of a petroleum reservoir, a couple of typical production scenarios and some typical scales of the reservoir. In this section we will develop mathematical models which describe the flow and displacement of fluids on the reservoir scale ($10^1 - 10^4$ m). In principle we could describe all displacement processes on the pore scale, but then we are limited by computational power to run simulations on the scale of millimetres and centimetres. In addition, the geometry and fluid distributions on the pore scale are not known to us. Instead, we will consider all our equations on the continuum level, where the petrophysical parameters are defined and measured over some average volumes (REVs).

The differential equations for flow and transport are governed by conservation laws, such as conservation of mass, momentum and energy. In the following we describe some of the mathematical models which are being used for reservoir simulation today.

2.1.1 Single-phase flow

The single-phase model is an important test model, which is extensively used to validate the performance of computational methods for flow in porous media. The practical importance of this model is perhaps limited as the petroleum reservoir contains several chemical species of several phases. However, a possible application could be the case of primary recovery of oil, when the reservoir pressure is higher than the bubble point pressure for oil and the influence of the water-phase is negligible small. In this case, it can be practical to approximate a single-phase system of equations.

The single-phase flow in porous media is driven by pressure gradients and governed by the Darcy's law, which is a deterministic model for the reservoir. The phase-velocity $\mathbf{v}$ is written as

$$\mathbf{v} = -\frac{\mathbf{K}}{\mu}\left(\nabla p - \rho g \nabla z\right), \tag{2.1}$$

where $\mathbf{K}$ is the *permeability*, μ is the phase *viscosity*, p is the pressure, ρ is the phase density, g is the gravitational acceleration and z is the vertical coordinate. The Darcy's law, first established in [24], can be considered as an upscaled model for the porous medium, as long as there is a distinct separation of scales, between the pore-scale and the continuum-scale. Under certain assumptions on the flow, the Darcy model can be derived from the Navier-Stokes equations. The Navier-Stokes equations are normally used to model flow on the pore scale.

The averaged parameters, $\mathbf{K}$ and μ, describe the conductivity of the rock and fluid, respectively. The permeability acts as a tensor, and may have large spatial variations. In general, the permeability is directional dependent, in which case it is called *anisotropic*. However, for many applications it is convenient to approximate an *isotropic* permeability tensor aligned with the grid, in which case $\mathbf{K}$ becomes diagonal. In practise, the permeability tensor can be hard to determine precisely, and it is usually approximated through a series of lab- and/or numerical experiments. More information on how to define and approximate an averaged permeability tensor is provided in Chapter 3.

The viscosity represents the internal friction of the fluid. In general, the viscosity is a function of pressure and temperature. For primary and sec-

ondary recovery of near-isothermal reservoirs, the phase viscosity is usually represented by a constant value. In enhanced oil-recovery, however, the aim is often to change the value of the phase viscosity, e.g. to lower the viscosity of the oil-phase or to increase the viscosity of the injected water, in order to get a more efficient displacement of the oil.

From the conservation of mass we know that the change in the concentration within a closed volume Ω must be balanced by the fluxes over the boundaries $\partial\Omega$ (inflow minus the outflow), and the internal sources and sinks, which we denote by f. This gives us the following relation for the phase density ρ:

$$\frac{\mathrm{d}}{\mathrm{dt}}\int_{\Omega}\phi\rho\mathrm{dV}+\int_{\partial\Omega}\rho\mathbf{v}\mathrm{dS}=\int_{\Omega}f\mathrm{dV}. \tag{2.2}$$

Here, ϕ is the porosity and is a measure of the accessible volume for the fluid-phase to the total rock volume (see also discussion in Section 1.1.1). The conservation law holds for any arbitrary volume Ω, and by using the divergence theorem together with Darcy's law we can formulate the following single phase flow equation:

$$\frac{\partial\left(\phi\rho\right)}{\partial t}-\nabla\cdot\left(\frac{\rho}{\mu}\mathbf{K}\left(\nabla p-\rho g\nabla z\right)\right)=f. \tag{2.3}$$

The density is a function of the pressure, and the compressibility of a fluid is defined by

$$c=\frac{1}{\rho}\frac{\partial\rho}{\partial p}. \tag{2.4}$$

For an ideal fluid under isothermal conditions, the compressibility c will be constant. For weakly compressible fluids and non-deformable porous media, i.e ρ and ϕ are treated as constant in time, we can approximate the non-linear parabolic equation (2.3) by a linear parabolic equation for the pressure,

$$c\phi\frac{\partial p}{\partial t}-\nabla\cdot\left(\frac{1}{\mu}\mathbf{K}\nabla\left(\nabla p-\rho g\nabla z\right)\right)=\frac{f}{\rho}. \tag{2.5}$$

Moreover, for incompressible fluids ($c=0$), we can introduce the potential $u=p-\rho gz$ and the single-phase equation becomes a static elliptic equation for the potential u:

$$-\nabla\cdot\left(\mathbf{K}^{*}\nabla u\right)=f^{*}. \tag{2.6}$$

For simplicity, we have here scaled the permeability tensor and the right hand side. In the following we will denote by x^{*} the scaled or upscaled parameter of x. Thus, the pressure equation takes the form of a Poisson equation.

In general, we always deal with compressible phases, in which case the flow is governed by non-linear parabolic equations. However, after discretising the parabolic equation in time, we are left with an elliptic equation for the pressure. Equation (2.6) is often refered to as the model elliptic problem and is extensively used for testing the convergence of spatial discretisation techniques and the efficiency of linear solvers stemming from elliptic differential equations.

2.1.2 Two-phase flow

We now consider the case of two-phase immiscible flow. For secondary oil recovery, when the reservoir pressure is higher then the bubble point pressure for oil, we may consider such a two-phase system of oil and water. Thus, we are interested in the displacement of the oil phase by the water phase. In the following we will use the subscripts o and w to denote the oil and water phase, respectively.

In a water-oil system, the water phase will tend to flow close to the pore walls, while the oil phase flows in the pore throats [76]. Both fluids can flow simultaneously and they will occupy a fraction of the same pore volume. The flow of each of the phases will now depend on the presence of the other phase, and we need to introduce three additional parameters; capillary pressure, phase saturation and relative permeability.

The interaction between fluid and rock, results in a slightly higher pressure in the water-phase, compared to the oil-phase. This difference in phase pressures is refered to as the *capillary pressure* and denoted by

$$p_c = p_o - p_w. \tag{2.7}$$

In general, the pore-scale capillary pressure is a highly complex variable, depending on the pore geometry and the surface tension between the fluids. The notion of an upscaled capillary pressure on the Darcy scale is less understood, see e.g. [47] for more information. However, it is clear that the capillary pressure is a function of saturation. In reservoir simulations it is common to use capillary curves, which are based on experiments or mathematical relations of the saturation.

The *saturation* of phase α is denoted by $S_\alpha \in [0, 1]$ and represents the fraction of the total pore space occupied by phase α. The two phases will together fill the entire pore volume.

We define the effective permeability of phase α by $\mathbf{K}_\alpha = \mathbf{K}_{\mathbf{r},\alpha}\mathbf{K}$, where $\mathbf{K}_{\mathbf{r},\alpha}$ is called the *relative permeability* of phase α. While the relative permeability is an upscaled reservoir parameter, it is in general a tensor, however

it is most often considered as a scalar variable. In the following we will only consider the scalar relative permeability, $k_{r,\alpha} \in [0,1]$. For more information on tensor relative permeabilities, see e.g. [56]. The fluid-fluid interaction reduces the total flow. Note that $\sum_\alpha k_{r,\alpha} < 1$, in regions where more then one phase are present.

For simplicity we will also introduce the *mobility* of phase α as

$$\lambda_\alpha = \frac{k_{r,\alpha}}{\mu_\alpha}. \tag{2.8}$$

If we assume that flow of each phase is governed by the individual phase pressures, we can represent the phase velocity by the generalised Darcy's Equation,

$$\mathbf{v}_\alpha = -\lambda_\alpha \mathbf{K} \left(\nabla p_\alpha - \rho_\alpha g \nabla z \right). \tag{2.9}$$

Similar to the case of single phase flow, we can use the conservation of mass to model the evolution of the phase density per time:

$$\frac{\partial \left(\phi \rho_\alpha S_\alpha \right)}{\partial t} - \nabla \cdot \left(\rho_\alpha \mathbf{v}_\alpha \right) = f_\alpha, \qquad \alpha = o, w. \tag{2.10}$$

Note that, $\phi \rho_\alpha S_\alpha$ is the mass of phase α, where ϕS_α represents the fractional pore-space of phase α. To solve the two-phase problem for the pressures p_o and p_w and the saturations S_o and S_w, we also need to define some constitutive relationships for the petrophysical parameters:

$$\begin{aligned} S_o + S_w &= 1, && (2.11)\\ p_o - p_w &= p_c(S_w), && (2.12)\\ \mu_\alpha &= \mu_\alpha(S_w), && (2.13)\\ \rho_\alpha &= \rho_\alpha(P_w), && (2.14)\\ k_{r,\alpha} &= k_{r,\alpha}(S_w), && (2.15) \end{aligned}$$

2.1.3 Pressure and saturation equation

The two-phase flow model (2.10) exhibits two parabolic equations, one for each of the phases $\alpha = o, w$. By applying the constitutive relations (2.12-2.13) we can manipulate the system of parabolic equations and rewrite the two-phase model into one elliptic equation for the pressure and one hyperbolic equation for the saturation. If we expand the time derivative in Equations (2.10) and add $\left(\frac{1}{\rho_o}\right)$ times the first equation with $\left(\frac{1}{\rho_w}\right)$ times the second equation, we arrive at the elliptic pressure equation:

$$\frac{\partial \phi}{\partial t} + \phi \left(S_w c_w \frac{\partial p_w}{\partial t} + S_o c_o \frac{\partial p_o}{\partial t} \right) + \frac{1}{\rho_o} \nabla \cdot (\rho_o \mathbf{v}_o) + \frac{1}{\rho_w} \nabla \cdot (\rho_w \mathbf{v}_w) = \frac{f_o}{\rho_o} + \frac{f_w}{\rho_w}. \quad (2.16)$$

Here, c_α is the compressibility of phase α given by relation (2.4). The first constitutive relation (2.12) was here used to remove the saturations from the equation. The second constitutive relation (2.13) is then used together with Equation (2.16) to solve for p_o and p_w.

In order to derive an equation for the saturation, we will express the Darcy velocity of the water phase in terms of the total velocity $\mathbf{v} = \mathbf{v}_o + \mathbf{v}_w$. By combining (2.9) and (2.13) we can write

$$\mathbf{v}_w = \frac{\lambda_w}{\lambda_o + \lambda_w} \mathbf{v} + \frac{\lambda_o \lambda_w}{\lambda_o + \lambda_w} \mathbf{K} \left(\nabla p_c + (\rho_w - \rho_o) \, g \nabla z \right), \quad (2.17)$$

If we insert (2.17) into (2.10) for the water phase, we arrive at the hyperbolic equation for the saturation

$$\frac{\partial \left(\phi \rho_w S_w \right)}{\partial t} - \nabla \cdot \left(\rho_w \left(\frac{\lambda_w}{\lambda_o + \lambda_w} \mathbf{v} + \frac{\lambda_o \lambda_w}{\lambda_o + \lambda_w} \mathbf{K} \left(\nabla p_c + (\rho_w - \rho_o) \, g \nabla z \right) \right) \right) = f_w.$$

If we assume incompressible fluids and incompressible porous media, we can simplify the elliptic pressure equation. We define $\lambda_t = \lambda_o + \lambda_w$ as the total mobility and write

$$- \nabla \cdot (\lambda_t \mathbf{K} \nabla p_w) = \frac{f_o}{\rho_o} + \frac{f_w}{\rho_w}. \quad (2.18)$$

Note that in the case of $\lambda_o = \lambda_w$, i.e. the total mobility is constant, the pressure equation is time-independent and needs only be solved once. Thus, the saturations may be updated repeatedly in time, from a fixed velocity field.

The pressure is a global elliptic variable which in general needs to be recalculated everywhere at each time-step, whenever λ_t has changed. The saturation S_w is a local variable, and needs only be changed locally in parts of the reservoir, at every time-step. Typically, the saturation needs to be updated close to the water-oil contact. The total mobility $\lambda_t = \lambda_t(S_w)$ is a function of the saturation, thus it changes locally. A good approximation of the pressure may be obtained by only updating the pressure close to the water front where λ_t has changed significantly.

In the multiscale simulations, which will be explained in Chapter 5, we couple a global pressure variable on a coarse scale with local pressure variables on the fine-scale (continuum scale). Thus, in each time step, we only need to solve one smaller (upscaled) global problem and those local fine-scale problems in which λ_t has changed. This enable us to run simulation on the fine-scale much more efficiently. The accuracy of the multiscale simulations may be controlled by expressing the multiscale method in the framework of domain decomposition. The development of robust multiscale methods for reservoir simulation has been the main objective of this thesis.

2.2 Numerical reservoir simulation

The mathematical models described above, are too complicated to be solved analytically and need to be treated numerically by means of computational methods. Even for the case of single phase flow, only special situations of simplified geometry and rock properties can be solved by analytical methods. Those cases are, however, important as benchmark tests for the computational methods.

In the following we will describe the numerical reservoir simulator. The purpose of the reservoir simulator is to efficiently solve the mathematical equations, for a given characterisation of the reservoir and the fluids inside, and to provide robust and accurate solution strategies. In principle we want to take large time-step in our simulations, while keeping the number of Newton iterations low. In the following we will describe two important features of the numerical reservoir simulator; the time-stepping method and the numerical grid. A more detailed description of numerical reservoir simulation can be found in [37, 76].

2.2.1 Time-stepping

The discretisation in time is carried out by a first order finite difference scheme, typically the Euler's formula. At each time-step of the simulation, we then have to solve for the dynamic variables, like saturation and pressure. In principle, the equations for flow and transport in porous media are highly non-linear, and most reservoir simulators solve these equations by applying the Newton's method. This is an iterative method for calculating a certain steady state of a non-linear function by taking linear steps. At each time step of the simulation, the Newton's method linearises the equation and iterates until the residual (of the non-linear equation) is lower than some pre-defined tolerance value. The number of iteration needed can be controlled by the

size of the time-step.

There are several approaches on how to solve for the dynamic variables, pressure and saturation. The pressure exhibit an elliptic character, and needs to be treated implicitly, i.e. at the end of the time-step. Note that the calculation of an elliptic variable result in a globally coupled problem which is very time-consuming to solve. The linear pressure solver often constitute the main computational cost when running a full-field reservoir simulation and efficient methods for solving the elliptic problem is the main focus of the reminding of the chapter. However, the choice of solution strategy for the saturation variable is also important, as it defines the time-stepping approach.

The saturation can be solved either implicitly or explicitly, resulting in different numerical formulations with different properties. In general, a fully implicit treatment of the dynamic variables leads to the most robust and stable method. The increased stability, give us the possibility of choosing longer time steps for our simulations. These favourable properties make the fully implicit method (FIM), the method of choice for the petroleum industry. The formulation is unconditionally stable and has no direct time-step limitations. However, each Newton-iteration requires solving the fully coupled system, which can be time-consuming for large compositional systems.

If we treat the saturations explicitly, we arrive at the *implicit in pressure, explicit in saturation* (IMPES) formulation. The pressure, because of its elliptic nature, is always evaluated at the end of the time-step, while the saturation is evaluated at the beginning of the time-step. This allows us to solve for the pressure and saturations independently, in a sequential manner. As a consequence, the time-step is restricted by the propagation speed of the saturations and space discretisation of the computational grid. This is known as the CFL-criterion. Each linear time-step, using IMPES, is in general much faster to solve, however, the time-step restrictions may kill the efficiency of this method.

In general, there are many more strategies for efficiently solving the dynamic variables, specially for compositional models. For more detailed about these and other time-stepping strategies, see e.g. [22].

2.2.2 The grid

A significant part of the numerical reservoir simulator is the grid. For the mathematician, it is known as the computational grid, and has a direct implication on the validity of the discretisation techniques and performance of the numerical methods. For the geologist, the grid is sometimes refered to as the flow grid or reservoir grid, and is designed to adapt to the geological layers and heterogeneities of the reservoir. Generally speaking,

the geologists design the grid, while the mathematicians discretise the equations in accordance with the grid, and try to build numerical schemes which are robust with respect to the given grid.

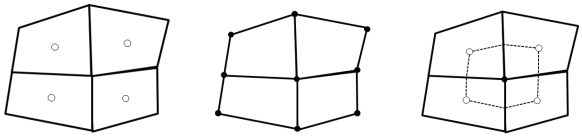
Flow grid

In principle, the flow grid is designed to represent the geology and geometry of the reservoir as best as possible. The main flow will follow the layering of the reservoir and it is natural that the grid also follows these layers. A typical flow grid for a petroleum reservoir is based on hexahedral grid cells. The 2D cross section of such a grid gives a quadrilateral grid. The geological layers are mostly horizontal and parallel, in which case it is simple to represent hexahedral grid cells in 3D and quadrilateral grid cells in 2D.

Commercial simulators like Eclipse [85] have traditionally been based on a natural numbering of the grid cells, in which case the flow grid needs to be stratigraphical. These constraints make it difficult to represent heterogeneous structures like fractures and faults explicitly in the simulations. Moreover, it is not uncommon to have crossing fractures or faults. A stratigraphical gridding of fractured and faulted reservoirs can result in many small and skewed grid cells with possibly large aspect ratio. Such grid cells are typically very difficult to represent in the numerical methods and may contribute to errors and large CFL-numbers.

When dealing with heterogeneous structures, like fractures, faults and collapsing layers, we experience abrupt changes in the rock properties which we know will have a strong effect on the fluid flow. Thus, it is especially important to model these cases correctly. Triangular grids have been shown to be very flexible w.r.t. adapting to different heterogeneous structures. A unique Delaunay grid can be generated, by connecting any distribution of grid points [87]. However, the number of grid cells can become large, and the increased number of connecting cells in the numerical scheme can also become a challenge. Another possibility is to used local grid refinement (LGR) in the neighbourhood of local heterogeneities, where increased resolution is required. All these adaptive gridding strategies usually result in unstructured grids, which can be difficult to handle for many numerical methods; it is difficult to find a natural numbering of the grid cells and the numerical scheme do not have a predefined structure. Algebraic methods, which can adapt to general sparse matrix-structures are then preferred.

Over the last couple of decades, much work has been devoted to the design of robust numerical methods for general grids. In Eclipse the algebraic multigrid (AMG) preconditioner has replaced the previous nested



(a) Cell-centred (b) Mesh-centred (c) Dual grid

Figure 2.1: An illustration of different grid types.

factorisation, used together with ILU. Such numerical methods in general allow for more complicated grids.

Computational grid

While the flow grid represents a discretisation of the petrophysical properties of the reservoir, the computational grid is related to the discretisation of the dynamical variables, like pressure, velocity and temperature. These variables are represented at discrete points on the grid, normally refered to as nodes. There is a clear distinction between cell-centred, and mesh-centred grids, where the dynamical variables are represented at nodes in the centres of the cells or at centres of the mesh. For some numerical methods, both of these grids are applied , where the one grid is considered the dual representation of the other. This is the case for the multiscale control-volume methods to be discussed in Chapter 5. Examples of different computational grids are shown in Figure 2.1.

In most conventional reservoir simulators they apply the control-volume discretisation in the space dimensions. Control-volumes are cell-centred grids, representing integrated quantities. The control-volume discretisation will be discussed in the next chapter.

2.3 Control volume discretisation of the elliptic equation

For a given mathematical model and a given computational grid, the next step is to discretise the equations. While the main work of this thesis is devoted to numerical multilevel methods for solving elliptic problems, we consider the discretisation of the Elliptic equation (2.6):

$$-\nabla \cdot \mathbf{K}\nabla u = f, \tag{2.19}$$

which is often refered to as the model problem for elliptic equations. For simplicity, we neglect gravity and refer to the elliptic variable u as the pressure and $\mathbf{K}$ as the permeability of a porous media. Hence, Equation (2.19) is also often refered to as the pressure equation. A delicate issue for elliptic problems for flow in porous media, is that the tensor coefficients of $\mathbf{K}$ are discontinuous and the spatial variability of $\mathbf{K}$ can be large. A major challenge lies in computing the fluxes across these discontinuities.

For the time-discretisation, the accuracy of the computations is controlled by lowering the time-step. Similarly, we seek spatial discretisation techniques which are convergent in the limit as the spatial grid size $\Delta x \to 0$. However, it should be mentioned that a typical discretisation of the pressure equation for reservoir simulation yields large coupled systems of equations to be solved, and refinement of the grid is not always possible. In fact, a large class of mathematical techniques aims at coarsening the elliptic problem, to a scale more suitable for conducting numerical simulations. Such techniques are investigated in Chapter 3.

The method of choice for most reservoir simulators is the control volume method. This is a class of discretisation techniques, based on the integral formulation of the principle of mass conservation (2.2). If we integrate Equation (2.19) across a control volume Ω and use the divergence theorem, we can write the control-volume formulation as

$$-\int_{\partial\Omega_i} \mathbf{v}\cdot\mathbf{n}\,\mathrm{dS} = \int_{\Omega_i} f\,\mathrm{dV}. \tag{2.20}$$

Equation (2.20) states that the mass transfer between the control-volumes is balanced between the accumulation term, f, on the right hand side, and the fluxes, across the interfaces. We define the flux q across a cell-interface Γ by

$$q = -\int_{\Gamma} \mathbf{v}\cdot\mathbf{n}\,\mathrm{d}\sigma = -\int_{\Gamma} \mathbf{n}^T\mathbf{K}\nabla u\,\mathrm{d}\sigma \tag{2.21}$$

The control-volume formulation is based on integrated quantities and allows for discontinuous permeability and discontinuous pressure across the interfaces. The formulation only requires that the source term $f \in L^\infty(\Omega)$ and that the boundary $\partial\Omega$ is Lipschitz continuous, such that the integrals are bounded and well-defined. The class of numerical schemes based on the control volume formulation, needs to discretise the flux-expression (2.21) for each cell interface such that relation (2.20) is satisfied. The flux is normally considered as a linear function of the pressure, in witch case the discretisation of (2.20) takes the following linear form:

$$\sum q_i = \sum_i \sum_j t_{ij} u_j = \sum f_i. \tag{2.22}$$

Here, the index i runs over all cells of the grid, while j represents the indices of all neighbouring grid-cells which contribute to the flux across the interfaces of grid cell i. The coefficients $t_{i,j}$ are refered to as the *transmissibillity coefficients*, corresponding to the flux across the boundary of grid cell i. If we assembly all flux expressions in Equation (2.22) and (2.20) we get a globally coupled system of equations,

$$Tu = f. \tag{2.23}$$

The global system matrix T contains all local transmissibillity coefficients t_{ij}. This matrix will be sparse, and if the grid cells have a local numbering along the coordinate axis, the sparsity pattern will be diagonal. The number of connections $\{t_{ij}\}$ per grid cell i will vary with the choice of control-volume scheme; We will denote this number as the *dimension of the numerical scheme*. In the following we will introduce two different discretisation schemes for the control-volume method.

2.3.1 The two-point flux approximation method

The *two-point flux approximation* (TPFA) method is perhaps the simplest, and it is the most widely used of all the control volume methods. As the name indicates, the method approximates the flux across a given interface, or surface in 3D, by applying the physical state of the two adjacent grid cells. By considering the situation in Figure 2.2, we express the flux $q_{i+\frac{1}{2}}$ across the interface $\Gamma_{i+\frac{1}{2}}$ as

$$q_{i+\frac{1}{2}} = -\frac{u_{i+1} - u_i}{\frac{1}{2}\left[\frac{\Delta x_i}{k_i} + \frac{\Delta x_{i+1}}{k_{i+1}}\right]} = -t_{i,i+1}\left(u_{i+1} - u_i\right). \tag{2.24}$$

Here, the denominator denotes the harmonic averaged permeability defined on the interface between cell i and $i+1$. Note also, that the TPFA-formulation requires a scalar permeability coefficient in the direction perpendicular to the interface $\Gamma_{i+\frac{1}{2}}$. This is only the case if the grid is aligned with the principle directions of the permeability tensor, in which case the grid is said to be $\mathbf{K}$-*orthogonal*. A necessary condition for $\mathbf{K}$-orthogonal grids is that the following relationship is satisfied:

$$\nu^T \mathbf{K} \mathbf{n} = 0, \tag{2.25}$$

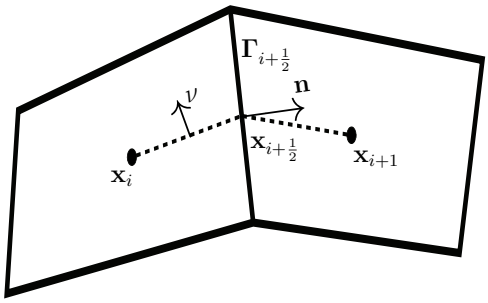


Figure 2.2: The two-point flux approximation

where ν and $\mathbf{n}$ are the normal vectors shown in Figure 2.2. If we sum over all the cell interfaces, Equation (2.20) will result in a five-point scheme for 2D-flow and a 7-point scheme for 3D-flow.

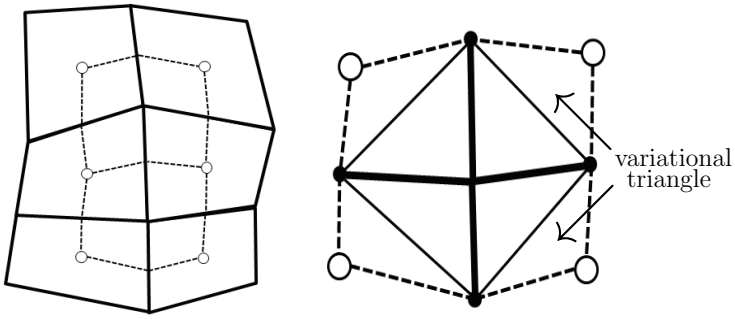
2.3.2 The multi-point flux approximation method

For non-$\mathbf{K}$-orthogonal grids, the TPFA-method is not a consistent approximation of the flux, and we need to consider a larger pressure-stencil for the numerical discretisation. The *multi-point flux approximation* (MPFA) method was introduced independently in [6] and [31], and can be regarded as a generalisation of the TPFA-formulation to the more general case of full-tensor permeability $\mathbf{K}$. In the following, we will describe the MPFA-method for structured 2D quadrilateral grids, as it is presented in [5]. For further extensions to general 3D grids, see e.g. [7].

The MPFA-method uses 6 pressure corner-points to approximate the flux across a single interface Γ. By splitting the interface in two half-edges, we define two *interaction regions* by joining the nodes in the cell centres with the midpoints on the interfaces. Half-edge fluxes are computed on each interaction region and summed together for the flux on the entire interface. The flux-stencil and a single interaction region is illustrated in Figure 2.3. On each sub-cell i the pressure is approximated by a linear function on the *variational triangle*. The variational triangle is also shown for the interaction region in Figure 2.3. If we consider x_j, $j = 1, 2, 3$, as the three corners of the triangle, we can approximate a linear pressure variation on sub-cell i as

$$u_i(\mathbf{x}) = \sum_i u(x_j)\phi_j(\mathbf{x}), \tag{2.26}$$

where ϕ_j^i is a linear basis function satisfying $\phi_j^i(x_k) = \delta_{jk}$, for $k = 1, 2, 3$. By



(a) MPFA (b) Interaction region

Figure 2.3: Illustration of the flux stencil (left) and the interaction region (right) for the MPFA method.

requiring continuity in flux across the interfaces and continuity in pressure at the interface midpoints, we can express the fluxes as a linear combination of the pressure points. On each interaction region the local flux-expression reads

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}. \tag{2.27}$$

By subassembly of all the flux expressions, the MPFA method results in a 9-point scheme for flow-problems in 2D and a 27-point scheme for flow problems in 3D. For general unstructured grids, even more pressure points may be required, but for **K**-orthogonal grids, the MPFA-method reduces to the TPFA-method.

2.3.3 Monotonicity

When designing the discretisation method, we want to honour certain properties of the elliptic equation (2.19). One such property is called the maximum principle. If we consider an elliptic boundary value problem with non-zero Dirichlet conditions, $u \geq 0$ on the boundary $\partial\Omega$, and with non-negative right hand side, $f \geq 0$. Then, the maximum principle will tell us that the solution u is everywhere non-negative in Ω and will attain its minimum value on the boundary. Moreover, the maximum principle holds for any sub-region $\Omega_i \in \Omega$, thus, the solution must be everywhere monotone inside of Ω. We

would like to have the same property for the discrete system of equations (2.23), and we say that the numerical method is *monotone*, if it satisfies the discrete maximum principle.

A sufficient criterion for monotonicity is that the system matrix T is an M-matrix, i.e. all elements of the inverse matrix T^{-1} is non-negative. The numerical scheme provided by the TPFA-method is guaranteed to result in an M-matrix, thus, the TPFA-method is unconditionally monotone. The nine-point scheme of the MPFA-method is not guaranteed to give an M-matrix, and for certain anisotropy ratios the solution may contain spurious oscillations. Sufficient criteria for monotonicity of the 9-point scheme are given in [74]; these are less restrictive than the condition of an M-matrix.

In general, MPFA methods represent a class of linear discretisation methods which reproduce the correct linear pressure field for homogeneous media and general grids. That is, if the numerical scheme is monotone. In Section 2.3.2 we presented the general MPFA-O(η) method, where η represents the continuity point of the pressure on the interface. The most common choice for the MPFA-O method, is $\eta = 0$, which denotes continuity in pressure at the mid-point of the interfaces, but in general, η can be chosen arbitrary between 0 and 1. For instance for triangles, the MPFA-O(1/3) method is shown to be more beneficial, as it provides a symmetric MPFA-scheme. Much work has been devoted to optimise the MPFA-methods w.r.t. convergence and to develop robust MPFA-schemes for problems involving anisotropic porous media. It has been shown that no 9-point schemes can be made unconditionally monotone for every grid and permeability fields [55], however, the violation of the monotonicity criteria do not necessarily result in large errors. In fact, there do not exist any explicit methods of quantifying the size of the spurious oscillations that may occur. For quadrilateral grids in 2D, the MPFA-L method [8] results in a compact 7-point scheme. This method has shown improved properties with respect to monotonicity. While it is difficult to construct robust MPFA-schemes for general grids, often small perturbations of the grid may transform a non-monotone scheme into a monotone scheme [96]. For a more thorough investigation on robust control-volume methods on general grid structures, consult [55].

2.4 Linear solvers

In the following we will focus on different solution strategies for solving the linearised equations. The linear system of equations arising from the linearisation step in the Newton's method is typically very large, up to $10^6 - 10^7$ number of unknowns is not uncommon. The linear solver of the reservoir

simulator is often the main bottleneck of the simulations and it may account for more than 70 percent of the total running time [14]. Thus, the choice of an efficient linear solver is crucial.

2.4.1 Direct solvers

The linear system of equations to be solved can be written as

$$Au = b, \tag{2.28}$$

where A in general is a $(n \times n)$ non-symmetric and sparse system matrix and b is the corresponding $(n \times 1)$ right-hand side vector. In principle, the exact solution ($u = A^{-1}b$) is simple to compute by inversion of the system matrix. This can be carried out by e.g. a Gaussian elimination of Equation (2.28) at the cost of $O(n^2)$ floating operations, if the sparsity of A is taken into account.

An efficient direct solver is based on the factorisation of A, into matrices which are faster to invert. Important techniques include the LU- and Cholesky-factorisation. By factorising $A = LU$, the linear system (2.28) can be solved sequentially by the following two equations:

$$Ly = b, \tag{2.29}$$
$$Uu = y. \tag{2.30}$$

Thus, the solution can be expressed as $u = U^{-1}L^{-1}b$. Various choices of L and U are possible. If the system matrix is badly scaled, or close to singular, a scaling and permutation of the system matrix is also possible. This results in a PLU-factorisation $A = PLU$, where the P-matrix can be regarded as a preconditioning, prior to the LU-factorisation. If A is a symmetric and positive definite matrix, the Cholesky factorisation $A = LL^T$ can be applied, resulting in faster algorithms for solving the linear system.

A direct solver, oppose to an indirect iterative solver is regarded as the most robust solution strategy, however, as the number of linear equations becomes large, a direct calculation of (2.28) gets extremely expensive. Because of the poor scalability of direct solvers with respect to the number of grid cells n, indirect solvers are almost always preferable in reservoir simulations.

2.4.2 Iterative solvers

Iterative methods aim at solving the linear system (2.28) without inverting the system matrix A. The problem is then formalised as a minimization

problem of the error function $e = u - \hat{u}$ evaluated in some norm, where $\hat{u}$ denotes the approximation to the exact solution u. Another possibility is to minimize the residual function

$$r = b - A\hat{u}, \tag{2.31}$$

where the actual error is related to the residual through

$$e = A^{-1}r. \tag{2.32}$$

The iterative methods can be classified into two major classes; the relaxation methods and the prolongation methods. In the following we briefly discuss two of the most popular iterative methods, the Richardson iteration and the Krylov subspace method, as representatives for the two classes of iterative methods.

Richardson Iteration

The Richardson iteration represents a class of iterative methods based on relaxation of the approximated elements of the solution vector. Starting from an arbitrary initial vector $u^{(0)}$, the iteration can be written on the form

$$u^{(i+1)} = u^{(i)} + \omega \left(b - Au^{(i)}\right), \tag{2.33}$$

where $\omega > 0$ is an arbitrary scaling parameter. Both the Jacobi, Gauss-Seidel and SOR- method are based on this algorithm, which aims at improving each individual element of the approximated solution vector by local corrections of the error function. In order to guarantee converge to the true solution u, i.e. $\frac{u^{i+1}-u}{u^i-u} < 1$, we must choose $\omega < \frac{2}{\rho(A)}$, where $\rho(A)$ is the spectral radius of A. Thus, for ill-conditioned problems, $\omega << 1$, and the convergence rate will be slow. For large elliptic problems where the solution is highly dependent upon the boundary conditions, the Richardson iteration typically has a slow convergence. A multigrid method is then typically applied as a preconditioner, to accelerate the convergence. In addition, while certain elements of the approximated solution vector may converge faster then others, local index-sets can be used. Thus, we only need to iterate on a smaller part of the global vector.

The Krylov subspace methods

Designed to solve large systems of equations, the Krylov subspace method is probably the most extensively used linear solver for reservoir simulations

today. The methods belong to a class of projection methods which seek an approximation $u^{(m)}$, of the solution u on the subspace $u_0 + \mathcal{K}_m$, such that the Petrov-Galerkin condition

$$b - Au_m \perp \mathcal{L}_m. \tag{2.34}$$

is satisfied. Here,

$$\mathcal{K}_m(A, r_0) = span\{r_0, Ar_0, A^{(2)}r_0, \cdots, A^{(m-1)}r_0)\} \tag{2.35}$$

is the Krylov subspace of dimension m. The subspace $\mathcal{L}_m$ is usually related to the Krylov subspace, but may be chosen independently for the different Krylov methods.

Consider the functional $F(u) = \|b - Au\|_2^2$, representing the norm of the residual vector. The *generalised minimal residual* (GMRES) method seeks approximations $u^{(m)}$ which minimises the residual,

$$u^{(m)} = arg\{\min_{u \in u^{(0)} + \mathcal{K}_m} F(u)\}. \tag{2.36}$$

It follows that the minimization problem is equivalent to finding the orthogonal projection of the residual onto the subspace $\mathcal{L}_m = A\mathcal{K}_m$. The GMRES algorithm uses the Arnoldi process to generate orthogonal basis functions to the Krylov subspace $\mathcal{K}_m$. Similar to all orthogonalized Krylov methods, the method converges monotonically in at most n iterations, where n is the size of the system matrix A. The actual number of required iterations is typically much smaller, and will depend on the condition number of A. The use of a preconditioner, to lower the condition number of the system matrix, can substantially reduce the number of required iterations. The drawback of the GMRES algorithm, is that in general all search directions need to be stored in memory. For large linear systems, a modified version of GMRES is applied, which restarts the iterative procedure every $n_r < n$ iteration step, using $u^{(n_r)}$ as the new initial vector. The optimal choice of n_r is however not always clear.

If A is symmetric and positive definite, the *conjugate gradient* (CG) method is probably the most efficient Krylov method to use. The CG method seeks approximations $u^{(m)}$ such that the error functional $\left(u^{(m)} - u\right)^T A \left(u^{(m)} - u\right)$ is minimized. Since A is symmetric ($A = A^T$), it follows from (2.32) that this is equivalent to minimizing the residual expression

$$r^{(m)} A^{-1} r^{(m)}. \tag{2.37}$$

Thus, we only need to constructs a sequence of orthogonal residual vectors in $\mathcal{L}_m = \mathcal{K}_m$. The main advantage of CG, is that the search directions can be updated in a recursive manner, and we only need to store information from the last iteration step. Similar to the GMRES method, the CG method is seldom used without a preconditioner.

Several generalisations of the CG method to non-symmetric systems have also been proposed. The perhaps most popular of these methods is the BiCGSTAB method. This method requires less storage capacity then the GMRES method for large number of iterations, however, the convergence is not guaranteed to be monotone. For a more complete discussion about these and other iterative linear solvers, see e.g. [81]

2.4.3 Preconditioners

For the application to reservoir simulation, the system matrix A is often ill-conditioned. A suitable preconditioner, is then the key ingredient for any efficient iterative linear solver, when solving such problems. Instead of solving Equation (2.28), we now seek the solution of

$$(MA)\, u = Mb, \tag{2.38}$$

where $M \approx A^{-1}$, such that the new preconditioned system (2.38) is better conditioned and requires less number of iterations to converge. Of course, the preconditioned matrix M should be fast to construct. Indeed, if $M = A^{-1}$, we have in fact solved Equation (2.28) by a direct solver. An equivalent expression for the preconditioned system can be written in terms of the residual:

$$Bz = r \tag{2.39}$$

where $B = M^{-1} \approx A$, and z is the preconditioned residual. Note that, Equation (2.39) indicates that we do not need to invert the preconditioned matrix B, we only need to efficiently solve the preconditioned linear system such that the preconditioned residual z represent a better approximation to the exact solution.

Incomplete factorisations

While the system matrix A typically has a sparse banded structure, the elements of the factorisation $A = LU$ and the inverse $A^{-1} = U^{-1}L^{-1}$ are not sparse. However, the off-diagonal elements usually have a decreasing value away from the diagonal. Based on this observation many incomplete, sparse factorisations have been proposed, which are fast to invert. These,

include the diagonal Jacobi approximation, the Incomplete LU-factorisation (ILU), the Incomplete Cholesky (IC) factorisation and the nested factorization [10]. Simple as they may seem, they usually work quite well as preconditioners M for the problem (2.38). For the ILU-preconditioner, it is further common to apply a threshold for which element to fill. In the case of ILU(0), only the non-zero elements of A will be represented in the ILU-factorisation. If A has a block structure, block versions of the incomplete ILU and the incomplete Cholesky factorisation also exist.

Domain decomposition

Domain decomposition (DD) is a special type of a block ILU decomposition, where the different blocks also represent a decomposition in space. It can also be written in the form of a Richardson iteration and applied as an iterative method. This will be investigated further in Chapter 4.

Multigrid

The Multigrid method can also operate as a linear solver itself, but is most often applied as a preconditioner for the Krylov subspace methods or the Richardson iteration. In fact, it can be seen as a special type of DD-preconditioner, where only a single sub-domain is considered. Both the multigrid and DD preconditioners are of the form (2.39).

Motivated by the fact that the local relaxation of the discrete node values only efficiently smooths out the high-frequency part of the error, the multigrid algorithm includes coarse-scale solvers on a multiple of coarse grids, to efficiently reduce oscillations in the error at all scales. In general, the multigrid algorithm consists of two main components, fine-scale smoothing and coarse-scale corrections. Local relaxation solvers, also called smoothers, are used initially to smooth out the fine-scale oscillations of the residual. Typically a couple of Richardson type iterations are being used at the finest scale. The residual is then step-wise being restricted to the coarser levels, where coarse-scale corrections are being computed on each level. The coarse-scale corrections are further being interpolated back and collected at the fine-scale. This procedure can further be repeated in several multigrid cycles, until the residual is sufficiently small. An efficient multigrid algorithm consists of n_1 pre-smoothing steps, n_c multigrid cycles and n_2 post-smoothing steps, where the numbers n_1, n_c and n_2 are not necessarily known in advance.

The interpolation functions and coarse-scale equations are not always trivial to construct. For geometric multigrid methods, these are formed through some pre-defined knowledge of the geometry of the grids. Typically

the coarse-scale operators stems from discretisations on the respective coarse-scale grids. For the *algebraic multigrid* (AMG) method, the coarse grids are not known in advance, and all the operators belonging to the coarse grids are calculated through algebraic manipulations of the fine-scale system matrix. Thus, the AMG method can be applied to any type of unstructured grids.

Chapter 3

Upscaling

The geostatistical model describing the rock properties of a given reservoir is often too detailed to be resolved within conventional reservoir simulators. Upscaling is a mathematical (or numerical) technique for developing coarse-scale reservoir models which are more suitable for conducting reservoir simulations. In this chapter we will discuss some of the fundamental techniques for upscaling reservoir parameters. The emphasis will here be on the Pressure equation for single-phase incompressible flow.

3.1 Coarse scale parameters

The various upscaling methods are usually classified in terms of the type of parameters that are upscaled. Typical, for multi-phase flow problems we need to upscale parameters like permeability, porosity, relative permeability and capillary pressure curves. The permeability and porosity are called single-phase parameters, as they are independent upon the different fluids that are present in the reservoir. The relative permeability and capillary pressure are denoted multi-phase parameters, as they are dependent upon the saturation of the different phases. Single-phase parameter upscaling is by far the most studied area within upscaling, and the best understood. In general, for multi-phase flow we also need to upscale relative permeability and capillary pressure for the time-dependent parabolic equation. However, in some cases for moderate level of coarsening, single-phase upscaling methods have shown to be reasonable approaches for upscaling these parameters as well [28]. In this thesis we have studied the multiscale methods. For the multiscale methods, the coarse-scale pressure solution is calculated on the coarse scale and interpolated down onto the fine-scale. Thus, the transport can then be solved on the fine-scale. A coarse model is only considered for the

pressure equation. In the following we will consider upscaling of the elliptic pressure equation and the parameters that are involved. For a summary on general techniques for multiphase upscaling, see e.g. [23, 25].

3.1.1 Coarse scale pressure equation

The coarse scale elliptic equation is usually assumed to be on the same form as the fine-scale elliptic equation,

$$-\nabla\cdot(\mathbf{K}\nabla u_{fine}) = f, \tag{3.1}$$
$$-\nabla\cdot(\mathbf{K}_{eff}\nabla u_{coarse}) = f^*, \tag{3.2}$$

where $\mathbf{K}_{eff}$ denotes the homogenised or effective permeability tensor over the coarse block and f^* is the source term integrated over the coarse block. In the following we neglect gravity and refer to u as the pressure p. The coarse-scale pressure equation is here based on the assumption of scale-separation and spatial periodicity. Let us consider that the fine-scale permeability tensor $\mathbf{K}(\mathbf{x},\mathbf{y})$ oscillates on two distinct and well-separated scales, $\mathbf{x}$ and $\mathbf{y}$, where $\mathbf{x} \ll \mathbf{y}$. In the limit, as $\mathbf{x} \to 0$, there exists a coarse scale model and a representation of the homogenised permeability tensor $\mathbf{K}_{eff}(y)$, which is independent upon the fine scale fluctuations [48].

This is in general not the case for porous media flow, however, we can still try to represent a block permeability tensor (also called equivalent permeability tensor) for the coarse model which preserves certain principle features of the flow on the fine scale. Rubin and Gomez-Hernandez [80] required that the flow rate calculated across the homogenized block should be equal to the average fine-scale flow rate calculated across the same block volume. For an isotropic medium, where the flux and gradient vectors coincide, they defined the block permeability $\mathbf{K}_b$ by the following formula:

$$\mathbf{K}_b = \frac{1}{V}\int_V \mathbf{v}(\mathbf{x})d\mathbf{x}\left(\frac{1}{V}\int_V -\nabla p(\mathbf{x})\right)^{-1}, \tag{3.3}$$

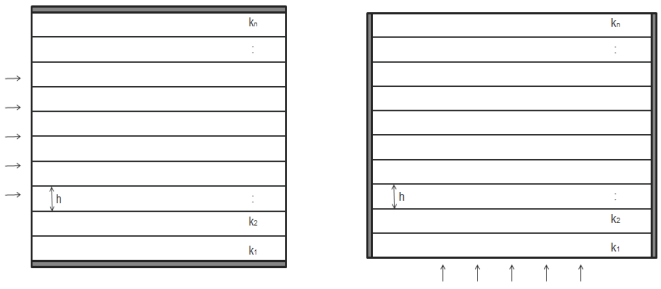
where V is the block volume and $\mathbf{v}(\mathbf{x})$ is the Darcy velocity. Indelman and Dagan [52] further suggested to use the equality of the dissipation of energy as a criterion for calculating the equivalent block permeability. The equality of dissipated energy has the favourable property that the upscaled block permeability, $\mathbf{K}_b$, will be equal to the actual effective permeability, $\mathbf{K}_{eff}$, of the medium, whenever it exists. Based on these and other criteria, several methods for calculating the upscaled permeability have been proposed.

Some of the main challenges of upscaling are: How to generate information about the local fine-scale flow field on the coarse block? How to calibrate that information in order to calculate an upscaled permeability tensor $\mathbf{K}_b$ with the desired equivalent properties, and how to choose a coarse-scale grid for these effective parameters? In the next subsections we will discuss some of the techniques for solving these problems. For a thorough discussion on single-phase upscaling methods, see also [36, 79].

3.1.2 Averaging techniques

First we present a couple of simple analytical averaging techniques for calculating the equivalent block permeabilities. For simplified configurations of porous media, where we have uniform flow, analytical expressions for the effective permeability may exist. Let us consider flow through an idealised layered reservoir, where each layer is assumed to have constant thickness h. We assign a scalar permeability coefficient, k_i, to each of the layers i as illustrated in Figure 3.1 for horizontal and vertical flow.

First we consider the case of horizontal flow. We apply a constant pressure on the left and on the right boundary, $p(0, y) = 1$ and $p(L_x, y) = 0$, and no-flux boundary conditions on the lower and upper boundaries. The pressure gradient in the y-direction is zero, thus, we know that the flow will be uniform and parallel to the layers, and the fluid velocity in each layer will be proportional to the permeability, $v_x^i = k_i$. By requiring that the integrated flux across the external outflow boundary should be equivalent for the fine-scale and coarse-scale model, we can write the expression for the effective permeability,



(a) Horizontal flow (b) Vertical flow

Figure 3.1: Directional flow through a layered reservoir

$$k_{eff} = \frac{1}{n} \sum_i k_i, \tag{3.4}$$

which is the arithmetic average of the fine-scale permeabilities. Now, we can apply the same strategy for the case of vertical flow. Here the boundary conditions are $p(x,0) = 1$, $p(x,L_y) = 0$ and $\mathbf{v} \cdot n_1 = \mathbf{v} \cdot n_2 = 0$. Within each layer we will have a constant velocity $\mathbf{v}_i = k_i \left(p_{i-\frac{1}{2}} - p_{i+\frac{1}{2}} \right)$, where $p_{i+\frac{1}{2}}$ denotes the pressure at the interface between layer i and $i+1$. The flux $\mathbf{v}$ is a conserved quantity and is continuous between the layers, thus the pressure solution will be piecewise linear. If we integrate over all the pressure differences, we can observe that the effective permeability in the y-direction is represented by the harmonic average,

$$\frac{1}{k_{eff}} = \sum_i \frac{1}{k_i}. \tag{3.5}$$

Note that in the case of one-dimensional flow, the harmonic average always gives the correct representation of the effective permeability. That is why the harmonic average of the permeability is used in several spatial discretisation techniques, e.g. the control-volume methods. If the permeability is not aligned, or perpendicular, to the principle direction of the flow, these procedures are not sufficient. This is generally the case for flow in two and three dimensional porous media. A generalization of the two analytical approaches described above is found by the power averaging law, in which

$$k_{eff} = \left[\frac{1}{V} \int_V (k_i)^\omega \, dV \right]^{\frac{1}{\omega}} \tag{3.6}$$

is the effective permeability. Here V represents the volume of the block. In the case of $\omega = 1$ we retain the arithmetic average, while $\omega = -1$ gives us the harmonic average. In [26] they use this formula to calculate the effective permeabilities for shale layered reservoirs. The analytical techniques do not require any information about the underlying fine-scale flow field, thus they are very efficient, but in general these techniques are not sufficiently accurate.

3.1.3 Properties of the effective permeability

For general two and three dimensional flow problems, the effective permeability do not satisfy an additive relationship of fine-scale permeabilities and standard averaging techniques do not apply. $\mathbf{K}_{eff}$ is no longer only a function of $\{\mathbf{K}_i\}$, but also the position of $\{\mathbf{K}_i\}$. The heterogeneous nature of porous

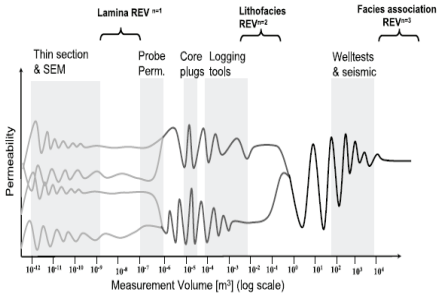


Figure 3.2: The effective permeability measured on different scales. Taken from Nordahl et. al. [72].

rock formations results in preferential flow paths and permeability variations on different scales. If the scales are well-separated, we can identify representative elementary volumes (REVs) for the permeability, similar to that of porosity. Thus, we can calculate an effective permeability tensor $\mathbf{K}_{eff}$ on that scale. However, it should be mentioned that the concept of REV is less understood for permeability, as the result not only depend on the size of the core sample, but also on how the lab-experiment on that sample is being conducted. Figure 3.2 gives an example of how the permeability may change as a function of the sample size. See Nordahl et. al. [72] for more information on the REV for permeability.

If we assume that we have scale-separation, the homogenization theory states that in the limit as the fast scale goes to zero there exists a coarse scale pressure equation (3.2), with a unique effective permeability tensor $\mathbf{K}_{eff}$ which is independent of the fast-scale. The effective permeability tensor can further be proven to be symmetric and positive definite [48]. Note, that even though the fine-scale medium is isotropic and homogenised, the effective permeability on the coarse scale may still be anisotropic with a full tensor representation. Thus, we transform our fine-scale problem to a more complicated one on the coarse scale.

3.2 Flow-based upscaling

In practise, we do not have information about any possible REV scales and the geometry is to complex to be able to analytically compute the effective permeability. Thus, we seek numerical techniques to calculate equivalent block permeability tensors $\mathbf{K}_b$. Flow-based upscaling is a numerical upscaling

technique, which preserves the main features of the flow on the fine scale. It consists of two steps. The first step is to gather information about the fine-scale flow field on the coarse block. This can be done either by solving a local or a global fine-scale flow problem. Based on the information from several fine-scale flow fields, we can approximate an equivalent permeability tensor $\mathbf{K}_b$ for that coarse block. The calibration of $\mathbf{K}_b$ is the second step and may also be done either locally or globally. Hence, the different approaches are divided into four groups; local-local, local-global, global-local and global-global upscaling methods.

We will here describe the local-local and local-global upscaling methods. These methods will further be useful for the description of multiscale and domain decomposition methods in the following chapters. Global upscaling methods (global-local and global-global) are considered too computationally expensive for solving large multiscale problems, and is not treated here.

3.2.1 Local-local upscaling

Local-local upscaling consists of solving a set of local flow problems, from which we can calibrate an equivalent permeability tensor, purely based on local information. We can think of the local flow problems as a type of numerical lab-experiments. We would like these experiments (or flow problems) to be independent from each other and as close as possible to actual reservoir conditions. A critical issue for these methods, is the choice of boundary conditions for the numerical tests. It is important to understand that for most upscaling problems, no unique solution exists and each set of different boundary conditions will in general result in different block permeabilities $\mathbf{K}_b$. While the actual boundary conditions are not known a priori and will most

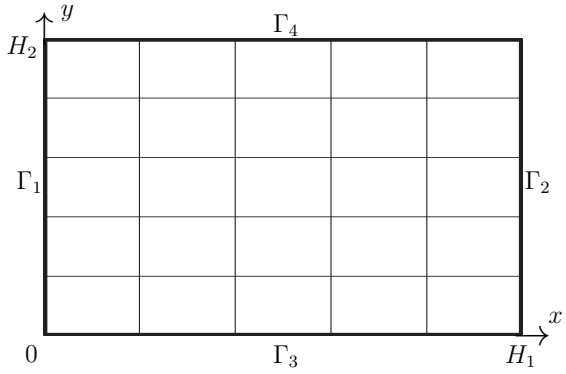


Figure 3.3: Coarse block

likely vary throughout the simulation, we seek natural boundary conditions which we believe will capture the main features of the flow on the fine-scale. For simplicity we will here explain the various methods in two dimensions and for regular Cartesian grids, however, the methods are naturally extended to three dimensions and general grids. Note that the implementation becomes more challenging on unstructured grids.

The most common choice of boundary conditions, is to consider fixed pressure on two opposite boundaries, and no-flux conditions on the two remaining boundaries. Consider the single block domain shown in Figure 3.3. The flow in the x-direction is approximated by solving a fine-scale pressure equation (3.2), subject to the following boundary conditions,

$$p(0, y) = 1, \tag{3.7}$$

$$p(H_1, y) = 0, \tag{3.8}$$

$$\mathbf{v} \cdot \mathbf{n}_3 = \mathbf{v} \cdot \mathbf{n}_4 = 0, \tag{3.9}$$

where $\mathbf{n}_3$ and $\mathbf{n}_4$ are the normal vectors corresponding to the interfaces Γ_3 and Γ_4, respectively. Similarly we can define boundary conditions for flow in the y-direction, and in three dimensions we would have a third flow problem in the z-direction.

The calibration of a coarse-scale permeability tensor $\mathbf{K_b}$ can be done in several ways. The two most common local procedures are: to evaluate the total flux across the outflow boundaries, and to evaluate the volume averaged flux across the entire block volume. By considering the total flow across the two outflow boundaries (Γ_2 and Γ_4) we can compute an approximation to the effective block permeability in the two coordinate directions. The total flow across the outflow boundary Γ_2, for flow in x-direction, can be calculated as

$$q_1 = \int_0^L \mathbf{v}(L, y) \cdot \mathbf{n}_2 dy = \sum_j (\mathbf{v}_j \cdot \mathbf{n}_2), \tag{3.10}$$

where j is an index for the fine-scale cells on the boundary Γ_2. The flux q_1 represents the integrated flow in the x-direction, calculated from numerical experiment 1. Using the Darcy's law for flow in one dimension, we can calculate the equivalent permeability $k_1 = \frac{q_1 L_1}{L_2}$ for the flow in x-direction across the homogeneous block. Similarly we can compute the equivalent permeability for flow in the y-direction. This simple construction gives us a diagonal permeability tensor on the coarse scale. In general, the flow across the coarse block will be anisotropic and a diagonal approximation of the block permeability is not sufficient. Thus, we need to allow for a full-tensor representation of $\mathbf{K}_b$. This may be achieved by calculating the volume

averaged flux across the block volume. Thus, from the solution of numerical experiment i ($\mathbf{v}_i$ and p_i), we can compute

$$\bar{\mathbf{v}}_i = \frac{1}{V}\int_V (\mathbf{v}_i)\,\mathrm{dV} = \frac{1}{V}\sum_{n_c} \mathbf{v}_i, \tag{3.11}$$

$$\overline{\nabla p_i} = \frac{1}{V}\int_V \nabla\,(p_i)\,\mathrm{dV} = \frac{1}{V}\sum_{n_c} \nabla p_i \tag{3.12}$$

where n_c now sums over all fine-cells belonging to the coarse block. While each of the volume averaged fluxes ($\bar{\mathbf{v}}_1$ and $\bar{\mathbf{v}}_2$) and pressure gradients ($\overline{\nabla p}_1$ and $\overline{\nabla p}_2$) has two components, one in each coordinate direction, we get a (4×4) linear system to solve for the four components of $\mathbf{K}_b$,

$$\begin{bmatrix} \overline{\nabla p}_1^1 & \overline{\nabla p}_1^2 & 0 & 0 \\ 0 & 0 & \overline{\nabla p}_1^1 & \overline{\nabla p}_1^2 \\ \overline{\nabla p}_2^1 & \overline{\nabla p}_2^2 & 0 & 0 \\ 0 & 0 & \overline{\nabla p}_2^1 & \overline{\nabla p}_2^2 \end{bmatrix} \begin{bmatrix} k_{11} \\ k_{12} \\ k_{21} \\ k_{22} \end{bmatrix} = - \begin{bmatrix} \bar{\mathbf{v}}_1^1 \\ \bar{\mathbf{v}}_1^2 \\ \bar{\mathbf{v}}_2^1 \\ \bar{\mathbf{v}}_2^2 \end{bmatrix}. \tag{3.13}$$

This calibration do not however account for the symmetry of $\mathbf{K}_b$. In order to guarantee a symmetric tensor, an additional constraint ($k_{12} = k_{21}$) must be made, resulting in a fifth equation in (3.13).

The no-flux boundary conditions used in the example above forces the flow in one coordinate direction at a time, thus, restricting the influence of transversal flow. Several Dirichlet and Dirichlet-Neumann boundary conditions have been studied, which allow for flow across all boundaries of the coarse block, s.a. linear or fixed pressure boundary conditions [43] and periodic boundary conditions [29, 16]. Both of these boundary conditions will lead to a full-tensor permeability whether we consider equality of total flux across the boundaries or average flux across the entire block volume. The periodic boundary conditions also result in a symmetric and positive definite tensor. This has been rigorously proved by Bøe [16] for the case of conservation of Dissipation. For general boundary conditions, a symmetric and positive definite permeability tensor is not guaranteed and must be obtained through a post-processing of the permeability coefficients.

All the techniques discussed above are local and regarded as relatively cheep to solve. In general, three local fine-scale problems need to be solved in order to compute an equivalent full-tensor permeability for each block in three dimensions. These methods give reasonable approximations to the equivalent block permeability whenever the sub-scale heterogeneities are well captured within the coarse-block [95]. Thus, an important issue of local-local upscaling

techniques is the generation of a coarse-scale grid. In Sandvin et. al. [84] we developed an algebraic technique of generating local boundary conditions (interface approximations) which are independent upon the underlying grid. These boundary conditions showed to be more robust with respect to flow on irregular grids and heterogeneous log-normal permeability.

3.2.2 Extended local and local-global upscaling

If the fine-scale problem involves large correlation lengths in the permeability (e.g. channelised reservoir), local-local upscaling methods may yield large errors and global information is necessary to take into account. The solution of global numerical experiments is considered too computationally expensive and will not be treated here, but there are other ways of incorporating global or extended local information into the calibration of the equivalent permeability.

The extended local upscaling methods were not discussed in the previous subsection. These methods are similar to local-local upscaling methods, but solve the local problem on an extended local domain. We can think of it as having a boundary layer around the coarse block. By knowing the permeability distribution in the vicinity around the coarse block of interest, we can more accurately approximate the local fluxes which contributes to the local boundary conditions for the local-local upscaling. The calibration of the block permeability tensor is done locally on the coarse block. In [95] they refer to this as oversampling, and demonstrates how extended local upscaling can reduce the resonance error of the larger scale heterogeneities. This oversampling technique was first introduced in [49] for the multiscale methods, and will be discussed more in detail in Section 5.1.2.

The local-global upscaling method [20] takes global information into account, but without solving any global fine-scale simulations. The global boundary conditions are coupled to the local sub-domains through the coarse scale only, and the local boundary conditions that are used to locally calculate the equivalent block permeability tensor on the coarse-scale, is updated through an iterative solution process. The local-global approach consists of the following steps:

- Local-local upscaling
- Solve the coarse problem for the pressure
- Interpolate the coarse scale pressure onto an extended local boundary
- Recalculate the upscaled parameters (extended local-local upscaling with new pressure boundary conditions)

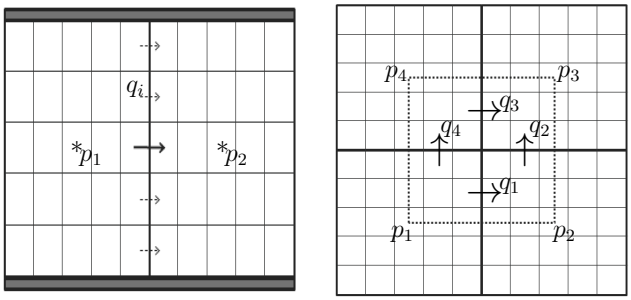
- Recalculate the coarse scale pressure

The procedure continues until the coarse scale pressure satisfies a predefined tolerance value for the upscaling. The main motivation of this approach is to efficiently communicate the influence of the global boundary conditions onto the local problems. For problems involving highly heterogeneous formations with long correlation lengths, e.g. channelised reservoirs, the local-global method may yield considerably more accurate approximations compared to local-local methods. Finally, an adaptive local-global method [19] further increase the efficiency of the iterative procedure. By examine the convergence locally, the adaptive method only recalculate those local problems which are necessary.

A critical point in the local-global approach, is the local interpolation of the coarse-scale solution. Unlike domain decomposition method, the local-global method only converge on the coarse scale. While the fine-scale solution is sensitive to the choice of the interpolation, the method is not in general able to fully resolve the details of the fine-scale. In [84] we show that by expressing the upscaling technique in terms of a domain decomposition preconditioner, it is possible to develop a local upscaling technique which exactly reproduces pre-defined flow fields.

3.2.3 Upscaling of transmissibillities

Sub-scale heterogeneities may give rise to complex flow paths on the fine-scale, which can not be represented by a single block permeability tensor.



(a) Two-domain (b) Dual coarse block

Figure 3.4: The figure to the left is a two-domain region, often used for transmissibillity upscaling of the TPFA formulation. The figure to right illustrates a coarse scale interaction, applied for upscaling to the MPFA method.

An alternative approach is then to directly try to approximate the flow on the coarse scale, by calculating the upscaled transmissibillity coefficients in the control-volume discretisation scheme.

When upscaling the transmissibillity coefficients we follow the same procedure of first approximating the local flow field, and then calibrating that information in order to approximate the equivalent coarse parameter. Since the transmissibillities are located at the interfaces between neighbouring cells, we now have to consider local numerical flow experiments on the two-domain region. See Figure 3.4(a). Hence, the local problems are twice as large compared to permeability upscaling, but for each coarse interface it is enough to solve one flow problem.

TPFA formulation

By applying the boundary conditions ((3.8)-(3.9)) we can calculate the fluxes q_i across the coarse interfaces. These, together with the average pressures $\bar{p}_j$ from each coarse block can then be used to calculate the upscaled transmissibillity coefficient of the TPFA method. From Equation (2.24) we have that

$$t^*_{12} = \frac{\sum_i q_i}{\bar{p}_1 - \bar{p}_2} \tag{3.14}$$

MPFA formulation

In order to approximate the coarse scale transmissibillity coefficients of the MPFA method we need to take into account more coarse scale pressure nodes. Recall, from Section 2.3.2 that the MPFA-method approximates half-edge fluxes on the interaction region, corresponding to the fine-scale dual grid. Similarly, we can form coarse scale interaction regions by constructing a dual coarse grid, as shown in Figure 3.4(b).

The local transmissibillity matrix T of the flux stencil (2.27) contains 16 elements for 2D quadrilateral grids and can be approximated by solving 4 independent flow problems on the coarse dual-block. These problems can be refered to as the basis problems for the construction of a coarse scale system matrix, and the solutions u of these problems are often called the *basis functions*. We will here denote the basis functions by ϕ_i, $i = 1, \ldots, 4$, and they need to be linearly independent. Thus, from (2.27) we have the following relation for the basis functions of the MPFA method,

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} t^*_{11} & t^*_{12} & t^*_{13} & t^*_{14} \\ t^*_{21} & t^*_{22} & t^*_{23} & t^*_{24} \\ t^*_{31} & t^*_{32} & t^*_{33} & t^*_{34} \\ t^*_{41} & t^*_{42} & t^*_{43} & t^*_{44} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix}. \tag{3.15}$$

To ensure linear independence, it is common to require that

$$\phi_i(x_j) = \delta_{ij} \tag{3.16}$$

and

$$\sum_i \phi_i(\mathbf{x}) = 1. \tag{3.17}$$

Thus, for the basis problem 1 we have that $\phi_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$, and from (3.15) we obtain

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} t^*_{11} \\ t^*_{21} \\ t^*_{31} \\ t^*_{41} \end{bmatrix}. \tag{3.18}$$

Hence, it is also known as upscaling of fluxes.

The accuracy of the upscaling methods is dependent upon the choice of boundary conditions for these basis problems. Typical choices are linear pressure conditions, no-flow conditions and periodic boundary conditions. The different approaches can also be improved by solving the basis problems on extended local or semi-global problems. The local-global approach explained in the previous section applies both to upscaling of permeability and transmissibillity.

Finally, it should be noted that even though more complex fine-scale flow fields may be taken into account in the upscaling of the transmissibillities, the coarse scale pressure cannot resolve these fine-scale details. Obviously, the coarse scale solution only account for coarse scale flow, and is only equivalent to some average of the fine-scale solution. For multiphase flow, the fine-scale flow field changes with saturation in each time-step and an upscaling of the fluxes may only be valid for a certain steady state of the petrophysical properties.

However, the basis functions may be updated, as a function of saturation, and that is what is being done in the multiscale methods. These will be discussed more in depth in Chapter 5. In the next chapter we will explain the domain decomposition methods. These methods couple the coarse scale

solution to the fine-scale, which makes it possible to efficiently solve the fine-scale solution through iterative algorithms.

Chapter 4

Domain Decomposition

In the previous chapter we have discussed how the fine-scale parameters of the geological model are being upscaled and represented on an effective coarse scale. The coarse scale model is often refered to as the simulation model, as it is much more efficient to work with, and thus more practical for every day engineering work. The geological model is then used to validate the results from the simulation model.

Domain decomposition represents another mathematical technique, for efficiently solving large linear systems of equations arising from the discretisation of the elliptic problem. It can be used on both the fine-scale and coarse scale discrete problems. The main idea is to decouple the large global boundary-value problem into several smaller local problems, and to carry out the solution by means of an iterative solution process. Domain decomposition has shown to be very suitable for parallel computing, and is also a convenient framework to build and analyse multi-level methods.

The methods of domain decomposition can be classified in several ways; in terms of the decomposition strategy (overlapping or non-overlapping), the iterative solution strategy (solver or preconditioner) or the number of coarse levels.

There are two main classes of domain decomposition methods; Schwarz methods for overlapping sub-domain and substructuring methods for non-overlapping sub-domains. We will start by giving a brief introduction to the framework of Schwarz methods for general overlapping sub-domains in Section 4.1, before directing our attention to the theory of substructuring methods for non-overlapping decompositions in Section 4.2. For the extension to many sub-domains, we need to accelerate the convergence by including a coarse space, this will be the focus of Section 4.3 and 4.4. In the end of the chapter, we discuss several choices of coarse-scale basis functions, and their ability to capture sub-scale heterogeneities. This will further be useful for

our discussion of multiscale methods in Chapter 5.

4.1 Introduction to Schwarz methods

In this section we introduce the general framework of Schwarz methods for overlapping sub-domains. We will show how domain decomposition can be formulated as an iterative solver for solving the coupled elliptic problem. Different iterative schemes lead to different Schwarz algorithms. We discuss the various forms of the Schwarz method in Section 4.1.2. Finally, we also show how the Schwarz method can be applied as a preconditioner for an iterative solver.

Over the last 3-4 decades there has been an intense research on the convergence properties and robustness for these methods. We will not go into details on the rather broad theory on Schwarz methods, but try to highlight some of the more important components and tools for constructing efficient multi-level Schwarz methods. For a more thorough presentation on the theory of Schwarz methods, consult e.g. [78, 88, 92] and the references therein.

For simplicity we will consider the Poisson problem

$$-\Delta u = f \quad \text{in} \quad \Omega \tag{4.1}$$
$$u = 0 \quad \text{on} \quad \partial\Omega \tag{4.2}$$

where the solution $u \in \mathrm{H}^1(\Omega)$, the right-hand side vector $f \in \mathrm{L}^2(\Omega)$ and $\partial\Omega$ is the Lipschitz continuous boundary of Ω.

4.1.1 The Schwarz method for overlapping sub-domains

The pioneering method of Schwartz (1870) [86], also known as the alternating Schwarz method, is considered to be the first contribution to the domain decomposition literature. He proved the convergence of an iterative procedure of solving the decoupled elliptic boundary value problem. The motivation at that time was not to establish an iterative framework for parallel computing, but he used the algorithm to show the existence of harmonic functions with prescribed boundary values [92]. Never the less, the algorithm of Schwarz illustrates many of the important features of domain decomposition as it is used today.

The iterative method of Schwarz can be described as follows. Consider Ω to be a closed region consisting of two overlapping sub-domains Ω_1 and Ω_2 with corresponding boundaries $\partial\Omega_1$ and $\partial\Omega_2$. Denote the overlapping

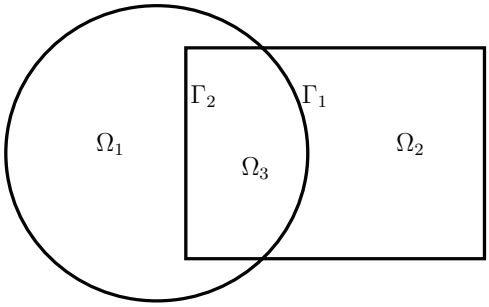


Figure 4.1: Two overlapping sub-domains

region Ω_3, as it is shown in Figure 4.1 and let Γ_i be the part of $\partial\Omega_i$ that also corresponds to the boundary of Ω_3. Given an initial guess u^0, which vanishes on $\partial\Omega$, we calculate u^{k+1} by sequentially solving:

$$\begin{aligned} -\Delta u^{k+\frac{1}{2}} &= f && \text{in} && \Omega_1, \\ u^{k+\frac{1}{2}} &= u^k && \text{on} && \Gamma_1, \\ u^{k+\frac{1}{2}} &= 0 && \text{on} && \partial\Omega_1 \cap \partial\Omega, \\ -\Delta u^{k+1} &= f && \text{in} && \Omega_2, \\ u^{k+1} &= u^{k+\frac{1}{2}} && \text{on} && \Gamma_2, \\ u^{k+1} &= 0 && \text{on} && \partial\Omega_2 \cap \partial\Omega. \end{aligned} \tag{4.3}$$

After the first fractional step, the solution $u^{\frac{1}{2}}$ on Ω_1 is mapped onto the boundary Γ_2, and the solution u^1 is carried out by solving the boundary-value problem on Ω_2. In the same manner, the solution on Ω_2 is mapped onto Γ_1 for the updated boundary-value problem on Ω_1, and so on. The iteration converges, when the exact boundary conditions are found for the internal boundaries Γ_1 and Γ_2.

Under certain assumptions on Ω_1 and Ω_2 it is possible to show that there exist some constants C_1, $C_2 \in (0,1)$ such that the error of the alternating Schwarz method has the following bound:

$$\|u - u^{k+1}\|_{L^\infty(\Omega)} \leq C_1^k C_2^k \|u - u^0\|_{L^\infty(\Omega)}, \tag{4.4}$$

However, as it is pointed out in [78], the convergence may be very slow ($C_1, C_2 \lessapprox 1$) in the case of small overlapping region. For a complete derivation of the proof of (4.4) see e.g. [63].

4.1.2 The multiplicative and additive forms

The Schwarz method can be formulated as a Richardson iteration, where the rate of convergence of the method is dependent upon the projection of the local solutions onto the internal boundaries of Γ_i. In order to show that this is the case, we will follow the arguments in [92].

When analysing the Schwartz methods, it is convenient to rewrite the elliptic problem on the variational form:

$$a(u,v) = \int_\Omega fv, \quad \forall v \in H^1(\Omega). \tag{4.5}$$

where

$$a(u,v) = \int_\Omega (\nabla u \cdot \nabla v)\,\mathrm{dx}, \tag{4.6}$$

is the bilinear form and v is a test function belonging to the space $H^1(\Omega)$. For the discretisation we introduce a triangulation T on Ω, such that the boundaries Γ_1, Γ_2 and $\partial\Omega$ do not cut through any of the elements of T. We also need to define a corresponding space V on T, such that $u, v \in V$. We let V be the space of continuous and piecewise linear functions on Ω, that vanishes on $\partial\Omega$, and denote V_1 and V_2 as the spaces for the corresponding functions on Ω_1 and Ω_2. Thus, we can also define the local bilinear forms $a_i(u,v) = \int_{\Omega_i} (\nabla u \cdot \nabla v)\,\mathrm{dx}$, where $u, v \in V_i$. The extension operator

$$R_i^T : V_i \to V, \quad i = 1,2, \tag{4.7}$$

is used to extend, by zeros, the local functions $v_i \in V_i$ on Ω_i to the space V on Ω. Note also that $R_i : V \to V_i$ are the usual restriction operators, that maps the functions $v \in V$ onto the sub-spaces V_i. We now define two orthogonal projections $P_i = R_i^T \tilde{P}_i$, $i = 1,2$, where $\tilde{P}_i : V \to V_i$ are defined by the relation

$$a_i(\tilde{P}_i u, v_i) = a(u, R_i^T v_i) \tag{4.8}$$

By using the variational form (4.5) and the definition of the orthogonal projection, we can now express the fractional steps of the alternating Schwarz method as a Richardson iteration,

$$\begin{aligned} u^{k+\frac{1}{2}} - u &= P_1(u^k - u), \\ u^{k+1} - u &= P_2(u^{k+\frac{1}{2}} - u), \end{aligned}$$

Thus, by combining the two fractional steps, the complete iteration step becomes

$$u^{k+1} - u = (I - P_2)(I - P_1)(u^k - u),$$

where the error propagation term

$$(I - P_2)(I - P_1) = I - (P_1 + P_2 - P_1P_2)$$

is on a multiplicative form. Such methods are denoted as *multiplicative Schwarz* methods. The error propagation term, as it is written here, is not symmetric, but it can be made symmetric if we consider an additional fractional step in the algorithm. By including a third fractional step, solve the problem on Ω_1 with the updated boundary condition from the solution on Ω_2, the Richardson iteration takes the symmetric form

$$(I - P_1)(I - P_2)(I - P_1).$$

Another useful algorithm is obtained by not updating the internal boundary conditions at each fractional step. Thus, all sub-domain problems can be solved independently and simultaneously in each iteration step. If we consider u_i as the solution on Ω_i, we can write the *additive Schwarz* algorithm as

$$\begin{aligned}
-\Delta u_1^{k+1} &= f && \text{in} && \Omega_1, \\
u_1^{k+1} &= u_2^k && \text{on} && \Gamma_1, \\
u_1^{k+1} &= 0 && \text{on} && \partial\Omega_1 \cap \partial\Omega, \\
-\Delta u_2^{k+1} &= f && \text{in} && \Omega_2, \\
u_2^{k+1} &= u_1^k && \text{on} && \Gamma_2, \\
u_2^{k+1} &= 0 && \text{on} && \partial\Omega_2 \cap \partial\Omega.
\end{aligned} \tag{4.9}$$

By a similar finite element analysis of this algorithm, we can also write (4.10) as a Richardson iteration

$$u^{k+1} - u = (I - (P_1 + P_2))(u^k - u),$$

where the error propagation term is on additive form. Both the additive and multiplicative Schwarz methods are naturally extended to many sub-domains. The main idea behind all of these domain decomposition algorithms is to decouple large systems of equation, and solve them locally by an iterative process. The Schwartz methods have shown to be very suitable for

parallel implementation. While the additive formulation is relatively straight forward to implement and run on parallel computers [15, 57, 88], the parallel implementation of the multiplicative Schwarz method is a bit more involved. Prior to the local computations, we need to identify disjoint sub-domains which do not share any overlapping region. This can be done by a colouring technique [92]. The sub-spaces V_i corresponding to the disjoint sub-regions (of the same colour) will then be orthogonal, and the sub-problems corresponding to these sub-spaces can be solved in parallel.

4.1.3 The Schwarz method as a preconditioner

As we have seen from the previous subsections, domain decomposition can be used to build iterative solvers for the elliptic problem, where the boundary value problem on Ω is replaced by boundary value problems on Ω_i, $i = 1, 2$. We will refer to problems on Ω as global problems and problems on Ω_i as local problems. Since the local problems in general are faster to compute, domain decomposition methods can also be used to build effective preconditioners for the global problem. In order to show this we will now consider the discrete global problem. By a suitable discretisation of Poisson's Equation (4.2) we arrive at the linear system of equations,

$$Au = b, \tag{4.10}$$

where A is a symmetric and positive definite matrix. In the following we will represent the discrete restriction operators R_i as the matrices, consisting of zeros and ones, which map the global vectors and matrices on Ω to corresponding local vectors and matrices on Ω_i. Similarly, we define the discrete extension operators R_i^T. We introduce the two local problems $\hat{A}_i u = b^*$, $i = 1, 2$, where $\hat{A}_i$ is the approximated local stiffness matrices given by,

$$\hat{A}_i = R_i A R_i^T, \quad i = 1, 2,$$

and is associated with the space V_i. The Schwarz methods are defined by the projection-like operators,

$$P_i = R_i^T \hat{P}_i, \quad i = 1, 2, \tag{4.11}$$

where $\hat{P}_i$ now is given by the discrete form of Equation (4.8);

$$v_i^T \hat{A}_i \hat{P}_i u = \left(R_i^T v_i\right)^T A u, \qquad \forall \qquad u \in V, \quad v \in V_i. \tag{4.12}$$

We note from (4.12) that we can write $A_i \hat{P}_i = R_i A$, and by using Equation (4.11) we observe that the projection-like operator can be expressed as a preconditioned operator for the global system matrix A,

$$P_i = \left(R_i^T \hat{A}_i^{-1} R_i \right) A.$$

If we consider the additive Schwarz methods, $P_a = \sum_i P_i$, we can write the one-level additive Schwarz preconditioner as

$$B_a = \sum_i \left(R_i^T \hat{A}_i^{-1} R_i \right). \tag{4.13}$$

For the multiplicative formulation (4.1.2), the preconditioner takes a more complicated structure, where $P_m = I - \prod_i (I - P_i)$. However, the preconditioner can still be represented by a recursive routine, that applies each local preconditioner to the corresponding restricted vector on Ω_i.

4.2 Substructuring methods

We will now direct our attention to the case of non-overlapping sub-domains. That is, the case where two neighbouring sub-domains at most share a set of common boundary nodes on the internal interface Γ. The domain decomposition methods for non-overlapping sub-domains are refered to as substructuring methods.

4.2.1 Non-overlapping sub-domains

Let us again take a look at the continuous Poissons problem given by Equation (4.2). Consider two non-overlapping sub-domains Ω_1 and Ω_2, with common boundary Γ, as illustrated in Figure 4.2. Under certain assumptions on the regularity of the right-hand side term f and the sub-domain boundaries, the problem can be formulated as

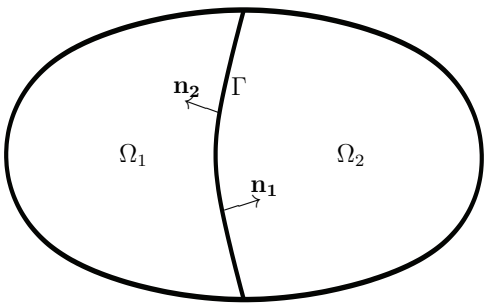


Figure 4.2: Non-overlapping sub-domains

$$
\begin{aligned}
-\Delta u_1 &= f && \text{in} && \Omega_1, && (4.14)\\
u_1 &= 0 && \text{on} && \partial\Omega_1 \cap \partial\Omega, && (4.15)\\
u_1 &= u_2 && \text{on} && \Gamma, && (4.16)\\
\frac{\partial u_1}{\partial \mathbf{n}_1} &= -\frac{\partial u_2}{\partial \mathbf{n}_2} && \text{on} && \Gamma, && (4.17)\\
u_2 &= 0 && \text{on} && \partial\Omega_2 \cap \partial\Omega, && (4.18)\\
-\Delta u_2 &= f && \text{in} && \Omega_2, && (4.19)
\end{aligned}
$$

where $\mathbf{n}_1$ and $\mathbf{n}_2$ are the normal vectors on Γ; $\mathbf{n}_1 = -\mathbf{n}_2$. The Equations (4.16) and (4.17) represent the *transmission conditions* for the solution on the boundary between the sub-domains. For second order elliptic problems where $u \in \mathrm{H}^1(\Omega)$, we require continuity of u and the normal derivative (flux) of u across Γ. In general, these conditions depend on the nature of the problem.

In the following we will only consider exact local solvers. Thus, the internal degrees of freedom can be eliminated prior to the iterative procedure. Similar to the overlapping Schwarz methods, the solution of this problem can be computed by an iterative algorithm for the interface equation. Moreover, it can be shown that the solution u on Γ has to satisfy the Steklov-Poincaré interface equation, see [9, 78]. In the next sub-sections we will consider iterative substructuring for the discrete system of linear equations. We will derive a Schur complement problem for the equation on the interface, which can be seen as a discrete approximation to the Steklov-Poincaré equation for the continuous problem.

4.2.2 Discrete formulations of the boundary equation

We will now derive two formulations for the discrete equation on the interface Γ. These formulations will form the basis for all iterative substructuring methods discussed in the remaining of this chapter.

We consider the equivalent discrete form of the Poisson's Equation, given by Equation (4.10). For non-overlapping sub-domains it is convenient to partition the degrees of freedom into those corresponding to the internal nodes ($u_I^{(i)}$) on each sub-domain Ω_i, and those related to the internal boundary nodes (u_B) on Γ. Thus, we can represent the coupled system of equations on the following matrix form:

$$\begin{bmatrix} A_{II}^{(1)} & 0 & A_{IB}^{(1)} \\ 0 & A_{II}^{(2)} & A_{IB}^{(2)} \\ A_{BI}^{(1)} & A_{BI}^{(2)} & A_{BB} \end{bmatrix} \begin{bmatrix} u_I^{(1)} \\ u_I^{(2)} \\ u_B \end{bmatrix} = \begin{bmatrix} b_I^{(1)} \\ b_I^{(2)} \\ b_B \end{bmatrix}. \tag{4.20}$$

Here, $u_B = u_B^{(1)} = u_B^{(2)}$ accounts for the continuity of the solution u_B on the internal boundary Γ, while the right hand side vector $b_B = b_B^{(1)} + b_B^{(2)}$, corresponds to the source terms on Γ. The global system matrix A can been constructed by subassembling the local system matrices $A^{(i)}$ corresponding to the discretisation on Ω_i, $i = 1, 2$. The local problems have the following form,

$$\begin{bmatrix} A_{II}^{(i)} & A_{IB}^{(i)} \\ A_{BI}^{(i)} & A_{BB}^{(i)} \end{bmatrix} \begin{bmatrix} u_I^{(i)} \\ u_B^{(i)} \end{bmatrix} = \begin{bmatrix} b_I^{(i)} \\ b_B^{(i)} + \lambda^{(i)} \end{bmatrix} \quad i = 1, 2. \tag{4.21}$$

Here $\lambda^{(i)} = A_{BI}^{(i)} u_I^{(i)} + A_{BB}^{(i)} u_B^{(i)} - b_B^{(i)}$ is the residual on the boundary due to the decoupling, which can be regarded as the discrete normal flux across Γ. The transmission conditions, given by the Equations (4.16) ans (4.17) can now by represented discretely by the following relationships

$$u_B = u_B^{(1)} = u_B^{(2)}, \tag{4.22}$$

$$\lambda_B = \lambda_B^{(1)} = -\lambda_B^{(2)}. \tag{4.23}$$

By a direct sum of the two local problems (4.21), we see that the flux terms cancel each other out and we retain the coupled formula given by Equation (4.20). Note also that the two upper rows of (4.20) represent the Dirichlet problems on Ω_1 and Ω_2, respectively.

We can now derive an algebraic equation for the solution u_B on Γ. By a simple block Gaussian elimination of (4.20), we express the system of equations on reduced form

$$\begin{bmatrix} A_{II}^{(1)} & 0 & A_{IB}^{(1)} \\ 0 & A_{II}^{(2)} & A_{IB}^{(2)} \\ 0 & 0 & S \end{bmatrix} \begin{bmatrix} u_I^{(1)} \\ u_2^{(2)} \\ u_B \end{bmatrix} = \begin{bmatrix} b_I^{(1)} \\ b_I^{(2)} \\ g \end{bmatrix}, \tag{4.24}$$

where

$$S = A_{BB} - A_{BI}^{(1)} \left(A_{II}^{(1)} \right)^{-1} A_{IB}^{(1)} - A_{BI}^{(2)} \left(A_{II}^{(2)} \right)^{-1} A_{IB}^{(2)} \tag{4.25}$$

and

$$g = b_B^{(1)} - A_{BI}^{(1)} \left(A_{II}^{(1)}\right)^{-1} b_I^{(1)} + b_B^{(2)} - A_{BI}^{(2)} \left(A_{II}^{(2)}\right)^{-1} b_I^{(2)} \tag{4.26}$$

are the Schur complement matrix and the modified right-hand side term on the boundary, respectively. The *Schur complement system*,

$$Su_B = g, \tag{4.27}$$

can be regarded as an equation for the trace of the exact solution u_B on Γ, where S is the discrete approximation to the Steklov-Poincaré operator [78]. Similar to A, the Schur matrix may also be assembled by the local Schur matrices constructed on each sub-domain Ω_i, which are given by

$$S^{(i)} = A_{BB}^{(i)} - A_{BI}^{(i)} \left(A_{BB}^{(i)}\right)^{-1} A_{IB}^{(i)}, \quad i = 1, 2. \tag{4.28}$$

Note that $S^{(i)}$ in general forms a full matrix, where all the boundary unknowns are coupled together through the internal of Ω_i.

We can also derive a dual formulation for the discrete flux on Γ. If we consider a block factorisation of the two local problems (4.21), we can write the local interface equations as

$$u_B^{(i)} = \left(S^{(i)}\right)^{-1} \left(g^{(i)} + \lambda^{(i)}\right), \quad i = 1, 2. \tag{4.29}$$

concerning the solution $u_B^{(i)}$, flux $\lambda_B^{(i)}$ and Schur complement $S^{(i)}$ on Γ. In stead of neglecting λ_B we can now use condition (4.23) and neglect u_B. By subtracting the second equation in (4.29) from the first and using the continuity conditions on Γ we arrive at the following *dual formulation*:

$$F\lambda = d \tag{4.30}$$

where

$$\begin{aligned} F &= \left(S^{(1)}\right)^{-1} + \left(S^{(2)}\right)^{-1}, \\ d &:= -\left(S^{(1)}\right)^{-1} g^{(1)} + \left(S^{(2)}\right)^{-1} g^{(2)}. \end{aligned}$$

The two formulations (4.27) and (4.30) forms the basis for the primal and dual iterative substructuring algorithms, to be discussed in the following subsections. While these algorithms involve the Schur complement, they are also known as Schur complement methods.

The Schur complement system is usually better conditioned then the original system. In fact, it can be shown that the condition number of the

Schur complement matrix S is bounded by the condition number of the original system matrix A [11].

The multiplication of the Schur complement S with a vector $\mathbf{v}$, corresponds to solving a local Dirichlet problem on each sub-domain Ω_i. Note that this can be done without explicitly forming the Schur complement. The multiplication of F (subassembly of the inverse of the local Schur matrices) with a vector $\mathbf{v}$ is equivalent with solving a Neumann problem on each sub-domain. The two primal and dual formulations are the starting points for a number of substructuring algorithms, which consists of solving local Dirichlet and Neumann problems in a certain order. The most popular of these methods will be presented in the following.

4.2.3 Primal algorithms

We will now discuss two algorithms for solving Equation (4.27); the Dirichlet-Neumann algorithm and the Neumann-Neumann algorithm. For simplicity we first consider the case of two non-overlapping sub-domains Ω_1 and Ω_2, as shown in Figure 4.2.

Dirichlet-Neumann

The Dirichlet-Neumann algorithm consists of solving one Dirichlet problem and one Neumann problem in each iteration. From a given initial solution $u_B^{(0)}$ on the boundary Γ, the Dirichlet-Neumann algorithm can be described by the following steps:

1. Solve the Dirichlet problem on Ω_1, with the given Dirichlet boundary data $u_B^{(k)}$:

$$u_I^{(1)} = \left(A_{II}^{(1)}\right)^{-1} \left(b_I^{(1)} - A_{IB}^{(1)} u_B^k\right).$$

2. Compute the corresponding flux λ_B on Γ corresponding to Ω_1:

$$\lambda_B^{k+\frac{1}{2}}|_{\Omega_1} = A_{BI}^{(1)} u_I^{k+\frac{1}{2}} + A_{BB}^{(1)} u_B^k - b_B^{(1)}.$$

3. Solve the resulting Neumann problem (4.21) on Ω_2, with the corresponding flux condition. Note that $\lambda_B^{k+\frac{1}{2}}|_{\Omega_2} = -\lambda_B^{k+\frac{1}{2}}|_{\Omega_1}$ on Γ.

$$\hat{u}_B^{k+1} = \left(S^{(2)}\right)^{-1} \left(g^{(2)} - \lambda_B^{k+\frac{1}{2}}|_{\Omega_2}\right)$$

4. Update the solution on the boundary:

$$u_B^{k+1} = \theta \hat{u}_B^{k+1} + (1-\theta)u^k,$$

where $\theta \in [0, \theta_{max}]$ is a relaxation parameter. The first two steps can be combined. If we eliminate $u_I^{(1)}$, we find that the flux across Γ is given from the Dirichlet solution as $\lambda_B^{k+\frac{1}{2}}|_{\Omega_1} = -(g_B^{(1)} - S(1)u_B^k)$. By substituting the expression for the flux into the third step, we observe that the algorithm can be rewritten as a preconditioned Richardson iteration $(u_B^{k+1} - u_B^k) = B\theta(g - Su_B^k)$ for the Schur complement system,

$$u_B^{k+1} - u_B^k = \theta B_{DN}(g - Su_B^k) \tag{4.31}$$

where $B_{DN} = \left(S^{(2)}\right)^{-1}$ is the preconditioner for the Schur complement matrix S. Note that, if $S^{(1)} = S^{(2)} = \frac{1}{2}S$, the Dirichlet-Neumann preconditioner will be exact.

Neumann-Neumann

The Neumann-Neumann algorithm consists of solving two Dirichlet and two Neumann problems in each iteration. Starting from a given initial boundary value u_B^0, the algorithm can be written in terms of the same three steps:

1. Solve a Dirichlet problem on both sub-domains Ω_1 and Ω_2, with prescribed Dirichlet boundary data $u_B^{(k)}$:

$$u_I^{(i)} = \left(A_{II}^{(i)}\right)^{-1} \left(b_I^{(i)} - A_{IB}^{(i)} u_B^k\right), \quad i = 1, 2.$$

2. Compute the corresponding fluxes $\lambda_B^{k+\frac{1}{2}}$ on Γ corresponding to Ω_i, $i = 1, 2$:

$$\lambda_B^{k+\frac{1}{2}}|_{\Omega_i} = A_{BI}^{(i)} u_I^{k+\frac{1}{2}} + A_{BB}^{(i)} u_B^k - b_B^{(i)}.$$

3. Solve the resulting Neumann problem (4.21) on Ω_i, $i = 1, 2$, with the corresponding flux conditions.

$$\hat{u}_B^{k+1}|_{\Omega_i} = \left(S^{(i)}\right)^{-1} \left(g^{(i)} - \lambda_B^{k+\frac{1}{2}}|\Omega_i\right)$$

4. Update the solution on the boundary:

$$u_B^{k+1} = \theta \hat{u}_B^{k+1} + (1-\theta)u^k,$$

In the same way as for the Dirichlet-Neumann algorithm, the Neumann-Neumann algorithm can be expressed as a preconditioned Richardson iteration:

$$u_B^{k+1} - u_B^k = \theta B_{NN}(g - Su_B^k) \tag{4.32}$$

where $B_{NN} = \left(S^{(1)}\right)^{-1} + \left(S^{(2)}\right)^{-1} = F$. Thus, the dual formulation acts as a preconditioner for the Schur complement system. Note that this is an additive Schwartz preconditioner, in which case all the sub-domain problems can be solved simultaneously and in parallel.

4.2.4 Dual algorithms

In a similar fashion we can construct algorithms for the dual formulation (4.30). The dual equivalent of the algorithms discussed above are the Neumann-Dirichlet and the dual Neumann-Neumann algorithm. Both of these algorithms can also be written in terms of preconditioned Richardson iterations,

$$\lambda_B^{k+1} - \lambda_B^k = B_F\theta\left(d - F\lambda_B^k\right) \tag{4.33}$$

for the dual formulation. In case of the Neumann-Dirichlet algorithm we solve one Neumann problem on Ω_1, with a repeating Dirichlet problem on Ω_2. Thus the preconditioner for the Dirichlet-Neumann methods is given by $B_{ND} = S^{(2)}$. Similar to the Neumann-Neumann method the dual version solves individual sub-problems on Ω_1 and Ω_2. Thus, the preconditioner for the dual Neumann-Neumann method takes the form: $B_{dNN} = S^{(1)} + S^{(2)}$.

4.3 Primal iterative substructuring methods for many sub-domains

In the previous section we have seen how the Schur complement S can be used to construct one-level algorithms for solving the two-domain problem. In this section we will extend the primal substructuring methods to many sub-domains. In general, the extension from two to many sub-domains can be viewed as solving a collection of many two-domain problems. However, as we will see, the construction of efficient iterative substructuring methods for the case of many sub-domains is not so straight forward.

4.3.1 Extension to many sub-domains

The Neumann-Neumann and dual Neumann-Neumann algorithms are both naturally extended to many sub-domains. All the local operations are independent and can be solved in parallel. The Neumann-Dirichlet and Dirichlet-Neumann methods can also be extended, but we need to keep track of which sub-domains that are coupled to each other. For structured grids, this can be done by using a colouring technique as discussed in Section 4.1.2.

For two and three dimensional problems we will have to deal with *floating sub-domains*. That is, sub-domains which are not connected to the global Dirichlet boundary $\partial\Omega$. The local system matrix $S^{(i)}$ corresponding to a floating sub-domain is singular and can not be directly inverted. Thus, in order to solve the pure Neumann problems corresponding to the inverse of $S^{(i)}$ we need to apply some kind of regularised inverse or preconditioner for the singular matrix.

While the Schur complement matrix S is much denser then A, the explicit calculation of S is expensive. The aim of primal iterative substructuring methods is to solve the Schur complement system S without explicitly forming the matrix S. Thus, we formulate our substructuring methods as Schur complement preconditioners for e.g. the Krylov-type algorithms. Various choices of local preconditioners will now be discussed in the next subsection.

4.3.2 Local Schur complement preconditioners

We consider the construction of local Schur complement preconditioners for the Neumann-Neumann method, in the case of many sub-domains. Similar approaches can also be applied to other variants of substructuring methods. The local preconditioner for the Neumann-Neumann method is analogous to the choice of local boundary conditions used in the local upscaling methods discussed in the Chapter 3. Moreover, the multiscale control-volume methods which have been the main focus of this thesis can be categorised as special types of Neumann-Neumann preconditioners. The localisation approximations applied in the multiscale framework can be seen as local Schur complement preconditioners.

For the general case of 3-dimensional elliptic problems, we need to introduce some additional notation on the geometry. We define the interface as $\Gamma = \cup_{i \neq j} \left(\partial\Omega_i \cap \partial\Omega_j \right)$. We further refer to sub-domain *faces* as the subset of Γ that is only shared by two sub-domains, sub-domain *edges* as the subset of Γ that is shared between more than two sub-domains and sub-domain *vertices* as the union of the endpoints of the sub-domain edges. Thus, we can sub-divide the degrees of freedom on Γ, $u_B = \begin{bmatrix} u_F & u_E & u_V \end{bmatrix}^T$, where u_F, u_E

and u_V are the degrees of freedom corresponding to sub-domain faces, edges and vertices, respectively. Thus, we can formulate the Schur complement problem (4.27) as

$$\begin{bmatrix} S_{FF} & S_{FE} & S_{FV} \\ S_{EF} & S_{EE} & S_{EV} \\ S_{VF} & S_{VE} & S_{VV} \end{bmatrix} \begin{bmatrix} u_F \\ u_E \\ u_V \end{bmatrix} = \begin{bmatrix} g_F \\ g_E \\ g_V \end{bmatrix}. \tag{4.34}$$

Recall form Section 4.2.3 that the Neumann-Neumann method requires the solution of a Neumann problem, involving $\left(S^{(i)}\right)^{-1}$ multiplied with a vector. The multiplication $\mathbf{x} = \left(S^{(i)}\right)^{-1}\mathbf{y}$ can be found by solving

$$S^{(i)}\mathbf{x} = \mathbf{y} \tag{4.35}$$

The aim of the local preconditioner, is to approximate the Schur complement S, such that the local problem (4.35) is fast to compute. In the following we will discuss two local Schur complement preconditioners, which has been successfully applied to the elliptic problem for flow in porous media.

Tangential component approximation

The tangential component (TC) approximation is a simple way of neglecting the global couplings, while preserving the local flow on Γ. The local Schur complement $S^{(i)}$ in (4.28) consists of a local term $A_{BB}^{(i)}$ and a global term $-\sum_i A_{BI}^{(i)}\left(A_{II}^{(i)}\right)^{-1}A_{IB}^{(i)}$. The TC approximation simply neglects the global term and approximates the local Schur complement by a sparse matrix $S_{TC}^{(i)} = \tilde{A}_{BB}^{(i)}$, which has the same sparsity pattern as $A_{BB}^{(i)}$. While the local matrix $A_{BB}^{(i)}$ can be significantly diagonal dominant, due to transversal flow, this matrix is not a good approximation to $S^{(i)}$. The TC approximation neglects the contributions to the flow normal to the boundary, corresponding to the elements of $A_{BI}^{(i)}$. In the preconditioner $\tilde{A}_{BB}^{(i)}$ we have neglected the part of the diagonal elements of $A_{BB}^{(i)}$. The TC approximation is algebraic, simple to implement and can be applied to general geometries and dimensions. The preconditioner may however be sensitive to variations in material constants on Γ.

The Multiscale Finite Volume Method (MSFV) applies a similar tangential flow approximation, denoted the reduced boundary condition [53]. This boundary condition is shown to be similar to the TC approximation for Cartesian grids, and the method is analogous to a special type of two-level additive Schwartz preconditioner.

Interface probing

A second sparse approximation to the Schur complement can be obtained by using the interface probing technique. This is also a completely algebraic technique, meaning that it only considers the local matrices $S^{(i)}$. By a careful choice of probing vectors $\{\mathbf{v}^j,\ 1 \leq j \leq d\}$, a low-bandwidth ($d-$diagonal) matrix $S_P^{(i)}$ is constructed such that

$$S_P^{(i)}\mathbf{v}^j = S^{(i)}\mathbf{v}^j = \mathbf{w}^j, \quad j = 1, \ldots, d. \tag{4.36}$$

Thus, the approximated Schur complement has the same application on $\mathbf{v}^j$ as the true Schur complement, and these d degrees of freedom are captured within the preconditioner. An important note is that the Schur complement $S^{(i)}$ do not need to be formed explicitly. The application of $S^{(i)}$ with the probing vectors is carried out by multiplying $\mathbf{v}^j$ through all the local matrix-components $S^{(i)}$. This is equivalent with solving d local Dirichlet problems. Usually $d = 3$, which results in a tri-diagonal approximation of $S^{(i)}$. Even though the resulting preconditioner is on local form, both the local and global term of the Schur complement are approximated, which makes the probing approximation more robust with respect to irregular grids and variations in the coefficients of K.

Other interface approximations

Several other choices of local Schur complement preconditioners have been proposed and many of them are designed for the model elliptic problem with constant coefficients.

In the case of uniform rectangular mesh on Ω, using piecewise linear functions for the discretisation, the Schur complement S, and its eigenvalues can be explicitly constructed in terms of the sine transform. Based on these analytical expressions for the Schur complement and eigenvalues, Dryja [27] and later Golub and Mayers [41] proposed two local Schur complement preconditioners which are spectral equivalent to the true Schur complement. While these preconditioners are superior for solving the Poisson's problem on uniform rectangular grids, their performance are in general highly dependent on the specific problem to be solved.

The Toepliez approximation is another preconditioner which takes advantage of the special structure of the Schur complement. [17]. This preconditioner is motivated by the observation that the local Schur complement matrix for homogeneous problems usually has a banded structure, where each diagonal band has small variations. Thus the local Schur complement

matrix is close to being a Toepliez matrix, for which the inverse matrix can be efficiently calculated. Other local Schur complement preconditioners also include the change of basis, from nodal basis to hierarchical basis [89] or multilevel basis (PBX) [91].

Finally, it should be repeated that domain decomposition is, for most cases, used as a preconditioner for the Krylow subspace methods (e.g. conjugated gradient methods or the generalised minimal residual (GMRES) method), where the goal is to bound the condition number.

In general, one-level preconditioners have a poor performance for elliptic problems where the condition number is proportional to $H^{-2}\left(1+\log\frac{H}{h}\right)^2$. Here $H<1$ is the sub-domain mesh size and $h<H$ is the fine-scale mesh size. Due to the term H^{-2}, the condition number will deteriorate with increasing number of sub-domains, or refinement of the grid.

4.3.3 Scalable preconditioners

Domain decomposition methods are most often used as preconditioners for solving large linear systems. Thus, it is crucial to bound the condition number for increasing number of sub-domains. The preconditioner is said to be *scalable* if the condition number is bounded. That is, if the condition number is close to constant when we refine the grid and keep the ratio between the fine-scale and coarse-scale mesh size $(\frac{H}{h})$ fixed.

We can illustrated this by a small scaling exercise. Let us consider a the single sub-domain $K\subset\mathbb{R}^n$ of size $H<1$, which is bounded and where the boundary ∂K is Lipschitz continuous. We further assume that there exist two functions $u,v\in H^1(K)$ such that,

$$|v|^2_{H^1(K)}\leq C\|u\|^2_{H^1(K)}. \tag{4.37}$$

The two functions u and v are bounded by a constant C that only depends on K. By the transformation $x=Hx^*$ from K to the domain $\tilde{K}$ with the same shape and with unit mesh size, we arrive at the new bound

$$|v|^2_{H^1(K)}\leq\frac{\tilde{C}}{H^2}\|u\|^2_{H^1(K)}, \tag{4.38}$$

where the constant $\tilde{C}$ only depends on the shape, and not the size, of K. In order to remove the dependence of H^{-2}, we need to impose a Poincaré inequality of the type $\|u\|^2_{L^2(K)}\leq\tilde{C}_2H^2|u|^2_{H^1(K)}$. Such a bound on the L^2-norm can found to hold, if the solution u has a zero mean over K [92]. For one-level methods, if K is not connected with the global Dirichlet boundary, we can not in general impose such a bound. For two-level methods, where the coarse

space is able to represent at least constant solutions exactly, such a bound may be found, and we can construct scalable preconditioners. That is why we need to consider multilevel methods, when we build domain decomposition preconditioners for solving large elliptic problems. For more complicated elliptic problems, e.g with oscillating coefficients, we need to require additional properties of the coarse space. This is discussed more in detail in Section 4.4.2.

4.4 Two-Level Substructuring Methods

We have now seen that the rate of convergence for one-level domain decomposition preconditioners deteriorates in the case of many sub-domains. In order to make the rate of convergence independent upon the size of the problem, we need to include a global coarse space component.

4.4.1 Coarse scale correction

From the previous section we have learned that iterative substructuring methods, when applied to large linear systems, should be used as preconditioners. Usually, these methods are applied as preconditioners for the Krylov methods, i.e. the conjugate gradient (CG) and generalised minimal residual (GMRES) method. The aim is to minimize the error function e, given by the relation $Se = r$, but since the error function involves the inverse of the matrix S, it is more convenient to work with the residual. The starting point for all the Krylov-type algorithms is the residual $r = g - Su$, on the fine-scale. The residual can be distributed to all sub-domains and minimized in an iterative procedure such that the updated solution takes the following form:

$$u^{k+1} = u^k + B_F r^k \tag{4.39}$$

In this equation B_F represents a one-level preconditioner (additive or multiplicative) on the fine scale, consisting of local solvers or preconditioners. Experience shows that many local solvers are efficient for reducing the high-frequency oscillations of the residual, but they are slow in reducing the low-frequency oscillations. This is due to the slow transportation speed of global information. If we can approximate the residual r_C on a coarser space, we can accelerate the convergence of the low-frequency error. Suppose that R_C represents the restriction operator from the fine-scale degrees of freedom to the coarse scale degrees of freedom and that A_C is an operator on the coarse scale. Then

$$\begin{aligned} u^{k+\frac{1}{2}} &= u^k + R_C^T A_C R_C r^k \\ u^{k+1} &= u^{k+\frac{1}{2}} + B_F r^{k+\frac{1}{2}} \end{aligned}$$

represents a simple two-step, two-level algorithm for minimizing the residual. We can write $B_C = R_C^T A_C^{-1} R_C$ as the coarse scale preconditioner and write the two-level iterative algorithm on the following compact form,

$$u^{k+1} = u^k + (B_C + B_F - B_F S B_C) r^k. \tag{4.40}$$

We clearly see that the full preconditioner $B_{2,MS} = B_C + B_F - B_F S B_C$ takes the form of a two-level, multiplicative Schwarz operator. In the same manner we can also construct two-level additive Schwarz preconditioners.

4.4.2 Space decomposition

The building blocks of an efficient two-level preconditioner consist of many local preconditioners and a suitable coarse-scale solver. A global coarse-scale component is essential for bounding the number of iterations for large elliptic problems, and the development of suitable coarse spaces has been one of the main topic of research within domain decomposition during the last couple of decades. See e.g. [93, 71] for an historical overview.

We will define V_0 as our *coarse space* and assume the following space decomposition:

$$V = R_0^T V_0 + \sum_{i=1}^{N} R_i^T V_i. \tag{4.41}$$

The operators R_i^T, $i = 1, \ldots, n$, are the usual extension operators defined in Section 4.1.3. The extension operator $R_0^T : V_0 \to V$ represents the interpolation from the coarse scale degrees of freedom to the fine-scale degrees of freedom and is also refered to as the coarse-scale basis function. The coarse-scale basis function is usually defined somewhat different from the local extension operators R_i; see Section 4.4.3 for more details.

The coarse space can be related to the local spaces in several different ways, leading to a number of different substructuring methods. In addition to the two-level additive and two-level multiplicative methods, we may construct hybrid Schwarz methods, where the coarse and local operators are treated in different ways.

It is the choice of V_0 and R_0^T which really defines the efficiency of the domain decomposition preconditioner. The construction of these parameters has been the main focus of all the papers [83, 84, 82] included in this thesis.

There are two principle properties, which has shown to be important when constructing the coarse space. The first principle states that all the sub-spaces $\{V_i,\ 0 \leq i \leq N\}$ should form a *stable splitting* of V. Thus, for every $v \in V$

$$v = \sum_{i=0}^{N} R_i^T v_i, \quad \text{where} \quad v_i \in V_i, \tag{4.42}$$

such that

$$\sum_{i=0}^{N} a_i(v_i, v_i) \leq C_0 a(v, v). \tag{4.43}$$

Here $a(\cdot\,,\cdot\,)$ is the same inner-product as used in Section 4.1.2. The principle is also refered to as the bounded energy condition, and it is important that the coarse space satisfies this condition. The second important principle is called the *null-space property*, which states that the coarse space should include the null-space of all the bilinear forms $a_i(\cdot\,,\cdot\,)$ [68]. For the scalar second order elliptic problems, the null-spaces are given by constant functions. This condition is directly related to the Poincaré inequality, discussed in Section 4.3.3, and can be thought of as fixing the constant for the inexact local Neumann problems.

The coarse scale operator $A_C \in V_0$ is often defined in terms of a Galerkin approximation $A_C = R_0 A R_0^T$, where the interpolation operator R_0^T consisting of coarse scale basis functions defines the coarse space. However, the coarse-scale operators can be defined quite freely, on some coarse mesh which may or may not coincide with the sub-domain mesh. In general, it is preferable that the coarse-scale elements are comparable in size with the sub-domains [92].

4.4.3 Coarse-scale basis functions

The coarse-scale basis functions span out the solution on the coarse scale. Thus, it is important that these functions capture the principle flow on the underlying fine-scale. In order for the domain decomposition preconditioner to be efficient, it is also important that the coarse scale basis functions satisfy the bounded energy condition and the null-space property. In the following we will discuss the piecewise constant basis function and the discrete harmonic basis function.

The space of piecewise constant basis-functions is used in the Balancing domain decomposition (BDD) preconditioner [69]. The method satisfies the null-space property for scalar elliptic problems. The advantage of this rather simple coarse space, is that it is applicable to very general geometries and the algorithm is rather simple to implement. The dual equivalent to BDD is the FETI algorithm [35]. It uses the same coarse space as BDD, is entirely algebraic, and applies to the dual formulation (4.29). These two methods capture the constant solution of the coarse scale, in which case it is possible to make the condition number independent upon the size of the problem.

Most domain decomposition preconditioners are efficient for solving large scalar elliptic problems, when the scalar coefficient a is constant on each sub-domain. For varying coefficients inside the sub-domains, it is not straight forward to satisfy the bounded energy condition. The BDDC (balancing domain decomposition by constraints) method of Dohrmann is a two-level Neumann-Neumann preconditioner, based on constrained energy minimization. The coarse space consists of piecewise discrete harmonic functions. The coarse preconditioner is additive rather then multiplicative, which makes it possible to choose a different bilinear form for the coarse space. Consequently, the coarse-scale operator of the BDDC preconditioner is less dense.

The coarse-scale basis-functions Φ_i are computed as a constrained minimization problem, minimizing the energy expression

$$E = \Phi_i^T A^{(i)} \Phi_i / 2. \tag{4.44}$$

The coarse scale operator $A_C^{(i)}$, corresponding to each sub-domain is further determined by the basis-functions, such that

$$A_C^{(i)} = \Phi_i^T A^{(i)} \Phi. \tag{4.45}$$

The dual equivalent to the BDDC is the FETI-DP [34], which was developed prior to the BDDC. The FETI-DP preconditioner uses primal unknowns on the corners and dual Lagrange multipliers for the continuity of the solution on the remaining part of the boundary Γ. A special version of the FETI-DP algorithm is called the P-FETI-DP [38] and can be shown to be similar to the BDDC method. In fact, all these methods have shown to be related and they give more or less the same eigenvalues. Thus, they are also comparable w.r.t. convergence rates [62, 70].

Discrete harmonic basis functions are also used in the multiscale methods. These methods are like domain decomposition preconditioners, but applied to elliptic problems with highly varying coefficients, i.e. porous media problems. The aim of the multiscale methods are to capture as much of the fine-scale information as possible inside the coarse-scale basis functions, and to use

the coarse-scale solution after only one iteration (upscaling) to recalculate an approximate fine-scale flow field which can be used for efficient reservoir simulations. These methods are the focus of the next Chapter 5.

Chapter 5

Multiscale Methods

In the previous two chapters we have investigated two different approaches for efficiently solving the elliptic problem for flow in porous media. By upscaling, we reduce the dimension of our fine-scale problem and seek a coarse-scale solution which have equivalent properties to the some pre-defined average of the fine-scale solution. In domain decomposition, we construct consistent interpolation functions between the scales and use the coarse-scale solution as a preconditioner to efficiently solve the fine-scale problem. In this chapter we introduce the multiscale methods, which can be seen as combined techniques of upscaling and domain decomposition.

We start by giving a general background to the multiscale methods, before focusing on the multiscale methods applied to the control volume formulation. We will show that these methods can be formulated as both upscaling methods and multiscale domain decomposition preconditioners.

5.1 Introduction to the multiscale methods

The multiscale methods for solving multi-phase flow in porous media are quite recent compared to upscaling and domain decomposition, but during the last 10-15 years these methods have gained an increasing interest. The main motivation is that while the fluid velocities in the reservoirs can have large variations on the fine-scale due to rapid variations in the fine-scale permeability, the pressure is in general smoother and may be well represented on an integrated coarse scale. Thus, when applied to e.g. an IMPES formulation, the elliptic pressure equation is solved primarily on the coarse scale, while the transport of fluids is carried out on the fine-scale.

The solution of the elliptic problem is often the main bottleneck of the simulations. The multiscale methods contribute to a substantial speed-up of

the fine-scale simulations by solving the elliptic equation for the pressure on the coarse scale. Thus, the multiscale method can be seen as a compromise, between pure upscaling and solving the entire problem on the fine-scale.

If we consider the case of two-phase flow, the mobility ratio and relative permeabilities will change as the saturation front evolves in time. Thus, the flow conditions and the premises for single-phase upscaling also changes. In this case, the updated fine-scale flow field from the multiscale simulation can be applied to recompute the basis functions used for the upscaling. Hence, the multiscale methods can also be seen as an alternative to two-phase upscaling.

5.1.1 The multiscale finite element method

The multiscale finite element method (MsFEM) for elliptic problems in porous media was first proposed in [49] as an upscaling technique. The objective at that time, was that the elliptic problem is too computationally expensive to be resolved at the fine-scale. Hence, upscaling is needed to reduce the large degrees of freedom of the fine-scale elliptic problem. However, we know that the small scale heterogeneities can have a significant influence on the fluid flow on the fine scale. By recalculating the fine-scale pressures, the transport of fluids can be carried out on the fine-scale.

The multiscale relation is written as,

$$u_{fine} = \sum_{i}^{n} \phi^i u^i_{coarse}, \tag{5.1}$$

for $i = 1, \ldots, n$, where n is the number of coarse nodes. The local coarse-scale basis functions ϕ^i express the coupling between the coarse scale and the fine scale. The aim of MsFEM is to capture fine-scale information of the elliptic operator into the local coarse-scale basis functions, thus, enforcing the correct fine-scale effects onto the coarse-scale. The MsFEM method indicated improved accuracy of the elliptic solution on the coarse scale, compared to standard upscaling methods for problems involving strong sub-scale heterogeneities. Furthermore, the approximate fine-scale solution recovered from the local basis functions showed comparable accuracy to the fully resolved solution for many problems. Especially flux dependent properties, like the water-cut, show fairly good match with the full-field fine-scale model. In the homogeneous limit, where the periodicity of the fine-scale heterogeneities goes to zero, the multiscale method gives the correct effective solution on the coarse-scale.

The key element of the multiscale method is the local basis-functions. We will here describe the MsFEM basis functions for a 2D regular Cartesian

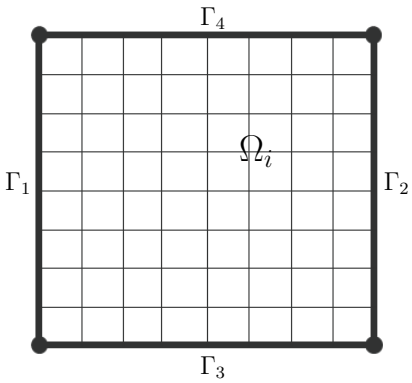


Figure 5.1: Illustration of a 2D Cartesian coarse block for the MsFEM method

grid, with coarse elements $\Omega_i \subset \Omega$ as illustrated in Figure 5.1. The local basis function ϕ^i used in the multiscale method is defined on Ω_i such that

$$-\nabla\cdot\left(\mathbf{K}\nabla\phi^i\right) = 0 \qquad \text{in} \qquad \Omega_i \subset \Omega. \tag{5.2}$$

We construct one local MsFEM basis function ϕ^i_j $(j = 1, \ldots, 4)$ for each vertex node of the coarse element Ω_i. For a 2D rectangular coarse element and isotropic permeability $\mathbf{K}$, the boundary condition on the horizontal boundaries Γ_3 and Γ_4 is given by the reduced 1D problem

$$-\frac{\partial}{\partial x}k_x\frac{\partial\phi^i}{\partial x} = 0. \tag{5.3}$$

Similarly we can construct the boundary condition for the vertical 1D boundaries. The uniqueness and continuity of ϕ^i is guaranteed through the conditions on the corner nodes x_j, where $\phi^i(x_j) = \delta_{ij}$. The MsFEM method can also be extended to 3D problems, however, only for Cartesian grids with isotropic permeability. The basis functions satisfy the following interpolation properties,

$$0 \leq \phi^i \leq 1, \tag{5.4}$$

$$\sum_i \phi^i = 1. \tag{5.5}$$

Note that, in the case of homogeneous permeability the oscillating basis functions of MsFEM reduces to the standard linear finite element basis functions.

The MsFEM method is often considered to be the first of what we will denote the multiscale methods for elliptic problems involving flow in porous media. Many of the key ideas were formulated in the pioneering works of Hou and Wu ([49] and [50]). In contrast to traditional upscaling methods, the multiscale methods do not assume scale-separation or periodicity of the porous rock. In fact, one of the main motivations is to capture the resonance error occurring at the sub-scale between the geological fine scale and the computational coarse scale. This is known as the *oversampling* technique.

5.1.2 The oversampling technique

An error is introduced on the boundary between the coarse elements due to the approximation induced in the local boundary conditions. The true boundary condition can only be found by actually solving the fine-scale problem. As a result, the error of the approximate solution, computed by the multiscale method, is always largest close to the local boundaries. Another important observation is that while the small-scale heterogeneities are well captured within the local-basis functions and the large scale heterogeneities are captured on the coarse scale, the intermediate-scale heterogeneities on the size of the coarse element corresponds to the largest error. This is known as the resonance effect.

The answer to these problems, as it is proposed in [50], is oversampling. That is, for each coarse element Ω_i we compute the local basis functions ψ_j^i on an extended local domain $\Omega_i' \subset \Omega_i$. Boundary conditions are then placed on Ω_i', away from the boundary of the coarse element Ω_i, i.e., moving the main approximation error further away from the coarse element. The local basis functions ϕ_j^i on Ω_i are then found by scaling the part of ψ_j^i which belongs to Ω_i. It can be shown that the interpolation properties (5.4) and (5.5) still hold on Ω_i with oversampling. However, two basis functions ϕ_j^i and ϕ_j^{i+1} corresponding to the same vertex node do not have to coincide on the boundary. I.e. the elements are allowed to be non-conformal.

Another way of thinking about oversampling, is that the local boundary conditions are moved closer to the global boundary, which for most elliptic problems are known. Thus, oversampling is a way of incorporating global information into the calculation of the local basis functions. In the limit, as the entire global domain is chosen as the oversampling region, we can compute the exact local boundary conditions.

In [30] they apply the local-global upscaling technique on the multiscale method. Thus, they apply a global approximation of the fine-scale solution in order to more precisely calculate the boundary conditions for the local basis-function problems. Furthermore, Efendiev et. al. [32] uses the ex-

act fine-scale solution of the first time step to more accurately represent the multiscale basis-functions which are being applied for the entire multiscale simulation. Both of these methods result in more accurate multiscale approximations, however, they are based on pre-calculated global information which is computationally more expensive. Another remark regarding these methods, is that the pre-defined global information needs to be updated for each new well that is introduced in the simulation.

The construction of the independent local basis functions in [30, 32] follows the exact same steps as the MsFEM method, in which the reduced boundary condition is applied on the interfaces. This local boundary condition will still provide an interpolation error on the fine-scale. In [84] we show that it is in fact possible to incorporate the global fine-scale solution into the coarse operator, such that the fine-scale information is exactly represented, i.e. removing the interpolation error from the multiscale simulation.

5.1.3 The mixed multiscale finite element method

The MsFEM method was further extended to flow and transport by the mixed multiscale finite element method (MMsFEM) of [21]. In the mixed formulation, the solution for the pressure and the flux is computed simultaneously on each interface Γ_{ij}, between two coarse elements T_i and T_j. The two-domain region is illustrated in Figure 5.2. Thus, the MMsFEM method applies a pair of basis functions $(\phi_{ij}, \psi_{ij}) \in (U \times V)$, for the pressure and velocity, respectively. Here $(U, V) \subset L^2(\Omega) \times H(div; \Omega)$ are the finite dimen-

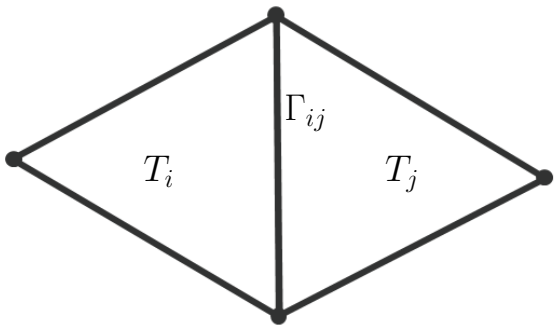


Figure 5.2: Upscaling region for the MMsFEM method

sional multiscale spaces of the discrete problem.

Assume a triangulation $\mathtt{T} = \{T\}$. The multiscale spaces are spanned by their local basis functions, where $\psi_{ij} = -K\lambda_t \nabla\phi_{ij}$. The local basis function ϕ_{ij} is calculated from the two-domain problem,

$$\nabla \cdot \psi_{ij} = \frac{1}{|T_i|} \quad \text{in} \quad T_i, \tag{5.6}$$

$$\nabla \cdot \psi_{ij} = -\frac{1}{|T_j|} \quad \text{in} \quad T_j, \tag{5.7}$$

subject to the boundary conditions

$$\psi_{ij} \cdot \mathbf{n}_{ij} = \frac{1}{|\Gamma_{ij}|} \quad \text{on} \quad \Gamma_{ij} \tag{5.8}$$

and

$$\psi_{ij} \cdot \mathbf{n} = 0 \quad \text{on} \quad \partial T_i \cup \partial T_j \setminus \Gamma_{ij}. \tag{5.9}$$

The Equations (5.6) and (5.7) form the homogeneous elliptic problem. This is the same equation which is also being used for the calculation of the local MsFEM basis functions.

In the MMsFEM framework, both the pressure and the velocity are computed on the coarse scale as well as interpolated onto the fine-scale. The approximate fine-scale solution is mass-conservative and may be applied for fine-scale simulations. However, as pointed out by Aarnes et. al. [1], the local conservation only exists on the coarse scale. Fine-scale source terms (non-zero right hand side term f) are not taken into account when calculating the coarse-scale basis functions. Thus, injection and production wells are only modelled on the coarse scale, and their contribution on the fine-scale will be averaged out across the entire coarse element.

Aarnes et. al. [1] modified the MMsFEM method, and included local conservation on the coarse elements which contained source terms. Thus, the MMsFEM method can be applied to simulate on the fine-scale. In the case of $\int_{T_i} f \ \mathrm{dx} \geq 0$, the modified basis function ψ_{ij} is defined on T_i such that

$$\nabla \cdot \psi_{ij} = \frac{f}{\int_{T_i} f \mathrm{dx}}, \tag{5.10}$$

subject to the boundary conditions (5.8) and (5.9). In the case of $\int_{T_i} f \ \mathrm{dx} = 0$, the Equations (5.6) and (5.7) are considered. The framework for calculating the local basis-functions for dynamical variables on the interface

is much more flexible than the nodal based basis functions with respect to the gridding. The MMsFEM method can be used to simulate on general grids.

Further extensions of this framework have focused on improved subscale capturing of large-scale structures [4], better modelling of the well-bore and near-well flow [59]. For efficient multiscale simulations, the MMsFEM method has also been coupled with streamline methods for transport [3, 90].

5.1.4 The variational multiscale method

In the variational multiscale (VMS) method we follow an alternative approach, different from traditional upscaling. Considering the variational formulation

$$a(u, v) = (f, v) \quad \forall \quad v \in V, \tag{5.11}$$

where $a(u, v)$ is a bilinear form, VMS seek a two-scale solution $u = u_c + u_f \in \mathbb{U} = U_f \oplus U_c$. Here, $u_c \in U_c$ represents the coarse-scale solution, and $u_f \in U_f$ corresponds to the remaining part of the solution on the fine-scale. Thus, from the spitting we can write the fine-scale and coarse-scale equations as follows:

$$\begin{aligned} a(u_f, v_f) + a(a_c, v_f) &= (f, v_f) \quad v_f \in V_f, & (5.12)\\ a(u_f, v_c) + a(a_c, v_c) &= (f, v_c) \quad v_c \in V_c, & (5.13) \end{aligned}$$

Note, that the multiscale framework is formulated in the continuous space. Most of the VMS methods originates from the work of Hughes [51], where the fine-scale solution u_f is written in terms of a Green's function. Different approximations of the Green's functions, will then in general result in different VMS methods. The spaces U_f and U_c are normally orthogonal, but this is not a requirement. In [60] they decompose the fine-scale solution, $u_f = \sum_i u_f^i$, where $u_f^i \in U_f^i(\omega_i) \subset U_f(\Omega)$ is a localised solution with support on $\omega_i \subset \Omega$. The ω_i is here refered to as a patch, and the size of these patches may be optimised based on a posteriori error estimates. Adaptive implementations based on the residual error have also been proposed [73].

5.2 Multiscale control-volume methods

In this section we will consider the multiscale methods designed for the control-volume discretisation, also known as the finite-volume discretisation.

Most of conventional reservoir simulators are based on the control-volume discretisation, thus one might claim that these multiscale methods have a greater potential of being implemented into existing reservoir simulator frameworks. The multiscale finite volume (MSFV) method is developed by [53], and is the most well-known of the multiscale methods for control volumes. This method has been of special interest for the study in this thesis.

The MSFV method can be shown to be similar to upscaling of transmissibillities, in which the coarse-scale system also represents a discretisation based on the control volume formulation. The method can also be seen as a special case of a mass-conservative domain decomposition preconditioner, which is termed MCDD [73]. Both of these numerical strategies will be discussed more in the following. In our research we have considered the MCDD preconditioner, and we have developed several new multiscale control-volume methods with different numerical properties.

5.2.1 The multiscale finite volume method

The MSFV method was developed by Jenny et. al. [53] as the finite volume equivalent of the MsFEM. The multiscale assumption of the MSFV method is the same as that of the MsFEM method and the construction of the local basis functions are the same. These are defined in Equations (5.1) and (5.5).

The reconstruction of the fine-scale fluxes is different, and a bit more involved for the MSFV method. Since finite volumes are cell centred, whereas the finite elements are mesh centred, the local basis functions of the MSFV method are constructed on a mesh-centred dual coarse-grid. The dual-grid is illustrated in Figure 5.3. While the coarse scale pressure and fluxes are represented on the cell-centred primal coarse-grid, a post-processing step is required to recover the fine-scale mass-conservative fluxes. The post-processing step is described in [53].

Much of the work on extending/improving the MSFV method has focused on application towards more realistic problems in the petroleum industry; mainly towards including more physics into the framework. One of the main challenges has been to incorporate physical features that are not linearly scalable with the pressure. Such properties can not be captured by the local basis-functions, as they are defined in Equation (5.1). Recall that the multiscale basis function is designed to capture local fine-scale information of the elliptic operator with zero right-hand side, i.e. the solution of the homogeneous elliptic problem (5.2). In [66] they propose a second set of functions, denoted the correction functions $\{\psi^i\}$. These additional degrees of freedom, one per coarse node, need to be included in order to capture gravity effects and capillary effects. They are calculated by solving the local

elliptic problem with the right-hand side term and zero boundary conditions. The fine-scale solution is then defined as

$$u_{fine} = \sum_i \left(\phi^i u^i_{coarse} + \psi^i \right). \tag{5.14}$$

Similar to the MMsFEM method, wells are not easily included into the MSFV framework. The pressure regime in the vicinity of a production well is non-linear, and is not captured by either the MSFV basis function or the correction function. One additional basis function per well needs to be included in order to recover the locally conservative flux field in the vicinity of each well [54, 94]. Much work has also been devoted to improve the MSFV basis function towards particularly difficult porous structures, e.g. anisotropic porous media [65], shale layers [64] and fractured reservoirs [45].

It can however be criticised that too little research has been devoted to improving the MSFV method for conducting simulations on irregular grids and more complex geometries. All applications of the MSFV method seem to be considering regular Cartesian grids. The MSFV method has further been extended to 3D multiscale simulations [61], however, restricted to regular Cartesian grids.

The reduced boundary condition which is used to construct the MSFV basis function has clear weaknesses towards simulations on anisotropic porous media and non-**K**-orthogonal grids [58, 84]. In addition, the boundary condition is a geometric approximation, which is less robust w.r.t. irregular grid structures and is not in general extendible to multilevel methods [83].

5.2.2 The MSFV as an upscaling technique

The MSFV method was first implemented as an upscaling method, similar to the MsFEM method. MSFV can be seen as an upscaling of fluxes or transmissibillities.

Consider a regular Cartesian grid in two dimensions. For each dual coarse-grid cell (see Figure 5.3) we compute one basis function related to each coarse node. The coarse nodes are located at the corners of the dual coarse cell. The MSFV method solve 4 local basis function problems on each dual-coarse cell Ω_i, subject to the multiscale conditions (5.4) and (5.5). Recall from Section 3.2.3, that this is equivalent to the upscaling of transmissibillities in the MPFA method. Thus, on each dual coarse cell (interaction region) we have the following relationship between the basis-functions and the fluxes across the primal coarse interfaces:

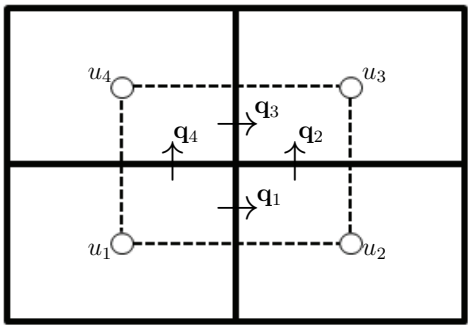


Figure 5.3: Illustration of a dual coarse grid cell for the MSFV method

$$\begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \\ \mathbf{q}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{t}_{11} & \mathbf{t}_{12} & \mathbf{t}_{13} & \mathbf{t}_{14} \\ \mathbf{t}_{21} & \mathbf{t}_{22} & \mathbf{t}_{23} & \mathbf{t}_{24} \\ \mathbf{t}_{31} & \mathbf{t}_{32} & \mathbf{t}_{33} & \mathbf{t}_{34} \\ \mathbf{t}_{41} & \mathbf{t}_{42} & \mathbf{t}_{43} & \mathbf{t}_{44} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix}. \tag{5.15}$$

In pure upscaling, the fluxes and transmissibillity coefficients across the coarse interfaces are summed together as coarse scale parameters. In the MSFV method, the fine-scale transmissibillities are stored in memory, for the recalculation of a mass-conservative fine-scale flux field. This post processing step is the only difference between an upscaling method as described in Chapter 3 and the MSFV method. If we sum the transmissibillities on the fine-scale together, we obtain the coarse operator of the MSFV method. This operator turns out to be similar to the MPFA-O(η) method, where $\eta = 1$ [46]. Recall from Chapter 2 that the numerical schemes resulting from the MPFA methods may have monotonicity issues for anisotropic flow, in cases where the principle directions of the permeability tensor is not aligned with the grid. Indeed this is also the case for the 9-point scheme of the MSFV coarse operator [84]. In [46] they propose a new compact coarse operator, which combined with the MSFV basis functions gives a more robust multiscale finite volume method for problems involving anisotropic porous media.

Similar to other upscaling methods, the accuracy of the MSFV method can be improved by means of oversampling [40], or inclusion of global information [30, 32]. Another approach, to be described next, is to formulate the MSFV method in an iterative framework.

5.2.3 The multiscale methods in an iterative framework

While the coarse scale operator of the MSFV method suffers from monotonicity issues, there might be large local errors in the approximate multiscale solution, but in many cases the multiscale solution can be substantially improved by applying a few local iterations. The first iterative formulation of the MSFV method was described in [44]. Here, they proposed to iterate on the boundary conditions for the elliptic problem of solving the local basis-function, and for this purpose they applied a line relaxation method. The MSFV method is easily extended to an iterative multiscale method. In fact, the original MSFV method itself, can be seen as a special case of a mass-conservative domain-decomposition (MCDD) preconditioner, which applies one numerical iteration before recovering the mass-conservative fine-scale solution [75].

The link between multiscale methods and domain decomposition has also been studied for the MsFEM method [2]. Similar to the MSFV method, the MsFEM method can be viewed as a non-overlapping domain decomposition preconditioner. Their study indicates that the reduced boundary condition gives a faster convergence than the use of linear boundary conditions for most problems in 2D and 3D. Note also that in 1D, the MsFEM method is exact, i.e. it fully resolves the solution after one iteration.

For 2D elliptic problems with highly oscillating coefficients, the condition number is proportional to the largest jump in material constants. To some extent, the harmonic basis functions of the multiscale methods (MsFEM and MSFV) can capture the fine-scale oscillations on the sub-domain and thus result in more robust preconditioners for flow in heterogeneous porous media [42].

In the framework of domain decomposition, the multiscale control volume methods are considered to be special types of Schur complement preconditioners, or MCDD-preconditioners. By a pre-processing of the fine-scale linear system, mass-conservation can be passed onto any hierarchical coarse grid [83]. If no approximations are considered for the coarse-scale nodes, a mass-conservative solution may be calculated on the coarse-scale, and recovered on the fine-scale. The recomputation of the fine-scale mass-conservative flux-field follows the post-processing step of the MSFV method [53]. Different approximations to the Schur complement is equivalent to choosing different boundary conditions, and will lead to different multiscale methods or different MCDD-preconditioners. The reduced boundary conditions of the MSFV method for problems involving for regular Cartesian grids, is equivalent with a special Schur complement approximation called the Tangential component (TC) approximation [84]. The TC approximation is most accurate when the

principle directions of the flow is aligned in the same direction as the local boundaries. In case of strong anisotropic flow and flow on irregular grids, other approximations are more accurate.

In our study we have studied the multiscale methods in the framework of MCDD. By developing algebraic preconditioners, we seek multiscale control volume methods that are more robust with respect to challenging grids and porous structures. Furthermore, we aim at constructing a flexible multiscale framework. Based on the problem, we should be able to optimise the size of each sub-domain, the number of coarse levels, the number of coarse-scale nodes on each coarse level, and the numerical interface approximation on each sub-domain boundary. These aspects have been studied in the scientific papers.

Chapter 6

Summary of papers and conclusion

The thesis includes three scientific papers. In the preceding chapters we have presented the theory and foundation for the study of this thesis. In this chapter we will give a short summary to each of the papers; present the problems that we have investigated and summarize the main results that we have obtained. In the end of this chapter we draw some general conclusions upon the results and insight that we have achieved.

In addition to the scientific papers, a book chapter is included as a supporting material to this dissertation. Here we give a detailed description of the mass-conservative domain-decomposition preconditioners, which has been our framework when studying the multiscale control-volume methods. The chapter should be regarded as a complement to the thesis and should be read after the three scientific papers, in the light of the results and the findings that we have obtained.

6.1 Summary of papers

Paper A: Multiscale Mass-Conservative Domain-Decomposition Preconditioners for Elliptic Problems on Irregular Grids. Published in *Computational Geosciences*, Vol. 15(3), Pages 587-602, June 2011.

Andreas Sandvin, Jan Martin Nordbotten and Ivar Aavatsmark

In Section 5.2 we showed how the multiscale control volume method can be seen as a Schur complement preconditioner for a DD-method. This was first demonstrated in [75], where they formulated the MSFV as a

special type of mass-conservative DD preconditioners, termed MCDD. In this paper we examine several multiscale control-volume methods within the framework of MCDD, we study the mathematical and numerical properties of these methods and develop new MCDD-preconditioners and multiscale control-volume methods. We demonstrate how the original MSFV method fails to capture the linear pressure field on irregular Cartesian grids, and that it is not in general robust with respect to long-range heterogeneities. We developed a multiscale framework based on probing, which is pure algebraic and a more robust with respect to simulations on irregular grid structures.

Recall from Section 4.3.2 that the probing technique applies certain linearly independent probing vectors on each sub-domain, and that these vectors will be exactly captured by the numerical method. We show that, based on the choice of these vectors, we can build either robust preconditioners, or accurate upscaling methods. In particular, we propose a new upscaling method based on pre-defined local flow fields. The resulting multiscale framework based on probing is convergent, more flexible and it is a natural way of incorporating global or non-local information into the multiscale simulation.

Paper B: A Unified Multilevel Framework of Upscaling and Domain-Decomposition. Proceeding paper of the XVIII International Conference on Water Resources, Barcelona 2010.

Andreas Sandvin, Jan Martin Nordbotten and Ivar Aavatsmark

The multiscale control-volume methods can be seen as either an upscaling method or a DD-preconditioner. This was discussed in Section 5.2 for the MSFV method. The same applies for the MCDD preconditioners, in which case we can stop the iteration process at any iteration step and recapture a fine-scale mass-conservative flux field. In this paper we show that the hierarchical multi-grid structure of the multiscale control-volume methods can easily be extended to build true multiscale methods in the framework of MCDD. In order to reconstruct a mass-conservative flux-field on the finer level, we have to require mass-conservative systems on the coarser grids. We only have to exactly compute the solution on the coarsest mass-conservative level. Furthermore, the mass-conservative solutions can be recovered stepwise onto the finest level. We demonstrated the multilevel framework of MCDD, together with the flexibility of the MCDD-preconditioner, by combining upscaling and preconditioning. On a three level grid, we upscale the finest level and apply a preconditioner on the intermediate level. We

observe that the multilevel framework is robust and that various upscaling techniques can be applied with different preconditioners.

Paper C: Auxiliary Variables for 3D Multiscale Simulations in Heterogeneous Porous Media. Submitted to Journal of Computational Physics, November 2011

Andreas Sandvin, Eirik Keilegavlen and Jan Martin Nordbotten

In Section 4.4 we discussed the importance of having a rich coarse space for the performance of the DD-preconditioners. In this paper we develop an new strategy for improving the robustness of the multiscale simulations in 3D, by means of including more information to the coarse space. In addition to the common mass-conservative variables on the coarse-scale, auxiliary coarse variables can be included, to capture non-linear effects which are not easily captured by the standard multiscale basis functions.

The construction of scalable domain decomposition preconditioners in 3D is non-trivial. In particular the vertex based coarse space, commonly applied with the multiscale methods, do not bound the condition number for increasing number of sub-domains. The multiscale framework using auxiliary variables is a generalisation of the vertex based method, which allows for additional degrees of freedom on the coarse scale. A special choice of the auxiliary variables results in the wire-basket method. This method is more robust in 3D and has been shown to yield scalable preconditioners for certain problems [89].

Our results clearly show that the wire-basket method results in a much better preconditioner than the usual vertex based method. The study also shows that we do not need to use all coarse-scale degrees of freedom related to the wire-basket. For problems involving flow in porous media, the method of auxiliary variables requires substantially less numbers of iterations when including a few additional coarse-scale degrees of freedom on the interfaces.

6.2 Conclusion

In this thesis we have analysed the multiscale control volume methods for flow in porous media. We have developed a flexible iterative multiscale framework, based on domain decomposition and we have implemented several multiscale preconditioners with increased robustness with respect to heterogeneous flow in porous media. The objective of this thesis has been to mature the multiscale control-volume methods towards applications on realistic porous media.

In particular, we have extended the multiscale framework toward simulations on general grids (paper A), general scales (paper B) and general dimensions (paper C and supporting material).

During this work we have identified several weaknesses of the multiscale framework of e.g. the MsFEM method and the MSFV method. In our opinion, the MCDD framework is more flexible and has several advantages over these methods. The multiscale method can be applied as an upscaling method, a domain decomposition preconditioner, or as a combination of upscaling and domain decomposition. Secondly, the framework is algebraic and the methods can be implemented as adaptive multilevel preconditioners. Finally, auxiliary variables may be included on the coarse scale, which enable the user to model the coarse scale more accurately.

For certain challenging problems of porous media flow, where the flow on the coarse scale is not aligned with the principle directions of the coarse-scale grid, the multiscale coarse-scale operator do not nescessarily satisfy the monotonicity criteria. This can result in large numerical errors in the coarse scale solution, and correspondingly pour approximations to the fine-scale flux field. These fluxes may not be applicable for conducting simulations. For such problems, it is absolutely crucial to be able to iterate on the coarse-scale solution. Within the framework of MCDD, we can iteratively improve the coarse-scale solution to whatever accuracy that we desire. At any iteration step, we can recover a mass-conservative fine-scale flux field and run fine-scale simulations accordingly.

The increased robustness and flexibility of the multiscale methods brings us closer to the ultimate goal of combining simulations on the geological fine-scale model and the coarse-scale simulation model. Moreover, efficient implementations of this framework can make it possible to include the influence of detailed fine-scale information, such as results from fractured network models and fault descriptions, into the reservoir simulator. This may substantially improve the accuracy of future reservoir simulations.

Bibliography

[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 (electronic), 2004.

[2] J.E. Aarnes and T.Y. Hou. Multiscale domain decomposition methods for elliptic problems with high aspect ratios. *Acta Math. Appl. Sin.*, 18(1):63–76, 2002.

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

[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 (electronic), 2006.

[5] I. Aavatsmark. An introduction to multipoint flux approximations for quadrilateral grids. *Comput. Geosci.*, 6(3-4):405–432, 2002.

[6] I. Aavatsmark, T. Barkve, Ø. Bøe, and T. Mannseth. Discretization on non-orthogonal, curvilinear grids for multi-phase flow. *Proc. of the 4th European Conf. on the Mathematics of Oil Recovery*, 1994.

[7] I. Aavatsmark, T. Barkve, and T. Mannseth. Control-volume discretization methods for 3D quadrilateral grids in inhomogeneous, anisotropic reservoirs. *SPE J*, 3(2), 1998.

[8] I. Aavatsmark, G.T. Eigestad, B.T. Mallison, and J.M. Nordbotten. A compact multipoint flux approximation method with improved robustness. *Numer. Methods Partial Differential Equations*, 24(5):1329–1360, 2008.

[9] V.I. Agoshkov and V.I. Lebedev. Poincaré-Steklov operators and methods of partition of the domain in variational problems. In *Computational processes and systems, No. 2*, pages 173–227. "Nauka", Moscow, 1985.

[10] J.R. Appleyard and Cheshire I.M. Nested factorization. *Proc. of the 7th SPE Symposium on Reservoir Simulation*, 1983. SPE 12264.

[11] O. Axelsson. *Iterative solution methods.* Cambridge University Press, New York, NY, USA, 1994.

[12] K. Aziz and A. Settari. *Petroleum reservoir simulation.* Applied Science Publisher, 1979.

[13] J. Bear. *Dynamics of Fluids in Porous Media.* Elsevier, 1972.

[14] S. Bhowmick, A.D. Kaushik, B.L. Mcinnes, B.B. Norris, and B.P. Raghavan. Parallel adaptive solvers in compressible petsc-fun3d simulations. *Proceeding of the 17th International Conference on Parallel CFD*, 2005.

[15] P.E. Bjørstad and M.D. Skogen. Domain decomposition algorithms of schwarz type, designed for massively parallel computers. In *in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, 1992.

[16] Ø Bøe. Analysis of an upscaling method based on conservation of dissipation. *Transport in porous media*, 17(1):77–86, 1994.

[17] J.R. Bunch. Stability of methods for solving Toeplitz systems of equations. *SIAM J. Sci. Statist. Comput.*, 6(2):349–364, 1985.

[18] S.A. Castro, J. Caers, C. Otterlei, H. Meisingset, T. Hoye, P. Gomel, and E. Zachariassen. Incorporating 4D seismic data into reservoir models while honoring production and geologic data. *The Leading Edge*, 28(12):1498–1505, December 2009.

[19] Y. Chen and L.J. Durlofsky. Adaptive local-global upscaling for general flow scenarios in heterogeneous formations. *Transp. Porous Media*, 62(2):157–185, 2006.

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

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

[22] 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.

[23] M.A. Christie. Upscaling for reservoir simulation. *Journal of petroleum technology*, 48(11), November 1996.

[24] H. Darcy. *Les Fontains Publiques de la Ville de Dijon*. Dalmont, Paris, 1856.

[25] D.B. Das and S.M. Hassanizadeh, editors. *Upscaling multiphase flow in porous media: from pore to core and beyond*. Springer, 2005.

[26] C. Deutsch. Calculating effective absolute permeability in sandstone/shale sequences. *SPE Formation Evaluation*, 1989.

[27] M. Dryja. A capacitance matrix method for Dirichlet problem on polygon region. *Numer. Math.*, 39(1):51–64, 1982.

[28] L.J. Durlofsky. Upscaling of geocellular models for reservoir flow simulation: A review of recent progress. In *7th International Forum on Reservoir Simulation*, pages 23–27, 2003.

[29] L.J. Durlofsky and E.Y. Chung. Effective permeability of heterogeneous reservoir regions. *Proc. of the 2th European Conf. on the Mathematics of Oil Recovery*, 1990.

[30] L.J. Durlofsky, Y. Efendiev, and V. Ginting. An adaptive local-global multiscale finite volume element method for two-phase flow simulations. *Advances in Water Resources*, 30(3):576 – 588, 2007.

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

[32] 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.

[33] O. Eiken, T. Stenvold, M. Zumberge, H. Alnes, and G. Sasagawa. Gravimetric monitoring of gas production from the troll field. *Society of Exploration Geophysicists*, 73(6), November 2008.

[34] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Linear Algebra Appl.*, 7(7-8):687–714, 2000.

[35] C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Internat. J. Numer. Methods Engrg.*, 32:1205–1227, 1991.

[36] C.L. Farmer. Upscaling: a review. *Internat. J. Numer. Methods Fluids*, 40(1-2):63–78, 2002. ICFD Conference on Numerical Methods for Fluid Dynamics (Oxford, 2001).

[37] C.L. Farmer. Geological modelling and reservoir simulation. In *Mathematical methods and modelling in hydrocarbon exploration and production*, volume 7 of *Math. Ind.*, pages 119–212. Springer, Berlin, 2005.

[38] Y. Fragakis and M. Papadrakakis. The mosaic of high performance domain decomposition methods for structural mechanics: Formulation, interrelation and numerical efficiency of primal and dual methods. *Computer Methods in Applied Mechanics and Engineering*, 192(35-36):3799 – 3830, 2003.

[39] M. Gerritsen and L.J. Durlofsky. Modelling fluid flow in oil reservoirs. *Fluid Mechanics*, 37:211–238, January 2005.

[40] V. Ginting. Analysis of two-scale finite volume element method for elliptic problem. *J. Numer. Math.*, 12(2):119–141, 2004.

[41] G.H. Golub and D. Mayers. The use of preconditioning over irregular regions. In *Computing methods in applied sciences and engineering, VI (Versailles, 1983)*, pages 3–14. North-Holland, Amsterdam, 1984.

[42] I.G. Graham, P.O. Lechner, and R. Scheichl. Domain decomposition for multiscale PDEs. *Numer. Math.*, 106(4):589–626, 2007.

[43] D.R. Guérillot and S. Verdière. Different pressure grids for reservoir simulation in heterogeneous reservoirs. *SPE Reservoir Simulation Symposium*, 1995.

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

[45] H. Hajibeygi and P. Jenny. A hierarchical fracture model for the iterative multiscale finite volume method. *J. Comput. Phys.*, 230(24):8729–8743, 2011.

[46] M.A. Hesse, B.T. Mallison, and H.A. Tchelepi. Compact multiscale finite volume method for heterogeneous anisotropic elliptic equations. *Multiscale Model. Simul.*, 7(2):934–962, 2008.

[47] R. Holm. *Modelling of three-phase flow functions for applications in enhanced oil recovery*. PhD thesis, University of Bergen, 2009.

[48] U. Hornung, editor. *Homogenization and porous media*. Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[49] 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(1):169–189, 1997.

[50] T.Y. Hou, X.-H. Wu, and Z. Cai. Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. *Math. Comp.*, 68(227):913–943, 1999.

[51] T.J.R. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Comput. Methods Appl. Mech. Engrg.*, 127(1-4):387–401, 1995.

[52] P. Indelman and G Dagan. Upscaling of permeability of anisotropic heterogeneous formations; 1. the general framework. *Water resources research*, 29(4):917–923, April 1993.

[53] 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 (electronic), 2004/05.

[54] P. Jenny and I. Lunati. Modeling complex wells with the multi-scale finite-volume method. *J. Comput. Phys.*, 228(3):687–702, 2009.

[55] E. Keilegavlen. *Robust control volume methods for reservoir simulation on challenging grids*. PhD thesis, University of Bergen, 2010.

[56] E. Keilegavlen, J.M. Nordbotten, and A. Stephansen. Tensor relative permeabilities: origins, modeling and numerical discretization. To be published in International Journal of Numerical Analysis & Modeling.

[57] D.E. Keyes and W.D. Gropp. A comparison of domain decomposition techniques for elliptic partial differential equations and their parallel implementation. *SIAM J. Sci. Stat. Comput.*, 8:166–202, March 1987.

[58] 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.

[59] S. Krogstad and L.J. Durlofsky. Multiscale mixed-finite-element modeling of coupled wellbore/near-well flow. *SPE J*, 14(1):78–87, March 2009.

[60] M.G. Larson and A. Målqvist. A mixed adaptive variational multiscale method with applications in oil reservoir simulation. *Math. Models Methods Appl. Sci.*, 19(7):1017–1042, 2009.

[61] S.H. Lee, C. Wolfsteiner, and H.A. Tchelepi. Multiscale finite-volume formulation of multiphase flow in porous media: black oil formulation of compressible, three-phase flow with gravity. *Comput. Geosci.*, 12(3):351–366, 2008.

[62] J. Li and O.B. Widlund. FETI-DP, BDDC, and block Cholesky methods. *Internat. J. Numer. Methods Engrg.*, 66(2):250–271, 2006.

[63] P.-L. Lions. On the Schwarz alternating method. II. Stochastic interpretation and order properties. In *Domain decomposition methods (Los Angeles, CA, 1988)*, pages 47–70. SIAM, Philadelphia, PA, 1989.

[64] I. Lunati and P. Jenny. Multi-scale finite-volume method for highly heterogeneous porous media with shale layers. *Proc. of the 9th European Conf. on the Mathematics of Oil Recovery*, 2004.

[65] I. Lunati and P. Jenny. Treating highly anisotropic subsurface flow with the multiscale finite-volume method. *Multiscale Model. Simul.*, 6(1):308–318 (electronic), 2007.

[66] I. Lunati and P. Jenny. Multiscale finite-volume method for density-driven flow in porous media. *Comput. Geosci.*, 12(3):337–350, 2008.

[67] J.L. Mallet. *Numerical Earth Models*. EAGE Publications, 2008.

[68] J. Mandel. Iterative solvers by substructuring for the p-version finite element method. *Comput. Methods Appl. Mech. Engrg.*, 80(1-3):117–128, 1990.

[69] J. Mandel. Balancing domain decomposition. *Comm. Numer. Methods Engrg.*, 9(3):233–241, 1993.

[70] J. Mandel, C.R. Dohrmann, and R. Tezaur. An algebraic theory for primal and dual substructuring methods by constraints. *Appl. Numer. Math.*, 54(2):167–193, 2005.

[71] J. Mandel and B. Sousedík. Coarse spaces over the ages. *ArXiv e-prints*, nov 2009.

[72] K. Nordahl and P.S. Ringrose. Identifying the representative elementary volume for permeability in heterolitic deposits using numerical rock models. *Math. Geosci.*, 40(7):753–771, 2008.

[73] J.M. Nordbotten. Adaptive variational multiscale methods for multiphase flow in porous media. *Multiscale Model. Simul.*, 7(3):1455–1473, 2008.

[74] J.M. Nordbotten, I. Aavatsmark, and G.T. Eigestad. Monotonicity of control volume methods. *Numer. Math.*, 106(2):255–288, 2007.

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

[76] D.W. Peaceman. *Fundamentals of numerical reservoir simulation*. Elsevier Scientific Publishing Company, 1977.

[77] M. Presho, S. Wo, and V. Ginting. Calibrated dual porosity, dual permeability modeling of fractured reservoirs. *Journal of Petroleum Science and Engineering*, 77(3-4):326–337, June 2011.

[78] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*. Numerical Mathematics and Scientific Computation. The Clarendon Press Oxford University Press, New York, 1999. Oxford Science Publications.

[79] Ph. Renard and G. de Marsily. Calculating equivalent permeability: a review. *Advances in Water Resources*, 20(5-6):253–278, 1997.

[80] Y. Rubin and J.J. Gómez-Hernández. A stochastic approach to the problem of upscaling of conductivity in disordered media: Theory and unconditional numerical simulations. *Water resources research*, 26(4):691–701, APRIL 1990.

[81] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

[82] A. Sandvin, E. Keilegavlen, and J.M. Nordbotten. Auxiliary variables for 3D multiscale simulations in heterogeneous porous media. To be published in J. Comput. Phys.

[83] A. Sandvin, J.M. Nordbotten, and I. Aavatsmark. A unified multilevel framework of upscaling and domain decomposition. *Proc. of the XVIII International Conference on Water Resources*, 2010.

[84] A. Sandvin, J.M. Nordbotten, and I. Aavatsmark. Multiscale mass conservative domain decomposition preconditioners for elliptic problems on irregular grids. *Comput. Geosci.*, 15(3):587–602, June 2011.

[85] Schlumberger. *Eclipse: Technical description*, 2007.

[86] H.A. Schwarz. *Gesammelte mathematische Abhandlungen. Band I, II*. Chelsea Publishing Co., Bronx, N.Y., 1972. Nachdruck in einem Band der Auflage von 1890.

[87] J.R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry-Theory and Applications*, 22(1-3), May 2002.

[88] B.F. Smith, P.E. Bjørstad, and W.D. Gropp. *Domain decomposition*. Cambridge University Press, Cambridge, 1996. Parallel multilevel methods for elliptic partial differential equations.

[89] B.F. Smith and O.B. Widlund. A domain decomposition algorithm using a hierarchical basis. *SIAM J. Sci. Statist. Comput.*, 11(6):1212–1220, 1990.

[90] V.R. Stenerud, V. Kippe, A. Datta-Gupta, and K.-A. Lie. Adaptive multiscale streamline simulation and inversion for high-resolution geomodels. *SPE Journal*, 13(1):99–111, March 2008.

[91] C.H. Tong, T.F. Chan, and C.-C.J. Kuo. A domain decomposition preconditioner based on a change to a multilevel nodal basis. *SIAM J. Sci. Statist. Comput.*, 12(6):1486–1495, 1991.

[92] A. Toselli and O. Widlund. *Domain decomposition methods—algorithms and theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2005.

[93] O. Widlund. The development of coarse spaces for domain decomposition algorithms. In *Domain Decomposition Methods in Science and Engineering XVIII*, volume 70, pages 241–248. Springer-Verlag, 2009.

[94] C. Wolfsteiner, S.H. Lee, and H.A. Tchelepi. Well modeling in the multiscale finite volume method for subsurface flow simulation. *Multiscale Model. Simul.*, 5(3):900–917 (electronic), 2006.

[95] X.H. Wu, Y. Efendiev, and T.Y. Hou. Analysis of upscaling absolute permeability. *Discrete Contin. Dyn. Syst. Ser. B*, 2(2):185–204, 2002.

[96] R.A. Yorgova and I. Aavatsmark. Modification of a reservoir grid to achieve monotonicity of the control volume method. *Proc. of the 12th European Conf. on the Mathematics of Oil Recovery*, 2009.

Part II

Included Papers

Paper A

Multiscale Mass-Conservative Domain-Decomposition Preconditioners for Elliptic Problems on Irregular Grids *

* Published in Computational Geosciences, Volume 15, Number 3, Pages 587-602, 2011.
DOI 10.1007/s10596-011-9226-6

Comput Geosci (2011) 15:587–602
DOI 10.1007/s10596-011-9226-6

ORIGINAL PAPER

Multiscale mass conservative domain decomposition preconditioners for elliptic problems on irregular grids

Andreas Sandvin · Jan Martin Nordbotten · Ivar Aavatsmark

Received: 25 May 2010 / Accepted: 26 January 2011 / Published online: 24 February 2011

Abstract Multiscale methods can in many cases be viewed as special types of domain decomposition preconditioners. The localisation approximations introduced within the multiscale framework are dependent upon both the heterogeneity of the reservoir and the structure of the computational grid. While previous works on multiscale control volume methods have focused on heterogeneous elliptic problems on regular Cartesian grids, we have tested the multiscale control volume formulations on two-dimensional elliptic problems involving heterogeneous media and irregular grid structures. Our study shows that the tangential flow approximation commonly used within multiscale methods is not suited for problems involving rough grids. We present a more robust mass conservative domain decomposition preconditioner for simulating flow in heterogeneous porous media on general grids.

Keywords Porous media · Reservoir simulation · Multilevel

1 Introduction

The heterogeneities at different scales in porous rocks make reservoir simulations computationally challenging, both with respect to time consumption and accuracy. The rapid variations in fine-scale permeability have big influence on the flow and need to be accounted for in the numerical methods. Various upscaling procedures have been developed to increase the efficiency of the flow calculations (see [12]). These techniques serve to construct coarse-scale flow parameters for the global problem on a coarser scale. However, for flow in complex geological media, it is crucial to solve the transport of fluids on the fine scale (Darcy scale). The idea behind multiscale methods as it was presented in [18] is to capture the fine-scale flow properties within independent local basis functions. After solving for the pressure on the coarse scale, the local basis functions then serve as accurate interpolation functions from the coarse-scale to the fine-scale pressure solution. Since the significant change in saturation often takes place in smaller parts of the global domain, only a few local basis functions need to be recalculated at each time step. Also, since the local basis functions are independent, the calculation of these may be carried out in parallel. Numerical experiments show that the multiscale technique can be efficient for solving multiphase flow problems in heterogeneous porous media [1, 3, 20].

The accuracy of the multiscale solution will, however, depend on the choice of localisation approximation, i.e. the choice of boundary conditions for the local basis function problems. Since the error in the local solutions is the largest close to the local boundaries, it has been shown in [19] that the error can be greatly reduced by calculating the local basis functions on larger overlapping domains. This corresponds to moving the local boundaries closer to the global boundary which determines the exact flow field. Another approach is to directly incorporate global information into the local

A. Sandvin (✉) · I. Aavatsmark
Centre for Integrated Petroleum Research,
University of Bergen, 5020, Bergen, Norway
e-mail: andreas.sandvin@cipr.uib.no

J. M. Nordbotten
Department of Mathematics, University of Bergen,
Johannes Brunsgate 12, 5008, Bergen, Norway

boundary value problems, by first solving an initial global fine-scale problem [11]. The use of local–global information has also been considered in [10]. While these methods may provide more accurate local solutions, they are in general more computationally expensive, specially in the case of repeated update of the global information, e.g. for changing global boundary conditions. A third approach is to improve the accuracy by means of local iterations on the domain interfaces [16, 28]. We will consider the latter approach.

It has been showed, in [28], that the multiscale finite-volume (MSFV) method of Jenny et al. [20] can be viewed as a special case of a mass conservative domain decomposition (MCDD) preconditioner, using a tangential flow approximation on the domain interfaces. The fine-scale solution for the MSFV method is expressed as a linear combination of local basis functions, which is equivalent with one iteration using the MCDD preconditioner. In the following, we will refer to the multiscale solution, as the approximation obtained after one iteration with the MCDD preconditioner. The class of MCDD preconditioners offers a general framework for approximating the flow on the interface, in which we can construct a wide range of different multiscale preconditioners with various properties. In this paper, we will focus on four principal properties for the multiscale preconditioner. The preconditioner should:

- *Be cheap to construct*. In this paper, we consider local sparse approximations to the flow on the boundary, i.e. sparse representations of the local Schur complement systems.
- *Be applicable as a multiscale method*. For many practical applications, it is computationally too expensive to iterate on the fine-scale solution, and the coarse-scale solution will be applied directly. The preconditioner should give a physically reliable approximation to the fine-scale flow field after only one iteration.
- *Possess good convergence properties*. Some of the local fine-scale features may be difficult to capture within a coarse-scale system. Thus, we are forced to iterate on the fine-scale residual.
- *Be applicable to realistic porous media*. Realistic flow problems for reservoir simulation involve irregular grid structures and heterogeneous anisotropic permeability fields. The multiscale approximations induced on the local domain interfaces have to be robust with respect to irregular geometries and fine-scale anisotropies not aligned with the grid.

We will consider the following elliptic problem for two-dimensional flow in porous media,

$$-\nabla\cdot(\mathbf{K}\nabla u) = q \qquad \text{in } \Omega \subset \mathbb{R}^2, \tag{1}$$

where $\mathbf{K}$ is a symmetric positive definite matrix representing the permeability of the media, u is the potential and q represents the source terms. The permeability is in general a full tensor describing the conductivity of an anisotropic porous medium, and it is the spatial variability of this parameter which represents the key challenge discussed in this paper. By integrating Eq. 1 over an arbitrary control volume $\omega \subset \Omega$ and applying Green's theorem, we obtain the integral equation for conservation of incompressible fluids,

$$\int_{\partial\omega} \mathbf{f}\cdot\nu \;\mathrm{d}\sigma = \int_{\omega} q \;\mathrm{d}\tau. \tag{2}$$

Here, $\mathbf{f} = -\mathbf{K}\nabla u$ represents the Darcy velocity and ν is the outward normal vector to $\partial\omega$. Methods based on the discretisation of Eq. 2 is referred to as control volume methods and yield local mass conservation within ω. The resulting discrete system of fine-scale equations, arising from a control volume discretisation, takes the following form:

$$Au = b. \tag{3}$$

We will assume that the solution $u \in V$, where V is the space of piecewise linear functions on $\Omega = \cup\omega_i$. The right-hand side term b represents the integrated sources over ω_i and belongs to the space of piecewise constant functions on Ω. Finally, the fine-scale operator $A \in V$ is a sparse and in general non-symmetric matrix, which contains information about the fine-scale geometry and variability of $\mathbf{K}$.

Previously, multiscale control volume methods have focused on the elliptic problem (see Eq. 1) on regular Cartesian grids. As far as we know, multiscale control volume methods have not been applied to problems involving irregular grid structures. In this paper, we investigate the accuracy, efficiency and robustness of different multiscale control volume approximations when applied to heterogeneous problems on irregular grids. Our study shows that the reduced boundary condition, commonly applied with the MSFV methods [10, 16, 20], is not robust with respect to perturbations on the fine-scale grid. Even for regular Cartesian fine grids and isotropic permeability tensor, the spatial variability in the fine-scale permeability may produce anisotropies on the coarse scale. The tangential flow on the local

boundaries is not sufficient to capture these coarse-scale anisotropies within the solution. We introduce a new multiscale framework, based on algebraic approximations to the Schur complement for constructing more accurate and robust interface approximations, for multiscale simulations on irregular grids. While this is a pure algebraic technique, it naturally extends to unstructured and multilevel grids.

The paper is organised as follows: In the next section, we give an introduction to the MCDD preconditioners. We show how the multiscale methods can be formulated as stand-alone upscaling methods, or as MCDD preconditioners for an iterative process. In Section 3, we compare the existing localisation approximation used for multiscale control volume methods with an algebraic interface approximation based on probing. A small comparison on the computational cost related to each of the preconditioners is given in Section 4, before testing the robustness and efficiency of the preconditioners for some numerical experiments in Section 5. Finally, we conclude the paper.

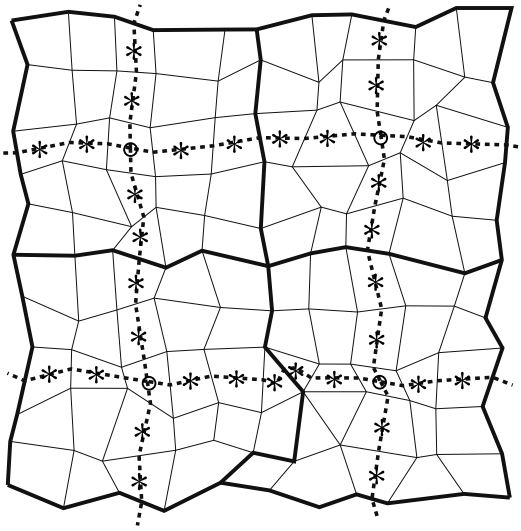


Fig. 1 The multiscale mesh. Here, the *bold faces* show the primal coarse grid, constructed on top of an underlying fine-scale grid. The *dashed lines* further indicate the dual coarse grid, on which the *circles* and *stars* refer to vertex and edge nodes, respectively

2 MCDD

In this section, we will consider the framework of non-overlapping domain decomposition methods and introduce the special class of MCDD methods introduced by [28]. Within this framework, we formulate the multiscale control volume methods as MCDD preconditioners.

2.1 Grids and scales

We will consider a cell-centred grid on the fine scale, consisting of control volumes $\{\omega_i, 1 \leq i \leq n\}$, such that they form a non-overlapping partition of Ω. Thus,

$$\Omega = \bigcup_i \omega_i; \quad \omega_i \cap \omega_j = \emptyset \quad i \neq j.$$

A primal coarse grid is then constructed on top of the fine grid, such that each primal coarse-grid cell $\{\Omega_i, 1 \leq i \leq N\}$ is a collection of fine-grid cells and the boundaries of Ω_i coincide with boundaries on the fine grid, as shown in Fig. 1. Moreover, we require that each fine-grid cell is represented in exactly one primal coarse-grid cells. Thus, the primal coarse grid also satisfies a non-overlapping partitioning of Ω. The centre-most fine-grid cell within each primal coarse-grid cell is further denoted as the coarse-grid node. Continuing in this manner, we note that we can construct a hierarchy of cell-centred coarse grids. However, for simplicity, we will here restrict our attention to two-scale methods. We consider true multiscale implementations of this framework in [30]. Note also that both the fine and the coarse grid may consist of arbitrarily shaped polygons.

We further introduce a dual coarse grid (indicated by dashed lines in Fig. 1) on which we will solve our local problems. We will refer to the dual coarse-grid cells as domains and denote them by Ω_i'. The primal coarse-grid nodes of Ω will then be located at the vertices of Ω_i'. On each domain Ω_i', the degrees of freedom corresponding to the boundary nodes will be denoted by subscript B and those corresponding to internal nodes by subscript I. The degrees of freedom related to the boundary unknowns will further be subdivided into those corresponding to vertex nodes and edge nodes, denoted by subscripts V and E, respectively.

Note that, while the primal coarse cells Ω_i are collections of cell-centred grid cells, the degrees of freedom on Ω_i are strictly separated from those on Ω_j, when $i \neq j$. For the dual coarse grid, the nodes are located on the vertices. Hence, the dual coarse cells Ω_i' form a non-overlapping partitioning of Ω in the classical sense, where the nodes on the dual-cell boundaries may be shared between neighbouring cells.

2.2 Mass conservative coarse-scale operator

For many applications, the multiscale method is applied as an upscaling procedure. At each time step of the simulation, the potential values are solved on the coarse scale, and the fine-scale solution is only reconstructed locally through a linear combination of coarse-scale basis functions. These basis functions are computed initially and seldom recomputed during the simulation. To reconstruct a mass conservative fine-scale flux field at a given time step, it is crucial that the coarse-scale operator also serves as a discretisation of the mass-conservation principle given in Eq. 2. Thus, as a preprocessing step, we will integrate the fine-scale equations (corresponding to fine-scale control volumes ω_i) associated with each primal coarse cell Ω_i into the row of the corresponding coarse node i. This will give us mass conservation on the primal coarse grid, represented by the coarse nodes.

We will consider a family of spaces $\{V_i,\ 1 \leq i \leq N\}$ corresponding to the primal coarse cells Ω_i and the extension operators $R_i^T : V_i \to V$, such that $V = \sum_{i=1}^{N} R_i^T V_i$. For each Ω_i, we define $M_i : V_i \to V_i$ as the integration operator adding all rows in $R_i A$, corresponding to fine-scale control volume equations on Ω_i, into the row of the coarse node i. More precisely, we can write the integration operator on matrix form

$$M_i = I + e_{iV}\left(\underline{1} - e_{iV}\right)^T, \tag{4}$$

where I is the identity matrix, e_{iV} is the unit vector identifying the row of the vertex (coarse) node and $\underline{1}$ is the vector entirely filled with ones. By applying the integration operator M_i on the fine-scale system (Eq. 3) restricted to each coarse cell Ω_i, we construct a system of equations, which is mass conservative on both scales. We write the MCDD system as

$$Cu = p, \tag{5}$$

where

$$C = \sum_{\Omega_i} (R_i)^T M_i R_i A \quad \text{and} \quad p = \sum_{\Omega_i} (R_i)^T M_i R_i b.$$

The fine-scale operator C belongs to the same space as A but has the additional property of preserving the mass balance on the coarse scale as well as the fine scale. It can be shown that this preprocessing step also acts as a good preconditioner for the fine-scale operator A, as it introduces a coarse space [30].

2.3 Schur complement system

The idea behind domain decomposition methods is to decouple the global fine-scale problem into independent local boundary value problems. The global fine-scale problem is then solved by iterating on the boundary unknowns of these local problems. To accelerate the iterative process, a global coarse-scale problem is constructed to capture the low-frequency error and to pass information between the local problems.

By grouping the unknowns corresponding to internal nodes in u_I and those corresponding to boundary nodes in u_B, we can reorder the unknowns $u = \begin{bmatrix} u_I & u_B \end{bmatrix}^T$, and write Eq. 5 as

$$\begin{bmatrix} C_{II} & C_{IB} \\ C_{BI} & C_{BB} \end{bmatrix} \begin{bmatrix} u_I \\ u_B \end{bmatrix} = \begin{bmatrix} p_I \\ p_B \end{bmatrix}. \tag{6}$$

All the internal unknowns u_I are now decoupled into local domains Ω_i', where the matrix C_{II} has a simple block diagonal structure. Thus, the internal degrees of freedom may be solved locally within each domain Ω_i' as

$$u_I = C_{II}^{-1}\left(p_I - C_{IB} u_B\right). \tag{7}$$

Hence, we can eliminate the internal degrees of freedom by substituting Eq. 7 into the second line of Eq. 6 and obtain the reduced Schur complement system

$$\left(C_{BB} - C_{BI} C_{II}^{-1} C_{IB}\right) u_B = p_B - C_{BI} C_{II}^{-1} p_I. \tag{8}$$

The matrix $S = C_{BB} - C_{BI} C_{II}^{-1} C_{IB}$ is referred to as the Schur complement of C, and for simplicity, we also denote $g = p_B - C_{BI} C_{II}^{-1} p_I$ as the modified right-hand side term. The Schur complement S is related to the space of discrete harmonic functions, where the multiplication of S to a vector $\mathbf{x}$ is equivalent to solve a local Dirichlet problem involving C_{II}^{-1} on each domain Ω_i'. The matrix S can be shown to yield better properties w.r.t. the condition number [4]; however, it is expensive to construct. In general, we never explicitly construct the Schur complement matrix S; we only do the necessary matrix–vector multiplications involving S.

By a similar reordering of the unknowns, $u_B = \begin{bmatrix} u_E & u_V \end{bmatrix}^T$, we can write the Schur complement system on matrix form

$$\begin{bmatrix} S_{EE} & S_{EV} \\ S_{VE} & S_{VV} \end{bmatrix} \begin{bmatrix} u_E \\ u_V \end{bmatrix} = \begin{bmatrix} g_E \\ g_V \end{bmatrix}. \tag{9}$$

Here, S_{EE} and S_{EV} have a block diagonal structure; however, each block is in general dense. In particular,

the Schur complement matrix S_{EE} can analytically be written as

$$S_{EE} = C_{EE} - C_{EI}C_{II}^{-1}C_{IE}. \tag{10}$$

This equation will be important later, when we discuss the properties of the different preconditioners. We observe that the first term in Eq. 10 contains the local couplings between the neighbouring edge elements, C_{EE}. This matrix will be sparse, with a predominantly tridiagonal structure. The second term in Eq. 10 is referred to as the global term, where internal information is interpolated onto the edge nodes. This term couples all the edge nodes together and forms a full matrix.

A direct solution of Eq. 9 is for most applications computationally too expensive, and we seek to construct a preconditioner for the Schur complement problem such that the number of local algebraic operations involving C_{II}^{-1} is as low as possible. We approximate Eq. 9 by

$$\begin{bmatrix} I & \hat{S}_{EE}^{-1}\hat{S}_{EV} \\ S_{VE} & S_{VV} \end{bmatrix} \begin{bmatrix} u_E \\ u_V \end{bmatrix} = \begin{bmatrix} \hat{S}_{EE}^{-1}g_E \\ g_V \end{bmatrix}, \tag{11}$$

where $\hat{S}_{EE}$ and $\hat{S}_{EV}$ now denote the approximations to S_{EE} and S_{EV}, respectively. Note that we have only modified the equations for the edge nodes, where

$$u_E = \hat{S}_{EE}^{-1}\left(g_E - \hat{S}_{EV}u_V\right). \tag{12}$$

Hence, the solution u_V of Eq. 11 still remains mass conservative on the coarse scale. In fact, any approximation of u_E will only affect the accuracy and not the property of mass conservation on the coarse scale. By substituting Eq. 12 into the second line of Eq. 11, we can write a mass conservative system of equations for the solution u_V on the coarse scale,

$$A_C u_V = g_V - S_{VE}\hat{S}_{EE}^{-1}g_E, \tag{13}$$

where

$$A_C = \begin{bmatrix} S_{VE} & S_{VV} \end{bmatrix} \begin{bmatrix} -\hat{S}_{EE}^{-1}\hat{S}_{EV} \\ I \end{bmatrix}. \tag{14}$$

The coarse-scale operator A_C is related to the space of piecewise discrete harmonic functions, where the approximated discrete harmonic extension on u_V is determined by the $\hat{S}_{EE}$ and $\hat{S}_{EV}$. For the multiscale methods, the coarse-scale equation (Eq. 13) will be solved directly. The columns of A_C contain the coarse-scale basis functions, which can be used to recover the fine-scale solution $u_f = \begin{bmatrix} u_I & u_E & u_V \end{bmatrix}^T$.

3 Interface approximations

The essential part in the construction of a good MCDD preconditioner, or efficient and accurate multiscale method, is the choice of interface approximation, which in our framework is $\hat{S}_{EB} = \begin{bmatrix} \hat{S}_{EE} & \hat{S}_{EV} \end{bmatrix}$. For most cases, the approximation error in $\hat{S}_{EE}$ will dominate. In this section, we will primarily focus on different approximations to S_{EE} and use $\hat{S}_{EV} = C_{EV}$. In Section 3.4, we will further discuss approximation techniques for the entire S_{EB} and how this can be related to flow-based upscaling techniques.

Recall from Eq. 10 that S_{EE} consists of two terms, a local term C_{EE} and a global term containing the couplings between edge and internal nodes. We will consider two types of interface approximations: a tangential flow approximation and an interface probing approximation. Both approximations result in low-band matrices, which are fast to invert. We have also applied other interface approximations, like the Toeplitz approximation [5] and the J-operator [6]. While they yield good results for the elliptic problem with constant coefficient on uniform Cartesian grids, they do not perform well for problems involving heterogeneous porous media and non-regular fine grids. Another approach, not considered here, is to directly approximate the second term of Eq. 10 by applying some local preconditioner on C_{II}. This would lead to a more expensive composite preconditioner, where the resulting approximation is not guaranteed to be sparse. For a broader discussion on different interface approximations, see [31].

3.1 Approximation properties

Our aim is to construct approximations $\hat{S}_{EE}$ with similar spectral properties as S_{EE} for which the system involving $\hat{S}_{EE}$ is fast to compute. In order to get a physically reliable solution after only one iteration (equivalent to solving the coarse-scale problem), we need to require some properties for the coarse-scale operator A_C. For single-scale methods, usual requirements for the system matrix A are that they are mass conservative and exactly reproduce constant and linear potential fields. As an example, most control volume methods are constructed precisely to satisfy these criteria.

As shown in Section 2, the MCDD preconditioners are constructed to be mass conservative on both the fine scale and the coarse scale. Furthermore, we will require that A_C is exact for constant solutions. For general heterogeneous elliptic problems, this is the only analytical solution that we can identify, which is

obtained by imposing zero boundary conditions and no internal source terms. For the multiscale methods, this implies that the local basis functions must form a partition of unity, i.e. the sum of the local basis functions is exactly equal to 1. For the MCDD preconditioners, we require that

$$\hat{S}_{EE}\mathbf{1} = S_{EE}\mathbf{1}, \tag{15}$$

which ensures that the approximation $\hat{S}_{EE}$ is exact for constant solutions. In the case of **K**-orthogonal fine grids, i.e. grids which are aligned with the principal directions of the permeability tensor, there are no couplings between vertex and internal nodes. Thus, $S_{EV} = C_{EV}$ and property 15 is a sufficient criterion for the resulting preconditioner to preserve constant solutions locally. However, for general grids, property 15 is not sufficient and we need to require that

$$\hat{S}_{EB}\mathbf{1} = S_{EB}\mathbf{1}. \tag{16}$$

The importance of capturing the constant solution has also been emphasized in domain decomposition, where it corresponds to capturing the null space of the local Schur complement matrices [24]. This is one of the important properties for the coarse space, which is needed to construct scalable two-level preconditioners with good convergence rates. Another important property for the coarse space, which is necessary to construct a robust preconditioner, is often referred to as the bounded energy condition [25] and is directly related to the capturing of sub-scale variations in the coefficients of **K** [15].

The motivation behind the multiscale methods is precisely to capture these local sub-scale variations within coarse-scale basis functions (i.e. within the coarse space of the corresponding operator A_c), by solving local PDEs on Ω_i' with pre-described boundary conditions. By solving extended local problems, e.g. by an oversampling procedure [9, 18], the local PDE-based fields on Ω_i' are made less sensitive to the boundary approximations, and the sub-scale information along $\partial\Omega_i'$ can also be well captured.

In Section 3.4, we will show how the interface probing technique can be applied to capture a few PDE-based fields governed by pre-defined boundary conditions on extended local domains. The difference from oversampling and global methods is that the local solutions are applied to approximate the Schur complement S_{EB}, rather than the basis functions.

3.2 Tangential component approximation

A frequently applied approximation for the multiscale methods, the reduced boundary condition (see [10, 11, 18, 20]), is to approximate the tangential flow along each local boundary. For the elliptic problem on regular Cartesian grids and with isotropic medium, the tangential flow along the boundary is found by discretising the elliptic equation (Eq. 1) directly along each local edge. As far as we know, the reduced boundary condition has only been applied to regular Cartesian grids, and it is not clear how to extend it to non-**K**-orthogonal grids. In order to test the reduced boundary condition within the MCDD framework, we discretise the elliptic equation (Eq. 1) along the local boundaries by a two-point flux approximation method. We denote the tridiagonal approximation resulting from the reduced boundary condition by S_{EE}^{RBC}. The preconditioner is denoted MCDD-RBC.

An equivalent approximation for regular Cartesian grids is the tangential component approximation, discussed in [28, 31]. This is an approximation to the first term of Eq. 10. Essentially, the tangential component (TC) approximation splits $C_{EE} = C_{EE}^T + C_{EE}^N$, where C_{EE}^N is a diagonal matrix containing the contribution to normal flow arising from the coupling between edge and internal nodes. By neglecting the flow normal to the local boundaries, the tangential component approximation is defined as

$$S_{EE}^{\mathrm{TC}} = C_{EE}^T. \tag{17}$$

The matrix S_{EE}^{TC} is tridiagonal when the edge nodes on the dual coarse grid have a natural numbering along each individual interface, and the expression is valid for general grids. We denote the resulting preconditioner, MCDD-TC. In the case of **K**-orthogonal fine grid, $S_{EE}^{\mathrm{TC}} = S_{EE}^{\mathrm{RBC}}$.

3.3 Probing technique

The interface probing technique (see [7] and references therein) represents a more general approach for approximating the flow on the boundary. The aim is to approximate the Schur complement matrix S_{EE} by a low-bandwidth matrix $\hat{S}_{EE}$, such that

$$\hat{S}_{EE}\mathbf{v}^i = S_{EE}\mathbf{v}^i = \mathbf{w}^i, \tag{18}$$

for some linearly independent probing vectors $\mathbf{v}^i$. The method was motivated by the observation that the Schur complement matrix often has a banded structure, where the Schur complement elements decay rapidly away from the diagonal. In fact, it has been shown by Golub [14] that $|S_{ij}| = O\left(|i-j|^{-2}\right)$, for $i \neq j$. The

probing vectors suggested in [7] for approximating the n-diagonal matrix S_{EE}^{nP} are

$$\mathbf{v}^i = \sum_{j=i \bmod(n)} \mathbf{e}_j. \tag{19}$$

In the case of tridiagonal probing, the probing vectors are $\mathbf{v}^1 = [1\,0\,0\,1\,0\,0\,\cdots]^T$, $\mathbf{v}^2 = [0\,1\,0\,0\,1\,0\,\cdots]^T$ and $\mathbf{v}^3 = [0\,0\,1\,0\,0\,1\,\cdots]^T$. While these probing vectors are linearly independent, Eq. 18 can be shown to yield a unique tridiagonal approximation $\hat{S}_{EE}$. Note also that Eq. 18 does not require the explicit formulation of the full matrix S_{EE}, which would be expensive; only those dimensions associated with the probing vectors $\mathbf{v}^i$ need to be calculated. In the case of three probing vectors, three local Dirichlet problems needs to be solved.

The interface probing technique can also be thought of as an efficient way of incorporating fine-scale properties into the coarse-scale operator. Indeed, since the sum of the probing vectors represented by Eq. 19 equals **1**, property 15 is naturally satisfied. We will denote the probing vectors of Eq. 19 as oscillating probing vectors because of their form and since they are designed to capture the fine-scale oscillations of the residual. We refer to MCDD-3P and MCDD-5P as the MCDD preconditioners, using S_{EE}^{nP} with $n = 3$ and 5, respectively.

3.4 Solution-based probing vectors

Originally, the interface probing technique was applied together with oscillating probing vectors, on the form given by Eq. 19, to approximate the diagonal structure of S_{EE}. For heterogeneous and anisotropic problems, there might be strong non-local couplings between boundary nodes, in which case the local Schur complement matrix does not have a diagonal structure. The oscillating probing vectors are designed to capture the fine-scale oscillations of the residual and therefore yield robust and good convergence properties for the iterative process. However, these probing vectors are not able to capture the correct features of the fine-scale solution after only one iteration. Motivated by standard flow-based upscaling strategies, see, e.g. [12], solution-based probing vectors are introduced to capture the underlying fine-scale variations within each local domain Ω_i'. Numerical experiments show that these probing vectors can be used to construct preconditioners with better approximation properties for the first iteration [30].

In order to guarantee that the multiscale preconditioners exactly reproduce constant solutions for general grids, we need to consider the probing of $S_{EB} = \begin{bmatrix} S_{EE} & S_{EV} \end{bmatrix}$ (see Eq. 16),

$$\hat{S}_{EB}\mathbf{v}^i = S_{EB}\mathbf{v}^i = \mathbf{w}^i. \tag{20}$$

Within an iterative solution process, we never construct the Schur complement matrix S_{EB} explicitly, nor do we construct the inverse of this matrix. We only apply the S_{EB} to some residual vectors $\mathbf{r}_B$ on the boundary. The multiplication of S_{EB} with a probing vector $\mathbf{v}$ can be regarded as solving a local Dirichlet problem with boundary values $\mathbf{v}$. Thus, the accuracy of the preconditioner will depend on how well the probing vectors $\mathbf{v}^i$ are able to capture the correct fine-scale variations on the boundary. For example, the choice $\mathbf{v} = \mathbf{1}$ will guarantee that the method preserves constant solutions locally, for general grids.

We compute the solution-based probing vectors $\mathbf{v}_{SB}^i$ on Ω_i' by solving local flow problems on $\Sigma_i \supset \Omega_i'$. The solution-based probing vectors $\mathbf{v}_{SB}^i$ are then defined as the restriction of the local solutions to the local boundary (see Fig. 2). It follows from Eq. 20 that the approximation $\hat{S}_{EB}$ will be exact for those particular boundary value problems defined by $\mathbf{v}_{SB}^i$. In this manner, the interface probing approximation can be constructed to be accurate for certain pre-defined boundary value problems.

The resulting multiscale preconditioner shares many similarities with the oversampling technique, applied in [9, 11, 13, 19]. In both strategies, a local elliptic problem is solved on an overlapping domain $\Sigma_i \supset \Omega_i'$, in order to reduce the approximation error introduced by the local boundary conditions. The difference of this strategy compared with other extended local or global multiscale methods is that we approximate the Schur complement, not the local basis functions. Thus, our multiscale method is formulated as a Schur complement preconditioner, which makes it a convergent method. Together with solution-based probing vectors, our preconditioner will be accurate for certain right-hand sides, in which case it has the additional property of an upscaling method. Results from Sandvin (to be submitted for publication) show that the proposed method is comparable in accuracy to multiscale control volume methods using oversampling.

In order to construct robust approximations to S_{EB}, we must require that the probing vectors are linearly independent and that the **1**-vector is represented through one or a sum of the probing vectors. A combination of solution-based and oscillating probing vectors is in our experience a good choice. In the original framework of the probing technique, a typical choice of $\hat{S}_{EE}$ would be a three-diagonal matrix, constructed from three

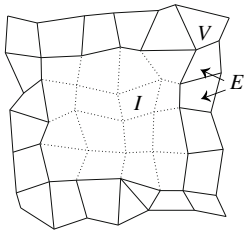


Fig. 2 Local domain Ω_i'

probing vectors, with the same sparsity pattern as the local discretisation C_{EE}. Thus, each edge segment connecting two vertex nodes (see Fig. 2) is decoupled, and the edge approximation $\hat{S}_{EE}$ can be inverted locally on the individual edge segments. The approximation $\hat{S}_{EB}$ is constructed in the same manner, from Eq. 20, filling only the diagonal elements and the element neighbours corresponding to the largest couplings in C_{EB}. While this interface approximation is only based on neighbour connections, it is a purely algebraic construction and independent on the underlying geometry. We denote the interface approximation S_{EB}^N and the corresponding MCDD preconditioner MCDD-N.

The construction of local approximations $\hat{S}_{EB}$, in the case of solution-based probing vectors, can be sensitive w.r.t. the requirement of linear independent probing vectors. To make the proposed MCDD-N preconditioner more robust, it is convenient to add a fourth probing vector and consequently a fourth non-local coupling. As a fourth coupling in S_{EB}, we choose an average value, representing the contribution from all local vertex nodes. Thus, the $\hat{S}_{EV}$ will be a full matrix, while the $\hat{S}_{EE}$ retain its local structure and can be inverted locally.

3.5 Inclusion of global information

For the application of multiphase flow, the elliptic problem needs to be solved repeatedly in time for slightly varying tensor coefficients K_{ij}. In this case, we may afford to spend some extra computational work initially, to construct an accurate multiscale method for the time-dependent problem. In [10, 11], they propose a more accurate multiscale method by incorporating information from a global fine-scale solution into the framework of the reduced boundary condition. However, the reduced boundary condition may not be able to capture the local variations of the global information correctly.

The interface probing technique represents a consistent framework for including fine-scale information into operators on coarser scales. In fact, if one of the probing vectors is chosen as the exact fine-scale solution restricted to the boundary, we have not done any approximations on the solution, and the fine-scale solution is solved exactly in one iteration.

3.6 Comparison of the two interface approximations

By applying the multiscale method as a preconditioner for a domain decomposition method, we have formulated each local problem as a Schur complement problem. In this framework, the reduced boundary condition, analogous to the tangential component approximation for **K**-orthogonal grids, is a purely local approximation, i.e. an approximation to the first term in Eq. 10. The algebraic approximation, resulting from the interface probing technique, can be regarded as a global approximation to the Schur complement matrix, where the resulting matrix has a local structure. Let us consider the tridiagonal probing technique described in Section 3.3. From the Eqs. 10 and 18, we have that

$$\begin{aligned} S_{EE}\mathbf{v}^i &= \left\{C_{EE}^T + \left(C_{EE}^N - C_{EI}C_{II}^{-1}C_{IE}\right)\right\}\mathbf{v}^i \\ &= \left(\mathbf{w}^T\right)^i + \left(\mathbf{w}^N\right)^i, \end{aligned} \tag{21}$$

where we have split C_{EE} into a tangential component C_{EE}^T and a normal component C_{EE}^N. While the tangential component C_{EE}^T has a tridiagonal structure, it is exactly represented by the 3 probing vectors $\mathbf{v}^i$ given by Eq. 19, resulting in the tangential component approximation. Thus,

$$S_{EE}^{3P} = S_{EE}^{\mathrm{TC}} + \hat{S}_{EE}^N, \tag{22}$$

where the second term is an approximation to the flux, accounting for the normal flow on the boundary. Since

$$\sum_i \mathbf{v}^i = 1, \tag{23}$$

we observe from Eq. 18 that property 15 is satisfied for all interface probing approximations S_{EE}^{nP}. However, from Eq. 21, we observe that the tangential component approximation only preserves constant solutions locally whenever

$$\left(C_{EE}^N - C_{EI}C_{II}^{-1}C_{IE}\right)\mathbf{1} = 0. \tag{24}$$

Here, C_{EE}^N is a diagonal matrix corresponding to the normal flow, where $C_{EE}^N\mathbf{1} = -C_{EI}\mathbf{1}$. Hence,

$$C_{EI}\left(\mathbf{1} + \mathbf{y_I}\right) = 0, \tag{25}$$

where $\mathbf{y_I}$ can be calculated from

$$C_{II}\mathbf{y_I} = C_{IE}\mathbf{1}. \tag{26}$$

In case of **K**-orthogonal grid along the domain boundaries, there are no contributions to the flow between vertex and internal nodes ($C_{IV} = \mathbf{0}$), and it follows that relation 26 is equivalent to a Dirichlet problem with constant boundary values -1,

$$\begin{bmatrix} C_{II} & C_{IB} \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{y_I} \\ \mathbf{y_B} \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \tag{27}$$

Thus, $\mathbf{y_I} = -\mathbf{1}$ and Eq. 25 is satisfied. However, for general grid structures, $C_{IV} \neq \mathbf{0}$, and the tangential component approximation does not satisfy property 15. By probing the S_{EB} (see Eq. 20), the interface probing technique also satisfies property 16 and exactly reproduces constant potential solutions locally, for general grid. Moreover, by choosing solution-based probing vectors, the preconditioner may be constructed to be exact for any predefined characterisation of the solution.

4 Computational efficiency aspects

The main objective of the multiscale methods is efficiency, and the approximations obtained in Section 3 will often be used directly to solve the reduced coarse-scale equation (Eq. 13). In the framework of domain decomposition, we may also use these approximative systems as coarse-scale preconditioners for solving the elliptic fine-scale problem iteratively. In this section, we study the computational work related to these preconditioners.

4.1 Cost of applying the preconditioners

The MCDD preconditioners discussed in this paper are non-overlapping and residual free on the internal degrees of freedom, meaning that the unknowns corresponding to internal nodes are solved exactly. For an iterative solution process, the internal degrees of freedom need only be resolved once. The degrees of freedom related to the edge nodes are eliminated by the different choices of interface approximations $\hat{S}_{EB}$. The linear system related to $\hat{S}_{EB}$ is assumed to be fast to compute. Thus, the main degrees of freedom are proportional to the number of coarse cells (vertex nodes), and the efficiency of the iterative procedure will be measured due to the number of fine-scale solves needed to solve the coarse-scale equation (Eq. 13).

The major computational cost involves C_{II}^{-1} applied to vectors $\mathbf{x}_i$. These operations require solving the local fine-scale problem

$$C_{II}\mathbf{y}_i = \mathbf{x}_i. \tag{28}$$

The solution vectors $\mathbf{y}_i$ can be stored and reused in an iterative process. On each local domain Ω_i', we have four degrees of freedom, one related to each vertex node. Thus, the left-hand side of the coarse-scale equation (Eq. 13) requires solving four fine-scale problems of the form (Eq. 28). The solutions $\mathbf{y}_i$ may be stored as coarse-scale basis functions. For each new iteration, we only need to solve one fine-scale problem corresponding to the new right-hand side term (residual). In practice, we will only update the residual locally on the boundary, in regions where the residual is large.

If the purpose is to solve the coarse-scale equation (Eq. 13) only, the right-hand side term p_I corresponding to internal source terms on the fine scale has to be interpolated onto the vertex nodes. This requires solving two additional local fine-scale problems on each local domain where $p_I \neq 0$. In the case of **K**-orthogonal fine grid, $S_{VE} = C_{VE}$, and we only need to solve one additional local fine-scale problem.

4.2 Cost of constructing the interface approximations

The accuracy and efficiency of the different MCDD preconditioners lie in the construction of $\hat{S}_{EB}$. In this study, we have only considered low-band approximations $\hat{S}_{EB}$, for which the system $\hat{S}_{EB}\mathbf{y}_i = \mathbf{x}_i$ is fast to compute. There is, however, an initial cost related to the construction of some of these approximations. For the tangential component approximation and the reduced boundary condition approximation, $\hat{S}_{EB}$ can be constructed directly from the global system matrix C of the fine-scale equation (Eq. 5). The probing technique (see Eq. 18) requires solving one fine-scale problem per probing vector, i.e. 3 and 5 fine-scale problems for the construction of S_{EB}^{3P} and S_{EB}^{5P}, respectively. The same applies to S_{EB}^{N}. However, the fine-scale problems

Table 1 Number of fine-scale solves related to the different MCDD preconditioners

	S_{EB}^{TC}	S_{EB}^{RBC}	S_{EB}^{3P}	S_{EB}^{5P}	S_{EB}^{N}
Interface approximation	0	0	3	5	3–5[a]
Multiscale method	4	4	7	9	7–9[a]
For each new iteration	+1	+1	+1	+1	+1

[a]The fine-scale problems related to the solution-based probing vectors can be larger, depending on the size of the overlapping region, on which they are computed

related to the construction of the solution-based probing vectors might be larger, depending on the size of the overlapping region on which they are computed.

All interface approximations can be reused in an iterative process. An overview of the cost related to each of the preconditioners is summarized in Table 1.

5 Numerical experiments and results

The MCDD preconditioners described in the previous subsections are tested for several test problems, involving both irregular grids and heterogeneous permeability fields. The different preconditioners will be compared with respect to the accuracy of the first iteration (equivalent to upscaling) and the number of fine-scale solves to obtain a certain tolerance value for the error (fine-scale solver). While there is an initial cost related to the construction of the preconditioners discussed in Section 4, we choose to compare the preconditioners with respect to the number of local fine-scale solves rather than the number of iterations. The global fine-scale solution will here serve as the reference solution for the approximated solutions.

For the iterative scheme, we apply the preconditioned generalized minimal residual (GMRES) [29], where we compare the efficiency of the different MCDD preconditioners based on the tangential component approximation (MCDD-TC), the reduced boundary condition (MCDD-RBC), the tridiagonal and pentadiagonal probing technique (MCDD-3P and MCDD-5P) and the interface probing technique based on neighbour connections (MCDD-N(n)). Here, n is the number of overlapping sub-domains, used to compute the solution-based probing vectors. For the numerical results of MCDD-N, we have considered two solution-based and two oscillatory (see Eq. 19) probing vectors. The two solution-based vectors are constructed, so to capture the principal flow in the horizontal and vertical direction, respectively. As boundary conditions for the overlapping domain, we have applied a unit pressure drop in one direction and no-flow conditions in the other. The MCDD preconditioners are also compared to the unpreconditioned GMRES method, referred to as MCDD-unprec. For large linear systems, the GMRES algorithm requires large information storage, and a restarted version of GMRES may improve the efficiency of the algorithm. For our numerical experiments, we do not consider restarts.

Monotonicity For the fine-scale discretisation, we consider the multipoint flux approximation (MPFA) method as described in [2]. More precisely, we have applied the MPFA O-method, which guarantees continuity of flux over each interface and continuity in pressure at the mid-point of each interface. For general quadrilateral grids, the discretisation will lead to a nine-point stencil on the fine scale. For **K**-orthogonal fine grid, the method will reduce to a five-point scheme similar to the two-point flux approximation method. In any case, the coarse-scale operators resulting from the various MCDD preconditioners will in general be nine-point stencils. Common for all these stencils is that they do not guarantee monotone methods [27] and may produce non-physical oscillations on the coarse scale for certain anisotropies on the coarse scale. This may again lead to incorrect flow fields on the fine scale. The solution can, however, be improved through local iterations on the fine scale. Monotonicity of the coarse-scale operator for the MSFV method has been studied in [17]. A compact coarse-scale operator with improved monotonicity properties was proposed, which reduces to a seven-point stencil in the limit of homogeneous permeability. However, the compact operator did not lead to improved robustness for the MCDD preconditioners when solving heterogeneous flow problems on irregular grids.

Error norms For the accuracy of the multiscale solution in potential and flux after one iteration, we

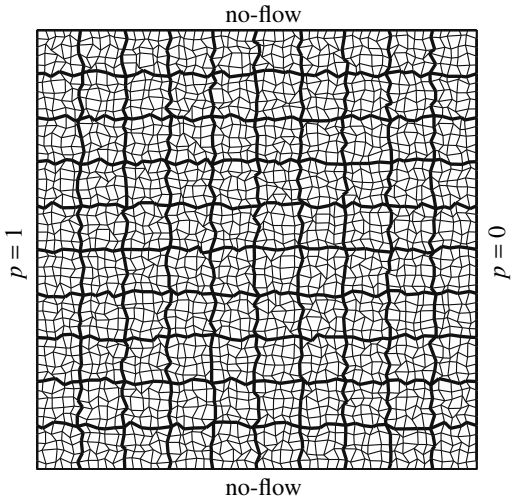


Fig. 3 Computational mesh. The figure shows a 50×50 irregular Cartesian grid on the fine scale. The degree of perturbation for this grid is $\epsilon = 0.4$. The *bold faces* show the coarse-scale grid, where we have applied a uniform coarsening of 5×5 grid cells

consider the error in L^∞-norm, as it relates to monotonicity. We denote

$$\delta_u = \frac{\| u^1 - u^\infty \|_\infty}{\max(u^\infty) - \min(u^\infty)}, \tag{29}$$

$$\delta_f = \frac{\| f^1 - f^\infty \|_\infty}{\max(f^\infty) - \min(f^\infty)}, \tag{30}$$

as the multiscale errors for the potential u and the flux f, respectively. In Eqs. 29 and 30, u^1 and f^1 represent the approximated multiscale solutions (solutions after one iteration), while u^∞ and f^∞ represent the converged fine-scale solutions for potential and flux. The reconstruction of a mass conservative flux on the fine scale is performed by a post-processing step similar to the MSFV method. The errors within the GMRES algorithm is evaluated in the L^2-norm.

Grids The MCDD preconditioners are tested on irregular rough grids. These grids are generated by random perturbations on the uniform Cartesian grid. If we consider $\mathbf{x}$ to be the coordinates of the initial uniform grid, irregular fine grids are generated by

$$\mathbf{x}_\epsilon = \mathbf{x} + \epsilon \mathbf{r} h, \tag{31}$$

where h is the fine-grid cell size of the initial uniform grid, $\mathbf{r} \in [-1, 1]$ is a random variable and $\epsilon \in [0, 0.5]$ is the degree of perturbation. Figure 3 shows one example of a simulation grid, where $\epsilon = 0.4$.

5.1 Uniform flow on rough grids

We first consider the elliptic problem (Eq. 1), where we neglect the source terms ($q = 0$) and apply a homogeneous and isotropic permeability tensor with diagonal elements, $k = 1$. We use no-flow boundary conditions on the top and bottom boundary and a unit pressure drop in the horizontal direction, as shown in Fig. 3. From the choice of boundary conditions, the flow on

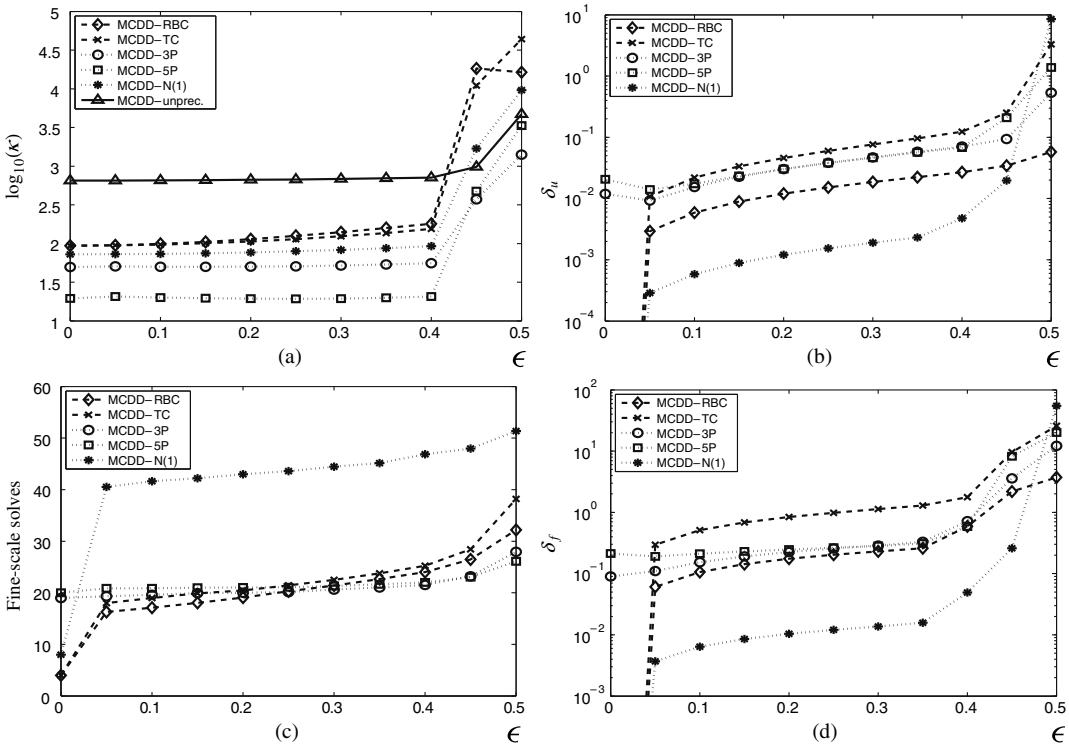


Fig. 4 A comparison of the different MCDD preconditioners for the elliptic problem with constant coefficient. Here, the *horizontal axis* represent the degree of roughness for the irregular grid. **a** The spectral condition number and **c** the number of fine-scale solves to meet a tolerance of 10^{-8} for the different preconditioners. In **b** and **d**, we plot the multiscale error, measured in L^∞-norm, obtained after one iteration with GMRES. All results are means of 50 realisations of random generated rough grids

the fine scale will be uniform. We will solve the homogeneous problem on irregular grids, to study the grid dependence of the different MCDD preconditioners. From the construction of control volume methods for elliptic problems, the expression for the flux depends on both geometry and the permeability tensor **K**, where perturbations on the grid will have similar effects as perturbing **K**.

Figure 4a shows the comparison of the spectral condition number for the fine-scale system, when applying the different MCDD preconditioners. We observe that the preconditioners based on oscillatory probing vectors (MCDD-3P and MCDD-5P) result in significantly lower condition numbers than the preconditioners based on harmonic (MCDD-RBC) and solution-based vectors (MCDD-N). We also observe that the MCDD-3P and MCDD-5P preconditioners are more robust with respect to perturbations on the fine-scale grid, for perturbations up to about 40% of the fine-grid cell size. This indicates that the interface approximations, commonly used for multiscale and upscaling methods, are not as well suited as multiscale preconditioners for the fine-scale problem. For perturbations above 40% of the fine-grid cell size, none of the low-band approximations for the Schur complement discussed in this paper are robust. Such rough grids will include non-convex and highly distorted grid cells (see Fig. 3).

In Fig. 4b, we show the number of fine-scale solves, required to solve the homogeneous fine-scale problem. For Cartesian grid ($\epsilon = 0$), the local tangential flow approximations used in MCDD-RBC and MCDD-TC reduce to the exact linear boundary conditions, and the problem is solved exactly in one iteration. However, for perturbed grids ($\epsilon > 0$), the interface probing approximation used in (MCDD-3P and MCDD-5P) are more robust, and the preconditioners need about the same amount of computational work. The MCDD-N preconditioner is designed as an upscaling technique and is not suited as a preconditioner for the iterative scheme. Note also that the MCDD-TC preconditioner, which is a consistent approximation to the tangential flow along perturbed local boundaries, is less accurate than the inconsistent two-point flux approximation used in MCDD-RBC.

In some cases, we cannot afford to iterate on the fine-scale solution, and we would like to reconstruct a fine-scale approximation from the solution of the coarse-scale problem. This is equivalent with one iteration on the fine-scale solution. Figure 4b, d show the accuracy of the multiscale solutions, obtained after one iteration on the fine-scale solution. While the oscillating probing vectors used in MCDD-3P and MCDD-5P result in more robust approximations for the iterative process, they are not as well suited for upscaling. The reduced boundary approximation (MCDD-RBC) has previously shown to be accurate for problems involving regular Cartesian grids; however, it is not as accurate for problems involving irregular grid. By introducing solution-based probing vectors in MCDD-N, we are able to obtain much more accurate approximations to the fine-scale solution at first iteration, for problems involving irregular grids.

5.2 Heterogeneous problems on rough grids

Next, we consider elliptic problems with variable coefficients $k(\mathbf{x})$. We consider isotropic, log-normal permeability on the fine scale, as shown in Fig. 5. Local sub-scale heterogeneities may introduce strong non-local couplings between boundary unknowns on the local sub-domains, as discussed in Section 3.4, which are more difficult to capture within local interface approximations. From Fig. 6, we see that the MCDD preconditioners resulting from the interface probing technique are able to better capture these non-local couplings and are more robust with respect to fine-scale perturbations on the fine grid. The tangential component approximation is not able to capture the correct flow normal to these local boundaries and has a larger dependency on the degree of perturbation ϵ. The multiscale methods have shown to have difficulties with capturing large anisotropies in the fine-scale flow field, especially for

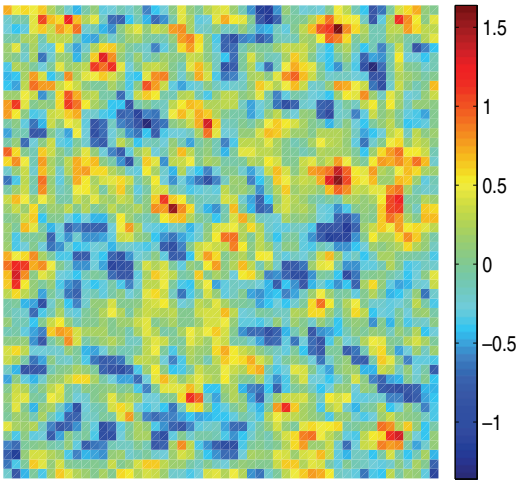


Fig. 5 A generated log-normal permeability field, with standard deviation of 0.5 and a correlation length of 3 fine-grid cells in both *x*- and *y*-direction. The figure shows the base 10 logarithm of the permeability

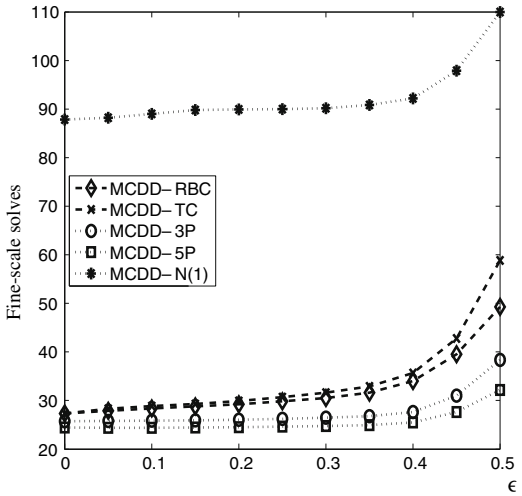


Fig. 6 The number of fine-scale solves needed to reach a tolerance of 10^{-8} when solving the elliptic problem for heterogeneous log-normal permeability when applying different MCDD preconditioners. The *horizontal axis* represents the degree of roughness for the perturbed grid. All results are mean values from 20 realisations of permeability fields and 20 realisations of randomly generated rough grids

problems with diagonal channels going through corner cells of the local sub-domains [22]. This is because the coupling, C_{VI}, between vertex and internal nodes, is neglected in the tangential component approximation. For problems involving irregular grids, MCDD-RBC and MCDD-TC do not preserve constant solution (see Section 3.6). By placing a high permeable channel between two opposite corners of the global fine grid, the tangential approximation (MCDD-TC and MCDD-RBC) is not able to capture the diagonal flow over the vertex nodes and requires solving many more fine-scale problems (iterations), even for Cartesian grid. The interface probing approximation is based on an algebraic approximation to the Schur complement and is more or less independent upon the geometry and principle directions on the fine scale. In Fig. 7a, b, we again observe that the MCDD-N preconditioner in general provides more accurate approximations to the solution after one iteration. The results in Fig. 7b even show that the interface probing technique, using oscillating probing vectors, results in more accurate approximations to the multiscale flux on the fine scale than the more commonly used tangential component approximation. As for regular upscaling techniques, the quality of the solution-based probing vectors depends on the induced local boundary conditions and the size of the overlapping domains.

5.3 Realistic porous media

In realistic porous media, we might be faced with complex geological layers, where the fine-scale permeability has long and anisotropic correlation lengths.

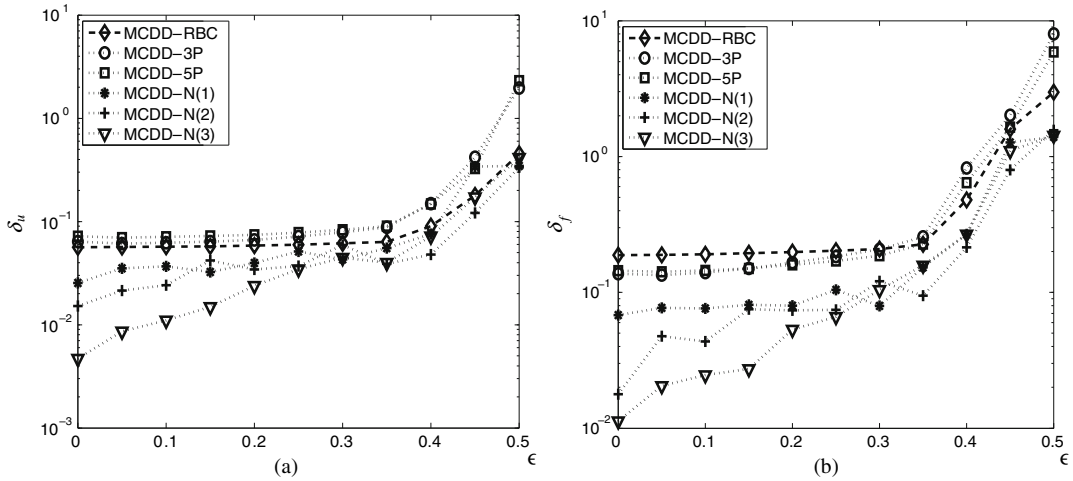


Fig. 7 Upscaling of log-normal permeability. The figures show the accuracy of the multiscale solution obtained after one iteration. In **a**, we plot the L^∞-error of the potential, and **b** shows the L^∞-error of the flux. All results are the truncated (80%) mean values from 20 realisations of permeability fields and 20 realisations of randomly generated rough grids

We consider the two-dimensional cross sections corresponding to the top and bottom layer of the 10th SPE comparative solution project [8] (see Fig. 8a, b). This test case is aimed at comparing different upscaling procedures and has been extensively used for testing multiscale methods. While the geometry is a simple uniform Cartesian grid, the permeability contrasts are rather challenging for reservoir simulators. The model for each layer consists of a 60×220 uniform Cartesian fine grid. To test the multiscale preconditioners, we further apply a 12×20 coarse grid. We use similar boundary conditions as in the previous test cases. The top layer is a Tarbert formation, while the bottom layer is fluvial. From the bottom layer (Fig. 8b), we clearly see channels with long correlation lengths throughout the reservoir. These channelized flow paths will result in anisotropic permeability on the coarse scale.

In Table 2, we compare the efficiency of the preconditioners based on the tangential flow approximation (MCDD-RBC) and the interface probing approximation (MCDD-3P and MCDD-5P). Since the simulation grid is **K**-orthogonal, the MCDD-TC and MCDD-RBC are equivalent formulations. Thus, we only report the results of the MCDD-RBC. For the top layer (Tarbert formation; see Fig. 8a), there are quite large correlation lengths in the fine-scale permeability, with small subscale variations on each coarse block. All the preconditioners give quite good results for this problem. Note that even though the construction of the MCDD-3P and MCDD-5P requires solving 3 and 5 local fine-scale problems initially, they are still more efficient for solving the fine-scale problem iteratively. The bottom layer contains a fine-scale permeability with anisotropic correlation lengths. This channelized reservoir does not have a clear scale separation, which makes it more challenging to construct efficient multilevel preconditioners. Our results clearly show that the oscillating vectors of the interface probing technique (MCDD-3P and MCDD-5P) are better suited for capturing the high-frequency error in the solution for channelized flow.

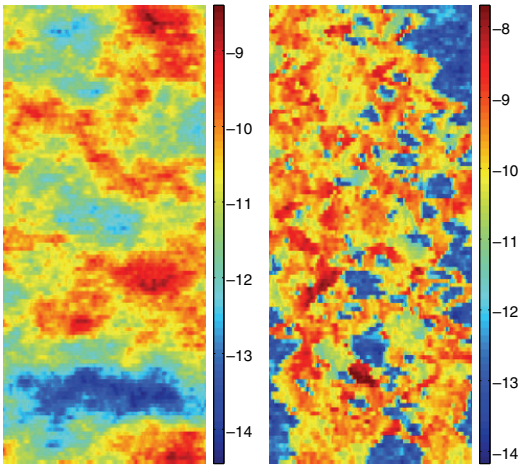


Fig. 8 Base 10 logarithm of permeability for the bottom and top layer of the SPE 10th comparative solution project. Here, **a** shows layer 1 (*top*) and **b** layer 85 (*bottom*)

Table 2 Number of fine-scale solves needed to reach a tolerance 10^{-8} in the L^2-norm

Layer	MCDD-RBC	MCDD-3P	MCDD-5P
1	33	30	30
85	347	167	146

Table 3 shows the accuracy of the multiscale solution (first iteration). The results from layer 1 (top layer) show that we can construct quite accurate multiscale approximations to the fine-scale solution after only one iteration. Thus, there exists an exact representation of the flow on the coarse scale. By increasing the region of the overlap, the solution within the target region is less effected by the boundary conditions and more determined by the local variations in the fine-scale permeability. Thus, different boundary value problems will give more similar results, and the solution-based probing vectors can get close to linearly dependent. This may result in an ill-conditioned system for computing the interface approximation of MCDD-N, in which case we will have to reduce the number of solution-based probing vectors. A natural extension, which is beyond the scope of this paper, is to adapt the overlapping regions based on the fine-scale residual. This idea has been studied in [26].

In the results for layer 85 (bottom layer), the anisotropies in the upscaled permeability produce non-physical oscillations in the solution for the coarse scale. In fact, the coarse-scale operators, for all the tested multiscale methods, fail to meet the requirements for monotonicity [28]. For completeness, we also tested the multiscale control volume method with linear boundary conditions as well as the MCDD-N preconditioner including global information. The MSFV method using linear boundary conditions has shown to give smaller errors then the MSFV method using reduced boundary conditions, for some problems involving highly anisotropic porous media [23]. By including global information in the MCDD-N preconditioners, the local approximations $\hat{S}_{EB}$ exactly reproduces the fine-scale flow field. However, these preconditioners also yield

Table 3 The error of the multiscale solution after one iteration, measured in the L^∞-norm

Layer	MCDD-RBC	MCDD-3P	MCDD-5P	MCDD-N(1)	MCDD-N(2)	MCDD-N(3)
1 (u)	0.15	0.14	0.19	0.032	0.043	0.018
1 (f)	0.069	0.098	0.12	0.012	0.014	0.037
85 (u)	0.47	11	155	2.6	0.54	0.53
85 (f)	0.43	0.58	8.1	2.8	0.26	0.17

Here, (u) denotes the fine-scale potential solution and (f) the fine-scale flux

non-monotone coarse-scale operators and non-physical oscillations in the coarse-scale potential solution. Table 3 shows the large errors in the fine-scale potential solution, caused by non-physical oscillations on the coarse scale. These oscillations in turn result in wrong boundary conditions for the recalculation of fine-scale fluxes. As it is known that there exist cases where no nine-point scheme is monotone [21], changing the discretisation scheme might not be enough. In order to construct accurate upscaling methods for problems with anisotropic coarse-scale permeability, one approach might be to adapt the coarse-scale grid to the principal directions of the flow on the coarse scale. However, it is not clear how to construct a coarse grid which will guarantee monotonicity. In any case, we would need a robust multiscale framework which can handle simulations on general grids.

6 Conclusion

We have tested the efficiency and accuracy of MCDD preconditioners for two-dimensional heterogeneous elliptic problems on irregular grids. In the case of solving only one iteration, the MCDD preconditioners are similar to standard multiscale control volume methods where only the global coarse-scale equation is solved. Since we approximate the Schur complement, rather than the local basis functions, the proposed multiscale methodology is convergent to the fine scale. Our numerical experiments have shown that the reduced boundary condition (MCDD-RBC), commonly applied for constructing multiscale methods, is not robust with respect to perturbations on the fine scale. In the case of non-**K**-orthogonal grids, the reduced boundary condition does not preserve constant solutions, which makes it unsuitable for constructing multiscale preconditioners for flow problems on irregular grid structures or anisotropic permeability.

We have presented a more robust multiscale framework, based on the interface probing technique, for solving heterogeneous elliptic problems on irregular grids. While the MCDD-3P and MCDD-5P, using oscillating probing vectors, act as more efficient multiscale preconditioners for the fine-scale problem, solution-based probing vectors can be constructed (MCDD-N) to give a more accurate representation of the coarse scale. Both of these preconditioners are purely algebraic upscaling techniques; thus, they are independent upon geometry and extendible to multiscale simulations on unstructured grids. Moreover, the interface probing technique can be seen as a more consistent way of incorporating global fine-scale information into the coarse-scale basis functions, as it is designed to exactly reproduce the solution of given boundary value problems.

When applied as a preconditioner for Krylov-type algorithms, the objective is to efficiently reduce the residual. At first iteration, the residual vector is exactly equal to the boundary conditions provided by the right-hand side vector p. Thus, multiscale methods based on global information are accurate. After the first iteration, the residual is proportional to the error and is similar to oscillating noise. Thus, a simple fine-scale preconditioner, which effectively smooths the error, is to be preferred.

For certain anisotropy relations on the coarse scale, it is not possible to construct monotone nine-point stencils on the coarse scale. Thus, we may need to iterate on the fine-scale residual, in order to reduce the non-physical oscillations produced by the coarse scale. The MCDD preconditioners can be adapted to serve as either an accurate approximation to the coarse-scale problem (multiscale method) or an efficient preconditioner for the fine-scale problem.

References

1. Aarnes, J.E.: 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 (electronic) (2004)

2. Aavatsmark, I.: An introduction to multipoint flux approximations for quadrilateral grids. Comput. Geosci. **6**(3-4), 405–432 (2002). Locally conservative numerical methods for flow in porous media
3. Arbogast, T.: Implementation of a locally conservative numerical subgrid upscaling scheme for two-phase Darcy flow. Comput. Geosci. **6**(3-4), 453–481 (2002). Locally conservative numerical methods for flow in porous media
4. Axelsson, O.: Iterative Solution Methods. Cambridge University Press, New York (1994)
5. Bunch, J.R.: Stability of methods for solving Toeplitz systems of equations. SIAM J. Sci. Statist. Comput. **6**(2), 349–364 (1985)
6. Chan, T.F.: Analysis of preconditioners for domain decomposition. SIAM J. Numer. Anal. **24**(2), 382–390 (1987)
7. Chan, T.F.C., Mathew, T.P.: The interface probing technique in domain decomposition. SIAM J. Matrix Anal. Appl. **13**(1), 212–238 (1992)
8. Christie, M., Blunt, M.: Tenth SPE comparative solution project: A comparison of upscaling techniques. SPE Reserv. Evalu. Eng. **4**(4), 308–317 (2001)
9. Chu, J., Efendiev, Y., Ginting, V., Hou, T.Y.: Flow based oversampling technique for multiscale finite element methods. Adv. Water Resour. **31**(4), 599–608 (2008)
10. Durlofsky, L., Efendiev, Y., Ginting, V.: An adaptive local–global multiscale finite volume element method for two-phase flow simulations. Adv. Water Resour. **30**(3), 576–588 (2007)
11. Efendiev, Y., Ginting, V., Hou, T., Ewing, R.: Accurate multiscale finite element methods for two-phase flow simulations. J. Comput. Phys. **220**(1), 155–174 (2006)
12. Farmer, C.L.: Upscaling: a review. Int. J. Numer. Methods Fluids **40**(1–2), 63–78 (2002). ICFD Conference on Numerical Methods for Fluid Dynamics (Oxford, 2001)
13. Ginting, V.: Analysis of two-scale finite volume element method for elliptic problem. J. Numer. Math. **12**(2), 119–141 (2004)
14. Golub, G.H., Mayers, D.: The use of preconditioning over irregular regions. In: Computing Methods in Applied Sciences and Engineering, VI (Versailles, 1983), pp. 3–14. North-Holland, Amsterdam (1984)
15. Graham, I.G., Lechner, P.O., Scheichl, R.: Domain decomposition for multiscale PDEs. Numer. Math. **106**(4), 589–626 (2007)
16. Hajibeygi, H., Bonfigli, G., Hesse, M.A., Jenny, P.: Iterative multiscale finite-volume method. J. Comput. Phys. **227**(19), 8604–8621 (2008)
17. Hesse, M.A., Mallison, B.T., Tchelepi, H.A.: Compact multiscale finite volume method for heterogeneous anisotropic elliptic equations. Multiscale Model. Simul. **7**(2), 934–962 (2008)
18. Hou, T.Y., Wu, X.H.: A multiscale finite element method for elliptic problems in composite materials and porous media. J. Comput. Phys. **134**(1), 169–189 (1997)
19. Hou, T.Y., Wu, X.H., Cai, Z.: Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. Math. Comput. **68**(227), 913–943 (1999)
20. Jenny, P., Lee, S.H., Tchelepi, H.A.: Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. J. Comput. Phys. **187**(1), 47–67 (2003)
21. Keilegavlen, E., Nordbotten, J.M., Aavatsmark, I.: Sufficient criteria are necessary for monotone control volume methods. Appl. Math. Lett. **22**(8), 1178–1180 (2009)
22. Kippe, V., Aarnes, J.E., Lie, K.A.: A comparison of multiscale methods for elliptic problems in porous media flow. Comput. Geosci. **12**(3), 377–398 (2008)
23. Lunati, I., Jenny, P.: Treating highly anisotropic subsurface flow with the multiscale finite-volume method. Multiscale Model. Simul. **6**(1), 308–318 (electronic) (2007)
24. Mandel, J.: Balancing domain decomposition. Commun. Numer. Methods Eng. **9**(3), 233–241 (1993)
25. Mandel, J., Sousedík, B.: Coarse spaces over the ages Arxiv e-prints (2009)
26. Nordbotten, J.M.: Adaptive variational multiscale methods for multiphase flow in porous media. Multiscale Model. Simul. **7**(3), 1455–1473 (2008)
27. Nordbotten, J.M., Aavatsmark, I., Eigestad, G.T.: Monotonicity of control volume methods. Numer. Math. **106**(2), 255–288 (2007)
28. Nordbotten, J.M., Bjørstad, P.E.: On the relationship between the multiscale finite-volume method and domain decomposition preconditioners. Comput. Geosci. **12**(3), 367–376 (2008)
29. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. **7**(3), 856–869 (1986)
30. Sandvin, A., Nordbotten, J.M., Aavatsmark, I.: A unified framework of upscaling and domain decomposition. In: XVIII Conference on Computational Methods in Water Resources (2010)
31. Smith, B.F., Bjørstad, P.E., Gropp, W.D.: Domain Decomposition. Cambridge University Press, Cambridge (1996). Parallel multilevel methods for elliptic partial differential equations

Paper B

A Unified Multilevel Framework of Upscaling and Domain Decomposition *

* In Proceedings of the XVIII International Conference on Water Resources, CIMNE Barcelona, 2010.

XVIII International Conference on Water Resources
CMWR 2010
J. Carrera (Ed)

A UNIFIED MULTILEVEL FRAMEWORK OF UPSCALING AND DOMAIN DECOMPOSITION

Andreas Sandvin*†, Jan M. Nordbotten†* and Ivar Aavatsmark*

*Centre for Integrated Petroleum Research (CIPR)
Allegt. 41, 4. etg. 5007 Bergen, Norway
e-mail: andreas.sandvin@uni.no, web page: http://www.cipr.uni.no/

†University of Bergen, Department of Mathematics
Johannes Brunsgate 12, 5008 Bergen, Norway
web page: http://www.uib.no/math

Key words: Multiscale, preconditioner, porous media

Summary. We consider multiscale preconditioners for a class of mass-conservative domain-decomposition (MCDD) methods. For the application of reservoir simulation, we need to solve large linear systems, arising from finite-volume discretisations of elliptic PDEs with highly variable coefficients. We introduce an algebraic framework, based on probing, for constructing mass-conservative operators on a multiple of coarse scales. These operators may further be applied as coarse spaces for additive Schwarz preconditioners. By applying different local approximations to the Schur complement system based on a careful choice of probing vectors, we show how the MCDD preconditioners can be both efficient preconditioners for iterative methods or accurate upscaling techniques for the heterogeneous elliptic problem. Our results show that the probing technique yield better approximation properties compared with the reduced boundary condition commonly applied with multiscale methods.

1 INTRODUCTION

Challenges within flow in porous media include complex geological structures with spatial variability on multiple scales. Reservoir simulations (i.e. groundwater flow, oil recovery, CO_2 storage) often involve large spatial scales, where we need to solve large linear systems repeatedly in time. The potential u within the reservoir is governed by an elliptic PDE, with highly variable tensor coefficients $k(\mathbf{x})$,

$$-\nabla\cdot(\mathbf{K}(\mathbf{x})\nabla u(\mathbf{x})) = q \qquad \mathbf{x}\in\Omega. \tag{1}$$

Here Ω is a two dimensional domain, $\mathbf{K}$ is the permeability and q represents the source terms. Standard two-level domain-decomposition methods, using e.g. piecewise linear

basis functions for the coarse space, where the oscillating coefficients are assumed to be resolved at the coarse scale, in general perform poorly for these problems[5], where the condition number will have a dependence on the largest ratio in coefficients k. Multiscale methods[3] is introduced as an upscaling technique for constructing robust coarse spaces, with harmonic basis functions. The multiscale problem is solved directly on the coarse scale, and resolved on the fine scale as a linear combination of the basis functions. This is equivalent with one fine-scale iteration, using the multiscale method as a two-level additive Schwarz preconditioner for domain decomposition[8].

In some cases the solution may be too expensive to compute on the fine scale at each time step, and we are forced to do upscaling. However, the coarse-scale operator may produce non-physical oscillations in the solution[6], which can only be reduced by iterating on the fine-scale residual. We will introduce an adaptive framework for constructing coarse spaces for the class of mass-conservative domain-decomposition (MCDD) methods introduced by Nordbotten and Bjørstad[8], which can act as either an accurate upscaling method, or an efficient preconditioner. The framework is based on algebraic approximations to the Schur complement, by using the interface probing technique[2]. Most multiscale methods are based on a geometric upscaling of fine-scale information, however, this does not naturally generalize from two- to multi-level methods, or arbitrary geometries and dimensions. The probing technique on the other hand is only based on neighbour element relations, which is independent upon the underlying geometry, and we show in Section 3 how this approach can be extended to construct multilevel preconditioners. For the application of upscaling, we observe that we can obtain much more accurate coarse spaces by applying a set of solution-based probing vectors. A more detailed discussion is given in Section 4. To demonstrate the flexibility of the proposed methodology, we show in Section 5 a two-step preconditioner, where the first step is an upscaling of the fine-scale system, and the second step is a preconditioner for the upscaled system.

2 MCDD

2.1 Fine-scale system

We consider linear systems arising from fine-scale discretisation on cell centred grids, consisting of finite volumes ω_i. Here, the permeability tensors K are assumed to be constant on each volume ω_i, but may be discontinuous at the interfaces γ_{ij}, between two neighbouring volumes ω_i and ω_j. By integrating (1) over ω_i, and applying Green's theorem we obtain the integral equation for conservation of incompressible fluids,

$$\int_{\partial\omega_i} \mathbf{F}\cdot\nu_{\mathbf{i}} = \int_{\omega_i} q. \tag{2}$$

Here, $\mathbf{F} = -\mathbf{K}(\mathbf{x})\nabla u(\mathbf{x})$ represents the Darcy flux and ν_i is the outward normal vector to $\partial\omega_i$. A discretisation of (2) yields local mass conservation within ω_i, and the global discrete system of fine-scale equations takes the form

$$Au = b. \tag{3}$$

The system matrix A is in general non-symmetric.

2.2 Grids and scales

A primal coarse grid $\Omega = \bigcup \Omega_i$ is defined, such that each primal coarse cell Ω_i is a set of finite volumes ω_j on the fine scale and the interfaces of Ω_i align with the interfaces on the fine scale. The centre-most volume on the fine scale within Ω_i is defined as the coarse node V. By repeating the process, we can form a hierarchy of cell centred coarse grids Ω^l. As a preprocessing step we require mass conservation between all cells Ω_i^l on each level l, on which we will compute the solution. This gives us the possibility to construct a mass-conservative flow field on level l, from the approximate solution $\hat{u}$ at the same level. Let $A_i^l = R_i^l A$ be the restriction of the system matrix A to Ω_i^l. Acting on each primal coarse cell Ω_i^l, the integration matrix M_i^l sums all the rows of A_i^l into the row of the coarse node V. More precisely,

$$M_i^l = I + e_{iV}^l \left(\underline{1} - e_{iV}^l\right)^T, \tag{4}$$

where I is the identity matrix, e_{iV}^l is the unit vector identifying the row of the coarse node and $\underline{1}$ is the vector entirely filled with ones. We apply (4) on the linear system (3), which gives us the MCDD system

$$Cu = p, \tag{5}$$

where

$$C = Q^{(l)} A; \quad p = Q^{(l)} b \quad \text{and} \quad Q^{(l)} = \sum_l \sum_i \left(R_i^l\right)^T M_i^l R_i^l.$$

A dual coarse grid Ω' is also introduced, s.t. all the coarse nodes defined on the primal coarse grid Ω represent vertex nodes on the dual grid. A continuous path of connecting cells on the finer level, connecting two neighbouring primal coarse nodes, further define the interfaces on the dual grid. The boundary of Ω_i' consist of boundary nodes, sub-divided into edge and vertex nodes, as shown in Fig. 1(b). While the MCDD system of equations is defined on the primal grid, all local operations will be carried out on the dual grid.

3 SCHUR COMPLEMENT SYSTEM

We consider non-overlapping sub-domains on Ω', where the sub-domains only share common sub-interfaces. We will denote the boundary nodes and internal nodes on each sub-domain by subscript B and I, respectively. The boundary nodes B are further sub-divided into edge nodes E and vertex nodes V. The vertex nodes will here be of special

importance, since they define our coarse-scale degrees of freedom. By grouping the unknowns corresponding to the internal nodes I in u_I, and the unknowns located on the local interfaces of Ω_i' in u_B, we reorder the fine-scale problem (5), writing

$$\begin{bmatrix} C_{II} & C_{IB} \\ C_{BI} & C_{BB} \end{bmatrix} \begin{bmatrix} u_I \\ u_B \end{bmatrix} = \begin{bmatrix} p_I \\ p_B \end{bmatrix}. \tag{6}$$

All internal unknowns are now decoupled on each local sub-domain, and can formally be eliminated by a block Gaussian elimination of (6). This gives us the Schur complement system $Su_B = p_B - C_{BI}C_{II}^{-1}p_I$ on the interface, where $S = C_{BB} - C_{BI}C_{II}^{-1}C_{IB}$. The Schur complement system can be shown to be better conditioned[1], however the system is still quite expensive to solve. The multiplication of S with a vector $\mathbf{x}$ will require solving a Dirichlet problem on each local sub-domain Ω_i'.

By a similar grouping of unknowns u_E and u_V on the edge nodes E and vertex nodes V, respectively, we can write

$$\begin{bmatrix} S_{EE} & S_{EV} \\ S_{VE} & S_{VV} \end{bmatrix} \begin{bmatrix} u_E \\ u_V \end{bmatrix} = \begin{bmatrix} g_E \\ g_V \end{bmatrix}. \tag{7}$$

The reordered Schur complement matrix now has a sparse block structure, however each block is in general dense. We want to construct a simple approximation to $S_{EE}^{-1}S_{EV}$, and a reduced system-matrix A_c on a coarser scale, where A_c can be solved directly or applied as a coarse space for an additive Schwarz preconditioner. Note that this only modifies the matrices belonging to the equations for the edge unknowns. Thus, our solution still have the property of conserving mass on the coarse scale. Another observation is that the coarse-scale operator A_c will have the same general structure as the fine-scale operator A. This means that the same operations may be applied for A_c, and we can recursively construct mass-conservative operators on a hierarchy of levels.

For the construction of 2-level additive Schwarz preconditioners, numerical experiments indicate that the property of mass conservation may result in better conditioned problems for the fine-scale, see Figure 1(a), however this does not necessarily apply for multi-level Schwarz preconditioners. The approximation induced on the edges for the highest level, may destroy the property of mass conservation on all intermediate levels, meaning mass conservation can only be guaranteed on two scales simultaneously, that being the coarsest and finest scale. Consequently, the MCDD operators may be better suited as input parameters for a multigrid-type preconditioner, where the restricted residual is applied directly on each mass-conservative level.

4 INTERFACE PROBING APPROXIMATION

For the interface approximation on the local edge nodes, we consider the probing technique introduced in Chan and Mathew[2] and references therein. The aim is to construct an approximation of the Schur complement matrix on the edge, such that

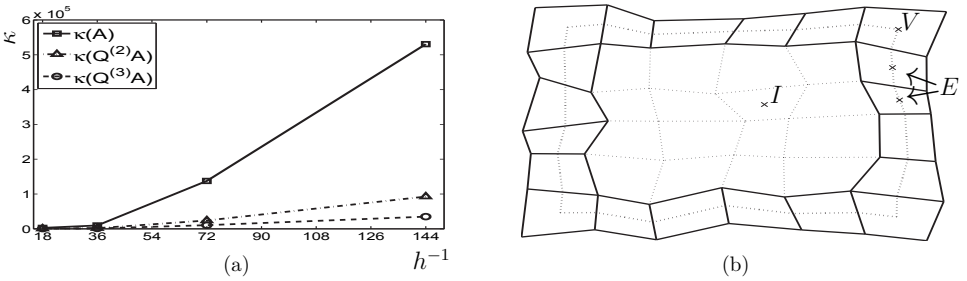


Figure 1: Figure (a) shows the condition number of the fine-scale solution with mass conservation on multiple levels, as we refine the grid. Figure (b) shows a single sub-domain Ω_i', where the bold lines indicate the boundary. The boundary cells are shared between two sub-domains, where each sub-domain only compute half-fluxes along the boundary.

$$\hat{S}_{EB}\mathbf{v}^i = S_{EB}\mathbf{v}^i = \mathbf{w}^i, \tag{8}$$

for some carefully chosen linearly independent probing vectors $\mathbf{v}^i$. Originally, the probing technique was applied on the square matrix S_{EE}, where the choice of probing vectors $\mathbf{v}^i = \sum_{j=i \bmod(n)} \mathbf{e}_j$ would lead to a low-band approximation of the Schur complement, which is fast to invert. The method was motivated by the observation of Golub and Mayers[4], that the coefficients of the Schur complement often had a rapid decay away from the diagonal, following the relation $|S_{ij}| = O\left(|i-j|^{-2}\right)$. In the case of anisotropic coefficients K_{ij} in the elliptic problem (1), we may have large off-diagonal elements, and the relation for the coefficients does not apply. If we instead probe S_{EB} on each Ω_i', the probing vectors $\mathbf{v}^i$ can be interpreted as boundary values for a Dirichlet problem on Ω_i'.

4.1 Solution-based probing vectors

We introduce solution-based probing vectors, to mimic upscaling techniques and therefore provide better approximation properties for heterogeneous and anisotropic problems. Let S_{EB} be the Schur complement on the edge, restricted to a single domain Ω_i', where B denotes the boundary of Ω_i' (see Figure 1(b)). The application of a probing vector $\mathbf{v}^i$ with the Schur complement matrix S_{EB} in (8), requires solving a Dirichlet problem on Ω_i', with boundary values $\mathbf{v}^i$. Thus, we may construct accurate upscaling methods by choosing probing vectors $\mathbf{v}^i$ that capture the important physical features of the local fine-scale solution. In fact, if the probing vectors could be chosen as the exact fine-scale solution restricted to the local boundary of Ω_i', the local approximation is exact, and the solution converges in one iteration. For the construction of solution-based probing vectors, we solve flow problems on a local domain covering the local support of S_{EB} (see Figure 1(b)). The restriction of the local solution to the boundary B will then be used as a probing vector, which will belong to the null space of the S_{EB}. We construct the

interface approximation $\hat{S}_{EB}$, from (8), filling only the diagonal elements and the element neighbours, corresponding to the largest couplings in C_{EB}. If the number of probing vectors are larger then the number of non-zero couplings in C_{EB}, we represent the additional neighbour connection(s) by an average of the remaining boundary elements. For each local problem, the calculation of a solution-based probing vector requires solving a local fine-scale problem on $\Sigma \supset \Omega_i'$. However, as the resulting probing vector $\mathbf{v}^i \in \text{null}(S_{EB})$ we do not need to solve the local fine-scale problem relating to the multiplication with S_{EB}. It follows from relation (8), that the preconditioner will be exact for those fine-scale problems captured by the local solutions. We will denote the preconditioner, MCDD-N, since its construction only depends on Neighbour relations.

4.2 Oscillating probing vectors

The interface probing preconditioners discussed in Chan and Mathew[2], are based on oscillating vectors, $\mathbf{v}^i = \sum_{j=i \bmod(n)} \mathbf{e}_j$. These preconditioners may give more robust approximations to the Schur complement, however they lack the physical interpretation provided by the solution-based probing vectors. Similar to standard upscaling techniques, the quality of the solution-based probing vectors will be case dependent. For robustness, all the probing vectors should not belong to the null space of S_{EB}. In fact, experience shows that by applying a few oscillating vectors, we get a more robust method. Thus, a combination of oscillating and solution-based probing vectors seems to be favourable. It is important that the chosen probing vectors are linearly independent. If two of the vectors $\mathbf{v}^i$ are close to being linearly dependent, the system (8) for calculating the approximation $\hat{S}_{EB}$ will be ill-conditioned.

5 NUMERICAL EXPERIMENTS

We demonstrate the Multilevel MCDD-N preconditioner for a heterogeneous flow problem generated by a random Log-Normal permeability field, with standard deviation 1 and a correlation length of 3 fine-grid cells in both the x- and y-direction. All calculations are performed on a (50x50) uniform mesh, with a coarsening factor of 5. For the global boundary conditions we consider $u = 1$ at the left boundary, $u = -1$ on the right boundary and no-flow conditions on the top and bottom boundary. We consider a two-step 3-level coarsening strategy, consisting of upscaling from the fine level, and preconditioning the intermediate level. For upscaling, we construct two solution-based probing vectors belonging to the null space of S_{EB}. We solve one problem with unit pressure drop in the horizontal direction and no-flow conditions on the vertical boundaries. Similarly, we solve a second problem with unit pressure drop in the vertical direction and no-flow conditions on the horizontal boundaries. All the local solutions are solved on a region with an overlap of 1, 2 and 3 sub-domains. We refer to MCDD-N(n) as the preconditioner with n sub-domains overlap. Additionally, two oscillating probing vectors are used. As preconditioners for the intermediate level we consider the interface probing preconditioner of

Chan and Mathew[4], using 3 and 5 oscillating vectors. We denote them MCDD-3P, and MCDD-5P, respectively. Both strategies are compared with an MCDD preconditioner using reduced boundary conditions. This is the same interface approximation applied with e.g. the multiscale finite-volume method of Jenny[7] for problems on regular Cartesian fine grid. It is also equivalent with the tangential component approximation $\hat{S}_{EB} = C_{EB}^T$, (see e.g. Smith et al.[9]). We denote the preconditioner MCDD-TC.

	MCDD-TC	MCDD-3P	MCDD-N(1)	MCDD-N(2)	MCDD-N(3)
mean	$4.2 \cdot 10^{-1}$	$9.5 \cdot 10^{-2}$	$4.8 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$
mean (92%)	$1.5 \cdot 10^{-1}$	$9.2 \cdot 10^{-2}$	$3.1 \cdot 10^{-2}$	$9.9 \cdot 10^{-3}$	$4.5 \cdot 10^{-3}$
mean (80%)	$1.4 \cdot 10^{-1}$	$9.1 \cdot 10^{-2}$	$2.1 \cdot 10^{-2}$	$8.1 \cdot 10^{-3}$	$3.7 \cdot 10^{-3}$

Table 1: Upscaling; We analyse the error in L^2-norm after one fine-scale iteration. The results for each method are the mean of 50 realisations of random Log-normal permeability fields. We also show the truncated means, where 2 and 5 realisations of both the low and high end of the results are discarded.

	MCDD-TC	MCDD-3P	MCDD-N(1)	MCDD-N(2)	MCDD-N(3)
unprec.	78	71	74	74	74
MCDD-TC	31	30	28	28	35
MCDD-3P	23	22	22	22	22
MCDD-5P	17	16	17	17	17

Table 2: Preconditioning; The table shows the number of iterations on the intermediate level, to meet a tolerance of 10^{-8}. Here each column represents different upscaling procedures, while the rows represent different preconditioners. All results are means of 50 realisations of random Log-normal permeability fields.

6 DISCUSSION

The results in Table 1 show that considerably more accurate coarse spaces can be achieved by applying only a few solution-based probing vectors, capturing the most important features of the fine-scale solution. While the upscaling method resulting from the tangential component approximation (MCDD-TC), fails to capture the correct flow field for many problems involving heterogeneous permeability, the probing technique (MCDD-3P) represents a more robust framework for approximating the flow on the boundary. Harmonic probing vectors (MCDD-N) can be applied to give better approximation properties for the interface probing technique. As for standard upscaling methods, the overall accuracy of the solution-based vectors relies on the localisation assumptions for the local problems. In general, the overall accuracy will increase with the size of the overlapping

region; The results show an improved accuracy of about a factor 2.5, per sub-domain. For robustness of the preconditioner, we need an independent set of probing vectors. For large overlapping regions, the local solution within the target region is less influenced by the boundary conditions and we may get similar flow behaviour for different boundary set up. This may cause inaccurate approximations to the local Schur complement. The residual on the local boundaries can be used to build local error estimates and adaptive strategies for constructing accurate operators on the coarse scale or efficient smoothers for the fine-scale. Table 2 shows that algebraic preconditioners may be constructed and applied to coarser levels, independently of the choice of upscaling procedure. The oscillating probing vectors applied with the (MCDD-3P and MCDD-5P) seem to be efficient to capture the oscillating nature of the residual. However, a systematic investigation of the quality of the MCDD preconditioners is beyond the scope of this paper, and a more systematic study of the localisation approximation and the properties of $\hat{S}_{EB}$ is needed.

REFERENCES

[1] O. Axelsson. *Iterative solution methods*. Cambridge University Press, New York, NY, USA (1994).

[2] T.F.C. Chan and T.P. Mathew. The interface probing technique in domain decomposition. *SIAM J. Matrix Anal. Appl.*, **13**(1), 212-238 (1992).

[3] Y. Efendiev and T.Y Hou. *Multiscale Finite Element methods: Theory and Applications*. Springer Science+Business Media, New York, USA (2009).

[4] G.H. Golub and D. Mayers. The use of preconditioning over irregular regions. *Computing method in applied sciences and engineering*, VI (Versailles, 1983), pp. 3-14. North-Holland, Amsterdam (1984).

[5] I.G. Graham, P.O. Lechner and R. Scheichl. Domain decomposition for multiscale PDEs. *Numer. Math.*, **106**(4), 589-626 (2007).

[6] M.A. Hesse, B.T. Mallison and H.A. Tchelepi. Compact multiscale finite volume method for heterogeneous anisotropic elliptic equations. *Multiscale Model. Simul.*, **7**(2), 934-962 (2008).

[7] 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**(1), 47-67 (2003).

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

[9] B.F. Smith, P.E. Bjørstad and W.D. Gropp. *Domain decomposition*. Cambridge University Press, Cambridge (1996).

Paper C

Auxiliary Variables for 3D Multiscale Simulations in Heterogeneous Porous Media *

* Submitted to Journal of Computational Physics, November 2011.

Auxiliary variables for 3D multiscale simulations in heterogeneous porous media

Andreas Sandvin[a,b], Eirik Keilegavlen[a,*], Jan M. Nordbotten[a]

[a] *Department of Mathematics, University of Bergen,Johannes Brunsgate 12, 5008 Bergen*
[b] *Centre for Integrated Petroleum Research, University of Bergen, Allegaten 41, 5007 Bergen*

Abstract

The multiscale control-volume methods for solving problems involving flow in porous media have gained much interest during the last decade. Recasting these methods in an algebraic framework allows one to consider them as preconditioners for iterative solvers. Despite intense research on the 2D formulation, few results have been shown for 3D, where indeed the performance of multiscale methods deteriorates. The interpretation of multiscale methods as vertex based domain decomposition methods, which are non-scalable for 3D domain decomposition problems, allows us to understand this loss of performance.

We propose a generalized framework based on auxiliary variables on the coarse scale. These are enrichments the coarse scale, which can be selected to improve the interpolation onto the fine scale. Where the existing coarse scale basis functions are designed to capture local sub-scale heterogeneities, the auxiliary variables are aimed at better capturing non-local effects resulting from non-linear behavior of the pressure field. The auxiliary coarse nodes fits into the framework of mass-conservative domain-decomposition (MCDD) preconditioners, allowing us to construct, as special cases, both the traditional (vertex based) multiscale methods as well as their wire basket generalization.

Keywords: multiscale methods, control volume methods, porous media, preconditioning, domain decomposition

1. Introduction

Geological porous media are typically characterized as heterogeneous at virtually every scale. This reflects the process by which geological formations are created, where natural sedimentation processes spanning kilometers horizontally and millennia in time lead to composite materials that are intrinsically complex in structure. Compounding the difficulties introduced by multiscale geological parameterizations are the strongly non-linear equations that describe multi-phase flow in porous media. These equations lead to challenges that are manifested in discontinuous solutions as well as both gravitational and viscous instabilities. Such phenomena are frequently best understood as multiscale in nature. As a consequence of the complexity in modelling multi-flow in geological porous media, virtually every text-book on the subject address issues of scale. We refer to [1] for classical examples.

Two main avenues are typically followed when confronting multiscale phenomena. The most classical approach, multiscale modelling, is to manipulate equations defined at a finest, verified, scale, and attempt to derive effective equations valid on coarser scales. These equations are typically stated for derived variables. These derived variables broadly fall into three categories: Conserved (extensive) quantities, auxiliary (intensive) state variables, and problem specific variables. This final category of variables may be unique to the problem, or to the coarser scales, and can be interpreted to represent emerging properties of the

*Corresponding author
Email addresses: Andreas.Sandvin@uni.no (Andreas Sandvin), Eirik.Keilegavlen@uib.no (Eirik Keilegavlen), Jan.Nordbotten@math.uib.no (Jan M. Nordbotten)

system. In some cases these emerging properties are parameterizations of what would otherwise be seen as hysteretic, or non-unique, behavior. In the context of multi-phase flow in porous media, component masses are conserved at all scales, pressure is an intensive state variable at all scales, and finally various parameterizations of hysteresis or dynamical behavior are introduced to make the models appropriate in practice [2]. This classical approach has seen several formalizations in recent years, among the most instructive of which is the Heterogeneous Multiscale Method [3].

A more recent approach to handling multiscale characteristics is through adaption of the numerical methods themselves. Classically introduced as generalized finite elements by Babuska and Osborn [4], it was first made into a useful concept through the residual-free bubble methods [5], where multiscale features of the solution can be handled. Later, this concept was also applied to multiscale coefficients, in what is termed multiscale finite element and multiscale finite volume methods (see [6] for an introduction). While multiscale numerical methods have shown good properties on academic problems, they often fail to live up to their promise on real problems [7]. By exploiting the link between multiscale numerical methods and domain decomposition (DD), multiscale control volume methods can be framed in an iterative setting which greatly increases the potential for robust implementations [8]. However, an improved multiscale representation without iterations is still the ultimate goal.

In this paper, we propose to enhance the common understanding of multiscale numerical discretizations through an analogy to multiscale modelling. In particular, as multiscale control volume methods inherently discretize conserved quantities, it is natural to ask if the discrete approximation, like its modelling counterpart, can be enhanced through introducing problem-specific additional variables. We term these additional variables auxiliary, and the remainder of the paper is devoted to developing and verifying this concept. In particular we consider the issue of assigning boundary conditions to the local problems based the state in the coarse variables. This poses challenges for multiscale numerical methods, especially in the presence of long correlation pathways that renders non-local dependence of the solution. The problem is difficult already in two spatial dimensions, where the state in the coarse variables must be mapped onto a 1D boundary. Strategies proposed to remedy the situation include oversampling [9, 10], utilizing global information [10, 11] and using specialized boundary conditions [7, 12]. The situation becomes worse in three spatial dimensions, since a mapping to a 2D boundary is needed. In this paper, auxiliary coarse variables are used to address these challenges. By exploiting links between multiscale control volume methods and domain decomposition, the auxiliary variables can easily be introduced in the linear solver. We consider grids with relatively few primary coarse variables (corresponding to aggressive coarsening), and enhance the coarse space by auxiliary variables. Thus the number of internal boundaries decreases, while there is enough degrees of freedom in the coarse space to capture details in the solution. Our numerical experiments involves model problems as well as industrial benchmark data. The results show that auxiliary coarse variables can improve the performance of the linear solver considerably.

The rest of the paper is structured as follows: In Section 2, multiscale methods for three-dimensional problems are discussed and difficulties are pointed out. A multiscale linear solver is introduced in Section 3, and the extension to coarse spaces is introduced in Section 4. Simulation results are presented in Section 5, and the paper is concluded in Section 6.

2. Challenges of 3D multiscale elliptic problems

In this study we consider the following elliptic problem for flow in three dimensional porous media,

$$-\nabla\cdot(\mathbf{K}\nabla u) = q, \tag{1}$$

where $\mathbf{K}$ is the permeability of the medium, u is the potential and q represents the source terms of the system. The heterogeneous structure of porous rocks is reflected in the permeability $\mathbf{K}$, which can vary by several orders of magnitude on different scales. It is the variation of this parameter which represents the major challenge, and has been the main focus of the multiscale methods for problems involving flow in porous media. Hou and Wu [13] showed that the sub-scale information of the elliptic operator can be captured within a few coarse-scale basis functions, which increases the accuracy of the recovered fine-scale

solution. Several multiscale numerical methods have later been developed for the capturing of sub-scale information into local basis-functions. We refer to [6] for an overview of these methods.

A special focus of this paper will be on problems involving long correlation lengths of the parameter $\mathbf{K}$, e.g. fractures, faults and channels which occupies several coarse-scale grid blocks. The discretization of such problems are particularly difficult to upscale, and local iterations are required to guarantee accurate solutions [14]. Due to the difficulties involved, the primary focus of previous work has been the 2D problem. In the remainder of this section we will briefly discuss some of the existing challenges of multiscale numerical methods, and highlight some of the main challenges of extending these methods to 3D simulations.

2.1. Multiscale numerics

In general porous media there are rarely only two or a couple of distinct scales, but rather a continuum of physical scales which needs to be taken into account. However, for practical purposes we need to define a finest (geological) scale for the discretization of our problem. Usually the fine-scale discretization leads to a large coupled problem which is extensive to solve and often too computationally expensive to compute. For multiscale methods, one or a couple of coarse scales are added to speed up the calculation of a fine-scale conservative solution.

The solution space for each coarse scale is spanned by local coarse-scale basis functions. This is referred to as the coarse space. As for upscaling, the coarse scale equations for the multiscale method are stated for conserved variables, thus the multiscale methods can be categorized as upscaling methods. In addition, through the construction of coarse-scale basis functions, the multiscale methods can be applied as multilevel DD-preconditioners. In this paper we will consider the multiscale control volume methods, which can be written as DD-preconditioners in the framework of mass conservative domain decomposition (MCDD) [8].

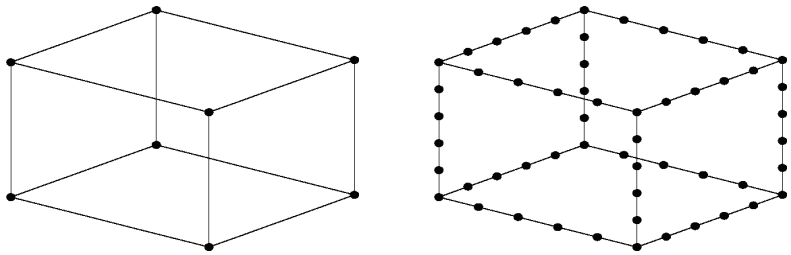
While the multiscale methods are designed to capture local heterogeneities within the sub-domains, one of the difficulties has been to include non-local and global information, which may influence the solution locally. An oversampling strategy was proposed by [9] to include non-local information into the calculation of the local basis-function, thus reducing the resonance effect of the intermediate scale heterogeneities. Adaptivity w.r.t the size and structure of these oversampling regions has further improved the accuracy and limited the number of additional computations [15]. The use of global information has also been considered [10, 11]. Finally, local iterations seams to be inevitable when constructing robust implementations of the multiscale methods.

2.2. Extension to 3D

Despite intense research on challenging problems in 2D, few results have been reported for problems in 3D. The capturing of sub-scale information naturally becomes more challenging for three dimensional flow. However, the extension from 2D to 3D problems also introduces new challenges w.r.t. the construction of the coarse-scale basis functions.

Concerning upscaling, the third dimension results in a larger gap between the numerical scales, where the number of fine cells per coarse cell grows as n^{dim}. Any aggressive coarsening strategy is obviously more challenging for 3D problems. The boundary conditions for the local elliptic problems are difficult to approximate. The localization approximations embedded in the multiscale simulations are usually lower-dimensional approximations of the flow on the sub-domain faces and edges and the errors due to these approximations become more severe for three dimensional flow.

From a domain decomposition perspective, it is well known that the extension of the coarse-scale basis functions from 2D to 3D problems is non-trivial. The multiscale methods are in general vertex-based (VB) methods (Fig. 1(a)), meaning that the coarse-scale degrees of freedom (dof) are associated with the vertices of the local sub-domains. For these methods, the standard piecewise linear interpolation is not robust, and do not result in scalable preconditioners for 3D-problems [16]. While the 2D coarse scale solution is mapped directly onto the boundary, the 3D coarse-scale solution, defined on vertices, needs to be mapped in two stages. The solution is first interpolated onto the edge dof, and secondly onto the face dof. The harmonic basis functions of the multiscale methods have shown to be more robust than the usual piecewise linear basis functions, w.r.t. problems involving sub-scale heterogeneities in 2D [17], however, this has not been



(a) (b)

Figure 1: Figure (a) and (b) show the vertex cells and wire basket cells, respectively.

verified for 3D problems. Moreover, in the homogeneous limit, the harmonic basis functions degenerates to the usual piecewise linear basis functions, which are not scalable with the size of the problem.

In domain decomposition, various extensions of the coarse space have been proposed to retain the approximation properties obtained in 2D simulations. Of special interest to this paper is the wire basket (WB) method [18], where both the vertex and edge dof belong to the coarse space, see Fig. 1(b). Thus, we only have one interpolation from the coarse-scale dof (wire basket) onto the boundary dof, which is similar to VB-methods for 2D-problems. Other extensions of the coarse space involves defining special interpolations on the sub-domain faces [16, 19]. Such methods will not be discussed in this paper. Instead we will show how the traditional VB-methods with standard interpolations can be improved through the use of auxiliary coarse variables.

While the WB-method is attractive w.r.t. convergence of the fine-scale solution, it uses a large static coarse space resulting in a much larger and more dense numerical scheme on the coarse scale. In many cases, it is not necessary to use all these dof related to the wire basket. Thus, by discarding the less important dof on the coarse scale, we may save many unnecessary computations.

In this paper, we develop a generalised framework for multiscale simulation by introducing *auxiliary variables* to the coarse space. The auxiliary variables are strategic sampling points of the coarse scale and they are meant as a supplement to the existing conserved variables on the vertices. Thus, it can be seen as a generalization of the existing vertex-based multiscale control volume methods. Furthermore, we observe that the WB-distribution of coarse variables becomes a special choice of the auxiliary coarse variables. The auxiliary variables fit naturally into the framework of MCDD, in which case they represent an additional flexibility with respect to multiscale simulation and multiscale numerical modelling. However, before we can introduce the auxiliary coarse variables, we need to formulate the domain decomposition framework.

3. MCDD

The discretization of Eq. (1) results in a sparse system of linear equations,

$$Au = b, \tag{2}$$

where the system matrix A is in general large. Hence, a direct calculation of Eq. (2) is typically very time-consuming. In reservoir simulation, several linear systems on this form must be solved within each time-step, in which case inexact solvers are always used. We seek a fast inexact solver for approximating the solution of the fine-scale problem (2).

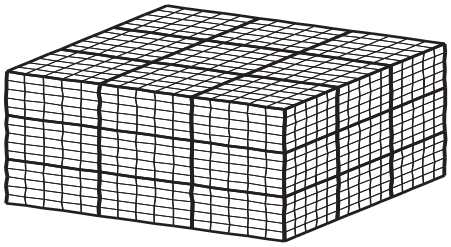


Figure 2: The figure shows the multiscale grid. Here, the bold faces indicate the primal coarse grid, constructed on top of an underlying fine-scale grid.

3.1. Mass conservative coarse-scale operator

Mass conservation is an essential property for reservoir simulation and fluid flow. In our case the equation is discretised into a set a fine-scale control-volumes ω_i, resulting in a linear system of equations which is mass conservative on the fine-scale. Each row of the linear system (2) will represent the mass balance equation over ω_i.

For the multiscale methods, the coarse model is defined on a primal coarse grid Ω, where each cell Ω_i is a collection of fine-scale control volumes and the coarse-scale interfaces coincide with interfaces on the fine-scale as seen in Fig. 3. If the fluxes on the coarse scale are mass conservative, it is possible to recalculate a mass conservative flux field on the fine-scale by a pre-processing of the fine-scale system (2) [20].

If all the equations corresponding to the fine-cells contained within a primal coarse cell Ω_i are added, the resulting equation will represent mass balance for Ω_i. This coarse scale equation is substituted for the row of the coarse cell (centermost cell of Ω_i). More precisely, we can write the pre-processed fine-scale system of equations as

$$Cu = p, \tag{3}$$

where

$$C = \sum_{\Omega_i} (R_i)^T M_i R_i A \quad \text{and} \quad p = \sum_{\Omega_i} (R_i)^T M_i R_i b.$$

Here, M_i is the integration matrix, while R_i represents the usual restriction matrix, consisting of zeros and ones, such that, acting on the global dof, R_i picks out the degrees of freedom corresponding to Ω_i. The integration matrix M_i is written as

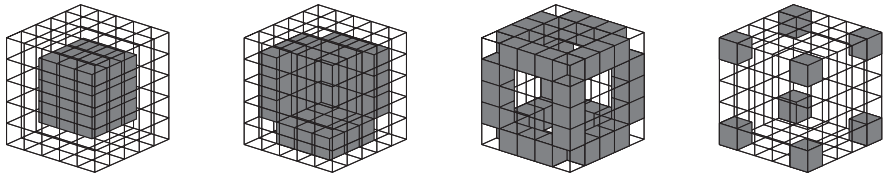
$$M_i = I + e_{iV} \left(\underline{1} - e_{iV}\right)^T, \tag{4}$$

where I is the identity matrix, e_{iV} is the unit vector identifying the row of the vertex cell and $\underline{1}$ is the vector entirely filled with ones.

3.2. Schur complement system

Several domain decomposition techniques coupled with various iterative schemes have been developed. For the majority of cases, the domain decomposition techniques are applied as preconditioners, where the iterations are accelerated by the use of Krylov subspace iterative methods. In this paper we will focus on the two-level additive Schwartz MCDD preconditioner, applied to Eq. (3). For information about other DD preconditioners, see e.g. [16, 21, 22]. The preconditioner will be accelerated using GMRES [23].

To formulate the MCDD preconditioner, we need to introduce the dual coarse grid Ω', consisting of dual coarse cells Ω'_i. The dual coarse grid is defined such that the centers of the the coarse cells Ω_i are located at the vertices of Ω'. Thus the coarse scale conservation of mass is associated with the vertices of Ω'. The



(a) Internal (b) Face (c) Edge (d) Vertex

Figure 3: An illustration of the internal, face, edge and vertex cells used in the domain decomposition method. For clarity of the visualization, only some of the face cells are shown.

fine-scale cells joining the vertex cells belong to the boundary of the dual-grid. The dual coarse-grid Ω' is a corner-cell grid and represents a non-overlapping decomposition of the global domain. For simplicity, we will refer to Ω'_i as a sub-domain. For each sub-domain, the fine-scale dof corresponding to internal cells are denoted as internal cells, while the fine-scale dof located on the sub-domain boundaries are denoted as boundary cells. The boundary cells are further sub-divided into face cells, edge cells and vertex cells, see Fig. 3. While the internal cells are localized on each sub-domain, the boundary cells are shared by neighboring sub-domains and are globally coupled.

By rearranging the system of equations into those dof corresponding to internal and boundary unknowns we can rewrite Eq. (3) as

$$\begin{bmatrix} C_{II} & C_{IB} \\ C_{BI} & C_{BB} \end{bmatrix} \begin{bmatrix} u_I \\ u_B \end{bmatrix} = \begin{bmatrix} p_I \\ p_B \end{bmatrix}. \tag{5}$$

The internal cells belonging to different sub-domains are now decoupled, and the block-diagonal matrix C_{II} can formally be inverted directly on each sub-domain Ω'_i. The Schur complement system is found by eliminating these internal cells, resulting in a system for the boundary cells only:

$$Su_B = g, \tag{6}$$

where

$$S = C_{BB} - C_{BI}C_{II}^{-1}C_{IB}, \tag{7}$$

and

$$g = p_B - C_{BI}C_{II}^{-1}p_I.$$

We can further split the Schur complement matrix into cells corresponding to the face (F), edge (E) and vertex (V):

$$S = \begin{bmatrix} S_{FF} & S_{FE} & S_{FV} \\ S_{EF} & S_{EE} & S_{EV} \\ S_{VF} & S_{VE} & S_{VV} \end{bmatrix}. \tag{8}$$

Written on this form, the Schur complement S still has a block diagonal structure, but all the blocks are now full. An explicit construction of S will in fact require the same amount of work as solving the entire fine-scale problem (5). Hence, the various Schur complement matrices must be approximated by sparse matrices, such that we can decompose the Schur problem. This gives us a MCDD or Schur complement preconditioner. A special requirement for the MCDD preconditioner is that the equation for the vertices (conserved variables on the coarse scale) must be solved exactly, providing a mass conservative coarse scale solution.

3.3. Interface approximations

The accuracy of the multiscale methods depends highly on the choice of boundary conditions for the local elliptic problems. A successful choice of boundary conditions may result in an accurate representation of the coarse scale problem and a good first approximation to the fine-scale solution. However, within an iterative setting, an accurate representation of the boundary conditions only affects the first iteration; after the first iteration, the residual does not represent any actual physics but should be regarded as noise. Thus, the best multiscale approximation does not necessarily give the most efficient preconditioner [7].

The most common interface approximation for the multiscale control volume methods is similar to the tangential component (TC) approximation for DD-preconditioners. The TC-approximation of the Schur matrix S (see Eq. (7)), can be written as

$$\hat{S} = C_{BB}^T, \tag{9}$$

where C_{BB}^T is denoted the tangential part of the C_{BB}. While C_{BB} is a diagonally dominant matrix, C_{BB}^T has reduced diagonal elements such that the row-sum equals to zero, i.e. $C_{BB}^T \cdot \underline{1} = 0$. The tangential part of C_{BB} arises in the multiscale methods as a lower dimensional discretization (in the "tangential" direction), also called the reduced boundary condition. This is in general only possible for regular Cartesian grids, however the TC-approximation is also valid for general grids [7]. We can write the TC-approximation for the Schur complement (8) on matrix form,

$$\hat{S} = \begin{bmatrix} C_{FF}^T & C_{FE} & 0 \\ 0 & C_{EE}^T & C_{EV} \\ S_{VF} & S_{VE} & S_{VV} \end{bmatrix}. \tag{10}$$

While S in general is a dense and globally coupled matrix, $\hat{S}$ is sparse and locally decoupled.

The coarse scale operator

For multiscale methods, the coarse cells represent integrated and conserved quantities on Ω. We will denote the coarse cells as C. Normally they are spanned out by the vertex cells, in which case we write $C = V$. By using Eq. (6) and (10) we can write up the equations for the face, edge and vertex cells:

$$\begin{aligned} C_{FF}^T u_F &= g_F - C_{FE} u_E, \\ C_{EE}^T u_E &= g_E - C_{EV} u_V, \\ A_V u_C &= b_V, \end{aligned} \tag{11}$$

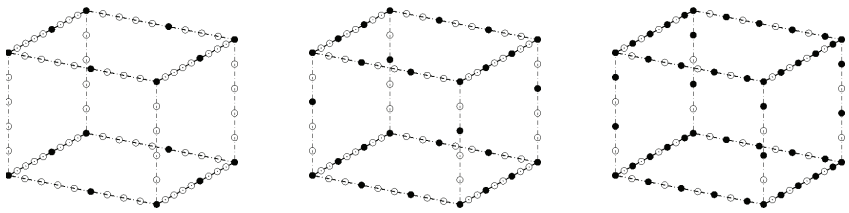
where

$$A_V = \begin{bmatrix} S_{VF} & S_{VE} & S_{VV} \end{bmatrix} \begin{bmatrix} \left(C_{FF}^T\right)^{-1} C_{FE} \left(C_{EE}^T\right)^{-1} C_{EV} \\ -\left(C_{EE}^T\right)^{-1} C_{EV} \\ I \end{bmatrix}, \tag{12}$$

is the coarse-scale operator and

$$b_V = g_V + \left(S_{VF} \left(C_{FF}^T\right)^{-1} C_{FE} - S_{VE} \right) g_E - S_{VF} g_F. \tag{13}$$

is the corresponding right-hand side vector on the coarse scale.



(a) (b) (c)

Figure 4: Figure (a)-(c) shows a coarse cell with three sets of auxiliary coarse cells.

4. Auxiliary coarse variables

Due to the more aggressive coarsening that necessarily follows from considering higher-dimensional problems, the multiscale methods become increasingly sensitive to the approximations introduced to the Schur complements. This is analogous to the situation that arises for traditional static condensation type methods, where in three spatial dimensions a discussion between vertex-based and wire basket methods arises. In this section, we introduce a framework for addressing this problems for multiscale preconditioners, in which the coarse space can be enriched by what we term auxiliary coarse variables.

4.1. Generalized preconditioner

Originally, the multiscale grid has been structured in a hierarchical way, where the coarse cells are fixed and represent coarse scale volumes or elements, on which mass conservation is satisfied. Indirectly, the coarse scale equation represents a special mass conservative discretization on a coarse scale grid.

The MCDD preconditioner satisfies the same hierarchical structure, where the vertex cells preserve the coarse-scale mass conservation, however, the MCDD framework is not restricted to those coarse scale degrees of freedom. An extension of the coarse-scale degrees of freedom has been studied in domain decomposition, where certain configurations of the coarse space can improve the interpolation of the coarse scale solution and give scalable and more robust preconditioners; see e.g. [24, 25] for more information.

This is specially important in 3D, where the usual vertex-based domain decomposition method, like most multiscale numerical methods, fail to be scalable for linear interpolations on the homogeneous problem. To address this shortcomming we generalize the multiscale iterative methods by introducing auxiliary variables on the coarse scale. We define the coarse cells as $C = \begin{bmatrix} X & V \end{bmatrix}$, where X represents the non-conserved auxiliary variables and V is the conserved variables on the vertices. Any fine-scale cell can in general be chosen as an auxiliary coarse cells. If $X = E$ we have the wire basket distribution of the coarse cells, which can be used to build preconditioners that are provably scalable [18]. On the other hand, if X is empty, we retain the VB-method.

The general Schur complement preconditioner can now be written

$$B = \hat{S}^{-1} = \left\{ \begin{bmatrix} C_{FF}^T & C_{FE} & C_{FC}|_{(C=E)} \\ 0 & C_{EE}^T & C_{EC} \\ S_{CF} & S_{CE} & S_{CC} \end{bmatrix} \right\}^{-1}, \tag{14}$$

where the approximate Schur complement matrix $\hat{S}$, is on the same form as (10). We observe that if X contains any edge cells, these will no longer be represented in the set E and connections to these cells will be stored in the third block column of (14). In the case of X being empty, the preconditioner (14) is similar to the multiscale control-volume preconditioner using vertex dof on the coarse scale (see Eq. (10)). In the special case of $X = E$, we can simplify the Schur complement preconditioner, writing

$$B_W = \left\{ \begin{bmatrix} C_{FF}^T & \begin{bmatrix} C_{FE} & 0 \end{bmatrix} \\ S_{WF} & S_{WW} \end{bmatrix} \right\}^{-1},$$

where $W = \begin{bmatrix} E & V \end{bmatrix}$ is the wire basket. The auxiliary coarse variables (ACVs) X, as applied in the framework of the Schur complement preconditioners, thus generalizes multiscale methods to include enriched coarse spaces. In general, adaptivity with respect to the selection of coarse cells is not new. In algebraic multigrid methods, the coarse scale dof are algebraically chosen within the iterative method. Several multilevel methods monitor the residual, and chooses the most inaccurate cells to be corrected for on a "coarse" scale. In this setting, the coarse scale is nothing more than a bounded set of the fine-scale. The idea of extending the coarse space are also considered in [26, 27] in the setting of multiscale finite element methods.

4.2. Properties of the auxiliary coarse variables

The auxiliary variables give us the flexibility of including additional fine-scale information, which might have large influence on the coarse scale. In principle, we now have the freedom of designing the coarse scale that we want, like in the case of numerical multiscale modelling. In this framework, the coarse cells represent distributed sampling points of the fine-scale problem, and the distribution of these points are important for the accuracy of the method.

We envision at least three strategies for the use of auxiliary coarse variables:

• *Parameter based.* The auxiliary coarse variables can be applied by means of refinement of the coarse space. In particular, this strategy should be considered in regions with strong contrasts in permeability.

• *Source based.* The inclusion of auxiliary coarse cells could also be motivated by the right hand side of the problem. In particular, auxiliary cells may be placed according to source terms of the system, normally represented by non-zero right-hand side elements. By placing an auxiliary coarse variable on the position of a Dirichlet well, one basis function will capture the fine-scale pressure in the vicinity of that well. This is analogous to constructing additional well-basis functions [28], which is normally performed for the MSFV method. The support for the additional basis function corresponding to the well is here limited to the corresponding sub-domain of the well.

• *Algebraic construction.* A third approach is to choose the auxiliary variables algebraically within an iterative procedure based on error estimates. The auxiliary variables are dynamic, in the sense that we can include or discard these variables at any stage of an iterative procedure. In any case, we only include or discard single basis-functions, which by construction are independent. This will introduce another adaptivity with respect to solving large linear systems effectively.

The placement of the additional coarse cells on the sub-domain boundaries is also analogous to multiscale methods with polygonal coarse elements. For each additional coarse cell on the sub-domain, we get one additional basis function. In the simplified case of linear basis functions, we observe that the extra auxiliary cells on the boundary will result in piecewise linear functions on the boundary. In the limit, as all the boundary cells are filled with auxiliary coarse variables, every cell is treated exactly and the true solution is captured. Even though this strategy is aimed at capturing non-linear and higher order functions of the solution, this is not similar to higher order multiscale methods, in which case we would need to incorporate higher order interpolations between the added coarse cells. In this paper we want to move the discussion away from only focusing on improved interpolations or boundary conditions. Here we apply the same boundary conditions everywhere, but in areas where the boundary conditions fail to capture the fine-scale physics, we show that enrichment of the coarse space may be applied directly or within an iterative procedure to improve the sub-scale capturing and convergence of the fine-scale solution.

Grid	Type	# PCV	# ACV	# (Coarse nodes)	# it
V1	Vertex	64	1	65	212
V2	Vertex	1728	1	1729	151
V3	Vertex	46656	1	46657	45
ACV1	ACV	64	385	449	170
ACV2	ACV	64	1537	1601	107
ACV3	ACV	1728	10369	12079	74
WB1	WB	64	4993	5057	65
WB2	WB	1728	41473	43201	42

Table 1: The number of primary (PCV) and auxiliary coarse variables for the log-normal grid. The last column shows the number of GMRES iterations needed to achieve a relative residual of 10^{-8}.

5. Numerical results

In this section we demonstrate the method of auxiliary variables through numerical experiments of increasing complexity. Our aim is to demonstrate the capability of auxiliary coarse variables to stabilize the multiscale methods for challenging problems in 3D. The simulations are conducted on parameters describing porous media exhibiting both short and long correlation lengths, as well as large jumps in the permeability coefficients. As for the geometry, only uniform Cartesian grids are considered. For all the numerical tests we apply auxiliary coarse variables restricted to either the wire basket or Dirichlet sources (fixed pressure wells).

When adding ACVs on an edge, the coarse space on the edge goes from representing linear (or reduced operators) functions for the vertex method to a piecewise linear (or piecewise reduced operators) functions. Thus ACVs can therefore be considered a refinement of the coarse space in the form of a piecewise linear polynomial and increasing number of degrees of freedom, bearing some similarity with traditional p-refinement for finite elements. Therefore we chose to compare ACV refinement of the coarse operator to classical h-refinement of a vertex method, which is realized by reducing the disparity between the fine and coarse scales. For simplicity, the additional ACVs are placed only on the subdomain edges. This keeps the refinement strategy relatively simple, and the wire basket scheme provides an upper limit of the possible improvement compared to the vertex scheme.

Some considerations regarding computational cost of the two refinement schemes are also in order. The main computational effort in the MCDD preconditioner is split into three parts: Solving local subproblems on edges, faces and the interior of subdomains, solving the coarse system, and creating restriction and prolongation operators (basis functions) between fine scale and coarse scale nodes. Of these, creating the mappings between coarse and fine scale will be more costly for ACV refinement than for vertex refinement, in that the support associated with a coarse scale node will be larger in the ACV refinement case. The impact of the two other sources of computational cost will be highly dependent on issues such as the degree of parallelization and available memory (for problems where many iterations are needed, it may be more effective e.g. to factorize the coarse scale matrix if one can afford to store the factorization). The final cost comparison between refinement of the coarse space and addition of auxiliary variables therefore comes down to implementation and hardware; we chose to use the total number of coarse variables as a proxy for computational cost.

5.1. Log-normal test case

Our first test case is a 3D grid with 108 cells in each direction, having in total 1.259.712 fine scale cells of unit size. The permeability has a log-normal distribution, as illustrated in Fig. 5. A Dirichlet injection well is placed in a corner, while a Neumann producer is located in the middle of the domain. For simplicity, periodic boundary conditions are applied.

The grid size facilitates vertex grids with a coarsening of 27, 9 and 3 cells in each direction, and we refer to these grids a V1, V2 and V3, respectively. Thus the coarse grids V2 and V3 can be seen as vertex refinements of grid V1. We will also refine grid V1 by placing 2 and 8 ACVs on each edge, rendering coarse

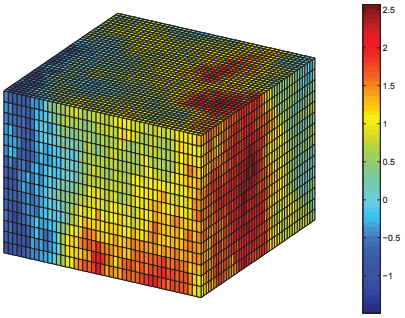


Figure 5: A log-normal permeability field. The base-10 logarithm of the permeability is plotted.

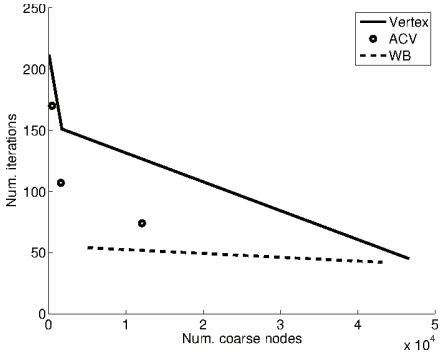


Figure 6: Number of iterations vs coarse unknowns for vertex and ACV grids for the log-normal permeability.

grids denoted ACV1 and ACV2, respectively. Similarly, coarse grid ACV3 is creating by adding 2 ACVs to each edge of grid V2. Finally, grid WB1 and WB2 are wire basket refinements of grid V1 and V2, respectively. A wire basket refinement of V3 was not considered, due to the large number of coarse nodes. The coarse grid configurations, together with the number of coarse unknowns and the number of GMRES iterations needed are summarized in Table 1.

Our first observation is that the vertex-based multiscale preconditioner is highly sensitive to the degree of coarsening, and indeed performs very poorly for the coarsest cases. This is consistent with our notion of low robustness for the vertex-based method in 3D. We observe that in this case, the numerical experiments indicate that the performance of the preconditioner can be improved by adding ACVs instead of vertex nodes. This is clearly seen by comparing grid V2 and ACV2: Both coarse grids are refinements of grid V1, with grid V2 having somewhat more coarse unknowns than grid ACV2. Nevertheless, the ACV strategy needs fewer iterations to reach the desired residual.

For convenience, Fig. 6 plots the iteration count versus the number of coarse unknowns. The figure illustrates that the ACV scheme is a middle course between the vertex and wire basket schemes, and that the preconditioner can be significantly improved compared to the vertex method by adding relatively few coarse variables.

5.2. SPE 10

As our second test case, we consider the permeability field from the tenth SPE comparative benchmark problem [29]. The grid consists of $60 \times 220 \times 85$ cells. Of these, the permeability in the upper 35 layers

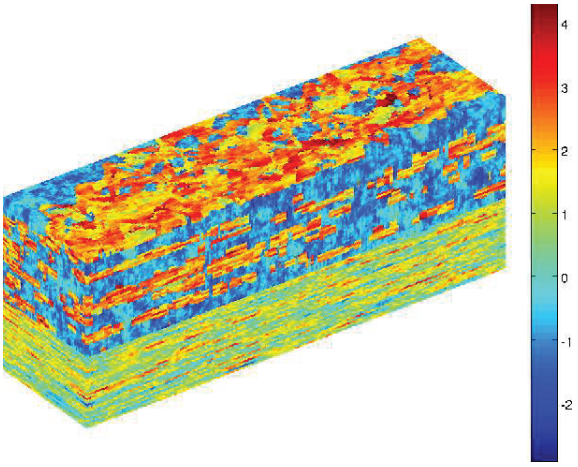


Figure 7: The base-10 logarithm of the SPE10 permeability field. The formation is viewed upside down to emphasize the channelized features in the lower part of the reservoir.

in the z-direction resembles a log-normal distribution, while the lower 50 layers are characterized by highly permeable channels and high permeability contrasts, see Fig. 7. Again, periodic boundary conditions are assumed in all tests. We note that to our knowledge, this represents the first systematic investigation of the multi-scale finite volume type methods applied as a preconditioner reported for problems of this size and complexity.

To investigate the impact of auxiliary coarse variables on this test case, we consider both the full model, as well as submodels from the upper and lower part of the formation. In all test cases, we will place an injection well in one corner, and a producer in the middle of the domain. A fixed pressure is applied in the injection well, and the injection cell is modeled as an ACV for all simulations.

5.2.1. Refinement of the coarse space

As a first test, we investigate how adding ACVs to a vertex grid impacts the number of iterations. We define three cases, the two first being subsets of the full model, corresponding respectively to a log-normal and channelized permeability field. The third case represents the full benchmark dataset:

Case A: A grid of $60 \times 220 \times 35$, permeability from SPE10 layers 1-35
Case B: A grid of $60 \times 220 \times 35$, permeability from SPE10 layers 36-70
Case C: A grid of $60 \times 220 \times 85$, permeability from SPE10 layers 1-85

On all these cases, we apply a coarse grid of $15 \times 11 \times 5$. We test vertex and wire basket preconditioners, as well as three preconditioners with an increasing number of ACVs denoted ACV 1, 2 and 3. The ACV refinement is the same for each edge, that is, permeability variations etc are not taken into account when defining the extra coarse nodes. The GMRES iterations are halted when the relative residual reaches 10^{-6}.

The number of coarse scale nodes and the results are summarized in Table 2. We see that the number of iterations needed to reach the desired tolerance decrease as auxiliary coarse variables are added. With the ACVs placed on the edges, the wire basket method is the limiting case with the densest coarse space, and thus a reduction factor of 2-3 compared to the vertex scheme is optimal for these tests. A large part of that improvement is achieved already when going to ACV 1. Also, the extra computational cost stemming from

Case A and B

	# ACV	# (Coarse nodes)	# it, Case A	# it, Case B
Vertex	1	561	193	1207
ACV 1	2341	2801	99	876
ACV 2	5041	5601	96	931
ACV 3	8401	8961	62	618
WB	14001	14561	63	600

Case C

	# ACV	#(Coarse nodes)	# it, Case C
Vertex	1	1361	1516
ACV 1	6801	8161	1082
ACV 2	13601	14961	1091
ACV 3	20401	21761	734
WB	34001	35361	780

Table 2: The number of GMRES iterations needed to reach a relative residual of 10^{-6} for the SPE10 test cases.

adding the extra coarse variables needed to go from ACV 3 to a full wire basket scheme does not seem to be justifiable.

The reduction in the iteration count is highest in the upper layers, where there are few abrupt changes in the permeability field, and thus the pressure solution has highest regularity. In the lower layers, the solution is erratic, and to best capture this behavior by ACVs, the auxiliary nodes should likely be placed on strategic locations in the reservoir, such as highly permeable channels. Despite that no adaptivity was applied to position the ACVs in the current test, the number of iterations needed is still reduced significantly, even when a relatively small number of ACVs are added.

5.2.2. Comparison of refinement strategies

It is worth comparing vertex and ACV refinement for the SPE 10 case, building on the analogy between h and p refinement. In this test, only cases A and B are considered, to emphasize the role of the different structure in the permeability fields.

We consider three vertex grids with an increasing number of coarse unknowns. Furthermore, we consider two ACV schemes, created by refining the coarsest vertex grid in such a way that the ACV grids and refined vertex grids have a similar number of coarse unknowns. Thus the simulations illustrates the performance of vertex and ACV refinement for a fixed number of coarse cells.

The results for both Case A and B are shown in Table 3, together with the number of coarse unknowns for the different schemes. For the log-normal-type permeability in Case A, vertex refinement renders a higher number of iterations needed compared to the strategy of adding auxiliary coarse variables. However, for the channelized Case B, the picture is less clear. This qualitative behavior is consistent with the notion of h-refinement for multiscale methods [6], where we note that for the case with high regularity (the upper layers) it is beneficial to go refine the coarse grid, while for the case of less regularity (the lower layers), the resonance effect between the characteristic length scales in the parameter field may influence the optimal grid spacing in a way where it is not always beneficial to reduce the spacing on the coarse grid.

In contrast, for both upper and lower layers, adding ACVs to the coarsest vertex grid renders fewer iterations. Compared to the vertex grids with a similar number of coarse unknowns, the ACV scheme for most cases has about half the number of iterations; in one case the reduction factor is almost 8. These results can be seen in the context of ACV providing a bridge between the vertex-based approximation and the relatively more robust wire basket approximation, leading to a consistently better approximation of the Schur complement systems.

Case A and B

Type	# PCV	# ACV	# (Coarse nodes)	# it, Case A	# it, Case B
Vertex	400	1	401	249	1481
Vertex	1680	1	1681	391	1984
Vertex	3698	1	3699	832	1732
ACV	400	1201	1601	171	1220
ACV	400	3201	3601	106	832

Table 3: The number of GMRES iterations needed to reach a relative residual of 10^{-6} for the SPE10 test cases. Three vertex preconditioners were tested, as well as two ACV schemes.

5.3. Standalone multiscale methods

As discussed in Section 3.3, the multiscale preconditioners derived herein can be seen as standalone multiscale approximations, which indeed gives the classical Multiscale Finite Volume Method [20]. As such, it is of interest to also discuss the performance of the multiscale preconditioner as an approximate solution by itself, not just its convergence properties.

While for 2D cross sections of the testcases discussed above, the multiscale methods can be made reasonably robust [7], the same is not true in 3D. In particular, for the regular Cartesian grids considered herein, the Multiscale preconditioner as a stand-alone solver gives unacceptable approximations even on the log-normal type permeability fields, and completely fails to give reasonable results for channelized problems such as the lower layers of SPE10. For this reason, we have chosen to emphasize the utility of multiscale methods in the iterative framework.

In practical implementations, the great advantage of the mass conservative types of preconditioners discussed herein is that converged solutions to the linear system of equations are not needed in order to have a locally conservative flow-field. Thus, in the iterative framework the tolerance of the linear solver can be chosen well above the tolerance used both when applying traditional preconditioners, but also above the tolerance used in our examples. This allows for considerable computational savings.

6. Conclusion

We have developed a new way of constructing coarse spaces for 3D multiscale simulations. While the existing coarse space for multiscale control volume methods based on vertex variables become unstable for three dimensional problems, we propose a generalised framework for including additional variables enriching the coarse space. We have denoted the additional coarse variables as auxiliary. The extended coarse space, based on auxiliary variables enable us to more accurately and more directly transfer fine-scale heterogeneous information onto the coarse scale. Moreover, it gives us the flexibility of constructing suitable coarse scale systems based on the complexity of the problem.

Numerical results show that the proposed framework can be used to construct efficient numerical methods for flow in 3D porous media; specially for problems involving long heterogeneous structures. In particular, the robustness of the multiscale framework in 3D is significantly enhanced with the novel formulation.

The wire basket multiscale method is a special case of the auxiliary coarse space, in which all edge cells are sampled on the coarse scale. This method degenerates to a robust preconditioner for problems involving homogeneous permeability on each sub-domain. While the WB-method usually gives the lowest number of iterations, much of the captured fine-scale information is often unnecessary. Numerical experiments show that much of the non-linear information may be captured within the local basis-functions by including a few auxiliary variables on the sub-domain boundaries. As such, the proposed methodology forms a flexibel and reasonable compromise between computational cost and iterative efficiency, which cannot be obtained by classical multiscale methods.

References

[1] D. Das, H. S.M. (Eds.), Upscaling multiphase flow in porous media: from pore to core and beyond, Springer, 2005.

[2] J. Bear, Dynamics of Fluids in Porous Media, Elsevier, 1972.
[3] W. E, B. Engquist, The heterogeneous multiscale methods, Commun. Math. Sci. 1 (1) (2003) 87–132.
[4] I. Babuska, G. Caloz, J. E. Osborn, Special finite-element methods for a class of 2nd-order elliptic problems with rough coefficients, SIAM J. Numer. Anal. 31 (4) (1994) 945–981.
[5] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method—a paradigm for computational mechanics, Comput. Methods Appl. Mech. Engrg. 166 (1-2) (1998) 3–24.
[6] Y. Efendiev, T. Y. Hou, Multiscale finite element methods, Vol. 4 of Surveys and Tutorials in the Applied Mathematical Sciences, Springer, New York, 2009, theory and applications.
[7] A. Sandvin, J. M. Nordbotten, I. Aavatsmark, Multiscale mass conservative domain decomposition preconditioners for elliptic problems on irregular grids, Comput. Geosci. 15 (3) (2011) 587–602.
[8] J. M. Nordbotten, P. E. Bjørstad, On the relationship between the multiscale finite-volume method and domain decomposition preconditioners, Comput. Geosci. 12 (3) (2008) 367–376.
[9] T. Y. Hou, X.-H. Wu, Z. Cai, Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients, Math. Comp. 68 (227) (1999) 913–943.
[10] Y. Efendiev, V. Ginting, T. Hou, R. Ewing, Accurate multiscale finite element methods for two-phase flow simulations, J. Comput. Phys. 220 (1) (2006) 155–174.
[11] L. Durlofsky, Y. Efendiev, V. Ginting, An adaptive local-global multiscale finite volume element method for two-phase flow simulations, Advances in Water Resources 30 (3) (2007) 576 – 588.
[12] I. Lunati, P. Jenny, Treating highly anisotropic subsurface flow with the multiscale finite-volume method, Multiscale Model. Simul. 6 (1) (2007) 308–318 (electronic).
[13] T. Y. Hou, X.-H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, J. Comput. Phys. 134 (1) (1997) 169–189.
[14] H. Hajibeygi, G. Bonfigli, M. A. Hesse, P. Jenny, Iterative multiscale finite-volume method, J. Comput. Phys. 227 (19) (2008) 8604–8621.
[15] J. M. Nordbotten, Adaptive variational multiscale methods for multiphase flow in porous media, Multiscale Model. Simul. 7 (3) (2008) 1455–1473.
[16] A. Toselli, O. Widlund, Domain decomposition methods—algorithms and theory, Vol. 34 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 2005.
[17] I. G. Graham, P. O. Lechner, R. Scheichl, Domain decomposition for multiscale PDEs, Numer. Math. 106 (4) (2007) 589–626.
[18] B. F. Smith, A domain decomposition algorithm for elliptic problems in three dimensions, Numer. Math. 60 (1) (1991) 219–234.
[19] C. Farhat, F.-X. Roux, A method of finite element tearing and interconnecting and its parallel solution algorithm, Internat. J. Numer. Methods Engrg. 32 (1991) 1205–1227.
[20] P. Jenny, S. H. Lee, H. A. Tchelep, Multi-scale finite-volume method for elliptic problems in subsurface flow simulation, J. Comput. Phys. 187 (1) (2003) 47–67.
[21] A. Quarteroni, A. Valli, Domain decomposition methods for partial differential equations, Numerical Mathematics and Scientific Computation, The Clarendon Press Oxford University Press, New York, 1999, oxford Science Publications.
[22] B. F. Smith, P. E. Bjørstad, W. D. Gropp, Domain decomposition, Cambridge University Press, Cambridge, 1996.
[23] Y. Saad, M. H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (3) (1986) 856–869.
[24] J. Mandel, B. Sousedík, Coarse spaces over the ages, in: Domain Decomposition Methods in Science and Engineering XIX, Vol. 70, Springer-Verlag, 2009, lecture Notes in Computational Science and Engineering.
[25] O. Widlund, The development of coarse spaces for domain decomposition algorithms, in: Domain Decomposition Methods in Science and Engineering XVIII, Vol. 70, Springer-Verlag, 2009, lecture Notes in Computational Science and Engineering.
[26] J. Galvis, Y. Efendiev, Domain decomposition preconditioners for multiscale flows in high-contrast media, Multiscale Model Simul 8 (4) (2010) 1461–1483.
[27] J. Galvis, Y. Efendiev, Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces, Multiscale Model Simul 8 (2010) 1621–1344.
[28] C. Wolfsteiner, S. H. Lee, H. A. Tchelepi, Well modeling in the multiscale finite volume method for subsurface flow simulation, Multiscale Model. Simul. 5 (3) (2006) 900–917 (electronic).
[29] M. Christie, M. Blunt, Tenth spe comparative solution project: A comparison of upscaling techniques, SPEREE 4 (4) (2001) 308–317.

Part III

Supporting Material

Report 1

Mass Conservative Domain Decomposition for Porous Media Flow *

* Accepted for publication as a chapter in the book Finite Element Method, InTech, November 2011.

CHAPTER NUMBER ?

Mass conservative domain decomposition for porous media flow

Jan M. Nordbotten, Eirik Keilegavlen, Andreas Sandvin
University of Bergen
Norway

1. Introduction

Understanding flow in subsurface porous media is of great importance for society due to applications such as energy extraction and waste disposal. The governing equations for subsurface flow are a set of non-linear partial differential equations of mixed elliptic-hyperbolic type, and the parameter fields are highly heterogeneous with characteristic features on a continuum of length scales. This calls for robust discretization methods that balance the challenges in designing efficient and accurate methods. In this chapter we focus on a class of linear solvers for elliptic systems that aims at providing fast approximate solutions, preferably in one iteration, but fall back to being iterative methods with good convergence properties if higher accuracy is needed.

We consider flow of flow of a single fluid in a porous media, transporting a passive particle. This can for instance represent flow of a pollutant as a result of groundwater contamination. Governing equations for the flow will be presented in the next section. Analytical solutions to the flow problem can only be found in very special cases, and in general, numerical approximations must be sought. The primary numerical schemes for commercial simulations are control volume methods. These methods are formulated such that conservation of mass is ensured, which is considered crucial in applications. After discretizing, an elliptic equation needs be solved for the pressure. This process is computationally expensive and may constitute the majority of the simulation time.

The permeability (fluid conductivity of the rock) in subsurface porous media has a truly multiscale nature, with highly permeable pathways with significant correlation lengths. Hence the elliptic pressure equation will experience strong non-local effects, posing a challenge for linear solvers. Moreover, the permeability field constructed by geologists is highly detailed; the number of cells in the geo-model can easily be several orders of magnitude higher than what is feasible to handle in a flow simulation. The traditional approach to this problem has been to upscale the permeability, e.g. to compute a representative permeability on a coarser grid. For the pressure equation this gives a linear system that is much smaller and computationally cheaper to solve. The drawback is of course that details in the geological characterization may be lost during upscaling, and these details are known to have significant impact on transport. An alternative approach is offered by the so-called multiscale methods, which have been a highly popular research field in the last decade (Tchelepi & Juanes, 2008). Like upscaling, multiscale methods perform a coarsening to end up with a relatively small linear system to solve. However, a multiscale method also provides a mapping from the coarse solution onto the fine grid. This projected solution will not be equal to a direct solution on the fine grid, but the two solutions will share many properties; in particular, many multiscale methods provide a velocity field that is mass conservative on the fine scale. Hence it can be used to solve fine scale transport equations. Numerical experiments have shown that this strategy can be extremely effective and highly accurate when measured in metrics that are important for applications (Kippe, et al., 2008; Efendiev & Hou, 2009).

Despite the success of multiscale methods in porous media flow, the strategy has certain weaknesses. In this chapter we highlight the quality of the coarse operator: If this does not represent essential features of the flow field, the quality of the fine scale velocity field may be poor. In particular, long and high permeable pathways are difficult to capture in the coarse scale operator. A natural approach would therefore be to introduce a scheme that allows for iterations on the multiscale solution. The idea of a multi-level iterative method resembles domain decomposition, and in Nordbotten & Bjørstad, 2008, the equivalence between the multiscale finite volume method (Jenny, et al., 2003) and a special domain decomposition strategy was shown. The resulting iterative scheme was termed mass conservative domain decomposition (MCDD), and it can be classified as an additive Schwarz preconditioner with minimal overlap. Contrary to classical domain decomposition

methods, MCDD will produce solutions that are mass conservative at any iteration step, thus it is not necessary to reduce the pressure residual to a very low value before solving transport equations. Various aspects of MCDD have been tested for two-dimensional problems (Kippe, et al., 2008; Sandvin, et al., 2011; Lunati, et al., 2011). However, to formulate multiscale methods for three-dimensional problems has turned out to be considerably more difficult in general, and to our knowledge, no applications of MCDD-type methods within an iterative setting have been reported in three dimensions.

In this chapter, we consider multiscale methods and preconditioners defined for arbitrary number of spatial dimensions. We show how the multiscale method can be formulated both as a top-down and as a bottom-up method, and that these formulations give rise to different interpretations of the resulting approximations and preconditioners. Numerical examples illustrate the main strengths and weaknesses of the approach. Moreover, the numerical examples highlight the capabilities of the framework in terms of producing quick calculations when possible, but also producing more accurate results when needed.

2. Governing equations and discretization

The primary focus of the current chapter is linear solvers. The particular linear solvers we discuss are designed to preserve certain properties from the physical problem. Therefore, the linear solvers cannot be discussed without first specifying both the governing equations and the particular discretizations we are concerned with.

2.1 Governing equations

We consider flow of an incompressible fluid in a porous medium. For an introduction to flows in porous media, see e.g. (Bear, 1972); a reference focusing on appropriate numerical methods for this problem, confer (Chen, et al., 2006). Here we will only provide a brief review of the main ideas of importance to this chapter. Conservation of mass (volume) for each phase can be modeled by the equation:

$$\nabla \cdot \boldsymbol{u} = v.$$

Here the flux of each phase is represented by $\boldsymbol{u}$, and v denotes the volumetric source / sink terms. The flux is usually assumed to be given by Darcy's law, which reads

$$\boldsymbol{u} = -\boldsymbol{k}\nabla p,$$

where the permeability is denoted $\boldsymbol{k}$, and p is the fluid potential. Additionally, we consider a dissolved concentration c which is passively advected with the velocity field,

$$\phi \frac{\partial c}{\partial t} + \nabla \cdot (c\boldsymbol{u}) = \psi, \tag{1}$$

where ϕ is the fraction of the void space available for fluid flow, referred to as porosity, and the material source term is given by ψ. We note that by introducing the particle velocity $\boldsymbol{v} = \phi^{-1}\boldsymbol{u}$ the advection of the dissolved concentration can be written in terms of the material derivative on Lagrangian form

$$\frac{Dc}{Dt} = \frac{\psi}{\phi} - c\frac{\nabla \cdot \boldsymbol{u}}{\phi}.$$

By eliminating the fluid flux from the statement of volume conservation we obtain an elliptic equation for pressure

$$-\nabla \cdot (\boldsymbol{k}\nabla p) = v. \tag{2}$$

In the sequel, we will study numerical solution techniques for Eq. (2), while keeping in mind that the methods must also be applicable for complex situations. Specifically, by integrating the Lagrangian version of the transport equation, we see that volume balance errors lead to exponential growth of errors in the dissolved concentration. Thus, it is of importance for the problems we consider that the velocity field must always be mass conservative in order to be suitable for use with most transport schemes.

2.2 Discretization

To discretize Eq. (1), we consider scalar discretizations, and in particular control volume schemes as they are particularly well suited for an exact representation of the conservation equation. Introducing the usual L_2 inner product, we can write the elliptic equation on a weak form as: Find $p \in H_0^1$ such that

$$(\boldsymbol{k}\nabla p, \nabla w) = (v, w) \quad \text{for all } w \in W. \tag{3}$$

Here, W represents a suitably chosen space, and we have for simplicity assumed zero Dirichlet boundary conditions to simplify the exposition. From this equation, we obtain control volume methods by choosing the finite subset of $W_h \in W$ to be the piece-wise constants forming a partition of unity on a cell-based grid, from which we obtain

$$\int_{\partial\omega} \boldsymbol{u} \cdot \boldsymbol{n} \; d\sigma = \int_{\omega} v \, d\tau,$$

for each (primal) cell ω. Note that $\boldsymbol{n}$ is the unit normal vector pointing out from the cell, and the product $\boldsymbol{u} \cdot \boldsymbol{n}$ is the normal flux over the boundary. Various control volume methods can now be defined by their approximation to the (flux) boundary integrals, most of which can be interpreted as particular choices of the finite space for p_h. We will in the following assume that such a choice has been made (for concreteness, one may consider the control-volume finite element method which is defined by p_h lying in the space spanned by piece-wise linears with nodes forming a dual grid to the partition induced by W_h). Furthermore, we assume for simplicity that the choice of flux approximation leads to a local approximation of the flux, in the sense that fluxes can be explicitly represented as a combination of fluid potentials in near-by cells.

We have now described a general setting for discrete representations of volume balance and Darcy's law which lead to a sparse linear system for the scalar variable p_h, which can be given on vector-matrix form as

$$\boldsymbol{A}\boldsymbol{p} = \boldsymbol{v}. \tag{4}$$

Remark 1: The control-volume finite element method, while attractive for educational purposes, is not very accurate in practice. Therefore, in reservoir simulation, the flux over a face has traditionally been approximated as driven by the pressure difference in the two adjacent cells only, giving rice to two-point schemes for the flux (Aziz & Settari, 1979). For logically Cartesian grids, this gives a classical 5- or 7-point cell stencil in 2 and 3 dimensions, respectively. However, in situations when the principal axis of the permeability tensor deviates considerably from the orientation of the grid, two-point schemes are known to produce inaccurate results. As a remedy, so-called multi-point schemes have been introduced (Edwards & Rodgers, 1998; Aavatsmark, 2002). These produce more accurate results to the price of a larger computational stencil, for Cartesian grids, the resulting linear system will have 9 and 27 bands in 2 and 3 dimensions, respectively. As we will see, the reduced accuracy of two-point schemes for rough grids is not only important for discretizing Eq. (2); similar considerations are also important when constructing fast linear solvers.

Remark 2: If a method is defined by choosing both p_h and w_h to lie in the same finite-dimensional space, the classical finite element method is recovered. In particular, the simplest choice, the piece-wise (multi)-linear functions give a system of equations that have a similar algebraic structure to the control-volume methods discussed above, but do not explicitly represent conservation.

3. Mass conservative domain decomposition

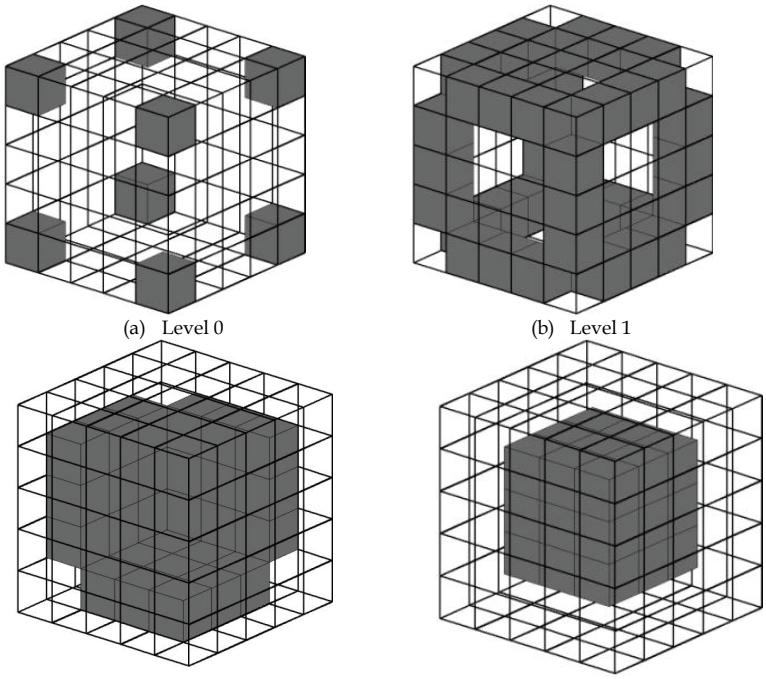
Here we describe the ingredients for the mass conservative domain decomposition (MCDD). We will start our presentation with describing the development of Schur-complement systems for N-dimensional problems. Readers may of course chose to disregard this generality, and consider only the special cases $N = 2, 3$. From this, we will see how we can form classical domain decomposition methods as well as multiscale control-volume methods. Note that due to the large number of matrices and vectors involved in the presentation, these will no longer be marked as bold as long as there is no room for confusion.

3.1 Schur complement systems

Being consistent with the discretization outlined above, we assume that computational domain is partitioned into a fine scale grid, and that control volume discretization enforces conservation of mass on the fine grid cells. In order to proceed in the construction of a two-level method, we need to introduce the notion of coarse grids.

Consider a continuous collection of cells, referred to as internal boundary cells, which partition the domain into isolated subdomains. By isolated, we mean in the sense of the discretization of the elliptic operator, such that no cell in one subdomain is dependent on any cells of any other subdomains. We interpret the subdomains as cells of the coarse dual grid, and the internal boundary cells thus form the nodes, edges, faces etc. of the dual coarse grid. We will identify cells and variables with a numerical subscript dependent on what part of the dual coarse grid they form part of: 0 indicates dual

coarse nodes, 1 indicates dual coarse edges, 2 indicates faces, and so on, until N denotes cells that lie in the subdomains (where N is the dimension of the problem). Confer Fig. 1 for an illustration. When considering only the internal boundary cells, we will refer with subscript B to all subscripts less than N. Note that we have not yet introduced a coarse primal grid; this will not be needed before a later section.



(a) Level 0 (b) Level 1

(c) Level 2 (d) Level 3

Fig. 1: Illustration of cells on different levels in a three-dimensional Cartesian grid. For clarity of visualization, only some of the cells on level 2 are indicated.

We now start to manipulate the linear system of equations (4), with the ultimate goal of obtaining a coarse linear system that captures non-local structures. By a reordering of the unknowns based on the dual coarse grid, Eq. (4) can then be written as

$$\begin{pmatrix} A_{00} & \dots & A_{0N} \\ \vdots & \ddots & \vdots \\ A_{N0} & \dots & A_{NN} \end{pmatrix} \begin{pmatrix} p_0 \\ \vdots \\ p_N \end{pmatrix} = \begin{pmatrix} v_0 \\ \vdots \\ v_N \end{pmatrix}. \tag{5}$$

In the last row, by construction, A_{NN} is a sparse block diagonal matrix, with each block representing the interactions within each isolated subdomain. This implies that we can find the values p_N by a local calculation given the variables on the internal boundary cells. We write these local calculations as

$$p_N = A_{NN}^{-1}(v_N - A_{NB}p_B).$$

We use this expression to formally eliminate internal cells from our system of equations. Thus, by substitution into Eq. (5) we have the *Schur Complement* system

$$\begin{pmatrix} S_{00} & \dots & S_{0(N-1)} \\ \vdots & \ddots & \vdots \\ S_{(N-1)0} & \dots & S_{(N-1)(N-1)} \end{pmatrix} \begin{pmatrix} p_0 \\ \vdots \\ p_{(N-1)} \end{pmatrix} = \begin{pmatrix} \tilde{v}_0 \\ \vdots \\ \tilde{v}_{(N-1)} \end{pmatrix},$$

where the Schur complement matrices S_{ij} are defined as

$$S_{ij} \equiv A_{ij} - A_{iN}A_{NN}^{-1}A_{Nj},$$

and the right hand side has been updated to reflect the elimination of the internal nodes by

$$\tilde{v}_i \equiv v_i - A_{iN}A_{NN}^{-1}v_N.$$

We make a few comments about the Schur complement system.

Remark 3: By the Schur complement formulaiton the number of unknowns has been reduced, from what was essentially an N dimensional problem to an $(N-1)$ dimensional problem. This significant reduction in model complexity comes at the cost of the Schur complement system being in general much denser than the original system. Furthermore, the computational cost of calculating the full Schur complement matrices if frequently prohibitive. As such, the Schur complement formulation by itself is seldom used.

Remark 4: For local discretizations the direct coupling between variables p_i and p_j, where i and j are more than one integer apart, is usually small (and indeed there is no coupling for the two-point flux approximation methods). This implies that the matrices A_{ij} and thus also S_{ij} are for many practical problems essentially zero for $|i-j| \geq 2$, and the full Schur complement system is therefore essentially block tri-diagonal. Furthermore, we see that S_{ij} only differs significantly from A_{ij} in the case where $i = j = N-1$.

In the particular case of two spatial dimensions, the matrix S describes interaction between edge and vertex nodes only. In the case of three dimensions, it describes the interaction between faces, edges and vertexes, where we expect that the interactions between vertexes and faces are weak.

While the Schur complement system itself may be prohibitive to form and solve, it provides the framework for developing approximate solvers. Classically, these fall in the category of domain decomposition preconditioners (Smith, et al., 1996; Quateroni & Valli, 1999; Toselli & Widlund, 2005). In this chapter, we see how this framework also gives us both multiscale methods and preconditioners based on them.

Recall that the Schur complement system is (essentially) tridiagonal. The main approximation strategies to this system fall in two categories: The top down strategy gives a low-rank approximation to S based on only the degrees of freedom associated with vertexes of the coarse dual grid, which are then identified as the coarse degrees of freedom. This essentially forms a multiscale subspace based on the lower-diagonal component of S, and is the approach we will emphasize in the following. The bottom-up strategy goes the other way, successively applying Schur complement strategies to eliminate all variables until only a system for p_0 remains. Since the Schur complement matrices themselves are too expensive to calculate, the bottom-up approach requires introducing low-rank approximations to the Schur complement (e.g. probing based techniques (Chan & Mathew, 1992)) at every stage in the succession. The class of domain decomposition methods known as substructuring methods is often formulated in terms of the bottom-up framework.

3.2 Multiscale basis approximations

The multiscale basis approximations to the Schur complement system use the (block) lower diagonal component of S. Retaining the dependence on p_0, which we hereafter identify as the coarse variable, we then see that we obtain an explicit expression for the remaining degrees of freedom. In the block tri-diagonal case, this can be written compactly as:

$$p_i = \left(\prod_{j=1}^{i} -\hat{S}_{jj}^{-1}\hat{S}_{j(j-1)}\right)p_0 + \sum_{k=1}^{i}\left(\prod_{j=k+1}^{i} -\hat{S}_{jj}^{-1}\hat{S}_{j(j-1)}\right)\tilde{v}_k.$$

In this expression, the matrix products are ordered right to left, and we have marked the Schur complement matrices with a hat, indicating that approximate choices of these matrices can be used in order to define different multiscale bases. In the general case, where a block tri-diagonal system is not assumed, the above expression is defined recursively. Either way, for conciseness, we denote the linear operator associated with the reconstruction of the full approximation p by its homogeneous and heterogeneous parts,

$$p = \Psi p_0 + \Upsilon\tilde{v}.$$

At this point we make the following remarks.

Remark 5: The space spanned by the projection of p_0 to the full set of variables defined by Ψ is termed the *multiscale space* W_H^{MS}. It can be characterized by the basis functions obtained by setting $p_0 = e_i$, where e_i is the elementary vectors. The resulting product allows us to define the *multiscale basis function* ψ_i^{MS} as columns of Ψ,

$$\psi_i^{MS} \equiv \Psi e_i.$$

Given suitable choices of $\hat{S}_{ij}$, this gives various multiscale basis functions from literature, as seen in the following remarks. In the terminology of domain decomposition, these basis functions are often referred to as prolongation operators.

Remark 6: The natural interpretation of $\hat{S}_{jj}^{-1}\hat{S}_{j(j-1)}$ is to solve local problems at the level j using level $j-1$ as boundary conditions. This motivates the usual approximations to these Schur complements. Three important alternatives exist.

1. $\hat{S}_{jj}$ can be chosen as a discretization of the original differential operator restricted to the part of the internal boundary associated with j. This is the original multiscale basis functions of Hou & Wu 1997, and this is also the strategy we will apply in our numerical experiments.
2. For arbitrary operators, the differential operator restricted to a lower dimension may not be a good approximation to the problem, and this approximation is unstable. For such cases, a simple linear interpolation on internal boundaries can be suggested (Lunati & Jenny, 2007), and $\hat{S}_{jj}$ is then chosen as any matrix which admits the relevant (multi-)linear solutions.
3. Both the preceding operators require knowledge about the original geometry of the problem, and can thus be seen as geometric methods. If it is desired to implement multiscale methods strictly algebraically, then it is possible to construct algebraic approximations $\hat{S}_{jj}$ based on the information in S_{jj}, as was explored in Sandvin, et al., 2011.

Remark 7: It is common to not approximate the last Schur complement S_{NN}. Note that this does not imply that this Schur complement matrix needs to be computed, as we only need to know its action on the elements of the multiscale basis and on the right hand side. If this component is retained exactly, then the method becomes residual-free on the subdomains, which is an important aspect that can be exploited at later stages.

Keeping in mind that we now have an explicit representation of the solution covering the domain given the knowledge of the coarse nodes, we can use this representation to obtain a coarse system of equations.

3.3 Coarse scale equations retaining conservation form

From the last section, we see that we can use the Schur complement system to obtain a multi-scale basis. This is essentially a low-dimensional approximation of the solution space for the homogeneous part of the discrete differential operator. What remains in order to get an approximate solution is to consider coarse equations that constrain the remaining degrees of freedom in the multiscale space.

The original system of equations provides us with the first option for a set of coarse equations, since the equations associated with the coarse variables are simply our fine-scale discretization. Recalling the notation Ψ that indicates the linear operator that reconstructs the homogeneous part of the solution from the coarse basis we see that our original system of equations is simply

$$\begin{pmatrix} A_{00} & \cdots & A_{0N} \end{pmatrix} \Psi\, p_0 = v_0 - \begin{pmatrix} A_{00} & \cdots & A_{0N} \end{pmatrix} \Upsilon \tilde{v}.$$

This is however a poor choice of constraint for our coarse variables, as it physically represents only the differential operators locally around the coarse node. From the perspective of the variational derivation of the control volume method, this solution thus satisfies Eq. (3) with $p \in W_H^{MS}$ and w in the space of piecewise constants with support around the *local cells associated with* p_0. From this understanding, we are motivated to think of reformulating the system such that the test functions for the coarse equations have a larger support, and in particular also form a partition of unity.

To be precise, consider a coarse partition of the domain, referred to as the primal coarse grid, which has the following properties: Each cell in the primal coarse grid consists of a set of cells from the fine grid, and contains exactly one vertex (cell on level 0) of the dual coarse grid, see Fig. 2. Since the primal coarse grid is a subset of the fine-scale grid, we know that the

space of piecewise constant functions W_H on the primal coarse grid is a sub-space of the space of piecewise constant functions W_h on the fine grid. Therefore, by a change of representation, we could write the original discretization such that the discrete equations for the coarse variables satisfied Eq. (3) for all piecewise constant functions of the primal coarse grid. This leads to our desired coarse equations.

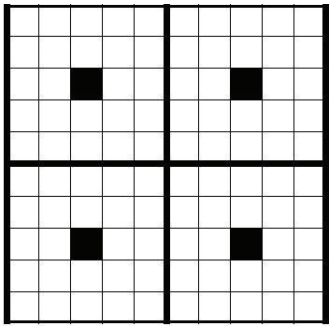


Fig. 2: A two-dimensional fine scale grid with a primal coarse grid imposed on it (bold lines). Black cells denotes center cells in the primal cell, these are on level 0 in the dual topology.

More practically, let the A, as before, represent a standard control volume discretization. Then let R_i be the restriction matrix to primal coarse cell i, and let $A_i = R_i A$. If furthermore M is an integration matrix that sums all rows in A_i into the row of the center cell, that is

$$M_i = I + e_{i0}(\mathbf{1} - e_{i0})^T, \tag{6}$$

where I is an identity matrix, e_{i0} is a unit vector identifying the center of the coarse cell, and $\mathbf{1}$ is a vector of ones. Multiplication with M_i for all primal coarse cells, and mapping the result back to the whole domain gives a linear system

$$Cp = v, \tag{7}$$

where $C = \sum_i R_i^T M_i R_i A$, and $v \longrightarrow \sum_i R_i^T M_i R_i v$. The linear system (7) is the original conservation of mass on the fine scale for all variables p_i where $i \geq 1$, however it represents conservation on the coarse scale for variables p_0. Note in particular that this means that $C_{ij} = A_{ij}$ for $i \geq 1$, and that this linear transformation does not change the solution p.

We now see that the coarse equations, as given by

$$(C_{00} \quad \cdots \quad C_{0N})\Psi\, p_0 = v_0 - (C_{00} \quad \cdots \quad C_{0N})\Upsilon\tilde{v},$$

solve the problem given by Eq. (3) for with $p \in \text{span}\, \psi^{MS}$ and $w \in W_H$, the space of piecewise constant functions on the dual coarse grid. We have thus derived a coarse control volume discretization, utilizing exactly a multiscale basis function to represent the solution. As a direct method, this is the so-called Multiscale Control Volume (Finite Element) Method as was first discussed (assuming $v_i = 0$ for $i \geq 1$) in Jenny, et al., 2003. The multiscale control volume methods described in the context of linear preconditioners are the Mass Conservative Domain Decomposition preconditioners derived in Nordbotten & Bjørstad, 2008.

Remark 8: In *Remark 1* at the end of the discretization section we saw that the standard finite element method is obtained by choosing test functions that are in the same multiscale space W_H^{MS} as the solution space. One may ask if the same is the case for multiscale methods. The answer is that yes, in the sense that if the integration on the primal grid defined in Eq. (7) is replaced by a weighted sum, using the multiscale basis itself as weights, the classical Multiscale Finite Element method of Hou & Wu, 1997 is recovered.

3.4 Recovering a conservative fine-scale flux field

The method as outlined so far constructs a two-level set of control volume methods. This can be seen from several perspectives: Either as the basis for a multi-level method, as the basis for a preconditioner in an iterative method, but also

from the perspective of deriving a new, (coarse) single-scale control-volume method. We will consider the third perspective in this section.

When discussing the coupled set of equations outlined in Section 2.1, we pointed out the importance of retaining local mass conservation. This property is often necessary to consider (almost) point-wise, while by construction, the control-volume methods consider this only on the primal cells of the grid. It is therefore natural to consider whether a post-processing can be performed to extend this cell-wise property to a more local property, and whether this operation can be conducted locally.

For a local post-processing, it is natural to use the (cell-wise conservative) fluxes over boundaries of the primal grid as the basis of solving Neumann boundary problems inside each cell. The Neumann problem for the elliptic problems we consider is well-known to only admit solutions if the compatibility condition is satisfied, which is to say that the boundary conditions exactly integrate to the sum of all internal sources or sinks. The control-volume methods satisfy the compatibility condition by construction. Note that after post-processing, we will obtain a flux field that is everywhere conservative, but as a consequence will not everywhere satisfy Darcy's law.

In the case of single-scale control-volume methods, the permeability coefficient k is usually considered constant inside each primal cell, and locally post-processed fluxes can be calculated analytically for some cell shapes. While this is not used much from the perspective of practical simulation, it is an invaluable tool in the derivation of error estimates.

For the multiscale control-volume method, the permeability is of course possibly heterogeneous inside each coarse primal cell, and a numerical calculation must be performed as a post-processing step. This can be achieved using the same grid and discretization as used when obtaining the multi-scale basis functions, and leads to an approximation with the following important properties: A post-processed flux which is conservative on the fine-scale primal grid. This post-processed flux allows for transport simulations to be performed on a significantly finer grid than the coarse control-volume scheme that was derived.

It is important to note that the possibility of post-processing the fluxes is the most important property of the multiscale control-volume method. Moreover, the construction of the MCDD preconditioner explicitly preserves this property, such that at any iteration of an iterative approach, the approximate solution to the fine-scale problem can also be post-processed in an identical manner.

3.5 Multiscale methods as iterative solvers

The domain decomposition method formulated in Section 3.3 can be applied as a stand-alone solver for the pressure system (4). This was the approach advocated in the early multiscale papers (Hou & Wu, 1997; Jenny, et al., 2003; Aarnes, 2004). Since the action of the method on a vector can be evaluated solving local systems related to the Schur complements, as well as a (relatively small) coarse linear system, we understand that the method offers an efficient way to obtain a pressure approximation and a mass conservative fine scale velocity field. Indeed, simulations of petroleum recovery indicate that in some cases, this strategy provides a fairly accurate and very cheap alternative to traditional approaches.

However, the above strategy is insufficient for more challenging problems. A particular weakness of the multiscale methods is the reliance on somewhat arbitrary approximations to the Schur complements $\hat{S}_{ii}$. Indeed, since the approximate Schur complements determine the subspace W_H^{MS}, we understand that for any approximate Schur complement, cases exist where the solution to the fine-scale problem lies in a space orthogonal to the multiscale space. Thus multiscale methods as direct solvers will always have problems with robustness. The practical performance of multiscale methods unfortunately deteriorates with the number of spatial dimensions; to be specific, multiscale methods have turned out to perform significantly worse in three spatial dimensions than in 2D.

When faced with these issues, there are a few techniques that can be applied to improve the solution. One is to consider sophisticated ways to construct $\hat{S}_{ii}$, using in particular non-local information and information about the right hand side. Another approach, to be described next, is to apply the MCDD preconditioners in an iterative setting to improve the approximation. The simplest such strategy is a Richardson scheme, where we, equipped with an initial guess p^0, define the iterative scheme

$$p^{l+1} = p^l + \tau B_{MS}(b - Ap^l),$$

where B_{MS} represent one application of the multiscale method and τ is a damping factor. We observe that when the multiscale method is applied as a stand-alone solver, this corresponds to applying a single Richardson iteration with the MCDD preconditioner and $\tau = 1$. The Richardson scheme will in general exhibit poor convergence for our problem. A better utilization of the multiscale method is as a preconditioner inside an iterative solver such as GMRES (Saad & Schultz, 1986). Since the problem is likely to be more difficult in some parts of the domain, the application of the preconditioner can be restrained to those parts, if they can be identified by error estimates.

An important and often time consuming ingredient of GMRES is to ensure orthogonality of the basis vectors for the Krylov subspace in which the approximated solution lies. When GMRES is preconditioned with the multiscale method, this computational cost can be reduced considerably by exploiting a special feature of the solution: If the internal nodes p_N are eliminated using an exact solver, e.g. introducing no approximation to S_{NN}, the residual in the interior will be zero after one application of the preconditioner as discussed in *Remark 7*. This does not mean the pressure is exact for those cells, but rather that the influence of nodes on level N on the residual is lumped into the higher levels. This also means that GMRES does not need to minimize the residual for cells on level N, the orthogonalization needs only consider levels $0, \dots, N-1$, leading to an often significant reduction of the computational cost. Note that level N cannot be totally ignored, since some nodes there are a part of the flux expression for level $N-1$.

We now realize that the MCDD applied as a preconditioner in an iterative setting possess several advantageous features in comparison to standard preconditioners:

1. For relatively simple problems, where standard multiscale methods are applicable, the iterative procedure can be terminated after a single iteration.
2. For moderately complex problems, the iterative method can be terminated at any point where the solution is deemed accurate enough, and a locally conservative flux field can be recovered.
3. For truly challenging problems, the MCDD preconditioner is comparable to standard non-overlapping domain decomposition based preconditioners for these problems.

Thus we see, for applications where the exact solution to the linear system is not necessary, the current methodology allows for a substantial savings in number of iterations. This is of great practical importance, since the error introduced by a discrete approximation to (2) can frequently be orders of magnitude larger than the tolerance used in traditional linear solvers.

3.6 Computational cost

While a full assessment of the computational cost is beyond the scope of the chapter, we will make some brief comments that allow the reader to get a general impression of the cost of both the multiscale methods as well as their application as preconditioners.

The computational cost of the MCDD preconditioner is composed of three components. First, the approximate Schur complement system involves approximating the action of S_{NN} on a (small) set of vectors. Physically, this corresponds to solving the local problems inside the internal subdomains for given boundary conditions. Denoting the number of internal subdomains as N_{SD}, a naïve estimate of cost would be $N_{SD} \cdot \dim p_0$. However, by construction, most approximate Schur complements will be local, such that each subdomain typically only has non-zero boundary conditions associated with the variables in p_0 that are associated with cells on the boundary of the subdomain. For Cartesian coarse subdomain, this is identified as the corners, such that the computational cost is proportional to $N_{SD} \cdot 2^d \cdot C_F$. Here C_F is defined as the coarsening factor, which is the ratio of degrees of freedom in the fine and coarse spaces, $C_F \equiv \frac{\dim p}{\dim p_0}$. The multiscale basis is only calculated once.

Secondly, there is a cost associated with the right-hand side, which needs to evaluated at every iteration. As seen in Section 3.2, the right hand side is also associated with local calculations, forced by source terms in contrast to the multiscale basis functions. The cost is thus proportional to $N_{SD} \cdot C_F \cdot N_I$, where N_I is the number of iterations.

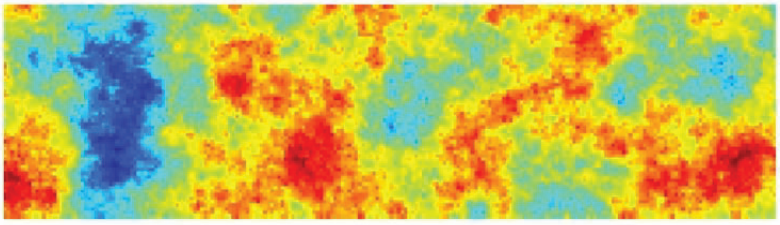
Finally, there is the cost associated with solving the coarse set of equations. Here, there are two contrasting strategies. The domain-decomposition strategies argue for aggressive coarsening, where the coarse problem has (almost) negligible size and cost. This has the advantage that the cost of the coarse solve can be neglected, at the expense of more costly construction

of the multiscale basis. However, as the multiscale basis calculation is trivially parallel, this may be a good strategy on some computational architectures, and in particular if the selection of coarse grids is hard to automate. A contrasting strategy is in the multi-grid flavor, where a much less aggressive coarsening is applied, which leads to a non-negligible cost in the coarse problem. However, since the coarse problem has the same control-volume structure as the fine-scale discretization, the multiscale method can be called recursively. The resulting algorithm has a better performance from the perspective of computational cost, but may be more difficult to implement as the problem is no longer trivially parallel. Note that for a conservative approximation to be obtained, the reconstruction of the flux field must also be conducted recursively.

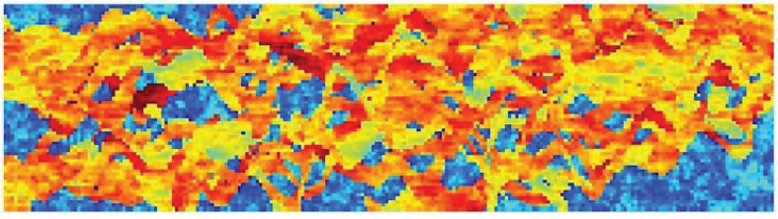
In general, the multiscale methods are designed for problems where there is a coupling between the permeability k and the concentration field c. As c evolves locally, the multiscale basis functions may only need to be updated locally in space, allowing for further computational savings compared to a generic linear solver that is not adapted to these features. These aspects have been carefully highlighted in a suite of 2D test cases (Kippe, et al., 2008) .

4. **Numerical examples**

In this section, we show numerical examples illustrating the properties of the domain decomposition method. For these examples, we have chosen the permeability field defined according to the SPE 10th comparative benchmark study much used to study upscaling and multiscale methods (Christie & Blunt, 2001). This test case involves a Cartesian 60 x 220 x 85 grid. The permeability in the upper 35 layers have a somewhat smooth distribution (consistent with a shallow marine depositional system), whereas the 50 lower layers are characterized by sharp permeability contrasts and highly permeable channels with long correlation length (consistent with a fluvial depositional system). The lower layers are expected to pose challenges for linear solvers. Representative layers from the upper and lower parts of the formation are shown in Fig. 3. The permeability field spans more than 10 orders of magnitude, rendering a challenging test problem for our methods.



(a) Uppermost layer



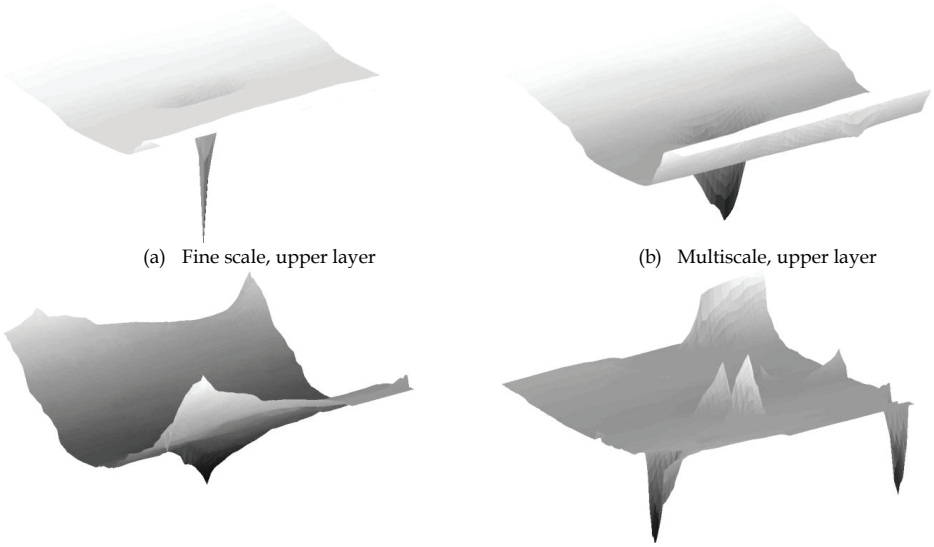
(b) Lowermost layer

Fig. 3: The base-10 logarithm of the permeability from the uppermost (a) and lowermost (b) layers in the SPE 10 test case. These are used in the 2D tests, and they are representative for the upper and lower part of the 3D formation, respectively. Blue and red corresponds to high and low-permeable regions, respectively. For convenience, the figures are rotated **90°**.

On this grid, we will consider simple setups, with one injection well and one producer. Both for 2D and 3D tests the injector is located along the boundary (the position differs somewhat between the tests, as we avoid injecting into low-permeable cells), and the producer is located in the middle of the domain. The pressure equation (2) is discretized using a two-point scheme, and for simplicity, periodic boundary conditions are assumed. For all test cases, post-processing of the flux field as discussed in Section 3.4 will be applied to ensure mass conservation on the fine scale.

4.1 2D examples

We start with two instructive examples in 2D, using permeability from the uppermost and lowermost layer of the SPE10 dataset, as pictured in Fig. 3. Thus the fine scale grid has 60 x 220 cells, and we use a coarse grid with 4x20 cells, rendering a coarsening factor of 165. Fig. 4 shows the pressure profiles obtained by a fine scale solution and the multiscale solver. For the upper layer, the multiscale solution is similar to that of the true solution; and has a quality that is as good as can be expected keeping in mind that the multiscale method is essentially a coarse discretization. In both solutions the pressure contours clearly indicate flow from injector to the producer, although again, the resolution of the local flow around the producer is better refined on the fine-scale grid. For the lower layer, the multiscale solution is highly oscillatory with false local minima in the solution. This can be interpreted as a case where the approximation to the Schur complement $\hat{S}_{ii}$ is not good enough, where a better approximation, or iterations, are needed to produce a pressure profile that resembles that of the fine scale solution.



(a) Fine scale, upper layer (b) Multiscale, upper layer

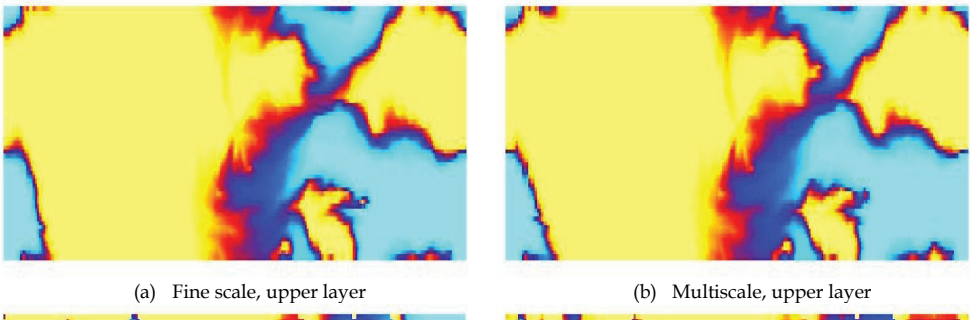
(c) Fine scale, lower layer (d) Multiscale, lower layer

Fig. 4: Pressure solutions obtained by a fine scale and a multiscale solution for the uppermost and lowermost permeability layers. The injection well is located along the left boundary for all plots, while the producer is associated with the downward spike visible in the middle of the domain visible in all figures except (d).

For the uppermost layer of SPE10, the relatively good MS approximation to pressure is reflected in the post-processed fluxes. We illustrate this by the solution to the transport equation (1), as displayed in Fig. 5 (a) and (b). Note that despite the relatively coarse grid used for the multiscale control-volume approximation, the reconstruction of the fine-scale fluxes leads to a flow field with no visible artifacts. From the perspective of practical simulation, the solutions are indistinguishable.

Surprisingly, despite the relatively poor approximation to the pressure field, quite satisfactory fluxes can be obtained also for the lowermost layer as shown in Fig. 5 (c-d). This illustrates that the coarse scale conservation of mass combined with post-processing of the velocity field, leads to a multiscale approximation that is applicable to transport problems also for highly challenging problems. Note however that in these lower layers, the multiscale approximation leads to some cases where flow-channels are either suppressed or exaggerated.

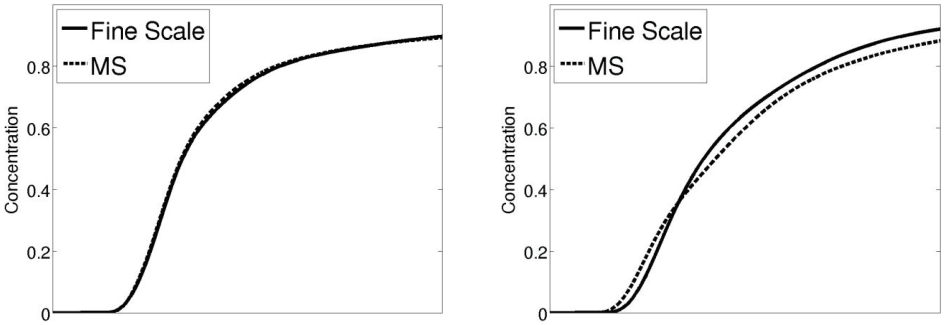
The results from the concentration maps in Fig 5 are further confirmed by considering time series of the concentration in the production well, as shown in Fig 6. For the upper layer, the curves corresponding to the fine scale and multiscale solutions are almost identical, and the differences are relatively small also for the lower layer.



(a) Fine scale, upper layer (b) Multiscale, upper layer

(c) Fine scale, lower layer (d) Multiscale, lower layer

Fig. 5: Concentration profiles obtained by solving the transport equation based on the post-processed pressure solutions. High concentration of the injected species is indicated with blue. We emphasis that for the multiscale solution, the velocity field is post processed to achieve local conservation of mass.



(a) Upper layer (b) Lower layer

Fig. 6: Time series of the concentration in the upper and lower layer.

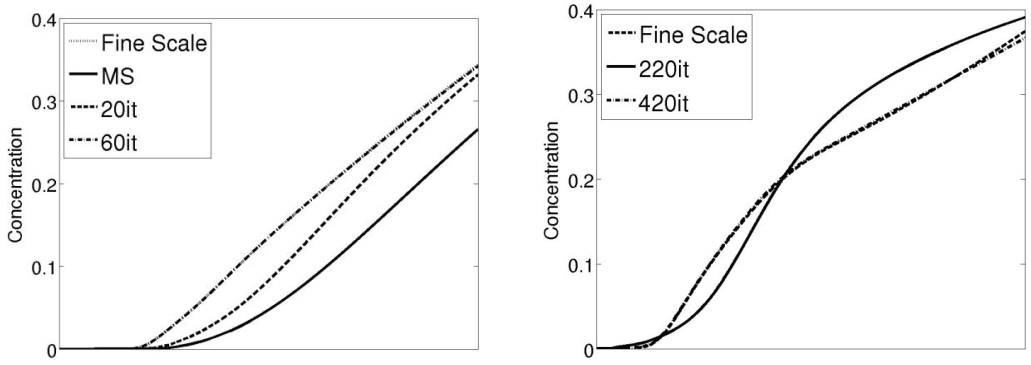
Remark 9: The appearance of oscillatory behavior in the multiscale solution is not unexptected. Again, we can analyze the multiscale control-volume method as simply being a single-scale control volume method on the coarse primal grid. It is known that for problems where the anisotropy in k is not aligned with the grid, local control-volume methods (and indeed this also holds for some other discretization families) in general cannot be constructed that are both consistent, as well as oscillation-free (Nordbotten, et al., 2007; Keilegavlen, et al., 2009). The channelized features that are shown in Fig. 3b are clearly not aligned with the general directions of the domain, and therefore they will lead to an effective permeability on the coarse grid that is also not aligned with the grid. The argument from the single-scale methods can thus be lifted to the multi-scale setting, which then informally may be states as: *No approximation $\hat{S}$ can be defined that leads to a local coarse-level control-volume method that is monotone for general channelized media*.

4.2 3D examples

The 2D examples showed that the multiscale method can provide reliable solutions for challenging permeability fields and relatively high coarsening ratios. As previously mentioned, the performance of the multiscale method deteriorates significantly when going from 2D to 3D. As we will see, the multiscale solution may be insufficient for transport purposes, and the application as a MCDD preconditioner inside an iterative solver is essential in order to recover accuracy. For all 3D simulations, we consider coarse grid cells composed of 15 x 11 x 5 fine cells, rendering a coarsening ratio of 825.

4.2.1 Multiscale method as preconditioner

We first consider simulations in the 10 uppermost and lowermost layers of the SPE10 formation, extending the two cases considered in the 2D case. Again there is an injection well in a corner of the domain, and a producer in the middle of the domain. We consider transport solutions based on a fine scale solution, a pure multiscale solution, and from MCDD preconditioned GMRES iterations. Since visualization is more difficult in 3D than in 2D, we will in 3D only give the time-series type plots similar to Figs. 6. The time series of concentration in the production cells are shown in Fig. 7 both for the upper and lower layers. For the upper layers, we observe that in contrast with the 2D examples, the multiscale solution now deviates significantly from the fine scale solution. Applying some GMRES iterations improves the quality of the solution somewhat, until after a sufficient number of iterations renders a curve that is indistinguishable from the fine scale. For the lower layers, the stand-alone multiscale solver produce a time series that is vastly different from the fine scale solution, and it is therefore not shown in the figure. For this difficult problem, it takes more iterations to produce a time series that resembles that of the fine scale solve.



(a) Upper layers (b) Lower layers

Fig. 7: Time series of the concentration in the production well for simulations in the upper and lower part of the SPE10 formation. The solutions obtained from the fine scale are located on top of those from 60 and 420 GMRES iterations for the upper and lower parts, respectively. A multiscale solution is not shown for the lower part, due to the low quality of the results produced.

The above test shows that in 3D the multiscale method does not reproduce concentration curves that are comparable to the fine-scale curves even when for relatively easy case of the upper layers of SPE10. Thus the present test shows the utility of having a framework that, when the fast multiscale solution is insufficient, can fall back to an iterative scheme, and fairly quickly recover a velocity field that is good enough for transport purposes. The increased difficulty in approximating the solution is also shown in the development of the residual during the corresponding GMRES iterations, see Fig. 8. In the lower layers the residual decreases slower and more iterations are needed to obtained what might be deemed a satisfactory solution. Based on these two simple tests, we observe that the relative residual error in the linear solver of 10^{-3} to 10^{-4} is needed in order to reproduce a good transport solution. This value is many orders of magnitude higher than the typical residual errors used in iterative solvers for the linear system. Indeed, the residual error as a function of iterations is shown in Fig. 8, where we see that more than 600 iterations are needed to obtain a converged iteration for the upper layers. The ability to truncate the iterations early when using MCDD as a preconditioner will therefore in this case represent a savings of about 90% in terms of number of iterations in the iterative solver. Also for the lower layers, an early truncation saves a majority of the computational effort.

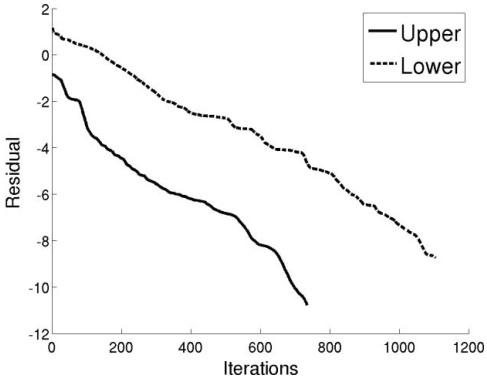


Fig. 8: The residual as a function of GMRES iterations for parts of the upper and lower part of the formation. The iterations terminates when the relative residual is reduced to a factor $\mathbf{10^{-10}}$.

At this point, it is appropriate to mention a third option to improve the multiscale solution, in addition to advanced approximations of the Schur complements $\hat{S}_{ii}$ and increasing the number of iterations: By increasing the number of coarse variables (in essence moving variables to level 0), the range of the multiscale basis functions ψ^{MS} can be increased to capture more of the solution. These ideas are exploited in (Sandvin, et al., Submitted), and show promising results in that the number of iterations needed can be reduced significantly with only a minor increase in the computational cost.

4.2.2 Quality control of MCDD solution

Returning to the original formulation of the system, we realize that an MCDD-based solution can be interpreted as the *exact* solution of Equations (2), for a modified permeability k^*. Using the pressure solution from the iterative solver together with the post-processed fluxes, we can calculate k^*. As the permeability is typically a value associated with great uncertainty for geological applications, we can compare the difference between k and k^* as a metric on the quality of the MCDD-based approximation. The most basic version of this comparison is to recall that from the physical motivation of the problem, the original permeability k is symmetric positive definite, and we can thus assess the quality of the approximate solution based on whether the modified permeability k^* also satisfies this physical constraint.

In Table 1, iteration counts and the number of sign changes are shown for a series of residual tolerances for GMRES. The grid consist of 60x220x10 cells, and the permeability is found from the channelized part of the SPE10 formation. Note that for the SPE10 dataset, the permeability tensor is diagonal, so positive definiteness is equivalent to positive diagonal elements. The table shows that for a high residual tolerance, more than a third of the fluxes change sign during post processing. Moreover, even for high accuracy of the GMRES solution there are some sign changes during flux post-processing.

$\log_{10}$(Tolerance)	-2	-4	-6	-8	-10
Iterations	224	429	717	915	1079
Negative elements in k^*	37%	9.4%	1.8%	.17%	.0018%

Table 1: The relative residual in GMRES, together with numbers for iterations and percentage of negative elements in the modified permeability.

The deviation of the flux and potential from a physical flow field, as measured by k^*, represents one attractive metric for assessing the approximation quality. However, more classical *a posteriori* error bounds and estimates are also applicable in this setting, and may be of equal importance for practical applications.

5. Concluding remarks

The present chapter has reviewed the construction of multiscale control volume methods in arbitrary dimensions from an algebraic perspective, allowing for a completely decoupled implementation of the fine-level (control volume) discretization and the multi-scale framework. We have emphasized several important aspects, including the points where key

approximations are made, together with both their algebraic and physical interpretations. By bringing attention to the formulation of multiscale methods in this general setting, we have been able to highlight aspects of how multiscale control volume methods relate to classical single-scale discretizations, iterative preconditioners, and multi-level approximations. Through carefully chosen numerical examples, we have sought to illustrate both the quality of the multiscale approximation to the primary variable (pressure), but more importantly the role of the multiscale approximation in the setting of a coupled system of equations. These examples clearly illustrate the increasing complexity faced with problems in 3D over 2D, and the care with which one needs to deal with notions of approximate solvers and multiscale numerics.

In closing this chapter we wish to take the opportunity to discuss some of the main obstacles and benefits of multiscale methods as one considers more challenging problems.

As a stand-alone solver for a single elliptic problem, it is difficult for multiscale methods and preconditioners to compete with multigrid methods. The advantage of the methodology lies therefore in different aspects.

- A coarse discretization is obtained directly, with explicit coarse flux expressions, leading to an understanding of the nature of the effective coarse-scale operator for the system.
- For time-dependent, where multiple (similar) problems need to be solved in succession, a large amount of calculations can be re-used from previous time-steps.
- For (locally) spatially periodic problems, sub-domain problems may be identical and computational savings can be obtained through re-use again.
- For problems with scale-separation (where homogenization is applicable), the multiscale method gives a good approximation to both the homogenized and true solutions after a single iteration.

Despite the initial promise, and the evidence that the advantages can be realized for model problems, several challenges remain before multiscale methods attain the robustness required for practical applications. Some of the major limitations, together with their potential remedies are:

- For irregular grids (both on the fine and coarse scale) and for anisotropic media, the multiscale approximation is again less robust, especially when local Schur approximations are applied. To some extent, this can be overcome by oversampling, through enriching the coarse space, or by bottom-up approaches such as matrix probing, although as noted in *Remark 9*, a local and consistent coarse operator can in general never be designed.
- For non-linear elliptic equations (e.g. is the permeability coefficient is a function of the pressure or its gradients), the method is no longer residual-free in the interior if the multiscale basis functions are re-used. Recalculating multiscale basis functions in an iterative setting is prohibitively expensive, and it remains unclear if good multiscale approximations can be constructed.
- For higher-dimensional problems (more than 3), the quality of multiscale approximations has yet to be addressed at all.

With these perspectives in mind, it is clear that multiscale methods and preconditioners are still a topic of very active research. As such, there will most certainly be aspects of the current chapter that later research will both clarify and improve upon. Nevertheless, we hope that the present text succeeds in giving a current perspective on multiscale methods that will have value for both the general and specialized reader.

Acknowledgement

This research was financed in part through the Norwegian Reseach Grant #180679, "Modelling Transport in Porous Media over Multiple Scales".

Bibliography

Aarnes, J., 2004. On the use of a mixed multiscale finite element method for greater flexibility and increased speed or improved accuracy in reservoir simulation. *SIAM Multiscale Model. Simul.*, 2(3), pp. 421-439.

Aavatsmark, I., 2002. An introduction to the multipoint flux approximations for quadrilateral grids. *Comput. Geosci.*, 6(3-4), pp. 405-432.

Aziz, K. & Settari, A., 1979. *Petroleum Reservoir Simulation.* s.l.:Chapman & Hill.

Bear, J., 1972. *Dynamics of Fluids in Porous Media.* s.l.:Elsevier.

Chan, T. & Mathew, T., 1992. The interface probing technique in domain decomposition. *SIAM J. Matrix Anal. Appl.,* 13(1), pp. 212-238.

Chen, Z., Huan, G. & Ma, Y., 2006. *Computational methods for multiphase flows in porous media.* s.l.:SIAM.

Christie, M. & Blunt, M., 2001. Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques. *Proc. of SPE Reservoir Simulation Symposium.*

Edwards, M. & Rodgers, C., 1998. Finite volume discretization with imposed flux continuity for the general tensor pressure equation. *Comput. Geosci.,* 2(4), pp. 459-490.

Efendiev, Y. & Hou, T., 2009. *Multiscale finite element methods: theory and applications.* s.l.:Springer.

Hou, T. & Wu, X.-H., 1997. A multiscale finite element method for elliptic problems in composite materials and Porous media. *J. Comput. Phys.,* 134(1), pp. 169-189.

Jenny, P., Lee, S. & Tchelepi, H., 2003. Multi-scale finite-volume method for elliptic problems in subsurface flow simulations. *J. Comput. Phys.,* 187(1), pp. 47-67.

Keilegavlen, E., Nordbotten, J. & Aavatsmark, I., 2009. Sufficient criteria are necessary for monotone control volume methods. *Appl. Math. Lett.,* 22(8), pp. 1178-1180.

Kippe, V., Aarnes, J. & Lie, K.-A., 2008. A comparison of multiscale methods for elliptic problems in porous media flow. *Comput. Geosci.,* 12(3), pp. 277-298.

Lunati, I. & Jenny, P., 2007. Treating highly anisotropic subsurface flow with the multiscale finite-volume method. *SIAM Multiscale Model. Simul.,* 6(1), pp. 308-318.

Lunati, I., Tyagi, M. & Lee, S., 2011. An iterative multiscale finite volume algorithm converging to the exact solution. *J. Comput. Phys.,* Volume 230, pp. 1849-1864.

Nordbotten, J., Aavatsmark, I. & Eigestad, G., 2007. Monotonicity of control volume methods. *Numer. Math.,* 106(2), pp. 255-288.

Nordbotten, J. & Bjørstad, P., 2008. On the relationship between the multiscale finite-volume method and domain decomposition preconditioners. *Comput. Geosci.,* 12(3), pp. 367-376.

Quateroni, A. & Valli, A., 1999. *Domain decomposition methods for partial differential equations.* s.l.:Oxford University Press.

Saad, Y. & Schultz, H., 1986. A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.,* 7(3), pp. 856-869.

Sandvin, A., Keilegavlen, E. & Nordbotten, J., Submitted. Auxiliary variables in multiscale simulations. *J. Comput. Phys..*

Sandvin, A., Nordbotten, J. & Aavatsmark, I., 2011. Multiscale mass conservative domain decomposition preconditioners for elliptic problems on irregular grids. *Comput. Geosci.,* 15(3), pp. 587-602.

Smith, B., Bjørstad, P. & Gropp, W., 1996. *Domain Decomposition.* s.l.:Cambridge University Press.

Tchelepi, H. & Juanes, R., 2008. *Special issue on multiscale methods for flow and transport in heterogeneous porous media.* 3 ed. Comput. Geosci.: s.n.

Toselli, A. & Widlund, O., 2005. *Domain Decomposition Methods - Algorithms and Theory.* s.l.:Springer.