Adaptive Multiscale Methods Based on A Posteriori Error Estimates

Thesis for the degree of Master of Science
in Applied and Computational Mathematics

Sergey Alyaev

Department of Mathematics
University of Bergen
Norway

Tuesday 1st June, 2010

Contents

Preface

Abstract

We give an overview of different methods for solving highly heterogeneous elliptic problems with multiscale structure and no intuitive scale separation. We compare different finite element variational multiscale methods and prove equivalence between the methods proposed by Larson et al. [LM07] and Nolen et al [NPP08]. We also discuss properties of different multiscale methods depending on the choice of scale separation and ways to represent the fine scale correction. Additionally, in this work we give an overview of *a posteriori* error estimates for the finite element method as well as newly proposed by Larson et al. estimates for the variational multiscale method [LM07]. As an illustration of the theory we show our numerical results for using theoretical estimates to construct adaptive algorithms: adaptive refinement of finite elements and adaptive overlap control for variational multiscale methods in the formulation of Nolen et al. There is no known implementation of the latter published at the moment.

Acknowledgments

First of all I would like to thank my supervisors Jan M. Nordbotten and Jan Valdman for wise guidance and support in my research as well as inspiration. Without their help this work would not exist. I would like to thank many other people for help: Petter Bjørstad for sharing his experience and giving references in the field; Alexander Malyshev, Talal Rahman, Eirik Keilegavlen, Mauricio Godoy and Georgy Ivanov for help in figuring out many small but important details concerning the work; Marie Saltnes for sharing her lecture notes.

I also want to thank Alexander Vasiliev for establishing the agreement with Saratov State University that gave me a chance to study at the University of Bergen. I am also very thankful to my parents and all my friends for

moral support during my studies, and especially Leonid Vasilyev and Gleb Berdnikov for sharing tea and conversations during the work.

Chapter 1

Introduction

In the first part of this chapter we will discuss where multiscale problems arise. We also give an overview of so-called multiscale methods, and how and why they were developed. We discuss the motivation for this work and challenges in the field.

The second part of this chapter gives some major applications of multiscale methods for problems of flow in porous media. We also comment on how the methods, studied in this work, can be applied to those problems.

1.1 Motivation

Problems with highly oscillatory coefficients appear in many applications such as quantum physics, composite materials, and fluid mechanics including flow in porous media[LM07]. In many cases they have so-called multiscale structure meaning that the features of parameters influencing coarse solution cannot be represented on this scale and the problem is to be solved in multiple scales. Problems with multiscale nature are very expensive to solve as the amount of variables on the fine scale can be too large for the memory of modern desktop computers. That is why during last decade there was an increase of popularity of approximate, so-called, multiscale methods that allowed achieving a rough but acceptable solution in short time even on desktops and quite many papers describing them were published, such as [HW97, HFMQ98, LM07, NPP08, Nor08]. The main idea of this class of methods is to compute the fine scale correction also known as fine scale Green's functions on the preprocessing step. In the main part of the algorithm they only perform a much cheaper work of solving a perturbed coarse scale problem.

Though multiscale methods were introduced in nineties and were being

used for practical engineering applications they still remain controversial. There are many ways of doing multiscale computations that were proposed and implemented by different authors but there is not so much comparison done. There is no knowledge of how to do it best, either. Another open issue about approximate multiscale methods is whether they should be preferred to traditional exact multilevel methods such as domain decomposition or multigrid. The ambition of this work is to overview some examples of exact and inexact two-level methods for solving multiscale elliptically driven problems and compare them in some aspects.

Another important thing during solving expensive problems is to optimize the computational time that is required to achieve the result. In order to improve the computational time, it is very important to use theoretical knowledge available. In this work we discuss *a posteriori* error estimates and their applications for creating adaptive algorithms.

1.2 Applications to flow in porous media

Here we will highlight the applications of multiscale methods to flow in porous media and formulate related problems going from their physics as it is done in [Bea88, Aav]. These problems are very important for a number of applications such as ground water flow, reservoir simulation and simulation of geological storage of carbon dioxide.

1.2.1 Incompressible single phase flow in porous media

First we consider single phase flow. Mass conservation for any domain Ω implies

$$\int_{\Omega} \frac{\partial}{\partial t}(\varphi\rho)d\tau + \int_{\partial\Omega} \rho v \cdot n d\sigma = \int_{\Omega} Q d\tau, \tag{1.1}$$

where φ is porosity, ρ is density of fluid, v volumetric flow density, n is an outer normal to the boundary of Ω and Q is mass source density.

For classical problems of flow in porous media the phenomenologically derived Darcy's law is valid

$$v = -\frac{K}{\mu}(\nabla p - \rho g \nabla D), \tag{1.2}$$

where μ is viscosity, K - the permeability tensor, p - pressure, g - gravitational acceleration, D - depth. If we substitute the formula for flow density into the

mass conservation equation (1.1) and apply Gauss-Green theorem [Eva98] to its second term we will end up with a general law for flow in porous media

$$\int_{\Omega} \left[\frac{\partial}{\partial t}(\varphi\rho) - \nabla \cdot \left(\frac{K}{\mu}(\nabla p - \rho g \nabla D) \right) - Q \right] d\tau = 0. \tag{1.3}$$

If the integrated functions are sufficiently smooth it is possible to rewrite (1.3) in divergent form:

$$\frac{\partial}{\partial t}(\varphi\rho) - \nabla \cdot \left(\frac{K}{\mu}(\nabla p - \rho g \nabla D) \right) - Q = 0. \tag{1.4}$$

If we consider incompressible flow that is described by $\varphi \equiv const$ and $\rho \equiv const$; and introduce a potential $u = p - \rho g D$ we will end up with an elliptic equation:

$$-\nabla \cdot \left(\frac{K}{\mu} \nabla u \right) = \frac{Q}{\rho}. \tag{1.5}$$

In the later chapters considering this problem we will use the notations: $a(x) = \dfrac{K(x)}{\mu}$ and $f(x) = \dfrac{Q(x)}{\rho}$.

Equation (1.5) and, especially, its weak form are very important for many applications. In many cases of porous media we have that K is highly oscillatory and contains features that are not always captured on the scale of discretization that is reasonable for performing simulations. This makes this problem interesting for solving by multiscale or multilevel algorithms.

1.2.2 Weakly compressible fluid

In the case of weakly compressible fluid we introduce a parameter for compressibility:

$$c = \frac{1}{\rho} \frac{d\rho}{dp}. \tag{1.6}$$

We assume that flow is weakly compressible, which in the mathematical model is equivalent to assuming that c is constant and it is small. c is small in the following sense [Aav]:

$$c \left(\frac{\partial p}{\partial x_i} \right)^2 \ll \left| \frac{\partial^2 p}{dx_i^2} \right|. \tag{1.7}$$

We also assume that we can neglect gravity. Applying these assumptions to the divergence form of the flow equation (1.4) we get

$$\begin{aligned} 0 &= \frac{\partial}{\partial t}(\varphi\rho) - \nabla\cdot\left(\frac{\rho}{\mu}K\nabla p\right) - Q \\ &= \varphi c\rho\frac{\partial p}{\partial t} - \rho\nabla\cdot\left(\frac{K}{\mu}\nabla p\right) - \frac{c\rho}{\mu}\nabla p\cdot\frac{K}{\mu}\nabla p - Q \\ &= \varphi c\rho\frac{\partial p}{\partial t} - \rho\nabla\cdot\left(\frac{K}{\mu}\nabla p\right) - Q, \end{aligned} \tag{1.8}$$

as $\frac{c\rho}{\mu}\nabla p\cdot\frac{K}{\mu}\nabla p$ is negligible due to our assumptions. After dividing this equation by ρ we end up with parabolic equation with dominating elliptic term [Aav]:

$$\varphi c\frac{\partial p}{\partial t} = \nabla\cdot\left(\frac{K}{\mu}\nabla p\right) + \frac{Q}{\rho}, \tag{1.9}$$

here ρ can be computed by integration of compressibility equation (1.6).

As the equation (1.9) is dominated by elliptic term the methods for solving elliptic equations can be applied to it after introducing time discretization. As K for this problems still behaves badly one needs to invent something to solve them fast and efficiently. Here multiscale methods are even more applicable than in purely elliptic problems as it is usually enough to compute corrections only once or update them rarely.

1.2.3 Two-phase flow in porous media

Here we will consider a case of two-phase flow in porous media assuming that fluids are incompressible and porosity is constant. In this case we will have conservation equation for each phase, similar to (1.1):

$$\int\limits_{\Omega}\frac{\partial}{\partial t}(\varphi s_i)d\tau + \int\limits_{\partial\Omega} v_i\cdot n d\sigma = \int\limits_{\Omega}\frac{Q_i}{\rho_i}d\tau, \quad i=1,2; \tag{1.10}$$

where i represents the indices of two fluids e.g. oil and water and parameters with i here and later correspond to fluid i; s_i is saturation of phase i. Then we write Darcy's law for each phase similarly to equation (1.2)

$$v = -\frac{K_{r,i}}{\mu_i}K(\nabla p_i - \rho_i g\nabla D), \quad i=1,2; \tag{1.11}$$

where $K_{r,i} = K_{r,i}(s_i)$ denotes relative permeability dependent on saturation of the phase for which the following inequalities should hold

$$K_{r,0}(s_0) + K_{r,1}(s_1) \leq 1, \tag{1.12}$$

$$\frac{\partial K_{r,i}}{\partial s_i} \geq 0, \quad i = 1, 2. \tag{1.13}$$

We introduce total volumetric flow density $v = v_0 + v_1$ and from mass conservation equation applying assumptions of incompressibility get

$$\int\limits_{\partial\Omega} v \cdot n d\sigma = \int\limits_{\Omega} \left(\frac{Q_0}{\rho_0} + \frac{Q_1}{\rho_1} \right) d\tau, \tag{1.14}$$

that is an elliptic equation.

To come to final set of equations we need to introduce new variables and parameters:

- mobility $\lambda_i = \dfrac{K_{r,i}}{\mu_i}$,
- fractional flow function $f_1 = \dfrac{\lambda_1}{\lambda_0 + \lambda_1}$,
- capillary pressure $p_c = p_0 - p_1$.
- $h = \dfrac{\lambda_0 \lambda_1}{\lambda_0 + \lambda_1}$

Computing the explicit formula for volumetric flow density of phase 1 from Darcy's law (1.11) and using the introduced parameters we get

$$v_1 = f_1 \cdot v + hK(\nabla p_c + (\rho_1 - \rho_0) g \nabla D). \tag{1.15}$$

Inserting this expression for flow density into conservation equation with $i = 1$ from (1.10) we have

$$\int\limits_{\Omega} \varphi \frac{\partial s_1}{\partial t} d\tau + \int\limits_{\partial\Omega} [f_1 \cdot v + hK(\nabla p_c + (\rho_1 - \rho_0) g \nabla D)] \cdot n d\sigma = \int\limits_{\Omega} \frac{Q_1}{\rho_1} d\tau. \tag{1.16}$$

Rewriting the equation in divergent form gives

$$\varphi \frac{\partial s_1}{\partial t} + \nabla \cdot [f_1 \cdot v + hK(\nabla p_c + (\rho_1 - \rho_0) g \nabla D)] - \frac{Q_1}{\rho_1} = 0 \tag{1.17}$$

where after simplifications we get

$$\varphi \frac{\partial s_1}{\partial t} + \nabla \cdot (fv) + \nabla \cdot [h(\rho_0 + \rho_1) g K \nabla D + hK \nabla p_c)] = \frac{Q_1}{\rho_1}, \qquad (1.18)$$

that is a non-linear parabolic equation with hyperbolic term $\nabla v f_1$ coupled with an elliptic equation (1.14) and algebraic equations for parameters that are usually derived experimentally gives formulation of the problem for two-phase incompressible flow in porous media. The methods for elliptic equations that are discussed in this work cannot be applied directly to solving this type of problems however they can be useful for solving for elliptic part of the problem or creating iterative method. When the hyperbolic nature dominates equation (1.18), this implies that changes happen locally and not all of the fine scale corrections should be recomputed on each step which greatly saves computational time.

Successful numerical experiments for multiscale solving of similar equations are presented by Nordbotten in [Nor08], that motivates further work in the direction of more complicated problems. Also during the last years there was a rapid development of multilevel methods for solving nonlinear equations such as the additive Schwarz preconditioned inexact Newton algorithm proposed by Cai et al. in [CK02] and generalization of variational multiscale methods to nonlinear problems proposed by Nordbotten in [Nor]. Good understanding of abilities and restrictions of the algorithms for linear equations lead to possibilities of applying similar techniques for solving more challenging problems.

Chapter 2

Theory

In this chapter we first in section 2.1 state the model problem for this work, that is an elliptic partial differential equation (PDE). This work focuses on solving this equation by finite element method (FEM) that is also described in section 2.1.

In section 2.2 we give an idea of *a posteriori* error estimates that can be used to characterize how good is the numerical solution compare to the exact solution. We also give an example of a classical estimate for finite element method.

The rest of the chapter is devoted to theory of multilevel methods. In section 2.3 we give an overview of multilevel solvers for elliptic equations and some theory lying beyond them. And section 2.4 is devoted to the variational multiscale method that is the main focus of this work. We show a derivation of the method, different forms of it, and *a posteriori* estimates for the class of multiscale methods.

2.1 Model problem

In this section we describe a numerical model that is the model problem of this work. It consists of an elliptic partial differential equation in the weak form that is described in the first part of the section, and the finite element method that concludes the section.

2.1.1 Weak formulation

In this work we will primarily consider the Poisson equation - our model problem:

Formulation 1 *Find* $u(x)$*, solving*

$$-\nabla \cdot a(x)\nabla u(x) = f(x), \quad \forall x \in \Omega, \tag{2.1}$$

with zero Dirichlet boundary conditions

$$u(x) = 0, \quad x \in \partial\Omega, \tag{2.2}$$

where $\Omega \subset R^2$ is a domain of the problem and $\partial\Omega$ is its boundary; a is a weight function, that will in general be a symmetric positive definite matrix; $f(x)$ is a load function.

Let us first reformulate (2.1) in the weak form. This can be easily obtained by multiplying it by a test function v and integrating over the whole domain Ω.

$$\int_\Omega (-\nabla \cdot a\nabla u)\, v dx = \int_\Omega f v dx, \tag{2.3}$$

and integration by part of the left hand side and applying boundary conditions (2.2) gives us:

$$\int_\Omega a\nabla u \cdot \nabla v dx = \int_\Omega f v dx. \tag{2.4}$$

In many textbooks, for example [Joh87, Bra07], it is proved that both u and v should lie in Sobolev space $H_0^1(\Omega)$.

It is easy to see that the integral on the right of (2.4) is a bilinear form in u and v. From now on, we use the notation:

$$a(u, v) \equiv \int_\Omega a\nabla u \cdot \nabla v dx. \tag{2.5}$$

We also notice that the right hand side of (2.4) is a linear functional in v. It is also an L_2 inner product over domain Ω

$$F(v) \equiv (f, v) \equiv \int_\Omega f v dx, \tag{2.6}$$

$(\cdot, \cdot)$ here and later denotes the L_2 inner product over default domain Ω; in the case when not both functions f and v are in L^2 we will still use this notation for a functional applied to v. We can check 4 properties of the formulation that are sufficient for existence uniqueness and stability of solution [Aav]. Let us list and check them.

1. Symmetry of the bilinear form

$$a(u,v) = a(v,u), \tag{2.7}$$

which is obvious.

2. Continuity of the bilinear form

$$\exists \gamma = const, \quad \forall u, v \in H_0^1, \quad |a(u,v)| \leq \gamma \|u\|_{H^1} \|v\|_{H^1}. \tag{2.8}$$

We perform a check:

$$|a(u,v)| = |(a\nabla u, \nabla v)| \leq \|au\|_{H^1} \|v\|_{H^1} \leq \left\|\sqrt{a}\right\|_{L_\infty} \|u\|_{H^1} \|v\|_{H^1}; \tag{2.9}$$

this gives a finite constant γ while a is in L_∞

3. V-ellipticity of the bilinear form

$$\exists \alpha = const, \quad \forall v \in H_0^1, \quad a(v,v) \geq \alpha \|v\|_{H^1}^2 . \tag{2.10}$$

Taking out the weight a from the bilinear form and applying Poincare-Friedrich's inequality (see for example [Bra07]) we have

$$a(v,v) \geq \left\|a^{-1}\right\|_{L_\infty} (\nabla v \cdot \nabla v) \geq (1+D)^2 \left\|a^{-1}\right\|_{L_\infty} \|v\|_{H^1}^2 , \tag{2.11}$$

where D is the diameter of our domain. If the a^{-1} is bounded then condition 3 is satisfied. If a is a tensor a similar argument should be applied to its eigenvalues.

4. Continuity of $F(v)$

$$\exists \Lambda = const, \quad \forall v \in H_0^1, \quad |F(v)| \leq \Lambda \|v\|_{H^1} . \tag{2.12}$$

This is also easy to check:

$$|F(v)| = (f,v) \leq \|fv\|_{L_2} \leq \Lambda \|v\|_{H^1} , \tag{2.13}$$

if the integral $\|fv\|_{L_2}$ is finite. This condition is satisfied for all $v \in H^1$ if $f \in H^{-1}$.

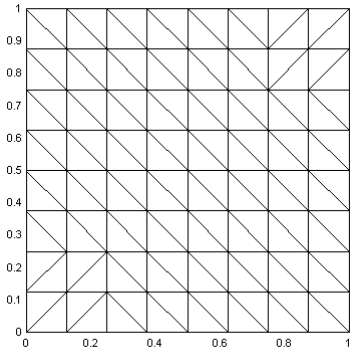

Figure 2.1: An example of triangular grid on a square domain.

2.1.2 Finite element method

The Ritz-Galerkin method suggests solving the problem (2.4) on a finite dimensional space V_h that is a subspace of the original problem space H^1. In case of finite element method we introduce a grid on Ω and so cold finite element basis. Let us give a precise mathematical definition of finite element method [Joh87]:

Definition 1 *Let*

- $K \subset \Omega$ *be a geometrical domain with piecewise smooth boundary,*
- P_K *be a finite-dimensional linear space of functions on* K*,*
- Σ *be a set of degrees of freedom chosen for any* $v \in P_K$ *to be uniquely defined by it;*

then the triple (K, P_K, Σ) *is called a finite element.*

Usually the basis associated to Σ is chosen to be piecewise-polynomial with support on the cells adjusted to one grid point.

In this work we stick to triangular grids (K is a triangle) as shown in Figure 2.1 as it is well suited for resolving complex geometry. In general, other types of grids can be used.

The example of a finite element basis that we use in numerical simulations for this work is linear basis, which is sometimes called nodal [BS94], constructed as follows

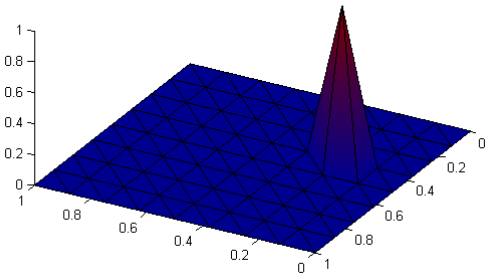

Figure 2.2: Linear basis function

$$\phi_i = \begin{cases} 1, & if \quad x = x_i \\ 0, & if \quad x = x_j, j \neq i \\ \text{linearly interpolated}, & \text{elsewhere.} \end{cases} \tag{2.14}$$

One such basis function is demonstrated in Figure 2.2. The linear basis provides enough smoothness of the solution on one hand and is simple on the other. In cases with highly oscillatory coefficient a that we consider higher smoothness is usually not necessary, so piecewise linear functions seem to be a good choice.

Having a basis we can represent any function in the space V_h as

$$u = \sum_i z_i \phi_i, \tag{2.15}$$

If we put this expression into our equation (2.4) and use the basis functions as test functions then we will get the system of linear equations

$$a\left(\sum_i z_i \phi_i, \phi_j\right) = \sum_i a(\phi_i, \phi_j) z_i = F(\phi_j). \tag{2.16}$$

If we take this for all j from the set of basis function we will end up with linear system of equations with respect to vector z

$$Az = b, \tag{2.17}$$

where $A = (a)_{i,j} = \sum_i a(\phi_i, \phi_j)$ and $b_j = F(\phi_j)$.

As the support of ϕ_i in case of our finite element method is local then, in the sums forming $(a)_{i,j}$ from (2.16) almost all the terms will be equal to zero

and hence the matrix A form (2.17) will be sparse. As our problem satisfies properties 1-4 described by (2.7), (2.8), (2.10), (2.12) the resulting matrix A is an M-matrix for the chosen basis on nice grids. A is symmetric and positive definite as proved in [Aav], and the linear system can be solved by the conjugate gradient (CG) method. The algorithm of the method and its derivation can be found, for example, in [KC91] or later in this work. CG method sets a number of assumptions on the system but is known to be the most efficient solver when applicable.

2.2 A posteriori error estimates for the finite element method

In this section we first give an overview of the error estimate concept and main classes of them. Then we go through a derivation of a well known *a posteriori* error estimate for the finite element method that will be used later in numerical examples of this work.

2.2.1 Different types of error estimates

Repin in his book [Rep08] gives a good comparison of two main estimate approaches that we will consider here: *a priori* and *a posteriori* error estimates.

If we consider an abstract problem with some data $\mathcal{D}$ we need to determine what the accuracy of an approximate solution v is.

- An *a priori* estimate then tells us if we increase the resolution to some extent, say α times, the error will decrees α^β times.
- An *a posteriori* estimate, in contrast gives a formula of the following kind:

$$\|e\|_V = \|u - v\|_V \leq \overline{\mathfrak{M}}(\mathcal{D}, v), \tag{2.18}$$

  where e stands for error; u is a correct unknown solution to the problem and $\overline{\mathfrak{M}}$ is a majorant that limits the error from above.

If we are able to compute $\overline{\mathfrak{M}}$ exactly it is called an error bound, if it is also dependent on parameter C independent of $\mathcal{D}$ that is not determined - it is an estimate.

Historically only direct, *a priori* estimates were used. They usually show whether the method will converge to a solution, and with what rate of convergence. The classical numerical analysis also studies uniqueness and existence of solution. These studies are very important but are usually not enough, as in many cases we know the problem and some solution, but we do not know how good they are. This question is answered by *a posteriori* error estimates that were developed in the last decades.

2.2.2 Error estimate for the finite element method

The popularity of *a posteriori* error estimates grew during the last decade, however some of the results in this theory are quite well-known. An example of the latter is the energy norm estimate for the finite element method that can be found in the textbooks [Joh87, BS94, EG04, Rep08]. It is also referred to as the explicit residual method or a special case of Céa's theorem. This result, in the way it is described in this work and in the textbooks, is obtained for the classical Poisson equation, for which $a \equiv 1$:

$$a(u^*, v) \equiv \int_\Omega \nabla u^* \cdot \nabla v dx = \int_\Omega f v dx, \quad \forall v \in H_0^1 \equiv V_0. \tag{2.19}$$

where u^* denotes the exact solution.

Let us obtain the aforementioned estimate, following the derivation in [Rep08, Chapter 2]. First observe that the difference between an approximate solution u and the exact solution u^* in equation (2.19) gives us:

$$a(u^* - u, v) = \int_\Omega (fv - \nabla u \cdot \nabla v)\, dx \equiv \mathcal{G}_u(v), \quad \forall v \in V_0, \tag{2.20}$$

where $\mathcal{G}_u(v)$ is a linear functional on V_0 that we will call the error functional. The norm of the functional is defined as

$$\|\mathcal{G}_u\| \equiv \sup_{v \neq 0 \in V_0} \frac{|\mathcal{G}_u(v)|}{\|v\|_a}, \tag{2.21}$$

and is equal to zero, if $u^* = u$; $\|\cdot\|_a$ is the norm induced by the bilinear form $a(\cdot, \cdot)$.

We notice that applying (2.20) to $v = u^* - u$ gives

$$a(u^* - u, u^* - u) = \mathcal{G}_u(u^* - u) \leq \|\mathcal{G}_u\| \, \|u^* - u\|_a, v \in V_0, \tag{2.22}$$

and as $a(u^* - u, u^* - u) \equiv \|u^* - u\|_a^2$, it follows that $\|u^* - u\|_a$ is bounded:

$$\|u^* - u\|_a \leq \|\mathcal{G}_u\| . \tag{2.23}$$

From the definition of $\mathcal{G}_u$ we get an inequality that is opposite to (2.23):

$$|\mathcal{G}_u(v)| = |(u^* - u, v)| \leq \|u^* - u\|_a \|v\|_a , \tag{2.24}$$

and applying the definition of the norm gives

$$\|\mathcal{G}_u(v)\| \leq \|u^* - u\|_a . \tag{2.25}$$

Having two opposite inequalities (2.25) and (2.23) means that $\|u^* - u\|_a = \|\mathcal{G}_u\|$, however it is practically impossible to compute $\|\mathcal{G}_u\|$ directly using its definition (2.21). So to compute the norm of the error functional we need to use some tricks. We introduce a Galerkin approximation of solution for finite dimensional space $V_h \subset V_0$ then for all functions from this subspace the weak form of the equation will be satisfied

$$a(u_h, v_h) = F(v_h), \quad \forall v_h \in V_h \tag{2.26}$$

and the error $u^* - u_h$ will be a-orthogonal to any $v_h \in V_h$. For this choice of space, $\mathcal{G}$ can be presented as

$$\mathcal{G}_{u_h}(v) = a(u^* - u_h, v) = a(u^* - u_h, v - \pi_h v), \tag{2.27}$$

where $\pi_h : V_0 \to V_h$ is an interpolant to the finite dimensional space. For finite element spaces we can take Clément's interpolant and use its properties on each finite element as it is done in [Rep08]. We will finally arrive to the following inequality

$$\|u^* - u_h\|_a^2 \leq |\mathcal{G}_{u_h}|^2 \leq C \sum_{k \in \mathcal{L}} \eta_k^2, \tag{2.28}$$

where η_k represents the error on an individual element k and has the following form

$$\eta_k^2 := \operatorname{diam}(\Omega_k)^2 \|\Delta u_h + f\|_{\Omega_k}^2 + \frac{1}{2} \sum_{l \in \mathcal{L}} |\Gamma_{k,l}| \left\| \left[\frac{\partial u_h}{\partial n_{k,l}} \right]_{\Gamma_{k,l}} \right\|_{\Gamma_{k,l}}^2 . \tag{2.29}$$

Here Ω_k stands for individual finite elements that are numbered by index set $\mathcal{L}$; $\Gamma_{k,l}$ for the boundaries between two elements, diam is diameter of the element, $[\cdot]$ represents jump. The first term of (2.29) represents the error

accumulated by the approximation inside the element, and the second term, that is sometimes called the flow term, is the error on the border between the elements.

This well known estimate illustrates that finite element approximation, actually makes sense and as it is shown in numerical results the estimate can be used for adaptive error control.

2.3 Inexact solvers vs. fast solvers

In this section we give an overview of different approaches to solving partial differential equations on large domains. We discuss both exact and inexact methods and highlight their differences and similarities. In the second part of this section we focus on domain decomposition methods in particular, since they are important for the numerical part of this work.

2.3.1 Overview

Many areas of modern science and industry require solving partial differential equations on large grids. Often the scale that is reasonable for numerical simulation is many times larger than the scale of the involved parameters. In these cases the fine solution is too expensive to be calculated directly but the coarse solution is not good enough to capture complex behavior of the exact solution. This leads to numerical problems with the large number of unknowns. Such problems cannot be solved exactly with modern desktop computers using classical solution methods. However, rapid development of supercomputers during the last decades made this possible and gave a new challenge of making efficient and preferably parallel algorithms that take advantages of their computational power and use this power wisely and efficiently.

In order to solve large problems several approaches have been developed

- upscaling,
- multigrid,
- domain decomposition,
- multiscale methods.

Let us discuss the main idea of each of them.

Upscaling is an inexact approach to problems with multiscale variability of the parameters. It does not give a solution to the original problem but

gives some approximation of associated coarse scale problem. The name of the method is imposed by its principle which is taking the parameters from the fine scale and somehow upscaling them to the coarse scale of the problem. The coarse scale is chosen to fit in a computer memory or to satisfy similar size requirements. There are several way to do the upscaling. One possibility is heuristics rely upon an assumption of periodicity or nice behavior of a parameter that gives a starting point for the mapping of the parameters to coarse scale. The assumptions can be based on physics of the problem or on the homogenization theory. The homogenization theory is based on the principle of continuity of the media and hence possibility of scale separation for the coefficients.

Multigrid is a mathematical approach to large problems. In contrast to upscaling it is aimed to solve large complicated problems and corresponding linear systems of equations rather then simplify them. The multigrid methods are based on properties of iterative solvers. The sharp oscillations of residual are decaying rapidly but long "waves" stay long. The idea of the method is to consider halved grid together with the original; make a mapping of the residual between the grids and do this operation recurrently we obtain a grid with only a few nodes. For many problems this purely algebraic operation of mapping gives outstanding results. More information on multigrid methods can be found in [BHM00]. Multigrid algorithms are used in many modern solvers and probably are the most efficient method for solving large problems. However its efficient implementation requires fine tuning of a number of parameters that are highly dependent on the problem.

As multigrid, domain decomposition is aimed at solving large problems exactly. The idea of domain decomposition methods is to divide the original domain of the problem into smaller sub domains, solve those local subdomain problems, and use the local solutions to reconstruct the solution of the whole problem. Modern algebraic approach to domain decomposition is to use the information obtained from solving local problems to create a preconditioner for original problem. More details on domain decomposition methods will be given in the following section. Multilevel implementations of domain decomposition show linear convergence and are relatively easy to implement, that is why they are chosen as a benchmark in this work.

Finally we will discuss multiscale methods that are relatively new. They form a large class of methods that are becoming popular during recent years. Original concept of the methods was combining known techniques of upscaling with the opposite operation called downscaling that transfers data back to fine scale and hence improves the resulting solution.

A multiscale method is not a completely different approach compare to the methods described above. Relation of special cases of multiscale methods

to homogenization theory is discussed in detail by Nolen et al. in [NPP08] and relationship with non-overlapping domain decomposition can be found in [NB08]. As a conclusion we can say that multiscale methods are some kind of compromise between inexact solvers and fast solvers which can provide a nice control over error if used adaptively. Further, the reader can find description of variational multiscale method and numerical result of adaptive error control.

2.3.2 Domain decomposition

Domain decomposition methods are known to be a very efficient way of solving elliptic equations. They are particularly interesting in this thesis as they can work as a good benchmark for multiscale methods. Firstly, they give the exact Galerkin solution to compare an approximate multiscale solution to. Secondly, they give the time in which it is possible to compute this correct solution. Exceeding this time asymptotically makes no sense for an approximate algorithm.

The detailed description of this class of methods can be found in textbooks such as [SBG96, TW05]. Here we give only a short overview of them. We will discuss a two level overlapping domain decomposition method used as a preconditioner. This method is very important as it is the simplest approach to domain decomposition that provides linear scalability.

To solve the linear system obtained by the finite element (or in general any other discrete) method (2.17) we need to find $A^{-1}b$. Iterative method will converge faster if we are able to find a matrix that approximates A^{-1} and use it as a preconditioner for original system.

Let us consider domain Ω and some triangulation of it as it is done in model problem section. We then introduce a decomposition of it into subdomains Ω_i so that $\Omega = \cup_{i\in\mathcal{I}}\Omega_i$. In the preferable case of matching grids that we will stick to, Ω_i have their borders along triangulation. Tests show that in order to make a domain decomposition method more efficient, intersections between Ω_i should be non-empty but relatively small; nevertheless non-overlapping methods are still possible. It is also known, that for optimal performance of the algorithm number of subdomains should be proportional to the unknowns in each of them.

We want to approximate the solution by a sum of solutions on subdomains. To do so, we need to project the residual onto subdomains and have a way to perform the opposite operation. We introduce so-called restriction operators $R_i : \Omega \to \Omega_i$. For the case of matching grids these operators are permutation matrices $(R_i)_{k,j} = 1$, if node k in original numbering is projected to node j in Ω_i. It is not hard to notice that matrix R_i^T, called the

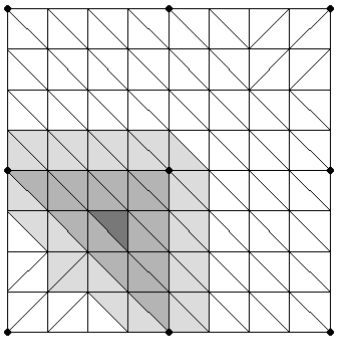

Figure 2.3: Example of conformal grid partition

reconstruction matrix will project elements back from Ω_i to Ω. Then we approximate A^{-1} as

$$\widetilde{A^{-1}} = \sum_{i \in \mathcal{I}}^{*} R_i^T A_i^{-1} R_i, \tag{2.30}$$

where A_i is a matrix that is generated by a discrete method on subdomain Ω_i. Algebraically, A_i can be obtained as $A_i = R_i A R_i^T$. In case of our model problem this is the finite element method on domain Ω_i with zero Dirichlet boundary condition.

In formula (2.30) summation with a star means that we may not want to sum the repeated components. More precisely, if we have a non zero component on position j, k in both $(R_i^T A_i^{-1} R_i)_{jk}$ and $(R_l^T A_l^{-1} R_l)_{jk}$ it is not sensible to sum them, as they correspond to the same point of the actual solution. Instead of summing, it is reasonable to take their average or just neglect one of them. This smart summation is not a necessary condition for the method to work but it often improves the preconditioner and hence the rate of convergence.

The way of approximation of A^{-1} described above, is associated to one level additive Schwarz method. We should note that the inverse matrix is never found or stored explicitly and A^{-1} basically mean solving iteratively associated linear system.

For the case of two level methods additionally to subdomains we introduce so-called coarse domain. We consider another grid and construct a system matrix by the method associated with it. In the simplest case we have conforming grids meaning that the fine grid can be obtained from the coarse grid by a refinement of the coarse grid. In general, this is not always true

and an interested reader can find more information in [SBG96]. An example of a subdomain partition can be found in Figure 2.3; one of the subdomains in the partition is highlighted with gray, the overlap area is marked in light gray, the points of the coarse grid are marked by black dots. In addition to coarse domain we also need to have a way to project functions given in original fine coordinate basis into the coarse basis. There exists a number of possibilities to perform this operation. Here we will discuss only one of them that is similar to projection and reconstruction used in multigrid methods [BHM00].

If the grids are conforming, there is an intuitive way to interpolate a function from the coarse grid onto the fine, that gives a matrix that we will call the reconstruction matrix for coarse domain and denote as R_0^T. As commonly done in multigrid methods to provide the opposite operation a weighted average is used, that is equal to cR_0. In our case, the resulting matrix is used as a preconditioner, and hence the constant multiplier c can be neglected, as it has no influence on the structure of the resulting matrix.

The two level preconditioner for the two-level domain decomposition additive Schwarz method will look like

$$M^{-1} = \sum_{i \in \{0\} \cup \mathcal{I}}^{*} R_i^T A_i^{-1} R_i. \tag{2.31}$$

where A_0 as preciously A_i, is the matrix for the coarse discrete method. The feature of the two level additive Schwarz method is that all the problems $A_i z_i = b_i$ can be solved independently and in parallel which gives great advantage in case of multi-processor systems. As we use solutions to subdomain problems only as a preconditioner, it is not necessary to solve them precisely or to the same accuracy as the original system [SBG96].

The solution of the original system in case of domain decomposition methods can be obtained by Krylov subspace or other iterative methods using M^{-1} as a preconditioner. In Figure 2.4 we give a preconditioned conjugate gradient algorithm that converges for symmetric and positive definite matrices. A matrix obtained from the model problem will satisfy these properties under condition of regularity of grid that can be easily fulfilled [Aav]. In CG algorithm given in Figure 2.4 u^k is an approximation of solution, u^0 is an initial approximation, and M is a preconditioner matrix.

1. Initialize: $r^0 = b - Au^0$

2. Iterate $k = 1, 2, \cdots$ until convergence

Precondition: $z^{k-1} = M^{-1} r^{k-1}$

$$\beta^k = \langle z^{k-1}, r^{k-1} \rangle / \langle z^{k-2}, r^{k-2} \rangle \quad [\beta^1 = 0]$$
$$p^k = z^{k-1} + \beta^k p^{k-1} \quad [p^1 = z^0]$$
$$\alpha^k = \langle z^{k-1}, r^{k-1} \rangle / \langle p^k, Ap^k \rangle$$
$$u^k = u^{k-1} + \alpha^k p^k$$
$$r^k = r^{k-1} - \alpha^k Ap^k$$

Figure 2.4: Preconditioned conjugate gradient method. (Taken from Toselli and Widlund [TW05].)

2.4 Variational multiscale method

In this section we discuss variational multiscale methods, which are the main focus of this work. We give a derivation of the VMS, give formulations described by Larson et al. and Nolen et al. and look at the relation between them. We also describe possible choices of decompositions between the coarse and the fine scale. Finally, we discuss approximations of the fine scale that play an important role for a multiscale method and describe Larson et al. error estimate that gives a clue of how good the approximations are.

2.4.1 Introducing scales

Let us consider the weak formulation of the model problem

$$a(u, v) = (f, v), \quad \forall v \in V, \tag{2.32}$$

where V is some finite-dimensional finite element subspace of the original space $H_0^1(\Omega)$.

In order to formulate a variational multiscale method (VMS) as it is done in many papers such as [HFMQ98, LM05, LM07, NPP08] we first introduce spaces:

- $V_c \subset V$ - the coarse space;
- $V_f \subset V$ - the fine space;

with the requirement that any function in V is uniquely decomposed into these subspaces:

$$V = V_c \oplus V_f. \tag{2.33}$$

As the weak formulation holds for all functions in the original space V it will also hold for both V_c and V_f. If we use the uniqueness of decomposition and represent the solution as $u = u_c + u_f$, and test functions as $v = v_c + v_f$, where $u_c, v_c \in V_c$ and $u_f, v_f \in V_f$, we end up with the following two equations:

$$\begin{aligned} a(u_c, v_c) + a(u_f, v_c) &= (f, v_c), \quad \forall v_c \in V_c, \\ a(u_c, v_f) + a(u_f, v_f) &= (f, v_f), \quad \forall v_f \in V_f. \end{aligned} \tag{2.34}$$

It is easy to notice that the equations are fully coupled and are to be solved together. It is natural that we want to decouple them as this will reduce computational cost. To do so we will use linear properties of the bilinear forms and functionals that enter into the equations (2.34).

2.4.2 Decoupling by introducing coarse scale residual

In this subsection we consider the way of decoupling described by Larson and Målqvist in [LM07]. We introduce a residual function $R(v)$ in a weak sense as follows

$$(R(v), w) = (f, w) - a(v, w), \quad \forall w \in V. \tag{2.35}$$

If we consider the residual of coarse scale, $R(u_c)$ and put it into the fine scale equation from (2.34) we have

$$a(u_f, v_f) = (R(u_c), v_f), \quad \forall v_f \in V_f. \tag{2.36}$$

We substitute $\mathcal{M}R(u_c) \equiv u_f$, obtained by solving (2.36) into the coarse equation from (2.34) and get

$$\begin{aligned} a(u_c, v_c) + a(\mathcal{M}R(u_c), v_c) &= (f, v_c), \quad \forall v_c \in V_c, \\ \mathcal{M}R(u_c) &\equiv u_f. \end{aligned} \tag{2.37}$$

$\mathcal{M}$ is an affine operator and has influence on both the right hand side and the left hand side.

In order to reduce computational cost we would like to decrease the complexity of the fine scale problem. One of the ways to obtain it is the divide and conquer principle, that is to divide the original problem into sub-problems and solve them separately. Following the derivation in [LM07] we introduce a partition of unity $\{\psi_i\}_{i \in \mathcal{I}}$ over the domain Ω. In this case we require that functions ψ_i forming the partition of unity are non-negative and belong to C^∞ inside the elements, but we do not require C^∞ continuity on the whole

domain Ω. We apply functions from the partition of unity to localize the test functions on the right hand side (2.36) and hence decompose it into sub-problems as follows:

$$a(u_{f,i}, v_f) = (R(u_c), \psi_i v_f), \quad \forall v_f \in V_f. \tag{2.38}$$

If the functions from the chosen partition of unity ψ_i have local support we can hope that the solutions for the corresponding problems u_i either have local support or decrease rapidly outside of support of ψ_i; more detailed discussion can be found later in this chapter and in numerical experiments.

We obtain the fine scale solution as $u_f = \sum_{i \in \mathcal{I}} u_{f,i}$ and notice that it is the solution to (2.36) by the following computations

$$a(u_f, v_f) = a\left(\sum_{i \in \mathcal{I}} u_{f,i}, v_f\right) = \sum_{i \in \mathcal{I}} (R(u_c), \psi_i v_f) = (R(u_c), v_f). \tag{2.39}$$

2.4.3 Decoupling by introducing basis corrections

In this section we will describe a way to decouple the multiscale problem given in (2.34) as it is suggested by Nolen et al. in [NPP08].

First of all, we will treat the coarse scale residual given in (2.35) as two separate terms: one coming from the remainder of right hand side that goes to the fine scale and the other from the coarse solution. Then we rewrite the fine scale equation from (2.34) and get two equations instead:

$$a(u_{f,0}, v_f) = (f, v_f), \quad \forall v_f \in V_f, \tag{2.40}$$

where $u_{f,0}$ is the fine scale part corresponding to the right hand side, and

$$a(\mathcal{M}^0 R(u_c), v_f) = -a(u_c, v_f), \quad \forall v_f \in V_f, \tag{2.41}$$

where $\mathcal{M}^0 R(u_c)$ is the part of fine scale corresponding to the coarse scale error. $\mathcal{M}^0$ here is linear in u_c and later we sometimes use notation $\mathcal{M}^0 u_c$ instead of $\mathcal{M}^0 R(u_c)$ which is valid due to linearity. But in the latter case, $\mathcal{M}^0$ is different and obtained from

$$a(\mathcal{M}^0 u_c, v_f) = -a(u_c, v_f), \quad \forall v_f \in V_f, \tag{2.42}$$

Function $u_{f,0}$ is the part of the solution independent from u_c that can be treated separately and gives no coupling. Nolen et al. claim in [NPP08] that for many practical problems this term can be negligible or of minor importance. The reduced equation for the coarse scale can be written as:

$$a(u_c, v_c) + a(\mathcal{M}^0 R(u_c), v_c) = (f, v_c), \quad \forall v_c \in V_c, \tag{2.43}$$

and the solution is computed as

$$u = u_c + \mathcal{M}^0 R(u_c) + u_{f,0}. \tag{2.44}$$

as $\mathcal{M}^0 R(u_c)$ is linear in u_c we can rewrite the equation for the fine scale as

$$a((I + \mathcal{M}^0)u_c, v_c) = (f, v_c), \ \forall v_c \in V_c, \tag{2.45}$$

where I is the identity operator and $\mathcal{M}^0$ is the linear operator computed from (2.42). We can think of $(I + \mathcal{M}^0)v_c$, where $v_c \in V_c$, as of a new perturbed basis and solve the problem with respect to it. We should notice that the span of this basis describes some subspace of V.

As in the previous case we want to split the fine scale problem (2.42) into sub-problems. In this case we stop on a particular case rather then on some abstract choice of partition of unity. We pick out the corrections that correspond to different basis functions from V_c as follows

$$a(u_{f,i}, v_f) = -a(\phi_i z_i, v_f), \quad \forall v_f \in V_f, \tag{2.46}$$

where ϕ_i is a coarse basis function corresponding to variable z_i. Using this splitting we can easily compute the matrix for the finite element method by introducing a new basis $\widetilde{\phi}_i = \phi_i + z_i^{-1} u_{f,i}$, and putting $\tilde{u}_c \equiv \sum_{i \in \mathcal{I}} \widetilde{\phi}_i z_i$ into (2.45) instead of $(I + \mathcal{M}^0)u_c$.

2.4.4 Comparison of formulations

Here we are interested in comparing the formulations discussed above. The coarse scale equations are the same, so we need to study how different the fine scale equations are.

For the formulation by Larson et al. we put the explicit representation of residual $R(u_c)$ from its definition in (2.35) into the equation (2.38) and get

$$a(u_{f,i}, v_f) = (f, \psi_i v_f) - a(u_c, \psi_i v_f), \quad \forall v_f \in V_f. \tag{2.47}$$

In the formulation by Nolen et al. we have (2.46) and (2.40) that we rewrite here for comparison purposes

$$a(u_{f,i}, v_f) = -a(\phi_i z_i, v_f), \quad \forall v_f \in V_f,$$

and

$$a(u_{f,0}, v_f) = (f, v_f), \quad \forall v_f \in V_f.$$

In the Nolen formulation fine correction is treated separately and can also be decomposed into several problems but so far we leave it alone. We compare the basis correction (2.46) with the second term with (2.47). We notice that choosing partition of unity ψ_i satisfying $\psi_i u_c = \phi_i z_i \equiv u_{c,i}$ will give us

$$a(u_{c,i}, v_f) = a(\psi_i u_c, v_f) = a(u_c, \psi_i v_f), \tag{2.48}$$

due to bi-linearity of $a(,)$. It is not so easy to write out an explicit formula for ψ_i, but it is not needed for any practical purposes.

If we add the right hand side correction from (2.40) to (2.46) we notice that the Nolen et al. formulation, which is nicer for implementation and understanding purposes, is equivalent to formulation used by Larson et al. with a special choice of ψ_i.

2.4.5 Symmetric formulation

The derivation of the variational multiscale method that is given above ends up with (2.45), which is referred to as the non-symmetric formulation. One can also consider a symmetric form of the multiscale method that is considered for example in [NPP08]:

$$a((I + \mathcal{M}^0)u_c + u_{f,0}, (I + \mathcal{M}^0)v_c + u_{f,0}) = (f, (I + \mathcal{M}^0)v_c + u_{f,0}), \quad \forall v_c \in V_c. \tag{2.49}$$

This equation directly follows from the original problem formulation (2.32) and the decomposition of Galerkin solution given by (2.44) and the fact that $(I + \mathcal{M}^0)v_c + u_{f,0}$ lies in space V. In [NPP08] Nolen et al. notice that if we replace $\mathcal{M} = \mathcal{M}^0 + u_{f,0}$ by $\mathcal{M}^0$ in this equation it will still remain valid and will take a form

$$a((I + \mathcal{M}^0)u_c, (I + \mathcal{M}^0)v_c) = (f, (I + \mathcal{M}^0)v_c), \quad \forall v_c \in V_c. \tag{2.50}$$

It can be proved by subtracting fine correction equation (2.40) with test function $u_{f,0}$, namely $a(u_{f,i}, u_{f,i}) = (f, u_{f,i})$; and also noticing that due to independence of $u_{f,i}$ from $v_c \in V_c$,

$$a((I + \mathcal{M})v_c, u_{f,0}) = 0 = a(u_{f,0}, (I + \mathcal{M})v_c)$$

as noticed in [NPP08]. Depending on the requirements imposed by the problem we can either ignore $u_{f,0}$ term or solve the equation for the right hand

side correction (2.40) approximately. Computing the exact solution of (2.40) makes no sense as it is at least as expensive as the original problem.

Solving the multiscale problem in symmetric formulation generally gives a denser matrix on the fine scale, however the resulting matrix is symmetric. This is important since using approximation of fine scale corrections $\mathcal{M}^0$ gives a well-posed problem on the coarse scale [NPP08] which for non-symmetric formulation is not always the case.

2.4.6 Introducing approximations

So far, no approximations have been made, and if we solve the final system in any of the two considered formulations exactly, we will get the best fit solution in the chosen finite element space V. However the complexity of the problem in its exact multiscale formulation is much higher then the complexity of the original problem, and hence solving it exactly makes no sense. Experiments [HW97, NPP08] show that if we solve some of the problems approximately the results may still remain acceptable but the computational time required will be much lower.

Though the partition of unity ψ_i is chosen so that the support of each function is local, the solution of each sub-problem due to its elliptic nature will be, generally non-local. It is rapidly decaying, however, outside of the support of the load function. A reasonable way to make the problem less complex is to solve the local problems on the patches, denoted by ω_i, that should at least contain the support of ψ_i. The size of the patches will be a trade-off between the accuracy of the solution and the computational cost. Example of how the correction decay can be found later in Figures 3.7 or 3.8.

A natural question that one may have is: "How do I choose the size of the patches?"; and it is indeed important. It follows from the minimum principle for elliptic equations (see [Eva98]) that if both the solution and its gradient on the border of the patch come to zero, then the solution will not increase any further and will stay equal to zero outside this patch. Some examples of dependency of error on the patch size as well as an adaptive algorithm to control it can be found in numerical results of this work.

2.4.7 Choice of space decomposition

In the formulation of VMS we mention that there is some space decomposition $V = V_c \oplus V_f$. The properties of the method will be influenced greatly by the choice of it. We will focus on two, probably most natural, choices:

- the hierarchical basis [LM05, LM07],

- the fine space that is H_0^1 orthogonal to the coarse space [NPP08].

By the hierarchical basis we mean that we exclude nodes of the coarse grid, from the fine grid, constraining fine scale functions to be zero at those points.

H_0^1-orthogonal basis can be described as follows: we take $v_c \in V_c$ and $v_f \in V_f$, we require that their inner product in H^1 is zero; $(v_c, v_f)_{H^1} = 0$. This can be obtained, for example, by solving a constrained system using Lagrange multipliers as it is shown in the numerical part of the work.

It is hard to answer which choice of decomposition is better. One can claim that the hierarchical basis is easier to implement end the associated linear system has better properties. In contrast, H_0^1-orthogonal decomposition gives a natural property of fine scale correction of being zero in mean and vanishes for homogeneous problem, but gives a non-symmetric linear system representing saddle point problem. We conclude that the choice remains problem-dependent and experiments are needed to compare effectiveness these two approaches.

2.4.8 A posteriori error estimates for VMS

Larson et al. in [LM07] prove a theorem giving error estimates for variational multiscale method. It states that the error in $a(,)$ norm, equal to $\|u\|_a = \sqrt{a(u,u)}$, will look like

$$\begin{aligned}\|e\|_a^2 \leq C \sum_{i \in \mathcal{L}} \left(\|H\mathcal{R}(u_c)\|_{\omega_i}^2 + \|(a-\bar{a})\nabla u_c\|_{\omega_i}^2 \right) \left\| \frac{1}{\sqrt{a}} \right\|_{L^\infty(\omega_i)}^2 \\ + C \sum_{i \in \mathcal{F}} \left(\left\| \sqrt{H}\Sigma(u_{f,i}) \right\|_{\partial\omega_i}^2 + \|h\mathcal{R}_i(u_{f,i})\|_{\omega_i}^2 \right) \left\| \frac{1}{\sqrt{a}} \right\|_{L^\infty(\omega_i)}^2 \\ + C \sum_{i \in \mathcal{F}} \left(\|(a-\bar{a})\nabla u_c\|_{\omega_i}^2 + \|(a-\bar{a})\nabla u_{f,i}\|_{\omega_i}^2 \right) \left\| \frac{1}{\sqrt{a}} \right\|_{L^\infty(\omega_i)}^2 .\end{aligned} \tag{2.51}$$

Let us describe all notation that appears in the error estimate (2.51) and discuss the meaning of each term.

H and h are naturally the typical grid resolution for the coarse and the fine grid respectively. In the general case, the parameter a that is associated with permeability lies in $L^\infty(\Omega)$. As in many formulas we need to compute L^2 norm containing a as a multiplier, we would like to approximate permeability with $\bar{a}$ that is piecewise polynomial in Ω and hence square integrable.

$$\mathcal{R}(u) = |f + \nabla \cdot \bar{a}\nabla u| + 1/2 \max_{\partial K \setminus \Gamma} H_k^{-1} \left| [\bar{a}\partial_n u] \right| \tag{2.52}$$

is the residual on the element and on the element's border where discontinuity occurs. This formula is similar to formula (2.29) that appears when we study classical *a posteriori* error estimates for the finite element formulation of the problem.

Now let us discuss the sets that appear as enumeration for coefficients in sums. Say, for our partition of unity ψ_i, index i lies in some set $\mathcal{N}$. We consider direct decomposition of $\mathcal{N}$ into two sets $\mathcal{N} = \mathcal{L} \sqcup \mathcal{F}$, where $\mathcal{L}$ is an index set for which we do not compute fine scale approximations and $\mathcal{F}$ is an index set where we solve the fine scale problem on some sub-domain ω_i.

$\Sigma(u_{f,i})$ is computed by solving the problem

$$(-\Sigma(u_{f,i}), v_f)_{\partial\omega_i} = (R(u_c), \psi_i v_f)_{\omega_i} - a(u_{f,i}, v_f)_{\omega_i}, \quad \forall v_f \in V_f(\overline{\omega_i}). \quad (2.53)$$

In the equation above the problem is solved for the boundary of the considered patch ω_i and test functions v_f are taken from the space $V_f(\overline{\omega_i})$ that is $V_f(\omega_i)$ without the requirement that the function is equal to zero on the border of the patch not coinciding with the boundary of the domain Ω. Formally

$$\overline{\omega_i} = \omega_i \cup (\partial\omega_i \setminus (\partial\Omega \cap \partial\omega_i)) .$$

Let us shortly stop on the meaning of each term (sum of terms) of the estimate (2.51). The first sum contains $\|H\mathcal{R}(u_c)\|^2_{\omega_i}$ that correspond to the error caused by not solving the fine scale equations for $i \in \mathcal{L}$ and the term associated with corresponding approximation of permeability a. The second sum shows the main part of the error coming from the approximate solution of fine problems. Its second term gives unavoidable error of Galerkin solution on the fine mesh, or, in other words, approximation of replacing the original problem by a finite dimensional one. The first term of the second sum is associated with the remaining flux through the border of the patch ω_i and corresponds to the error from solving the sub-problems locally. And the third sum gives the error from approximation of permeability for $i \in \mathcal{L}$.

In this work we mainly consider a bit simplified version of the estimate. We will assume that permeability a is given as a piecewise polynomial functions which, is true for the most of the applications. Then in (2.51) $a \equiv \bar{a}$ and a simplified version of the estimate looks as follows

$$\begin{aligned}\|e\|_a^2 \leq C \sum_{i \in \mathcal{L}} \left(\|H\mathcal{R}(u_c)\|_{\omega_i}^2 \right) \left\| \frac{1}{\sqrt{a}} \right\|_{L^\infty(\omega_i)}^2 \\ + C \sum_{i \in \mathcal{F}} \left(\left\| \sqrt{H}\Sigma(u_{f,i}) \right\|_{\partial\omega_i}^2 + \|h\mathcal{R}_i(u_{f,i})\|_{\omega_i}^2 \right) \left\| \frac{1}{\sqrt{a}} \right\|_{L^\infty(\omega_i)}^2 .\end{aligned} \tag{2.54}$$

As we have noticed above in Section 2.4.4, the formulation of Nolen et al. is equivalent to formulation by Larson et al. with a special choice of partition of unity. This means that the estimate can be used for both formulations. We also notice that in the proof of the estimate in [LM07] the restrictions are set only to the original finite dimensional space and coarse space, but not on space decomposition so we conclude that the estimate (2.51) and its simplified version (2.54) are applicable to a sufficient class of multiscale methods. In the numerical results chapter we have examples of its practical applications.

Chapter 3

Numerical results

In this chapter we present various numerical examples of the methods that are described in this work. In the first part we present an adaptive refinement for finite elements. In the rest of the chapter we focus on variational multiscale methods: we formulate a numerical model, argue and give examples of typical space decompositions and discuss convergence and performance of VMS with and without adaptive overlap control.

3.1 Examples of adaptive refinement

In this section we discuss an example of using the estimates for the finite element method that are discussed in section 2.2.2 for creating an adaptive refinement algorithm.

For this section we will consider a test problem that is a special case of the model problem for this work. The classical estimate derived for the finite element method in Section 2.2.2 is given for the Poisson equation for which $a \equiv 1$ and in this section we stick to it. As an example we will take a problem given on an L-shaped domain with the right hand side equal to one everywhere

$$a(u, v) = (1, v), \quad \forall v \in V = H_0^1(\Omega), \tag{3.1}$$

where Ω is L-shaped domain as illustrated by Figure 3.1. This problem is interesting as a benchmark for adaptive algorithm, as it has a singularity at the corner $(0.5, 0.5)$. The solution at this point is not in H^2 but only in H^1 [Aav], and we expect that the error for an approximate solution will be greater in this point. As there is only continuity in the first derivative, use of more expensive high-order elements makes no sense and we will use linear elements given by formula (2.14).

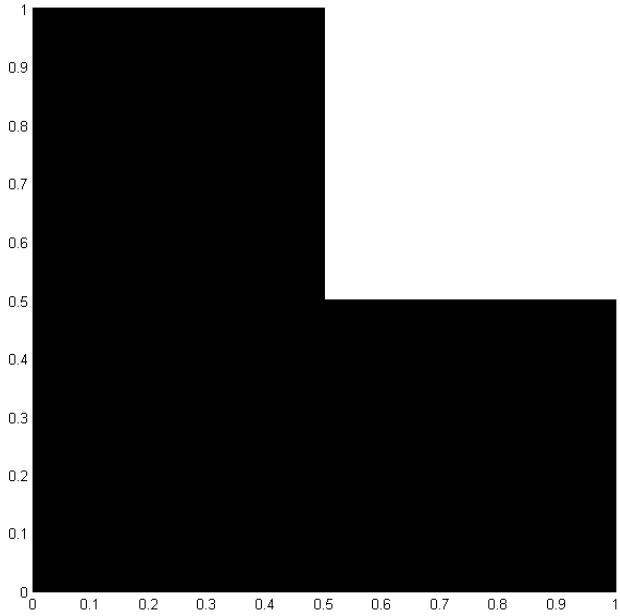

Figure 3.1: L-shaped domain

For the choice of the right-hand side and the elements made in our example, the estimate for the finite element method (2.28), (2.29) is slightly simplified and can be written as

$$\|u^* - u_h\|_a^2 \leq C \sum_{k \in \mathcal{L}} \eta_k^2, \tag{3.2}$$

$$\eta_k^2 := C\mathrm{diam}(\Omega_k)^4 + \frac{1}{2} \sum_{l=\mathcal{L}} |\Gamma_{k,l}| \left\| \left[\frac{\partial u_h}{\partial n_{k,l}} \right]_{\Gamma_{k,l}} \right\|_{\Gamma_{k,l}}^2 . \tag{3.3}$$

Since the diameter of an individual element Ω_k is small we can, in practice, neglect the fourth-order term $C\mathrm{diam}(\Omega_k)^4$.

To make an adaptive refinement algorithm one can use the following argument: as the estimate (3.2) is split into terms corresponding to one element each, it makes sense to refine elements for which the error is greater.

Knowing what element to refine one still needs a strategy of how to perform this operation. A common way of doing it on a triangular grid is to split triangles for which the error is greater than some threshold, in four parts as it is shown in Figure 3.2 in the center. After doing that, use splitting into two or three parts to remove vertices inside edges, as it is shown in Figure 3.2 in triangles adjusted to the central. We choose splitting in two or

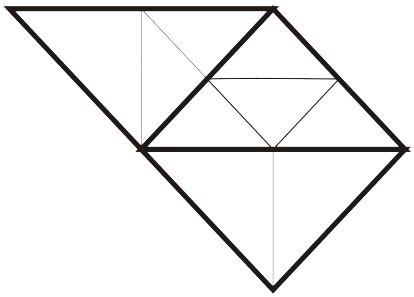

Figure 3.2: Splitting triangles in four, two or three parts keeping their form.

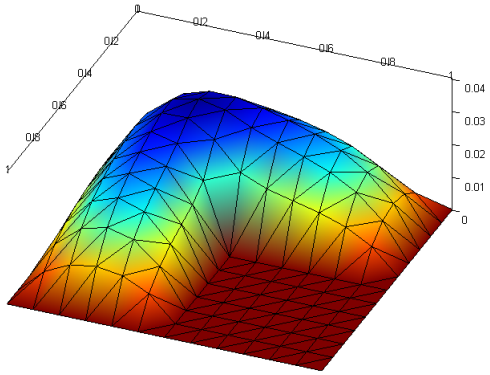

Figure 3.3: Solution on a simple grid

three parts so that the edges proportion is preserved and the resulting grid remains well-shaped.

Now let us consider some simple results of numerical simulations. In Figure 3.3 we can see a solution on a simple coaxial triangular 12x12 grid, and the structure of the grid itself. As it was mentioned above, we expect that the biggest error occurs near the mid-point. If we then take 6x6 and 12x12 grids and run several steps of refinement we will get Figure 3.4. We notice that only few new vertices (and hence degrees of freedom) are introduced. However, as we can see further, the solution is improved greatly.

We now consider the point error that is computed with comparison to the solution on a very fine grid in several points of the L-shaped domain, see Figure 3.5. We notice that linear finite element method with refinement for points closer to the singularity (plotted on the right) not only overtakes the non-refined method, but performs almost as good as more computationally expensive FEM with quadratic elements. Away from singularity, the results for the refined grid are still much better than the results for non-refined grid,

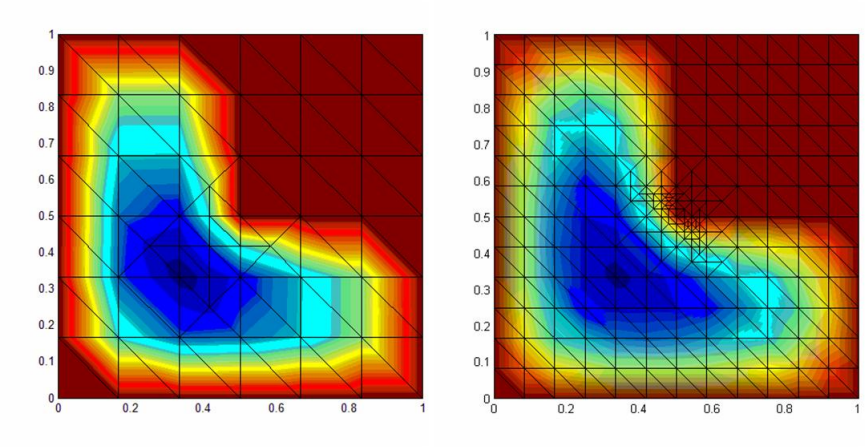

Figure 3.4: Grid and solution after several steps of refinement (6x6 in the left and 12x12 in the right)

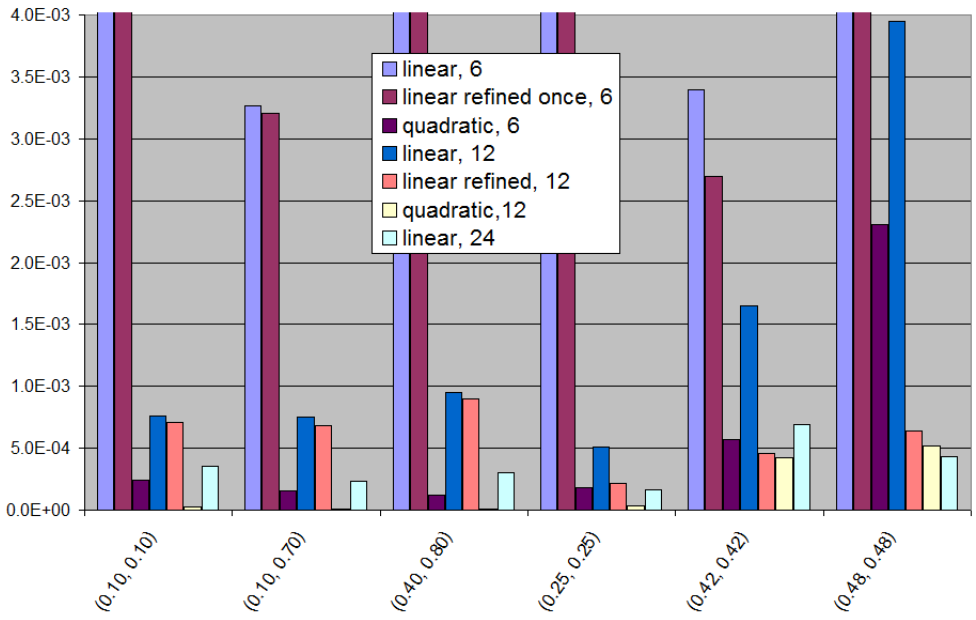

Figure 3.5: Error compared to the fine solution computed for different methods at several points

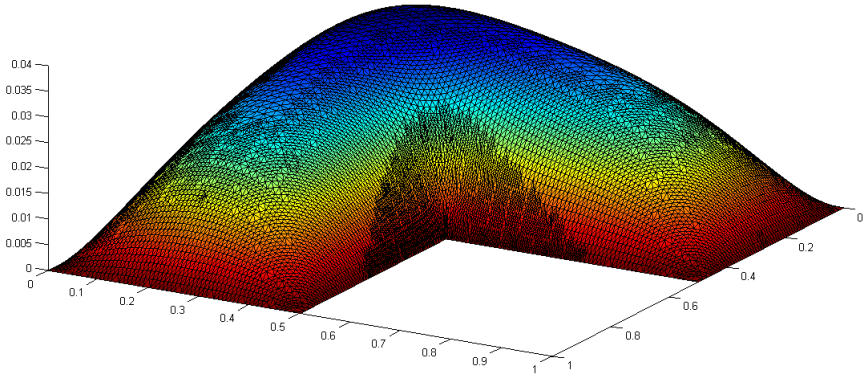

Figure 3.6: Solution on a very fine refined grid

though they are no longer the match for smother quadratic methods. Finally we look at a very fine solution on a grid built adaptively, see Figure 3.6; we notice that the smallest sells are located around point of singularity, confirming the theory.

3.2 Performance of variational multiscale method

In this section we describe some of numerical examples obtained for one of possible formulations of variational multiscale algorithms from different aspects using domain decomposition as a benchmark. We start with formulating the problem for numerical simulation. Then we briefly stop on comparison of the behavior of fine scale corrections. And finally we give examples of convergence and performance of the VMS in its non-adaptive and adaptive formulations.

3.2.1 Model problem

The formulation that we focus on is a symmetric two level method with decoupling of spaces by introducing basis corrections, as it is described in Section 2.4.3. We use piecewise-linear hat functions (2.14) as the finite element basis on the coarse scale and H_0^1 orthogonal piecewise linear functions for the fine scale corrections to both: coarse basis functions, and the right hand side. Finally, the formulation looks as follows:

Formulation 2 *Find the solution* $u = (\mathcal{I} + \mathcal{M}^0)u_c + u_{f,0}$ *by solving*

$$a((I + \mathcal{M}^0)u_c, (I + \mathcal{M}^0)v_c) = (f, (I + \mathcal{M}^0)v_c), \quad \forall v_c \in V_c; \tag{3.4}$$

where parts of the correction $\mathcal{M}^0 u_{c,i} = \mathcal{M}^0 \phi_i z_i$ *are found from*

$$\begin{cases} a(\mathcal{M}^0\phi_i, v) + \sum\limits_{j\in\mathcal{I}} \lambda_j(\nabla\phi_j, \nabla v) = -a(\phi_i, v), \\ (\nabla\phi_j, \nabla v) = 0, \quad j \in \mathcal{I}, \end{cases} \quad \forall v \in V_h, i \in \mathcal{I}, \tag{3.5}$$

where V_h *is the fine solution space and* $\mathcal{I}$ *is the enumeration set for the coarse scale basis* ϕ_i*; and* $u_{f,0}$ *from*

$$\begin{cases} a(u_{f,0}, v) + \sum\limits_{j\in\mathcal{I}} \lambda_j(\nabla\phi_j, \nabla v) = (f, v), \\ (\nabla\phi_j, \nabla v) = 0, \quad j \in \mathcal{I}, \end{cases} \quad \forall v \in V_h. \tag{3.6}$$

The problems in (3.5) and (3.6) are constrained problems, and λ_j are the Lagrange multipliers [Bra07]. To force fine scale corrections to be H_1^0-orthogonal to the coarse scale we instead solve a problem on fine scale with introduced constrains on the test function v, $(\nabla\phi_j, \nabla v) = 0$ which correspond to H_1^0 semi-norm. It is known [Eva98] that $(\nabla\phi_j, \nabla v) = 0$ is equivalent to $(\phi_j, v)_{H_0^1} = 0$, which denotes H_0^1 inner product.

We take the symmetric formulation as it gives a symmetric and better conditioned matrix. And, as it is noticed in [NPP08], the symmetric formulation is well posed even if $\mathcal{M}^0$ is computed approximately. We choose this way of space decomposition as it has zero average which is natural from homogenization and there is no known results of adaptive algorithms for them.

3.2.2 Fine scale corrections for different types of space decomposition

As it was discussed before in Section 2.4.7, the choice of decomposition between the coarse and the fine space is crucial for variational multiscale methods. In this section we will present two computational examples that correspond to different choices of decomposition.

Let us first introduce the notion of patch radius. We make an assumption that the partition of unity ψ_i introduced in (2.38) is constructed so that for each i it lies within one triangle from point j. In case of the formulation in

the form of basis corrections, i will always be equal to j and the assumption will be automatically fulfilled for the linear basis that we consider in the model problem (2.14).

Definition 2 *We give a recurrent definition:*

1 *A subdomain is called a patch with radius* 1 *around point* j *if it consists of triangles adjusted to point* j*.*

i *A subdomain is called a patch with radius* i *around point* j *if it is constructed as the intersection of the patch with radius* $i-1$ *around point* j *and all triangles around vertexes that lie inside that patch of radius* $i-1$*.*

Under the assumptions that we have made, the right hand side for the localized fine scale equation (3.5) will be zero outside the patch of size greater then 1 around corresponding point.

Let us consider two examples of localized solutions $u_{f,i}$ solved on different size of patches, see Figures 3.8 and 3.7. We notice that for both choices of space decomposition for the patch radius equal to one, we get steep gradient on border of the patch which is an unpleasant effect meaning that the approximation is very inaccurate, so for oscillatory a in the model problem the patch size should be at least of size 2, which we notice later applying adaptive algorithm.

As to differences of space decompositions, we notice that in case of H_0^1 orthogonal basis as shown in Figure 3.7 we, as expected, get zero average in contrast to the other choice. In case of the hierarchical basis, the fine scale correction has an overshoot near the border of a patch of radius one, see Figure 3.8. The author's experience as well as some early publications on VMS show that the overshoot causes so-called resonance behavior that ruins even coarse scale solutions.

As it was mentioned in the numerical model formulation, our choice of space decomposition is the one implied by H_1^0 orthogonality. The drawback of it is that we have a more complicated saddle point problem (3.5) on the fine scale. However it gives a great advantage as well. The second set of equations from (3.5) implies that for homogeneous parts of the domain the correction is always zero, as for this case the right-hand side

$$-a(u_{c,i}, v) = -C(\nabla u_{c,i}, \nabla v) = 0,$$

which follows directly from one of constrains. So, for i in homogeneous regions we may not solve the problem (3.5) at all.

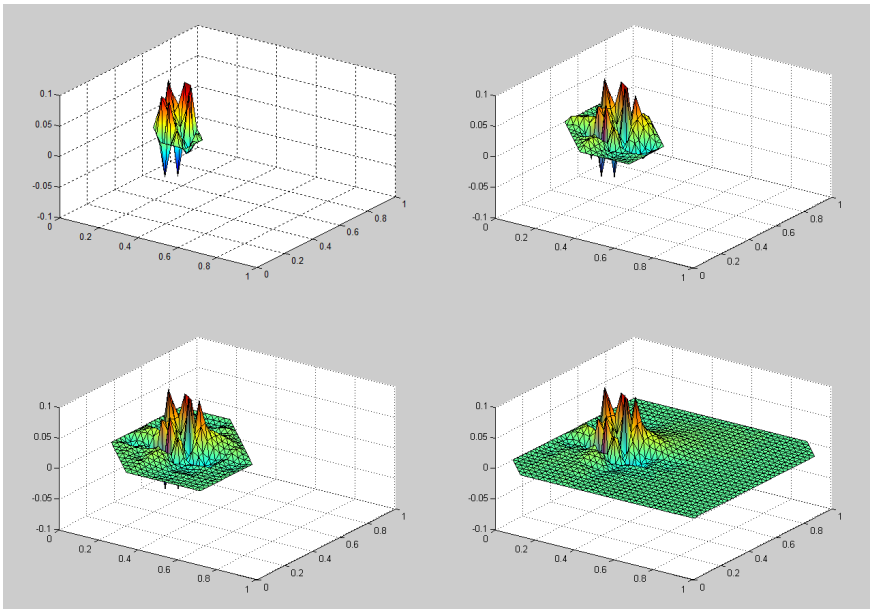

Figure 3.7: A typical localized solution $u_{f,i}$ for fine scale correction with H_0^1 orthogonality on patches with radius 1, 2, 3 and equal to entire domain.

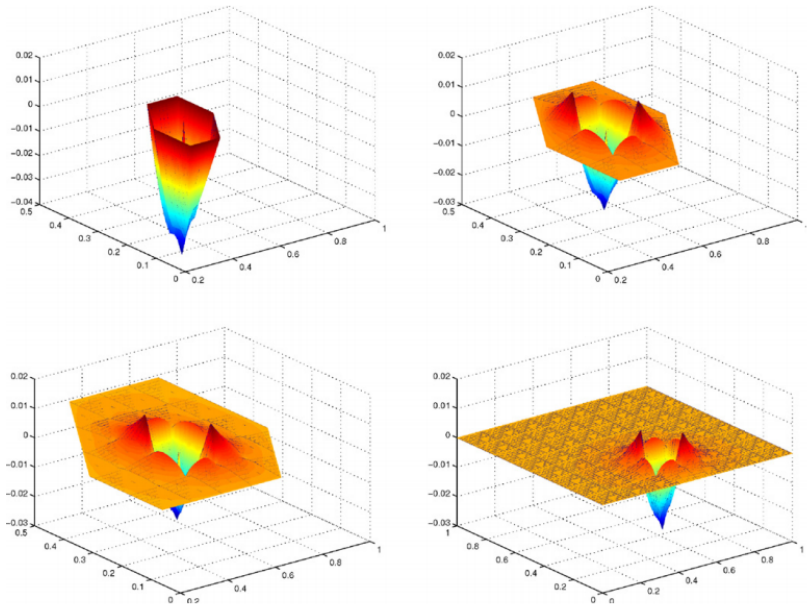

Figure 3.8: A typical localized solution $u_{f,i}$ for fine scale correction with hierarchical orthogonality on patches with radius 1, 2, 3 and equal to entire domain. (Taken from Larson and Målqvist [LM07].)

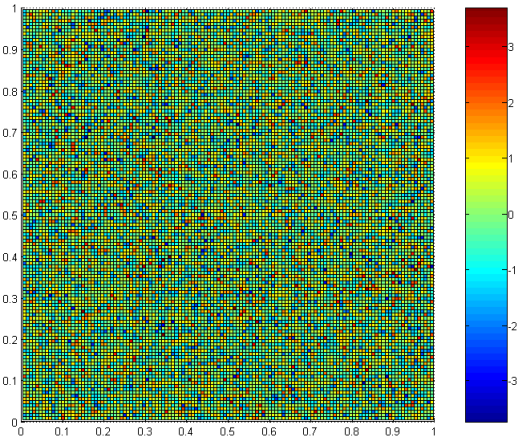

Figure 3.9: Logarithm of piecewise constant random permeability field.

3.2.3 Convergence and performance of the variational multiscale method

In this section we look at examples of how fast VMS converges and how accurate it is for different patch sizes. The results are obtained from solving the model problem presented in Section 3.2.1 and are compared to the solution obtained by two-level domain decomposition method from Section 2.3.2 with linear hat functions as bases on both the coarse and the fine scales.

Let us consider choices of parameters used for the following tests. We take a square domain of size one. As permeability a in the model equation we take highly oscillatory piecewise constant random field such as one shown in Figure 3.9. As the right-hand side of the equation, we take the Dirac δ-function at point $(0.5, 0.5)$ that in case of reservoir interpretation of the problem represents a point injection well. For these choices, Galerkin solution looks as shown in Figure 3.10.

As mentioned above, we focus on the formulation by Nolen et al., for which $z_i^{-1}u_{f,i}$ is the correction for the basis. We also need to remember about the correction for the right hand side u_{f_0} that cannot be neglected in general, but in many cases can be approximated. For the considered case the right hand side correction is presented in Figure 3.11 and, as one can notice, it is far from being zero but is mainly localized around the injection point. As this work does not focus on approximating $u_{f,0}$ we will assume we have it *a priori*. We will not include the computation of it as being expensive but will add it to solution $u_{multiscale}$ when comparing it to the Galerkin solution

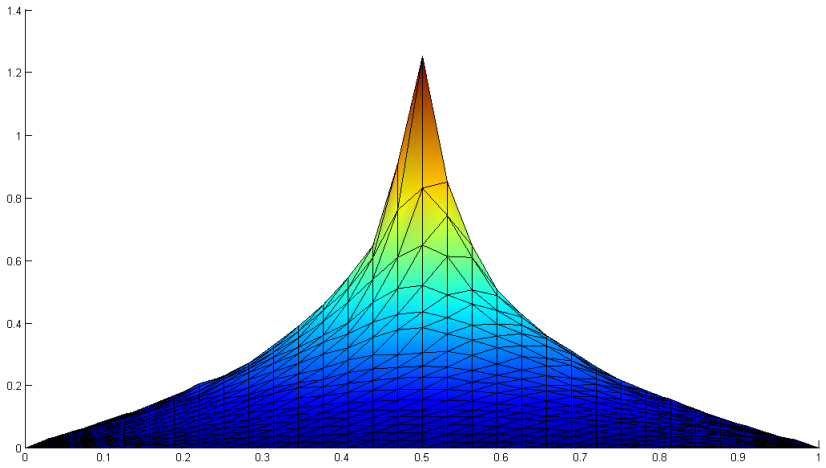

Figure 3.10: The solution for test problem.

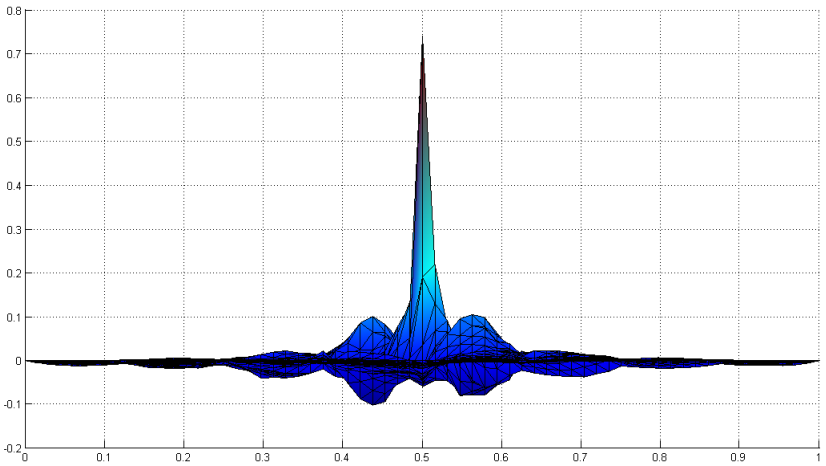

Figure 3.11: Correction for the right-hand side with a point well.

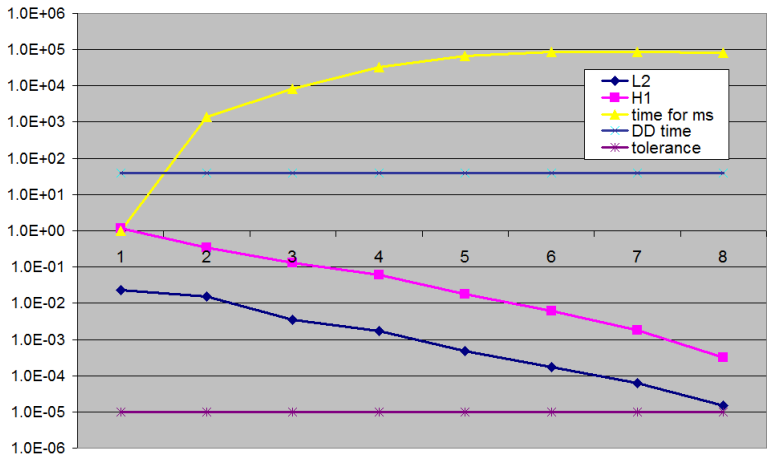

Figure 3.12: Convergence of the variational multiscale method to the Galerkin solution with increase of patch diameter and time spent for solving the problem, ms.

u^*. This assumption is very important as we need to filter out results for the method being tested.

Theory implies that if we make no approximation such as localizing basis functions' corrections on patches or, in other words, take those patches equal to the whole domain Ω, we should still get the exact Galerkin solution. We also hope that the multiscale solution will converge monotonically to exact solution with increase of patch radius.

Let us take a look at how the method converges (see Figure 3.12). We notice that the error in both L_2 and H^1 decreases monotonically and considerably fast with increase of patch diameter. The diameter 8 corresponds to the discrete resolution of the coarse grid. We can also notice that the error decreases in L_2 up to residual tolerance, meaning that we actually get the exact solution up to a computational error, which implies that the multiscale method works correctly.

Also in Figure 3.12 we see the time needed for variational multiscale method to compute the solution, and there is a similar plot of the time for domain decomposition given for comparison. We notice that domain decomposition for patch diameter greater than one is faster then VMS. The reason why we consider elliptically dominant time dependent problems in introduction, is that for purely elliptic problems exact algorithms such as domain decomposition perform better. The most of the time multiscale method spends to compute corrections (3.5), but the time for solving coarse problem with perturbed basis (3.4) for this problem is not greater then a millisecond

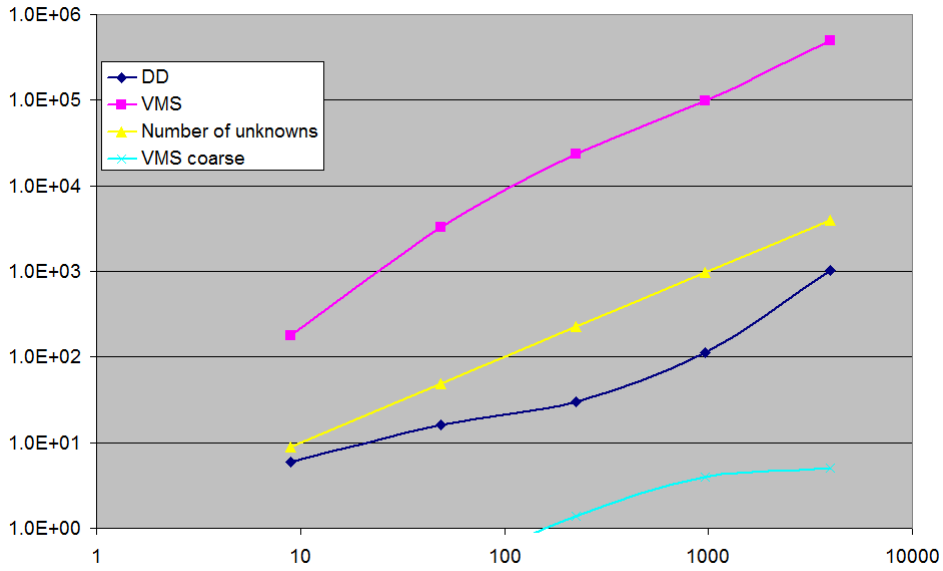

Figure 3.13: Time performance of the methods, ms.

and is hard to compare. For small sizes of the problem time for the coarse solution of VMS is almost equal to the time needed for the classical finite element solution of the coarse problem, though the matrix is less sparse. So, for problems with time dependency, where we can keep the corrections as in the parabolic case or recompute only some of them as in case of coupling with transport, a multiscale method can gain advantage over classical fast solvers.

Another important aspect of performance is scalability. So, let us consider how the VMS scales with increase of unknowns on the fine scale and compare it to the result of domain decomposition. A correction on patches of different sizes shown in Figure 3.7 studied by an "eye norm"implies that for patch sizes greater then 2 the correction is small. So we will compare the speed of domain decomposition to VMS with patch radius set to 3. The plot can be found in Figure 3.13. We can again see that the domain decomposition being an exact algorithm is faster then the variational multiscale, but we need to remember that the most of the time to solve multiscale formulation is spent on the phase of precalculation. This means that in time dependent elliptically driven problems on each time step we only need to resolve for the coarse scale, which is very cheap. And in Figure 3.13 we see that coarse solution of VMS is many times faster then the domain decomposition and if the number of steps, for which we can keep corrections is sufficiently large the multiscale method will gain advantage over an exact solver.

Together with plots of performance of the domain decomposition and the multiscale algorithm in Figure 3.13 we see plot of the number of unknowns. It is done to indicate linear scalability on logarithmic plot. We notice that all three curves are almost parallel to this line, which means that both algorithms have linear scalability.

Also we should notice that for the simulations described in this work no preconditioning is used for acceleration of variational multiscale methods. So in industrial applications if the preconditioner is applied the computational time can be seriously reduced. There exist effective domain decomposition methods for preconditioning saddle point problems involved in computation of corrections, but their implementation can be quite complicated and they are not studied in this particular work.

3.2.4 Examples of adaptive overlap control

As we have discussed above we have a Larson et al. *a posteriori* estimate (2.51) for variational multiscale methods which is applicable to the problem formulation that we are using. In this section we will discuss numerical results for applying its simplified version (2.54):

$$\begin{aligned}\|e\|_a^2 \leq C \sum_{i\in\mathcal{L}} \left(\|H\mathcal{R}(u_c)\|_{\omega_i}^2\right) \left\|\frac{1}{\sqrt{a}}\right\|_{L^\infty(\omega_i)}^2 \\ + C \sum_{i\in\mathcal{F}} \left(\left\|\sqrt{H}\Sigma(u_{f,i})\right\|_{\partial\omega_i}^2 + \|h\mathcal{R}_i(u_{f,i})\|_{\omega_i}^2\right) \left\|\frac{1}{\sqrt{a}}\right\|_{L^\infty(\omega_i)}^2,\end{aligned} \tag{3.7}$$

to control the diameter of the patches for the local correction functions.

In this work we will only consider adaptivity with respect to the size of overlap or, in other words, the patch sizes of the local problems. To construct an error indicator for the algorithm we will use the estimate (3.7). So, let us point out main driving terms for our error indicator.

- For sub-problems that are not solved on the fine scale at all or, equivalently, for $i \in \mathcal{L}$, it is $\|H\mathcal{R}(u_c)\|_{\omega_i}^2$.
- For those, that have a solution for $\mathcal{M}^0\phi_i \equiv z_i^{-1}u_{f,i}$, it is $\left\|\sqrt{H}\Sigma(u_{f,i})\right\|_{\partial\omega_i}^2$.

We also assume that constant C and $\left\|\frac{1}{\sqrt{a}}\right\|_{L^\infty(\omega_i)}^2$ go into the limiting threshold. We need to notice that $\left\|\frac{1}{\sqrt{a}}\right\|_{L^\infty(\omega_i)}^2$, that requires some effort to be

computed, is not the main driving part of the estimate as the heterogeneity influences the solution and comes into $\Sigma(u_{f,i})$ term. So, we neglect it for simplicity.

To compute $\Sigma(u_{f,i})$ we need to solve problem (2.53) that can be rewritten for the basis correction formulation as:

$$(-\Sigma(u_{f,i}), v_f)_{\partial\omega_i} = -a(\phi_i z_i^{-1}, v_f)_{\omega_i} - a(\mathcal{M}^0\phi_i z_i^{-1}, v_f)_{\omega_i}, \quad \forall v_f \in V_f(\overline{\omega_i}),$$

where ω_i is a patch on which the corresponding problem is solved. For our choice of space decomposition solving this problem is not quite straightforward so we solve a saddle point problem instead, as we did in formulation of our numerical model:

$$\begin{cases} (-\Sigma(u_{f,i}), v)_{\partial\omega_i} & + \sum\limits_{j\in\mathcal{I}_i} \lambda_j(\nabla\phi_j, \nabla v) \\ & = -a(\phi_i z_i^{-1}, v)_{\omega_i} - a(u_{f,i}, v)_{\omega_i}, \quad \forall v \in V_h(\overline{\omega_i}), \\ (\nabla\phi_j, \nabla v) = 0, & j \in \mathcal{I}_i \end{cases} \tag{3.8}$$

where $\mathcal{I}_i$ is the set of finite element coarse basis functions that lie inside the patch ω_i. We also notice that the problem in (3.8) is solved only on the boundary of the patch and time to solve it is negligible compare to the problems for $\mathcal{M}^0\phi_i$ on the same patch.

As earlier for adaptive refinement we argue that we can treat the estimate separately for different parts of the domain which we call patches. We start out with no multiscale corrections applied and increase the patches for parts exceeding the threshold.

For testing of our adaptivity technique we used test cases similar to those used by Nordbotten in [Nor08]. For the right hand side of our equation we use two point wells located symmetrically from the center of the unit square domain production and injection. From mathematical point of view they are represented as $-\delta(x_p)$ and $\delta(x_i)$ respectively, where $\delta(x)$ is the Dirac delta. This setup, from point of view of applications, is a model representing a common production pattern.

Let us consider a domain with constant permeability first, where $a \equiv 1$ and the model equation becomes the Poisson equation. For this case the adaptive algorithm even for very small thresholds finishes with the maximum patch size equal to 1 and L^2 error equal to $e_{L^2} \approx 2.3E-6$ which is below residual tolerance of the solver. This goes well with the theory: for the homogeneous case we expected H_1^0 orthogonal corrections to be zero and hence no corrections are needed. The algorithm finds that the flux error Σ on the boundary of any patch is zero and stops after first iteration.

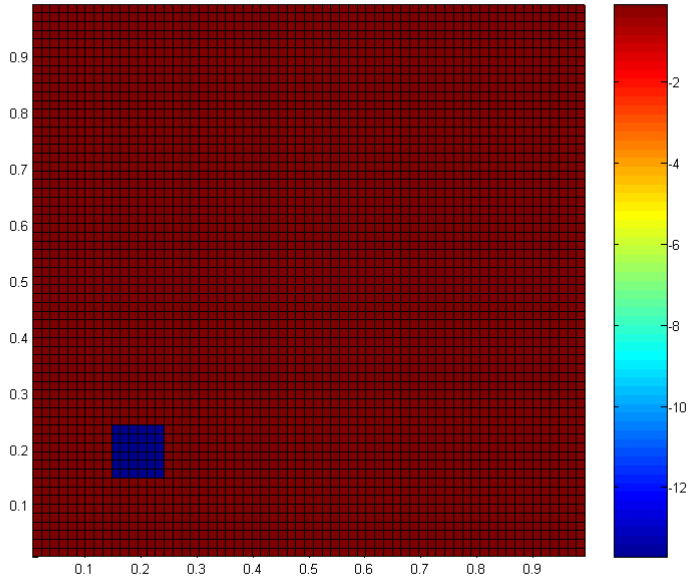

Figure 3.14: Logarithm of a partially homogeneous permeability field.

Another important numerical result is that the error indicator that we are using does not suggest that we stop after the first iteration when subdomain problems are not solved, meaning that some study of constants in the estimate (3.7) is still required.

Now we consider a more complicated case where the parameter a is piecewise constant and is equal to $a_1 \equiv 1$ in all the domain except for one square where it is $a_2 \equiv 1e-4$, see Figure 3.14. We consider the plot of L_2 error against the amount of computational effort needed to computed, where the latter is expressed as fraction of fine scale needed to be solved, see Figure 3.15. The plot shows that the adaptive strategy actually works well for domains with local heterogeneity. With less then 10% of the fine correction being used, it converges up to an accuracy close to numerical errors. However, we also notice that the behavior is not monotone for small amount of overlap. A possible reason is that for a certain choice of the threshold the corrections computed approximately form a wrong system for the coarse problem.

A good illustration of how the adaptive algorithm decides on the choice of patch sizes is shown in Figure 3.16. For corrections far from discontinuity the radius is not greater then 1, and they are almost equal to zero. They are not shown in the figure.

The corrections shown in Figures 3.7 and 3.16 as well as later in Fig-

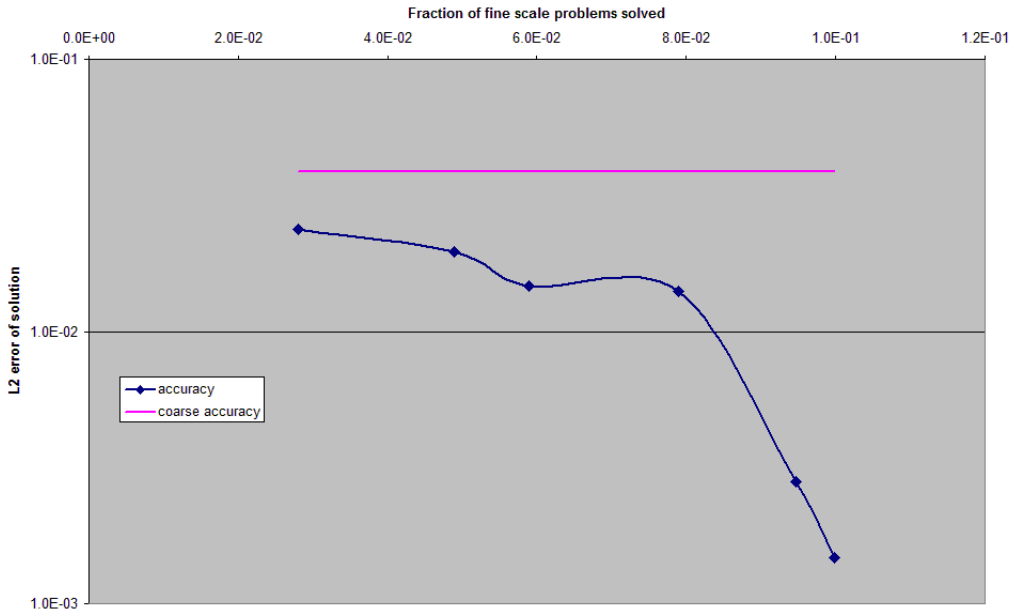

Figure 3.15: L_2 error of the adaptive algorithm depending on computational cost for the domain with local heterogeneity.

ure 3.20 require some comment. They show not correction to the solution as it is done in Figure 3.8 in [LM07] but correction to the basis function as described in the Nolen et al. formulation of the algorithm. The correction $z_i^{-1}u_{f,i}$ that is described in equation (2.46) should be understood as operator $\mathcal{M}^0$ for the basis function ϕ_i. In order to get correction to the solution, one should simply multiply it by the corresponding value of the coarse scale solution z_i, computed from (2.45) or equivalently (2.50).

For highly oscillatory random data as shown in Figure 3.9 and discussed in the previous section for the non-adaptive algorithm, the results of using adaptive algorithm are not so impressive, see Figure 3.17. As expected, the classical uniform refinement strategy gives results similar to adaptive one and in order to have a good VMS approximation of the solution we need to pay big computational cost as radiuses grow proportionally in all the domain. The only advantage of adaptivity in this case is that the patches corresponding to higher values of coarse solution increase first.

Finally we consider a more complicated case of realistically heterogeneous media. As data for parameter a we take a square from layer 1 of well-known SPE10 benchmark, that is shown in Figure 3.18. If we change the threshold value for the adaptive algorithm we get a plot of error depending on computational effort as shown in Figure 3.19. For this complicated test

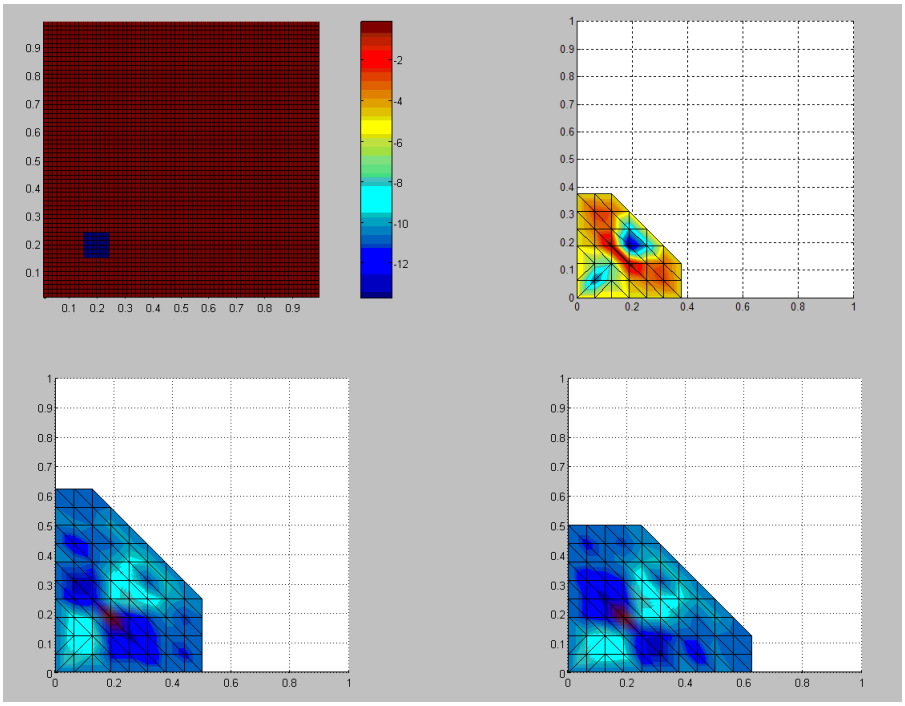

Figure 3.16: The form of some patches achieved by the adaptive algorithm on the inhomogeneous case compared to logarithm of permeability.

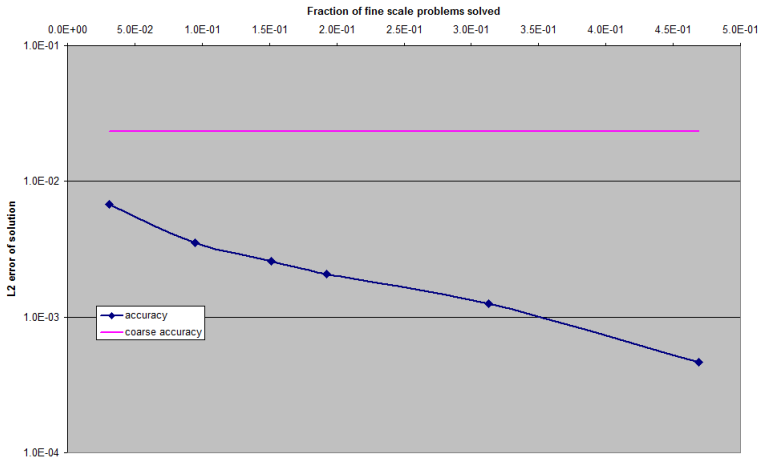

Figure 3.17: L_2 error of the adaptive algorithm depending on computational cost for random permeability field.

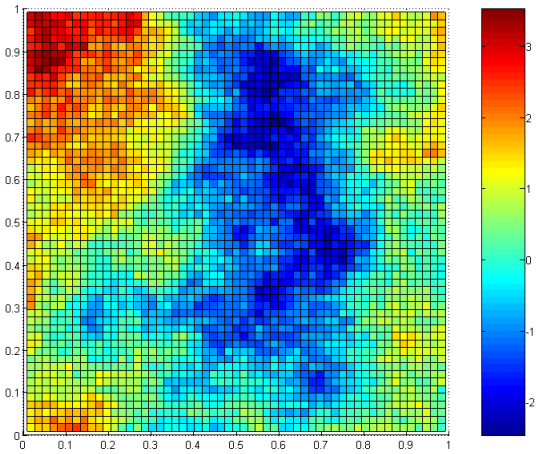

Figure 3.18: Base 10 logarithm of permeability for layer 1 of the SPE 10

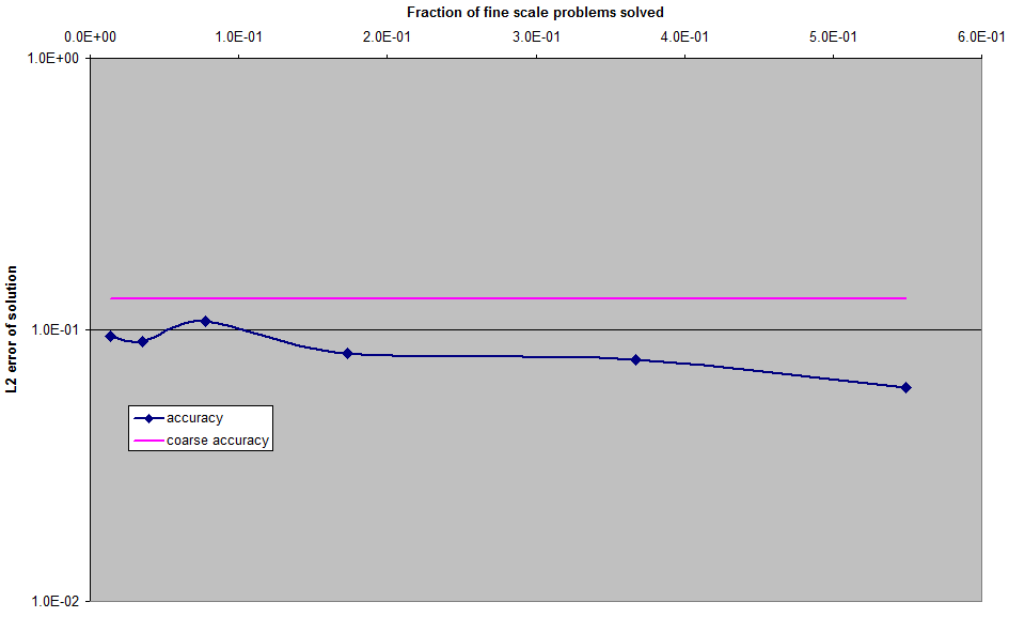

Figure 3.19: L_2 error of the adaptive algorithm depending on computational cost for SPE10 layer 1.

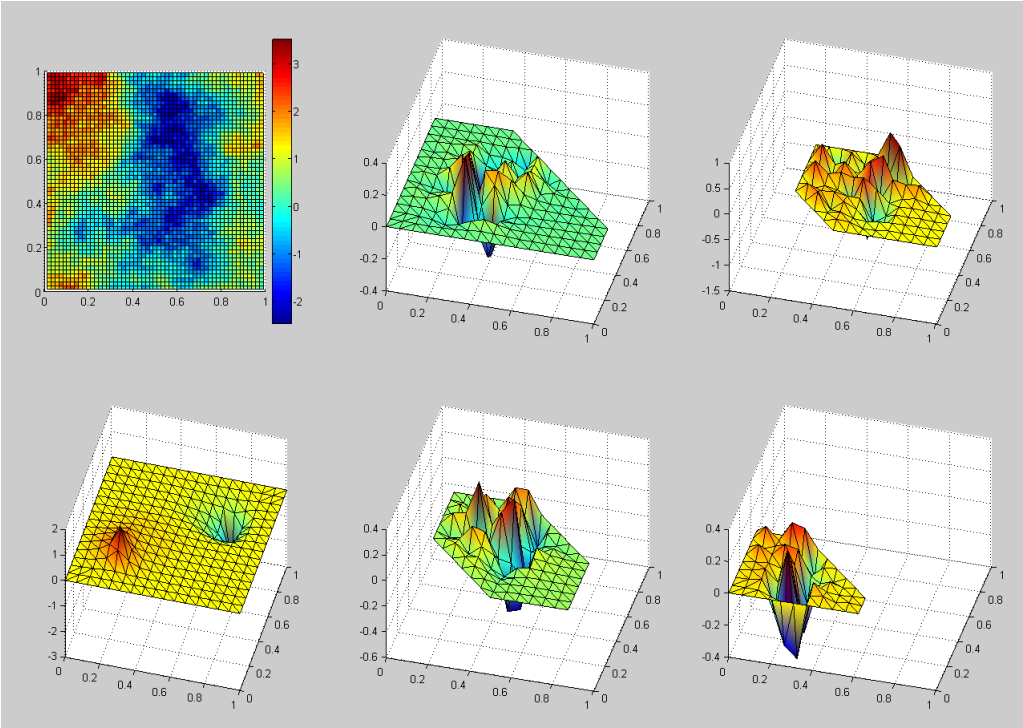

Figure 3.20: The form of some patches achieved by the adaptive algorithm for SPE10 level 1 compared to base 10 logarithm of permeability and to the finite element solution.

we notice that there is a serious improvement of the multiscale algorithm over the coarse solution even for small percentage of the fine scale being solved. However we can notice that convergence to the finite element solution is not monotone for smaller values of the overlap. This problem can be explained by non-compensated corrections as in case with local heterogeneity. As noticed in theoretical section, to compute a true multiscale solution we need to solve fine scale corrections on the whole domain. Sometimes taking one of corrections larger than its neighbors leads to getting extra error in the final multiscale solution.

For 17% of fine scale solved, the corrections on adaptively constructed subdomains look as it is shown in Figure 3.20. We notice that the largest radius of subdomain where corrections are solved corresponds to areas where the oscillations of parameter a are high or close to solution peaks. This choice of the patch sizes is intuitively correct and goes well with theory.

We can conclude that adaptive patch control with the suggested error indicator in many cases gives an improvement over non-adaptive methods.

Chapter 4

Conclusions

In this work we present a class of variational multiscale methods. This class includes two methods of decoupling equations for the fine and the coarse problems:

- by introducing residual (proposed by Larson and Målqvist [LM07]),
- by introducing correction to coarse basis (proposed by Nolen Papanicolaou and Pironneau [NPP08]).

We prove, that the method used by Nolen et al. is equivalent to the special case of the method used by Larson et al. We also consider two different choices for splitting between the fine and the coarse space:

- the hierarchical basis,
- H_0^1-orthogonal spaces.

We adopt an *a posteriori* error estimate for the VMS method that was recently proposed by Larson et al. in [LM07] and show that it is applicable to the whole class of VMS methods discussed in this work.

For the numerical tests we use the Nolen et al. formulation with H_0^1 orthogonality between the fine and the coarse space. We chose this formulation because it is more intuitive for understanding and using. In this formulation they suggest excluding the correction for the right-hand side from the multiscale solution. This is an important assumption for elliptically driven time-dependent problems for which the right-hand side changes on each time step. Computing this correction is expensive and according to [NPP08] is not important to many applications of multiscale methods. For the decomposition between the spaces we use H_0^1 orthogonality because it gives a natural zero

average of the corrections and does not give an overshoot on the border of support of a basis function. This decomposition also gives a trivial solution for the case of homogeneous parameter.

We also demonstrate an implementation of adaptive patch size control based on the error indicator induced by the estimate from [LM07]. Developing an adaptive algorithm for the considered problem is a new contribution of this work. Numerical results show that the adaptive algorithm reduces computational time for many cases. For the considered formulation notably good results are achieved for the parameter fields with local heterogeneities. For these cases adaptive strategy recognizes the patches for which the parameter is constant and keep them minimal, which decreases the total computational time.

However, for relatively big threshold values we notice that its decrease does not imply decrease of the actual error will decrease. In other words, increasing the patches does not always guarantee improvement in the resulting multiscale solution. This issue does not mean that the error estimate does not work, but points out that the estimate is rough. So in order to ensure monotonic behavior of the error we should force the size of the patches to be large.

4.1 Reflections on the method

Now, after discussing advantages of adaptive error control, let us summarize how our numerical results characterize the behavior of the considered VMS method. As we showed in Numerical results the method performs relatively well for several test problems and shows close to linear scalability. It is not efficient enough for purely elliptic problems, since the approximate computation of the corrections takes more time then the Galerkin solution for the fine scale. However, for-time dependent problems this can be understood as a precalculation which is done only once and pays off for a sufficient number of time steps.

It is worth saying that there are numerical simulations, not included in the work, for which the VMS method does not converge to the Galerkin solution for a large number of unknowns. In these problems, distribution of the parameter a has complicated non-local patterns with high jumps (e.g., layer 65 from SPE10, see Figure 4.1). For these problems small numerical errors in fine scale corrections influence considerably the matrix of the coarse scale, which, finally, results in large errors in the multiscale solution. In these cases adaptive patch control does not help, either.

One of possible reasons for the poor performance is the choice of decom-

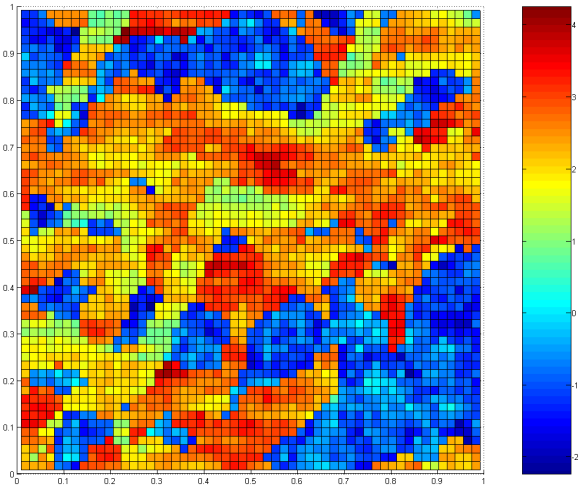

Figure 4.1: Base 10 logarithm of permeability for layer 65 of the SPE 10.

position. In order to find the fine corrections for the H_0^1 decomposition we need to solve a saddle point problem. For this type of problems the associated linear system matrix is quite complicated and can have high dispersion of its eigenvalues. This makes iterative methods slow to converge and also increases the ratio between the solution error and the residual tolerance, which cannot be totally avoided by preconditioning. Composed together, errors in the solutions of the the fine scale effect the coarse scale which in the end, for complicated problems, may ruin the multiscale solution. One of the ways to avoid this is, for instance, to use hierarchical space decomposition. We have not performed proper numerical simulations for this choice of decomposition. Our early implementations show that for small patch sizes the multiscale method with hierarchical decomposition seriously suffer from the resonance effect due to the overshoot, and show poor results even for simple tests.

Another issue of the algorithm is the assumption that the fine scale correction to the right hand side of the model equation is small or easy to approximate, which is not true in general. For instance it is large for point well injection, see Figure 3.11. This additionally limits the applicability of the method.

Finally we need to notice that it is possible that the VMS for finite element formulation is not the best idea for complicated cases. For complicated parameter fields the corrections have a strong non-local influence on the coarse

system, regardless of the choice of decomposition, and hence for obtaining accurate results the corrections cannot be localized. A possible solution to this is to choose a different formulation for the model problem, which we touch briefly in the following section.

4.2 Relation to other works

Let us look at the relation of this work to previous publications.

4.2.1 Adaptivity for variational multiscale methods

We have shown that the error estimate proposed by Larson et al. can be successfully applied to a larger class of methods than was initially considered in in [LM07]. We see that for the H_0^1 orthogonal space decomposition the adaptive strategy works well for many test problems. Due to its zero mean property the algorithm needs no refinement for problems with locally homogeneous parameter in zones with homogeneity.

4.2.2 Behavior of the method for different problems

As noted above we have noticed poor performance of the finite element VMS method for the problems with structured distribution of the parameter with high jumps of discontinuity in it. The numerical tests for the finite element multiscale method were published in several papers including [HW97, LM07, NPP08]. So, let us have a brief comparison of the results. In all of these papers they claim good performance of the method, however the tests performed are quite simple. They use either random or periodic distribution of parameters, for which in our work we also see a good performance of the method. None of these papers use realistic heterogeneous test cases that are both challenging and needed in practical applications.

4.2.3 A different formulation of multiscale method

In conclusion we suggest trying different problem formulations for multiscale method to be applicable for more challenging problems as SPE10. For instance Nordbotten in [Nor08] achieves good results using a problem formulation with explicit formulation for the flux. This suggests that the finite element method is not the best choice as the base for constructing VMS methods for complicated cases.

Bibliography

[Aav] Ivar Aavatsmark. Bevarelsesmetoder for elliptiske differensialligninger. Lecture notes.

[Bea88] Jacob Bear. *Dynamics of fluids in porous media*. Dover Publications, New York, 1988.

[BHM00] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A multigrid tutorial*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2000.

[Bra07] Dietrich Braess. *Finite elements*. Cambridge University Press, Cambridge, third edition, 2007. Theory, fast solvers, and applications in elasticity theory, Translated from the German by Larry L. Schumaker.

[BS94] Susanne C. Brenner and L. Ridgway Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1994.

[CK02] Xiao-Chuan Cai and David E. Keyes. Nonlinearly preconditioned inexact Newton algorithms. *SIAM J. Sci. Comput.*, 24(1):183–200 (electronic), 2002.

[EG04] Alexandre Ern and Jean-Luc Guermond. *Theory and practice of finite elements*, volume 159 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2004.

[Eva98] Lawrence C. Evans. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 1998.

[HFMQ98] Thomas J. R. Hughes, Gonzalo R. Feijóo, Luca Mazzei, and Jean-Baptiste Quincy. The variational multiscale method—a paradigm for computational mechanics. *Comput. Methods Appl. Mech. Engrg.*, 166(1-2):3–24, 1998.

[HW97] Thomas Y. Hou and Xiao-Hui Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.*, 134(1):169–189, 1997.

[Joh87] Claes Johnson. *Numerical solution of partial differential equations by the finite element method*. Cambridge University Press, Cambridge, 1987.

[KC91] David Kincaid and Ward Cheney. *Numerical analysis*. Brooks/Cole Publishing Co., Pacific Grove, CA, 1991. Mathematics of scientific computing.

[LM05] Mats G. Larson and Axel Målqvist. Adaptive variational multiscale methods based on a posteriori error estimation: duality techniques for elliptic problems. In *Multiscale methods in science and engineering*, volume 44 of *Lect. Notes Comput. Sci. Eng.*, pages 181–193. Springer, Berlin, 2005.

[LM07] Mats G. Larson and Axel Målqvist. Adaptive variational multiscale methods based on a posteriori error estimation: energy norm estimates for elliptic problems. *Comput. Methods Appl. Mech. Engrg.*, 196(21-24):2313–2324, 2007.

[NB08] 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.

[Nor] Jan Martin Nordbotten. Variational and heterogeneous multiscale methods for non-linear problems. In *Proceedings of ENUMATH 2009*.

[Nor08] Jan M. Nordbotten. Adaptive variational multiscale methods for multiphase flow in porous media. *Multiscale Model. Simul.*, 7(3):1455–1473, 2008.

[NPP08] James Nolen, George Papanicolaou, and Olivier Pironneau. A framework for adaptive multiscale methods for elliptic problems. *Multiscale Model. Simul.*, 7(1):171–196, 2008.

[Rep08] Sergey Repin. *A posteriori estimates for partial differential equations*, volume 4 of *Radon Series on Computational and Applied Mathematics*. Walter de Gruyter GmbH & Co. KG, Berlin, 2008.

[SBG96] Barry F. Smith, Petter E. Bjørstad, and William D. Gropp. *Domain decomposition*. Cambridge University Press, Cambridge, 1996. Parallel multilevel methods for elliptic partial differential equations.

[TW05] Andrea Toselli and Olof Widlund. *Domain decomposition methods—algorithms and theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2005.