Two Level Additive Schwarz Preconditioner For Control Volume Finite Element Methods

Master of Science Thesis in Applied Mathematics

Kristian Gundersen

Department of Mathematics

University of Bergen

September 22, 2010

Preface

Contents

Chapter 1

Introduction

Domain decomposition methods are flexible methods for solving linear or nonlinear systems of equations arising from the discretization of partial differential equations. In the linear cases which are studied in this thesis, the domain decomposition method can be viewed as a preconditioner for the Krylov subspace accelerators such as the Conjugate gradient or the General minimal residual method. The term domain decomposition has different meanings when the topic is discussed by mathematicians and engineers. In this thesis we refer domain decomposition as a preconditioning method. I.e. the process of subdividing the solution of large linear systems into smaller problems or subdomains where the solutions can be used to produce a preconditioner or solver for the system of equations that results from the discretizing the entire domain.

It is not always satisfying to use domain decomposition as preconditioner alone. With solving only on each subdomain the accuracy of a preconditioner may be poor. To compensate for this, multilevel methods have been introduced . These methods solve the linear system of equations on a larger scale such that the problems of the accuracy on the artificial boundary of each sub-problem is reduced. A domain decomposition method with two levels, i.e. which decompose the problem into smaller sub-problems and solve it on coarse scale is referred as a two level domain decomposition method. The benefits of domain decomposition methods are especially the parallelization properties, simplification on complicated geometry and the fact that convergence the properties are excellent.

In this thesis we will focus on the convergence properties and the implementation of a two level additive Schwarz method and investigate numerically the results provided by *Chou, SH and Huang, J* in the paper, *A domain*

decomposition algorithm for general covolume methods for elliptic problems
[9]. Although the topic of this paper states general control volume methods,
we present a method based on the finite element method and its discretiza-
tion, namely the control volume finite element method.

The control volume finite element method (CVFEM) was a solution of prob-
lems associated with the control volume finite difference method [22] i.e. the
geometric constraints of the finite difference method. With the CVFEM it is
possible to construct control volumes on unstructured mesh that conforms to
an arbitrary shaped domain. The method was invented in the 1960's for solv-
ing electromagnetic field problems. In the early 1980's Bagli and Patankar
[2] linked the method to heat transfer and fluid flow. The CVFEM are some-
times referred as the finite volume method FVM in the literature. The basics
of the CVFEM is that we use the discretization techniques associated with
the finite element method and additionally construct a dual partition or con-
trol volumes around each discretization point. The flux over each edge in the
dual partition is then calculated, stored and used to generate a linear sys-
tem of equations which can be computed together with a source term (right
hand side) and give an approximate solution of the problem in question. In
this thesis we restrict the implementation and numerical experiments to the
steady state diffusion equation on the form,

$$\begin{cases} -\nabla \cdot (A\nabla u) = f & \text{in } \Omega \\ \qquad\qquad u = 0 & \text{on } \partial\Omega, \end{cases}$$

with zero Dirichlet boundary conditions. This is sufficient to test the conver-
gence properties in the two level additive Schwarz framework together with
the CVFEM. Simultaneously an error of the discretization can be estimated.
The diffusion equation appears in many compound equations and the space
discretization of this term is essential when solving such systems numerically.
For the actual space discretization it is convenient to consider both regular
and irregular triangulation of the domain. The benefits with irregular trian-
gulations are that they can tessellate any planar surface and for that reason
is widely used in many discretization applications.

With a heterogeneous diffusion coefficient A the CVFEM generates an un-
symmetric coefficient matrix. There is a variety of methods for solving linear
systems of equations. The general minimal residual method is one of them
and this method is suitable for solving system of linear equations with un-
symmetric structure.

This thesis investigates numerically the convergence properties of the CVFEM

preconditioned with the two level additive Schwarz method for the CVFEM method and compares it to other preconditioners for the GMRES. Domain decomposition on such methods are not common, and to see the numerical results in this context is interesting. There have been some research regarding the domain decomposition capabilities for the mortar finite element method in [11]. In [19] the authors look at the finite volume variational formulation and its application to domain decomposition methods for rectangular grids. However there have been little research regarding the CVFEM capabilities on two level additive Schwarz methods with overlap. Further we will explain terms and conditions such that the work done here can be easily reproduced, understood and tested by others. Now follows an outline of the thesis.

Chapter 2: This chapter is entirely devoted to the CVFEM. In the first part we describe some important concepts such as mesh, region of support and interpolating shape functions. Next we formulate a CVFEM scheme through a model problem, and at the end comment on the convergence of the method.

Chapter 3: This chapter explains the idea behind preconditioning, condition numbers and the GMRES method. In the part about the GMRES method, we explain how the algorithms works as well as give a convergence estimate.

Chapter 4: Here we explain generally about domain decomposition and the Schwarz methods, especially the two level additive Schwarz method. We look at an abstract view of the two level additive Schwarz for use in chapter 6.

Chapter 5: In chapter 5 the implementation aspects of CVFEM, especially the assembly of the coefficient matrix are presented to the best of ability. A full review of how we have implemented the two level additive Schwarz method is included. Some of the important terms such as restriction and interpolation operators are explained in detail.

Chapter 6: We present the analysis of the convergence rate from [9], and state the theorem to be tested numerically in chapter 7.

Chapter 7: We present the numerical results of the implementation of both the CVFEM and two level additive Schwarz method, and also a comparison with other well known preconditioners are done to show the convergence properties.

Chapter 8: Summary of the thesis and conclusion of the work.

Chapter 2

Control Volume Finite Element Method (CVFEM)

In this section we introduce a variation of the finite element method (FEM). The difference from this method is the introduction of a dual mesh, and that we approximate it with the flux over the so called control volumes generated around every node instead of using the original triangulation to calculate the solution. For more about the FEM see [3, 7]. In the following chapter we will study the essence of the control volume finite element method. Basic concepts of the method will be explained. First important definitions and syntax such as mesh, basis, shape functions, region of support, control volume will be explained before we explain the method through a model problem. In later chapters, important aspects of implementation will be discussed. A fundamental part of the CVFEM is the discretization and how this is applied. The first part of this chapter will be dedicated to how we prepare the continuous problem to suit the discrete version of the method. Much of the theory about CVFEM shown in this chapter are inspired from [21, 9, 7, 13].

2.1   Important Concepts Regarding the CVFEM

In this section we will explain some of the important concepts around the CVFEM. For understanding the method it can be convenient to be confident on the different expressions and definitions. Before the method is explained in an abstract manner, we introduce some of the important terms. As defined in [21].

2.1.1   Discretization

With discretization we mean the process which starts out from continuous models and equations and transforms them into discrete variants of the problem. This is often the first step toward making the problems suitable for nummerical experimentation, implementation and evaluation. There are various ways to discretisize a domain.

In this thesis we are going to use a finite element discretization and introduce a dual partion to obtain the CVFEM. A basic approach is using structured mesh where the nodes are located such that each node is uniquely specified by a row and a column vector in a uniform manner. Meshing of triangular elements will be in focus. The benefits of a triangular mesh, especially unstructured triangular mesh, is that it will give geometric flexibility. An unstructured mesh is a mesh which is not regular, and the triangles are not uniform over the domain. In the next section we will see the difference between regular and structured, and irregular and unstructured meshes. An $2-D$ unstructured mesh can tessellate any planar surface. A common choice is to place the nodes in the vertices[3], but it is also possible to let the nodes lie on the midpoint of an edge. We have to be careful not make too acute angles in the triangulation. This is due to the numerical solution obtained is critically dependent on the quality of the mesh[9].

2.1.2 Mesh

A mesh which is built up with only one type of triangular elements is called a regular triangulation or a regular mesh. On the other hand, we call a domain discretizised of various types of triangles, an unstructured mesh or unstructured triangulation. A node is a part in our data structure which we are using to compute the solution. Each triangle consists of three nodes. These nodes are our building block, and the placing of the nodes gives us the mesh.

Structured Mesh
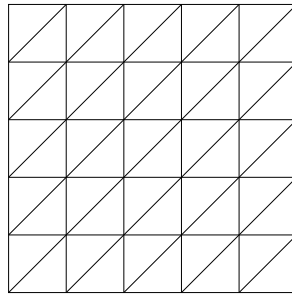
Figure 2.1: Here we see a structured mesh in 2-D with 36 nodes on a domain $0 < x, y < 1$.

Unstructured Mesh
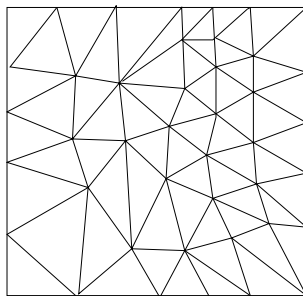
Figure 2.2: Here we see a unstructured mesh in 2-D with 42 nodes on a domain $0 < x, y < 1$.

2.1.3 Region of Support

The region of support is the list of nodes that share a common element with a given node. This region is used to gain and produce the dual partition. In figure 2.3 we see an example of a triangular mesh where the region of support with basis i. Around each node, it is possible through the region of support, to make a closed polygon. This polygonial is used to produce what we call the control volume CV.
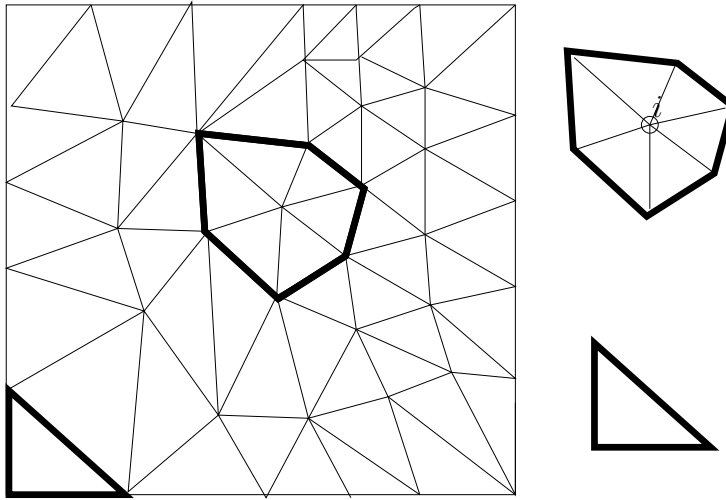
Figure 2.3: Illustration of the region of support.

2.1.4 Control Volume

In the region of support a control volume (CV) is created by joining the centre of each element in the support to the mid-points of the element sides that pass through the nodes. See figure 2.4. In general this creates a closed polygonial CV with $2m$ sides or faces, where m is the number of elements in the support. This can bee seen in figure 2.6. Each CV contributes 1/3 of its area to the CV area if we use the circumcenter of the triangles as base. The volumes from all the nodes tessellates the domain without overlap. See figure 2.6 for vizualation.
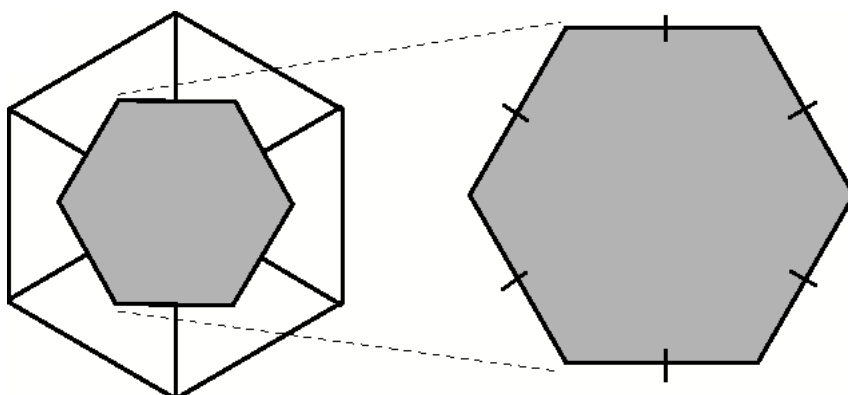
Figure 2.4: Because of the uniform structure of the triangular mesh, this control volume look like it only has m sides.

2.1.5 Interpolation Shape Functions

The building block of the discretization is the triangular element. For linear triangular elements, which are used in this thesis, the node points are placed at the vertices of the triangle [3] and the nodes are labelled $1, 2$ and 3 as in figure (2.5). Values of the dependent variable ϕ, the so called basis functions, are calculated and stored at these node points. An arbitrary point (x, y) within an element can be approximated with linear interpolation

$$\phi_i = ax_i + by_i + c \qquad i = 1, 2, 3 \tag{2.1}$$

It is more convenient to rewrite the basis function in terms of the Shape function N_1, N_2 and N_3 where

$$N_i(x, y) = \begin{cases} 1 & \text{, at node } i \\ 0 & \text{, at all points opposite to node } i \end{cases}$$
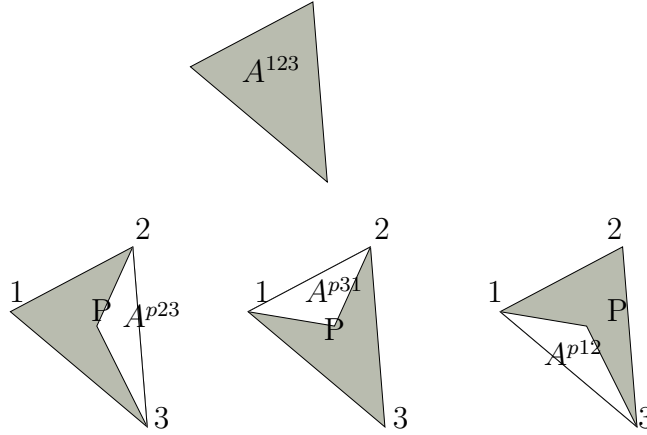
Figure 2.5: Here we see an illustration of how the shape functions are defined. The shape functions are used to calculate the flux over each line segment or face in one element.

$$\sum_{i=1}^{3} N_i(x, y) = 1, \qquad \text{at every point in the element} \qquad (2.2)$$

The continuous unknown field can now be expressed as the linear combination of the values at nodes $i = 1, 2, 3$

$$\phi(x, y) \approx \sum_{i=1}^{3} N_i(x, y)\phi_i \qquad (2.3)$$

A straightforward geometric derivation for the shape functions can be obtained such that

$$A^{123} = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

$$= \frac{1}{2}[(x_2 y_3 - x_3 y_2) - x_1(y_3 - y_2) + y_1(x_3 - x_2)] \qquad (2.4)$$

where A^{ijk} denotes the area of one triangle in one particular element. Similarly we obtain the the sub−elements with vertices at points $(p, 2, 3)$, $(p, 3, 1)$, and $(p, 1, 2)$ where p is an arbitrary and variable point in the element. With these definitions it follows that the shape functions are given by

$$N_1 = \frac{A^{p23}}{A^{123}}, \qquad N_2 = \frac{A^{p31}}{A^{123}}, \qquad N_2 = \frac{A^{p12}}{A^{123}}. \qquad (2.5)$$

For actual implementation it is the derivatives of the shape functions which are of intrest.

$$N_{1x} = \frac{\partial N_1}{\partial x} = \frac{(y_2 - y_3)}{2A^{123}}, \quad N_{1y} = \frac{\partial N_1}{\partial y} = \frac{(x_3 - x_2)}{2A^{123}} \qquad (2.6)$$

$$N_{2x} = \frac{\partial N_2}{\partial x} = \frac{(y_3 - y_1)}{2A^{123}}, \quad N_{2y} = \frac{\partial N_2}{\partial y} = \frac{(x_1 - x_3)}{2A^{123}} \qquad (2.7)$$

$$N_{3x} = \frac{\partial N_3}{\partial x} = \frac{(y_1 - y_2)}{2A^{123}}, \quad N_{3y} = \frac{\partial N_3}{\partial y} = \frac{(x_2 - x_1)}{2A^{123}} \qquad (2.8)$$

2.1.6   Sobolev Spaces

When we characterize a solution of a partial differential equation it is necessary to know what class of functions we are seeking the solution in. It is often possible to show that the solution is the limit of an Cauchy sequence [8]. The space in which we seek the solution has to be complete i.e a Banach space. In this thesis we also require that the metric is induced by an inner product such that the space is a Hilbert space. We will look at integrable functions, and their derivatives. At the same time we will demand that the derivatives are integrable. In this thesis we look at domains $\Omega \subset \mathbb{R}^2$, but the theory developed applies $\mathbb{R}^d$ in general. We denote the closure of Ω with $\overline{\Omega}$. Here u is a scalar real functions, i.e. $u : \Omega \rightarrow \mathbb{R}$.

We start out by defining the space of integrable functions. For every positive p, we define the space $L^p(\Omega)$ which is the quantity of measurable functions u, where

$$L^p(\Omega) = \int_\Omega |u(x)|^p \, dx < \infty. \qquad (2.9)$$

The norm of the space $L^p(\Omega)$ is defined by

$$||u||_{L^p(\Omega)} = \left(\int_\Omega |u(x)|^p \, dx \right)^{1/p}. \qquad (2.10)$$

If $p = \infty$, $L^\infty(\Omega)$ is defined as the quantity of measurable functions u, where

$$\sup_{x \in \Omega} |u(x)| < \infty. \qquad (2.11)$$

The norm of the space $L^\infty(\Omega)$ is given by

$$||u||_{L^\infty} = \sup_{x \in \Omega} |u(x)|. \qquad (2.12)$$

The spaces $L^p(\Omega)$ where $1 \le p \le \infty$ is Banach spaces, while the space $L^2(\Omega)$ is a Hilbert space. An important space is when $p = 2$ and for real functions

the inner product of the space $L^2(\Omega)$ becomes

$$(u, v)_{L^2(\Omega)} = \int_\Omega u(x)v(x) \; dx. \tag{2.13}$$

We also want the integrable functions to be in some sort differentiable. It is not possible to differentiate (2.13) in classical manner for all functions. We have to introduce the term weak derivative. If $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_d)$, then α_j is a d-tuple of a non-negative integer . The term α is an multi index. With the multi index in hand we can use the following notation for the derivative

$$D^\alpha u = \frac{\partial^{|\alpha|} u}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_d^{\alpha_d}}. \tag{2.14}$$

The term $|\alpha|$ is defined as

$$|\alpha| = \sum_{j=1}^d \alpha_j \tag{2.15}$$

With this way of describe the derivative of we can for a arbitrary local integrable function $u \in \Omega$ define the weak derivative with the relation,

$$\forall \phi \in C_0^\infty(\Omega) \quad \int_\Omega D^{|\alpha|} u\phi \; dx = (-1)^{|\alpha|} \int_\Omega u D^\alpha \phi \quad dx, \tag{2.16}$$

We see that if u is differentiable in classical manner, then equation (2.16) is obtained with partial integration since $\phi = 0$ on the boundary $\partial\Omega$. The nice property of the weak derivative is that it gives a unambiguous definition of the derivative of quantities close to zero. With quantities close to zero we mean functions that are almost equal everywhere except zero. The spaces L^p is is not actually the functions u itself, but what we refer as equivalent classes. If the L^p space is equal to the function u over the whole domain Ω except zero we say that they are similar on a quantity measures near zero. With the weak derivative we can define the Sobolev space which is where we want to seek a solution u to the problem (2.31). The Sobolev space is defined as $W^{m,p}(\Omega)$, where m is a non negative integer, and $1 \leq p \leq \infty$. The space of functions $u \in L^p(\Omega)$ where all the weak derivatives of u up order m also lies in $L^p(\Omega)$. The space $W^{m,p}(\Omega)$ is given as

$$W^{m,p}(\Omega) = \{u \in L^p(\Omega) | D^\alpha u \in L^p\Omega \quad \forall \quad |\alpha| \leq m\} \tag{2.17}$$

The norm of the Sobolev space $W^{m,p}(\Omega)$ is for $p \leq \infty$ given as,

$$||u||_{W^{m,p}(\Omega)} = \left(\sum_{|\alpha|=m} ||D^\alpha u||_{L^p(\Omega)}^p \right)^{1/p}. \tag{2.18}$$

For $p = \infty$ the Sobolev norm is given as

$$||u||_{W^{m,p}(\Omega)} = \max_{|\alpha| \leq m} ||D^\alpha u||_{L^\infty(\Omega)} \tag{2.19}$$

The spaces $W^{m,p}(\Omega)$ is Banch spaces. The important space $W^{m,2}(\Omega)$ which are used in this thesis is an Hilbert space, with inner product

$$(u, v)_{W^{m,2}(\Omega)} = \sum_{|\alpha| \leq m} (D^\alpha u, D^\alpha v)_L^2 (\Omega). \tag{2.20}$$

An usual notation for these Hilbert space is the $H^m(\Omega) = W^{m,2}(\Omega)$. We see that $H^0(\Omega)$ then becomes $L^2(\Omega)$.

Before we explain some important subgroups of the traditional Sobolev spaces, it can be convenient to give a theorem that explain the term embedding.

Theorem 1 *Let X and Y be two Banach spaces with norms $||\cdot||_X$ and $||\cdot||_Y$ respectivly. We say that X is embedded in Y, and use the notation $X \hookrightarrow Y$ if the two following conditions are satisfied:*

    i. $X \subset Y$

    ii. For all $x \in X$ there exists a constant c such that $||x||_Y \leq c||x||_X$.

Sometimes it is necessary to look at Sobolev spaces $H^s(\Omega)$ with non integer index s. These kind of Sobolev spaces are important when evaluating the boundary integrals. Let $\Omega \subset \mathbb{R}^d$ and assume that $s > 0$ but constitutes a non-integer. Let m be the integer value of s, i.e $m = s$, and let $\lambda = s - m$. Then $\lambda \ in(0, 1)$. We define the inner product

$$
\begin{aligned}
(u, v)_{H^s(\Omega)} &= \sum_{|\alpha| \leq m} \left[\int_\Omega D^\alpha u(x) D^\alpha v(x) dx \right. \\
&+ \left. \iint_{\Omega \times \Omega} \frac{[D^\alpha u(x) - D^\alpha u(y)][D^\alpha v(x) - D^\alpha v(y)]}{||x - y||_2^{d+2\lambda}} dx dy \right]
\end{aligned} \tag{2.21}
$$

with the norm

$$||u||_{H^s(\Omega)} = \sqrt{(u, u)_{H^s}(\Omega)} \tag{2.22}$$

Too characterize the space $H^s(\Omega)$, we look at

$$X_s = \{u \in C^\infty(\Omega) | ||u||_{H^s(\Omega)} < \infty\} \tag{2.23}$$

with the norm $||\cdot||_{H^s(\Omega)}$. The space $H^s(\Omega)$ is defined as the completion of the space X_s, i.e.

$$H^s(\Omega) = \overline{X_s}. \tag{2.24}$$

Sobolev spaces where s are non-integer is called Sobolev-Slobodeckij spaces. We have that $H^s(\Omega) \hookrightarrow H^m(\Omega)$. We will use $s = 1/2$ when we evaluate boundary integrals in the following.

Next we go on with explaining the important subspace $H_0^m(\Omega)$. We define H_0^m to be the closure in $H^m(\Omega)$ of the infinitely differentiable function compactly supported in Ω. We write

$$H_0^m(\Omega) = \overline{C_0^\infty(\Omega)}. \tag{2.25}$$

Let $\Omega \subset \mathbb{R}^d$ be a limited open set, and assume that the boundary is Lipschitz continuous

$$H^m(\Omega) \hookrightarrow C^0(\overline{\Omega}) \quad \text{for} \quad m > \frac{d}{2}. \tag{2.26}$$

If a function is defined on the domain Ω, then we call the restriction of the same function to the boundary $\partial\Omega$ for the trace to the function u on the boundary. If the trace of a function is well posed, the function has to have a natural expansion to the boundary. It is not enough to demand that the function is continuous or lies in L^2. However, if the function is uniformly continuous on a closed domain $\Omega \subset \mathbb{R}^d$, then we now that function has an unambiguous expansion to the boundary $\partial\Omega$. For a Lipschitz continuous boundary the embedding (2.26) holds, and it is therefore sufficient to demand that the function lies in $H^m(\Omega)$ with $m > d/2$. Although the function has to be uniformly continuous in the domain, the boundary integral treated do not have the same restrictions. We see that if the domain should hold the Lipschitz continuous conditions, $m > d/2$ that for $m = 2$, we have that $d = 2$ or $d = 3$. This is often a stricter condition than what we need for solving partial differential equations. To solve partial differential equations we need the trace of the functions u in $H^1(\Omega)$. The trace of such functions can not be evaluated directly since the boundary of a domain do not always have the quantity zero in relation to itself. It is however possible to evaluate this by an looking at a linear map $\gamma : H^1(\Omega) \to H^{1/2}(\partial\Omega)$, such that $\gamma u = u|_\gamma$ for all $u \in H^1(\Omega) \cap C^0(\overline{\Omega})$. Conversely each function $\phi \in \Omega$ be understood as the trace of a function of the function $u \in H^1(\Omega)$. Since the mapping γ is continuous and linear, i.e γu is embedded in u, we have the relation

$$||\gamma u||_H^{1/2}(\partial\Omega) \leq c||u||_{H^1(\Omega)}. \tag{2.27}$$

This means that functions which lies close to uniformly continuous functions is mapped in the same manner. This fact gives the H^1 functions a natural value on the boundary. If the trace of the functions defined in this manner, we can define the space $H_0^1(\Omega)$. If the Ω is limited and the boundary is Lipschitz continuous we define the space $H_0^1(\Omega)$ as,

$$H_0^1(\Omega) = \{u \in H^1(\Omega)|u|_{\partial\Omega} = 0\} \tag{2.28}$$

With this definition we can seek solutions in the H_0^1 space without concerning about the boundary integral problems if the functions is uniformly continuous.

An important subspace in the analysis of developing a discrete version of partial differential for control volume methods are the space $H(\text{div}, \Omega)$. We consider the space $\mathbb{R}^d$ and let $\Omega \subset \mathbb{R}^d$ be an open domain and $\mathbf{q} \in \mathbb{R}^d$ be a vector in this space. Then the space

$$H(\text{div}, \Omega) = \{\mathbf{q} \in (L^2(\Omega))^d | \text{div}\mathbf{q} \in L^2(\Omega). \tag{2.29}$$

This space has the norm

$$||\mathbf{q}||_{H(\text{div},\Omega)} = \left(\sum_{i=1}^{d} ||q_i||_{L^2(\Omega)}^2 + ||\text{div}\mathbf{q}||_{L^2(\Omega)}^2 \right)^{1/2}. \tag{2.30}$$

The space $H(\text{div}, \Omega)$ is a space where all the vectors components and the divergence to the vector lies in the space $L^2(\Omega)$.

2.2 Formulation of a CVFEM Scheme

In this section we will look at a CVFEM scheme through a boundary value model problem. We will look closer to the abstract view of the CVFEM and explain some important tools to treat the problem.

2.2.1 A Model Problem

We consider the self-adjoint elliptic problem on a polygon domain Ω with boundary $\partial\Omega$,

$$\begin{cases} -\nabla \cdot (A\nabla u) = f & \text{in } \Omega \\ \qquad\qquad u = 0 & \text{on } \partial\Omega, \end{cases} \tag{2.31}$$

where $f \in L^2(\Omega)$, $u \in H_0^1(\Omega)$ and A is finite symmetric positive definite matrix

2.2.2 Triangulation

We have to discretisize the domain Ω. Let $\tau_h = [K]$ be a triangulation of the domain Ω, where each triangle K does not have any interior angles greater than $\frac{\pi}{2}$. The triangulation is quasi uniform, which means that there exists

positive constant C_0, C_1 independent of h such that each triangle $K \in \tau_h$ is within a disk of radius $C_0 h$ and $C_1 h$. We can know formulate the piecewise linear conforming finite element spaces

$$S^h = \{ v \in C^0(\hat{\Omega}) : \quad v|_K \quad \text{is linear}, \forall K \in \tau_h \} \tag{2.32}$$

where $\hat{\Omega} = \cup_{K \in \tau_h} \hat{K}$ means that we are taking to account all the triangles on the domain, or the closure $S_0^h = S^h \cap H_0^1(\Omega)$.

2.2.3 Dual Mesh

We make a Donald dual partition such as in figure 2.6, and construct the control volumes $\tau_h^* = \{V_i\}_{i=1}^{M_1 + M_2}$. Here M_1 is internal nodes, and M_2 is boundary nodes. We denote the nodal points by $\{z_i\}_{i=1}^{M_1 + M_2}$. The dual mesh is constructed from the node points z_i. With Donald dual we mean that the lines defining the control volume meet in the barycentre of each triangle K. With the barycentre as base, the coefficient matrix will be non-symmetric if the diffusion coefficient is heterogeneous, but not necessary anisotropic. It will though be dependent on the mesh.
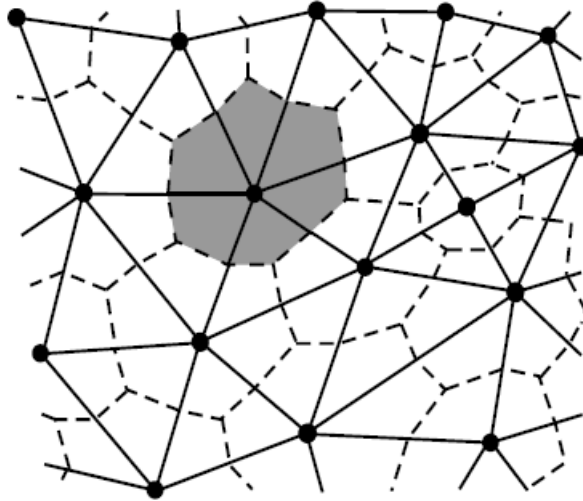

Figure 2.6: Visualization of the dual mesh. The shaded area denotes one control volume and the dotted lines the entire dual partition. The solid lines shows the original finite element mesh 2.32

2.2.4   Variational and Integral Formulation

To obtain the solution we determine the flux over the edges in each control volume. We assume that mass is conserved over the control volumes. We can state the model problem on integral form [1], which expresses conservation trough,

$$\int_{\partial \Omega} q \cdot n ds = \int_{\Omega} f dx. \tag{2.33}$$

where the flux $q = -A\nabla u$. For equation (2.33) to be valid we have to demand that $\mathbf{q} \in H(\text{div}, \Omega)$. We refer the reader to [1], for proof that this is sufficient regularity conditions such that the solution $u \in H_0^1(\Omega)$. With the right regularity conditions in place we want the equation to apply for the discrete problem. Equation (2.33) still applies if we look at one particular control volume V_i. The general equation is then modified to find $u_h \in S_0^h$ such that for V_i

$$-\int_{\partial V_i} (A\nabla u_h) \cdot \mathbf{n} ds = \int_{V_i} f dx \qquad i = 1, 2..., M_1. \tag{2.34}$$

We want equation (2.34) on variational form such that the we can use the powerful properties of this formulation. To get the problem on variational form we multiply both side of equation (2.31) with a test function and integrate over the domain. For the general problem this test function is defined as a function $v \in H_0^1(\Omega)$. If we then apply Greens formula we get the variational form of (2.31),

$$\forall v \in H_0^1(\Omega): \quad a(u, v) = \int_{\Omega} A\nabla v \cdot \nabla u dx = \int_{\Omega} f v dx. \tag{2.35}$$

The way equation (2.31) is defined it is zero on the boundary. This is the reason for why the boundary term not appear in (2.35). Equation (2.35) has much better regularity conditions than (2.31). We see that in the original problem we have to demand that the function u has to be two times differentiable, but in (2.35) it only has to be differentiable one time. The variational form of the equation (2.35) is stated on bilinear form. The reason for stating the variational formulation on bilinear form is the many application in analysis. It is easy through the bilinear form to show how we can extend the problem to operators which give the arise to solve the problem as an linear system of equations. Also the applications in the error analysis make it beneficial to state the problem in this way.
We now return to (2.34). As we have described it is useful to have the problem on variational form. Before we can multiply (2.34) with a test function,

we have to define this test function. Since equation (2.34) applies for the control volumes V_i where $i = [1, 2, \cdots, M_1]$ we have to consider this. We define the test function as for all $v \in S_0^h$

$$\pi_h v = \sum_{i=1}^{M_1} v(z_i) \chi_i. \tag{2.36}$$

We explain this test function a little more thoroughly. For every node in the discretization z_i, the term χ_i is the shape of the control volume surrounding this node. The term pi_h on the left hand side of equation (2.36) is a short hand notation applying the shape of each control volume to the right node. By choosing this test function, we make sure that we integrate over each control volume, and not over any other volume. Basicly this test function restricts the integration to the correct control volume. The summation means that we do this for every control volume V_i, and therefore this is equivalent with integrating over the whole domain Ω as in the (2.33). By applying equation (2.34) with a general test function $v \in H_0^1(\Omega)$ we get the integral form of (2.31).

$$a(u_h, v) = \int_{\partial \Omega} (A \nabla u_h) \cdot \mathbf{n} ds \quad v = \int_{V_i} f v dx. \tag{2.37}$$

We call it the integral formulation because we base the derivation on equation (2.33) and not the (2.31). At the same time we state the problem on bilinear form where we want to find $u_h \in S_0^h$ such that

$$a_h(u_h, v) = f_h(v) \qquad \forall v \in S_0^h \tag{2.38}$$

For all $v, w \in S_0^h$, where w represent the test function of (2.36) addition to some manipulation, equation (2.38) can be expressed as,

$$a_h(v, w) = -\sum_{i=1}^{M_1} \int_{\partial V_i} (A \nabla v) \cdot \mathbf{n} ds w(z_i), \quad f_h(v) = \int_{V_i} f \pi_h v dx. \tag{2.39}$$

This is a form of the problem (2.31) suitable for solving numerically. With the powerful tools of the bilinear form, it is easy to show how we can go from the abstract form to an matrix representation of the problem. This is shown in the next section.

2.2.5   From Abstract to Matrix Formulation

As equation (2.39) shows, we now have a system where we compute for every control volume and every node in the discretization. Further we have to

introduce a linear operator B_h from $S_0^h \rightarrow S_0^h$ such that

$$a_h(v, w) = (B_h v, w), \qquad v, w \in S_0^h, \tag{2.40}$$

where x is a vector in $\mathbb{R}^2$. Similarly we have for for $f_h(v)$ the linear operator f_h from $S_0^h \rightarrow S_0^h$ such that

$$f_h(v) = (f_h, v). \tag{2.41}$$

We can then write equation (2.39) on operator form

$$B_h u_h = f_h \tag{2.42}$$

where u_h can be written in terms of the interpolating shape functions N_i as in section 2.1.5. For all the nodes we can express the solution through

$$u_h = \sum_{i=1}^{M_1} u_i N_i. \tag{2.43}$$

With the introduction of the interpolation shape functions the matrix representation of (2.39) is

$$\overline{\mathbf{B}}_\mathbf{h} \overline{\mathbf{u}}_\mathbf{h} = \overline{\mathbf{f}}_\mathbf{h}. \tag{2.44}$$

In (2.44) the three different terms are respectively a matrix $\overline{\mathbf{B}}_\mathbf{h} = [a_{ij}]_{M_1 \times M_1}$, and vectors $\overline{\mathbf{u}}_\mathbf{h} = [u_1, u_2, \cdots, u_{M_1}]^T$, $\overline{\mathbf{f}}_\mathbf{h} = [u_1, u_2, \cdots, u_{M_1}]^T$ with, $a_{ij} = \int_{\partial V_i} (A \nabla N_j) \cdot n \, ds$ and $f_i = \int_{V_i} f(x) dx$. In general the CVFEM generates a coefficient matrix $\overline{\mathbf{B}}_\mathbf{h}$ which is non-symmetric. If the circumcenters had made up the control volumes we would get a symmetric coefficient matrix. Also if the diffusion coefficient are constant we will get a symmetric matrix.

2.3 Convergence Behaviour

In this section we briefly mention some error estimates in the H^1 and L_2 norms of the CVFEM. The formulation of the problem as in (2.38) allow us to use the techniques from finite element error analysis. We mention that for stricter conditions then what is envisaged here the result in the L_2 norm the order of convergence will not be the same. For analysis around CVFEM error estimates see [4, 13, 5, 10].

2.3.1   Estimation of the Error in the H^1 and L_2 Norm

If the triangulation have no interior angles grater than $\pi/2$, then [10] provides us with an estimate of the error in both the L_2 and H_1 norm. When $u \in H^2(\Omega)$, $u_h \in S_0^h$ where u and u_h denotes the solutions of respectively (2.33) and (2.34). The matrix A of (2.31) is a symmetric positive definite matrix, and $f \in L_2(\Omega)$, then the error in the H^1 norm is bounded by

$$||u - u_h||_{H^1} \le Ch||u||_{H^2}, \tag{2.45}$$

and in the L_2 norm it is defined as,

$$||u - u_h||_{L_2} \le Ch^2||u||_{H^2} \tag{2.46}$$

where constant C is independent of the mesh size h.

Chapter 3

Preconditioning and General Minimal Residual Method

In this chapter the essence of preconditioning [6] , condition number [6] and the general minimal residual method [20, 17] will be presented. In the section about preconditioning we described in general what the purpose of this term is, and in chapter 4 we introduce a specific preconditioner. We briefly talk about condition numbers, since the aim of preconditioner is to improve ill conditioned matrices. In the section about GMRES we describe the basics of how this may be implemented, and in the end give an estimate of the convergence without poof. Otherwise we talk about important terms in use for developing GMRES, such as Arnoldi iteration and Krylov subspaces.

3.1   Preconditioning

Preconditioning is about designing an effective matrix, the so-called preconditioner, in order to obtain a numerical solution with more accuracy or in less time [6, 20]. If we consider a linear system of n equations,

$$Ax = b, \tag{3.1}$$

where $A \in R^{n \times n}$, $b \in R^n$. Then a simple preconditioned system takes the following form,

$$MAx = Mb. \tag{3.2}$$

Where $M \in R^{n \times n}$ matrix that approximate A^{-1} so that only multiplication by M is needed. We begin with defining condition number of a matrix, since the goal of the preconditoner is exactly to lower this number. The target is to construct a preconditioned M such that the preconditioned matrix

$A_1 = MA$ has better matrix properties than A. There are two types of practical methods for solving (3.1), direct methods and iterative methods. Here the focus will be on iterative methods and especially on the GMRES. Our aim is to use a two level additive Schwartz method as a preconditoner for the global system (3.1). We often say that a preconditoner M of a matrix A is a matrix such that $M^{-1}A$ has a smaller condition number than A. These preconditoners are useful when solving large sparse linear systems. For iterative solvers as GMRES and others, the rate of converges decay as the condition number of the matrix A increases. If we find a good preconditoner it will reduce the condition number.

3.1.1   Condition of a Problem and Condition Number

If a small perturbation in input data leads to a small perturbation in output data we say the problem is well conditioned. One the other hand an ill conditioned problem is one where a small perturbations in the input gives large changes in the output data. The analogy "small" and "large" is dependent on the application and the problem. A problems condition can be measured in what we call the condition number.

Theorem 2 *Let $A \in \mathbb{C}^{m*m}$ be non-singular and consider the equation $Ax = b$. The problem of computing b, given x, has condition number*

$$\kappa = ||A|| \frac{||x||}{||b||} \leq ||A|| \, ||A^{-1}|| \tag{3.3}$$

with respect to perturbations of x The problem of computing x, given b, has condition number

$$\kappa = ||A^{-1}|| \frac{||b||}{||x||} \leq ||A|| \, ||A^{-1}|| \tag{3.4}$$

with respect to perturbations of b. If $||\cdot|| = ||\cdot||_2$, then equality holds in (3.3) if x is a multiple of a right singular vector of A corresponding to the minimal singular value σ_m, and equality holds in (3.4) if b is a multiple of a left singular vector of A corresponding to the maximal singular value σ_1.

The product $||A|| \, ||A^{-1}||$ is what we refer to as the condition number of A, and is denoted by $\kappa(A)$:

$$\kappa = ||A|| \, ||A^{-1}|| \tag{3.5}$$

If $\kappa(A)$ is small, A is said to be well conditioned, if $\kappa(A)$ is large, A is ill-conditioned. If A is singular, it is usual to denote $\kappa(A) = \infty$. The usage of

$||\cdot|| = ||\cdot||_2$ results in $||A|| = \sigma_1$ and $||A^{-1}|| = \frac{1}{\sigma_m}$. The condition number can then be written in the 2-norm,

$$\kappa(A) = \frac{\sigma_1}{\sigma_m}, \tag{3.6}$$

and is used to for computing 2-norm condition numbers of matrices. The aim in preconditioning is to reduce the condition number of the system. Typically one can use domain decomposition methods. If A is normal i.e. $A^T A = AA^T$, then the condition number can be stated as

$$\kappa(A) = ||\frac{\lambda_{max}(A)}{\lambda_{min}(A)}|| \tag{3.7}$$

where $\lambda_{max}(A)$ and $\lambda_{min}(A)$ are maximum and minimum eigenvalues of A. The condition number can be a measure of how long time the iterative solver will use on a desired system of equations. The point of using a preconditioner is exactly reduce the condition number such that the solver use less time on the solving the linear system. A good linear solver is the General Minimal Residual Method which is discussed in the next section. If the matrix A is poorly conditioned or close to singular it can be convenient to look at the the symmetric part of A, by threat the expression $(A + A^T)/2$ instead of A in the above theory.

3.2 General Minimal Residual Method(GMRES)

Trough the next section the method described the main theory and results are provided from [20, 17]. The goal of the GMRES is to solve (3.8) and when we have discretisized (2.31) by using the CVFEM, we obtain the linear system,

$$Px = b, \tag{3.8}$$

where $P \in \mathbb{R}^{n \times n}$ is a non symmetric but sparse $(n \times n)$ matrix, $b \in \mathbb{R}^n$, and $x \in \mathbb{R}^n$ is the unknown. Although the GMRES can solve symmetric problems, it is designed to solve non symmetric but sparse system. The coefficient matrix A in (2.44) is general non symmetric, and is suitable to be solved by a preconditioned GMRES method with the preconditioner given by a two level domain decomposition method. The GMRES method is well documented in the sense of finite element method, and all the applications and analysis is to be find in [20]. Hence it may be convenient to explore its applications to finite volume equations. The GMRES is used because of the non symmetric nature of the coefficient matrix, and in this section we

describe the most important aspects of the algorithm but refer to [6] for more details. In for example reservoir simulation, wells and irregular grid block structure destroy the natural sparse symmetric structure of the coefficient matrix. Through the past few decades we have seen a flourish of iterative techniques, many of them are the Krylov subspace methods[16]. There are several parameter-free Krylov subspace algorithms that have been developed for solving non symmetric systems of linear equations. With parameter-free refer to the lack of choosing a parameter such as required in successive over relaxation type methods [15]. GMRES is one of the parameter-free Krylov subspace methods. We want to minimize the residual in the Krylov subspace $\kappa_n = (b, Pb, \cdots, P^{n-1}b)$. We use the notation $x_* = P^{-1}b$ for the exact solution of the problem (3.8). The idea of the GMRES can be formulated in one sentence. At step n, the approximation of x_* by the vector $x_n \in \kappa_n$ shall minimize the norm of the residual $r_n = b - Px_n$. This is the same as solving a least square problem. We define the Krylov matrix K_n, which is nothing else but the Krylov subspace on a matrix formulation,

$$K_n = \left[\ b\ \middle|\ Pb\ \middle|\ \cdots\ \middle|\ P^{n-1}b\ \right]$$

3.2.1 GMRES as a Least Square Problem

We can now solve the least square problem, Let K_n be the defined as above, then

$$PK_n = \left[\ Pb\ \middle|\ P^2b\ \middle|\ \cdots\ \middle|\ P^nb\ \right]$$

The column space of PK_n is $P\kappa_n$. Then the problem is to find a vector $c \in \mathbb{C}^n$ such that

$$||PK_nc - b||_2 = \min \tag{3.9}$$

This can be done by a decomposition into a orthogonal and upper triangular matrix, or a so called QR factorization, where $K_n = Q_nR_n$. Once c is found, we can solve the problem $x_n = K_nc$. This is an unstable way of developing GMRES. The cost of this type of GMRES are also to high. Instead it is more efficient and benificial to use the Arnoldi iteration to construct a sequence of the Krylov matrices. The Krylov subspace is not linearly independent. We have to find another basis. For this purpose it is clever to use the so called Arnoldi iteration [20], which is a Gram-Schmidt style iteration [15] of finding orthonormal vectors forming the basis of κ_n. The basis is denoted with $(q_1, q_2, \cdots, q_n)$. The Arnoldi iteration construct a sequence of Krylov matrices Q_n, where the columns of the matrices $(q_1, q_2, \cdots, q_n)$ span the successive Krylov subspaces κ_n. The Arnoldi iteration intention is to transform

a matrix to Hessenberg form. A complete reduction of P to Hessenberg form by an orthogonal similarity transformation can be written $P = QHQ^*$ or $PQ = QH$. The term Q^* is the adjoint or the hermitian conjugate of Q, i.e. the transpose if Q is real. Because of cost, and the fact that m can be very large or even infinite, we consider only the first n columns of $PQ = QH$. Let the Q_n be the $(m \times n)$ matrix whose the first n columns of Q are given by

$$Q_n = \left[\begin{array}{c|c|c|c} q_1 & q_2 & \cdots & q_n \end{array} \right]$$

Let $\tilde{H}_n$ be the $(n+1) \times n$ upper left section of H, which also are a hessenberg matrix on the form,

$$\tilde{H}_n = \begin{bmatrix} h_{11} & & \cdots & & h_{1n} \\ h_{21} & h_{22} & & & \\ & \ddots & \ddots & & \vdots \\ & & h_{n,n-1} & & h_{nn} \\ & & & & h_{n+1,n} \end{bmatrix}$$

With this definition of $\tilde{H}_n$, we can write

$$PQ_n = Q_{n+1}\tilde{H}_n. \tag{3.10}$$

By multiplying the matrices in equation (3.10) we obtain the n'th column of the expression

$$Pq_n = h_{1n}q_1 + \cdots + h_{nn}q_n + h_{n+1,n}q_{n+1}, \tag{3.11}$$

where q_{n+1} is based on the previous term of the Krylov vectors and itself. It is a so called $(n+1)$ recurrence relation. The Arnoldi iteration is the task of implementing equation (3.11). With the concepts of the Arnoldi iteration and the Krylov matrices in mind, the task can be reformulated from $x_n = K_n c$ to $x_n = Q_n y$. Instead of solving equation (3.9) the new least square problem to solve is to find a vector $y \in \mathbb{C}^n$ such that

$$||PQ_n y - b||_2 = \text{is minimized}. \tag{3.12}$$

The problem (3.12) has dimension $m \times n$, because of the structure of Krylov subspaces, the actual dimension of the problem is $(n + 1) \times n$. This can be solved by applying definition (3.10) to equation (3.12), then we get the minimization task problem

$$||Q_{n+1}\tilde{H}_n y - b||_2 = \min. \tag{3.13}$$

Expression (3.13) can further be modified since both vectors of the norm are inside the column space of Q_{n+1}. Multiplying with Q_{n+1}^{-1} on both sides of the components in the norm does not change the result. The expression

$$||\tilde{H}_n y - Q_{n+1}^{-1} b||_2 = \min, \tag{3.14}$$

is equivalent with (3.13). Because of the orthogonality of Q_{n+1}, we have that the term Q_{n+1}^{-1} is equal to $||b|| e_1$, where $e_1 = (1, 0, 0, \cdots)$, which is the first standard basis of $\mathbb{R}^{n+1}$ The final form of the GMRES least square problem

$$||\tilde{H}_n y - ||b|| e_1||_2 = \min. \tag{3.15}$$

The overall algorithm of the GMRES solves this minimization task with respect to y at step n and then solve the problem $x_n = Q_n y$. At each step the GMRES minimize the norm of the residual $r_n = b - P x_n$ over all the vectors $x_n \in \kappa_n$. Minimization of the residual $r_n = b - P x_n$ is done in the minimization task (3.15), and not explicitly from x_n, i.e.

$$||r_n|| = ||\tilde{H}_n y - ||b|| e_1||_2.$$

The GMRES consists in general of four important steps:

1. For generating the $(n + 1) \times n$ upper Hessenberg matrix $\Longrightarrow$ Arnoldi iteration.

2. Find the y_n that minimize the residual $||r_n||$.

3. Computation of $x_n = Q_n y_n$.

4. Repeat until the residual is within the desired error.

3.2.2   Convergence of the GMRES

The nth iteration minimizes the residual in κ_n. As mentioned previously, every subspace is contained in the next subspace and consequently the residual decreases monotonically. For a positive definite matrix, the rate of convergence can be expressed as,

$$||r_n|| \leq (1 - \frac{\lambda_{min}(P^T + P)}{2\lambda_{max}(P^T + P)})^{n/2} ||r_0|| \tag{3.16}$$

λ denotes the largest and smallest eigenvalue of the matrix that consists of P^T and P. I additional the matrix P is symmetric, the rate of convergence is,

$$||r_n|| \leq (\frac{\kappa_2^2(P) - 1}{\kappa_2^2(P)})^{n/2} ||r_0||, \tag{3.17}$$

where κ_2^2 is the condition number in the euclidean way. In the general we can describe the convergence rate of the GMRES in the manner of Einsenstat, Elman and Shultz **??**. As for the CVFEM we let P be a linear operator in the finite dimensional element space $\mathbb{R}^n$ which is associated with an inner product $[\cdot, \cdot]$ and a norm $|| \cdot ||$. Both the norm and inner product shall reflect the peculiar properties of P. For the next theorem we also assume that P is non symmetric but positive definite. Then the GMRES method can be characterized in terms of the minimal eigenvalue symmetric part of the operator and the norm of the operator. They are defined as

$$c_p = \inf_{x \neq 0} \frac{[Px, x]}{[x, x]} \quad \text{and} \quad c_p = \sup_{x \neq 0} \frac{||Px||}{||x||} \tag{3.18}$$

By considering the decrease of the norm of the residual in a single step, the following theorem can be established for the convergence.

Theorem 3 *(Einsenstat, Elman, Shultz) If $c_p > 0$, then the GMRES method converges after m steps, the norm of the residual is bounded by*

$$||r_m|| \leq (1 - \frac{c_p^2}{C_p^2})^{1/2} ||r_0||. \tag{3.19}$$

Chapter 4

Domain Decomposition

Domain decomposition among other is used as preconditioner for Krylov space iterative methods, i.e. GMRES [17, 20] and Conjugate Gradient method [12], for solving boundary value problems. A detailed description of a domain decomposition method, a two level additive Schwartz method, will be described later in the chapter. The main idea of domain decomposition is to split the problem into smaller boundary value problems or subdomains, and then synchronize the solution from every subdomain. This makes these methods suitable for parallel computing. There are different ways of splitting the domain, and we separate between overlapping and non-overlapping domain decomposition. In overlapping methods the subdomains interfaces overlaps with each other, and in non-overlapping the intersection is only on the boundary. An advantage of overlapping methods are faster convergence, but it is also a more costly routine and the implementation is often more complex. In these methods it is common to distinguish between coarse and fine scale. The solution is solved both in the coarse scale and the fine scale on every subdomain. The coarse and fine scale can have many different levels and, a so called multilevel method. A two level method consist of only one fine scale and one coarse scale.

The reason for solving the method on both coarse and fine scale, is that if we only solve the problem on fine scale, the low frequencies of the solution will be lost [18]. To compensate for this, we also solve it on a coarse scale. If we only solve it on a coarse scale, the precision of the solution will probably not be satisfied, dependent of the problem and application. There are many types of domain decomposition, but we will here only talk about the Schwartz methods. Schwartz was the first person to develop a domain decomposition method in the late 19th century. The main theory in the next sections about the Schwarz methods are form [18, 14, 9].
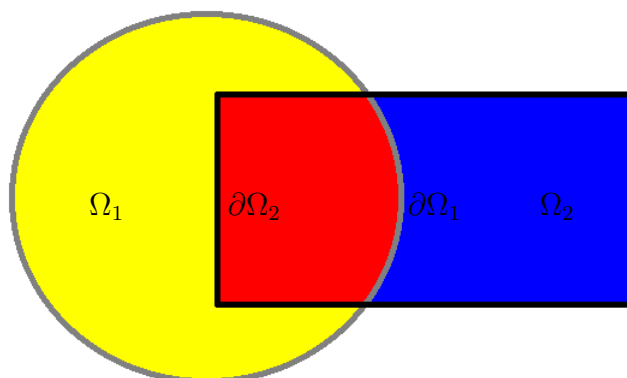
Figure 4.1: Schwarz's famuse figure, two overlapping subdomains Ω_1 and Ω_2 with two artificial interfaces $\partial\Omega_2$ and $\partial\Omega_1$.

4.1 General Theory About Schwarz Methods

For understanding a two level Schwars methods it is beneficial to understand the basics of general Schwarz methods, we briefly will go true some of the theory behind these methods. We will go true the overlapping Schwarz method, the additive Schwarz method and finally the two level additive Schwarz method, which is used in this thesis.

Consider the domain Ω as in figure 4.1. The domain Ω consists of two overlapping subdomains $\Omega_1 \cup \Omega_2$. The goal is to solve the linear PDE ,

$$Lu = f \quad \text{in} \quad \Omega, \tag{4.1}$$

$$u = g \quad \text{on} \quad \partial\Omega, \tag{4.2}$$

where $\partial\Omega$ denote the boundary of Ω. Also in the rest of this chapter we say that $u \in H_0^1(\Omega)$, $g, f \in L^2(\Omega)$ and L is a linear operator similar to the operators in equation (2.42). The artificial boundary $\partial\Omega_i$ is the part that is inside the domain Ω_i that is interior to Ω. The overlapping Schwarz method iterates by first selecting a initial guess u_2^0 for the values in $\partial\Omega_1$, and then solve the boundary value problem iteratively for u_1^n.

$$Lu_1^n = f \quad \text{in} \quad \Omega_1 \tag{4.3}$$

$$u_1^n = g \quad \text{on} \quad \partial\Omega_1\partial\Omega_1, \tag{4.4}$$

$$u_1^n = u_2^{n-1} \quad \text{on} \quad \partial\Omega_1, \tag{4.5}$$

The same procedure is done for u_2^n,

$$Lu_2^n = f \quad \text{in} \quad \Omega_2 \tag{4.6}$$
$$u_2^n = g \quad \text{on} \quad \partial\Omega_2\partial\Omega_2 \tag{4.7}$$
$$u_2^n = u_1^{n-1} \quad \text{on} \quad \partial\Omega_2 \tag{4.8}$$

This method described here is called the alternating Schwarz method. Often it is useful and possible to use a matching grid in the overlapping region such that the unknowns dose not duplicate each other. A method that has this feature is called the **multiplicativeSchwarz** method. The discretisized problem can be formulated as $Au = f$ similar as in chapter 2.2. The iteration can then be written in two steps:

$$u^{n+1/2} = u^n + \begin{pmatrix} A_{\Omega_1}^{-1} & 0 \\ 0 & 0 \end{pmatrix} (f - Au^n) \tag{4.9}$$

and

$$u^n = u^{n+1/2} + \begin{pmatrix} 0 & 0 \\ 0 & A_{\Omega_2}^{-1} \end{pmatrix} (f - Au^{n+1/2}). \tag{4.10}$$

A_{Ω_i} is the discrete form of the operator L, restricted to Ω_i. This basic method is not suitable for parallel computing since each iteration involves sequential fractional steps. As it is seen from the equation 4.9 and 4.10 this is nothing else than a Gauss-Seidel method divided into blocks. This generalization of the Gauss-Seidel method is called the block Gauss-Seidel method. A more robust method for parallel computing is the additive Schwarz method.

4.1.1 One Level Additive Schwarz Method

The additive Schwarz method (ASM) is the multiplicative Schwarz method where the iteration is done in one step. If we put **??** and **??** the result is the parallelized version of the multiplicative Schwarz or ASM.

$$u^{n+1} = u^n + \left[\begin{pmatrix} A_{\Omega_1}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & A_{\Omega_2}^{-1} \end{pmatrix} \right] (f - Au^n).$$

The expression above can be written in a compressed manner

$$u^{n+1} = u^n + (B_1 + B_2)(f - Au^n), \tag{4.11}$$
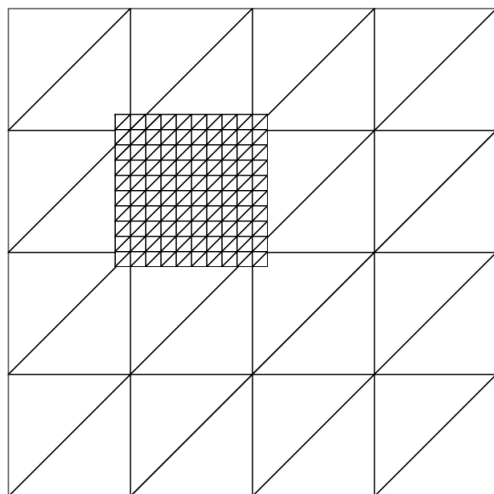
Figure 4.2: The fine scaled region represent one subdomain. On every subdomain the discrete version of the problem is solved and summarized. This is suitable as a preconditioner for the general fine scaled problem. The fine scaled domain can be viewed as the restriction operator R_i of the ASM

where is $B_i = R_i^T A_\Omega^{-1} R_i$. R_i is what we call the restriction matrix. The method will converge if we do not weighting the overlap, but with weighting the convergence properties will increase. The idea behind the restriction matrix, which we have looked at earlier, is to makes sure that that the components in the particular domain Ω_i is returned. The ASM can easily be extended to multiple subdomains. If the domain Ω is divided into $\Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_{i-1} \cup \Omega_i$ overlapping subregions, the ASM can be expressed as follows.

$$u^{n+1} = u^n + \sum_i B_i(f - Au^n). \tag{4.12}$$

The additive Schwarz method now apply for i subdomains. For the attentive reader one may see that the ASM is nothing else but a generalization of the Jacobi method. It is called the block Jacobi method.

4.2   Two Level Additive Schwarz Method

The problem with single level methods is that in large problems and cases with many subdomains the effictivity decreases dramaticly. There is no connection between the subdomains except the interface or the overlap. If there

are many subdomains it is critical for the convergence that there are some sort of interaction between the domains. The iterative linear solver needs some global way of communicate of information at each iteration. A efficient and reliable way of transmitting global information across the domains is the multilevel method. The idea of a multilevel method is to use coarse spaces to nurture the communication between the domains. It is possible to use several different coarse spaces, but the most basic multilevel method is to use one coarse space and is called a two level method. The combination of the ASM and one coarse space correction is the two level additive Schwarz method. We look at the two different systems $A_C u_C = f_C$ and $A_F u_F = f_F$, respectively the coarse and fine discretizations of the problem. If the error on the coarse grid is known, it is possible to interpolate it back to the fine grid and use this as a correction. The problem is that we do not know the error. However if the residual on the coarse grid is known the error on the coarse grid can be calculated from the error equation $A_C e_C = r_C$. Again we encounter a problem, we do not know the residual r_C on the coarse grid. We only now the residual on the fine grid, and by interpolation we can approximate the residual. R_0^T is a representation of the linear interpolation from coarse to the fine grid. A coarse grid correction can then be stated as follows;

$$\mathbf{u_F} = \mathbf{u_F} + \mathbf{R_0^T A_C^{-1} R_0}(\mathbf{f} - \mathbf{A_F u_F}) \tag{4.13}$$

This equation states that the residual is calculated, we restrict it to the coarse grid solves the coarse grid problem, and then at last interpolate the solution back to the fine grid. This is the basic of a coarse grid correction. The problem is just that this is not possible. What is possible is to use the coarse grid correction term $R_0^T A_C^{-1} R_0$ as a preconditioner. The idea is that the coarse grid part of the preconditioner has a large null space. The rank, or linearly independent columns of $R_0^T A_C^{-1} R_0$ has the same dimension as A_C. Dependent on the coarse grid, this is normally much fewer than the dimension of the fine scale. This principle makes sure that the error or residual which lies in the null space of the coarse grid term never is corrected. A specially the components that include most of the high frequency errors. This is the basic behind a two level method, a coarse grid correction which correct all the errors in the fine grid and a local or fine grid solver. A good rule is that the relation between the subdomains and the coarse grid should be scalable, which means that the performance should be insensitive to the number of subdomains.

Iteration of a Two Level Additive Schwarz Method

$$r_F^n = f - Au_F^n \qquad\qquad \text{Computation of the residual on} \Omega$$
$$r_C = R_0 r_F^n \qquad\qquad \text{Restrict the residual to the coarse space}$$
$$A_C v_c = r_C \qquad\qquad \text{Solve on the coarse space} \quad (A_C = R_0 A R_C^T)$$
$$v^n = R_0^T v_C \qquad\qquad \text{Interpolate the correctio}$$
$$u_F^n = u_F^n + v^n \qquad\qquad \text{Compute the new approximation}$$
$$u_F^{n+1} = u_F^n + \sum_i B_i r_F^n \qquad\qquad \text{ASM on} \Omega$$

4.3   The Abstract Schwarz Framework

In this section we will give an abstract view of the two level additive Schwarz method. The terms and conditions are also used in the convergence analysis chapter 6. The aim is to use the Schwarz framework a method solving equation (2.44) and study the convergence.

4.3.1   Two Level Overlapping Additive Schwarz Method

The domain Ω is portioned into subdomains $\tau_H = \{\Omega_i\}_{i=1}^N$ which are quasi uniform with mesh size H. The boundary of τ_H is aligned with τ_h. This coarse triangulation can be associated with the piecewise linear finite element space S^H and S_0^H in the same manner as in 2.2.2, similarly as S^h and S_0^h. To gain a overlapping method we have to extend the subdomains Ω_i into larger regions $\Omega_i{}'$, i.e. $\Omega_i \subset \Omega_i{}'$ such that $\overline{\Omega} = \subset_{i=1}^N$. The boundary $\partial\Omega_i{}'$ are also aligned with τ_h. The minimum distance between $\Omega_i{}'$ and Ω_i are denoted by

$$\delta = \min_i \text{dist}(\partial_i{}' \setminus \partial\Omega, \partial\Omega_i \setminus \partial\Omega). \qquad (4.14)$$

In [18] they define the difference between generous and small overlap. If δ is proportional with h we say the subdomains $\Omega_i{}'_{i=1}^N$ has a small overlap. On the other hand if there exists a $\beta > 0$, such that $\delta \geq \beta H$, we say that $\Omega_i{}'_{i=1}^N$ has a generous overlap. A quasi uniform finite element subdivision of each $\Omega_i{}'$, where $i > 0$ is obtained from the finite element triangulation τ_h. The new space is defined as

$$S_i^h = H_0^1(\Omega_i{}') \cap S_0^h. \qquad (4.15)$$

The space S_i^h extends the subspace S_0^h by zero since it S_i^h vanish on the boundary $\partial \Omega_i'$. The decomposed subspace S_0^h can be viewed as

$$S_0^h = S_0^H + \sum_{i=1}^{N} S_i^h. \tag{4.16}$$

As for the CVFEM method we define a bilinear $a_H(v, w)$ similarly as $a_h(v, w)$, with the difference that $v, w \in S_0^H$ and change τ_h with τ_H. A dual mesh are related with the τ_H, similarly as for τ_h. We let the linear operators $T_0 : S_0^h \to S_0^H$ and $T_0 : S_i^h \to S_i^H$, $1 \le i \le N$, be defined as

$$a_H(T_0 v, w) = a_h(v, w), \quad v \in S_0^h, \quad w \in S_0^H, \tag{4.17}$$

$$a_h(T_i v, w) = a_h(v, w), \quad v \in S_0^h, \quad w \in S_i^H. \tag{4.18}$$

$$\tag{4.19}$$

For (4.17) to be well posed we assume that there exist a constant H_0, such that for $h < H < H_0$. This is due to the fact that the subspace S_i^H and S_i^H are finite dimensional Hilbert spaces. The two level additive Schwarz method described has the intention to solve (2.38), by solving a similar equivalent problem,

$$T u_h = \tilde{f}_h, \tag{4.20}$$

where $T = \sum_{i=0}^{N} T_i$, $\tilde{f}_h = \sum_{i=0}^{N} T_i u_h$. We solve the terms $T_i u_h$ in the subspace such that,

$$a_H(T_0 u_h, v) = a_h(u_h, v) = f_h(v), \quad v \in S_0^H, \tag{4.21}$$

and,

$$a_h(T_i u_h, v) = a_h(u_h, v) = f_h(v), \quad v \in S_i^H. \tag{4.22}$$

It is not obvious that (4.20) and (2.38) are equivalent. We further show how the equation (4.20) are defined on operator and matrix forms. We define some linear bounded operators for use in the analysis later. Let $Q_0 : S_0^h \to S_0^H$, $Q_i : S_0^h \to S_i^H$, $1 \le i \le N$ which are the standard L^2 orthogonal projection operators. Let $B_0 : S_0^H \to S_0^H$, $B_i : S_i^H \to S_i^H$, $1 \le i \le N$ be operators such that,

$$(B_0 v, w) = a_H(v, w), \quad \forall v, w \in S_0^H, \tag{4.23}$$

$$(B_i v, w) = a_h(v, w) = (B_h v, w), \quad \forall v, w \in S_i^h. \tag{4.24}$$

The way the operators T_i, Q_i and B_i is given, we see that

$$B_i T_i = Q_i B_h, \quad i = 1, 2, \cdots, N, \tag{4.25}$$

or,

$$T_i = B_i^{-1} Q_i B_h, \quad i = 1, 2, \cdots, N. \tag{4.26}$$

The two level additive Schwarz method (4.20) then becomes

$$S_h B_h u_h = S_h f_h, \tag{4.27}$$

where $S_h = \sum_{i=0}^{N} B_i Q_i$. By the definition of preconditioners in section 3.1, we see that this is just equation (2.38) preconditioned by S_h.

4.3.2   A Matrix Representation of the Two Level Over- lapping Additive Schwarz Method

The number of interior nodes in S^h are defined as the the number of elements in Ω, and are denoted with $|\hat{\Omega}|$. For every Ω_i', $1 \leq i \leq N$ the interior nodes are defined as the number of elements in the specific subdomain and denoted with $|\hat{\Omega}'|$. Remember, with interior in one subdomain we do not mean the artificial boundary, only the fixed boundary from the global problem. We first apply a global ordering of the interior nodes in the domain $\hat{\Omega}$. Further we assume that the set of nodes in $\hat{\Omega}_j'$, where $[p_1, p_2, \cdots, p_M]$ are points in the domain has the global indices $[i_1, i_2, \cdots, i_M]$, respectively, with $i_1 < i_2 < \cdots < i_M$. Every subdomain has local index $[1, 2, \cdots, M]$ corresponding to the global index respectively. We can describe the local matrix as $\mathbf{\overline{B}_i} = (q_{kl})_{1 \leq k,l \leq M}$, where $q_{kl} = b_{i_k i_l}$ from equation (2.44). For all i an extension operators $R_i : E^{|\hat{\Omega}_i|} \rightarrow E^{|\hat{\Omega}|}$ is defined such that,

$$E_i v(k) = \begin{cases} v(k) & k \in \hat{\Omega}_i, \\ 0 & \text{otherwise}, \end{cases}$$

for arbitrary $v \in E^{|\hat{\Omega}_i|}$. Through the extension operator R_i the global representation of the particular domains is preserved. Basicly R_i transfer the local representation of the subspace S_i^h and associate it with the global indexing in S_0^h. By some simple manipulation we can obtain the matrix representation of (4.27) such that,

$$\mathbf{\overline{S}_h \overline{B}_h \overline{u}_h = \overline{S}_h \overline{f}_h}, \tag{4.28}$$

where,

$$\mathbf{\overline{S}_h = R_H \overline{B}_H^{-1} R_H^T + \sum_{i=1}^{N} R_i \overline{B}_i^{-1} R_i^T}. \tag{4.29}$$

Here $\mathbf{R_i^T}$ and $\mathbf{R_H^T}$ denote the transpose of the corresponding matrix.

Chapter 5

Implementation of the CVFEM and the Additive Schwarz Framework

In this chapter we go through the algorithm step by step, so that it is easy to reproduce. First we explain the assembly of the coefficient matrix in the CVFEM which consists of the most not evident parts of the implementation. In same section we describes how other aspects of implementing the CVFEM can be done. Later in the section a matrix deviation of how the tow level additive Schwarz with an default overlap could be designed.

5.1 Assembly of the Coefficient Matrix in the CVFEM

If there are M nodes the coefficient matrix will be a matrix of $(M \times M)$ elements. First of all the dual partition is obtained. This is done through the definition of what the region of support is. As described in 2.1.3 the region of support is the nodes surrounding a node. The nodes in a support is listed in an array, as shown below

$$R_{i,j} = [r_{i,1}, r_{i,2}, \cdots, r_{i,j}], \tag{5.1}$$

where $R_{i,j}$ is the the i'th node, with j nodes in the support. To distinguish between internal and boundary nodes, we add a zero on the end of the array of a boundary node. There is now M different nodes with each have a array containing the neighbouring nodes. Further there are some calculations that involves the above arrays and information used to calculate the stiffness

matrix that have to be obtained. Loop over all the nodes M, for every node there is assembled an array containing the neighbouring nodes. Then a new loop is applied, looping over the length of the array containing the neighbouring nodes. We now point at the different nodes in the region of support such that we have a system that creates the triangulation.

$$[1 \quad 2 \quad 3] = [i, R_{i,j}, R_{i,j+1}]. \tag{5.2}$$

We associates the nodes (5.2) with the coordinates such that,

$$x_1 = x_i \qquad\qquad x_2 = x_{R_{i,j}} \qquad\qquad x_3 = x_{R_{i,j+1}}$$
$$y_1 = y_i \qquad\qquad y_2 = y_{R_{i,j}} \qquad\qquad y_3 = y_{R_{i,j+1}}.$$

The information obtained is used to calculate the necessary parts to develop the coefficient matrix and the right hand side. In the model problem (2.31) the diffusion coefficients is defined as A, but to avoid confusion we redefine it to be κ. Further we have that one element has the volume

$$V_{tri} = \frac{1}{2} \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_{R_{i,j}} & y_{R_{i,j}} \\ 1 & x_{R_{i,j+1}} & y_{R_{i,j+1}} \end{vmatrix}$$
$$= \frac{(x_{R_{i,j}} y_{R_{i,j+1}} - x_{R_{i,j+1}} y_{R_{i,j}}) + x_i(y_{R_{i,j}} - y_{R_{i,j+1}}) + y_i(x_{R_{i,j+1}} - x_{R_{i,j}})}{2}.$$

The contribution of one element to the one specific control volume is.

$$V_i^j = \frac{1}{3} V_{tri} \tag{5.3}$$

To obtain an oveview we denote the labeling with the local labeling $1, 2, 3$, which represent one triangle. The derivatives of the shape functions can then be expressed and calculated.

$$N_{1x} = \frac{\partial N_1}{\partial x} = \frac{(y_2 - y_3)}{2V_{tri}}, \qquad N_{1y} = \frac{\partial N_1}{\partial y} = \frac{(x_3 - x_2)}{2V_{tri}}$$
$$N_{2x} = \frac{\partial N_2}{\partial x} = \frac{(y_3 - y_1)}{2V_{tri}}, \qquad N_{2y} = \frac{\partial N_2}{\partial y} = \frac{(x_1 - x_3)}{2V_{tri}}$$
$$N_{3x} = \frac{\partial N_3}{\partial x} = \frac{(y_1 - y_2)}{2V_{tri}}, \qquad N_{3y} = \frac{\partial N_3}{\partial y} = \frac{(x_2 - x_1)}{2V_{tri}}$$

With theses shape functions the gradient can be expressed, and the components are with respect to x and y,

$$\frac{\partial \phi}{\partial x} = N_{1x}\phi_1 + N_{2x}\phi_2 + N_{3x}\phi_3$$
$$\frac{\partial \phi}{\partial y} = N_{1y}\phi_1 + N_{2y}\phi_2 + N_{3y}\phi_3$$
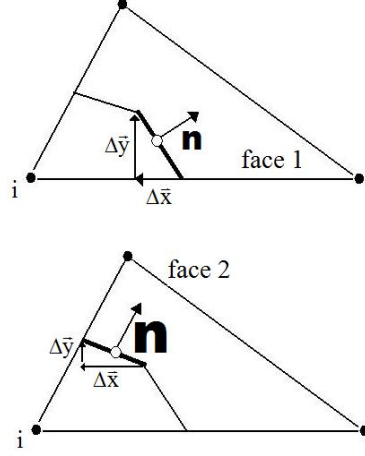
Figure 5.1: The faces of a element

It is possible to calculate the faces of the problem, refer to the figure 5.1.

$$\Delta\overline{x}_{f1} = \frac{x_3}{3} - \frac{x_2}{6} - \frac{x_1}{6} \qquad\qquad \Delta\overline{x}_{f2} = -\frac{x_3}{3} + \frac{x_2}{6} + \frac{x_1}{6}$$

$$\Delta\overline{y}_{f1} = \frac{y_3}{3} - \frac{y_2}{6} - \frac{y_1}{6} \qquad\qquad \Delta\overline{y}_{f2} = -\frac{y_3}{3} + \frac{y_2}{6} + \frac{y_1}{6}$$

When we know $\Delta\overline{x}_{f1}$, $\Delta\overline{y}_{f1}$, $\Delta\overline{x}_{f2}$ and $\Delta\overline{y}_{f2}$, the area or the length of the face can be derived by using Phytagoras as 5.1 shows. The footnote $f1$ and $f2$ refers to the face of the elements we are threatening. See figure (5.1) for illustration.

$$A_{f1} = \sqrt{\Delta\overline{x}_{f1}^2 + \Delta\overline{y}_{f1}^2} \tag{5.4}$$

$$A_{f2} = \sqrt{\Delta\overline{x}_{f2}^2 + \Delta\overline{y}_{f2}^2} \tag{5.5}$$

The flux over each face in there specified direction can be approximate by dividing the x and y component in each direction with the area over the face. On face 1 the flux can be expressed as

$$\mathbf{n}_x^{f1} = \frac{\Delta\overline{x}_{f1}}{A_{f1}}, \qquad n_y^{f1} = \frac{\Delta\overline{x}_{f1}}{A_{f1}}, \tag{5.6}$$

and on face 2 the flux can be expressed as,

$$\mathbf{n}_x^{f2} = \frac{\Delta\overline{x}_{f2}}{A_{f2}}, \qquad n_y^{f2} = \frac{\Delta\overline{y}_{f2}}{A_{f2}}. \tag{5.7}$$

At the midpoint at each face, the value of the diffusivity is determined by the value κ and the interpolating shape functions.

$$\kappa_{f1} = [N_1\kappa_1 + N_2\kappa_2 + N_3\kappa_3]_{f1} = \frac{5}{12}\kappa_1 + \frac{5}{12}\kappa_2 + \frac{2}{12}\kappa_3$$

$$\kappa_{f2} = [N_1\kappa_1 + N_2\kappa_2 + N_3\kappa_3]_{f2} = \frac{5}{12}\kappa_1 + \frac{2}{12}\kappa_2 + \frac{5}{12}\kappa_3$$

This is sufficient information to calculate the left hand side of the problem. The challenge is how to compute, and use the information above to get a complete closed system of equations,

$$B_h u_h = f_h, \tag{5.8}$$

and solve it with a iterative method. The integral formulation gives the rise to determine the components of the coefficent matrix.

$$B_h = a_{ij} = \int_{\partial V_i} (\kappa \nabla \phi_j) \cdot \mathbf{n} ds \tag{5.9}$$

The following will represent in practice how the coefficient matrix 5.9 can be developed by the information obtained previously in the chapter. The notation $\partial V_i|_{f1}$ and $\partial V_i|_{f2}$ means the area or the length of that particular face we are considering. I.e. $\partial V_i|_{f1} = A_{f1}$ for the i'th control volume, and similarly $\partial V_i|_{f2} = A_{f2}$ as in the equations (5.4) and (5.5).

$$\int_{\partial V_i} (\kappa \nabla \phi_j) \cdot \mathbf{n} ds = \int_{f1} (\kappa \nabla \phi_j) \cdot \mathbf{n} ds + \int_{f2} (\kappa \nabla \phi_j) \cdot \mathbf{n} ds$$

$$\approx \kappa \nabla \phi_j \cdot \mathbf{n} \quad \partial V_i|_{f1} + \kappa \nabla \phi_j \cdot \mathbf{n} \quad \partial V_i|_{f2}$$

Here we have that $\partial V_i|_{f1}$ is the same as multiplying with A_{f1}, and similarly with $\partial V_i|_{f2}$. To gain the result below, we also use the definitions of the $\mathbf{n}$ with respect to x and y as in (5.6) and (5.7) respectively. The same applies for $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial y}$.

$$\kappa \nabla \phi_j \cdot \mathbf{n} \quad \partial V_i|_{f1} = \kappa_{f1}\frac{\partial \phi}{\partial x}\Delta \overline{y}_{f1} - \kappa_{f1}\frac{\partial \phi}{\partial y}\Delta \overline{x}_{f1}$$

$$= \kappa_{f1}[N_{1x}\phi_1 + N_{2x}\phi_2 + N_{3x}\phi_3]\Delta \overline{y}_{f1}$$

$$- \kappa_{f1}[N_{1y}\phi_1 + N_{2y}\phi_2 + N_{3y}\phi_3]\Delta \overline{x}_{f1},$$

$$\kappa \nabla \phi_j \cdot \mathbf{n} \quad \partial V_{ij}|_{f2} = \kappa_{f2}\frac{\partial \phi}{\partial x}\Delta \overline{y}_{f2} - \kappa_{f2}\frac{\partial \phi}{\partial y}\Delta \overline{x}_{f2}$$

$$= \kappa_{f2}[N_{1x}\phi_1 + N_{2x}\phi_2 + N_{3x}\phi_3]\Delta \overline{y}_{f2}$$

$$- \kappa_{f2}[N_{1y}\phi_1 + N_{2y}\phi_2 + N_{3y}\phi_3]\Delta \overline{x}_{f2}.$$

The above expressions can be generalized to all the nodes and their respective support nodes.

$$\int_{\partial V_i} \kappa \nabla \phi ds \approx -a(i,i)\phi_i + a(i,R_{i,j})\phi_{R_{i,j}} + a(i,R_{j+1})\phi_{R_{i,j+1}}, \qquad (5.10)$$

$$a_1^\kappa = -\kappa_{f1} N_{1x} \Delta \overline{y}_{f1} + \kappa_{f1} N_{1y} \Delta \overline{x}_{f1} - \kappa_{f2} N_{1x} \Delta \overline{y}_{f2} + \kappa_{f2} N_{1y} \Delta \overline{x}_{f2},$$
$$a_2^\kappa = \kappa_{f1} N_{2x} \Delta \overline{y}_{f1} - \kappa_{f1} N_{2y} \Delta \overline{x}_{f1} + \kappa_{f2} N_{2x} \Delta \overline{y}_{f2} - \kappa_{f2} N_{2y} \Delta \overline{x}_{f2},$$
$$a_3^\kappa = \kappa_{f1} N_{3x} \Delta \overline{y}_{f1} - \kappa_{f1} N_{3y} \Delta \overline{x}_{f1} + \kappa_{f2} N_{3x} \Delta \overline{y}_{f2} - \kappa_{f2} N_{3y} \Delta \overline{x}_{f2},$$

$$a(i,i) = a(i,i) + a_1^\kappa,$$
$$a(i,R_{i,j}) = a(i,R_{i,j}) + a_2^\kappa,$$
$$a(i,R_{i,j+1}) = a(i,R_{i,j+1}) + a_3^\kappa.$$

With a well design coefficient matrix, the task of developing a closed system of equations is within reach. The solution,

$$u_h = [u_1, u_2, \cdots, u_{M_1}], \qquad (5.11)$$

is what we want to approximate, but the right hand side

$$f_h = [f_1, f_2, \cdots, f_{M_1}], \qquad (5.12)$$

remains to calculate. The footnote M_1 refer to the internal nodes as in 2. The elements of the right hand side can be calculated by integration

$$f_i = \int_{V_i} f(x)dx. \qquad (5.13)$$

All the parts of a closed linear system is now at place. The coefficent matrix A_h and the right hand side together with a iterative solver will give an approximated solution of the problem.

5.2 Implementation of the Two Level Additive Schwartz Method

We assume we want to solve the system,

$$\mathbf{Au} = \mathbf{f}.$$

The main tasks is to restrict all the subdomains and develop a coarse grid corrector. Restriction matrices R_i is defined as a matrix which restricts the solution to a given subdomain. The idea is that if the coefficient A and the restriction matrix R_i is multiplied, and then ones more multiplied by the transpose of the restriction matrix, the result will be the coefficient matrix on the particular subdomain such that

$$\mathbf{A_i} = \mathbf{R_i A R_i^T} \tag{5.14}$$

If there are i different subdomains, there are i different smaller systems to solve. The sum of all these subdomains are actually a one level preconditioner

$$\mathbf{B} = \sum_i \mathbf{R_i^T A_i^{-1} R_i}. \tag{5.15}$$

The second main task is to construct the coarse grid correction. The solution of the coarse and fine grid is not in the same dimension. This is the purpose with the restriction operator $\mathbf{R_0}$. The operator $\mathbf{R_0}$ transform or averaging the solution from the coarse level to the fine level. We can formulate the coarse problem as follows,

$$\mathbf{A_0} = \mathbf{R_0 A R_0^T}. \tag{5.16}$$

A summation of equation (5.15) and (5.16) results in a two level additive Schwarz method such that the preconditioner is defined as

$$\overline{\mathbf{B}} = \mathbf{R_0^T A_0^{-1} R_0} + \sum_i \mathbf{R_i^T A_i^{-1} R_i}. \tag{5.17}$$

5.2.1   Restriction Matrices $\mathbf{R_i}$

The restriction matrix will in general never be constructed for the subdomains. A partitioning algorithm is usual used for this purpose. This type of algorithm is choosing out the favourably nodes, or optimizes the partitioning, such that we get a good partitioning which is consistent with solving the system with the desired accuracy and convergence. A simple partitioning can be one where the coarse space is defined, and the fine space is just a refinement of the coarse space. All the fine nodes in each subdomain defines the particular subdomain. It is often practical to have a overlap between the subdomains, such that the convergence of the method increases. When all the domains is defined by the fine nodes, a loop choosing out the correct nodes of the coefficient matrix is applied, such that there are a minor coefficient matrix, solving the system on the desired place of the domain, or

the actual subdomain. This is beneficial and suitable for parallel computing. The system solves all the small systems instead of the large one. A disadvantage of this is of course that the overall error is pretty large, but this is solved by generating the coarse grid correction. Although it is normal to formulate the local solver, or subdomain solver with this restriction matrices, it is in general never constructed.

5.2.2   Interpolation Matrix R_0

For developing the interpolation matrix from coarse space to the fine space, we use linear interpolation. We can represent the coarse space basis as the identity matrix, where each column in the matrix is an basis for the coarse space. The basis spans the whole coarse space. If we interpolate the basis functions from the coarse space to the fine space, we get an approximated changed new basis, thus this now applies in the fine space. The output after this interpolation is an array which represents an averaging. This matrix is called the interpolation matrix and can be multiplied with the solution of the coarse space problem. This gives an solution that applies in the fine space.

It is important how the interpolation is applied, and it is important that the triangular elements are accounted for. This means that the input of the interpolation has to contribute for the triangles. If we interpolates with wrong functions, e.i. interpolates such that the nodes forms squares instead of triangles. In this case the change of basis apply for squares and not triangle on the fine scale. Interpolation of the coarse problem should apply for triangles and not squares, this averaging will be wrong although it will be approximately correct. The moral is that it is important to be aware of this, and if a wrong basis change is applied, we have to take account for this in our analyse, and the error will be of an another order.

We denote the coarse grid with $2h$ and the fine grid with h. The mapping from the coarse grid to the fine grid can be expressed as follows. The transfer from fine grid to coarse grid and vice-versa is made possible by the inter grid operators namely restriction and interpolation respectively. The standard interpolation operator is a linear operator from $R^{(N/2)-1}$ to R^{N-1}. This map takes coarse grid vectors v^{2h} and produce fine grid vectors v^h, by the rule $I_{2h}^h v^{2h} = v^h$ defined by,

$$v_{2i}^h = v_j^{2h} \tag{5.18}$$

$$v_{2j+1}^h = \frac{1}{2}(v_j^{2h} + v_{j+1}^{2h}) \tag{5.19}$$

for $0 \leq j \leq \frac{N}{2} - 1$. In the fine grid Ω^h, the value of v^h at the even numbered grid points are directly transferred from Ω^{2h} where the values at the vector v^h at the odd numbered grid points is the average of the adjacent coarse grid values. For two dimensions, the linear interpolation is defined by,

$$v^h_{2i,2j} = v^{2h}_{i,j} \tag{5.20}$$

$$v^h_{2i+1,2j} = \frac{1}{2}(v^{2h}_{i,j} + v^{2h}_{i+1,j}) \tag{5.21}$$

$$v^h_{2i,2j+1} = \frac{1}{2}(v^{2h}_{i,j} + v^{2h}_{i,j+1}) \tag{5.22}$$

$$v^h_{2i+1,2j+1} = \frac{1}{2}(v^{2h}_{i,j} + v^{2h}_{i+1,j} + v^{2h}_{i,j+1} + v^{2h}_{i+1,j+1}) \tag{5.23}$$

The restriction operator is linear operator from R^{N-1} to $R^{(N/2)-1}$. This means that the fine grid vector is v^h taken such that it produces the coarse grid vector v^{2h}, i.e. the coarse grid only takes its values from the fine grid. The standard full weighting operator defined by $I^{2h}_h v^h = v^{2h}$, where

$$v^{2h}_j = \frac{1}{4}(v^h_{2j-1} + 2v^h_{2j} + v^h_{2j+1}), \qquad 1 \leq j \leq \frac{N}{2} - 1 \tag{5.24}$$

The full weighted restriction operator and the linear interpolation operator are transpose of each other. For two dimensional problems, the full weighting operator is given by the rule, $I^{2h}_{ij} v^h = v^{2h}$, and is defined by

$$v^{2h}_{ij} = \frac{1}{16}[(v^h_{2i-1,2j-1} + v^h_{2i-1,2j+1} + v^h_{2i+1,2j-1} + v^h_{2i+1,2j+1}) \tag{5.25}$$

$$+ 2(v^h_{2i,2j-1} + v^h_{2i,2j+1} + v^h_{2i,2j+1} + v^h_{2i-1,2j} + v^h_{2i,2j-1}) + 4v^h_{2i,2j}] \tag{5.26}$$

$$\text{for} \quad 1 \leq i,j \leq \frac{N}{2} - 1. \tag{5.27}$$

As an illustration, we give a simple example. The restriction operator applied on a (1×7) vector produces a (1×3) vector, where $Rw = w_0$

$$R = \begin{bmatrix} 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 \end{bmatrix}, \quad w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \end{bmatrix}, \quad w_0 = \begin{bmatrix} w_1 + 2w_2 + w_3 \\ w_3 + 2w_4 + w_5 \\ w_5 + 2w_6 + w_7 \end{bmatrix}$$

The basis of for the null space of the restriction operator $N(I^{2h}_h)$ is: $(2, -1, 0, 0, 0, 0, 0)$, $(0, -1, 2, -1, 0, 0, 0)$, $(0, 0, 0, -1, 2, -1, 0)$, $(0, 0, 0, 0, 0, -1, 2)$. The interpolation operator on a (1×3) vector produces a (1×7) vector: $R'v = \frac{1}{2}v_0$ where

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix}, \quad w = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}, \quad w_0 = \begin{bmatrix} \frac{v_1}{2} \\ v_1 \\ \frac{v_1+v_2}{2} \\ v_2 \\ \frac{v_2+v_3}{2} \\ v_3 \\ \frac{v_3}{2} \end{bmatrix}$$

The basis for $R(I_{2h}^h)$ is nothing but the column vectors of the interpolation which represents the so called hat function or triangular function. The use of linear interpolation in multi grid schemes corresponds to a representation of the solution in terms of piecewise linear hat scaling functions.

5.2.3 Important Aspects of the Implementation of the Two Level Additive Schwarz Method

We will now in more simple terms explain the dynamic of how such a method is implemented. The idea here is to show what lies in the two terms in (5.17). We start out by defining our coarse problem,

$$\mathbf{A_H u_H = f_H} \tag{5.28}$$

where $\mathbf{A_H}$ is an $(m \times m)$ matrix, $\mathbf{u_H}$ and $\mathbf{f_H}$ is vectors of size m which is also the number of nodes in the problem. H defines the largest side of one triangle K_H in the discretization. The coarse problem, or the coarse discretization makes a foundation for the subdomains. $\mathbf{K_H}$ defines the triangles on coarse scale level. Each triangle made on the coarse grid, is defined as one subdomain $\mathbf{S_i}$. We define the boundary of a $\mathbf{S_i}$ as $\mathbf{S_i^B}$. Further the whole domain is refined such that we get a new system,

$$\mathbf{A_h u_h = f_h} \tag{5.29}$$

where $\mathbf{A_h}$ is an $n \times n$ matrix, $\mathbf{u_h}$ and $\mathbf{f_h}$ is vectors of size n which is the number of nodes in the refined version of the problem. As in (5.28), h in (5.29) defines the largest sides of the triangle $\mathbf{K_h}$ that makes out the discretization. The task now is to determine what triangle of the fine scaled problem (5.29) that lies in the coarse scale problem (5.28) such that the subdomains $\mathbf{S_i} = \{\mathbf{K_h} \in \mathbf{K_H}\}$. For gaining an overlap, simply define that for all $\mathbf{K_h}$ on $\mathbf{S_i^B}$, all $\mathbf{K_h}$ that share a common node with $\mathbf{S_i^B}$ should also be included in $\mathbf{S_i}$. If this is done several times, an overlap of desired size can be obtained. $\mathbf{S_i^N} = \{n \in \mathbf{S_{K_H}}\}$ and are used to determine the sub problems

$$\mathbf{A_{S_i^N} u_{S_i^N} = f_{S_i^N}}, \tag{5.30}$$

where $\mathbf{A_{S_i^N}}$ is a $(p \times p)$ matrix consisting of the elements in, $\mathbf{S_i^N}$, $\mathbf{u_{S_i^N}}$ and $\mathbf{f_{S_i^N}}$ is vectors of size p which is the number of nodes in $\mathbf{S_i^N}$. The system (5.30) is solved such that

$$\mathbf{u_{S_i^N}} = \mathbf{A_{S_i^N}^{-1}f_{S_i^N}} \tag{5.31}$$

for every subdomain $\mathbf{S_i}$. This creates a solution $\mathbf{u_{S_i^N}}$ which is a vector of size p. We define a new solution vector $\mathbf{v_h}$ which are of the same size as the fine scaled problem (5.29). We map the solution $\mathbf{u_{S_i^N}}$ from the sub problem to the great problem such that $\mathbf{Z} : \mathbf{u_{S_i^N}} \to \mathbf{v_h}(\mathbf{u_{S_i^N}})$. The vector $\mathbf{v_h}(\mathbf{u_{S_i^N}}$ is the term $\sum_{\mathbf{i}} \mathbf{R_i^T A_i^{-1} R_i}$ of the preconditioner $\mathbf{B}$. Further we solve the the coarse problem (5.28) such that,

$$\mathbf{u_H} = \mathbf{A_H^{-1}f_H} \tag{5.32}$$

The idea is that we add the two solutions $\mathbf{v_h}$ and $\mathbf{u_H}$ together, such that we get the two level preconditioner. This is not possible, and before this can be done, a interpolation of the solution $\mathbf{u_H}$ onto $\mathbf{v_h}$. We define triangular linear interpolation such that $\mathbf{I_H^h u_H} = \mathbf{u_h}$ approximately. In practice this interpolation is produced by define a identity matrix $\mathbf{I}$ of size $(m \times m)$ where the elements on the diagonal forms the basis of the nodes m at their actual position as in the discretization of equation (5.28). Then the triangular linear interpolation is applied to each column of $\mathbf{I}$. From each interpolation we get a "new" fine scaled basis of size n approximating the "old" coarse basis of size m. The interpolation $\mathbf{I_H^h}$ is a matrix of size $(m \times n)$, and are actually the interpolating matrix $\mathbf{R_0}$ from equation (5.17). The preconditioner B in equation (5.17) can be used together with the GMRES method as a matrix, or as an vector matrix product such that,

$$\mathbf{u} = \mathbf{I_H^h u_H} + \sum_{\mathbf{i=1}}^{\mathbf{m}} \mathbf{u_{S_i^N}} \tag{5.33}$$

Chapter 6

Convergence Rate of the CVFEM Preconditioned with Two Level Additive Schwarz method

In this section we present an analytical estimate of the rate of convergence of the combined method of the CVFEM and the two level additive Schwarz method as a preconditioner. The results in [9] are based of inexact solver unlike the theory presented here. In this analysis we have developed the theory to apply for exact solvers. Still, the theory follows the lines from [9], and the theorem and lemmas mentioned in this section, will not be proved. These are available in the same article. The theory will be tested with different triangulations and discretizations, both regular and irregular triangulation in chapter 7. The idea is to use the GMRES to solve (4.28), and to determine the coefficient c_p and C_p in the equations (3.18). First we have to define a proper inner product. In the analysis done in the next section we choose this inner product to be

$$[v, w] = \tilde{a}(v, w) = \sum_{K \in \tau_h} \text{meas}(K)(A(Q_K)\nabla v) \cdot \nabla w, \qquad (6.1)$$

where Q_K denote the barycentre of the finite element K. The benefits related to this choice of inner product is that it preserves good convergence properties. The task is then to solve (4.28) through the inner product (6.1).

47

6.1 Estimate of Convergence Rate

The first lemma introduced shows that the different bilinear forms are approximately equal.

Lemma 1 *For the bilinear forms $a(\cdot,\cdot)$, $a_h(\cdot,\cdot)$ and $\tilde{a}(\cdot,\cdot)$, the following estimates hold:*

$$|a(v,w) - \tilde{a}(v,w)| \leq h||v||_a||w||_a, \quad \forall v,w \in S_0^h, \tag{6.2}$$
$$|a(v,w) - a_h(v,w)| \leq h||v||_a||w||_a, \quad \forall v,w \in S_0^h. \tag{6.3}$$

where $||v||_a = a(v,v)^{1/2}$.

We can prove the first inequality of lemma 1 with the definitions of $a(\cdot,\cdot)$ and $\tilde{a}(\cdot,\cdot)$ and Friedrichs and Cauchy-Schwartz inequalities. The second inequality are more complex and refer to [9] for the proof. Thus an immediate consequence of lemma 1 is lemma 2.

Lemma 2 *There exists a positive constant $H_0 > 0$, such that for $h < H_0$, we have*

$$||v||_a \simeq ||v||_{\tilde{a}} \simeq ||v||_{a_h} \simeq ||v||_{1,\Omega}, \quad \forall v \in S_0^h, \tag{6.4}$$

where $||v||_{\tilde{a}} = \tilde{a}(v,v)^{1/2}$, $||v||_{a_h} = a_h(v,v)^{1/2}$.

For lemma 2 to hold, we assume from now that the terms h and H satisfy $h < H < H_0$. The same principle yields for the bilinear form of the coarse mesh, such that if we replace h with H in lemma 1 we get lemma 3.

Lemma 3 *For the bilinear form $a_H(\cdot,\cdot)$, we have the estimate*

$$|a_H(v,w) - a(v,w)| \lesssim H||v||_a||w||_a, \quad \forall v \in S_0^H. \tag{6.5}$$

Lemma 4 *For the operators T_i, $i = 0,1,\cdots,N$, we have*

$$||T_i v||_a \lesssim ||v||_a, \quad \forall v \in S_0^h, \quad i = 0,1,\cdots,N. \tag{6.6}$$

For the proof of lemma 4 lemmas we need the definitions of T_i and T_0. Consider T_i and T_0 separately and deduce from that base.

Lemma 5 *For all $v \in S_0^h, i = 0,1,\cdots,N$, we have*

$$a_h(v, T_i v) \gtrsim \tilde{a}(T_i v, T_i v), \tag{6.7}$$
$$||T_i v||_{a_h} \lesssim ||v||_{\tilde{a}}, \tag{6.8}$$
$$a_h(T_i v, T_i v) \gtrsim \tilde{a}(T_i v, T_i v). \tag{6.9}$$

The lemma can be proved through the definition of $a_h(\cdot, \cdot)$ and lemma 2 and lemma 4. The second inequality follows from lemma 3. The third inequality are also deduced with the same lemmas and definitions.

Lemma 6 *Let the linear operator T_i be defined as in (4.20). Then for all $v \in S_0^h$, we have*

$$\tilde{a}(T_i v, v) \geq C_3 \tilde{a}(T_i v, T_i v) - C_4 h \tilde{a}(v, v), \tag{6.10}$$

where C_3 and C_4 are two generic positive constants independent of H and h.

The lemma can be proved with the help of lemma 1,2,4 and 5. We further use some existing results for the two level additive Schwarz methods for symmetric problems such as in [18]. For $i \leq i, j \leq N$, define $\varepsilon_{ij} \in [0, 1]$ to be the minimal value that satisfy

$$a(v_i, v_j) \leq \varepsilon_{ij} a(v_i, v_i)^{1/2} a(v_j, v_j)^{1/2}, \qquad v_i \in S_i^h, v_j \in S_j^h. \tag{6.11}$$

We define $\rho(\varepsilon)$ to be the spectral radius of ε. It is given in [18] that $\rho(\varepsilon)$ is bounded above by a constant independent of H and h. Moreover, let $\alpha > 0$ be the minimal constant such that for all $v \in S_0^h$ there exists a representation $v = \sum_{i=0}^{N} v_i$, with $v_0 \in S_i^H$, $v_i \in S_0^h$, $i = 1, \cdots N$, such that

$$\sum_{i=0}^{N} a(v_i, v_i) \leq \alpha^2 a(v, v). \tag{6.12}$$

If $\{\Omega_i\}$ has a generous overlap, α can be bounded above independent of H and h, and if $\{\Omega\}$ has a small overlap, α can be bounded by above with $C(1 + \frac{H}{h})^{1/2}$. From (2) and (6.11) the result leads to a new lemma

Lemma 7 . *There exists a positive constant C_5 such that for $i \leq i, j \leq N$ the following estimate holds:*

$$a(v_i, v_j) \leq C_5 \varepsilon_{ij} a(v_i, v_i)^{1/2} a(v_j, v_j)^{1/2}, \qquad v_i \in S_i^h, v_j \in S_j^h. \tag{6.13}$$

Lemma 7 is used in the next lemma which are again used to full fill the convergence of the method.

Lemma 8 *There exists a positive constant C_6, such that for all $v \in S_0^h$,*

$$\sum_{i=0}^{N} \tilde{a}(T_i v, T_i v) \geq C_6 \alpha^{-2} \tilde{a}(v, v). \tag{6.14}$$

The problem is solved by the GMRES method trough the inner product $\tilde{a}(\cdot, \cdot)$, thus (3.18) becomes

$$c_p = \inf_{v \in S_0^h \backslash 0} \frac{\tilde{a}(Tv, v)}{\tilde{a}(v, v)} \quad \text{and} \tag{6.15}$$

$$C_p = \sup_{v \in S_0^h \backslash 0} \frac{\tilde{a}(Tv, Tv)}{\tilde{a}(v, v)}, \tag{6.16}$$

where $T = \sum_{i=0}^{N} T_i$ is given in (4.20). To obtain the convergence rate estimate, we must calculate c_p and C_p. This is done with lemma 6, 7 and 8, and the results are similar as in [18].

Lemma 9 *For the constants c_p and C_p defined in (6.15), we have the estimates*

$$c_p \geq C_3 C_6 \alpha^{-2} - C_4(1 + N)h, \tag{6.17}$$

$$C_P \geq C_7^2(1 + \rho(\varepsilon) + h)^2, \tag{6.18}$$

where C_7 is a genetric positive constant independent of h, H and α.

With lemma 9 we can conclude with the final theorem stating under certain conditions that we have good convergence properties. The theorem are given and proved in [9] and will be tested numerically in the next section.

Theorem 4 *There exist two positive constants $H_0 > h_0 > 0$ such that the GMRES method for solving problem (4.20) with respect to the inner product (6.1) is convergent as $h < h_0$ and $H < H_0$. Furthermore, If Ω_i has generous overlap, the convergence rate is optimal, i.e. independent of H and h.*

Chapter 7

Numerical Results

In this chapter we will test the theory described in the past chapters. The main focus will be to investigate how the CVFEM preconditioned with a two level additive Schwarz method works relative to other preconditioners, and test the theorem described in 4. We will look at heterogeneous problems with antisymmetric coefficient matrix such as generated by the CVFEM. We will both look at uniform and non uniform discretizations for the purpose to test the method more extensively. We will look how the convergence change with increasing overlap, and study the impact of different size of subdomains. The discretization error will be measured in L^2 and H_1 norms.

7.1 Reference Solution

Because of the lack of an exact solution in heterogeneous cases, we have to compare the convergence with an estimate of the exact solution. An estimate of the exact solution is here provided by solving the problem on a very fine scale, and using this solution as an "exact". We then have to interpolate the solution we get from the approximated solution to the fine reference solution. Obviously under this interpolation errors are introduced, but it should be the same every time we compute the norm, and we should recognize the error in the experiments.

7.2 Numerical Experiments on Regular Mesh

The first part of the numerical experiments are carried out on regular triangles. It can be interesting to investigate how different types of discretizations and how this affect the results. A regular discretization is often the first step in $2d$ implementation of elliptic partial differential equations.

7.2.1   The Problem

The numerical experiments in the rest of this section are carried out on the following equation

$$\begin{cases} -\nabla \cdot (D\nabla u) = f & \text{in } \Omega := (-1, 1) \\ \qquad\quad u = 0 & \text{on } \partial\Omega, \end{cases} \tag{7.1}$$

and the right hand side f is chosen to be

$$f = -2\pi^2 \sin(\pi x) \sin(\pi y), \tag{7.2}$$

where $x, y \in \Omega \bigcup \partial\Omega$ and D is the diffusion coefficient. The diffusion coefficient is set to be a function of the discretization. We apply this coefficient when we assemble the coefficient matrix, and to ensure that we get an antisymmetric problem, we let $D(z_i) = e^{x(z_i)}$. If we let B_h be the coefficient matrix we get an antisymmetry of order $0.1 - 0.01$ on the majority the of the off-diagonal elements. This is of coarse also dependent on the triangulation. This is one of the major reasons for testing the theory nummerically on both regular and irregular triangles. Since we choose the barycentre as the point in the elements that conforming the faces of each control volume, we ensure that we get an antisymmetric matrix. It is well known that this theory applies on symmetric cases, but for unsymmetrical cases it is not as proven.
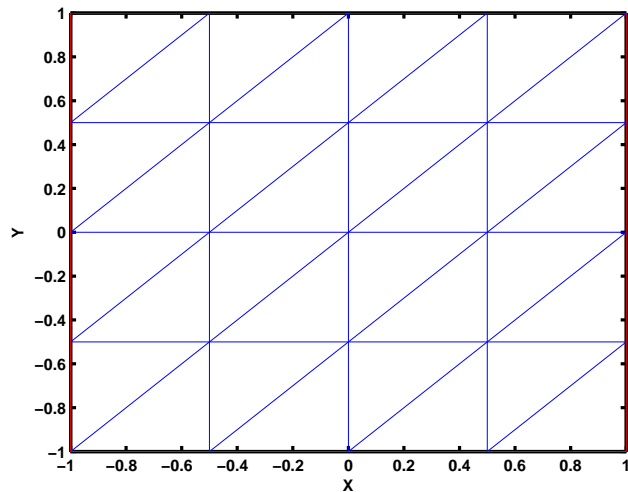
Figure 7.1: Typical coarse mesh used as preconditioner in the two level additive Schwarz method with regular triangles.
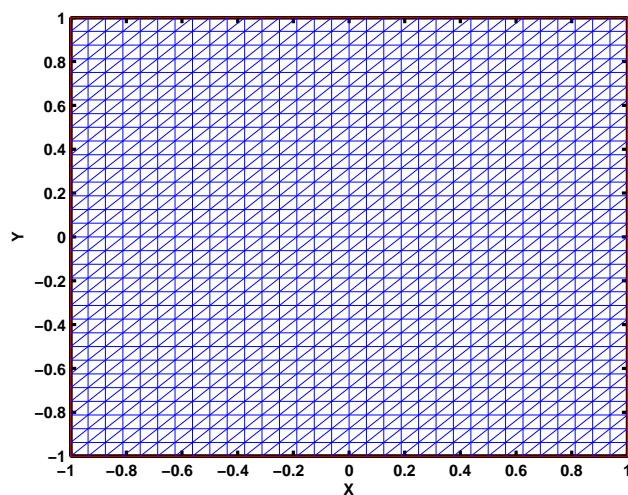

Figure 7.2: Here we see a refined version of the discretization in figure 7.1. In each triangle of the coarse grid we refine the triangle, this constitutes one subdomain of the preconditioner and shows the fine mesh discretization.
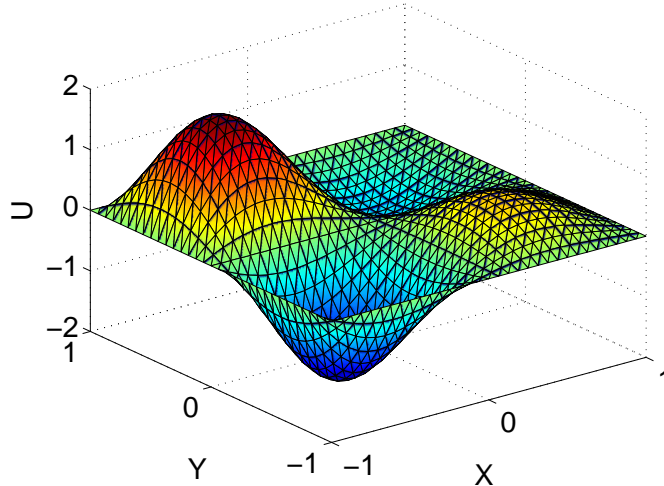
Figure 7.3: Solution of the test problem with regular triangulation

Error Estimate Regular Mesh

Table 7.1:

Unknowns	L_2 error	H^1 error	ROC L_2/h^2	ROC H_1/h
49	1.19150e-2	2.18016	2.43163e-4	3.11452e-1
225	4.10050e-4	0.38938	1.82245e-6	2.59593e-2
961	1.95947e-5	7.14862e-2	2.03899e-8	2.30601e-3
3969	1.07211e-6	1.21620e-2	2.70123e-10	1.93048e-4
16129	6.47259e-8	1.38927e-3	4.01302e-12	1.09391e-5

We observe that the error in the L_2 norm has a $O(h^2)$ behaviour with a constant term C from equation (2.46) approximately $\approx 10^{-2}$. The reason why we see this scaling is because of the interpolation we do when we compare the reference solution with the approximated solution. For the H^1 error estimate we see a convergence rate of order $O(h)$. Here the constant C in equation (2.45) is $\approx 10^{-1}$. This shows that the implementation of the CVFEM method is correct, and the error estimate is consistent with the theory as in section 2.3.1.

7.2.2 Convergence of the CVFEM Preconditioned with an Overlapping Two Level Additive Schwarz Method on Regular Mesh

In this section we look how the two level additive Schwarz method on regular mesh. We check the convergence when the subdomains are held, but with varying fine discretization. Another experiment involves holding the fine discretization but varying the subdomains and the coarse mesh. We look at the recommended situation where the coarse grid and fine grid are proportional to each other. To show the good convergence properties of the two level additive Schwarz method we compare the method against a couple of common preconditioners. In the rest of this we denote the space discretization of the coarse grid with H and the corresponding fine discretization with h. The notation for a generous overlap is $O(H)$ and for a small overlap we use $O(h)$.

Unchanged H with Varying h

Table 7.2:

Unknowns	SUBD	TRI TOT	TRI/SUB	ITR $O(h)$	ITR $O(H)$
121	50	200	4	16	14
441	50	800	16	18	14
1681	50	3200	64	22	15
6561	50	12800	256	27	16
25921	50	51200	1024	37	17

Here we clearly observe the independence of h in the generous overlap case. We conclude with a constant behaviour although we see small variations in the results for the $O(H)$ overlapping. The differences in the result occurs because of the nature of the GMRES. The GMRES only approximate the solution and can give varying results. With $O(h)$ overlap the method converge dependent of number of unknowns.

Change H, with h Unchanged

Table 7.3:

Unknowns	SUBD	TRI TOT	TRI/SUB	ITR $O(h)$	ITR $O(H)$
289	8	512	64	17	9
1681	50	3200	64	22	15
3249	98	6272	64	21	15
6561	200	12800	64	21	15
14641	450	28800	64	20	15
40401	1250	80000	64	19	14

In table 7.3 we fix the fine grid discretization and vary the coarse grid discretization. We see that both cases seem to converge independently of the mesh. This behaviour occurs since the coarse grid becomes more an more as a discretization itself, and not a smoothing factor. This is consistent with theorem 4.

Small and Generous Overlap with H and h Proportional

Table 7.4:

Unknowns	SUBD	TRI TOT	TRI/SUB	ITR $O(h)$	ITR $O(H)$
25	8	32	4	6	5
169	18	288	16	16	12
2401	72	4608	64	21	15
37249	288	73728	256	25	16

For the case where the subdomain and coarse discretization are proportional we observe that the convergence is constant for the $O(H)$ overlap, while it is dependent on the mesh with $O(h)$ overlap. When the number of subdomains and the number of triangles per subdomain are comparable, we have the optimal case for the Schwarz methods. In the view of parallization this is the best situation, as the subdomain and the coarse grid use the same amount of time to solve the problem.

Comparison Between Preconditioners

Table 7.5:

Unknowns	PLAIN	DIAG	LUI	ASM $O(h)$	ASM $O(H)$
25	9	9	6	6	5
169	55	35	19	16	12
2401	242	120	68	21	15
37249	941	437	242	25	16

Table 7.5 should be viewed together with figure 7.4. We see some simple, but usual preconditioners compared with the additive Schwarz methods. The GMRES without any preconditioner is also displayed. What we see at once is the low number of iterations the Schwarz methods use to get within the desired tolerance of the residual.
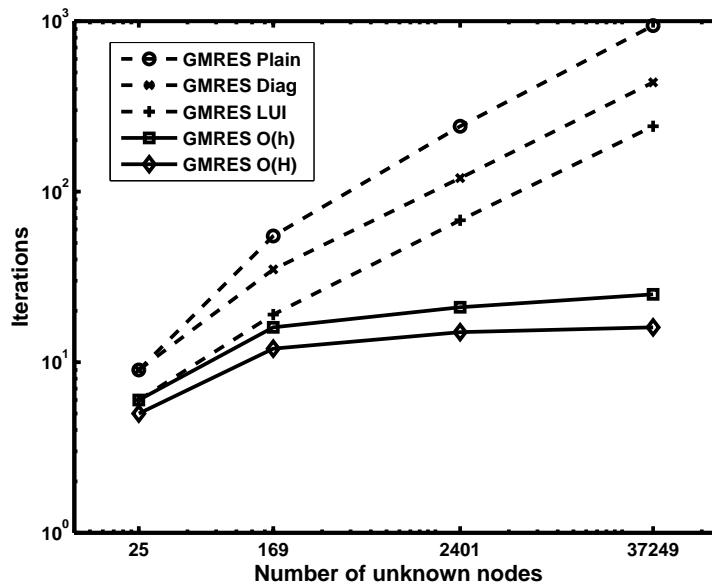

Figure 7.4: Comparing between the different preconditioners. GMRES Plain means no preconditioning at all, GMRES diag means a simple diagonal pre-conditioner, GMRES LUI means an incomplete lower-upper decomposition of the coefficient matrix as preconditioner and GMRES $O(h)$ and GMRES $O(H)$, means two level additive Schwarz method with respectively small and generous overlap.

7.3 Numerical Experiments on Unstructured Mesh

In this section we will study the behaviour of the solution of equation (7.2.1) when the discretization is unstructured. With the non-uniform structure it is interesting to see if this gives better or worse convergence conditions. First we look at the discretization error of the CVFEM.
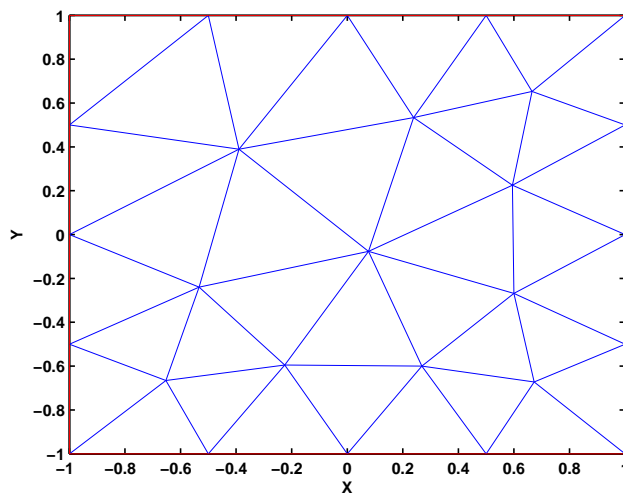

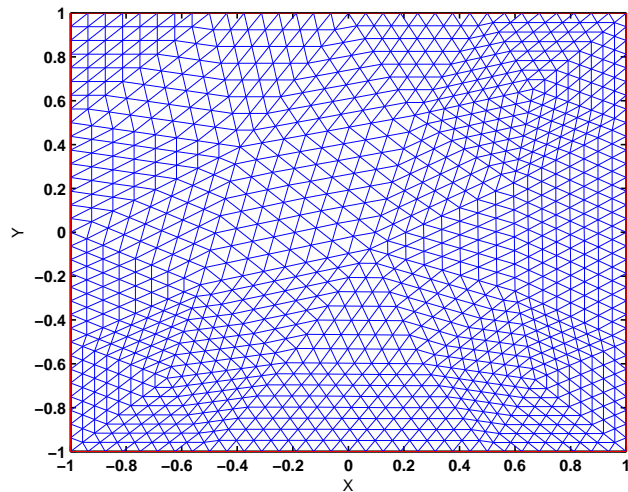Figure 7.5: Discretization of the coarse mesh

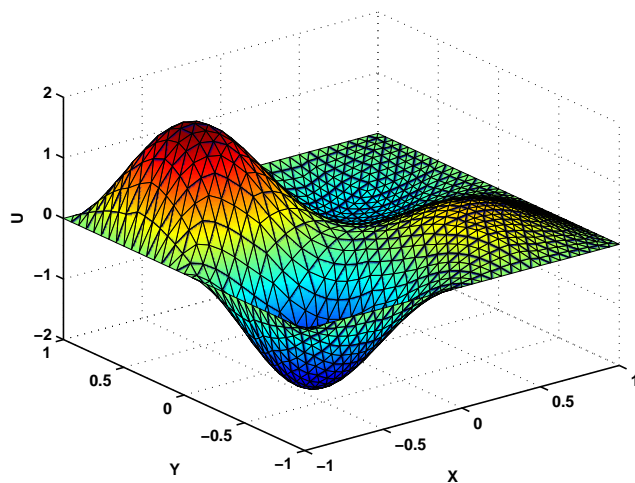Figure 7.6: The corresponding fine mesh of the coarse mesh

Figure 7.7: Solution of the of equation (7.2.1) with unstructured discretization where f is equal to equation (7.2)

Error Estimate on Unstructured Mesh

Table 7.6:

Unknowns	L_2 error	H^1 error	L_2/h^2	H^1/h
145	5.13586e-3	4.16804e-1	3.54198e-05	3.46137e-2
545	6.40902e-4	9.73264e-2	1.17597e-06	4.16901e-3
1601	3.56757e-05	2.34991e-2	2.22834e-08	5.87293e-4
6913	2.00948e-06	3.16133e-3	2.90681e-10	3.80222e-05
28097	1.21519e-07	2.32436e-4	4.32498e-12	1.38667e-06

We see the same behaviour here as for the regular triangulation. We have a $O(h^2)$ order of the convergence in the L_2 norm. We still see that the error is multiplied with a constant. We do not see the perfect match as in the regular case. This is because the triangles do not have the same shape over the whole domain. Especially we see this when the domain is roughly discretisized. We do observe a $O(h)$ behaviour in the H^1 norm as well.

7.3.1 Convergence of the CVFEM Preconditioned with an Overlapping Two Level Additive Schwarz Method on Unstructured Mesh

In this section we will investigate numerically the theory described in this thesis. The case when we have an unstructured mesh is more general than a regular, and since the coefficient matrix is dependent on the discretization it is important to look at more general cases. We look at how the numerical result are compared against the theory and also look at how this method behaves compared to other well-known preconditioners. The first case we consider is when the subdomains and coarse problem is proportional.

The Case When the Subdomain and Coarse Mesh is Proportional

Table 7.7:

Unknowns	SUBD	TRI TOT	TRI/SUB	ITR $O(h)$	ITR $O(H)$
13	4	16	4	1	1
145	16	256	16	11	9
2449	74	4736	64	18	14
32801	254	65024	256	23	15
134353	518	267456	520	24	15

In table 7.7 we see a clear independence of the mesh in the case with generous overlap , it is on the other side dependent of the discretization in the case of small overlap on each subdomain. Although the convergence of the method is significantly better and stabilizes after only a few iterations when the overlap is of order $O(H)$ it is not sure to be the right choice of overlap. It is important to take account the increasing computational time an $O(H)$ overlap implies.

Varying H, with h Unchanged

Table 7.8:

Unknowns	SUBD	TRI TOT	TRI/SUB	ITR $O(h)$	ITR $O(H)$
145	4	256	64	8	1
545	16	1024	64	14	10
2449	74	4736	64	18	14
8273	254	16256	64	18	14
13441	414	26496	64	18	14
44033	1366	87424	64	17	13

We see in the generous overlap case that we have a convergence which is constant. The discretization does not affect how many iterations the GM-RES uses. This is a behaviour that is consistent with theorem 4 in chapter 6. The convergence is not dependent on the mesh at all. In the first two iteration we get better convergence than for the larger problems. The reason for this is that there are few enough discretization points that the overlap almost stretches over the whole domain. In fact in the first row the overlap stretches over the whole domain and we obtain the desired tolerance after just one iteration. In row two, the overlap is still large compared to the domain, and we see better convergence properties for that reason.

For the case when we have a small overlap we see that the method stabilizes when we increase the coarse grid and fix the fine discretization. We see good convergence properties because the coarse grid catches most of the characteristics of the solution and we do not need so many iterations before we get the residual within the desired tolerance.

Varying h, with Unchanged H

Table 7.9:

Unknowns	SUBD	TRI TOT	TRI/SUB	ITR $O(h)$	ITR $O(H)$
53	20	80	4	12	12
185	20	320	16	16	12
689	20	1280	64	19	13
2657	20	5120	256	25	14
10433	20	20480	1024	35	14
41345	20	81920	4096	47	15

In table 7.9 we see a constant behaviour of the convergence when we have a generous overlap, consistent with the theory. We also see with an small overlap the dependence of the mesh. When we increase the number of fine nodes, we clearly see the dependence of the grid unlike when we have a generous overlap where we see the constant behaviour.

Comparison Between Preconditioners

Table 7.10:

Unknowns	PLAIN	DIAG	LUI	ASM $O(h)$	ASM $O(H)$
13	2	2	4	1	1
145	37	26	17	11	9
2449	277	173	72	18	14
32801	1124	695	281	23	15
134353	-	-	-	24	15

This table 7.10 should be seen together with figure 7.8. Here we see the good convergence properties of the two level additive Schwarz method, both for small and generous overlap. While all the preconditioners we have compared the method with are clear off the mark, the additive Schwarz method with both small and generous overlap is relatively stable. In fact the case where we have generous overlap we have that the number of iterations needed to be within the tolerance of the residual is bounded by a constant. For the case where we have a small overlap it is dependent on the mesh, but the properties are still very good compared to other the other preconditioners in the table 7.10.
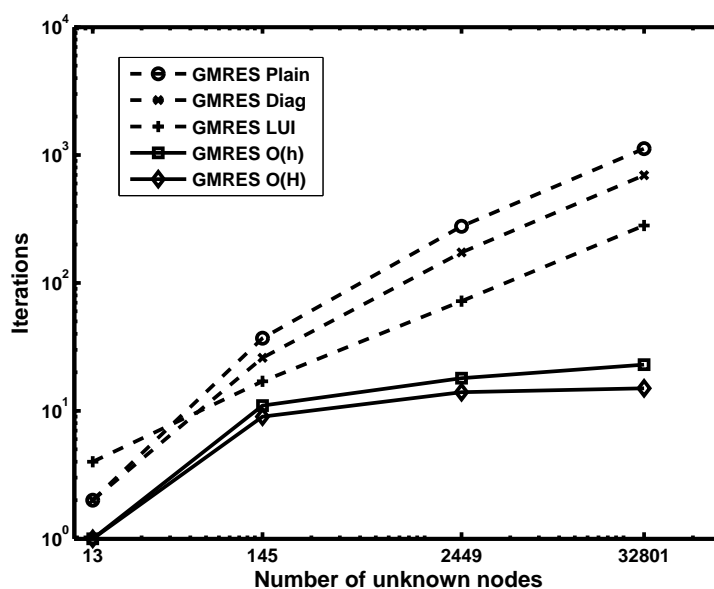
Figure 7.8: Comparing between the different preconditioners. GMRES Plain means no preconditioning at all, GMRES diag means a simple diagonal preconditioner, GMRES LUI means an incomplete lower-upper decomposition of the coefficient matrix as preconditioner, and GMRES $O(h)$ and GMRES $O(H)$, means two level additive Schwarz method with respectively small and generous overlap.

Chapter 8

Summary and Conclusion

In this chapter we summarize and conclude this thesis, and also give some indication of what interesting things to do about this topic in the future.

8.1 Summary

In this thesis we have looked at the steady state diffusion equation and applied it to a the CVFEM method. We have also looked at a domain decomposition algorithm to precondition the system of equations that arises. The CVFEM is an applicable method for solving elliptic partial differential equations. The fact that it is locally conservative on each control volume, and that it can tesselate any planar surface because, makes this method robust. We have seen how the method is built up, and explained important concepts.

Because the CVFEM generates an unsymmetric coefficient matrix we have looked at a solver suitable for solving unsymmetric linear problems. We have seen that the GMRES is such a method, and explained the basic concepts around how the algorithm works. In addition we have explained the basics of preconditioning and condition number.

Since the coefficient matrix that arises form the CVFEM often has a large condition number or is ill conditioned, we have looked at a domain decomposition method to precondition the system. Especially we have looked at two level additive Schwarz methods, both with a small and generous overlap. We have gone through some theory regarding the general Schwarz methods, and also given a detailed view of the abstract Schwarz framwork for a two level additive Schwarz method.

Further we have showed how we may assemble the coefficient matrix that arises from the CVFEM, We have also discussed some important aspects of how the two level additive Schwarz method can be implemented.

A theoretical estimate of the convergence of the Schwarz framework together with the GMRES has been shown without proof and tested thoroughly. We have also seen how the two level additive Schwarz method provides compared to other preconditioners.

8.2   Conclusion

In this thesis we have seen how the control volume finite element method pre-conditoned with a two level additive Schwarz method performs numerically. We have seen that the discretization error of the CVFEM converges with order $O(h^2)$ in the L_2 norm, both for regular and irregular triangulations. In the H^1 norm it also performs as expected, as the error converges with a rate of order $O(h)$. The focus has not been on discretization errors but more on the convergence of the preconditioned system.

If we compare the results with the theory there has been no great difference. The numerical result shows that if there is a generous overlap on each subdomain, the method is independent on the mesh as theorem 4. For the example described in chapter 7 we have seen that the method stabilizes around 15 iterations for small enough h and H. A two level additive Schwarz method with generous overlap has very good convergence properties. The second best preconditioner tested here, the LUI decomposition, is outperformed by huge factors when the discretization is refined.

For the case with small overlap, the theory states that it converges, but dependent on the mesh size H and h. This is exactly what we see, both for regular and unstructured mesh. Still, the method converges significantly better than the preconditioners it is compared against. We can conclude with that the theory match the numerical results. The method is robust and converges after relatively few iterations. These types of domain decomposition methods are tailor-made for parallization.

There are benefits and drawbacks with both cases. For the case with generous overlap we have that the convergence is bounded by a constant and independent of the mesh size. The drawbacks is that the linear system to solve at each subdomain is larger and the computational time to solve the

subdomains increase. For the case with small overlap we do not have a fixed convergence rate. Still, the method with small overlap has outstanding convergence properties. There is a fine nuance between what kind of overlap to choose.

Overall the two level additive Schwarz method as preconditioner on the CVFEM with both generous and small overlap performs well and according to the theory described in the thesis. The method is light years ahead regarding convergence compared with other regular preconditoners. We have seen no greater difference between regular and unstructured discretization.

The method is tested against the most common discretizations regarding triangles, and the results are clear. There is no sense in testing the theory against discretization techniques that are not in use, or not beneficial. In the introduction we mentioned that the domain decomposition as preconditioning for control volume methods is not common. Throughout this thesis we have seen that this actually is a robust and convenient method on the CVFEM. The problems that arise with the unsymmetric coefficient matrix is solved by using the GMRES as an iterative solver. In fact the CVFEM preconditioned with the two level additive Schwarz method performs very well, and there are no good reasons not using domain decomposition on such methods. Actual in the light of the results we recommend the two level additive Schwarz method as preconditioner for the CVFEM. Further testing will indicate if other domain decomposition methods are suitable as preconditioner for the CVFEM and other control volume methods.

Domain decomposition on control volume methods is not widespread. Further work along this line may be testing and develop theory around other domain decomposition methods in this framework.

Bibliography

[1] I. Aavatsmark and U. i Bergen. Bevarelsesmetoder for elliptiske di eren-sialligninger.

[2] BR Baliga and SV Patankar. A new finite-element formulation for convection-diffusion problems. *Numerical Heat Transfer, Part A: Applications*, 3(4):393–409, 1980.

[3] D. Braess. *Finite elements: Theory, fast solvers, and applications in solid mechanics.* Cambridge Univ Pr, 2001.

[4] Z. Cai, J. Mandel, and S. McCormick. The finite volume element method for diffusion equations on general triangulations. *SIAM Journal on Numerical Analysis*, 28(2):392–402, 1991.

[5] P. Chatzipantelidis and RD Lazarov. Error estimates for a finite volume element method for elliptic PDEs in nonconvex polygonal domains. *SIAM Journal on Numerical Analysis*, 42(5):1932–1958, 2005.

[6] K. Chen. *Matrix preconditioning techniques and applications.* Cambridge Univ Pr, 2005.

[7] Z. Chen, G. Huan, and Y. Ma. *Computational methods for multiphase flows in porous media.* Society for Industrial Mathematics, 2006.

[8] W. Cheney, EW Cheney, and W. Cheney. Analysis for applied mathematics. 2001.

[9] SH Chou and J. Huang. A domain decomposition algorithm for general covolume methods for elliptic problems. *Journal of Numerical Mathematics*, 11:179–194, 2003.

[10] S.H. Chou and Q. Li. Error estimates in L2, H1 and L in covolume methods for elliptic and parabolic problems: a unified approach. *Math. Comp*, 69(229):103–120, 2000.

[11] R.E. Ewing, R.D. Lazarov, T. Lin, and Y. Lin. Domain decomposition capabilities for the mortar finite volume element methods. In *11-th International Conference on Domain Decomposition Methods in Science and Engineering, London,(CH Lai, PE Bjorstad, M. Cross, and O. Wtdlund, Eds.), DD Press, Bergen*, pages 220–227. Citeseer, 1999.

[12] P.B. Hansen. Conjugate gradient solution of linear equations. *Concurrency: Practice and Experience*, 10(2):139–156, 1998.

[13] H. Jianguo and X. Shitong. On the finite volume element method for general self-adjoint elliptic problems. *SIAM Journal on Numerical Analysis*, 35(5):1762–1774, 1998.

[14] Department of Mathematics Louisiana State University Kimn. J. Additive Schwarz and multilevel methods. *www.math.lsu.edu/ kimn/DD.pdf*.

[15] D. Kincaid and W. Cheney. *Numerical analysis: mathematics of scientific computing*. Amer Mathematical Society, 2009.

[16] Y. Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Mathematics of computation*, 37(155):105–126, 1981.

[17] Y. Saad and M.H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.

[18] B.F. Smith, P. Bjorstad, and W. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge Univ Pr, 2004.

[19] JM Thomas and D. Trujillo. Finite Volume Variational Formulation. Application to Domain Decomposition Methods. In *Domain decomposition methods in science and engineering: the sixth International Conference on Domain Decomposition, June 15-19, 1992, Como, Italy*, volume 157, page 127. Amer Mathematical Society, 1994.

[20] L.N. Trefethen and D. Bau. *Numerical linear algebra*. Society for Industrial Mathematics, 1997.

[21] V.R. Voller. *Basic control volume finite element methods for fluids and solids*. World Scientific Pub Co Inc, 2009.

[22] A.M. Winslow. Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh* 1. *Journal of computational physics*, 1(2):149–172, 1966.