UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

Multilevel Elliptic Solvers on Unstructured Grids

Tony F. Chan
Susie Go
Ludmil Zikatanov

August 1997

CAM Report 97-36

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

Multilevel Elliptic Solvers on Unstructured Grids*

Tony F. Chan† Susie Go† Ludmil Zikatanov†

August 18, 1997

Abstract

An overview of multilevel methods on unstructured grids for elliptic problems will be given. The advantages which make such grids suitable for practical implementations are flexible approximation of the boundaries of complicated physical domains and the ability to adapt the mesh to resolve fine-scaled structures in the solution. Multilevel methods, which include multigrid methods and overlapping and non-overlapping domain decomposition methods, depend on proper splittings of appropriate finite element spaces: either by dividing the original problem into subproblems defined on smaller subdomains, or by generating a hierarchy of coarse spaces. The standard splittings used in structured grid case cannot be directly extended for unstructured grids because they require a hierarchical grid structure, which is not readily available in unstructured grids.

We will discuss some of the issues which arise when applying multilevel methods on unstructured grids, such as how the coarse spaces and transfer operators are defined, and how different types of boundary conditions are treated. An obvious way to generate a coarse mesh is to re-grid the physical domain several times. We will propose and discuss different and possibly better alternatives: node nested coarse spaces and agglomerated coarse spaces.

*This paper is an abridged version of the lecture notes which were prepared for the lecture course "28th Computational Fluid Dynamics", 3-7 March, 1997, at the von Karman Institute for Fluid Dynamics, Belgium [1].

†Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095-1555, USA. E-mail: chan@math.ucla.edu, sgo@math.ucla.edu, lzikatan@math.ucla.edu. ONR under contract ONR-N00014-92-J-1890, and the Army Research Office under contract DAAL-03-91-C-0047 (Univ. of Tenn subcontract ORA4466.04 Amendment 1 and 2). The first two authors acknowledge support from RIACS under contract number NAS 2-13721 for visits to RIACS/NASA Ames. The third author is supported by Grant ONR-N00014-92-J-1890 and NSF Grant Int-95-06184.

1 Introduction

In this article, multilevel methods applied to elliptic problems on general unstructured grids will be discussed. We will describe various approaches for dealing with the solution of discrete equations arising from unstructured grids. Our interest will be in the performance of multilevel methods, including multigrid and domain decomposition methods.

The beauty of multilevel methods is that the convergence speed can often be proven to be independent of the problem size and they can be naturally parallelized. This makes them the most powerful and useful tool for a wide variety of applications. On the other hand, these methods require a hierarchical grid structure, which is not readily available in unstructured grids. In our context, we use them not as solvers on their own, but rather as preconditioners for Krylov subspace iterative methods.

Various approaches for dealing with these issues and their effect on the convergence properties of these methods will be covered. This article is organized as follows: Section 1 begins with an introduction to Krylov subspace methods and multilevel methods, followed by some two-level theory in Section 2. Specific examples of how to deal with node-nested multilevel methods are covered in Section 3. Section 4 concerns agglomerated multigrid methods.

Many of the topics described here represent previous and continuing joint work with Barry Smith and Jun Zou [2, 3, 4, 5], (Section 3) and with Jinchao Xu [6] (Section 4).

1.1 Elliptic problems

Elliptic problems are one of the most extensively investigated problems in applied mathematics. Their relation to many physical models is well known and the theoretical and numerical results obtained in this area are very useful in practice. As a first approximation to more complicated physical and mathematical models

(such as those in computational fluid dynamics), elliptic problems are sometimes the only ones for which rigorous theoretical results are known. The design of numerical methods for such model problems can often be adapted and applied to more complicated situations. Elliptic problems are also important in their own right, for example in computational fluid dynamics in the solution of the pressure equation, implicit time integration schemes, etc.

In this section, we will state the model problems we consider. Our goal is to design effective solvers for the resulting systems of linear equations, and we will not pay much attention to the discretization techniques. Detailed discussions of the finite element element discretizations that we use can be found in [7, 8, 9, 10].

Let $\Omega \subset \mathbb{R}^d$ be a polygonal (polyhedral) domain, $d = 2, 3$. We consider the following variational (or Galerkin) formulation of an elliptic problem: Find $u \in H_0^1(\Omega; \Gamma_D)$ such that:

$$a(u, v) = F(v) \text{ for all } v \in H_0^1(\Omega; \Gamma_D), \tag{1.1}$$

where

$$\begin{aligned} a(u, v) &= \int_\Omega \alpha(x) \nabla u \nabla v dx, \\ F(v) &= \int_\Omega F(x) v dx. \end{aligned} \tag{1.2}$$

Here $H_0^1(\Omega; \Gamma_D)$ denotes the Sobolev space which contains functions which vanish on Γ_D with square integrable first derivatives. It is well known that (1.1) is uniquely solvable if $\alpha(x)$ is a strictly positive scalar function and F is square integrable.

We will use the simplest finite element discretization of the elliptic problem (1.1). First, we cover Ω with simplicial finite elements (triangles in $\mathbb{R}^2$ and tetrahedra in $\mathbb{R}^3$). Then the discrete problem can be formulated as follows:

Find $u_h \in V_h$ such that:

$$a(u_h, v_h) = F(v_h) \text{ for all } v_h \in V_h, \tag{1.3}$$

where V_h is the finite dimensional subspace of $H_0^1(\Omega; \Gamma_D)$ consisting of continuous functions linear on each of the simplexes forming the partition.

The values of the discrete solution on the grid nodes are then determined by solving the resulting system of linear equations:

$$Au = f, \tag{1.4}$$

where A is a symmetric and positive definite matrix, f is the right hand side and the nodal values of the discrete solution u_h will be obtained in u after solving the system (1.4). To obtain an accurate enough approximate solution of (1.1), one often has to solve huge discrete problems which are badly conditioned, with condition number growing like $O(h^{-2})$, where h is the characteristic mesh size. Our goal in the next sections will be to construct robust and effective methods for solving the discrete equations (1.4).

1.2 Unstructured grids

With the vast improvements in computational resources today, the motivating reasons for using structured grids over unstructured grids become less obvious. Cartesian or mapped Cartesian grids are popular because they are directional, so efficient methods can be used, such as the alternating direction implicit methods (ADI) and fast Fourier transforms (FFT). This structure, however, imposes limitations on the types of domains which can be considered. In addition, local refinement cannot be easily done without affecting large portions of the grid, so the ability to adapt the grids for resolving steep gradients in the solution is a source of difficulty.

One of the alternative approaches for dealing with complicated geometries is the composite grid method as proposed by Brown, Chesshire, Henshaw and Kreiss [11] and Chesshire and Henshaw [12].

Unstructured grids provide the flexibility needed to adapt to rapidly changing or dynamic solutions as well as complex geometries. These grids have irregular connectivity and so do not have to adhere to the strict structure of Cartesian-based grids, see Figure 1.1. The tradeoff is that, computations on unstructured grids require more complicated data structures and possibly modifications in some solvers, e.g. multilevel methods. We will discuss some such modifications in multilevel methods in this article.

1.3 Preconditioned iterative methods

As mentioned in the introduction, multilevel methods will be used in our framework as preconditioners in

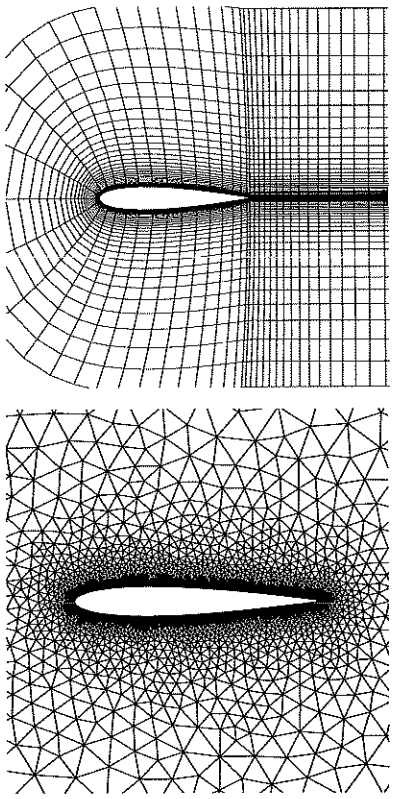

Figure 1.1: A structured grid (top) and unstructured grid (bottom).

Krylov subspace iterative methods, the most popular of which are: the Conjugate Gradient (CG) and the Generalized Minimum Residual (GMRES) method. In addition, there are many other Krylov subspace methods which can be used as alternatives, especially for non-symmetric A's. Some popular methods are BiCGSTAB, CGS, QMR, TFQMR, BCG and their many variants. At this point there is no widespread agreement on the relative merits of these methods. For more details concerning these methods we refer to the recent book by Saad [13].

It is often beneficial, before applying any iterative method, to write (1.4) in the following *preconditioned* form:

$$M^{-1}Au = M^{-1}f. \tag{1.5}$$

where M is called the preconditioner for A. The choice of M is very important because it can improve the convergence rate of the iterative method. A good preconditioner M for A should have the following properties:

- The action of $M^{-1}v$ for a given vector v should be less expensive to compute than $A^{-1}v$.
- The condition number $\kappa(M^{-1}A)$ should be as close to 1 as possible, preferably uniformly bounded above (with respect to the mesh size h).
- If A is SPD then M should be SPD.

When A and M are both SPD, it is more convenient to work with the symmetrized version of (1.5):

$$\begin{aligned} &\hat{A}\hat{u} = \hat{f}, \quad \text{where} \quad \hat{A} = M^{-\frac{1}{2}}AM^{-\frac{1}{2}}, \\ &\hat{u} = M^{\frac{1}{2}}u \quad \text{and} \quad \hat{f} = M^{-\frac{1}{2}}f. \end{aligned} \tag{1.6}$$

Oftentimes, it is useful to apply different preconditioners M_i at each step (e.g. inner inexact solvers). In this way, one obtains flexible solvers which can handle a wider class of problems. The standard Krylov subspace methods must be modified to handle such non-stationary preconditioners. One such method is known as *flexible* GMRES (see Saad [13]).

In the next sections, our particular interest will be focused on multilevel methods (such as domain decomposition methods and multigrid methods) used as preconditioners in PCG. The popularity of these methods as preconditioners is based on the fact that they exactly fit in the applications where finite element or finite difference method is used. In other words, the design of such preconditioners uses the properties of finite element spaces which allow precise optimal constructions and theoretical analysis to be done.

1.4 Multilevel methods

For many practical problems, the system of linear equations which arises from finite element or finite difference discretizations might be huge. A challenge is how to effectively solve such large systems of linear equations. Direct methods face the problem of excessive memory requirements and number of the floating point operations needed. In this connection, iterative methods, and especially multilevel methods such as multigrid and domain decomposition methods, are very attractive. These methods are popular because the amount of work required to solve a problem is on the order of the number of unknowns, the convergence rates are independent of the problem size and they can be easily parallelized.

1.4.1 Multigrid methods.

In this section, we briefly describe the multigrid methods for solving linear systems of discrete equations.

We will consider the case where these systems are obtained via finite element discretization of an elliptic partial differential equation. Detailed discussion on multigrid methods can be found in standard references, e.g. Briggs [14], Bramble [15], Hackbusch [16], and Xu [17, 18].

The idea behind multigrid methods is based on the fact that simple relaxation schemes such as Gauß-Seidel, Jacobi and Richardson possess a good smoothing property: they reduce the highly oscillatory part of the error very well in few iterations. This part of the error lies in the subspace spanned by the eigenvectors corresponding to large eigenvalues, i.e. the high frequencies. The global error, or the low frequencies unfortunately cannot be corrected well by such iterative schemes and this is where multigrid helps. The low frequencies from fine grid (say original one) are transfered to the coarse grid, where they behave like high frequencies, and are smoothed quickly by a simple relaxation scheme. Recursive application of this idea leads to the multigrid method.

We will denote the space which contains the solution u by V_J. We assume that the coarse grids are given and with each grid we associate a finite dimensional space (like V_J for the fine grid). We denote these spaces by $V_0, \ldots, V_{J-1}$. To unify the notation in this section we define $A_J := A$. We assume that the operators A_k, $k = 0, \ldots J-1$, are given (these operators correspond to different approximations of A on the coarse grids). We also assume that the prolongation operator R_k^T and the smoothing operators S_k are also given. One can consider the action of the smoother on $g \in V_k$ as a fixed number of Gauß-Seidel or Jacobi iterations with right-hand side g and zero initial guess.

We view the multigrid method as a way of defining a preconditioner M_J. We will describe in matrix notation the action $M_J^{-1}g$ in the simplest case when one pre- and post-smoothing steps are applied.

The action of M_k^{-1} is then obtained through the following steps:

ALGORITHM 1.1 *(V-cycle multigrid preconditioner $M_k^{-1}g$)*

0. *If $k = 0$ then $M_0^{-1}g = A_0^{-1}g$*
1. Pre-smoothing*: Apply one transposed smoothing iteration with initial guess $x^0 = 0$ and right hand side g, i.e.*

$$x^1 = S_k^T g$$

2. Coarse grid correction*:*

(a) Restrict the residual:

$$q^0 = R_k(I - A_k S_k^T)g.$$

(b) "Solve" on the coarse grid:

$$\begin{aligned} q^1 &= M_{k-1}^{-1} q^0 \\ &= M_{k-1}^{-1} R_k (I - A_k S_k^T) g. \end{aligned}$$

(c) Interpolate back and correct:

$$\begin{aligned} x^2 &= x^1 + R_k^T q^1 \\ &= \left[S_k^T + R_k^T M_{k-1}^{-1} R_k (I - A_k S_k^T)\right] g. \end{aligned}$$

3. Post-smoothing*: Apply one smoothing iteration with initial guess x^2 and right hand side g, i.e.*

$$\begin{aligned} M_k^{-1} g &= x^2 + S_k(g - A_k x^2) \\ &= \left[S_k + S_k^T - S_k A_k S_k^T \right. \\ &\quad \left. + (I - S_k A_k) R_k^T M_{k-1}^{-1} R_k (I - A_k S_k^T)\right] g. \end{aligned}$$

Note that the above definition is recursive, the action of $M_k^{-1}g$ is defined in terms of $M_{k-1}^{-1}g$. Let us now consider the simplest case: a two-level method (when $J = 1$). For the sake of simplicity we omit the index 1 in the next equation. The preconditioner then is defined as follows:

$$\begin{aligned} M^{-1} g &= \left[S + S^T - SAS^T \right. \\ &\quad \left. + (I - SA) R^T A_0^{-1} R (I - AS^T)\right] g. \end{aligned}$$

1.4.2 Domain decomposition methods

Domain decomposition (DD) methods are divide-and-conquer methods which take a large problem defined on a physical domain, and appropriately decompose it into many smaller problems defined on subdomains. These smaller subdomain problems can then be solved quickly and independently of each other and their solution suitably combined, usually via an iterative process to obtain the solution to the original problem. Domain decomposition methods fall into two broad categories: overlapping DD (Schwarz methods) and nonoverlapping DD (substructuring or Schur complement methods). Our description here follows that in Chan-Mathew [19]; see also the recently published book by Smith, Bjørstad and Gropp [20]. We will not discuss the nonoverlapping domain decomposition methods here. For a detailed description and investigation of these methods we refer to [19, 20].

1.4.2.1 Overlapping DD In overlapping DD methods, a set of p overlapping subdomains are formed by taking a set of nonoverlapping subdomains $\{\Omega_i'\}_{i=1}^p$, and extending them to larger subdomains, $\{\Omega_i\}_{i=1}^p$ by some small distance, δ, see Fig. 1.2. The partitioning induced by such a decomposition amounts to an overlapping block decomposition of the system (1.4). Thus, the overlapping DD methods can be thought of as block iterative solvers, either overlapping block Jacobi or block Gauß-Seidel, depending on whether or not the the most updated iterates are used for boundary conditions.

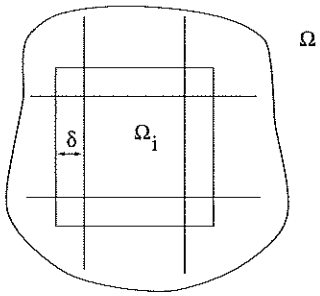

Figure 1.2: Generating a set of overlapping subdomains.

The main ingredients required in all DD methods are:

- *Restriction matrices*: Let R_i be the $n_i \times n$ restriction matrix of 1's and 0's which takes a full-length vector in $\mathbb{R}^n$ and maps it to a restricted vector in $\mathbb{R}^{n_i}$, where n_i denotes the number of unknowns in subdomain Ω_i. The effect on an n-vector is injection onto the subdomain, Ω_i.

- *Extension matrices*: Let R_i^T be the $n \times n_i$ extension matrix, which is defined as the transpose of the restriction matrix, R_i. The effect on an n_i-vector is identity on the subdomain, Ω_i, and zero extension outside the subdomain, i.e. on $\Omega \setminus \Omega_i$.

- *Subdomain matrices*: Define the local stiffness matrix on Ω_i to be $A_i = R_i A R_i^T$, where $A_i \in \mathbb{R}^{n_i \times n_i}$. Because the restriction and interpolation matrices consist only of 0's and 1's, the local stiffness matrices are simply principal submatrices of A.

- *Subdomain solvers*: Let A_i^{-1} symbolically denote the solver for the restricted operator. These can be either exact or inexact solvers (see Sec. 1.4.2).

The additive Schwarz (block Jacobi) method on p subdomains is given by:

$$u^{k+i/p} = u^{k+(i-1)/p} + R_i^T A_i^{-1} R_i (f - A u^k), \quad i = 1, ..., p.$$

In this form, it is seen that corrections are done simultaneously on p subdomains. Rewriting this as one equation reveals the preconditioned iterative method:

$$u^{k+1} = u^k + M_{as}^{-1}(f - Au^k)$$

where the preconditioner M_{as} is given by:

Additive Schwarz preconditioner.
(*block Jacobi on A*)

$$M_{as}^{-1} = \sum_{i=1}^{p} R_i^T A_i^{-1} R_i. \qquad (1.7)$$

Instead of simultaneous corrections, the corrections can also be done successively, to yield the multiplicative Schwarz (block Gauß-Seidel) method, for $i = 1, ..., p$:

$$u^{k+i/p} = u^{k+(i-1)/p} + R_i^T A_i^{-1} R_i (f - A u^{k+(i-1)/p}).$$

Because the most currently updated information is used, this method will generally converge faster than additive Schwarz. The drawback is that it is less parallel (but this can be remedied by appropriate coloring of the subdomains).

For the multiplicative Schwarz method on p subdomains, the preconditioner can be written as:

Multiplicative Schwarz preconditioner.
(*block Gauß-Seidel on A*)

$$M_{ms}^{-1} = [I - \prod_{i=1}^{p} (I - R_i^T A_i^{-1} R_i A)] A^{-1}. \qquad (1.8)$$

1.4.2.2 Coarse grid The domain of dependence for elliptic problems is the entire domain, but because Schwarz methods decompose the problem into smaller, independent problems, information from one subdomain must travel large distances to reach another subdomain. To avoid deterioration of the convergence rates of these methods, some sort of mechanism for the global transfer of data is needed. This is achieved, to some degree, by the overlapping of subdomains in the Schwarz methods. More overlap leads to more coupling between subdomains. However, this adds redundant work and communications overhead if too much overlap is introduced. Dryja and Widlund [21, 22] showed

that the condition number for additive Schwarz (1.7) is given by:

$$\kappa(M_{as}^{-1}A) = O\left(H^{-2}\left(1+\left(\frac{H}{\delta}\right)^2\right)\right).$$

The condition number is independent of h. For sufficient amount of overlap (choosing $\delta = O(H)$), the condition number is $O(H^{-2})$ and so will increase as H tends to zero. This means that the method will not be scalable to a large number of processors.

This deterioration can be remedied by introducing a coarse grid to achieve additional global coupling. In addition to the subdomain restriction, interpolation and stiffness matrices used in the one-level Schwarz methods, we need coarse versions of them: $R_H, R_H^T, A_H = R_H A R_H^T$, and A_H^{-1}. Here, R_H and R_H^T will instead be the full weighting restriction and linear interpolation matrices, respectively, which are commonly used in multigrid methods. The two-level additive Schwarz preconditioner can then be written as:

Additive Schwarz preconditioner with coarse grid.

$$M_{asc}^{-1} = R_H^T A_H^{-1} R_H + \sum_{i=1}^{p} R_i^T A_i^{-1} R_i.$$

It can be shown that the condition number for this two-level method is (see Section 2):

$$\kappa(M_{asc}^{-1}A) = O(1+(H/\delta)^2),$$

and the method can be made independent of H, h with sufficient overlap by choosing $\delta = O(H)$.

1.4.2.3 Multilevel Schwarz Multilevel Schwarz is an extension of two-level Schwarz with L different coarse levels, each level being decomposed into p_l subdomains as previously described. We will denote the i^{th} subdomain on the l^{th} level as: Ω_i^l. Several different variants of multilevel Schwarz can be created, depending on when the most currently updated information is used:

- Fully additive multilevel methods would be additive among subdomains on the same level as well as additive between levels.
- Multilevel methods which are multiplicative among subdomains on the same level, but additive between levels can be viewed as "additive MG".
- Classical V-cycle MG can be viewed as a multilevel Schwarz method which is multiplicative both among subdomains on the same level as well as between levels.

The fully additive multilevel Schwarz preconditioner can be written as:

Fully additive multilevel Schwarz preconditioner.

$$M_{mlas}^{-1} = \sum_{l=1}^{L}\sum_{i=1}^{p_l}(R_i^l)^T(A_i^l)^{-1}(R_i^l).$$

1.4.2.4 Inexact subdomain solves In all of the domain decomposition methods described above, subdomain solves, A_i^{-1}, are required. These can be done either exactly or inexactly. Though the subdomain and coarse problems are much smaller than the original problem, it can still be quite expensive to attempt exact solves on these problems.

In the two-level additive Schwarz methods, we can simply replace the exact solves with inexact solves. Let $M_i \approx A_i$, $M_H \approx A_H$ represent the inexact solves. Then the preconditioner is given by:

$$\tilde{M}_{asc}^{-1} = R_H^T M_H^{-1} R_H + \sum_{i=1}^{p} R_i^T M_i^{-1} R_i.$$

1.5 Approaches for designing multilevel methods on unstructured grids

Multilevel methods require a hierarchical grid structure. For structured grids, the hierarchy can be recovered from the fine grid. For unstructured grids, however, there is no natural grid hierarchy. In addition, their lack of structure prevents these methods from exploiting regularity and using fast solvers as with structured grids. Difficulties exist in identifying coarse grid problems/spaces/boundary conditions which do not occur when using structured grids. The algorithms which are based on unstructured grids must be redesigned to handle these issues without sacrificing too much in terms of complexity and performance.

There are several approaches for constructing the coarse spaces for unstructured grids. One approach is

to simply apply algebraic versions of multigrid methods [23]. However ignoring the geometric and differential nature of the underlying problem may lead to suboptimal performance.

Another approach (see Mavriplis [24]) is based on independently generated coarse grids and piecewise linear interpolation between the grids. The advantage of this approach is convenience: the coarse grids can be generated by using the same grid generator which produced the original fine grid. The disadvantage is that the interpolations can be expensive to apply since the set of nodes in the coarse grids are not related in any way to the nodes in the fine grid. Thus no fast search routines can be applied and the implementation will be $O(n^2)$.

An alternative approach is based on generating node-nested coarse grids, which are created by selecting subsets of a vertex set, retriangulating the subset, and using piecewise linear interpolation between the grids (see [25, 26]). This still provides an automatic way of generating coarse grids and now faster implementations of the interpolation (can be implemented in $O(n)$ time). The drawback is that in three dimensions, retetrahedralization can be problematic.

Another effective coarsening strategy proposed by Bank and Xu [27] uses the geometrical coordinates of the fine grid (which is available in most cases).

New coarsening strategies based on the algebraic approach recently were published by Hackbusch [28], Braess [29] and Reusken [30].

In many of these approaches, problems may occur in producing coarse grids which are valid and with boundaries which preserve the important features of the fine domain. One of most popular and promising new coarsening techniques which avoids this problem is based on the agglomeration technique (see Koobus, Lallemand and Dervieux [31]). Instead of constructing a proper coarse grid, neighboring fine grid elements are agglomerated together to form macroelements. Since these agglomerated regions are not standard finite elements, appropriate basis functions and interpolation operators must be constructed on them. Such algorithms have also been investigated by Mandel, Vaněk, Brezina [32]) and Vaněk, Křížková [33].

2 Introduction to convergence theory

As mentioned in Section 1.3, the estimate of the convergence rate of the PCG requires an estimate of the upper bound of $\kappa(M^{-1}A)$. In particular, estimates on the extreme eigenvalues of $M^{-1}A$ must be obtained. In this section, we first give a general framework for bounding $\kappa(M^{-1}A)$ and then we show how such an analysis can be carried out for the overlapping domain decomposition method. Such an analysis can show (or predict) the convergence rate and in most cases gives a good guess as to how the parameters and approximate operators should be chosen in order to get an optimal iterative method. For a similar approach in analyzing the convergence properties of iterative methods using general subspace splittings for structured meshes, we refer to [17].

We shall adopt a matrix approach for analyzing the domain decomposition methods, in the hope that it is more intuitive and easier to understand. In the analysis, we state the important theoretical results omitting their proofs. Detailed presentations of the analysis, rigorous proofs of the results quoted here, and more references can be found in [19], [20].

Although we will present the domain decomposition methods in matrix formulation, the use of Sobolev norms and semi-norms in $H^k(\Omega)$ cannot be avoided in a few places, so we will denote them by the conventional notation: $\|u\|_k$ (or $\|u\|_{k,\Omega}$) and $|u|_k$ (or $|u|_{k,\Omega}$), respectively (see [20]).

2.1 Subspace correction framework: matrix formulation

Our initial setting in matrix form is as follows: Let Ω be covered by p overlapping subdomains Ω_i, $i = 0, 1, \ldots, p$. Each subdomain Ω_i corresponds to a subspace $V_i \subset \mathbb{R}^n$. The subspaces are defined through the restriction operators $R_i \in \mathbb{R}^{n_i \times n}$, $i = 0, 1, ..., p$, and we set $V_i = \text{Range}(R_i)$ Note here that we will interchangeably use the notation for the coarse grid versions denoted with subscript H in the previous section, with the subscript 0, when convenient. Here, V_0 denotes the coarse space.

We wish to construct a preconditioner for solving the following linear algebra problem

$$Au = f, \quad A \in \mathbb{R}^{n \times n} \text{ is SPD.} \tag{2.1}$$

Let us first explain the intuition behind the construction of a preconditioner based on this splitting of $\mathbb{R}^n$. It is natural to take the best approximation to the solution from each subspace, and then extend these different approximations to the whole $\mathbb{R}^n$ somehow in order to get a global solution. Thus the question is: What is the best correction to the k-th iterate u^k from V_i?

If we measured the error in the A-norm, $\|\cdot\|_A$, then this question can be reformulated as the following minimization problem:

$$\min_{y_i} \|(u^k + R_i^T y_i) - A^{-1} f\|_A. \tag{2.2}$$

The solution is given by:

$$\begin{aligned} y_i &= (R_i A R_i^T)^{-1} R_i (f - A u^k) \\ &= A_i^{-1} R_i (f - A u^k). \end{aligned} \tag{2.3}$$

The next iterate is then obtained via the equation (note that we correct here only in one subspace V_i)

$$u^{k+1} = u^k + R_i^T A_i^{-1} R_i (f - A u^k) \tag{2.4}$$

Example. The Jacobi iteration (see Section 1.3) corresponds to the splitting $V_i = span\{e_i\}$ where e_i is the i-th unit coordinate vector. The restrictions, R_i, in this case are defined as $R_i v = (v, e_i) e_i$.

Performing these subspace corrections simultaneously gives the additive subspace correction preconditioner:

$$M_{asc}^{-1} = \sum_{i=0}^{p} R_i^T A_i^{-1} R_i.$$

Defining now the projections $P_i \equiv R_i^T A_i^{-1} R_i A$, for $i = 0, \ldots p$, we get

$$M_{asc}^{-1} A = \sum_{i=0}^{p} P_i.$$

As we pointed out earlier, the convergence of the PCG method depends on the condition number of $M^{-1}A$. Thus our goal is to find an upper bound for $\kappa(M^{-1}A)$ which amounts to finding an upper bound for $\lambda_{\max}(M_{asc}^{-1}A)$ and a lower bound for $\lambda_{\min}(M_{asc}^{-1}A)$.

The estimate on the upper bound for $\lambda_{\max}(M_{asc}^{-1}A)$ is easier and it follows directly from the following simple lemmas.

Lemma 2.1 *P_i is a projection in $(\cdot,\cdot)_A$, i.e.*

$$AP_i = P_i^T A, \quad P_i^2 = P_i \quad \|P_i\|_A \leq 1.$$

Lemma 2.2 *The maximal eigenvalue of the preconditioned matrix satisfies the following inequality:*

$$\lambda_{\max}(M_{asc}^{-1}A) \leq p + 1.$$

REMARK. The bound given in the previous lemma can be easily improved to:

$$\lambda_{\max}(M_{asc}^{-1}A) \leq n_c + 1 \equiv c_1,$$

where n_c is the *number of colors to color Ω_i's* in such a way that no two neighboring subdomains are colored the same color.

We next give an estimate on the lower bound for $\lambda_{\min}(M_{asc}^{-1}A)$. This estimate is based on the following *Partition Lemma* which plays a crucial role in the convergence analysis of the domain decomposition methods.

Lemma 2.3 (Partition Lemma). *(Matsokin-Nepomnyaschikh [34], Lions [35], and Dryja-Widlund [36, 21]). Assume that there exists a constant c_2 such that*

$$\min_{\substack{u = \sum u_i \\ u_i \in V_i}} \sum_{i=0}^{p} \|u_i\|_A^2 \leq c_2 \|u\|_A^2. \tag{2.5}$$

then

$$\lambda_{\min}(M_{asc}^{-1}A) \geq \frac{1}{c_2}.$$

The assumption made in the partition lemma (equation (2.5)) means that for any given u, a stable decomposition must exist in the sense that the sum of the "energy" of all the pieces u_i lying in V_i is bounded by the global energy norm of the decomposed vector. This assumption can be viewed as a condition on the V_i's, i.e. the subspaces must not introduce oscillations (high energy components) in u_i.

Combining lemmas 2.1– 2.3, we get our main theorem:

Theorem 2.1 *If assumption (2.5) holds, then the condition number $\kappa(M_{asc}^{-1}A)$ can be bounded by:*

$$\kappa(M_{asc}^{-1}A) \leq c_1 c_2. \tag{2.6}$$

This result suggests how to construct the decompositions in order to obtain optimal preconditioners: we want the constants c_1, c_2 to be independent of the problem parameters such as the number of subdomains, the characteristic mesh sizes h and H, jumps in the coefficients of the underlying PDE, etc. It is also desirable to make c_1 and c_2 as small as possible in order to get a condition number close to 1. But c_1 and c_2 depend on the size of overlaps in the subspaces V_i. More overlap will decrease c_2, but the number of colors c_1 will increase. On the other hand, small overlap will lead to large c_2 and small c_1. Thus the space decompositions have to be made in such a way to ensure that the product $c_1 c_2$ is as small as possible.

2.2 Application to two-level overlapping domain decomposition methods

As an example of the application of the above theory, we will present a detailed estimate for the condition number of the two-level Schwarz method.

2.2.1 The intuitive idea

As in the previous section, we first present the basic intuitive idea using a simple 1D version of (1.1).

We want a splitting which satisfies the partition assumption (2.5). Take Ω to be a fixed open interval on the real line and cover Ω with p overlapping subdomains Ω_i, $i = 1, \ldots, p$ (see fig 2.1). Consider the partition of unity θ_i corresponding to this covering. By construction the functions θ_i satisfy

$$\sum_{i=1}^{p} \theta_i = 1, \quad 0 \le \theta_i \le 1, \quad |\theta_i|_{1,\infty} \le \delta^{-1}, \qquad (2.7)$$

where δ is the size of the overlap and $|\theta|_{s,\infty}$ denotes the maximum norm, i.e. the maximum of the s-th derivative of θ_i. We define $u_i = \theta_i u$, so we have $u = \Sigma_{i=0}^{p} u_i$.

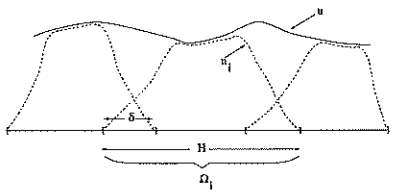

Figure 2.1: The partitioned u_i's without coarse grid.

Since A corresponds to a discretization of the second order elliptic operator, $\frac{d}{dx}\alpha(x)\frac{d}{dx}$, it is easy to see that the A-norm and the H^1-seminorm are equivalent in this case: $\|u\|_A \approx \|du/dx\|_0$. Our goal is to bound $\|u_i\|_A$ by $\|u\|_A$. Looking at Fig. 2.1, we see that the function u_i changes from $\|u\|_0$ to 0 over a distance δ and we get:

$$\|u_i\|_A^2 = \left\| \left(\frac{du}{dx} \right) \right\|_0^2 \le c \left(\frac{\|u\|_0}{\delta} \right)^2 .$$

We still need to bound $\|u\|_0$ by $\|u\|_A$. But the function u, satisfying homogeneous Dirichlet boundary conditions, cannot change rapidly over the interval Ω if there is no significant change in the derivative. The well-known Poincaré inequality estimates the norm of the function with the norm of derivatives and its application leads to the following:

$$\left(\frac{\|u\|_0}{\delta} \right)^2 \le \tilde{c} \left(\frac{\|u\|_A}{\delta} \right)^2 .$$

After summing over all subdomains, we get:

$$\sum_{i=1}^{p} \|u_i\|_A^2 \le O\left(\frac{1}{\delta^2} \right) \|u\|_A^2 .$$

Therefore,

$$c_2 \;=\; O\left(\frac{1}{\delta^2} \right) = O\left(\left(\frac{H}{\delta} \right)^2 \frac{1}{H^2} \right) .$$

From these inequalities, one may conclude that if the overlap is of size $O(H)$, then $\kappa(M^{-1}A) = O(H^{-2})$, which is an improvement over $O(h^{-2})$, but is still unsatisfactory. We can see that the overlapping subdomains alone cannot provide a stable partition of u.

It turns out that this dependence on H can be eliminated by using a global coarse space, V_H, which couples all the subdomains. The idea is to construct a coarse grid approximation u_H to u satisfying the following two important properties:

$$\|u_H\|_A \;\le\; c\|u\|_A \qquad (2.8)$$

$$\|u - u_H\|_0 \;\le\; cH\|u\|_A \qquad (2.9)$$

Define $w = u - u_H$ and the following partition of u:

$$u_i = \theta_i(u - u_H), \quad u = u_H + \sum_{i=1}^{p} u_i . \qquad (2.10)$$

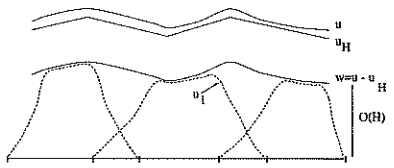

Figure 2.2: The partitioned u_i's with coarse grid.

Proceeding as before and taking into account that now the u_i's change from 0 to $O(H)$, we have

$$\|u_i\|_A^2 \leq c\left(\frac{H\|u\|_A}{\delta}\right)^2 \leq c\left(\frac{H}{\delta}\right)^2 \|u\|_A^2.$$

If we make the natural assumption that $\delta = O(H)$, the bound for c_2 now reads

$$c_2 = O\left(\left(\frac{H}{\delta}\right)^2\right) = O(1).$$

Thus, we can see that the role of the coarse grid V_H is to make $\|u - u_H\|$ small enough ($O(H)$), so that it can be partitioned in a stable manner.

We would like to comment on the choice of u_H. As it can be seen, u_H is a purely theoretical construction and there is no need of its use in the algorithm. One possible choice is $u_H = \mathcal{R}_H u$, where $\mathcal{R}_H$ is a kind of interpolation or projection operator. Of course, $\mathcal{R}_H$ must satisfy properties similar to (2.8) and (2.9) namely:

$$\begin{aligned} |\mathcal{R}_H u|_1 &\leq c|u|_1 \quad \text{(stability)} \qquad (2.11)\\ \|u - \mathcal{R}_H u\|_0 &\leq cH|u|_1. \quad \text{(approx.)} \qquad (2.12) \end{aligned}$$

A natural candidate for such an operator is the nodal value interpolant on the coarse grid, $u_H = I_H u$. A drawback of such a choice is that in $3D$ this interpolation does not satisfy the stability property (2.11). To see this, we take w to be the basis function ψ_i^h associated with the grid node x_i. Then we have

$$\begin{aligned} |w|_1^2 &= \int_{\Delta_h} \left(\tfrac{1}{h}\right)^2 = O(h)\\ |I_H w|_1^2 &= \int_{\Delta_H} \left(\tfrac{1}{H}\right)^2 = O(H). \end{aligned}$$

The last estimate shows that the stability requirement is violated.

If the grid is structured, then a good and stable coarse grid approximation to the elements of V_h is the *L_2-projection* Q_H from $V_h \to V_H$ and we can define $\mathcal{R}_H = Q_H$, i.e. $u_H = Q_H u$. It is known that this u_H satisfies the stability and approximation properties (2.11) and (2.12) (see Xu [17] or Dryja and Widlund [36]).

We now state the main result of this section.

Theorem 2.2 *Under the above assumptions (2.11) and (2.12) for the condition number $\kappa(M_{asc}^{-1}A)$, the following estimate holds*

$$\kappa(M_{asc}^{-1}A) = O(1 + (H/\delta)^2). \qquad (2.13)$$

2.2.2 Some extensions

Inexact subdomain solves can easily be accommodated by using $\|\cdot\|_{M_i}$ satisfying:

$$\|u\|_A \leq \omega\|u\|_{M_i} \qquad \forall u \in V_i.$$

Then the constant ω in the above inequality will be absorbed in the resulting bound for $\kappa(M^{-1}A)$.

The extension to *multilevel* Schwarz method is straightforward. An additional assumption, however is needed in this case:

$$(u_i, u_j)_A \leq \varepsilon_{ij}\|u_i\|_A\|u_j\|_A \qquad \forall u_i \in V_i, \forall u_j \in V_j.$$

Such inequalities measure the abstract angles between the subspaces and are known in the literature as "strengthened Cauchy Schwarz inequalities". For a detailed discussion of the issues concerning multilevel theory we refer to Xu [17, 18], Chan-Mathew [19]. Note that $\rho(\{\varepsilon_{ij}\})$ will enter in the bound for $\kappa(M^{-1}A)$.

We now briefly comment on the convergence of the *multiplicative* Schwarz method which was described in Section 1.4.2. From (1.8), for the error e^{k+1}, we have:

$$e^{k+1} = (I - P_p)\cdots(I - P_0)e^k.$$

Since each $(I - P_i)$ is a projection in the A-norm, it immediately follows that $\|e^{k+1}\|_A \leq \|e^k\|_A$. The following result gives a bound for the damping factor of the multiplicative iteration.

Theorem 2.3 *(Bramble, Pasciak, Wang, Xu [37]; and Xu [17]) Let V_i's satisfy the assumptions of the Partition Lemma. Then the following estimate is true:*

$$\|(I - P_p)\cdots(I - P_0)\|_A \leq 1 - \frac{c}{c_2}.$$

where c depends on the number of colors for coloring the Ω_i's but is independent of p.

2.3 Convergence of multigrid methods

The convergence properties of *multigrid* methods (see Section 1.4.1) depend on many parameters. One can vary the number of smoothing steps, the smoothing operators, the interpolation and restriction operators, coarse grid operators, etc. There are two main approaches in constructing multigrid preconditioners. One of them uses nested subspace splittings of V_h and the other one uses non-nested spaces or specially interpolated bilinear forms (coarse grid matrices). The discussion of the convergence in both these cases is given in [15, 37, 17, 18]. Here we give the simplest convergence result in the case of *nested* spaces (i.e. $V_0 \subset V_1 \subset \ldots V_{J-1} \subset V_J = V_h$). and so-called full elliptic regularity assumption:

$$\|u\|_2 \leq c\|F\|_0,$$

where u is the solution, F is the right hand side of (1.1).

Again, the stability and approximation properties (2.11) and (2.12) are crucial in the convergence theory. The stability property (2.11) is automatically satisfied when the spaces are nested. It turns out that from the regularity assumption the following approximation property follows (see Xu [17, 18]):

There exists a constant, c_1, independent of the mesh parameters (i.e. of the mesh size h) such that

$$\|I - P_{k-1}v\|_A^2 \leq c_1 \frac{1}{\rho(A_k)}\|A_k v\|_2^2 \quad \forall v \in V_k, \quad (2.14)$$

where P_k denotes the elliptic projection defined by $(AP_k u, v) = (Au, v) \ \ \forall v \in V_k$.

The other operator involved in the definition of M_J^{-1} is the smoother and we make the following assumption on it:

$$\frac{c_0}{\rho(A_k)}(v, v) \leq (S_{symm,k}v, v) \leq (A_k^{-1}v, v). \quad (2.15)$$

The smoother $S_{symm,k}$ is the symmetric version of S_k and is defined as: $S_{symm,k} := S^T + S - S^T AS$. Inequalities of the type (2.15) are satisfied by the Gauß-Seidel method.

An important thing to mention for the choice of the smoother is that we are trying to choose smoother which will quickly capture the high frequency components of the error, and we are not going to use it as a solver. For example if the matrix A_J corresponds to the five point finite difference stencil, it can be seen that the Jacobi method (with $\omega = 1$) is not good for a smoother. One must use the damped Jacobi method with $\omega < 1$.

The subspace correction framework applies to multigrid methods as well. As long as the stability and approximation properties are verified, the following convergence result holds:

Theorem 2.4 *Under the assumptions (2.14) and (2.15), the following estimate is true:*

$$\|I - M_J^{-1}A\|_A \leq 1 - \frac{c_0}{c_1}. \quad (2.16)$$

We want to point out that a convergence result similar to Theorem 2.4 is also true in the *non-nested* case. We refer to work by Bramble et.al. [37] or Chan-Zou [38, 39] for unstructured grids. The construction of interpolation operators satisfying the stability property might be an issue. Some possible stable definitions can be found in [38] and [39] and in Section 3.2.

In the next sections, we will define subspaces satisfying the above assumptions or directly satisfying the approximation property similar to (2.14). The verification of these assumptions is easy for structured grids and can be found in many papers. On unstructured grids, however, this might be rather complicated and tricky. Special attention should be paid to the construction of the spaces themselves, rather than using standard spaces and verifying the above inequalities.

3 Node-nested coarse spaces

Unstructured multilevel methods for solving linear systems like (1.4) require a hierarchy of coarse grids. Grids which are node-nested have the advantage that they can be automatically generated and that efficient methods can be used to create the interpolation and restriction operators needed to transfer information from one level to the other. Disadvantages are that for complicated geometries, particularly in three dimensions, special care must be taken to ensure that the coarse grids which are produced are valid and preserve the important geometric features of the fine domain. With unstructured meshes, the grid hierarchy can allow general grids which are non-quasiuniform and coarse grids whose boundaries may be non-matching to the boundary of the fine grid, so care must be applied when constructing intergrid transfer operators for various types

of boundary conditions. In this section, we will discuss some possibilities.

3.1 Maximal independent set (MIS) coarsening

A maximal independent set of vertices in a graph is a subset of vertices which is *independent* in the sense that no two vertices in the subset are connected by an edge, and *maximal* if the addition of a vertex results in a dependent subset. An automatic approach to generating node-nested coarse grids is to take a maximal independent set (MIS) of the vertices and call this set, the set of coarse grid nodes, and then retriangulate it [25, 26]. A sequence of coarse grids can thus be created by repeated application of this technique.

A simple technique for finding a MIS of vertices is to first choose a MIS of the boundary vertices by choosing every other boundary vertex and eliminating all its nearest neighbors, and then find a MIS of the interior vertices by selecting a random interior vertex and eliminating all its nearest neighbors, and repeating the process until all vertices are either eliminated or selected. The resulting vertex subset is then retriangulated using for example, the same triangulation routine which generated the original fine grid.

3.2 Coarse-to-fine interpolations

In general, the resulting coarse grid domain will have boundaries which will not match the boundaries of the fine grid so the coarse space V_H is usually not a subspace of the fine space V_h. Indeed, even if $\Omega_H = \Omega_h$, V_H may still not be a subspace of V_h since the coarse elements are generally not the unions of some fine elements in unstructured grids. To construct a coarse-to-fine transfer operator, one can use the standard nodal value interpolant associated with the fine space, V_h.

ALGORITHM 3.1 *(Standard nodal value interpolation)*

1. **For** *each fine grid node,*
2. Search *through all coarse grid elements until one which contains it is found.*
3. **If** *the fine grid node is a coarse grid node, then*
4. Set *the interpolant to be equal to that nodal value,*
5. **Else**
6. Set *it to be a linear interpolation of the 3 nodal values making up that coarse grid element (see Fig. 3.1).*

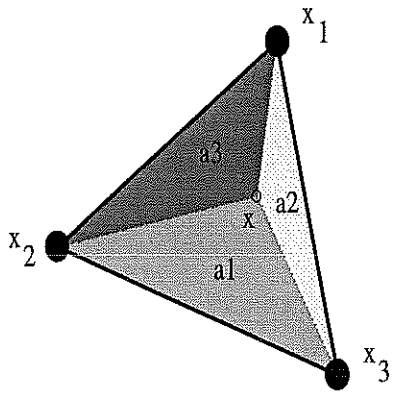

Figure 3.1: Barycentric (natural) coordinates: $\lambda_i(x) = \frac{\text{area}(a_i)}{\text{area}(\Delta_{123})}$, for $i = 1, 2, 3$, where Δ_{123} is the simplex with vertices x_1, x_2, x_3. The value of a function f at a point x is given by: $f(x) = \lambda_1(x)f(x_1) + \lambda_2(x)f(x_2) + \lambda_3(x)f(x_3)$.

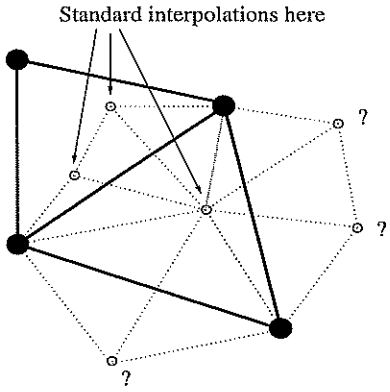

Figure 3.2: Use standard interpolation for fine grid nodes interior to a coarse element. What should be done for fine grid nodes which are not interior to any coarse grid element? One way is simply with extension by zero.

A naive implementation of this routine requires $O(n^2)$ time, but exploiting the node-nested property of the grids, one can implement this in $O(n)$ time, since only nearest coarse grid elements of a fine node need to be searched.

3.3 Interpolations on non-matching boundaries

Notice, however, that the standard nodal value interpolant is only well defined for those fine nodes lying also in the coarse domain $\bar{\Omega}_H$, but undefined for those fine nodes lying outside $\bar{\Omega}_H$. That is, in Step 2 of the standard nodal value interpolation (Algorithm 3.1), there is no provision for what to do if all the coarse grid

elements have been searched, and none contains the fine grid node. A simple and natural way to remove this barrier is to assign those fine node values by zero. We denote this interpolant as the coarse-to-fine interpolant, $\mathcal{I}_h^0$. This zero extension interpolant works well for Dirichlet boundary conditions [26, 2] but will not be accurate nor stable for other boundary conditions.

We provide a simple one dimensional example to illustrate why better interpolants are needed at non-matching boundaries. This example has a Dirichlet boundary condition at the left boundary point and a homogeneous Neumann boundary condition at the right boundary point. The fine grid function, u, and the coarse grid approximation to it, U_H are shown. For Neumann boundary conditions, the elements from V_h which have to be interpolated are generally not zero at the Neumann part of the boundary. Recall from Section 1.1 that V_h is a subspace of $H_0^1(\Omega, \Gamma_D)$, whose elements are restricted to vanish only on Dirichlet boundary. Thus using a zero extension interpolant at a Neumann boundary will not be accurate enough and introduces a correction with high energy ($\|u - u_H\|$ is no longer $O(H)$), (see Fig. 3.3).

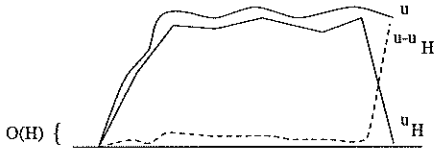

Figure 3.3: Non-matching boundaries: Zero extension interpolation is not accurate at the right end of the coarse domain.

To achieve better efficiency, we need to modify this intergrid operator to account for the Neumann condition. Two general ways to treat such boundaries are:

1. Modify the coarse grid domain to cover any fine grid boundaries of Neumann type and use standard nodal value interpolation.

2. Increase the accuracy of the interpolants by accounting for the Neumann condition for those fine nodes in $\Omega \backslash \Omega^H$.

The first approach is motivated by the fact that standard nodal value interpolants can still be used with efficiency, provided the coarse grid covers the Neumann boundary part of the fine grid (see Fig. 3.4). This was first proposed and justified in [2]. We shall denote this operator as the coarse-to-fine interpolant, $\mathcal{I}_h^1$. Let us still denote the modified coarse grid domain by $\bar{\Omega}_H$. Then for all $v^H \in V^H$, the interpolant $\mathcal{I}_h^1$ is defined as:

$$\mathcal{I}_h^1 v^H(x) = \begin{cases} v^H(x) & \text{for } x \in \Omega \cap \bar{\Omega}_H, \\ 0 & \text{for } x \in \Omega \setminus \bar{\Omega}_H. \end{cases}$$

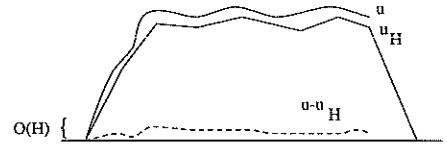

Figure 3.4: More accurate interpolation with $\mathcal{I}_h^1$ done at Neumann boundary.

This is a natural extension of v_H by zero outside the Dirichlet boundary part of the coarse grid domain. Similar zero extensions were used in Kornhuber-Yserentant [40] to embed an arbitrarily complicated domain into a square or cube in constructing multilevel methods on nested and quasi-uniform meshes for second order elliptic problems with purely Dirichlet boundary conditions.

Although the coarse-to-fine operator $\mathcal{I}_h^1$ works well for mixed boundary conditions, one has to modify the original coarse grid so that it covers the Neumann boundary part of the fine grid domain (see [4] for a description for modifying boundaries). This can be very difficult to do for complicated domains. To avoid modifying the original coarse grid, we now consider standard finite element interpolants which are modified only near Neumann boundaries. The idea is as follows: Let us consider a fine grid point, x, which lies outside the coarse grid domain. Find a nearby coarse grid triangle to x (say, τ_H with vertices x_1, x_2, x_3), and *extrapolate* $u(x)$ using the values $u(x_1), u(x_2)$ and $u(x_3)$. Note that such an extrapolation should depend on the type of boundary condition at x.

We define the interpolant at x by using the nodes of the coarse boundary edge closest to x:

$$\mathcal{I}_h^2 v^H(x) = \lambda(x) v^H(x_1^H) + (1 - \lambda(x)) v^H(x_2^H),$$

where the coarse edge has endpoints x_1^H and x_2^H, and λ is the ratio of the lengths of two segments of this edge cut off by the normal line passing through x to the edge. This kind of interpolation was also used by Bank and Xu [41] in their construction of a hierarchical basis on a unstructured mesh.

We can also use a non-zero extension by extrapolation using barycentric functions:

$$\mathcal{I}_h^3 v^H(x) = \lambda_1(x) v^H(x_1^H) + \lambda_2(x) v^H(x_2^H)$$

$$+\lambda_3(x)v^H(x_3^H),$$

where $\lambda_1, \lambda_2, \lambda_3$ are the three barycentric coordinate functions corresponding to τ_H (see Figure 3.1). Note that the functions λ_1, λ_2 and λ_3 used in the definition of $\mathcal{I}_h^3$ satisfies $\lambda_1, \lambda_2, \lambda_3 \geq 0$ for $x \in \tau_H$, but not so for $x \notin \tau_H$. The barycentric coordinates may still be defined, provided we consider the area of a simplex to be orientation-dependent. That is, area is > 0 for "right-handed" triangles (clockwise) and area is < 0 for "left-handed" triangles (counter-clockwise).

We summarize the various interpolants:

$\mathcal{I}_h^0$: Zero extension with unmodified coarse boundaries,

$\mathcal{I}_h^1$: Zero extension with modified coarse Neumann boundaries,

$\mathcal{I}_h^2$: Nearest edge interpolation, and

$\mathcal{I}_h^3$: Nearest element interpolation.

3.4 Stability and approximation of the non-nested interpolation

The convergence theory for overlapping multilevel domain decomposition and multigrid methods require the coarse-to-fine grid transfer operator to possess the local optimal L^2-approximation and local H^1-stability properties as introduced in Sec. 2. The locality of these properties is essential to the effectiveness of these methods on highly non-quasi-uniform unstructured meshes.

Because the spaces are non-nested (they are *node-nested*, but still *non-nested*, as coarse grid elements are not unions of fine grid elements), in the theory discussed in Section 2, the u_H coarse space approximation to u should be defined as:

$$u_H = \mathcal{I}_h \mathcal{R}_H u.$$

Since $\mathcal{R}_H \in V_H \not\subset V_h$, we need to use the interpolation operator $\mathcal{I}_h$ to map $\mathcal{R}_H u$ back to V_h. The convergence theory now requires both $\mathcal{I}_h$ and $\mathcal{R}_H$ to possess the stability and approximation properties. When the mesh is quasi-uniform, the usual L^2-projection, Q_H, can be used for $\mathcal{R}_H$. But when the mesh is highly non-quasi-uniform, the constant in the approximation property (2.12) can deteriorate if we use Q_H. The trick then is to use a localized version of the L^2-projection, i.e. the so-called Clément's projection. It is known that this projection provides local stable and good approximations. We refer to Clément [42] for its definition and Chan-Zou [38] and Chan-Smith-Zou [2] for its use in domain decomposition contexts.

For $\mathcal{I}_h$, we can take the coarse-to-fine interpolants introduced in Sec. 3.3 The key step then is proving the stability and approximation properties for $\mathcal{I}_h$. The proof that the non-nested standard interpolation used in the interior is stable and accurate can be found in Cai [43] and Chan-Smith-Zou [2]. The proof for the boundary-specific interpolations can be found in [5].

3.5 Numerical results

In this section, we provide some numerical results of domain decomposition and multigrid methods for elliptic problems on an unstructured airfoil mesh: see Figure 3.5. The well-known NASA airfoil mesh was provided by T. Barth and D. Jesperson of NASA Ames. Coarse grids were generated by the MIS approach as described above. All numerical experiments were performed using the Portable, Extensible Toolkit for Scientific Computation (PETSc) [44], running on a Sun SPARC 20. Piecewise linear finite elements were used for the discretizations and the resulting linear system was solved using either multilevel overlapping Schwarz or V-cycle multigrid as a preconditioner with full GMRES as an outer accelerator. In the experiments, the initial iterate is set to be zero and the iteration is stopped when the discrete norm of the residual is reduced by a factor of 10^{-5}.

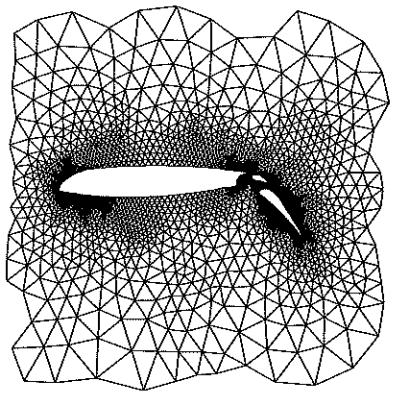

Figure 3.5: Unstructured NASA airfoil with 4253 nodes.

For partitioning, all the domains (except the coarsest) were partitioned using the recursive spectral bisection method [45], with exact solves for both the subdomain problems and the coarse grid problem. To generate overlapping subdomains, we first partition the domain into nonoverlapping subdomains and then extend

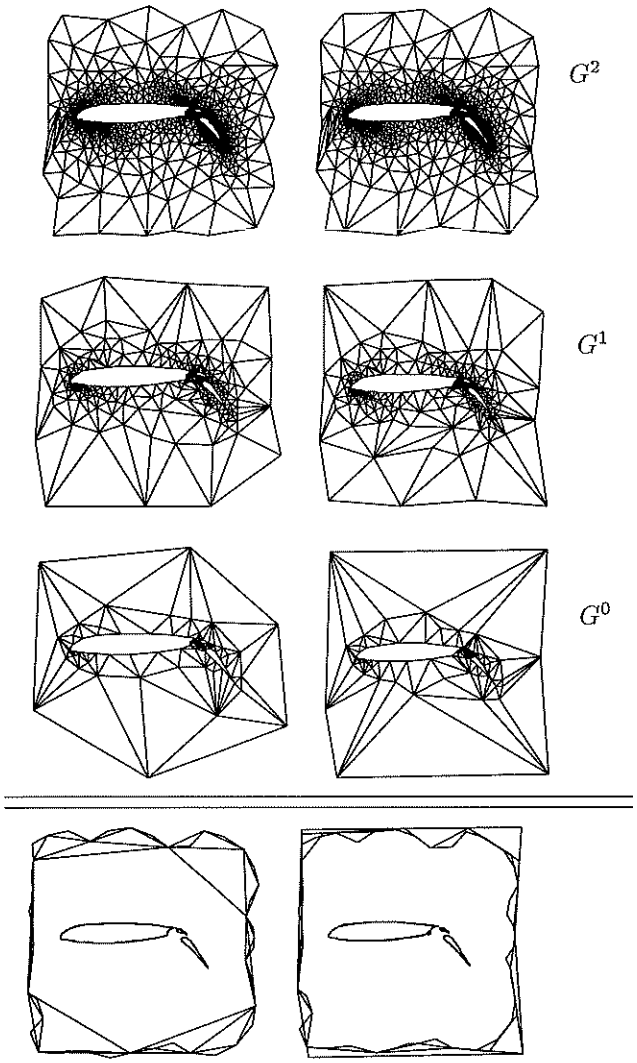

Figure 3.6: *Airfoil* grid hierarchy with unmodified boundaries (left) and modified boundaries (right).

each subdomain by some number of elements.

We solve a mildly varying coefficient problem on the airfoil:

$$\frac{\partial}{\partial x}((1+xy)\frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}((\sin(3y))\frac{\partial u}{\partial y}) = F(x,y),$$

where

$$F(x,y) = (4xy+2)\sin(3y) + 9x^2\cos(6y),$$

with either a purely Dirichlet boundary condition or a mixed boundary condition: Dirichlet for $x \leq 0.2$ and homogeneous Neumann for $x > 0.2$. For this problem, the non-homogeneous Dirichlet condition is $u = 2 + x^2\sin(3y)$.

Fig. 3.7 shows results when a hybrid 4-level multiplicative-additive Schwarz method is used (multiplicative between levels but additive among subdomains on the same level). As can be seen, deterioration of the method occurs with interpolant $\mathcal{I}_h^0$ when mixed boundary conditions are present. The next figure (Fig. 3.8) shows results for the multiplicative Schwarz method (both on the subdomains and between levels). This method behaves much like multigrid (see Table 1). In fact, this is nothing more than standard V-cycle multigrid with a block smoother used as a preconditioner. A V-cycle multigrid method with pointwise Gauss-Seidel smoothing and 2 pre- and 2 post-smoothings per level was used to produce the results in Table 1.

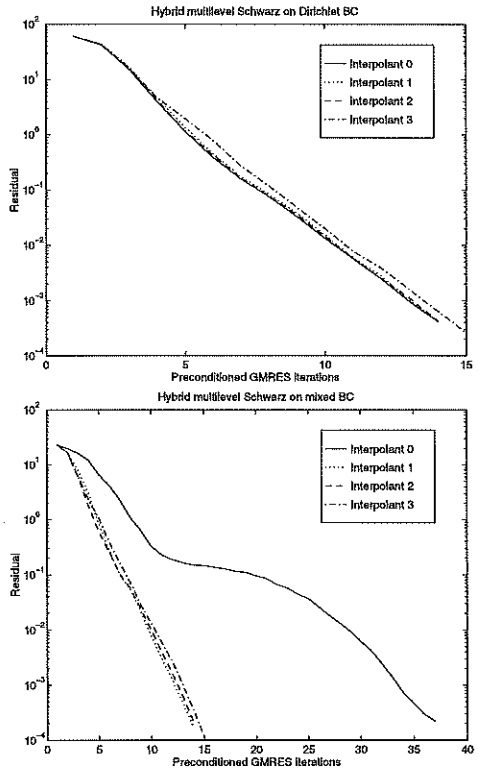

Figure 3.7: **Hybrid multiplicative-additive 4-level Schwarz**. Convergence history with purely Dirichet boundary conditions (top) or mixed boundary conditions (bottom).

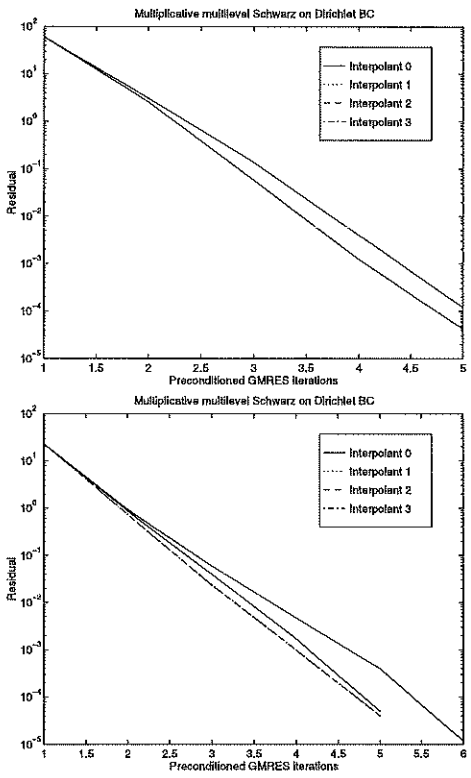

Figure 3.8: **Multiplicative 4-level Schwarz.** Convergence history with purely Dirichet boundary conditions (top) or mixed boundary conditions (bottom).

Table 1: **Multigrid.** Tables show the number of GMRES iterations to convergence with Dirichet or mixed boundary conditions.

Dirichlet boundary conditions

# of fine grid nodes	MG levels	# of coarse grid nodes	Interpolant Used			
			$\mathcal{I}_h^0$	$\mathcal{I}_h^1$	$\mathcal{I}_h^2$	$\mathcal{I}_h^3$
4253	2	1170	4	4	4	4
	3	340	4	4	4	4
	4	101	4	4	4	4

Mixed Dirichlet/Neumann boundary conditions

# of fine grid nodes	MG levels	# of coarse grid nodes	Interpolant Used			
			$\mathcal{I}_h^0$	$\mathcal{I}_h^1$	$\mathcal{I}_h^2$	$\mathcal{I}_h^3$
4253	2	1170	6	5	4	4
	3	340	6	4	5	5
	4	101	7	5	5	5

4 Agglomerated coarse spaces

4.1 Agglomerated multigrid methods on unstructured grids

In this section, we will consider a general agglomeration approach for constructing nested coarse spaces and transfer operators. The difference between this technique and the node-nested coarse spaces from the previous section is that here, we want to produce a nested sequence of spaces to be used in the multigrid method. The common point will be that our construction must satisfy the approximation and stability properties mentioned in the subspace correction framework in Section 2.

4.2 Coarse points and construction of macroelements

The agglomeration technique is based on the construction of a coarse grid with "macro-elements" consisting of unions of fine grid elements (triangles). An example of such a coarse grid is given on Fig. 4.1. Then, as in the standard finite element method, the basis functions in each coarse grid macroelement are appropriately defined. The coarse space V_H is then determined as the space spanned by these functions. If the coarse grid basis functions are defined as linear combinations of fine grid basis (i.e. the usual finite element basis), then V_H is a proper subspace of V_h, i.e. we obtain nested spaces by construction.

The construction of a basis in V_H is equivalent to the definition of the restriction matrix R_H, because the coordinates of these basis functions with respect to the fine grid basis form the rows of the restriction matrix. Thus, once the basis functions are defined, we have the restriction R_H, the interpolation (or prolongation) R_H^T, the coarse grid operator $R_H A R_H^T$ and we can apply the V-cycle algorithm from section 1.4.1.

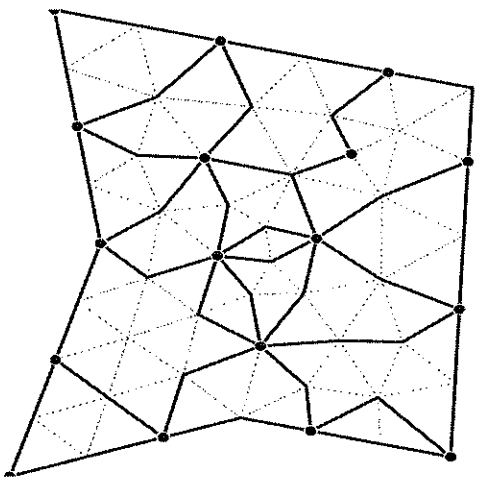

Figure 4.1: An example of macroelements

Although the nestedness is assured by construction, there are some important rules which should be followed in order to ensure good convergence rate of the resulting multilevel methods:

- *Smoothness:* The basis functions have to be smooth enough. This requirement is needed because the elements from V_H have to satisfy the stability property (2.11), which involves the A-norm of the coarse grid function.
- *Approximation:* The functions in V_H have to satisfy the approximation property (2.12). An implication of this is that the individual basis function in V_H cannot be independently chosen, and there must be some global relation which couples the basis functions.
- *Small supports:* The functions in V_H must have compact support. This requirement is based on the fact that once the basis $\{\Phi_i\}$ in V_H is given, then the coarse grid matrix elements are defined to be $a(\Phi_i, \Phi_j)$ (see (1.3)). Thus, if the basis functions have large supports, the coarse grid matrix will be dense, and the coarse grid problem is expensive to solve.
- *Conformity:* For finite element discretizations, it is desirable that the resulting coarse grid is formed by conformal macroelements, an analogue of conforming triangulations in finite element methods. This facilitates the analysis and construction of the algorithms.
- *Recursion:* The coarse grid should allow the recursive application of the algorithm to construct a multilevel method.

A careful look at these rules shows that it is difficult (even impossible) to satisfy all of them simultaneously. Usually some of them have to be weakened in order to satisfy others. For example, to have conforming macroelements of size $\approx 2h$ on an unstructured grid (which is a desirable choice in multigrid) is almost impossible. Taking smoother basis functions will increase the supports, and make the coarse grid operator denser.

A recent paper by Koobus, Lallemand and Dervieux [31] deals with agglomeration with the finite volume discretizations. The basis functions are piecewise constant on each cell, and the coarse grid cells are formed as unions of fine grid cells. A drawback of this algorithm is that the stability of the coarse grid basis functions is not easy to control.

An algebraic agglomeration algorithm can be found in the recent papers by Mandel, Vaněk, Brezina (see [32]) and Vaněk, Křížková [33]. Their approach uses an algebraically smoothed basis functions, and the coarse grid nodes are not explicitly defined. This allows the process of the basis construction to be more automatic, but it is more difficult to control the sparsity of the coarse grid operators.

Our approach is based first on the definition of the coarse grid points (using the MIS described in Section 3.1) and then using them to define the macroelements. The difference between the algorithm presented here and the agglomeration algorithms quoted above is in the more "geometrical" nature of our coarsening strategy. We will define our coarse grid space using macroelement edges and macroelement vertices (coarse points). In the numerical examples presented here, the definition the macroelement edges and coarse grid points is done using the dual graph of a given triangulation. Such an algorithm does not pretend to be computationally the best one, and we do not describe here this algorithm in detail. Our main concern will be the definition of a proper coarse space.

4.3 Coarse space basis functions.

Given the set of macroelements, we will now introduce three different ways to define the coarse grid basis functions. Let us first focus on meeting the first two rules given in the previous section: we need to define smooth basis functions so that the coarse space satisfy the approximation and stability properties.

To assure the approximation property, we should take a basis which preserves at least the constant function, i.e. the constant function must be always in the coarse space V_H. To do this, we first define basis functions possessing this property on the the macroelement boundaries, and after that we extend them into the interior of the macroelement as discrete harmonic functions. This extension obviously will not destroy the constant preserving property, because the constant function is harmonic.

We define the coarse basis functions on these edges as linear functions minimizing some quadratic functional. The $H^{1/2}$ norm on the boundary is one good choice for the quadratic functional, as it is the interface analogue of the A-norm.

Let the macroelement boundary be formed by ℓ edges from the fine grid, (see Fig 4.2) connecting two coarse grid points, x_0 and x_ℓ, and let this path contains the vertices $x_0, \ldots, x_\ell$. We define the basis function corresponding to the coarse grid node x_0 as follows:

Φ_0 is a linear function in $\mathbb{R}^2$:

$$\Phi_0 = a\xi + b\zeta + c, \quad (\xi, \zeta) \in \mathbb{R}^2.$$

We want : $\Phi_0(x_0) = 1$, $\Phi_0(x_\ell) = 0$. These are only two conditions and we have three parameters: a, b and c. To complete the set of conditions, we require that the function Φ_0 minimizes the functional (discrete $H^{1/2}$ norm):

$$F_{1/2}(\Phi_0) = \sum_{i=1}^{\ell} \sum_{j=i+1}^{\ell} \frac{h_i h_j}{h_{ij}^2} (\Phi_0(x_i) - \Phi_0(x_j))^2, \quad (4.1)$$

where h_i is the length of the edge (x_i, x_{i+1}) and $h_{ij} = |x_i - x_j|$. After using the first two conditions, this minimization is equivalent to minimization of a simple quadratic function of one variable which can be easily done analytically.

In the interior of the macroelements, we extend the basis functions by solving the equation:

$$a(\Phi_0, \phi) = 0, \qquad \text{for any } \phi \in V_h. \quad (4.2)$$

In this way, we define the basis in V_H and thus also defining the restriction operator R_H.

ALGORITHM 4.1 *(Geometric coarsening-Harmonic extension I)*

1. *The boundary values minimize the $H^{1/2}$-discrete norm.*
2. *For the nodes internal to the macroelements, the values of the basis functions are obtained via the harmonic extension, see Fig. 4.2.*

We will now give two simpler variants of this algorithm. The first one uses simpler boundary conditions: The function is defined on the boundary using the graph distance (see Fig 4.3). The graph distance $dist(i, j)$ is equal to the number of edges forming the shortest path connecting the vertices i and j, (in the case we consider these vertices are x_0 and x_ℓ).

ALGORITHM 4.2 *(Geometric coarsening-Harmonic extension II)*

1. *The boundary values are taken using the graph distance interpolation, see Fig. 4.3.*
2. *For the nodes internal to the macroelements, the values of the basis functions are obtained via the harmonic extension, see Fig. 4.3.*

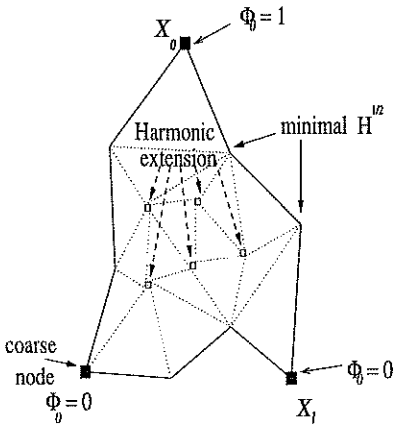

Figure 4.2: Minimize $H^{1/2}$ on the boundary and harmonic extension in the interior.

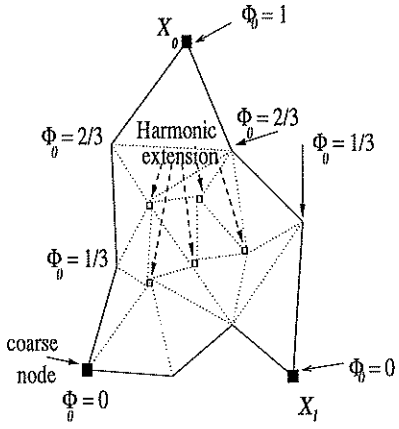

Figure 4.3: Graph distance on the boundary and smooth extension in the interior.

The second variant, which does not include harmonic extension is as follows:

ALGORITHM 4.3 *(Geometric coarsening)*

1. *On the macroelement boundaries the graph distance interpolation is used.*
2. *For the nodes internal to the macroelements and fictitious faces (see* Step 4*), the values of all basis functions whose supports form the macroelement take one and the same value: the reciprocal of the number of coarse points forming the macroelement, see Fig. 4.4.*

The approximation and stability properties of the agglomerated spaces given above are assured by the next lemma. A detailed proof will be included in the paper [6].

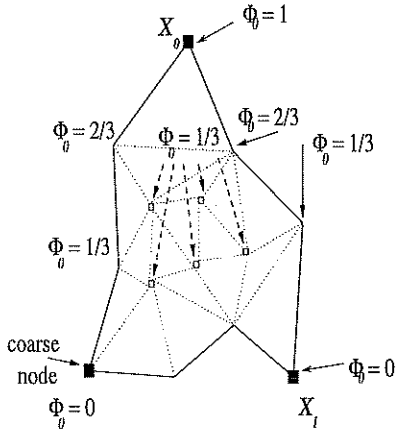

Figure 4.4: Graph distance on the boundary and in the interior.

Given a triangulation $\mathcal{T}_h$ and the corresponding linear finite element space $V_h \subset H^1(\Omega)$. Let $V_H \subset V_h$ be obtained by any of the three agglomeration algorithms described above. Let $Q_H : H^1(\Omega) \to V_H$ be the L^2-projection.

Lemma 4.1 *Assume that the constructed basis preserves the constant function. Then the following stability and approximation properties hold for the agglomerated subspaces:*

$$\|Q_H v\|_{1,\Omega} \leq c\|v\|_{1,\Omega}, \tag{4.3}$$

$$\|v - Q_H v\|_{0,\Omega} \leq cH|v|_{1,\Omega}, \quad \forall v \in H^1(\Omega). \tag{4.4}$$

4.4 Numerical examples

We consider an elliptic equation of following type:

$$\begin{cases} -\dfrac{\partial}{\partial x}\left(a(x,y)\dfrac{\partial u}{\partial x}\right) - \dfrac{\partial}{\partial x}\left(b(x,y)\dfrac{\partial u}{\partial x}\right) = 0, \\ u(x,y) = 1 \text{ on } \partial\Omega, \end{cases} \tag{4.5}$$

where $\Omega \subset \mathbb{R}^2$.

We use three types of coefficients for the equation (4.5) on three different grids. As a standard example we take $a = b = 1$, i.e. the Laplace operator.

In *Example 1*, the coefficients are mildly varying: $a(x,y) = (x^2 + y^2 + 1 + sin(x+y))$ and $b(x,y) = (x^2 + y^2 + 1 + cos(x+y))$. In *Example 2*, the coefficients are varying in the range $[10^{-3}, 30]$. For the grid

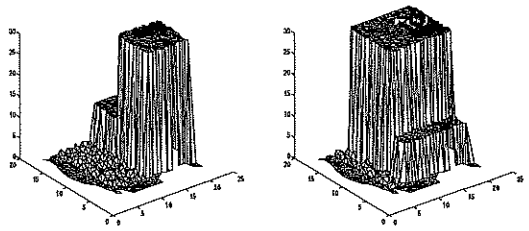

Figure 4.5: Surface plot of the coefficients for Example 2 — rapidly varying coefficients, $a(x,y)$ on the left, $b(x,y)$ on the right.

given in Fig. 4.6, the surface plot of the coefficients for *Example 2* are given in Fig. 4.5. For all the grids, the coefficients vary within the same range. In these experiments, we use the standard V-cycle preconditioner and the outer acceleration is done by the CG method. In the V-cycle , we use 1 pre- and 1 post-smoothing steps. The smoothing operator is forward Gauß-Seidel. The PCG iterations are terminated when the relative residual is less then 10^{-6}.

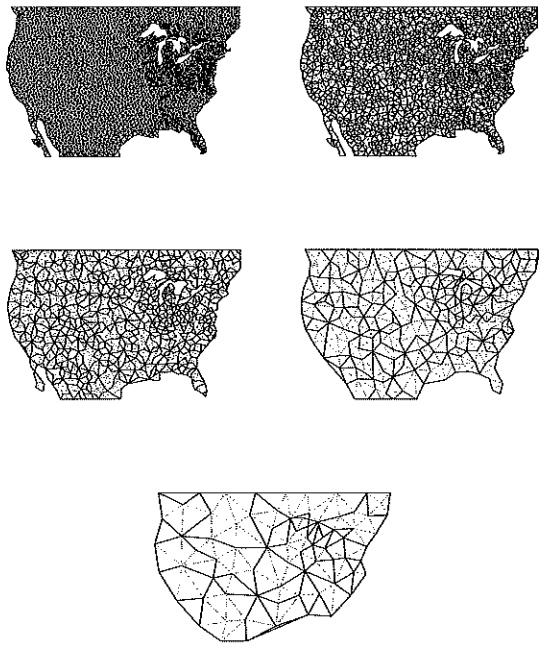

Figure 4.6: Macroelements for an unstructured grid level= 4 $N_h = 3422$; level= 3 $N_H^1 = 938$; level= 2 $N_H^2 = 268$; level= 1 $N_H^3 = 77$; level= 0 $N_H^4 = 21$.

In Figures 4.6–4.8, the macroelements are shown for different unstructured grids and different number of levels. Figure 4.9 shows the convergence histories for

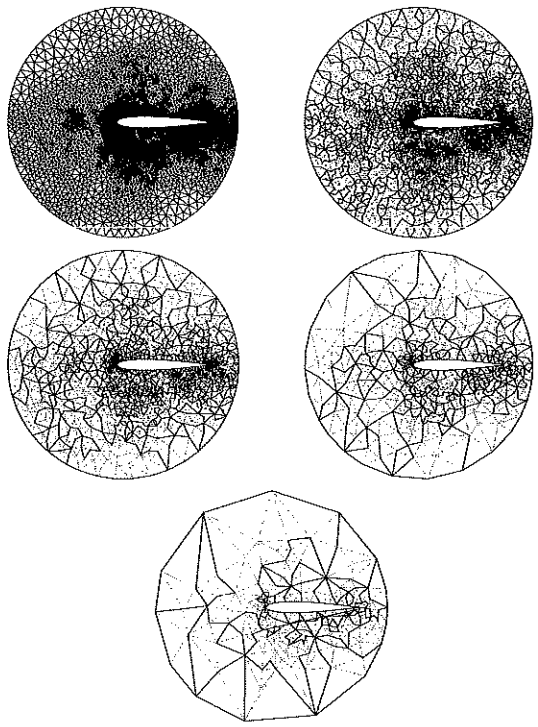

Figure 4.7: Macroelements for one element airfoil: level= 5 $N_h = 12665$; level= 4 $N_H^1 = 3404$; level= 3 $N_H^2 = 928$; level= 2 $N_H^3 = 257$; level= 1 $N_H^4 = 74$; level= 0 $N_H^5 = 24$.

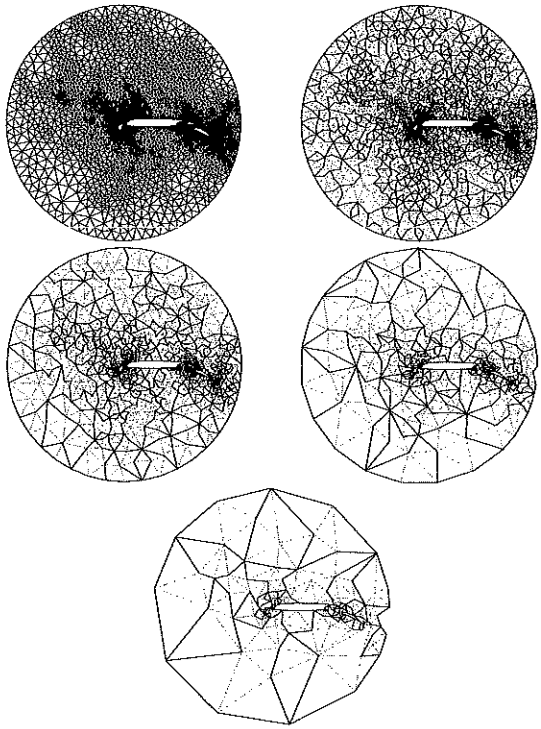

Figure 4.8: Macroelements for four element airfoil: level= 5 $N_h = 12850$; level= 4 $N_H^1 = 3444$; level= 3 $N_H^2 = 949$; level= 2 $N_H^3 = 270$; level= 1 $N_H^4 = 80$; level= 0 $N_H^5 = 26$.

the different types of coefficients and different grids. All these experiments were done using the simplest interpolation algorithm, Alg. 4.3. Figure 4.10 shows the convergence histories for a varying number of unknowns on two different grids: a one-element airfoil with one internal boundary, and a four-element airfoil with four internal boundaries. The numerical experiments were done using Algorithm 4.2.

These computational results show that the convergence is uniform with respect to the mesh size h. The convergence in the experiments shown in Figure 4.10 is a little better because the aspect ratio of the grids is better and Algorithm 4.2 was used instead of Algorithm 4.3. The behavior for roughly varying coefficients is not as good, as seen in Fig. 4.9. The airfoil grids used for the experiments in this section were produced using Barth's SIMPLEX2D mesh generator. We note here that these grids were generated at random, with no special attention being paid to the quality of the meshes, and does not reflect any deficiencies in the mesh generator.

4.5 Extensions

Other interpolations can also be constructed in an algebraic way. They are known as matrix-dependent prolongations (see M. Griebel in [46]). In this case, the prolongation operator is defined by $R_H^T = [A_{11}^{-1}A_{12}, I]^T$. Here, A_{11} and A_{12} are the the blocks in A_J formed by the natural splitting of the unknowns into two non-overlapping subsets corresponding to fine and coarse grid unknowns respectively:

$$A_J = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}. \tag{4.6}$$

The block A_{11} corresponds to the contributions $fine-fine$. Then it is straightforward to see that the coarse grid matrix is equal to the Schur complement of A: $S = A_{J-1} = A_{22} - A_{21}A_{11}^{-1}A_{12}$. Unfortunately, A_{11}^{-1} is generally a dense matrix, which is a serious drawback of using this approach. There are ways of defining approximations to this type of matrix-dependent prolongations as proposed by A. Reusken [30] and M.

Grid-Fig. 4.6

Example	Reduction factor
Laplace	0.11754
Example 1	0.12298
Example 2	0.24887

Grid-Fig. 4.7

Example	Reduction factor
Laplace	0.20701
Example 1	0.20724
Example 2	0.42600

Grid-Fig. 4.8

Example	Reduction factor
Laplace	0.21144
Example 1	0.21454
Example 2	0.46994

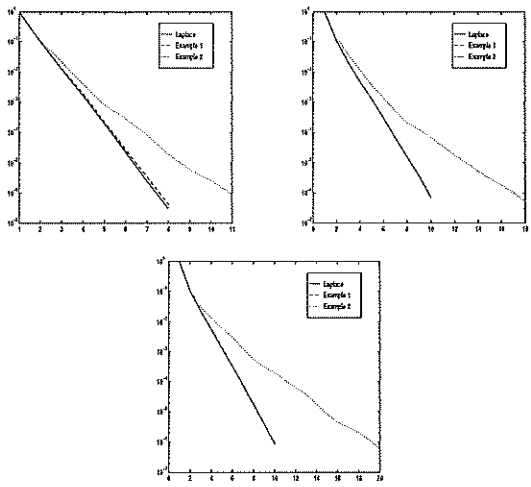

Figure 4.9: Convergence history and average reduction per iteration for Laplace equation, Example 1 and Example 2 on the different grids

Griebel in [46]. The agglomeration coarsening algorithm presented here and the aggregation given in [32] also might be viewed as ways of approximating the first entry $A_{11}^{-1}A_{12}$ in the prolongation with a sparse matrix.

4.5.1 Anisotropic problems

Another class of problems we have studied are the *anisotropic problems*. The problem in applying multigrid methods for such problems is that the smoother does not smooth the proper range of the high frequencies. A semi-coarsening (i.e. coarsening only in one direction) is often used to remedy this.

For anisotropic problems, the relevant changes in the agglomeration algorithm are straightforward. A dropping strategy can be used for the small off-diagonal

1-element airfoil

nodes	Reduction factor
$N_h = 72139$	0.16834
$N_h = 18152$	0.14236
$N_h = 4683$	0.14911
$N_h = 1315$	0.14087

4-element airfoil

nodes	Reduction factor
$N_h = 72233$	0.15454
$N_h = 18328$	0.13836
$N_h = 4870$	0.15488
$N_h = 1502$	0.14727

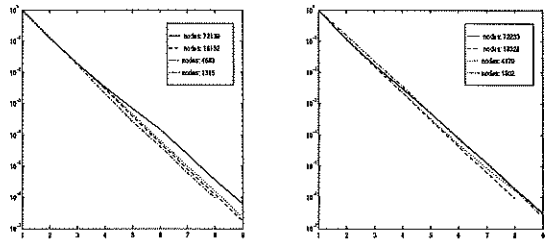

Figure 4.10: Convergence history and average reduction per iteration for varying number of unknowns

elements in A_k on each level. A new coarse grid operator $\tilde{A}_k$ is then obtained and this matrix corresponds to a new graph which is disconnected. Different dropping strategies can be applied (see [13]). Here we apply a simple one:
If

$$\sqrt{\frac{a_{ij}a_{ji}}{a_{ii}a_{jj}}} \leq 0.0001,$$

then set

$$a_{ii} := a_{ii} + a_{ij};\ a_{jj} := a_{jj} + a_{ji};\quad a_{ij} := 0;\ a_{ji} := 0.$$

Once this is done, we apply the usual *algebraic coarsening* algorithm [23]. In the next example, the algorithm which uses the dropping strategy is called *reduced graph algorithm*. Similar approaches for handling anisotropic problems can be found in [32].

The last numerical example in this section solves the Laplace equation with anisotropy introduced by the grid (see Fig. 4.11). The geometrical aspect ratio is of order 10^4. It can be seen that the algorithm which uses the anisotropic agglomeration is faster than the others.

4.6 Remarks

The agglomeration algorithms can provide a good approach for developing multilevel methods on unstructured grids. We have presented here a general technique for constructing basis for the coarse space satisfying stability and approximation properties. We have

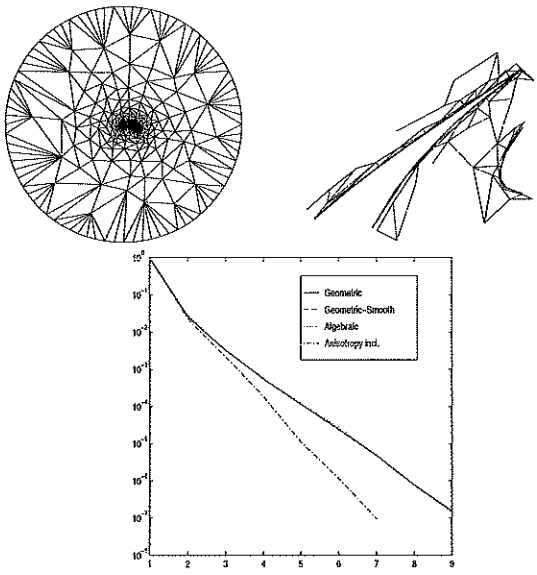

Figure 4.11: Macroelements (geometrical algorithm), second level geometric coarsening, zoomed 200 times near the airfoil. Convergence history for the stretched mesh: Geometrical coarsening; Geometrical coarsening-harmonic extension II; Algebraic coarsening; Reduced graph coarsening (anisotropy).

to point out that the general theory for the construction of agglomerated spaces on unstructured grids is still not fully developed. On the other hand, numerically these methods have good performance and can be applied to a large set of problems, including elliptic, anisotropic and convection dominated problems. For such experiments, we refer to Koobus, Lallemand and Dervieux [31], Mandel, Vaněk, Brezina (see [32]) and the experiments presented in this section.

We presented three different types of basis construction over agglomerated macroelements: Algorithm 4.1, Algorithm 4.2, Algorithm 4.3. We prefer to use Algorithm 4.3 for isotropic problems, because the convergence rate is as good as with the other two algorithms and this algorithm is simpler. The last numerical experiment we performed shows that for more complicated problems, such as anisotropic problems, the interpolation must be done depending of the direction of the anisotropy. In this case, the algorithms for constructing the coarse grid also need to be done very carefully, following the anisotropy direction.

5 Conclusions

There is no doubt that unstructured grids will be increasingly popular and the development of robust multilevel solvers on unstructured grids is important. We have presented two different approaches for constructing such solvers and they have their own areas of strength and weakness:

1. In some sense, the retriangulation algorithm is more natural and interfaces well with existing software (reusability). But problems may occur when solving 3D discrete equations because the retriangulation is not an easy task. The treatment of different types of boundary conditions must be done very carefully if one wishes to obtain a uniformly convergent iterative method.

2. The agglomeration algorithms offer many advantages: they are more algebraic, they produce nested coarse spaces, and are very robust. We have presented a framework for the design of agglomeration multilevel methods based on the stability and approximation properties of the underlying subspaces. Some of these methods have straightforward extension to 3D problems. The convergence of these types of methods is still not completely understood from a theoretical point of view.

Most importantly, we have shown that it is possible to design robust multilevel methods on unstructured grids which perform as efficiently as for structured grids.

References

[1] T. F. Chan, S. Go, and L. Zikatanov. Lecture notes on multilevel methods for elliptic problems on unstructured grids. Technical Report CAM 97–11, Department of Mathematics, University of California at Los Angeles, March 1997.

[2] T. F. Chan, B. Smith, and J. Zou. Overlapping Schwarz methods on unstructured meshes using non-matching coarse grids. *Numerische Mathematik*, 73(2):149–167, April 1996.

[3] T. F. Chan, B. Smith, and J. Zou. Multigrid and domain decomposition methods for unstructured meshes. In I.T. Dimov, Bl. Sendov, and P. Vassilevski, editors, *Proc. of the 3rd Int'l Conf. on Advances in Numerical Methods and Applications, Sofia, Bulgaria.*, pages 53–62. World Scientific, August 1994.

[4] T. F. Chan, S. Go, and J. Zou. Multilevel domain decomposition and multigrid methods for unstructured meshes: Algorithms and theory. Technical Report CAM-95–24, Department of Mathematics, University of California at Los Angeles, May 1995.

also in Proceedings of the Eighth International Conference on Domain Decomposition, May 1995, Beijing.

[5] T. F. Chan, S. Go, and J. Zou. Boundary treatments of multilevel methods on unstructured meshes. Technical Report CAM 96–30, Department of Mathematics, University of California at Los Angeles, September 1996.

[6] Tony F. Chan, J. Xu, and Ludmil Zikatanov. Agglomeration strategies in multigrid method. in preparation.

[7] G. Strang and G. Fix. *An Analysis of the Finite Element Method.* Prentice Hall, 1973.

[8] P. Ciarlet. *The finite element method for elliptic problems.* North-Holland, 1978.

[9] S. Brenner and L.R. Scott. *The mathematical theory of finite element methods.* Springer-Verlag, 1994.

[10] Claes Johnson. *Numerical solution of partial differential equations by the finite element method.* Cambridge University Press, Cambridge, 1990.

[11] D.L. Brown, G. Chesshire, W.D. Henshaw, and H.-O. Kreiss. On composite overlapping grids. In T.J. Chung and G.R. Karr, editors, *Finite Element Analysis in Fluids*, pages 544–559. UAH Press, 1989.

[12] G. Chesshire and W. D. Henshaw. Composite overlapping meshes for the solution of partial differential equations. *J. Comp. Phys.*, 90(1):1–64, 1990.

[13] Yousef Saad. *Iterative methods for sparse linear systems.* PWS Publishing company, 1996.

[14] William Briggs. *A multigrid tutorial.* Society for Industrial and Applied Mathematics, Philadelphia, 1987.

[15] J. Bramble. *Multigrid methods.* Pitman, Notes on Mathematics, 1994.

[16] W. Hackbusch. *Multi-grid methods and applications.* Springer Verlag, New York, 1985.

[17] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34:581–613, December 1992.

[18] J. Xu. *An introduction to multilevel methods.* To be published by Oxford University Press, 1996. Lecture notes: VIIth EPSRC Numerical analysis summer school, Leicester University, UK.

[19] T. F. Chan and T. Mathew. Domain decomposition algorithms. *Acta Numerica*, pages 61–143, 1994.

[20] B. Smith, P. Bjorstad, and W. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations.* Cambridge University Press, Cambridge, 1996.

[21] Maksymilian Dryja and Olof B. Widlund. Some domain decomposition algorithms for elliptic problems. In Linda Hayes and David Kincaid, editors, *Iterative Methods for Large Linear Systems*, pages 273–291, San Diego, California, 1989. Academic Press.

[22] Maksymilian Dryja and Olof B. Widlund. Additive Schwarz methods for elliptic finite element problems in three dimensions. In Tony F. Chan, David E. Keyes, Gérard A. Meurant, Jeffrey S. Scroggs, and Robert G. Voigt, editors, *Fifth Conference on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1992. SIAM.

[23] J. W. Ruge and K. Stüben. Algebraic multigrid. In S. F. McCormick, editor, *Multigrid methods*, Frontiers in applied mathematics, pages 73–130, Philadelphia, Pennsylvania, 1987. SIAM.

[24] D.J. Mavriplis. Unstructured mesh algorithms for aerodynamic calculations. Technical Report 92-35, ICASE, NASA Langley, Virginia, July 1992.

[25] H. Guillard. Node-nested multi-grid method with Delaunay coarsening. Technical Report RR-1898, INRIA, Sophia Antipolis, France, March 1993.

[26] T. F. Chan and Barry Smith. Domain decomposition and multigrid methods for elliptic problems on unstructured meshes. In David Keyes and Jinchao Xu, editors, *Domain Decomposition Methods in Science and Engineering, Proceedings of the Seventh International Conference on Domain Decomposition, October 27-30, 1993, The Pennsylvania State University.* American Mathematical Society, Providence, 1994. also in Electronic Transactions on Numerical Analysis, v.2, (1994), pp. 171–182.

[27] R. E. Bank and J. Xu. An algorithm for coarsening unstructured meshes. Technical report, Dept. of Math., Univ. of Calif. at San Diego, 1994.

[28] W. Hackbusch. *Iterative solution of large sparse systems of equations.* Springer Verlag, Heidelberg, 1993. Applied Mathematical Sciences, Vol. 95.

[29] D. Braess. Towards algebraic multigrid for elliptic problems of second order. *Computing*, 55:379–393, 1995.

[30] A. A. Reusken. A multigrid method based on incomplete Gaussian elimination. Technical Report RANA 95-13, Eindhoven University of Technology, October 1995.

[31] B. Koobus, M. H. Lallemand, and A. Dervieux. Unstructured volume-agglomeration MG: solution of the Poisson equation. *International Journal for Numerical Methods in Fluids*, 18(1):27–42, 1994.

[32] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multi-grid by smoothed aggregation for second and forth order elliptic problems. *Computing*, 1995. to appear.

[33] P. Vaněk and J. Křížkovà. Two-level method on unstructured meshes with convergence rate independent of the coarse space size. Technical Report 33, University of Colorado at Denver, January 1995.

[34] A. M. Matsokin and S. V. Nepomnyaschikh. A Schwarz alternating method in a subspace. *Soviet Mathematics*, 29(10):78–84, 1985.

[35] P. Lions. On the Schwarz alternating method. I. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1988.

[36] M. Dryja and O. Widlund. An additive variant of the Schwarz alternating method for the case of many subregions. Technical Report 339, also Ultracomputer Note 131, Department of Computer Science, Courant Institute, 1987.

[37] James H. Bramble, Joseph E. Pasciak, Junping Wang, and Jinchao Xu. Convergence estimates for multigrid algorithms without regularity assumptions. *Math. Comp.*, 57(195):23–45, 1991.

[38] T. F. Chan and J. Zou. Additive Schwarz domain decomposition methods for elliptic problems on unstructured meshes. *Numerical Algorithms*, 8:329–346, 1994.

[39] T. F. Chan and J. Zou. A convergence theory of multilevel additive Schwarz methods on unstructured meshes. Technical Report 95-16, Department of Mathematics, University of California at Los Angeles, March 1995. to appear in Numerical Algorithms.

[40] Ralf Kornhuber and Harry Yserentant. Multilevel methods for elliptic problems on domains not resolved by the coarse grid. In David Keyes and Jinchao Xu, editors, *Domain Decomposition Methods in Science and Engineering, Proceedings of the Seventh International Conference on Domain Decomposition, October 27-30, 1993, The Pennsylvania State University*, volume 180, pages 49–60. American Mathematical Society, Providence, 1994.

[41] R. Bank and J. Xu. An algorithm for coarsening unstructured meshes. *Numer. Math.*, 73:1–23, 1996.

[42] P. Clément. Approximation by finite element functions using local regularization. *R.A.I.R.O. Numer. Anal.*, R-2:77–84, 1975.

[43] Xiao-Chuan Cai. The use of pointwise interpolation in domain decomposition methods with nonnested meshes. *SIAM J. Sci. Comp.*, 16(1), 1995.

[44] William D. Gropp and Barry F. Smith. Portable Extensible Toolkit for Scientific Computation (PETSc). Available by anonymous ftp at `info.mcs.anl.gov` in the directory `pub/pdetools` or at `http://www.mcs.anl.gov/Projects/petsc/petsc.html`.

[45] A. Pothen, H. Simon, and K.P. Liou. Partitioning sparse matrices with eigenvector of graphs. *SIAM J. Mat. Anal. Appl.*, 11(3):430–452, 1990.

[46] T. Grauschopf, M. Griebel, and H. Regler. Additive multilevel preconditioners based on bilinear interpolation, matrix-dependent geometric coarsening and algebraic multigrid coarsening for second-order elliptic PDEs. *Applied Numerical Mathematics*, 23(1):63–97, february 1997. special issue on multilevel methods.