Domain Decomposition Algorithms for the Partial Differential Equations of Linear Elasticity

Barry F. Smith

Technical Report 517

Courant Institute of Mathematical Sciences

September 1990

This work was supported in part by the National Science Foundation under grant NSF-CCR-8903003 and in part by the Norwegian Research Council for Science and the Humanities under grant D.01.08.054 and The Royal Norwegian Council for Scientific and Industrial Research under grant IT2.28.28484 at the University of Bergen.

Domain Decomposition Algorithms for the Partial Differential Equations of Linear Elasticity

Barry F. Smith

May 25, 1994

A dissertation in the Department of Mathematics submitted to the faculty of the Graduate School of Arts and Sciences in partial fulfillment of the requirements for the degree of Doctor of Philosophy at New York University.

Approved: ________________________

Olof Widlund
Adviser

Abstract

The use of the finite element method for elasticity problems results in extremely large, sparse linear systems. Historically these have been solved using direct solvers like Choleski's method. These linear systems are often ill-conditioned and hence require good preconditioners if they are to be solved iteratively. We propose and analyze three new, parallel iterative domain decomposition algorithms for the solution of these linear systems. The algorithms are also useful for other elliptic partial differential equations.

Domain decomposition algorithms are designed to take advantage of a new generation of parallel computers. The domain is decomposed into overlapping or non-overlapping subdomains. The discrete approximation to a partial differential equation is then obtained iteratively by solving problems associated with each subdomain. The algorithms are often accelerated using the conjugate gradient method.

The first new algorithm presented here borrows heavily from multi-level type algorithms. It involves a local change of basis on the interfaces between the substructures to accelerate the convergence. It works well only in two dimensions.

The second algorithm is optimal in that the condition number of the iteration operator is bounded independently of the number of subdomains and unknowns. It uses non-overlapping subdomains, but overlapping regions of the interfaces between subdomains. This is an additive Schwarz algorithm, which works equally well in two or three dimensions.

The third algorithm is designed for problems in three dimensions. It includes a coarse problem associated with the unknowns on the wirebaskets of the subdomains. The new method offers more potential parallelism than previous algorithms proposed for three dimensional problems since it allows for the simultaneous solution of the coarse problem and the local problems.

Acknowledgements

I would like to thank my adviser Olof Widlund for all the time and energy he has devoted to me and my work. I would also like to thank my Norwegian colleagues Petter Bjørstad and Anders Hvidsten who have made much of this work possible. In addition, I would like to thank Anne Greenbaum and all of the other numerical analysts at the Courant Institute, including Marsha Berger, Michael Overton, Jim Demmel, Leslie Greengard, Jonathan Goodman and Tom Hou.

I would also like to thank VERITAS SESAM SYSTEMS, Høvik, Norway, for the use and access to SESAM for the generation of test stiffness matrices which were used in the numerical experiments reported in this thesis.

Barry Smith

Contents

Chapter 1

Introduction

1.1 An Overview

The finite element method for elliptic partial differential equations results in extremely large, sparse linear systems. We propose and analyze three new, parallel, iterative algorithms for the solution of these linear systems. The underlying iterative scheme is often the preconditioned conjugate gradient method where we construct the preconditioner using domain decomposition techniques.

Domain decomposition algorithms are designed to take advantage of a new generation of parallel computers. The domain is decomposed into overlapping or non-overlapping subdomains. In the former case the algorithms are often referred to as Schwarz methods, in the latter they are called iterative substructuring methods. This distinction is not always clear. For instance, Dryja and Widlund [36] have shown that the basic iterative substructuring algorithm, see Section 3.2, can be analyzed as a Schwarz method. For a discussion of the relationship between the overlapping and non-overlapping schemes, see also Bjørstad and Widlund [10] and Chan and Goovaerts [21].

The discrete approximation to a partial differential equation is obtained iteratively by solving problems associated with each subdomain and passing information between neighbors. When a large number of subdomains are used, a global problem, which involves one or a few unknowns from each subdomain, must also be included since otherwise the convergence rate deteriorates rapidly with the number of subdomains. The best of these algorithms have condition numbers which are bounded independently of the number of subdomains and unknowns or which grow like $(1+\log(H/h))^2$. Here H is the diameter of a subdomain and h is the diameter of an element. These algorithms are often accelerated using the conjugate gradient method. The particular domain decomposition algorithm is then defined by the preconditioner for the conjugate gradient method.

The alternating Schwarz method [75] is believed to be the first domain decomposition algorithm. In its original form it involves solving problems on two subdomains serially. We discuss this algorithm more completely in Chapter 2. Domain decomposition algorithms where the subproblems are solved in such a serial fashion like the alternating Schwarz method are referred to as multiplicative schemes. In [35], Dryja and Widlund show how Schwarz's algorithm can be restructured so that all the subproblems may be solved in parallel. In this case the methods are referred to as additive algorithms. Xu discusses many domain decomposition algorithms in [86]. He uses the terms recursive subspace correction (RSC) and simultaneous subspace correction (SSC) for multiplicative and additive cases, respectively.

The first new algorithm presented here borrows heavily from the literature on multi-level, multigrid type algorithms. It involves a local change of basis on the interfaces between the substructures to accelerate the convergence. It works well only in two dimensions.

The second new algorithm is optimal in the sense that the condition number of the iteration operator is bounded independently of the number of subdomains and unknowns. The subdomains do not overlap, but overlapping subregions of the interfaces between the subdomains are used. This is an additive Schwarz algorithm; it works equally well in two or three dimensions.

The third new algorithm is an iterative substructuring algorithm designed for problems in three dimensions. It includes a coarse problem associated with the unknowns on the wirebaskets of the subdomains. The new algorithm offers more potential parallelism than previous, similar algorithms proposed for three dimensional problems because it allows for the simultaneous solution of the coarse problem and the local problems.

The structure of this thesis is as follows. In Chapter 1 we discuss the equations of linear elasticity and the finite element method. Chapter 2 is devoted to the basic algorithms and tools of domain decomposition. In Chapters 3, 4, and 5 we introduce and analyze our three new algorithms. We analyze the operation counts for some basic domain decomposition algorithms in Chapter 6.

Portions of this thesis have already appeared as technical reports, two of which have been accepted for publication. Parts of Chapter 3 are contained in [78]. The algorithm in Chapter 4 was introduced in [77]. The main new result in Chapter 5 will appear in [76].

One of the recurring themes in this thesis is the relative ease with which different domain decomposition ideas can be combined to produce a multitude of new algorithms. Another is the recognition that many of the proposed preconditioners are merely block diagonal preconditioners in a new basis suggested by the underlying geometry. A third is that just as the global stiffness matrix is constructed from pieces corresponding to the subdomains, the preconditioner can be constructed in a similar manner.

We now briefly review some of the literature. Papers which relate directly to this thesis will be discussed more completely in the following chapters.

Very early, Sobolev [79] showed that the Schwarz alternating method converges for the equations of linear elasticity. More recent work on elasticity has been done by several scientists. Using the Neumann-Dirichlet algorithm, Bjørstad and Hvidsten [6] have solved actual industrial problems with some success. De Roeck [28], De Roeck, Le Tallec, and Vidrascu [56] and De Roeck and Le Tallec [29] have implemented the algorithm proposed in Bourgat, Glowinski, Le Tallec, and Vidrascu [11] for elasticity problems. This algorithm involves solving a Dirichlet problem and Neumann problem for each subdomain at each iteration. Hughes and others have used element-by-element preconditioning [46],[82], on large structural problems. The preconditioned problems for these latter two algorithms can require hundreds of conjugate gradient iterations.

For the p-version finite element method, Mandel has analyzed iterative substructuring type algorithms for elasticity [60],[61],[62],[63]. Other work for the p-version finite element method has been carried out by Babuška, Craig, Mandel, and Pitkäranta [1], and Babuška, Griebel, and Pitkäranta [2]. For the p-version finite element method, the elements themselves, which are associated with many degrees of freedom, are treated as subdomains.

Russian mathematicians are also working on domain decomposition algorithms for elasticity and related elliptic problems. In [54], Kolotilina and Yermin have performed numerical experiments on the three dimensional Navier equations using a block SSOR with a particular ordering inspired by the geometry. This work is also for the p-version finite element method. In the preconditioned problems, they observe only a slight growth in the condition number as p, the degree of the finite element basis functions, increases. Their scheme has some similarities to a multiplicative version of Mandel's work cited above. However, Kolotilina, and Yermin work with a fixed number of p-elements and do not explore the behavior of the condition number as a function of the number of p-elements.

In [49], Kaporin, Kolotilina, and Yermin continue these experiments and advocate the merging of neighboring p-elements and the use of an iterative method to solve the reduced, Schur complement, system. They also find that for larger p performing an incomplete, approximate factorization of the blocks, rather than a complete factorization, is more computationally efficient. The incomplete factorization they use is discussed in Kolotilina and Yermin [53].

The work of Nepomnyaschikh [68] contains some important theory which is helpful in analyzing a variety of domain decomposition algorithms. One of his important results is introduced in detail in Chapter 2. The work by Matsokin and Nepomnyaschikh [66] discusses a Schwarz alternating algorithm which has some similarities to the algorithm presented in Chapter 4.

Much work using domain decomposition has focused on the scalar elliptic problem, including Bjørstad and Widlund [8],[9]; Bramble, Pasciak, and Schatz [12],[15]; Chan and Resasco [23],[24]; Dryja [33],[34]; Dryja and Widlund [35],[36]; and Widlund [84]. Experimental work on the relative effectiveness of various domain decomposition algorithms on several types of parallel computers have been performed by Gropp and Keyes [50],[51],[52]. Gropp and Keys have experimented with non-self-adjoint problems and with problems in fluid dynamics.

For recent extensions of the theory for domain decomposition algorithms to non-self-adjoint elliptic equations and parabolic problems, see Cai [19],[18] and Cai and Widlund [20].

1.2 Sobolev Spaces

Sobolev space methods provide powerful tools in the study of elliptic problems and are also extremely useful in analyzing finite element methods. We introduce the basic concepts in this section. To fix some notations, we let x represent an element in a bounded Lipschitz region $\Omega \subset R^n$ of diameter $O(H)$. Let u, v, f, g be scalar valued functions. Bold face (e.g. $\mathbf{u}$) will be reserved for vector valued functions and subscripted variables such as u_i will indicate the ith component of a vector valued function. The number of components of $\mathbf{u}$ will be denoted by q. C and c will be generic constants. We need only work with the real Sobolev spaces.

1.2.1 The Norms

The L^2 norm is defined by

$$||u||_{L^2(\Omega)}^2 = \int_\Omega u^2 dx.$$

The space L^2 is defined to contain all functions u for which $||u||_{L^2(\Omega)} < \infty$. The H^1 semi-norm is defined by

$$|u|_{H^1(\Omega)}^2 = \int_\Omega (\nabla u) \cdot (\nabla u) dx.$$

We define the weighted H^1 norm by

$$||u||_{H^1(\Omega)}^2 = |u|_{H^1(\Omega)}^2 + \frac{1}{H^2} ||u||_{L^2(\Omega)}^2.$$

The space $H^1(\Omega)$ consists of all functions u for which both u and all of its first derivatives are in $L^2(\Omega)$. The space $H_0^1(\Omega)$ is the subspace of $H^1(\Omega)$ consisting of all functions $u \in H^1(\Omega)$, which are the limits of $C_0^\infty(\Omega)$ functions. The space $C_0^\infty(\Omega)$ consists of infinitely continuously differentiable functions whose support lies in Ω.

The definitions are easily extended for vector valued functions by

$$||\mathbf{u}||_{L^2(\Omega)}^2 = \sum_{i=1}^q ||u_i||_{L^2(\Omega)}^2,$$

and

$$||\mathbf{u}||_{H^1(\Omega)}^2 = \sum_{i=1}^q ||u_i||_{H^1(\Omega)}^2.$$

When we speak of the H^1 space of vector valued functions, we write $(H^1(\Omega))^q$, however, when we explicitly express the norm, we will drop this notation and write

$$||\mathbf{u}||_{H^1(\Omega)} \quad \text{rather than} \quad ||\mathbf{u}||_{(H^1(\Omega))^q}.$$

1.2.2 Three Useful Lemmas

The following three inequalities establish equivalences of certain norms in some subspaces of $H^1(\Omega)$.

Lemma 1.2.1 (**Friedrichs' inequality**) *There exists a positive constant* $C(\Omega)$ *such that, for all* $u \in H_0^1(\Omega)$,

$$||u||_{H^1(\Omega)} \leq C(\Omega)|u|_{H^1(\Omega)}.$$

Lemma 1.2.2 (**Poincaré's inequality**) *There exists a positive constant* $C(\Omega)$ *such that, for all* $u \in H^1(\Omega)$,

$$||u||^2_{H^1(\Omega)} \leq C(\Omega)(|u|^2_{H^1(\Omega)} + \frac{1}{H^{2+n}}(\int_\Omega u dx)^2).$$

Lemma 1.2.3 (**Poincaré-Friedrichs' inequality**) *If* $\Gamma_0 \subset \partial\Omega$ *is of positive measure then there exists a positive constant* $C(\Omega, \Gamma_0)$ *such that, for all* $u \in H^1(\Omega)$,

$$||u||^2_{H^1(\Omega)} \leq C(\Omega, \Gamma_0)(|u|^2_{H^1(\Omega)} + \frac{1}{H}\int_{\Gamma_0} u^2).$$

Proofs of Poincaré's inequality may be found in [38],[69],[71], Friedrichs' inequality may be found in [71] and Poincaré-Friedrichs' inequality in [69].

1.3 Trace Spaces

Certain trace spaces play an important role in the analysis of many domain decomposition algorithms. We consider the same Lipschitz region Ω as in the previous section.

Let Γ be a simple, closed curve (surface) in Ω. We define

$$|u|^2_{H^{1/2}(\Gamma)} = \min_{\tilde{u}|_\Gamma = u} |\tilde{u}|^2_{H^1(\Omega)}.$$

An equivalent semi-norm is given by,

$$\int_\Gamma \int_\Gamma \frac{|u(x) - u(y)|^2}{|x-y|^n} ds(x) ds(y),$$

cf. Miranda [67]. To define a full norm on Γ, we use

$$||u||^2_{H^{1/2}(\Gamma)} = |u|^2_{H^{1/2}(\Gamma)} + \frac{1}{H}||u||^2_{L^2(\Gamma)}.$$

If $\hat{\Gamma} \subset \Gamma$ then we define

$$|\hat{u}|^2_{H^{1/2}_{00}(\hat{\Gamma})} = |u|^2_{H^{1/2}(\Gamma)}, \qquad \text{where } u|_{\hat{\Gamma}} = \hat{u} \text{ and } u|_{\Gamma\setminus\hat{\Gamma}} = 0.$$

$H^{1/2}_{00}(\hat{\Gamma})$ is a strict subspace of $H^{1/2}(\Gamma)$. A norm equivalent to $H^{1/2}_{00}(\hat{\Gamma})$ is given by

$$|\hat{u}|^2_{H^{1/2}_{00}(\hat{\Gamma})} = |\hat{u}|^2_{H^{1/2}(\hat{\Gamma})} + \int_{\hat{\Gamma}} \frac{u^2(x)}{\rho(x)} ds(x),$$

where $\rho(x)$ is the distance from x to the boundary of $\hat{\Gamma}$. For a more detailed discussion of trace spaces; cf. Grisvard [43] or Lions and Magenes [57].

1.4 Ellipticity

1.4.1 The Scalar Case

We are interested in partial differential equations of the form

$$-\sum_i \sum_j \frac{\partial}{\partial x_i} a_{ij}(x) \frac{\partial u}{\partial x_j} = f, \qquad \text{in } \Omega, \tag{1.1}$$

$$u = 0, \qquad \text{on } \Gamma_0 \subset \partial\Omega, \tag{1.2}$$

$$\sum_j \sum_i a_{ij}(x) \frac{\partial u}{\partial x_j} \hat{n}_i = g, \qquad \text{on } \Gamma_1 = \partial\Omega \setminus \Gamma_0. \tag{1.3}$$

Here $\hat{\mathbf{n}}$ is the unit outward normal to $\partial\Omega$. We require that the measure of Γ_0 be strictly greater than zero. This insures a unique solution to problem (1.1-1.3).

To obtain a variational form for this equation, we note that,

$$\frac{\partial}{\partial x_i} v a_{ij}(x) \frac{\partial u}{\partial x_j} = v \frac{\partial}{\partial x_i} a_{ij}(x) \frac{\partial u}{\partial x_j} + a_{ij}(x) \frac{\partial v}{\partial x_i} \frac{\partial u}{\partial x_j}.$$

Assuming enough smoothness, we can integrate over Ω to obtain

$$\int_{\partial\Omega} v \hat{n}_j a_{ij}(x) \frac{\partial u}{\partial x_j} = \int_\Omega \frac{\partial}{\partial x_i} v a_{ij}(x) \frac{\partial u}{\partial x_j} = \int_\Omega v \frac{\partial}{\partial x_i} a_{ij}(x) \frac{\partial u}{\partial x_j} + \int_\Omega a_{ij}(x) \frac{\partial v}{\partial x_i} \frac{\partial u}{\partial x_j}.$$

Summing over i and j, we obtain an important Green's formula

$$\int_{\partial\Omega} \sum_i \sum_j v \hat{n}_i a_{ij}(x) \frac{\partial u}{\partial x_j} = \int_\Omega \sum_i \sum_j v \frac{\partial}{\partial x_i} a_{ij}(x) \frac{\partial u}{\partial x_j} + \int_\Omega \sum_i \sum_j a_{ij}(x) \frac{\partial v}{\partial x_i} \frac{\partial u}{\partial x_j}. \tag{1.4}$$

Let V be the set of all v such that $v \in H^1(\Omega)$ and $v|_{\Gamma_0} = 0$. Using equation (1.4), a weak form of the partial differential equation (1.1-1.3) is

$$\int_\Omega \sum_i \sum_j a_{ij}(x) \frac{\partial v}{\partial x_i} \frac{\partial u}{\partial x_j} = \int_\Omega fv + \int_{\Gamma_1} gv, \quad u \in V, \ \forall v \in V. \tag{1.5}$$

We introduce the bilinear form $a_\Omega(u, v)$ and the linear form $f(v)$ by rewriting equation (1.5) as

$$a_\Omega(u, v) = f(v), \quad u \in V, \ \forall v \in V. \tag{1.6}$$

For any symmetric, positive definite problem the related minimization problem is given by

$$\min_{u \in V} J(u) = \min_{u \in V} \frac{1}{2} a_\Omega(u, u) - f(u). \tag{1.7}$$

We are interested in when solutions to equations (1.6) and (1.7) exist and whether they are the same. The related problem of whether the solution of equation (1.6) is a classical solution of (1.1-1.3) will not be discussed, cf. [38],[48],[57].

Let W be a Hilbert space with the inner product $(\cdot, \cdot)$ and norm $||\cdot||$. We review the following definitions.

- The bilinear form $a(\cdot, \cdot)$ is *continuous* if there exists a constant $C > 0$ such that
$$|a(u, v)| \leq C||u||||v||, \qquad \forall u, v \in W.$$
- The bilinear form $a(\cdot, \cdot)$ is *strongly elliptic* or *coercive* if there exists a constant $c > 0$ such that
$$a(v, v) \geq c||v||^2, \qquad \forall v \in W.$$
- The linear form $f(\cdot)$ is *continuous* if there exists $c > 0$ such that
$$|f(v)| \leq c||v||, \qquad \forall v \in W.$$

The following abstract theorem allows us to determine conditions on $a_{ij}(x)$ for existence and uniqueness of solutions to equation (1.6).

Theorem 1.4.1 (**Lax-Milgram Lemma** [26],[55]) *Let W be a real Hilbert space, let $a(\cdot, \cdot)$ be a continuous, strongly elliptic (coercive) bilinear form defined on W, and let f be a continuous linear form on W. Then there exists one and only one element $u \in W$ which satisfies*

$$a(u, v) = f(v), \qquad \forall v \in W. \tag{1.8}$$

Lemma 1.4.1 *If the same hypothesis hold as in the Lax-Milgram Lemma and $a(\cdot,\cdot)$ is symmetric then*

$$\min_{u \in W} \frac{1}{2} a(u,u) - f(u)$$

exists and the minimizing u satisfies (1.8).

To apply the Lax-Milgram Lemma to the variational problem (1.6), we use the fact that $H^1(\Omega)$ is a Hilbert space. We need to verify that $a_\Omega(\cdot,\cdot)$ is strongly elliptic (coercive), i.e.

$$a_\Omega(u,u) \geq c||u||^2_{H^1(\Omega)}, \qquad \forall u \in V.$$

We therefore require

$$\sum_i \sum_j y_i a_{ij}(x) y_j \geq c \sum_i y_i^2, \qquad \forall x \in \Omega,\ \forall y \in R^n.$$

It follows that

$$\begin{aligned} a_\Omega(u,u) &= \int_\Omega \sum_i \sum_j \frac{\partial u}{\partial x_i} a_{ij}(x) \frac{\partial u}{\partial x_j} \\ &\geq c \int_\Omega (\nabla u)^2 \\ &= c|u|^2_{H^1(\Omega)} \\ &\geq c||u||^2_{H^1(\Omega)}. \end{aligned}$$

The final step makes use of the Poincaré-Friedrichs' inequality, and the fact that u is zero on Γ_0.

1.4.2 The Vector Valued Case

We can apply the same procedure to the vector valued case. Consider the partial differential equation,

$$\begin{aligned} -\sum_j \sum_k \sum_l \frac{\partial}{\partial x_j} \beta_{ijkl}(x) \frac{\partial u_k}{\partial x_l} &= f_i, && \text{in } \Omega, \\ \mathbf{u} &= 0, && \text{on } \Gamma_0 \subset \partial\Omega, \\ \sum_j \sum_k \sum_l \beta_{ijkl}(x) \frac{\partial u_k}{\partial x_l} \hat{n}_j &= g_i, && \text{on } \Gamma_1 = \partial\Omega \setminus \Gamma_0. \end{aligned}$$

Again, using a Green's formula for each component of $\mathbf{u}$ separately, we obtain the variational problem,

$$a_\Omega(\mathbf{u}, \mathbf{v}) = f(\mathbf{v}), \quad \mathbf{u} \in \mathbf{V}, \ \forall \mathbf{v} \in \mathbf{V}, \tag{1.9}$$

where

$$a_\Omega(\mathbf{u}, \mathbf{v}) = \int_\Omega \sum_i \sum_j \sum_k \sum_l \frac{\partial v_i}{\partial x_j} \beta_{ijkl}(x) \frac{\partial u_k}{\partial x_l},$$
$$f(\mathbf{v}) = \int_\Omega \sum_i f_i v_i + \int_{\Gamma_1} \sum_i g_i v_i,$$

and the space $\mathbf{V} \subset (H^1(\Omega))^q$ is given by

$$\mathbf{V} = \{\mathbf{v} \in (H^1(\Omega))^q : \mathbf{v}|_{\Gamma_0} = 0\}.$$

To apply the Lax-Milgram Lemma in this case, we require that

$$\sum_{i,j} \sum_{k,l} y_i z_j \beta_{ijkl}(x) y_k z_l \geq c \sum_{i,j} (y_i z_j)^2, \qquad \forall x \in \Omega, \ \forall y \in R^n, \ \forall z \in R^q.$$

Again we get strong ellipticity (coercivity) by

$$\begin{aligned} a_\Omega(\mathbf{u}, \mathbf{u}) &= \int_\Omega \sum_i \sum_j \sum_k \sum_l \frac{\partial u_i}{\partial x_j} \beta_{ijkl}(x) \frac{\partial u_k}{\partial x_l} \\ &\geq c \int_\Omega \sum_{i,j} (\frac{\partial u_i}{\partial x_j})^2 \\ &= c|\mathbf{u}|^2_{H^1(\Omega)} \\ &\geq c||\mathbf{u}||^2_{H^1(\Omega)}. \end{aligned}$$

We conclude by using Poincaré-Friedrichs' inequality, and the fact that $\mathbf{u}$ is zero on Γ_0.

1.5 Elasticity

The equations of linear elasticity model the displacement $\mathbf{u}(x)$ of a body Ω which is fixed along a portion of its boundary, Γ_0, and is subject to a surface force of density $\mathbf{g}$ along the other portion of the boundary, $\Gamma_1 = \partial\Omega \setminus \Gamma_0$. The body is also subject to a volume force in its interior (such as gravity) given by $\mathbf{f}$, see Figure 1.1.

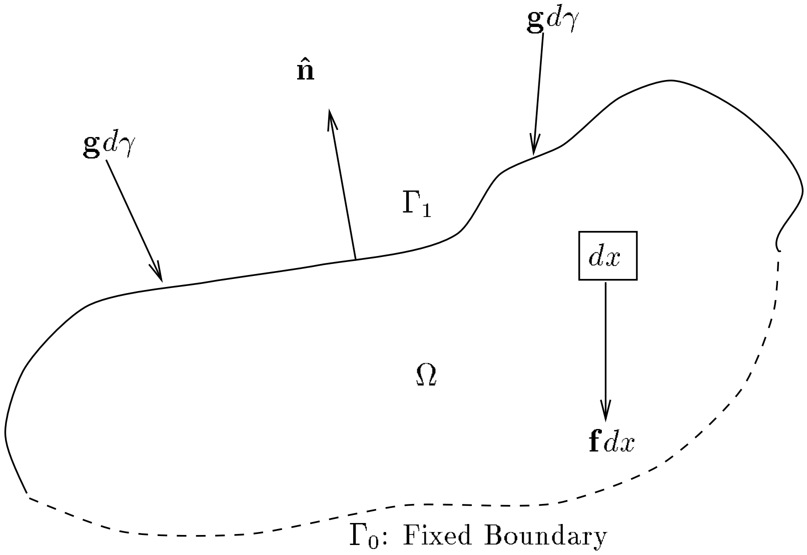


Figure 1.1: A Sample Domain

The equations of linear elasticity [25],[44],[45], are a coupled elliptic system, and can be written in the form

$$\sigma_{ij}(\mathbf{u}) = \sum_{k,l=1}^{3} \alpha_{ijkl}\epsilon_{kl}(\mathbf{u}),$$

$$\sum_{j=1}^{3} \frac{\partial \sigma_{ij}}{\partial x_j}(\mathbf{u}) = f_i, \quad \text{in } \Omega,$$

$$\mathbf{u}|_{\Gamma_0} = 0, \quad \text{and} \quad \sum_{j=1}^{3} \sigma_{ij}(\mathbf{u})\hat{n}_j = g_i \quad \text{on } \Gamma_1 = \partial\Omega \setminus \Gamma_0.$$

The symmetric three-by-three *stress* matrix, $\epsilon_{ij}(\mathbf{u})$, is given by

$$\epsilon_{ij}(\mathbf{u}) = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}).$$

The tensor α_{ijkl} is a symmetric, positive definite operator which operates on symmetric three-by-three matrices. σ_{ij} is called the *strain.* $\hat{\mathbf{n}}$ is the outward normal on Γ_1.

From a Green's formula, we obtain the variational formulation

$$a_\Omega(\mathbf{u}, \mathbf{v}) = f(\mathbf{v}), \quad \mathbf{u} \in \mathbf{V}, \ \forall \mathbf{v} \in \mathbf{V},$$

with

$$a_\Omega(\mathbf{u}, \mathbf{v}) = \int_\Omega \sum_{i,j=1}^{3} \sigma_{ij}(\mathbf{u})\epsilon_{ij}(\mathbf{v}), \quad f(\mathbf{v}) = \int_\Omega \sum_{i=1}^{3} f_i v_i + \int_{\Gamma_1} \sum_{i=1}^{3} g_i v_i,$$

and

$$\mathbf{V} = \{\mathbf{v} \in (H^1(\Omega))^3 \ : \ \mathbf{v}|_{\Gamma_0} = 0\}.$$

A special case of interest is when the material properties are independent of position, i.e. the material is *homogeneous*, and the behavior of the material is the same in all directions, i.e. the material is *isotropic.* In this the case the equations for the strain reduce to

$$\sigma_{ij}(\mathbf{u}) = \lambda(\sum_{k=1}^{3} \epsilon_{kk}(\mathbf{u}))\delta_{ij} + 2\mu\epsilon_{ij}(\mathbf{u}).$$

Here $\lambda \geq 0$ and $\mu > 0$ are the Lamé coefficients, which represent physical properties of the material. Using familiar notations, we find that

$$\sigma_{ij}(\mathbf{u}) = \lambda(\sum_{k=1}^{3} \frac{\partial u_k}{\partial x_k})\delta_{ij} + \mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}),$$

$$\sum_{j=1}^{3} \frac{\partial \sigma_{ij}}{\partial x_j}(\mathbf{u}) = \sum_{j=1}^{3}(\lambda(\sum_{k=1}^{3} \frac{\partial u_k}{\partial x_k})\delta_{ij} + \mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})) = f_i,$$

or

$$\lambda \sum_{k=1}^{3} \frac{\partial^2 u_k}{\partial x_i \partial x_k} + \mu \sum_{j=1}^{3} \frac{\partial^2 u_i}{\partial x_j^2} + \frac{\partial^2 u_j}{\partial x_i \partial x_j} = f_i.$$

These equations can be written as

$$\sum_{j=1}^{3} \mu \frac{\partial^2 u_i}{\partial x_j^2} + (\lambda + \mu)\frac{\partial^2 u_j}{\partial x_i \partial x_j} = f_i, \tag{1.10}$$

or

$$\mu \triangle \mathbf{u} + (\lambda + \mu)\nabla\nabla \cdot \mathbf{u} = \mathbf{f}. \tag{1.11}$$

To put the equation in variational form, we multiply equation (1.10) by v_i, integrate over Ω, sum over i and integrate by parts. We then obtain

$$\int_\Omega \mu \sum_{i=1}^{3}\sum_{j=1}^{3} \frac{\partial v_i}{\partial x_j}\frac{\partial u_i}{\partial x_j} + (\mu + \lambda)(\sum_{i=1}^{3} \frac{\partial v_i}{\partial x_i})(\sum_{j=1}^{3} \frac{\partial u_j}{\partial x_j}) = \int_\Omega \sum_{i=1}^{3} v_i f_i + \int_{\Gamma_1} \sum_{i=1}^{3} g_i v_i.$$

Again we express this in terse notation as,

$$a_\Omega(\mathbf{u}, \mathbf{v}) = f(\mathbf{v}), \quad \mathbf{u} \in \mathbf{V}, \ \forall \mathbf{v} \in \mathbf{V}. \tag{1.12}$$

An equivalent formula for $a_\Omega(\mathbf{u}, \mathbf{v})$ using the stress matrix $\epsilon_{ij}(\mathbf{u})$ is given by

$$a_\Omega(\mathbf{u}, \mathbf{v}) = \int_\Omega 2\mu \sum_{i=1}^{3}\sum_{j=1}^{3} \epsilon_{ij}(\mathbf{u})\epsilon_{ij}(\mathbf{v}) + \lambda(\sum_{i=1}^{3} \frac{\partial v_i}{\partial x_i})(\sum_{j=1}^{3} \frac{\partial u_j}{\partial x_j}). \tag{1.13}$$

The proof that $a_\Omega(\mathbf{u},\mathbf{v})$, as given in equation (1.13), is strongly elliptic (coercive) in the $(H^1(\Omega))^3$ norm is non-trivial. This is because we need to bound all combinations of $\frac{\partial u_i}{\partial x_j}$ by combinations of the form $\frac{1}{2}(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i})$. The following inequality, due to Korn (1909), establishes strong ellipticity (coercivity) and hence existence and uniqueness for the variational problem (1.12).

Lemma 1.5.1 (**Korn's inequality** [26],[38],[70]) *If $\partial\Omega$ is Lipschitz, cf. [70], then*

$$||\mathbf{u}||^2_{H^1(\Omega)} \le c(\Omega)(\int_\Omega \sum_{i,j} \epsilon^2_{ij}(\mathbf{u}) + ||\mathbf{u}||^2_{L^2}).$$

A consequence of Korn's inequality is, cf. [25],

Theorem 1.5.1 *If the measure of $\Gamma_0 \subset \partial\Omega$ is greater than zero and*

$$\mathbf{V} = \{\mathbf{v} \in (H^1(\Omega))^q : \mathbf{v}|_{\Gamma_0} = 0\},$$

then

$$c||\mathbf{u}||^2_{H^1(\Omega)} \le \int_\Omega \sum_{i,j} \epsilon^2_{ij}(\mathbf{u}) \le C||\mathbf{u}||^2_{H^1(\Omega)}, \quad \forall \mathbf{u} \in \mathbf{V}.$$

The strong ellipticity (coercivity) of the bilinear form follows from,

$$\begin{aligned} a_\Omega(\mathbf{u},\mathbf{u}) &= \int_\Omega 2\mu \sum_{i=1}^3\sum_{j=1}^3 \epsilon^2_{ij}(\mathbf{u}) + \lambda(\sum_{i=1}^3 \frac{\partial u_i}{\partial x_i})^2 \\ &\ge \int_\Omega 2\mu \sum_{i=1}^3\sum_{j=1}^3 \epsilon^2_{ij}(\mathbf{u}) \\ &\ge 2\mu c||\mathbf{u}||^2_{H^1(\Omega)}. \end{aligned}$$

Instead of the Lamé coefficients, we can work with Young's modulus, E, and Poisson's ratio, ν, defined implicitly by,

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{1+\nu}.$$

Using these parameters, the differential equation (1.11) becomes

$$\frac{E}{1+\nu}(\triangle\mathbf{u} + \frac{1-\nu}{1-2\nu}\nabla\nabla\cdot\mathbf{u}) = \mathbf{f}. \qquad (1.14)$$

Note that the second coefficient blows up as ν approaches 1/2.

1.6 Finite Elements for Elasticity

1.6.1 Formulation

The finite element formulation is obtained by replacing the infinite dimensional space $\mathbf{V}$ with a finite dimensional subspace $\mathbf{V^h} \subset \mathbf{V}$. We triangulate the domain Ω into non-overlapping regions called elements, generally triangles, rectangles, tetrahedra, or bricks. $\mathbf{V^h}$ is then defined to be the space of continous, piecewise linear, bilinear, trilinear, quadratic, etc. functions on the elements.

For $\mathbf{V^h}$ we need to define a convenient basis $\{\phi^j\}$. We use a *nodal* basis. In the simplest case the ϕ^j are vector valued functions, which are one in a single component at a single vertex of an element and zero in all the others, and piecewise linear, bilinear, etc. inside each of the elements.

The finite dimensional variational problem is obtained by representing an approximation to $\mathbf{u}$ in the basis ϕ^j,

$$\mathbf{u}^h = \sum_j y_j \phi^j,$$

and inserting this into the variational problem

$$a_\Omega(\sum_j y_j \phi^j, \phi^k) = f(\phi^k), \qquad \forall \phi^k.$$

We define the *stiffness matrix*

$$K_{jk} = a_\Omega(\phi^j, \phi^k),$$

and the *load*

$$b_k = f(\phi^k),$$

thus producing a linear system

$$Ky = b.$$

The solution $\mathbf{u}^h = \sum_j y_j \phi^j$ is called the Galerkin approximation and is the a_Ω-orthogonal projection of the true solution $\mathbf{u}$ onto the subspace $\mathbf{V^h}$. For a fuller discussion of the finite element method and its analysis, cf. Ciarlet [26] or Strang and Fix [81].

1.6.2 Types of Elements

We now briefly discuss several particular elements, which often arise in engineering practice. For a more detailed discussion of the elements and the exact forms of the arising stiffness matrices, see Przemieniecki [74].

The first of these consists of triangular elements with piecewise linear functions. The nodal basis functions for this space are one, or a unit vector, at a single vertex node, zero at all the others and linear on each element. For the Laplacian, when the elements are isosceles right triangles, the resulting stiffness matrix has exactly the same form as that arising from using finite differences with the five point stencil. This is a well known model problem.

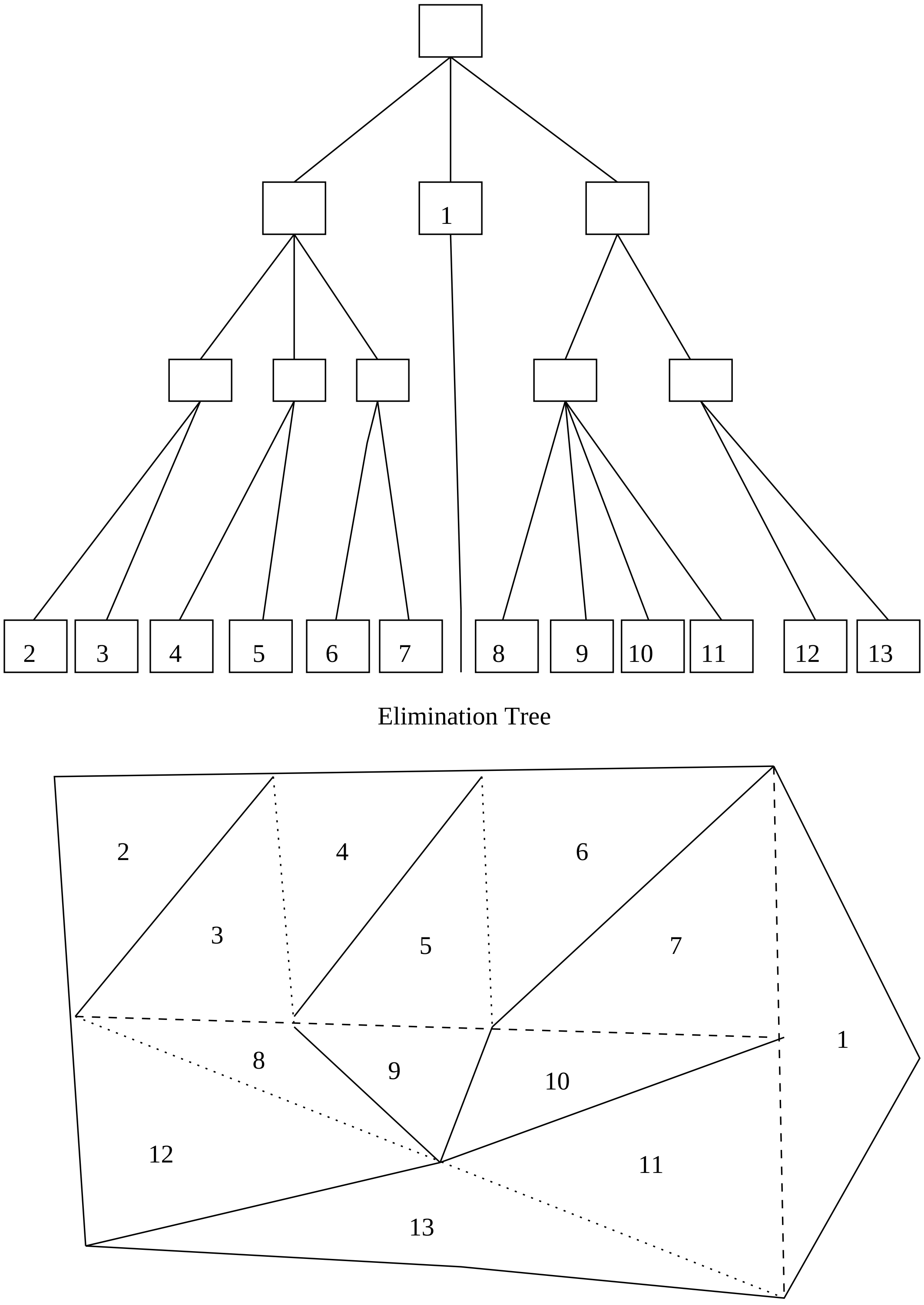
The membrane plane quadrilateral element is constructed for in-plane displacements at six nodes. The nodes are at the four quadrilateral vertices and two interior points. The basis functions are quadratic in the displacements x and y. The two interior nodes are eliminated by Choleski factorization before the element is combined with its neighbors; this elimination is referred to as static condensation. Hence it has eight degrees of freedom, two for each vertex node. This element has been used in some of the experiments reported in this thesis, see [4].

The flat, thin shell quadrilateral element is based on Kirchoff's plate bending theory, see [4],[74]. In general such an element has six degrees of freedom for each node, three displacements and three rotations. Again, the basis functions are quadratic and the two interior nodes are eliminated by static condensation. This element is generally ill-conditioned and its use increases the difficulties of using iterative solvers on the resulting linear system.

1.6.3 Substructuring Methods

Substructuring provides an efficient direct factorization method for the solution of the resulting linear system. It is based upon the fact that if we divide the domain Ω into substructures, which are larger than the elements, not only can we form the stiffness matrix for different substructures independently, but we can also eliminate the interior variables of the substructures independently. If we define

$$K_{jk}^{(i)} = a_{\Omega_i}(\phi^j, \phi^k)$$



Elimination Tree

Figure 1.2: Pictorial Representation of Substructuring

and

$$b_k^{(i)} = f_{\Omega_i}(\phi^k),$$

then, by properly merging the columns and rows of the $K^{(i)}$ and $b^{(i)}$, we can form K and the load b. Similarly the subvector of y associated with substructure Ω_i is $y^{(i)}$. **Symbolically**

$$K = \sum_i K^{(i)}, \qquad \text{and} \qquad b = \sum_i b^{(i)}.$$

This is the *subassembly* process.

We order the nodes on the interior of the substructures first and write this as

$$y^{(i)} = \begin{pmatrix} y_I^{(i)} \\ y_B^{(i)} \end{pmatrix}, \qquad b^{(i)} = \begin{pmatrix} b_I^{(i)} \\ b_B^{(i)} \end{pmatrix},$$

$$K^{(i)} = \begin{pmatrix} K_{II}^{(i)} & K_{IB}^{(i)} \\ K_{IB}^{(i)^T} & K_{BB}^{(i)} \end{pmatrix}.$$

Symbolically the linear system is,

$$\sum_i \begin{pmatrix} K_{II}^{(i)} & K_{IB}^{(i)} \\ K_{IB}^{(i)^T} & K_{BB}^{(i)} \end{pmatrix} \begin{pmatrix} y_I^{(i)} \\ y_B^{(i)} \end{pmatrix} = \sum_i \begin{pmatrix} b_I^{(i)} \\ b_B^{(i)} \end{pmatrix}.$$

We eliminate the interior nodes,

$$\sum_i \begin{pmatrix} K_{II}^{(i)} & K_{IB}^{(i)} \\ 0 & S_{BB}^{(i)} \end{pmatrix} \begin{pmatrix} y_I^{(i)} \\ y_B^{(i)} \end{pmatrix} = \sum_i \begin{pmatrix} b_I^{(i)} \\ b_B^{(i)} - K_{IB}^{(i)^T} K_{II}^{(i)^{-1}} b_B^{(i)} \end{pmatrix}.$$

$$S_{BB}^{(i)} = K_{BB}^{(i)} - K_{IB}^{(i)^T} K_{II}^{(i)^{-1}} K_{IB}^{(i)},$$

is a *Schur Complement* for substructure Ω_i and is independent of the other substructures.

The reduced linear system is now

$$\sum_i S_{BB}^{(i)} y_B^{(i)} = \sum_i \tilde{b}_B^{(i)}.$$

This method can be applied recursively, combining several substructures and again eliminating those nodes which lie in the new interior. We represent this process in Figure 1.2. For very large problems there may be 8 to 10 levels of this recursion. The substructures are sometimes referred to as *superelements* since once their interior nodes have been eliminated, they play the role of elements.

A great advantage of this method is that if several substructures are identical then they need to be formed and factored only once. A commercial code which uses this method is available from Veritas Sesam Systems, Oslo, Norway. This code has recently been parallelised for general multi-processor machines [47]. The parallelisation is done at two levels. Problems on separate subdomains are factored on individual processors and also larger factorizations are spread over several processors. For a good discussion of this approach, see Hvidsten [47].

On the lower levels the problems are small and sparse and can be factored quickly. As one moves up the tree the problems become larger and denser. The ease of parallelisation is also much higher at the lower levels. In practice most of the cpu time is spent at the higher levels of the tree. This suggests that we should stop the factorization at some point and solve the reduced system using a *preconditioned conjugate gradient* method. The development and analysis of such methods is a major goal of the thesis.

Chapter 2

Domain Decomposition Techniques

2.1 Iterative Methods

The iterative method of choice for many domain decomposition algorithms is the preconditioned conjugate gradient method (PCG). This is a well known algorithm so we will not discuss it in detail, see cf. [27],[41]. For completeness, we give one version below.

The conjugate gradient method is often an effective iterative algorithm to solve the symmetric, positive definite system

$$Ax = b.$$

For a well conditioned matrix A, the conjugate gradient algorithm may converge to a good approximate solution in relatively few iterations. When A is not well conditioned, which is generally the case for discretizations of elliptic problems, we can introduce a preconditioner M and solve the linear system

$$M^{-1}Ax = M^{-1}b.$$

The preconditioner M is chosen so that the linear system $My = Ax$ is easily solved and also so that $M^{-1}A$ is well conditioned. This latter requirement insures that the conjugate gradient algorithm will converge in a small number of iterations.

We note that in the conjugate gradient algorithm given below we need not explicitly form the matrices A and M^{-1}. We only need to know how to apply them to a given vector.

Preconditioned Conjugate Gradient Algorithm [73]

```
Choose $x^0$.
$r^0 = b - Ax^0$.
Solve $M\tilde{r}^0 = r^0$.
$p^0 = \tilde{r}^0$.
For $k = 0, 1, \ldots$
    $\alpha_k = -(\tilde{r}^k, r^k)/(p^k, Ap^k)$
    $x^{k+1} = x^k - \alpha_k p^k$
    $r^{k+1} = r^k + \alpha_k Ap^k$
    Test for convergence
    Solve $M\tilde{r}^{k+1} = r^{k+1}$
    $\beta_k = (\tilde{r}^{k+1}, r^{k+1})/(\tilde{r}^k, r^k)$
    $p^{k+1} = \tilde{r}^{k+1} + \beta_k p^k$
end $k$
```

The reduction in the energy norm of the error after n steps of PCG can be bounded by, see e.g. [41],

$$\frac{2}{\rho^n + \rho^{-n}}$$

where

$$\rho = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}$$

and

$$\kappa = \frac{\lambda_{\max}(M^{-1}A)}{\lambda_{\min}(M^{-1}A)}).$$

The PCG method works well for problems with small *condition number*, κ, so we should chose the preconditioner M so that $M^{-1}A$ is well conditioned and the operation $M^{-1}x$ is inexpensive.

2.2 Multiplicative Schemes

2.2.1 The Classical Alternating Schwarz Scheme

The earliest domain decomposition method was suggested in an existence proof by Schwarz in 1869, [75]. Let us consider the domain as shown in Figure 2.1 with

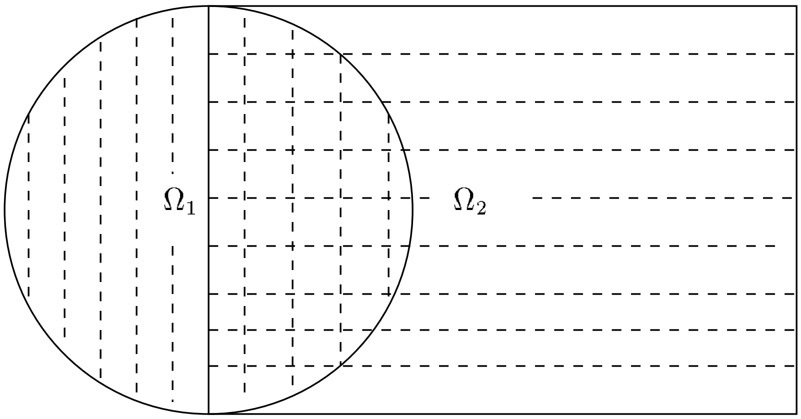


Figure 2.1: Subdomains for Schwarz's Method

$\Omega = \Omega_1 \cup \Omega_2$ on which we wish to solve

$$\begin{aligned} -\triangle u &= f && \text{in } \Omega, \\ u &= 0 && \text{on } \partial\Omega. \end{aligned}$$

The Schwarz alternating procedure approximates the solution iteratively by solving problems on the individual subdomains.

$$\begin{aligned} -\triangle u^{n+1/2} &= f && \text{in } \Omega_1, \\ u^{n+1/2} &= u^n && \text{on } \partial\Omega_1. \end{aligned}$$

and

$$\begin{aligned} -\triangle u^{n+1} &= f && \text{in } \Omega_2, \\ u^{n+1} &= u^{n+1/2} && \text{on } \partial\Omega_2. \end{aligned}$$

Schwarz established that this method converged using the Maximum Principle. In the 1930's Sobolev extended the result to the partial differential equations of linear elasticity, [79]. Recent work by Lions [58] has interpreted the algorithm in a variational framework. If we let

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \qquad \text{and} \qquad f(v) = \int_\Omega fv$$

then the algorithm can be written as

$$a(u^{n+1/2} - u^n, v) = f(v) - a(u^n, v), \qquad u^{n+1/2} - u^n \in H_0^1(\Omega_1), \forall v \in H_0^1(\Omega_1),$$

and

$$a(u^{n+1} - u^{n+1/2}, v) = f(v) - a(u^{n+1/2}, v), \qquad u^{n+1} - u^{n+1/2} \in H_0^1(\Omega_2), \forall v \in H_0^1(\Omega_2).$$

At each half-step the correction is thus the projection of the error onto the subspace $H_0^1(\Omega_1)$ or $H_0^1(\Omega_2)$. The projection P_i is given by

$$a(P_i u, v) = a(u, v), \qquad\qquad P_i u \in H_0^1(\Omega_i), \forall v \in H_0^1(\Omega_i).$$

The error propagation operator for one complete step of the alternating Schwarz method is simply

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

For numerical computation we would always work with some finite element subspaces $V^h \subset H_0^1(\Omega)$ and $V^h \cap H_0^1(\Omega_i)$.

2.2.2 The Abstract Multiplicative Schwarz Scheme

The alternating Schwarz method is a particular case of an abstract multiplicative Schwarz scheme. We consider the finite dimensional variational problem

$$a(u, v) = f(v), \qquad \forall\, v \in V, \quad u \in V. \tag{2.1}$$

Let V_i be subspaces of V so that $V = V_1 + \cdots + V_N$. Associated with each subspace is a corresponding orthogonal projection operator P_i, onto the subspace V_i, such that

$$a(P_i u, w) = a(u, w) = f(w), \qquad \forall\, w \in V_i, \quad P_i u \in V_i. \tag{2.2}$$

$P_i u$ can be determined by introducing a basis $\{\psi_j^{(i)}\}$ for V_i and expanding $P_i u$ in that basis, $P_i u = \sum_j \alpha_j^{(i)} \psi_j^{(i)}$. This results in the linear system

$$\tilde{K}^{(i)} \alpha^{(i)} = f^{(i)},$$

where $\tilde{K}_{jl}^{(i)} = a(\psi_j^{(i)}, \psi_l^{(i)})$, and $f^{(i)}$ is the vector defined by $f(\psi_j^{(i)})$. The abstract multiplicative Schwarz scheme proceeds by serially projecting the error onto the subspaces V_i. The error propagation operator for one complete step of the alternating Schwarz method is given by

$$\Pi_i (I - P_i).$$

New theoretical tools for the analysis of multiplicative Schwarz schemes have just recently been developed. Recent work on the two subspace case for the continuous problem has been done by Lions [58]; see also Nepomnyaschikh [68]. For the finite element formulation the two subspace case has been discussed in Bjørstad [5], Mandel and McCormick [64] and Bjørstad and Mandel [7]. More general theories for the case of more then two subspaces have been introduced by Widlund [85] and Mathew [65]. The most successful results so far have been obtained very recently by Bramble, Pasciak, Wang, and Xu [16]. We review their abstract theory which can be used in analyzing multiplicative variants of the algorithms presented in this thesis.

Assume we have J symmetric (in the $a(\cdot,\cdot)$ inner product), positive semi-definite linear operators T_i of norm bounded by $\omega < 2$. Define E_i by

$$E_i = (I - T_i)(I - T_{i-1}) \cdots (I - T_1).$$

The T_i are either projections onto the subspaces V_i or approximations thereof. The effectiveness of the particular multiplicative algorithm is determined by the reduction in the error for one complete iteration of the scheme, i.e.,

$$|E_J v|_a \leq \gamma |v|_a.$$

Explicit bounds on γ are now available. We demonstrate below how those bounds can be obtained.

We first note two useful identities

$$E_{i-1} - E_i = T_i E_{i-1} \tag{2.3}$$

and

$$I - E_i = \sum_{j=1}^{i} T_j E_{j-1}. \tag{2.4}$$

Using equation (2.3) Bramble, Pasciak, Wang, and Xu [16] establish the inequality,

Lemma 2.2.1 *With the above definitions of E_i and T_i,*

$$|E_J v|_a^2 \leq |v|_a^2 - (2 - \omega) \sum_{i=1}^{J} a(T_i E_{i-1} v, E_{i-1}).$$

We next need to make two assumptions on the subspaces V_i and the operators T_i. Assume there exists a C_0^2 such that

$$a(v,v) \leq C_0^2 \sum_{i=1}^{J} a(T_i v, v), \qquad \forall v \in V. \tag{2.5}$$

The second assumption involves a strengthened Cauchy-Schwarz inequality, i.e. let $\mathcal{E}$ be the matrix such that

$$|a(v_i, v_j)| \leq \epsilon_{ij} a(v_i, v_i)^{1/2} a(v_j, v_j)^{1/2}, \qquad \forall v_i \in V_i, \forall v_j \in V_j. \tag{2.6}$$

The relevant quantity in equation (2.6) is $\rho(\mathcal{E})$, the spectral radius of $\mathcal{E}$. We should note that as the overlap between the spaces increases the bound in equation (2.5) will generally improve while the bound in (2.6) will degrade, i.e. the spectral radius $\rho(\mathcal{E})$ will increase.

A main result of Bramble, Pasciak, Wang, and Xu is given in

Theorem 2.2.1 *Assume equation (2.5), holds then*

$$|E_J v|_a \leq \gamma |v|_a,$$

with

$$\gamma = 1 - \max\left(\frac{2-\omega}{C_0^2(J + \omega^2 J(J-1)/2)}, \frac{2-\omega}{2C_0^2(1 + \omega^2 J(J-1)/2)}\right).$$

Proof. This is Theorem 2.1 in Bramble, Pasciak, Wang, and Xu [16]. The proof makes use of Lemma 2.2.1 and the identity (2.4). ∎

When many of the subspaces do not intersect Bramble, Pasciak, Wang, and Xu [16] have strengthened their result to take this into account. Recently, Xu [86] has obtained an even stronger result.

Theorem 2.2.2 *Assume (2.5) and (2.6) hold. Then*

$$|E_J v|_a \leq \gamma |v|_a,$$

with

$$\gamma = 1 - \frac{2-\omega}{2C_0^2(1 + \omega^2 \rho(\mathcal{E})^2)}.$$

An important special case of this general result is when the T_i are projections P_i onto the subspaces V_i. For this result, we can replace assumption (2.5) with the assumption: for all $v \in V$ there exists a decomposition $v = \sum v_i$ with $v_i \in V_i$ such that

$$\sum_{i=1}^{J} a(v_i, v_i) \leq C_0^2 a(v, v).$$

For the proof of this result see Lemma 2.3.3 in the following section. In addition, we note, since the P_i are orthogonal projections, ω is one.

2.3 Additive Schwarz Schemes

One of the possible problems with the multiplicative Schwarz methods is the serial nature of the fractional steps of each iteration. The additive Schwarz schemes were designed by Dryja and Widlund [35] to remove the inherent sequential behavior of the fractional steps.

Again we consider the finite dimensional variational problem

$$a(u, v) = f(v), \qquad \forall v \in V, \quad u \in V. \tag{2.7}$$

As before, let V_i be subspaces of V such that $V = V_1 + \cdots + V_N$. Associated with each subspace is a corresponding projection operator P_i, which is the orthogonal projection in the $a(u, v)$ inner product onto the subspace V_i.

The additive Schwarz method, see Dryja and Widlund [36], of solving equation (2.7) is introduced in terms of an auxiliary problem

$$Pu = \sum_i P_i u = \hat{f}, \tag{2.8}$$

which has the same solution as equation (2.7). Since $P_i u$ can be found by equation (2.2) without knowing the solution of (2.7), we first compute $\hat{f}$ and then solve equation (2.8) using the conjugate gradient method, without further preconditioning.

The reason for going from problem (2.7) to problem (2.8) is that, by a suitable choice of the subspaces V_i, we can turn a large ill-conditioned system into a very well conditioned problem at the expense of solving many small independent linear systems. The following three lemmas allow us to develop bounds on the condition number of P.

Lemma 2.3.1 *Consider the undirected graph with a node for each subspace V_i, and an edge between node i and node j iff $V_i \cap V_j \neq 0$. Let p be the number of colors needed to color the graph so that no two nodes connected by an edge have the same color. Then*

$$\lambda_{\max}(P) \leq p.$$

Proof. All the subspaces for a particular color are disjoint, hence their corresponding projection operators are mutually orthogonal. Therefore the sum of the projection operators of a particular color is itself a projection operator. P then is the sum of p projection operators each of norm one. ∎

Working with a strengthened Cauchy-Schwarz inequality, i.e. as in the multiplicative case, we obtain the stronger result.

Lemma 2.3.2 *Assume equation (2.6) holds then*

$$\lambda_{\max}(P) \leq \rho(\mathcal{E}).$$

The following lemma allows us to obtain bounds on the smallest eigenvalue of P. This result is given in Nepomnyaschikh [68] and for the two subspace case in Lions [58], cf. also Dryja and Widlund [35].

Lemma 2.3.3 *Assume that for all $u \in V$, there exists a representation $u = \sum_i u_i$ with $u_i \in V_i$ such that*

$$\sum_i a(u_i, u_i) \leq C_0^2 a(u, u) \tag{2.9}$$

then

$$\lambda_{\min}(P) \geq C_0^{-2}.$$

Moreover, if C_0^2 is the best possible constant for equation (2.9), then $\lambda_{\min}(P) = 1/C_0^2$.

Proof.

$$|u|_a^2 = \sum_i a(u, u_i) = \sum_i a(u, P_i u_i) = \sum_i a(P_i u, u_i).$$

Therefore,

$$|u|_a^2 \leq (\sum_i |P_i u|_a^2)^{1/2} (\sum_i |u_i|_a^2)^{1/2}.$$

By the assumption of the lemma and a property of projections,

$$|u|_a^2 \leq C_0^2 \sum_i |P_i u|_a^2 = C_0^2 \sum_i a(P_i u, u) = C_0^2 a(Pu, u).$$

To show that the bound is strict, we construct explicit decompositions of u. Let u^k be the eigenvectors of P with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_{k-1} \leq \lambda_k \leq \cdots \leq \lambda_n$. Define a decomposition of u^k by

$$u_i^k = \frac{1}{\lambda_k} P_i u^k.$$

A simple calculation shows that

$$\sum_i a(u_i^k, u_i^k) = \frac{1}{\lambda_k} a(u^k, u^k).$$

In fact this is the best decomposition of u^k. To see this we assume that $u^k = \sum_i w_i^k$ is a better decomposition, i.e.,

$$\sum_i a(w_i^k, w_i^k) < \frac{1}{\lambda_k} a(u^k, u^k).$$

We then conclude that

$$a(u^k, u^k) < \frac{1}{\lambda_k} a(Pu^k, u^k) = a(u^k, u^k),$$

since u^k is an eigenvector.

We conclude the proof by explicitly constructing for all $u \neq u^1$, a decomposition so that

$$\sum_i a(u_i, u_i) < \frac{1}{\lambda_1} a(u, u).$$

Let $u \neq u^1$ be of norm one and expand it in the eigenvectors of P,

$$u = \sum_k \alpha_k u^k \qquad \text{with} \qquad \sum_k \alpha_k^2 = 1.$$

Define u_i by

$$u_i = P_i \sum_k \frac{\alpha_k}{\lambda_k} u^k.$$

Then using a property of projections and the orthogonality of the eigenvectors, we obtain,

$$\begin{aligned} \sum_i a(u_i, u_i) &= \sum_k \frac{\alpha_k^2}{\lambda_k} a(u, u) \\ &< \frac{1}{\lambda_1} a(u, u). \end{aligned}$$

∎

It is easy to construct and analyze preconditioners which use approximate solvers using additive Schwarz techniques. The preconditioned matrix P can be written as

$$P = \sum_i B_i^+ A.$$

The B_i^+ represents a pseudo-inverse since the individual terms in the sum are singular. We introduce new operators C_i^+ which are spectrally 'near' B_i^+, and define $\hat{P}$ by

$$\hat{P} = \sum_i \hat{P}_i = \sum_i \alpha_i C_i^+ A.$$

The α_i offer an additional chance to improve the condition number by adjusting the scaling between the various $\hat{P}_i$.

We can now give bounds on the condition number of $\hat{P}$. If

$$m_i x^T B_i^+ x \leq \alpha_i x^T C_i^+ x \leq M_i x^T B_i^+ x,$$

then

$$\lambda_{\max}(\hat{P}) \leq \max_i M_i \, \lambda_{\max}(P),$$

and

$$\lambda_{\min}(\hat{P}) \geq \min_i m_i \, \lambda_{\min}(P).$$

We wish to examine the structure of the B_i^+ a little more carefully. Let S_i be a matrix whose columns span the space V_i. Then B_i^+ can be written as

$$B_i^+ = S_i^T (S_i A S_i^T)^{-1} S_i.$$

For most of the approximate solver techniques, C_i^+ is of the form

$$C_i^+ = S_i^T (G_i)^{-1} S_i.$$

The G_i are chosen to be spectrally equivalent to the matrices $S_i A S_i^T$. This then insures the spectral equivalence of B_i^+ and C_i^+.

Remark: The case when the C_i^+ are chosen in another manner is not well understood. The problem is that the null spaces of the two operators no longer match. This then changes in a nontrivial manner the fundamental subspaces of the algorithm.

Remark: Nothing general can yet be said about the relative utility of the additive and multiplicative versions of the same scheme in a parallel environment. The same two parameters $\rho(\mathcal{E})$ and C_0^2 determine the convergence behavior of both variants; however, there is no general theory which explicitly relates the actual convergence rates of the two variants, except in the case of two subspaces where it is well understood, cf., [5],[7], and [64]. The architectures of the various parallel machines available seem more likely to determine which variant to prefer rather than a strict mathematical analysis would. We note that the multiplicative variant would, in general, be non-symmetric, therefore to accelerate it with conjugate gradient we must introduce additional fractional steps to make it symmetric. We could instead accelerate the nonsymmetric problem directly using GMRES.

2.4 The Need for a Global Problem

For a domain decomposition algorithm with a small number of subdomains good convergence may be obtained even with only local communication, e.g. at each iteration of the algorithm information is passed only between neighboring subdomains. If a large number of subdomains are used then the convergence rate deteriorates rapidly with the number of subdomains unless an additional feature is added to the algorithm to provide for global communication of information in each iteration. Assume that the diameter of the domain is of order one and the representative diameter of the subdomains is of order H. In [84] Widlund shows that without a global coarse problem the condition number of the iteration operator grows like $1/H^2$.

For iterative substructuring algorithms two types of global coarse problems have been proposed in the literature. We shall refer to the resulting algorithms as the vertex based methods and the wirebasket based methods. The wirebasket refers to all nodes which belong to the closure of more than two substructures. The vertex based algorithm introduces a coarse problem which is basically the discretization of the Laplacian (or the operator in question) on the coarse grid defined by the subdomains. This works well in two dimensions, but without additional enhancements it performs poorly in three dimensions, see Section 3.2. This has provided the impetus for the development of the second type of coarse problem. The wirebasket based approach involves first calculating an average value for each subdomain by solving a problem which involves a relatively small number of unknowns from each subdomain, the unknowns on the wirebasket, and then solving local problems. Such methods can have good asymptotic behavior in three dimensions. Both of these approaches are discussed in detail in this thesis and we believe important new insights are offered.

For additive Schwarz type algorithms a simple vertex based coarse problem works well in both two and three dimensions. The more generous overlap in Schwarz methods negates the need for a more elaborate coarse problem. This is because the extra overlap allows more freedom in the selection of the partition of u into the subspaces as required for Lemma 2.3.3. A disadvantage of the extra overlap is that the condition number may not be independent of the jumps in the coefficients of the differential equation between subdomains, while it is for the iterative substructuring algorithms. This is discussed in the next section.

2.5 Global Bounds from Local Bounds

For many domain decomposition algorithms it is possible to bound the condition number of the preconditioned problem by bounds obtained locally, i.e. by bounds obtained on individual substructures. This important observation is used in Bramble, Pasciak, and Schatz [12],[14] and presented very clearly in Mandel [59]. Iterative substructuring algorithms share this property. Additive Schwarz algorithms with overlapping regions do not have this property.

Let $K^{(i)}$ be the contribution of substructure Ω_i to the stiffness matrix K and $\hat{K}^{(i)}$ be the contribution of substructure Ω_i to the preconditioner $\hat{K}$. For instance, in the appropriate basis, $\hat{K}^{(i)}$ may be a block diagonal part of $K^{(i)}$. If

$$c_i \hat{K}^{(i)} \leq K^{(i)} \leq C_i \hat{K}^{(i)}, \qquad \forall i,$$

then

$$\min_i c_i \hat{K} \leq K \leq \max_i C_i \hat{K}$$

or equivalently,

$$\kappa(\hat{K}^{-1} K) \leq \frac{\max_i C_i}{\min_i c_i}.$$

We shall refer to the quantity $\frac{\max_i C_i}{\min_i c_i}$ as the local bound. For this technique to work it is necessary that

$$\text{null}(\hat{K}^{(i)}) = \text{null}(K^{(i)}) \qquad \forall i.$$

Whenever this condition is satisfied, we also obtain the important result that if the coefficients of the equation are slowly varying on each subdomain then the condition number of the preconditioned problem is independent of the jumps in the coefficients between substructures.

Several domain decomposition methods have been designed to insure that these null spaces coincide, cf. Bramble, Pasciak, and Schatz [15], Dryja and Widlund [37], and Mandel [59].

We note that for the scalar model problem any substructure which has no part of its boundary with given Dirichlet data has a null space of the constant functions. Those with any part of their boundary having given Dirichlet data have only the trivial null space. For linear elasticity the null space is more complicated, being the set of infinitesimal rigid motions, i.e. translations and rotations. In the plane this is a three dimensional space, in three space it is six dimensional. Boundary substructures may have reduced null spaces depending on how many possible motions are inhibited by the boundary conditions.

Chapter 3

A Hierarchical Domain Decomposition Method

3.1 Introduction

In this chapter, we consider second order, self-adjoint, uniformly elliptic differential equations on a two dimensional polygonal domain Ω. The problems are solved numerically by using continuous, piecewise linear finite elements. The domain is first subdivided into nonoverlapping, triangular substructures Ω_i, and these are further triangulated into elements. H denotes the diameter of a typical substructure and h the diameter of one of its elements.

We develop a domain decomposition algorithm similar to those considered by Bjørstad and Widlund [8],[9]; Bramble, Pasciak, and Schatz [12],[13]; Dryja and Widlund [36]; and Widlund [84]. When using these methods, the variables interior to individual substructures are first eliminated. The resulting reduced system, the Schur complement, therefore only involves the variables associated with Γ, the set of edges and vertices of the substructures. This system is then solved by a preconditioned conjugate gradient method, where the preconditioner is constructed from certain problems associated with the interfaces $\Gamma_{ij} = \partial\Omega_i \bigcap \partial\Omega_j$ between the substructures, and a global coarse problem associated with the vertices of the substructures. We note that it is shown in Dryja and Widlund [36], that such a preconditioner naturally can be viewed in terms of a splitting; cf. Varga [83]. In the splitting, the couplings between the groups of variables, associated with individual edges of the substructures, are eliminated. It is explained in [36] how results and algorithms for the two substructure case can be used to construct and analyze problems on many substructures.

Various preconditioners have been proposed for the subproblems associated with the edges Γ_{ij}. For each Γ_{ij}, this is essentially a two subregion problem, and we can therefore take advantage of a number of results obtained in early work on domain decomposition algorithms. Already in 1980, Dryja [31], see also [32], introduced an effective preconditioner J, which is the square root of a discrete, one dimensional Laplacian on Γ_{ij}. The same preconditioner was also discussed in Bjørstad and Widlund [9] and Bramble, Pasciak, and Schatz [12]. Other preconditioners for these two subregion subproblems, such as the Neumann-Dirichlet algorithm, were considered by Bjørstad and Widlund [9]; Bramble, Pasciak, and Schatz [13]; Chan and Resasco [23]; Chan and Keyes [22]; Dihn, Glowinski, and Périaux [30]; and Golub and Mayers [40]. A number of the resulting algorithms for the many substructure case are known to be almost optimal in the sense that the condition number is bounded by $C(1+\log(H/h))^2$. For a more complete discussion see Bjørstad and Widlund [9] and Widlund [84] for the two subregion and many subregion cases, respectively.

An alternative almost optimal algorithm, which uses a hierarchical basis, has been introduced by Yserentant [88]. His bound on the condition number is of the same form. When the standard finite element nodal basis is replaced by a hierarchical basis, the transformed matrix becomes much better conditioned. The preconditioner is then obtained by discarding the off diagonal blocks and by replacing all but one of the diagonal blocks by diagonal matrices.

In this chapter, we consider a hybrid method demonstrating that a successful and simple preconditioner can be obtained by changing the basis of the spaces associated with individual edges Γ_{ij}. Our proof uses only tools of linear algebra and Yserentant's main result. We show that the new method has a smaller condition number than Yserentant's original method; thus it grows no faster than $C(1+\log(H/h))^2$, a result confirmed in our numerical experiments.

Our work has been inspired by recent work of Babuška, Craig, Mandel, and Pitkäranta [1], Babuška, Griebel, and Pitkäranta [2] and Mandel [60],[61],[62] where efficient preconditioners for the p-version of the finite element method are developed by using hierarchical basis functions and partial orthogonalization of the basis functions. Similarly, our algorithm for the h-version involves a change of basis. It is extremely easy to carry out, and it results in a much better conditioned linear system.

Remark: We should note that recent results on extending hierarchical basis methods to more general multilevel schemes have been obtained by Bramble, Pasciak, and Xu [17]. They have successfully constructed multilevel schemes with performance exceeding that of Yserentant's hierarchical basis method for problems in both two and three dimensions. See Yserentant [89] for a comparison of the two methods. It may be possible to apply some of the techniques of this chapter to the schemes of Bramble, Pasciak, and Xu as well.

Remark: We note that an accelerated version of Yserentant's algorithm has been developed by Bank, Dupont, and Yserentant [3]. It can be viewed as a symmetric multiplicative scheme while the original formulation is an additive scheme. The algorithm proposed in this chapter is an additive scheme. Multiplicative versions can therefore be developed straight forwardly.

We consider a second order, self adjoint, coercive, bilinear form $a_\Omega(u,v)$ on Ω and, for simplicity, impose a homogeneous Dirichlet condition on $\partial\Omega$. The problem is then to find $u \in H_0^1(\Omega)$ such that

$$a(u,v) = (f,v), \quad \forall\, v \in H_0^1(\Omega).$$

For the two levels of triangulations into substructures Ω_i and elements introduced earlier, we assume shape regularity and that the elements and substructures satisfy the usual rules of finite element triangulations; see e.g. Ciarlet [26]. $V^H(\Omega)$ and $V^h(\Omega)$ are the spaces of continuous, piecewise linear functions, on the two triangulations, which vanish on the boundary $\partial\Omega$.

The discrete problem is then to find $u_h \in V^h(\Omega)$ such that

$$a(u_h,v_h) = (f,v_h), \quad \forall\, v_h \in V^h(\Omega). \tag{3.1}$$

3.2 A Partial Change to Hierarchical Basis

The space V^H plays a major role in many domain decomposition algorithms, since it allows the easy construction of a global coarse problem, which is needed for the global transportation of information. We will briefly discuss this space and the manner in which it is used. In the following we will always work only with the Schur complement, though the same results hold for the original stiffness matrices. Let $\tilde{V}^h$ be the subspace of functions in V^h which are discrete harmonic on the interiors of the subdomains. The basis functions $\{\phi^k\}$ are defined to be equal to the usual finite element basis functions on the interface between substructures and extended as discrete harmonic onto the interiors of the subdomains.

We first consider the substructures, which have diameters on the order of H, as large elements and consider their nodal basis functions, $\{\psi^j\}$. The space V^H is the space spanned by $\{\psi^j\}$. We decompose the space $\tilde{V}^h$ into

$$\tilde{V}^h = V^H \oplus (\tilde{V}^h \setminus V^H).$$

Any function in $\tilde{V}^h$ can be represented in the basis $\{\phi^k\}$ or in the partial hierarchical basis. We group the nodal basis functions into two groups, those associated with the vertex of a substructure, $\{\phi^k_V\}$, and all the rest, $\{\phi^k_E\}$. The two representations can then be written as

$$u^h = \sum_j x_{V_j} \psi^j + \sum_k x_{E_k} \phi^k_E,$$

and

$$u^h = \sum_j y_{V_j} \phi^j_V + \sum_k y_{E_k} \phi^k_E.$$

The mapping between the two sets of coefficients is given by

$$\begin{pmatrix} x_E \\ x_V \end{pmatrix} = \begin{pmatrix} I & -R^T \\ 0 & I \end{pmatrix} \begin{pmatrix} y_E \\ y_V \end{pmatrix}, \quad \begin{pmatrix} y_E \\ y_V \end{pmatrix} = \begin{pmatrix} I & R^T \\ 0 & I \end{pmatrix} \begin{pmatrix} x_E \\ x_V \end{pmatrix}.$$

In the case of linear basis functions, R^T simply represents linear interpolation. These mappings are completely local to each substructure, i.e., we can map the coefficients between the two different basis one substructure at a time without any information about other substructures. This we write algebraically as

$$\begin{pmatrix} x^{(i)}_E \\ x^{(i)}_V \end{pmatrix} = \begin{pmatrix} I & -R^{(i)^T} \\ 0 & I \end{pmatrix} \begin{pmatrix} y^{(i)}_E \\ y^{(i)}_V \end{pmatrix}, \quad \begin{pmatrix} y^{(i)}_E \\ y^{(i)}_V \end{pmatrix} = \begin{pmatrix} I & R^{(i)^T} \\ 0 & I \end{pmatrix} \begin{pmatrix} x^{(i)}_E \\ x^{(i)}_V \end{pmatrix}.$$

We now show how it is possible to obtain global bounds from local bounds (see Section 2.5) using the partial change to the hierarchical basis. Consider

$$S^{(i)} = \begin{pmatrix} S^{(i)}_{EE} & S^{(i)}_{EV} \\ S^{(i)^T}_{EV} & S^{(i)}_{VV} \end{pmatrix}.$$

We first make a partial change to hierarchical basis, by post-multiplying by

$$T^{(i)} = \begin{pmatrix} I & R^{(i)^T} \\ 0 & I \end{pmatrix},$$

and pre-multiplying by $T^{(i)^T}$. This gives,

$$T^{(i)^T} S^{(i)} T^{(i)} = \begin{pmatrix} S_{EE}^{(i)} & \text{Non-zero} \\ \text{Non-zero} & \tilde{S}_{VV}^{(i)} \end{pmatrix}.$$

We now replace $S_{EE}^{(i)}$ with any non-singular preconditioner and drop the coupling between the vertices and the other nodes. This gives,

$$\begin{pmatrix} \hat{S}_{EE}^{(i)} & 0 \\ 0 & \tilde{S}_{VV}^{(i)} \end{pmatrix}.$$

We can also replace the matrix $\tilde{S}_{VV}^{(i)}$ with any spectrally equivalent matrix. We note that $\tilde{S}_{VV}^{(i)}$ is spectrally equivalent to the discrete Laplacian on the coarse grid.

Finally, we return to the nodal basis by post-multiplying by $T^{(i)^{-1}}$ and pre-multiplying by $T^{(i)^{-T}}$ and obtain

$$\hat{S}^{(i)} = \begin{pmatrix} I & 0 \\ -R^{(i)} & I \end{pmatrix} \begin{pmatrix} \hat{S}_{EE}^{(i)} & 0 \\ 0 & \tilde{S}_{VV}^{(i)} \end{pmatrix} \begin{pmatrix} I & -R^{(i)^T} \\ 0 & I \end{pmatrix}. \tag{3.2}$$

This final form of the local contribution to the preconditioner has the same null space as $S^{(i)}$, as is needed, see Section 2.5. It is important to note that we never need to explicitly carry out this change to a partial hierarchical basis. The same procedure can be carried out for the stiffness matrices arising from linear elasticity without any difficulties, because the null space of infinitesimal rigid motions is contained in the V^H space.

By subassembly we can construct the global preconditioner

$$\hat{S} = \begin{pmatrix} I & 0 \\ -R & I \end{pmatrix} \begin{pmatrix} \hat{S}_{EE} & 0 \\ 0 & \tilde{S}_{VV} \end{pmatrix} \begin{pmatrix} I & -R^T \\ 0 & I \end{pmatrix}. \tag{3.3}$$

This procedure is possible because the actions of the $R^{(i)^T}$ for two adjacent subdomains along a shared edge (face) are identical, e.g. linear interpolation from the shared vertex nodes. This property is very important and is not shared by the wirebasket based algorithms of Mandel [59] and Bramble, Pasciak, and Schatz [15].

The preconditioner is easily inverted to give

$$\hat{S}^{-1} = \begin{pmatrix} R^T \\ I \end{pmatrix} \tilde{S}_{VV}^{-1} \begin{pmatrix} R & I \end{pmatrix} + \begin{pmatrix} I \\ 0 \end{pmatrix} \hat{S}_{EE}^{-1} \begin{pmatrix} I & 0 \end{pmatrix}.$$

Note that there are independent problems associated with each edge (face and edge) and the coarse problem.

In [36], Dryja and Widlund describe and analyze, using an additive Schwarz approach, the basic iterative substructuring method, which is a vertex based method. We will now briefly review this method and their results. The preconditioner is obtained by making a partial change to hierarchical basis as described above. The couplings between the different edges (edges and faces in three dimensions) are then dropped as are the couplings between the edges (edges and faces) and the vertices of the substructures. This produces a block diagonal preconditioner with a block for each edge (face and edge) and a block for the substructure vertices.

In the additive Schwarz analysis of this method the subspaces of V^h that Dryja and Widlund use are V^H and the spaces $V^h \cap H_0^1(\Omega_i \cup \Gamma_{ij} \cup \Omega_j)$, where Ω_i and Ω_j are neighboring subdomains. They proceed by partitioning u^h as required for Lemma 2.3.3 and obtain the result, for two dimensions,

$$\kappa \leq C(1 + \log(H/h))^2.$$

In three dimensions the bound is

$$\kappa \leq C(H/h)(1 + \log(H/h)).$$

These bounds are sharp.

We now present an argument why the bound in three dimensions is at least $C(H/h)$ in the case of piecewise linear elements. In constructing the partition of u^h as required for Lemma 2.3.3, we must use the interpolant of u^h at the vertex nodes because the only space with functions which can be nonzero at the vertices of the subdomains is the V^H space. Consider the finite element function u^h which is one at a single subdomain vertex and zero at all other nodes. Let α represent the elements for which u^h is nonzero, and β the substructures which contain the given vertex. The volume of α is $O(h^3)$ and that of β is $O(H^3)$.

The $H^1(\Omega)$ energy of u^h is

$$|u^h|_{H^1(\Omega)} = \int_\alpha (\nabla u^h)^2 = O(h),$$

and the energy of its interpolant onto V^H is

$$|I^h u^h|_{H^1(\Omega)} = \int_\beta (\nabla I^h u^h)^2 = O(H).$$

Therefore the energy of its interpolant is $O(H/h)$ as large, thus the bound for Lemma 2.3.3 must be at least $O(H/h)$. This is also the reason why Yserentant's hierarchical basis method is generally unsuccessful in three dimensions. For a detailed analysis of the hierarchical basis method in three dimensions, see Ong [72].

3.3 The Hierarchical Basis Method

The hierarchical basis method provides a general purpose preconditioner for second order elliptic problems in the plane; see Yserentant [87],[88]. The algorithm is given in terms of a set of spaces $V^{h_i}, i = 0, \cdots, j$, which are successive refinements by a factor of two of $V^{h_0} = V^H$. V^{h_i} is the set of piecewise linear finite element functions after i levels of refinement from the original coarse triangulation with $V^{h_j} = V^h$. V^h is a direct sum of subspaces

$$V^h = V_0^h \oplus V_1^h \oplus \cdots \oplus V_j^h,$$

where $V_i^h = V^{h_i} \backslash V^{h_{i-1}}$. In other words, V_i^h is the set of piecewise linear functions in V^{h_i}, which are zero at the nodal points of the triangles of all coarser triangulations, see Figure 3.1 for a one-dimensional case. For the spaces V_i^h, we choose a basis of standard nodal functions of V^{h_i} associated with the new nodes. The resulting basis for the entire space V^h is much closer to being orthogonal in the H^1 sense, than the standard nodal basis functions, and the stiffness matrix is therefore much better conditioned.

Yserentant's preconditioner is block diagonal in the new basis. The first block is defined by the finite element model for the subspace V^H and the others are diagonal. In matrix notation the resulting system, which is solved by a conjugate gradient method, is of the form

$$D^{-1/2} H^T K H D^{-1/2} \hat{x} = g,$$

or equivalently,

$$D^{-1} H^T K H \tilde{x} = \tilde{g}.$$

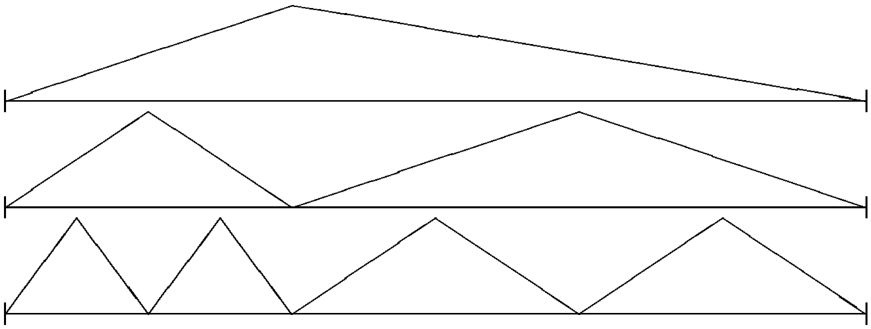


Figure 3.1: Hierarchical Basis Functions in One Dimension

Here K is the original stiffness matrix, H represents the transformation from the hierarchical to the nodal basis, and D is block diagonal and obtained from H^TKH, as described above. In an implementation, there is no need to represent the stiffness matrix explicitly in the new basis. Instead, we perform the basis change on the vectors as needed.

The important theorem due to Yserentant [88] is

Theorem 3.3.1 *For a uniformly elliptic, scalar partial differential equation which has been discretized using a quasi-uniform hierarchical h-version finite element method in two dimensions with $j = \log(H/h)$ levels of triangulation the condition number of the preconditioned stiffness matrix satisfies*

$$\kappa(\hat{K}) = \kappa(D^{-1/2}H^TKHD^{-1/2}) \leq C(j+1)^2.$$

In [88], Yserentant develops the algorithms needed for performing the basis change between the hierarchical and the nodal basis and demonstrates that each requires fewer than $2n$ additions and n divisions by 2, where n is the dimension of the finite element space. The following algorithm is valid for both one and two dimensions.

```
Algorithm to form $x \leftarrow Hx$
      for $k = 1$ to number of levels
            for $i$ with $x_i$ on level $k$
                  $x_i = x_i + (x_{I1_i} + x_{I2_i})/2$
            next $i$
      next $k$
```

The integer arrays $I1$ and $I2$ contain pointers to the two neighbors of x_i which are on the next coarser level. We can regard the algorithm as defining a factored form of the matrix H. The nodal to hierarchical transformation is similar. We note that the coarse mesh needs not be uniform; see Figure 3.1. If the refinements are not uniform, then the weights in the algorithms have to be adjusted. It is also possible to continue the refinement only in selected subregions.

3.4 Iterative Substructuring Methods

We again examine the iterative substructuring algorithms introduced in Section 3.2 from a slightly different perspective.

Iterative substructuring algorithms use a different splitting of the space V^h into $N+1$ subspaces;

$$V^h = V^h_{harm} \oplus V_0^h(\Omega_1) \oplus \cdots \oplus V_0^h(\Omega_N) \ .$$

For each substructure Ω_i, we thus have a subspace $V_0^h(\Omega_i) = V^h \bigcap H_0^1(\Omega_i)$. The elements of V^h_{harm} are piecewise, discrete harmonic functions, i.e. they are orthogonal, in the sense of the bilinear form $a(\cdot,\cdot)$, to all the other subspaces. It is easy to show that an element of V^h_{harm} is uniquely determined by its values on $\Gamma = \bigcup \partial\Omega_i$.

In a first step of many substructuring algorithms, the variables interior to the Ω_i are eliminated. We partition the vector $x = (x_I, x_B)$ and the stiffness matrix K accordingly. The system that remains to be solved is, after a block Gaussian elimination step,

$$Schur(K)x_B = g. \tag{3.4}$$

Here $Schur(K)$ is a Schur complement defined by

$$Schur(K) = K_{BB} - K_{IB}^T K_{II}^{-1} K_{IB}.$$

A particular iterative substructuring method is defined by the choice of a preconditioner for equation (3.4). Finally, when accurate enough values on Γ have been computed, the values elsewhere are determined by solving N separate Dirichlet problems on the individual substructures. We note that it is not necessary to compute the elements of $Schur(K)$ since, in the conjugate gradient iteration, this matrix is needed only in terms of matrix-vector products. Such a product can be found at the expense of solving one problem on each of the substructures.

A number of preconditioners can be described as follows: We first carry out a partial change of basis, associating the standard basis functions of V^H with the vertices of the substructures. As shown in Section 3.2, we represent the Schur complement, in the new basis, as

$$\begin{pmatrix} S_{EE} & S_{EV} \\ S_{EV}^T & S_{VV} \end{pmatrix} .$$

Here S_{VV} is denotes the part of the Schur complement associated with the vertices of the substructures and S_{EE} is that part associated with the edges between substructures.

The preconditioner for this system is given by

$$\begin{pmatrix} \hat{S}_{EE} & 0 \\ 0 & \hat{S}_{VV} \end{pmatrix} ,$$

where $\hat{S}_{VV}$ is the matrix for the coarse mesh finite element problem and $\hat{S}_{EE}$ is a block diagonal matrix. Each of its blocks is associated with the variables of a single edge Γ_{ij}. The operator J, mentioned before, can be used for this purpose; for other examples of such algorithms, see the references given in Section 3.1.

3.5 The Hybrid Algorithm

We can combine the two main ideas of Section 3.3 and 3.4, as follows: We first represent the stiffness matrix in the hierarchical basis and then eliminate the interior variables of all the substructures. We proceed by solving the remaining Schur complement system approximately without further preconditioning. Finally we use the resulting values as boundary data for the local problems on the individual substructures.

In the new algorithm, we proceed differently, but as we will see, we will obtain the same approximate solution without the considerable expense of explicitly converting the stiffness matrix into the hierarchical basis. In our algorithm we work with the standard nodal basis, while eliminating the interior variables, only changing to the hierarchical basis on Γ, the set of interfaces and vertices. The resulting linear system is similar to that of Section 3.4

$$D_{BB}^{-1} H_{BB}^T Schur(K) H_{BB} \tilde{x}_B = \tilde{g}$$

It is important to note that we do not use any further preconditioning of the variables associated with the edges Γ_{ij}.

This method offers several possible advantages over the standard hierarchical basis algorithm. The conjugate gradient iteration is carried out over a much smaller set of unknowns and as we will show that the condition number is smaller. The solution of the subproblems is easily parallelizable since they are independent. The hierarchical basis method in its original form appears to offer less opportunity for this trivial type of parallelization. The change of basis required in each iteration step now consists of completely independent one dimensional problems instead of a two dimensional problem. The basic observation is that the values at a node on Γ_{ij} can be computed using only the coefficients for the hierarchical basis functions related to that edge.

We now prove the almost optimality of our algorithm using two simple lemmas and Yserentant's result.

Lemma 3.5.1 *Let G represent a change of basis which leaves the space of variables on Γ invariant. Then the Schur complement associated with this set of unknowns is independent of the choice of bases for $V_0^h(\Omega_k)$.*

Proof. Let x_I be the vector of unknowns associated with $V_0^h(\Omega_k)$, $\forall\, k$, and x_B be those associated with Γ. The most general basis transformation considered here is of the form

$$\begin{pmatrix} x_I \\ x_B \end{pmatrix} = \begin{pmatrix} G_{II} & G_{IB} \\ 0 & G_{BB} \end{pmatrix} \begin{pmatrix} \tilde{x}_I \\ \tilde{x}_B \end{pmatrix} .$$

In the new basis, the stiffness matrix is

$$\tilde{K} = \begin{pmatrix} G_{II}^T & 0 \\ G_{IB}^T & G_{BB}^T \end{pmatrix} \begin{pmatrix} K_{II} & K_{IB} \\ K_{IB}^T & K_{BB} \end{pmatrix} \begin{pmatrix} G_{II} & G_{IB} \\ 0 & G_{BB} \end{pmatrix} .$$

A straightforward calculation shows that its Schur complement satisfies

$$Schur(\tilde{K}) = G_{BB}^T Schur(K) G_{BB} \ .$$

∎

The following result follows easily by a Rayleigh quotient argument.

Refinement levels	1	2	3	4	5	6	7
# of unknowns on Γ	3	7	15	31	63	127	255
H/h	4	8	16	32	64	128	256
New method	1.68	2.66	3.82	5.18	6.75	8.52	10.50
No preconditioning	3.05	6.88	14.20	28.63	57.37	114.79	230.49

Table 3.1: Condition Numbers for Two Subdomain Case

Lemma 3.5.2 *Let K be symmetric, positive definite. Then, the condition numbers of K and its Schur complement satisfy*

$$\kappa(Schur(K)) \leq \kappa(K).$$

Our main result is given in

Theorem 3.5.1 *The condition number of the hybrid algorithm, introduced in this section, is bounded by that of Yserentant's method. Thus, it is bounded by $C(1 + \log(H/h))^2$.*

Proof. We use Lemma 3.5.1 twice and the fact that D is block diagonal to obtain

$$\begin{aligned} Schur(D^{-1/2}H^TKHD^{-1/2}) &= D_{BB}^{-1/2}Schur(H^TKH)D_{BB}^{-1/2} \\ &= D_{BB}^{-1/2}H_{BB}^TSchur(K)H_{BB}D_{BB}^{-1/2}\ . \end{aligned}$$

By using Lemma 3.5.2, we obtain

$$\begin{aligned} \kappa(D_{BB}^{-1/2}H_{BB}^TSchur(K)H_{BB}D_{BB}^{-1/2}) &= \kappa(Schur(D^{-1/2}H^TKHD^{-1/2})) \\ &\leq \kappa(D^{-1/2}H^TKHD^{-1/2}), \end{aligned}$$

which is bounded by $C(1 + \log(H/h))^2$; see Yserentant (Thm. 4.1) [88]. ∎

3.6 Numerical Experiments

3.6.1 General Results

In a first set of experiments, we consider the domain $\bar{\Omega} = \bar{\Omega}_1 \bigcup \bar{\Omega}_2$ where $\bar{\Omega}_1$ and $\bar{\Omega}_2$ are unit squares aligned along an edge $\Gamma = \bar{\Omega}_1 \bigcap \bar{\Omega}_2$. We use the standard regular mesh and the usual five point discretization for the Laplacian. The results are listed in Table 3.1.

Remark: Our experiments show that the condition number grows faster than $(1 + \log(H/h))$ for the two subdomain case. We note that for a number of preconditioners the condition number remains bounded in this case. This is true for the preconditioner J if we solve a Dirichlet problem, cf. [9], but not for a Neumann problem. Yet our method and that based on the J operator both have condition numbers which grow like $(1 + \log(H/h))^2$ in the many subdomain case.

In a second set of experiments, we consider the case of many substructures. The unit square Ω is subdivided uniformly into 4, 16, 64, or 256 square subdomains and the same model problem is solved using uniform meshes. We compare our results with a set of experiments reported in Yserentant [88]. The results are given in Table 3.2.

The coarse problem and the problems associated with the edges which together make up the preconditioner are independent. We can therefore scale the contribution of the coarse model by a scalar factor α selecting the value of the parameter for which the convergence is fastest. In our numerical experiments, we have found that for our model problem $\alpha \approx 3.6$ is the best for a wide range of refinements. We note that the condition number grows quadratically in the logarithmic factor for all $\alpha > 0$. Our numerical results are reported for $\alpha = 3.6$.

3.6.2 On the Selection of α

Our hierarchical preconditioner is obtained by subassembly just as the preconditioner for the basic iterative substructing algorithm. The local contribution to the preconditioner has the same null space as $S^{(i)}$. Hence the global bound on the condition number of the preconditioned problem is bounded by the worst bound on the local problems, see Section 2.5. Therefore the optimal α obtained locally will likely be near the global optimal α. Since we are only dealing, in the theory, with bounds on condition numbers and not the condition numbers themselves it is not clear that the optimal α in the two cases need to correspond exactly. Numerically we observe that they often are close.

In Figure 3.2 we plot the global condition number and local condition number as a function of α for a large problem with $H/h = 32$ and 256 square subdomains and the model problem. The behavior in Figure 3.2 is representative of the relationship

Levels	3	4	5	6	7	8	9
# of unknowns in Ω	7^2	15^2	31^2	63^2	127^2	255^2	511^2
Yserentant's method	10.59	19.53	31.85	47.14	65.38	86.51	110.49
4 Subdomains							
New method	3.35	5.18	10.87	15.45			
No preconditioning	9.77	21.50	44.97	91.98			
H/h	4	8	16	32			
# of unknowns on Γ	13	29	61	125			
16 Subdomains							
New method		4.89	7.94	11.81	16.45		
No preconditioning		35	75	155	316		
H/h		4	8	16	32		
# of unknowns on Γ		81	177	369	753		
64 Subdomains							
New method			5.29	8.52	12.54	17.32	
No preconditioning			137	290	599	1217	
H/h			4	8	16	32	
# of unknowns on Γ			385	833	1729	3521	
256 Subdomains							
New method				5.46	8.71	12.78	17.61
No preconditioning				546	1152	2372	4766
H/h				4	8	16	32
# of unknowns on Γ				1665	3585	7425	15105

Table 3.2: Condition Numbers for the Many Subdomain Case

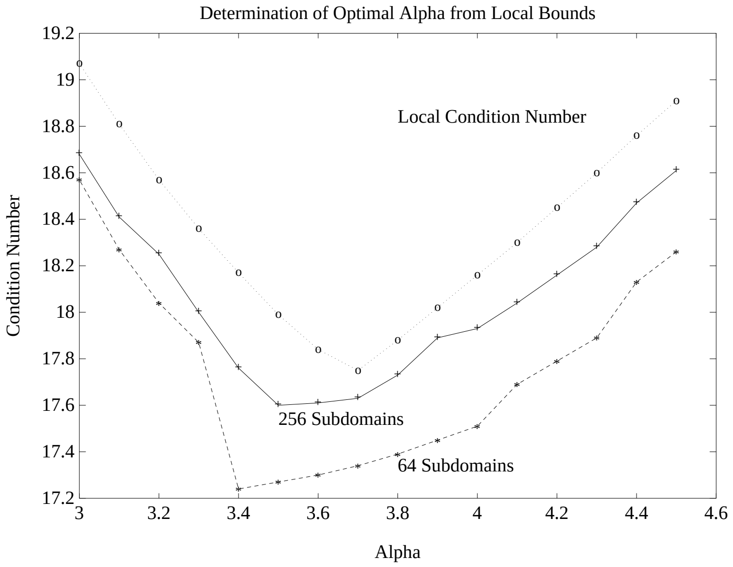


Figure 3.2: Local vs Global Condition Numbers

between the local condition number and the global condition number. We conclude by stating that, in certain situations, we may nearly determine the optimal α by solving only local problems. However, interesting questions still remain on the determination of optimal scaling.

3.6.3 Numerical Comparisons to Other Methods

We numerically compare four different edge preconditioners for the problem with many substructures. They are given below.

- The J operator first proposed by Dryja in [31] and used very successfully by Bramble, Pasciak, and Schatz in their important algorithm for the case with many substructures [12].
- The operator we shall call G proposed by Golub and Mayers [40].
- The hierarchical domain decomposition algorithm, which we call the H method.
- The basic iterative substructuring method, i.e. the method where the Schur complements for each edge are actually formed and factored, this we symbolize in the tables by V^H.

Edge	Unknowns along Edge							
Solver	3		7		15		31	
16 Subdomain Case								
J	5.44	7	8.99	9	13.44	10	18.44	11
H	4.89	7	7.94	10	11.81	11	16.45	13
G	4.83	8	7.34	9	10.53	9	14.39	9
V^H	4.92	7	7.59	9	10.89	10	14.84	11
No Preconditioning	35.26	18	75.10	33	155.20	53	315.82	78
Number of Unknowns	81		177		369		753	
H/h	4		8		16		32	

Table 3.3: Condition Numbers and Iteration Counts for Model Problem

Edge	Unknowns along Edge							
Solver	3		7		15		31	
64 Subdomain Case								
J	5.52	8	8.99	9	13.44	10	18.46	11
H	5.26	9	8.51	11	12.53	13	17.30	14
G	5.47	10	8.15	9	11.60	10	15.72	10
V^H	5.35	9	8.19	10	11.54	12	15.62	13
No Preconditioning	137.38	46	290.44	70	598.53	*	1216.67	*
Number of Unknowns	385		833		1729		3521	
H/h	4		8		16		32	

Table 3.4: Condition Numbers and Iteration Counts for Model Problem

Again the model problem is studied but we should keep in mind that the results will be the same in the case when the coefficients are constant on each substructure but have large jumps between substructures, assuming that the correct scaling of the preconditioner on each subdomain is used.

From Tables 3.3, 3.4 and 3.5 we conclude that the three preconditioners are basically equivalent. Our stopping tolerance is a relative decrease in the energy norm of the error of 10^{-5}.

3.6.4 Results for Elasticity

We first consider the two domain case, as discussed above, where we use membrane elements to model our elasticity problem. There are two degrees of freedom

Edge	Unknowns along Edge							
Solver	3		7		15		31	
256 Subdomain Case								
J	5.57	7	8.99	9	13.45	10	18.47	11
H	5.46	9	8.71	10	12.78	12	17.61	13
G	5.62	10	8.41	9	11.98	10	16.21	10
V^H	5.46	9	8.27	10	11.77	12	15.90	13
No Preconditioning	545.90	94	1151.95	*	2372.39	*	4766.38	*
Number of Unknowns	1665		3585		7425		15105	
H/h	4		8		16		32	

Table 3.5: Condition Numbers and Iteration Counts for Model Problem

Membrane Elements

Preconditioning	H/h							
	4		8		16		32	
None	2.37	4	4.96	7	10.06	13	20.20	21
Hierarchical	2.82	4	3.99	7	5.73	11	7.59	14
J	1.50	4	1.72	5	1.78	6	1.80	6
Number of unknowns	6		14		30		62	

Table 3.6: Condition Numbers/Iteration Counts for Two Subdomain Case

per node, one for horizontal and one for vertical displacement. These results are listed in Table 3.6, along with results using the J operator, in the style of Dryja [31].

In Table 3.7 we report on the case of many subdomains. Again we select a weight α on the 'coarse' problem which leds to good condition numbers. As with the Laplacian, the different methods are about equally effective.

Lastly we consider the two subdomain case using shell elements. We also use the hierarchical domain decomposition method and the J operator. The results are listed in Table 3.8. For shell elements there are three degrees of freedom per node; displacements in the x, y, and z directions. Since shell elements model physically thin materials the z displacements may be on a different scale than the x and y displacements. This is reflected in the poorer results obtained.

Number of Subdomains	Nodes along Edge	Number of Unknowns on Γ	V^H Method		Hierarchical Method		J Operator Preconditioning	
16	3	162	10.52	15	10.69	15	10.46	16
	7	354	14.84	17	16.08	20	14.88	16
	15	738	19.83	19	22.11	23	20.04	18
	31	1506	25.51	20	28.89	26	25.95	20
64	3	770	12.13	18	12.24	18	11.95	18
	7	1666	16.92	19	18.28	23	16.82	20
	15	3458	22.37	22	24.90	26	22.44	23
	31	7042	28.52	25	32.21	30	28.85	22
256	3	3330	12.42	18	12.72	18	12.39	18
	7	7170	17.31	19	18.16	23	17.01	20
	15	14850	22.88	22	24.73	26	22.70	23
	31	30210	29.15	25	32.30	29	29.07	23

Table 3.7: Condition Numbers and Iteration Counts for Membrane Elements

Shell Elements

Preconditioning	H/h							
	4		8		16		32	
None	41.82	6	40.74	12	86.98	23	396	42
Hierarchical	31.67	7	22.17	12	26.42	20	51.44	28
J	25.09	6	13.36	11	14.64	18	33.58	28
Number of unknowns	9		21		45		93	

Table 3.8: Condition Numbers/Iteration Counts for Two Subdomain Case

Chapter 4

An Optimal Preconditioner for Linear Elasticity

4.1 The Algorithm

We consider the Schur complement problem,

$$Sx_B = g,$$

obtained by a substructuring approach, see Chapter 1, after the interior variables have been eliminated. This problem will be solved using an additive Schwarz style scheme.

The variational problem is to find $\tilde{u}_h \in \tilde{V}^h$ such that

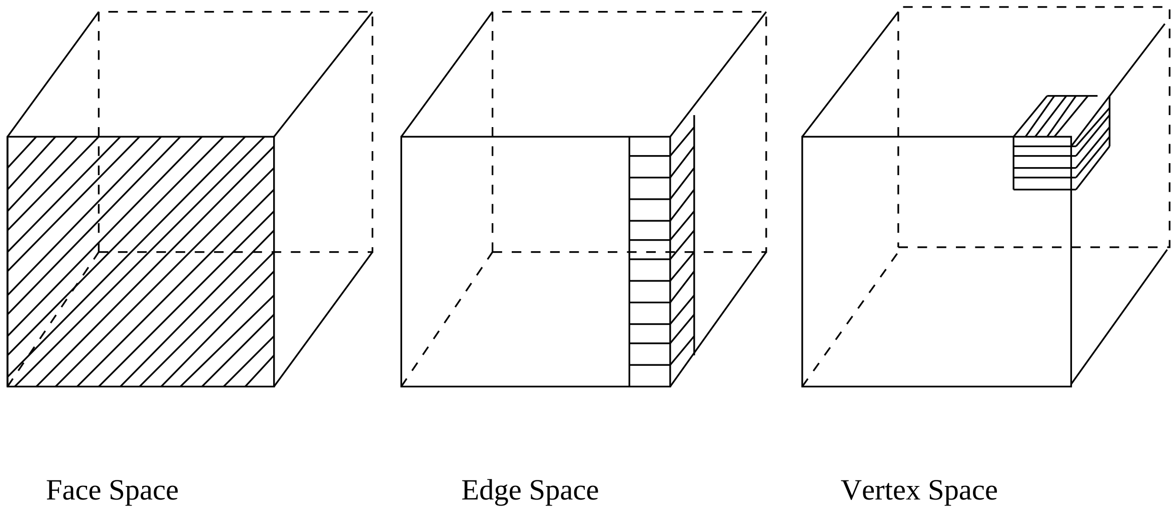
$$a_\Omega(\tilde{v}_h, \tilde{u}_h) = f(\tilde{v}_h), \qquad \forall\, \tilde{v}_h \in \tilde{V}^h,$$

where $\tilde{V}^h$ is the subspace of V^h of functions which are discrete harmonic in the interiors of the subdomains. The matrix formulation of the problem is to find x_B such that

$$y_B^T S x_B = y_B^T g, \qquad \forall\, y_B.$$

As in Chapter 1, the vectors y_B and x_B are the coefficients of the finite element functions associated with the interface Γ.

The numerical algorithm will be constructed in an additive Schwarz framework. We work with subspaces of $\tilde{V}^h$ associated with overlapping regions of Γ and a global coarse space. We cover the surface (or curve) Γ with overlapping patches, $\{\Gamma^{F_i}, \Gamma^{E_j}, \Gamma^{V_k}\}$ defined below. We can regard the boundaries of the substructures



Face Space Edge Space Vertex Space

Figure 4.1: Subspaces for a Cubic Substructure

as consisting of three parts: the substructure vertices, the edges between substructure vertices and the faces of the substructures (in two dimensions there are only vertices and edges). The Γ^{F_i} are the faces of the substructures. We choose Γ^{E_j} to be regions consisting of an edge and an overlap of order H onto all of the adjacent faces. The Γ^{V_k} are regions consisting of a vertex and an overlap of order H onto all adjacent faces and edges. We constrain the overlap so that no portion of Γ is covered more than four times. See Figure 4.1 where we represent the restriction of the face, edge and vertex regions to a single substructure.

The subspaces of $\tilde{V}^h$ are given by

$$V^H,$$

$$\tilde{V}^h_{F_i} = \{\phi \in \tilde{V}^h : \mathrm{supp}(\phi|_\Gamma) \subseteq \Gamma^{F_i}\},$$

$$\tilde{V}^h_{E_j} = \{\phi \in \tilde{V}^h : \mathrm{supp}(\phi|_\Gamma) \subseteq \Gamma^{E_i}\},$$

and

$$\tilde{V}^h_{V_k} = \{\phi \in \tilde{V}^h : \mathrm{supp}(\phi|_\Gamma) \subseteq \Gamma^{V_i}\}.$$

It is easy to see that each of these spaces is a subspace of $\tilde{V}^h$ and that

$$\tilde{V}^h = (\sum_i \tilde{V}^h_{F_i}) + (\sum_j \tilde{V}^h_{E_j}) + (\sum_k \tilde{V}^h_{V_k}).$$

The projection onto each 'face' subspace is given by

$$P_{F_i} = R_{F_i}^T S_{F_i}^{-1} R_{F_i} S,$$

where R_{F_i} is the restriction operator, which returns only those unknowns which are associated with Γ^{F_i}, and S_{F_i} is the principal minor of the Schur complement S which is associated with that same set of unknowns. The projections onto 'edge' and 'vertex' subspaces are formed in the same manner. Naturally the restriction operators R need never be explicitly represented. Instead we would use something like hardware scatter-gather.

For the coarse space V^H, the projection operator is of the form

$$P_H = R_H^T K_H^{-1} R_H S.$$

The operator R_H^T represents linear interpolation from V^H to $\tilde{V}^h$ and K_H is the stiffness matrix for the original problem treating the substructures as elements. K_H can also be obtained by making a partial change to a hierarchical basis of each $S^{(k)}$ so that the nodal functions associated with the vertices of the substructure are the nodal functions in the V^H space; cf. Smith and Widlund [78] and Section 3.2. The principal minor of the Schur complement, which is associated with these vertex nodes, then forms the contribution to the stiffness matrix K_H from the given substructure.

We now present a detailed description of the algorithm. We note that many opportunities exist for parallelism between and within each step.

The New Algorithm

1. Form the stiffness matrices
$$K^{(j)} = \begin{pmatrix} K_{II}^{(j)} & K_{IB}^{(j)} \\ K_{IB}^{(j)^T} & K_{BB}^{(j)} \end{pmatrix}$$
for each substructure by integration.
2. Factor $K_{II}^{(j)}$ for each substructure.
3. Form the Schur complements $S^{(j)} = K_{BB}^{(j)} - K_{IB}^{(j)^T} K_{II}^{(j)^{-1}} K_{IB}^{(j)}$.
4. Form and factor the coarse stiffness matrix K_H.
5. Form, by subassembly, S_{F_i}, S_{E_i}, and S_{V_i}.
6. Factor S_{F_i}, S_{E_i}, and S_{V_i}.
7. Form the right hand sides $b^{(j)}$ for each substructure by integration.
8. Modify the right hand sides; $b_B^{(j)} = b_B^{(j)} - K_{IB}^{(j)^T} K_{II}^{(j)^{-1}} b_I^{(j)}$.
9. Solve $Sx_B = b_B$ using a preconditioned conjugate gradient method with the preconditioner
$$\hat{S}^{-1} = R_H^T K_H^{-1} R_H + \sum_i R_{F_i}^T S_{F_i}^{-1} R_{F_i} + \sum_j R_{E_j}^T S_{E_j}^{-1} R_{E_j} + \sum_k R_{V_k}^T S_{V_k}^{-1} R_{V_k}.$$
10. Form the right hand sides for the problems on the interiors of the subdomains
$$b_I^{(j)} = b_I^{(j)} - K_{II}^{(j)^{-1}} K_{IB}^{(j)} x_B^{(j)}.$$
11. Solve for the interior unknowns $x_I^{(j)} = K_{II}^{(j)^{-1}} b_I^{(j)}$.

Steps 1 though 6 can be regarded as a preprocessing stage independent of the particular loads; they need not be repeated for different loads.

If we exclude from the algorithm all of the overlap and the 'vertex' spaces we obtain the iterative substructuring algorithm presented in Dryja and Widlund [36], see Section 3.2. That is, in the definition of $\hat{S}^{-1}$, we drop the terms
$$\sum_k R_{V_k}^T S_{V_k}^{-1} R_{V_k}$$
and restrict Γ_{E_j} to be the edge extended out to but not including the first nodes on the adjacent faces. This implies that the restriction operator R_{E_j} retrieves only those coefficients associated with the nodes along the edge. In the new

algorithm it also retrieves some coefficients associated with nodes on the adjacent faces. The iterative substructuring algorithm is also very similar to that introduced by Bramble, Pasciak, and Schatz [12]. In two dimensions these algorithms have condition numbers which grow like $(1+\log(H/h))^2$, while in three dimensions the condition numbers for these algorithms grow faster than (H/h), see Section 3.2.

4.2 A Proof of Optimality

The main result for this algorithm is now given in

Theorem 4.2.1 *The condition number of the preconditioned system is bounded independently of the size of the substructures H and the size of the elements h, i.e.*

$$\kappa(P) = \kappa(\hat{S}^{-1}S) \leq C.$$

Proof. We note that no node on Γ is contained in more than 4 of the regions $\Gamma^{F_i}, \Gamma^{E_j}$ and Γ^{V_k}. Therefore the graph, as described in Lemma 2.3.1, can be colored by a fixed finite number of colors. Hence, by Lemma 2.3.1, we have $\lambda_{\max}(P) \leq C$.

We now use the uniform ellipticity of our bilinear form $a_\Omega(u,u)$,

$$c||u||^2_{H^1(\Omega)} \leq a_\Omega(u,u) \leq C||u||^2_{H^1(\Omega)},$$

to show that we can work in the $(H^1(\Omega))^q$ norm instead of the equivalent norm induced by $a_\Omega(u,u)$. To obtain a lower bound, we must demonstrate that for all $\tilde{u}^h \in \tilde{V}^h$, there exists a representation

$$\tilde{u}^h = \tilde{u}^H + \sum_i \tilde{u}^h_{E_i} + \sum_j \tilde{u}^h_{F_j} + \sum_k \tilde{u}^h_{V_k},$$

so that

$$\begin{aligned} ||\tilde{u}^H||^2_{H^1(\Omega)} + \textstyle\sum_i ||\tilde{u}^h_{E_i}||^2_{H^1(\Omega)} + \sum_j ||\tilde{u}^h_{F_j}||^2_{H^1(\Omega)} + \sum_k ||\tilde{u}^h_{V_k}||^2_{H^1(\Omega)} \\ \leq C_0^2 ||\tilde{u}^h||^2_{H^1(\Omega)} \end{aligned} \tag{4.1}$$

where C_0 is independent of $\tilde{u}^h$, h and H.

We construct this representation as follows. We extend the boundary regions $\Gamma^{E_i}, \Gamma^{F_j}, \Gamma^{V_k}$ into the interiors of the neighboring substructures. This results in a large collection of overlapping regions. Now apply the result of Lemma 4.2.1, given below, to each component of the vector valued function $\tilde{u}^h$ separately to obtain a representation of $\tilde{u}^h$,

$$\tilde{u}^h = u^H + \sum_i u^h_{E_i} + \sum_j u^h_{F_j} + \sum_k u^h_{V_k},$$

which satisfies

$$||u^H||^2_{H^1(\Omega)} + \sum_i ||u^h_{E_i}||^2_{H^1(\Omega)} + \sum_j ||u^h_{F_j}||^2_{H^1(\Omega)} + \sum_k ||u^h_{V_k}||^2_{H^1(\Omega)} \leq C_0^2||\tilde{u}^h||^2_{H^1(\Omega)}.$$

The equivalence of the $H^1(\Omega)$ norm and semi-norm on V^h follows from Friedrichs' inequality and allows us to apply the result in Lemma 4.2.1.

We now define our representation by restricting these new functions to Γ and then extending them as discrete harmonic functions.

$$\tilde{u}^H|_\Gamma = u^H|_\Gamma, \quad \tilde{u}^h_{F_i}|_\Gamma = u^h_{F_i}|_\Gamma,$$

$$\tilde{u}^h_{E_j}|_\Gamma = u^h_{E_j}|_\Gamma, \quad \tilde{u}^h_{V_k}|_\Gamma = u^h_{V_k}|_\Gamma.$$

The definition of discrete harmonic as the minimizing extension then gives us the needed bound. We then apply Lemma 2.3.3 to conclude the proof. ∎

We note that by applying the techniques of Chapter 2, we can show that the multiplicative version of this newly proposed scheme also converges independently of h and H for the equations of linear elasticity. This follows because the necessary estimates for the multiplicative and additive versions are identical, see Chapter 2.

We now develop a partitioning result for finite elements in two and three dimensions which is needed for several of our proofs.

For two or three dimensions let Ω be a polyhedral (polygonal) domain which has been triangulated into substructures which are shape regular, with diameter $O(H)$. Continue the triangulation to obtain a triangulation with elements of diameter $O(h)$. Furthermore assume that Ω has been covered with N shape regular overlapping regions Ω_i, (not necessarily related to the coarse triangulation above) each with a diameter $O(H)$, each of which overlaps all its neighbors with an overlap of $O(H)$. Let $V^H(\Omega) \subset H^1_0(\Omega)$ and $V^h(\Omega) \subset H^1_0(\Omega)$ be the spaces of continuous, piecewise linear functions, on the two triangulations, which vanish on the boundary $\partial\Omega$. We then construct the following spaces

$$V^h_0 = V^H, \quad V^h_i = V^h \cap H^1_0(\Omega_i).$$

The following theorem is a variation of a result given in Dryja and Widlund [36].

Lemma 4.2.1 *For all $u^h \in V^h$ there exists $u_i^h \in V_i^h$ with $u^h = \sum_{i=0}^N u_i^h$ such that*

$$\sum_{i=0}^{N} |u_i^h|^2_{H^1(\Omega)} \leq C_0^2 |u^h|^2_{H^1(\Omega)},$$

where C_0 is independent of u^h, h and H.

Proof. From Strang [80], we know that there exists a linear map $\hat{I}_H : V^h \to V^H$ which satisfies

$$||u^h - \hat{I}_H u^h||^2_{L_2(\Omega)} \leq CH^2 |u^h|^2_{H^1(\Omega)} \tag{4.2}$$

and

$$|u^h - \hat{I}_H u^h|^2_{H^1(\Omega)} \leq C|u^h|^2_{H^1(\Omega)}. \tag{4.3}$$

We then define

$$u_0^h = \hat{I}_H u^h, \quad w^h = u^h - u_0^h$$

and

$$u_i^h = I_h(\theta_i w^h).$$

I_h is the linear interpolation operator onto the space V^h and the θ_i form a partition of unity with $\theta_i \in C_0^\infty(\Omega_i), 0 \leq \theta_i \leq 1$ and $\sum_{i=1}^N \theta_i = 1$. Since I_h is a linear operator, it is immediate that

$$u^h = \sum_{i=0}^{N} u_i^h.$$

Because of the generous overlap between subregions, we can insure that the gradients of θ_i are well behaved. That is, the θ_i can be constructed so that their gradients satisfy $|\nabla \theta_i|^2_{L^\infty} \leq C/H^2$. If we let K represent any single element in the triangulation this implies that

$$||\theta_i - \bar{\theta}_i||^2_{L^\infty(K)} \leq C(h/H)^2. \tag{4.4}$$

Here $\bar{\theta}_i$ is the average of θ_i on element K.

We now estimate the H^1 norm of u_i^h over a single element.

$$\begin{aligned} |u_i^h|^2_{H^1(K)} &= |I_h(\bar{\theta}_i w^h + (\theta_i - \bar{\theta}_i) w^h)|^2_{H^1(K)} \\ &\leq 2|\bar{\theta}_i w^h|^2_{H^1(K)} + 2|I_h(\theta_i - \bar{\theta}_i)w^h)|^2_{H^1(K)}, \end{aligned}$$

which can be bounded using an inverse inequality by

$$|u_i^h|^2_{H^1(K)} \leq 2|\bar{\theta}_i w^h|^2_{H^1(K)} + Ch^{-2}||I_h(\theta_i - \bar{\theta}_i)w^h)||^2_{L^2(K)}.$$

We now use equation (4.4) and the trivial inequality $||\bar{\theta}_i||_{L^\infty} \leq 1$ to obtain

$$|u_i^h|^2_{H^1(K)} \;\leq\; 2|w^h|^2_{H^1(K)} + CH^{-2}||w^h||^2_{L^2(K)}.$$

Since a finite bounded number of u_i^h are nonzero for any element K, we obtain, when summing over i,

$$\sum_{i=1}^{N} |u_i^h|^2_{H^1(K)} \leq C|w^h|^2_{H^1(K)} + CH^{-2}||w^h||^2_{L^2(K)}.$$

Next sum over the elements K,

$$\sum_{i=1}^{N} |u_i^h|^2_{H^1(\Omega)} \leq C|w^h|^2_{H^1(\Omega)} + CH^{-2}||w^h||^2_{L^2(\Omega)}.$$

To finish the argument, we use equations (4.2) and (4.3) to obtain

$$\sum_{i=0}^{N} |u_i^h|^2_{H^1(\Omega)} \leq C_0^2|u^h|^2_{H^1(\Omega)}.$$

∎

Theorem 4.2.2 *The classical additive Schwarz scheme, cf. Dryja and Widlund, [34],[35],[36], has a condition number bounded independently of h and H for the equations of linear elasticity.*

Proof. We note that no node on Ω is contained in more than a fixed number of the overlapping regions. Therefore no element of the space V^h can belong to more than that fixed number of subspaces; hence by Lemma 2.3.1 we have $\lambda_{\max}(P) \leq C$.

We next use the uniform ellipticity of our bilinear form $a_\Omega(u,u)$,

$$c|u|^2_{H^1(\Omega)} \leq a_\Omega(u,u) \leq C|u|^2_{H^1(\Omega)},$$

to make it possible to work in the $(H^1(\Omega))^q$ norm. To complete the proof we use the partitioning result of Lemma 4.2.1 and Lemma 2.3.3. ∎

We note that by applying the techniques of Chapter 2, we can show that the multiplicative version of the classical additive Schwarz scheme of Dryja and Widlund also converges at a rate independent of h and H for the equations of linear elasticity.

4.3 Numerical Experiments in Two Dimensions

We have performed numerous experiments with problems in two dimensions. The problems considered are

- The Laplacian using the usual five point stencil;
- The equations of linear elasticity using 4 node square membrane elements with 2 degrees of freedom per node; cf. [4];
- The equations of linear elasticity using 4 node square shell elements with 3 degrees of freedom per node; cf. [4].

Experiments have been performed on square and L-shaped regions; since there was no appreciable difference between the two cases results are only given for the square regions. The substructures are squares. For the elasticity problems the stiffness matrices were generated using the SESAM code [4], a large, reliable commercial structural analysis code, using a Poisson ratio of .3.

The experiments were run twice, once using all the subspaces as indicated in the algorithm and once excluding the 'vertex' spaces. As expected, the condition number remains bounded by a constant independent of H and h when all the spaces were included. When the 'vertex' spaces were excluded the condition number appears to grow like $(1 + \log(H/h))^2$, also as expected.

The selection of an appropriate stopping condition for the preconditioned conjugate gradient method is crucial. A stopping criterion based only on a norm of the residual can make comparisons between preconditioned and unpreconditioned results misleading since the eigenvalues of the two operators can be of completely different orders of magnitude. For instance, for elasticity problems the eigenvalues of the original stiffness matrices can be of order 10^{12} while the eigenvalues of the preconditioned problems generally are of order 1. We have therefore chosen to use the stopping condition

$$||\text{residual}||_{L^2} \leq \epsilon ||\text{approx. solution}||_{L^2} \lambda_{\min}(\hat{S}^{-1}S).$$

$\lambda_{\min}(\hat{S}^{-1}S)$ is calculated using the Lanczos method at very little extra expense. We have chosen to use $\epsilon = 10^{-5}$; this assures that roughly five digits of the solution are correct and not many more, regardless of the preconditioner used.

Number	Nodes	Number of	No		Without		With	
of	along	Unknowns	Preconditioner		‘Vertex’		‘Vertex’	
Subdomains	Edge	on Γ			Spaces		Spaces	
16	3	81	35.26	14	4.92	7	2.45	6
	7	177	75.10	24	7.59	9	2.55	7
	15	369	155	37	10.89	10	2.82	7
	31	753	315	51	14.84	11	2.99	7
64	3	385	137	32	5.35	9	2.60	8
	7	833	290	49	8.19	10	2.68	8
	15	1729	598	70	11.54	12	2.78	8
	31	3521	1216	*	15.62	13	2.87	8
256	3	1665	545	62	5.46	9	2.63	8
	7	3585	1151	91	8.27	10	2.70	7
	15	7425	2372	*	11.77	12	2.80	7
	31	15105	4766	*	15.90	13	2.89	7

Table 4.1: Condition Numbers and Iteration Counts for the Laplacian

Overlap in nodes	0	1	2	3	4	5	6	7	8
Condition number	15.62	4.49	4.01	3.78	3.52	3.38	3.01	2.92	2.81
Iterations	13	8	8	8	8	8	8	8	8

Table 4.2: Condition Number as Function of Overlap for the Laplacian

	No Preconditioning		Without ‘Vertex’		With ‘Vertex’	
Iter.			Spaces		Spaces	
i	$\|\|e^i\|\|_{L^2}$	$(\frac{\|\|e^i\|\|_{L^2}}{\|\|e^0\|\|_{L^2}})^{1/i}$	$\|\|e^i\|\|_{L^2}$	$(\frac{\|\|e^i\|\|_{L^2}}{\|\|e^0\|\|_{L^2}})^{1/i}$	$\|\|e^i\|\|_{L^2}$	$(\frac{\|\|e^i\|\|_{L^2}}{\|\|e^0\|\|_{L^2}})^{1/i}$
1	9.8×10^{-1}	.99	3.0×10^{-2}	.03	3.0×10^{-2}	.03
2	9.6×10^{-1}	.98	9.9×10^{-3}	.10	1.0×10^{-2}	.10
3	9.5×10^{-1}	.98	2.2×10^{-3}	.13	4.1×10^{-3}	.16
4	9.3×10^{-1}	.98	1.2×10^{-3}	.19	9.0×10^{-4}	.17
5	9.2×10^{-1}	.98	6.3×10^{-4}	.23	7.0×10^{-5}	.15
6	9.1×10^{-1}	.98	4.5×10^{-4}	.28	2.5×10^{-5}	.17
7	8.9×10^{-1}	.98	1.0×10^{-4}	.27	9.4×10^{-6}	.19
8	8.8×10^{-1}	.98	3.6×10^{-5}	.28	3.3×10^{-6}	.21
9	8.6×10^{-1}	.98	9.5×10^{-6}	.28	3.8×10^{-7}	.19
10	8.4×10^{-1}	.98	6.3×10^{-6}	.30	6.7×10^{-8}	.19
11	8.2×10^{-1}	.98	7.7×10^{-6}	.34	2.4×10^{-8}	.20
12	7.9×10^{-1}	.98	3.0×10^{-6}	.35	5.9×10^{-9}	.21

Table 4.3: Errors and Convergence Rates for Laplacian

Number of Subdomains	Nodes along Edge	Number of Unknowns on Γ	No Preconditioner		Without 'Vertex' Spaces		With 'Vertex' Spaces	
16	3	162	22.16	19	10.52	15	3.51	10
	7	354	46.63	28	14.84	17	3.51	10
	15	738	96.34	41	19.83	19	3.56	10
	31	1506	196	60	25.51	20	3.62	10
64	3	770	84.82	37	12.13	18	3.85	10
	7	1666	178	55	16.92	19	3.85	10
	15	3458	368	79	22.37	22	3.84	10
	31	7042	747	*	28.52	25	3.89	10
256	3	3330	334	75	12.42	18	3.91	10
	7	7170	705	*	17.31	19	3.90	10
	15	14850	1453	*	22.88	22	3.89	10
	31	30210	2921	*	29.15	25	3.94	10

Table 4.4: Condition Numbers and Iteration Counts for Membrane Elements

Overlap in nodes	0	1	2	3	4	5	6	7	8
Condition number	28.55	5.59	4.85	4.36	4.01	3.89	3.88	3.89	3.89
Iterations	25	12	11	10	10	10	10	10	10

Table 4.5: Condition Number as Function of Overlap for Membrane Elements

Iter.	No Preconditioning		Without 'Vertex' Spaces		With 'Vertex' Spaces	
i	$\|\|e^i\|\|_{L^2}$	$(\frac{\|\|e^i\|\|_{L^2}}{\|\|e^0\|\|_{L^2}})^{1/i}$	$\|\|e^i\|\|_{L^2}$	$(\frac{\|\|e^i\|\|_{L^2}}{\|\|e^0\|\|_{L^2}})^{1/i}$	$\|\|e^i\|\|_{L^2}$	$(\frac{\|\|e^i\|\|_{L^2}}{\|\|e^0\|\|_{L^2}})^{1/i}$
1	9.7×10^{-1}	.98	8.0×10^{-2}	.08	8.7×10^{-2}	.09
2	9.5×10^{-1}	.98	3.7×10^{-2}	.19	4.2×10^{-2}	.20
3	9.3×10^{-1}	.98	1.1×10^{-2}	.22	1.6×10^{-2}	.25
4	9.2×10^{-1}	.98	6.3×10^{-3}	.28	3.9×10^{-3}	.25
5	9.0×10^{-1}	.98	3.7×10^{-3}	.33	1.1×10^{-3}	.26
6	8.8×10^{-1}	.98	3.4×10^{-3}	.39	5.1×10^{-4}	.28
7	8.6×10^{-1}	.98	2.0×10^{-3}	.41	1.6×10^{-4}	.29
8	8.3×10^{-1}	.98	1.4×10^{-3}	.44	4.4×10^{-5}	.29
9	8.1×10^{-1}	.98	5.1×10^{-4}	.43	1.7×10^{-5}	.30
10	7.8×10^{-1}	.98	2.3×10^{-4}	.43	5.4×10^{-6}	.30
11	7.5×10^{-1}	.97	2.5×10^{-4}	.47	1.5×10^{-6}	.30
12	7.2×10^{-1}	.97	2.6×10^{-4}	.50	5.1×10^{-7}	.30

Table 4.6: Errors and Convergence Rates for Membrane Elements

Number of Subdomains	Nodes along Edge	Number of Unknowns on Γ	No Preconditioner		Without 'Vertex' Spaces		With 'Vertex' Spaces	
16	3	243	452	47	10.51	17	3.48	10
	7	531	442	66	14.83	18	3.48	10
	15	1107	970	101	19.81	21	3.53	10
	31	2259	*	*	25.48	23	3.60	10
64	3	1155	1751	118	11.99	18	3.72	10
	7	2499	1707	128	15.93	19	3.74	10
	15	5187	3800	193	22.12	23	3.83	10
	31	10563	*	*	28.24	26	3.88	10
256	3	4995	*	239	11.63	17	3.73	10
	7	10755	*	246	16.19	19	3.81	10
	15	22275	*	365	22.58	22	3.85	10

Table 4.7: Condition Numbers and Iteration Counts for Shell Elements

Overlap in nodes	0	1	2	3	4	5	6	7	8
Condition number	28.24	5.62	4.81	4.33	3.98	3.87	3.87	3.88	3.88
Iterations	23	12	11	10	10	10	10	10	10

Table 4.8: Condition Number as Function of Overlap for Shell Elements

Iter.	No Preconditioning		Without 'Vertex' Spaces		With 'Vertex' Spaces	
i	$\|\|e^i\|\|_{L^2}$	$(\frac{\|\|e^i\|\|_{L^2}}{\|\|e^0\|\|_{L^2}})^{1/i}$	$\|\|e^i\|\|_{L^2}$	$(\frac{\|\|e^i\|\|_{L^2}}{\|\|e^0\|\|_{L^2}})^{1/i}$	$\|\|e^i\|\|_{L^2}$	$(\frac{\|\|e^i\|\|_{L^2}}{\|\|e^0\|\|_{L^2}})^{1/i}$
1	*	*	6.8×10^{-2}	.07	7.4×10^{-2}	.07
2	*	*	3.2×10^{-2}	.18	3.6×10^{-2}	.19
3	*	*	9.4×10^{-3}	.21	1.4×10^{-2}	.24
4	*	*	5.3×10^{-3}	.27	3.4×10^{-3}	.24
5	*	*	3.1×10^{-3}	.32	1.0×10^{-3}	.25
6	*	*	3.1×10^{-3}	.38	4.7×10^{-4}	.28
7	*	*	1.9×10^{-3}	.41	1.4×10^{-4}	.28
8	*	*	1.4×10^{-3}	.44	3.9×10^{-5}	.28
9	*	*	5.5×10^{-4}	.43	1.6×10^{-5}	.29
10	*	*	3.3×10^{-4}	.45	4.9×10^{-6}	.29
11	*	*	2.5×10^{-4}	.47	1.3×10^{-6}	.29
12	*	*	2.5×10^{-4}	.50	4.6×10^{-7}	.30
13	*	*	2.4×10^{-4}	.53	1.5×10^{-7}	.30

Table 4.9: Errors and Convergence Rates for Shell Elements

In Table 4.1 the experiments are conducted for the Laplacian. The overlap of the 'vertex' spaces onto the 'edge' spaces is chosen to be $H/4$. In Table 4.2, we examine the effect of varying the amount of overlap of the 'vertex' spaces onto the 'edge' spaces for the case with 64 substructures and 31 nodes along the edge of each substructure. We see that the overlap is very important but a small overlap has almost as much effect as a larger overlap. We give a sample of the convergence behavior in Table 4.3, showing the discrete L^2 norm of the error as a function of the number of iterations. This is again for the case of 64 substructures and 31 nodes along the edge of each substructure. These tables are repeated for the linear elasticity problems.

For completeness, we give a sample of the cpu times on a Sun Sparcstation and Convex supermini computer. The times are only for the conjugate gradient portion of the algorithm and do not include the calculation of the stiffness matrices nor the time for the backsolves onto the interiors of the substructures. We note that the preconditioned problems are solved much faster than the unpreconditioned problems. See Table 4.10.

4.3.1 Effect of Aspect Ratios

The aspect ratios of substructures can strongly affect the conditioning of the preconditioned problem. An ideal preconditioner would generate condition numbers independent of the aspect ratio of the substructures.

We have run experiments on elasticity models which have substructures with large aspect ratios. The first set of experiments involve using rectangular substructures with square elements, hence there are a different number of elements along the different edges of the substructures, see Table 4.11. Also included in the same table are results which use a nonuniform mesh. All the results are for the case with 256 subdomains.

4.3.2 Effect of the Poisson Modulus

For certain two dimensional elasticity models, i.e. membranes and shells, the 'locking' phenomenon occurs when ν is 1 rather than .5. We examine the effect of this locking on the preconditioning.

Number	Nodes	Number of	No	Without	With
of	along	Unknowns	Preconditioner	‘Vertex’	‘Vertex’
Subdomains	Edge	on Γ		Spaces	Spaces
Timings on Sun Sparcstation					
16	3	243	7.03	2.38	2.04
	7	531	23.98	7.20	5.43
	15	1107	119.67	28.71	18.63
64	3	1155	73.16	7.23	4.93
	7	2499	181	7.23	4.93
	15	5187	945	115	59.69
256	3	4995	586	27.26	19.26
	7	10755	1394	103	62.46
	15	22275	7332	449	232
Timings on Convex (Vectorized only)					
16	3	243	1.73	.94	.75
	7	531	5.73	2.36	1.81
	15	1107	26.79	8.29	5.77
	31	2259	200	30.80	19.75
64	3	1155	16.72	3.02	2.21
	7	2499	39.38	8.13	5.68
	15	5187	196	30.41	16.99
	31	10563	1502	117	56.75
256	3	4995	131	11.76	9.54
	7	10755	301	32.93	22.61
	15	22275	1469	117	65.63

Table 4.10: Timings for Shell Problem (in seconds of CPU time)

Nodes	Number of	No		Without		With	
along	Unknowns	Preconditioner		‘Vertex’		‘Vertex’	
Edge	on Γ			Spaces		Spaces	
		Aspect Ratio 1 to 2					
3 by 7	5250	580	120	28.19	30	9.11	17
7 by 15	11010	1179	174	36.72	32	9.02	17
15 by 31	22530	2071	249	46.03	35	8.95	16
		Aspect Ratio 1 to 4					
3 by 15	9090	714	145	134	59	58.84	39
7 by 31	18690	1435	208	170	65	59.06	38
3 by 3	3330	1003	172	62.51	37	48.98	34
7 by 7	7170	2123	251	99.51	48	55.81	38
15 by 15	14850	4380	364	138	56	57.98	38
31 by 31	22906	8554	525	175	65	58.58	38
		Aspect Ratio 1 to 8					
3 by 31	16770	736	157	627	115	380	90

Table 4.11: Condition Numbers and Iteration Counts for Membrane Elements

Effect of Poisson Modulus

Poisson Modulus	Number of Unknowns on Γ	No Preconditioner		Without 'Vertex' Spaces		With 'Vertex' Spaces	
16 Subdomains							
.3	738	96.34	41	19.83	19	3.56	10
.5		98.54	45	23.95	21	4.18	11
.95		226	67	195	41	25.86	20
.99		670	96	954	57	79.80	27
64 Subdomains							
.3	3458	368	79	22.37	22	3.84	10
.5		374	87	28.41	24	4.70	12
.95		829	139	261	77	33.15	29
.99		2555	222	1334	147	101	49
256 Subdomains							
.3	14850	1453	*	22.88	22	3.89	10
.5		1475	173	29.39	24	4.84	12
.95		3250	270	281	96	34.61	34
.99		10126	451	1453	196	107	58

Table 4.12: Condition Numbers and Iteration Counts for Membrane Elements

Using membrane elements with an H/h of 16, we calculate the condition numbers and iteration counts as ν approaches 1. The numerical results are listed in Table 4.12.

Chapter 5

Iterative Substructuring Methods for Three Dimensions

5.1 Introduction

In this chapter we introduce a new iterative substructuring algorithm for problems in three dimensions. We then prove that the condition number of the resulting preconditioned problem is bounded by $C(1+\log(H/h))^2$. In addition, the condition number is independent of the jumps in the coefficients of the differential equation between substructures.

The following Sobolev type inequality holds for finite element functions in two dimensions,

$$||u^h - \alpha||^2_{L^\infty(\Omega_i)} \leq C(1 + \log(H/h))|u^h|^2_{H^1(\Omega_i)}.$$

Here α is any convex combination of the values of u^h in Ω_i. Using this inequality, it can be shown that a coarse mesh interpolant of u^h can have an energy which exceeds that of u^h by at most a factor of $C(1 + \log(H/h))$.

However in three dimensions we only have a much weaker bound

$$||u^h - \alpha||^2_{L^\infty(\Omega_i)} \leq C(H/h)|u^h|^2_{H^1(\Omega_i)}.$$

Therefore, in this case, interpolating the value of u^h from a vertex can result in a $O(H/h)$ change in the energy. See Section 3.2 for a detailed discussion of this point.

It was observed by Bramble, Pasciak, and Schatz [15] that, for three dimensions, there exists a much stronger bound. Let α be any convex combination of values of u^h at the wirebasket nodes, then

$$h||u^h - \alpha||^2_{l^2(W)} \leq C(1 + \log(H/h))|u^h|^2_{H^1(\Omega_i)}. \qquad (5.1)$$

Here W represents the wirebasket on substructure Ω_i. The wirebasket based schemes of Bramble, Pasciak, and Schatz [15] and Dryja [33] are based on calculating the average value of u^h on the wirebasket and then interpolating these values onto the faces of the substructures.

The use of the average value of u^h over the entire wirebasket of the substructure makes it impossible to perform the coarse problem solution in parallel with the local problems. We can see this by noting that the local contribution to the preconditioner can be written in the form of (3.2). However, the action of the of the two transformation matrices $R^{(i)^T}$ generally provide different values on the face, since each $R^{(i)^T}$ also depends on values not available to both substructures. We, therefore, cannot express the global form of the preconditioner as in (3.3). Therefore the coarse problem and face problems cannot be solved in parallel.

5.2 Wirebasket Based Methods

5.2.1 The New Method

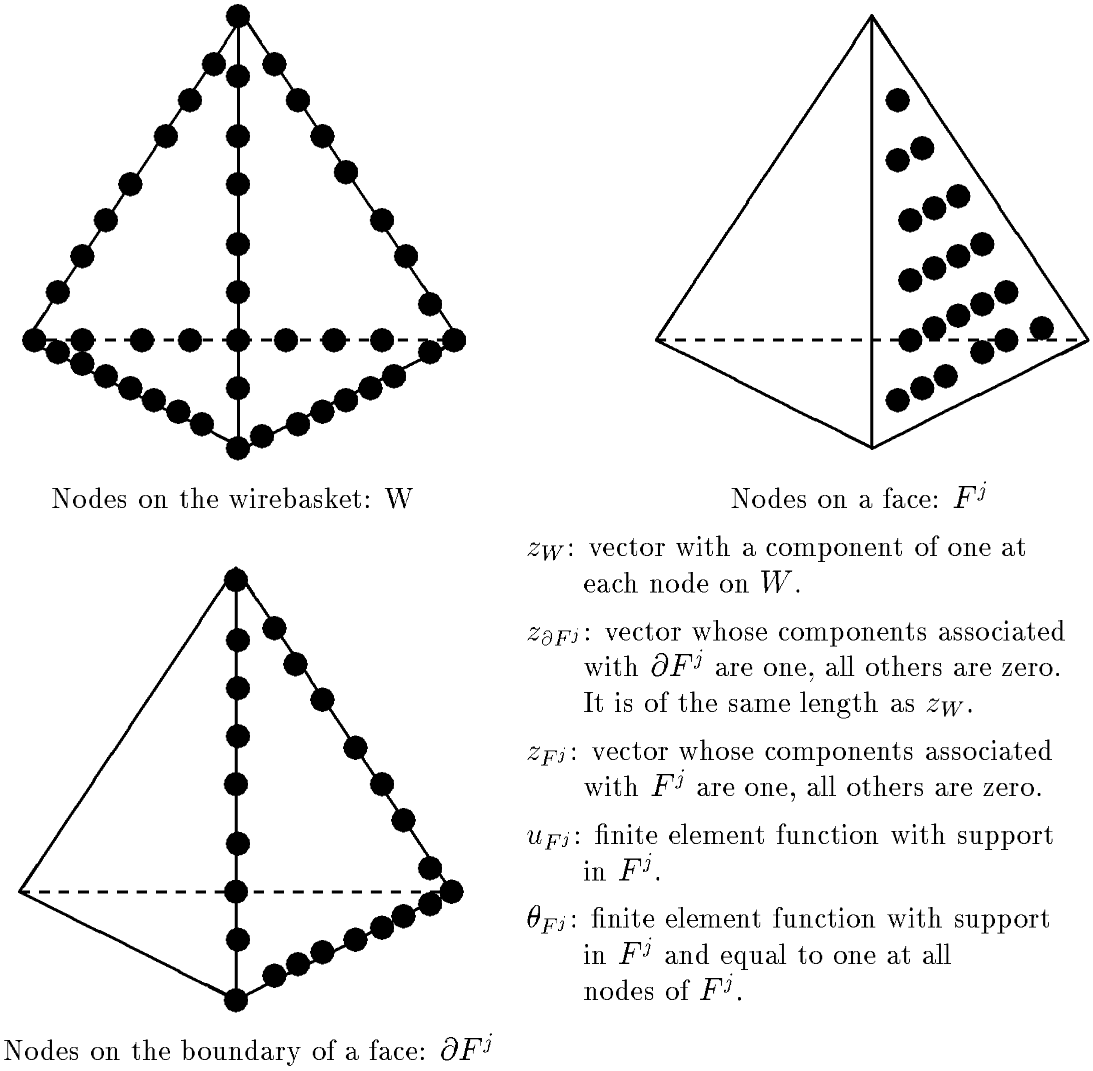
To introduce the new algorithm we observe that the estimate (5.1) remains valid if we replace the entire wirebasket by a line segment of length $O(H)$. In the new algorithm we will interpolate averages for the parts of the wirebasket adjacent to each face, see Fig. 5.1.

We construct the preconditioner one substructure at a time, using a method quite similar to that of Section 3.2, and obtain the preconditioner by subassembly. We first order the nodes on the faces and then the nodes on the wirebasket. The local contribution of the substructure Ω_i to the Schur complement is

$$S^{(i)} = \begin{pmatrix} S^{(i)}_{FF} & S^{(i)}_{FW} \\ S^{(i)^T}_{FW} & S^{(i)}_{WW} \end{pmatrix}.$$

We are solving the reduced, Schur complement problem so the nodes interior to the substructures play no role in the description.

Let $T^{(i)^T}$ map the weighted average of the values of the boundary nodes of each face (the adjacent vertices and edges) to the node of the corresponding face.



Nodes on the wirebasket: W

Nodes on a face: F^j

Nodes on the boundary of a face: ∂F^j

z_W: vector with a component of one at each node on W.

$z_{\partial F^j}$: vector whose components associated with ∂F^j are one, all others are zero. It is of the same length as z_W.

z_{F^j}: vector whose components associated with F^j are one, all others are zero.

u_{F^j}: finite element function with support in F^j.

θ_{F^j}: finite element function with support in F^j and equal to one at all nodes of F^j.

Figure 5.1: Nodal subsets associated with a substructure Ω

Let $\hat{S}_{FF}^{(i)}$ be the block diagonal part of $S_{FF}^{(i)}$ with a block for each face. The local contribution to the preconditioner is then,

$$\hat{S}^{(i)} = \begin{pmatrix} I & 0 \\ -T^{(i)} & I \end{pmatrix} \begin{pmatrix} \hat{S}_{FF}^{(i)} & 0 \\ 0 & \hat{G}^{(i)} \end{pmatrix} \begin{pmatrix} I & -T^{(i)^T} \\ 0 & I \end{pmatrix}.$$

Note that the structure is similar to that of the preconditioner introduced in Section 3.2.

We define $\hat{G}^{(i)}$ by

$$x_W^{(i)^T} \hat{G}^{(i)} x_W^{(i)} = \min_{\bar{w}^{(i)}} (x_W^{(i)} - \bar{w}^{(i)} z_W^{(i)})^T G^{(i)} (x_W^{(i)} - \bar{w}^{(i)} z_W^{(i)}). \tag{5.2}$$

$G^{(i)}$ is given by $\delta(H/h)I$ or $\delta(H/h)$ times a constant (block) diagonal matrix and $z_W^{(i)}$ is a vector of all ones of the same dimension as $x_W^{(i)}$, see Figure 5.1. The optimizing $\bar{w}^{(i)}$ corresponds to the weighted average of the values of x on the wirebasket of substructure Ω_i. The $\hat{G}^{(i)}$ is constructed in this manner to force the $\hat{G}^{(i)}$, and hence $\hat{S}^{(i)}$, to have a null space of the constants. Since this is also the null space for $S^{(i)}$ we immediately obtain that the condition number is independent of the number of subdomains, see Section 2.5. For subdomains with a Dirichlet boundary $\hat{G}^{(i)}$ is simply $G^{(i)}$.

The $\delta(H/h)$, a scalar function of (H/h), is chosen to optimize the scaling between the 'coarse' problem and the 'face' problems. Our results show that when $\delta(H/h) = (1 + \log(H/h))$, the condition number of the preconditioned problem is bounded by $C(1 + \log(H/h))^2$ in three dimensions. On the other hand, if $\delta(H/h)$ is chosen to be a constant then the bound is $C(1 + \log(H/h))^3$.

The global preconditioner is obtained by subassembly,

$$\hat{S} = \begin{pmatrix} I & 0 \\ -T & I \end{pmatrix} \begin{pmatrix} \hat{S}_{FF} & 0 \\ 0 & \hat{G} \end{pmatrix} \begin{pmatrix} I & -T^T \\ 0 & I \end{pmatrix}.$$

Here $\hat{S}_{FF}$ and $\hat{G}$ are obtained by subassembly from $\hat{S}_{FF}^{(i)}$ and $\hat{G}^{(i)}$ respectively. As in Section 3.2, we have used the fact that the action of the $T^{(i)^T}$, for two adjacent subdomains along a shared face, are identical. The preconditioner is easily inverted to give,

$$\hat{S}^{-1} = \begin{pmatrix} T^T \\ I \end{pmatrix} \hat{G}^{-1} \begin{pmatrix} T & I \end{pmatrix} + \begin{pmatrix} I \\ 0 \end{pmatrix} \hat{S}_{FF}^{-1} \begin{pmatrix} I & 0 \end{pmatrix}. \tag{5.3}$$

Note that the resulting operator has independent parts associated with each face and a coarse problem. This is not true of the algorithms of Bramble, Pasciak, and Schatz [15]; Dryja and Widlund [37]; or Mandel [59].

To derive an explicit formula for $T^{(i)}$, we need to introduce some more notations. Let $z_{Fj}^{(i)}$ be a vector of the same length as $x_F^{(i)}$ with zeros at all nodes except those associated with face F^j where the coefficients are one. Let $z_{\partial Fj}^{(i)}$ be a vector of the same length as $x_W^{(i)}$ which is zero at all nodes except those on the boundary of the face F^j where the coefficients are one, see Figure 5.1. Then

$$T^{(i)^T} = \sum_j z_{Fj}^{(i)} (z_{\partial Fj}^{(i)^T} G^{(i)} z_{\partial Fj}^{(i)})^{-1} z_{\partial Fj}^{(i)^T} G^{(i)}.$$

We use this complicated notation to make it possible generalize the method to the case of linear elasticity. In that case the $z^{(i)}$ are matrices with columns forming a basis for the null space of $S^{(i)}$ and the $\bar{w}^{(i)}$ are vectors with components for each column of $z^{(i)}$. The notation is essentially borrowed from Mandel [59]. We give the detailed analysis of our algorithm only for the scalar case. However the results for the elasticity problem follow immediately.

This trick of scaling a portion of the preconditioner with $\delta(H/h)$ has been previously used by Dryja [33] for a version of the Neumann-Dirichlet algorithm and has inspired this new result.

5.2.2 Previous Wirebasket Methods

In [59], Mandel, motivated by the desire to obtain condition numbers independent of the number of substructures, proposed the following algorithm for the p-version finite element method. Let $z^{(i)}$ be the vector of all ones of the same length as $x^{(i)}$. The local contribution to the preconditioner is written as

$$x^{(i)^T} \hat{S}^{(i)} x^{(i)} = \min_{\bar{w}^{(i)}} (x^{(i)} - \bar{w}^{(i)} z^{(i)})^T \begin{pmatrix} \hat{S}_{FF}^{(i)} & 0 \\ 0 & G^{(i)} \end{pmatrix} (x^{(i)} - \bar{w}^{(i)} z^{(i)}).$$

$G^{(i)}$ can be given by

$$I \qquad\qquad \text{or} \qquad\qquad \mathrm{diag}(S_{WW}^{(i)})$$

or a block diagonal part of $S_{WW}^{(i)}$. We no longer need a factor $\delta(H/h)$.

The method of Bramble, Pasciak, and Schatz [15], which predates Mandel's algorithm, is essentially a special case of his algorithm. We have chosen to present Mandel's version since it fits more easily into our framework. Mandel's method and those of Bramble, Pasciak, and Schatz [15] are inherently more sequential than

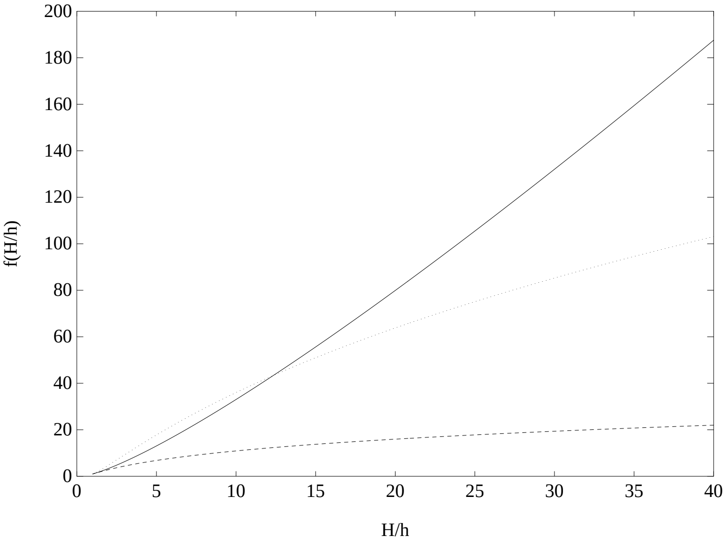


Figure 5.2: Graphs of $(1+\log(H/h))^2$, $(1+\log(H/h))^3$, and $(H/h)(1+\log(H/h))$

the vertex based method and the new method. This is because they all require the determination of an 'average' value for each substructure before the local problems can be solved. This might produce a bottleneck in the parallel implementation of these methods.

In Figure 5.2 we plot the functions $(1+\log(H/h))^2$, $(1+\log(H/h))^3$, and $(H/h)(1+\log(H/h))$ in order to see the relative utility of the asymptotically faster methods even for small values of H/h.

5.2.3 Solving the Coarse Problem

For all of these 'averaging' methods we need to solve a quadratic problem of the form

$$\min_x \sum_i \frac{1}{2}(x^{(i)} - \bar{w}^{(i)} z^{(i)})^T B^{(i)} (x^{(i)} - \bar{w}^{(i)} z^{(i)}) - x^T f,$$

with $\bar{w}^{(i)}$ defined by

$$\arg\min_{\bar{v}} (x^{(i)} - \bar{v} z^{(i)})^T B^{(i)} (x^{(i)} - \bar{v} z^{(i)}).$$

For the new method, this is the system associated with the matrix $\hat{G}$. For the other wirebasket based methods all the variables associated with the preconditioner are involved.

We use the solution technique due to Mandel [59]. We note that Bramble, Pasciak, and Schatz [15] used different tools and provide a technique which appears less generally applicable.

We write the problem as

$$\min_x \sum_i \min_{\bar{w}^{(i)}} \frac{1}{2}(x^{(i)} - \bar{w}^{(i)} z^{(i)})^T B^{(i)} (x^{(i)} - \bar{w}^{(i)} z^{(i)}) - x^T f.$$

We then take derivatives with respect to x and $\bar{w}^{(i)}$. This results in the linear system,

$$z^{(i)^T} B^{(i)} (x^{(i)} - z^{(i)} \bar{w}^{(i)}) = 0, \qquad \forall i,$$

$$Bx - \sum_i B^{(i)} z^{(i)} \bar{w}^{(i)} = f. \tag{5.4}$$

We then eliminate x and get the following system for the $\bar{w}^{(i)}$,

$$(z^{(i)^T} B^{(i)} z^{(i)}) \bar{w}^{(i)} - z^{(i)^T} B^{(i)} B^{-1} \sum_j B^{(j)} z^{(j)} \bar{w}^{(j)} = z^{(i)^T} B^{(i)} B^{-1} f.$$

Once the $\bar{w}^{(i)}$ are known x can be found by solving (5.4).

5.3 Proofs of Almost Optimality

In this section we establish the convergence properties of the new method and Mandel's method. We first give a series of lemmas and then state and prove our results. Many of the techniques are similar to those used in Bramble, Pasciak, and Schatz [15] and Dryja [33]. However, our algorithm is quite different as is some of the analysis.

We first use the approach of Section 3.2 to reduce the calculation of the condition number to a singe subdomain.

Lemma 5.3.1 *The vertex based method, the new method and the other wirebasket based methods satisfy*

$$\kappa(\hat{S}^{-1} S) \le \frac{\max_i \lambda_{\max}(\hat{S}^{(i)+} S^{(i)})}{\min_i \lambda_{\min}(\hat{S}^{(i)+} S^{(i)})}.$$

In the rest of this section $\Omega \subset R^3$ is a substructure of diameter H. We use the weighted H^1 norm,

$$||u||^2_{H^1(\Omega)} = |u|^2_{H^1(\Omega)} + \frac{1}{H^2}||u||^2_{L^2(\Omega)},$$

and the weighted $H^{1/2}$ norm,

$$||u||^2_{H^{1/2}(\Gamma)} = |u|^2_{H^{1/2}(\Gamma)} + \frac{1}{H}||u||^2_{L^2(\Gamma)}.$$

Lemma 5.3.2 *In two dimensions*

$$||u^h||^2_{L^\infty(\bar{\Omega})} \leq C(1 + \log(H/h))||u^h||^2_{H^1(\Omega)}.$$

Proof. See e.g. Bramble, Pasciak, and Schatz [12], or Dryja [33]. ∎

The following lemma is a consequence of Lemma 5.3.2 and is obtained by applying it to two-dimensional slices one at a time. Proofs of the first and second parts are given in Bramble, Pasciak, and Schatz [15] and Dryja [33], respectively.

Lemma 5.3.3 *Let I be a line segment, of length $O(H)$, in Ω. Then*

$$h||u^h||^2_{l^2(I)} \leq C(1 + \log(H/h))||u^h||^2_{H^1(\Omega)}.$$

If, in addition, $\bar{u}^h$ is the average of u^h on I then

$$h||u^h - \bar{u}^h||^2_{l^2(I)} \leq C(1 + \log(H/h))|u^h|^2_{H^1(\Omega)}.$$

The following extension theorem is given in Bramble, Pasciak, and Schatz [15].

Lemma 5.3.4 *Let u^h be a discrete harmonic, piecewise linear, finite element function in Ω. Then there exists a C, independent of h and H, such that*

$$|u^h|_{H^1(\Omega)} \leq C|u^h|_{H^{1/2}(\Gamma)},$$

for all u^h.

Lemma 5.3.5 *Let $\bar{u}^h$ be the average of u^h over an interval I, of length $O(H)$. Then*

$$(\bar{u}^h)^2 \leq C\frac{(1+\log(H/h))}{H}||u^h||^2_{H^1(\Omega)}.$$

In addition,

$$(\bar{u}^h)^2 \leq C\frac{h}{H}||u^h||^2_{l^2(I)}.$$

Proof.

$$\begin{aligned}(\bar{u}^h)^2 &\leq Ch^2H^{-2}(\sum_{i\in I}u^h(x_i))^2\\ &\leq Ch^2H^{-2}h^{-1}H\sum_{i\in I}u^h(x_i)^2\\ &= ChH^{-1}||u^h||^2_{l^2(I)}\\ &\leq C\frac{(1+\log(H/h))}{H}||u^h||^2_{H^1(\Omega)}.\end{aligned}$$

■

Let Γ be the boundary of Ω, and $F^i \subset \Gamma$ be a face. Let θ_{F^i} be the finite element function which is one at the interior nodes of F^i and zero at all of the other nodes on Γ.

Lemma 5.3.6 *For θ defined above*

$$|\theta_{F^i}|^2_{H^{1/2}_{00}(F^i)} \leq C(1+\log(H/h))H.$$

Proof. This is an intermediate result in the proof of Lemma 5.3.7. ■

Lemma 5.3.7 *Let $u^h_{F^i}$ be the finite element function which is equal to u^h on the interior nodes of F^i but zero on ∂F^i, then*

$$|u^h_{F^i}|^2_{H^{1/2}_{00}(F^i)} \leq C(1+\log(H/h))^2||u^h||^2_{H^{1/2}(\Gamma)}.$$

Proof. This is Lemma 4.3 in Bramble, Pasciak, and Schatz [15]. ■

Lemma 5.3.8 *Let I be a line segment in Ω of length $O(H)$ and let u_W^h be the finite element function which equal to u^h on I but which vanishes at all nodes not on I. Then*

$$|u_W^h|_{H^{1/2}(\Gamma)}^2 \le Ch||u^h||_{l^2(I)}^2.$$

Proof. This is Lemma 4.1 in Bramble, Pasciak, and Schatz [15]. ∎

Lemma 5.3.9 *Let $\bar{u}^h$ be the average of u^h on an interval I, of length $O(H)$. Then*

$$|u_{F^i}^h - \bar{u}^h \theta_{F^i}|_{H_{00}^{1/2}(F^i)}^2 \le C(1+\log(H/h))^2 |u^h|_{H^{1/2}(\Gamma)}^2.$$

Proof. We use Lemma 5.3.7, Lemma 5.3.5, and Lemma 5.3.6 to obtain

$$\begin{aligned} |u_{F^i}^h - \bar{u}^h \theta_{F^i}|_{H_{00}^{1/2}(F^i)}^2 &\le 2|u_{F^i}^h|_{H_{00}^{1/2}(F^i)}^2 + 2|\bar{u}^h \theta_{F^i}|_{H_{00}^{1/2}(F^i)}^2 \\ &\le C(1+\log(H/h))^2 ||u^h||_{H^1(\Omega)}^2. \end{aligned}$$

To conclude the argument, we note that the left hand side is unchanged if a constant is added to u^h. We therefore shift u^h by a constant so that it's average is zero. We can then replace the norm on the right hand side with a semi-norm using a Poincaré inequality. We return to the $H^{1/2}$ semi-norm using Lemma 5.3.4. ∎

The main result for the new algorithm is given in

Theorem 5.3.1 *For the new wirebasket method using $G^{(i)} = (1+\log(H/h))I$, (i.e. $\delta(H/h) = (1+\log(H/h))$), the condition number is bounded by*

$$\kappa(\hat{S}^{-1}S) \le C(1+\log(H/h))^2.$$

Moreover if $G^{(i)} = I$ then

$$\kappa(\hat{S}^{-1}S) \le C(1+\log(H/h))^3.$$

Proof. We prove the first result; the second result follows easily. We will establish

$$\frac{x^{(i)^T}\hat{S}^{(i)}x^{(i)}}{c(1+\log(H/h))^2} \le x^{(i)^T}S^{(i)}x^{(i)} \le Cx^{(i)^T}\hat{S}^{(i)}x^{(i)}. \tag{5.5}$$

We prove the result for an interior subdomain; a similar proof holds for subdomains on the boundary. We need the following definitions. Let $\bar{w}$ be the average value of u^h on the nodes of the wirebasket and $\bar{w}^{\partial F^i}$ be the average value of u^h on ∂F^i. Recall that $u^h_{F^i}$ is the finite element function which is equal to u^h at the interior nodes of face F^i, but is zero on ∂F^i. $W = \cup \partial F^i$ is the union of the line segments which make up the wirebasket and u^h_W is the finite element function which is equal to u^h at the nodes of W, but is zero at all of the other nodes of Γ. Using these notations the lower bound of (5.5) is equivalent to

$$\delta(H/h)h||u^h - \bar{w}||^2_{l^2(W)} + \sum_i |u^h_{F^i} - \bar{w}^{\partial F^i}\theta_{F^i}|^2_{H^{1/2}_{00}(F^i)} \leq c(1+\log(H/h))^2|u^h|^2_{H^{1/2}(\Gamma)}.$$

We bound each of the left hand terms separately.

$$\begin{aligned}\delta(H/h)h||u^h - \bar{w}||^2_{l^2(W)} &\leq c\delta(H/h)(1+\log(H/h))|u^h|^2_{H^1(\Omega)} \\ &\leq c(1+\log(H/h))^2|u^h|^2_{H^1(\Omega)} \\ &\leq c(1+\log(H/h))^2|u^h|^2_{H^{1/2}(\Gamma)}.\end{aligned}$$

This follows from Lemma 5.3.3 and Lemma 5.3.4. We bound the other terms using Lemma 5.3.9,

$$|u^h_F - \bar{w}^{\partial F^i}\theta_{F^i}|^2_{H^{1/2}_{00}(F^i)} \leq c(1+\log(H/h))^2|u^h|^2_{H^{1/2}(\Gamma)}.$$

We now must prove the upper bound,

$$x^{(i)^T}S^{(i)}x^{(i)} \leq Cx^{(i)^T}\hat{S}^{(i)}x^{(i)}.$$

We shift u^h, by a constant, so that the average of u^h on the wirebasket is zero, i.e. $\bar{w} = 0$. The upper bound is then equivalent to

$$|u^h|^2_{H^{1/2}(\Gamma)} \leq C(\delta(H/h)h||u^h||^2_{l^2(W)} + \sum_i |u^h_{F^i} - \bar{w}^{\partial F^i}\theta_{F^i}|^2_{H^{1/2}_{00}(F^i)}).$$

We show this bound by,

$$\begin{aligned}|u^h|^2_{H^{1/2}(\Gamma)} &= |\sum_i(u^h_{F^i} - \bar{w}^{\partial F^i}\theta_{F^i}) + u^h_W + \sum_i \bar{w}^{\partial F^i}\theta_{F^i}|^2_{H^{1/2}(\Gamma)} \\ &\leq C(\sum_i |u^h_{F^i} - \bar{w}^{\partial F^i}\theta_{F^i}|^2_{H^{1/2}(\Gamma)} + |u^h_W|^2_{H^{1/2}(\Gamma)} + \sum_i |\bar{w}^{\partial F^i}\theta_{F^i}|^2_{H^{1/2}(\Gamma)}) \\ &\leq C(\sum_i |u^h_{F^i} - \bar{w}^{\partial F^i}\theta_{F^i}|^2_{H^{1/2}_{00}(F^i)} + h||u^h||^2_{l^2(W)} + (1+\log(H/h))h||u^h||^2_{l^2(W)}) \\ &\leq C(\sum_i |u^h_{F^i} - \bar{w}^{\partial F^i}\theta_{F^i}|^2_{H^{1/2}_{00}(F^i)} + \delta(H/h)h||u^h||^2_{l^2(W)}).\end{aligned}$$

The bound follows from Lemma 5.3.8, Lemma 5.3.5 and Lemma 5.3.6. ∎

We conclude by proving the result for the previous wirebasket method. Note that this is essentially the result in Bramble, Pasciak, and Schatz [15].

Theorem 5.3.2 *For the previous wirebasket method, using $G^{(i)} = I$, the condition number is bounded by*

$$\kappa(\hat{S}^{-1}S) \leq C(1+\log(H/h))^2.$$

Proof. We will prove the result

$$\frac{x^{(i)^T}\hat{S}^{(i)}x^{(i)}}{c(1+\log(H/h))^2} \leq x^{(i)^T}S^{(i)}x^{(i)} \leq Cx^{(i)^T}\hat{S}^{(i)}x^{(i)}.$$

We shift u^h by a constant so that its weighted average is zero. Using the definitions given in the previous theorem the lower bound follows from

$$h||u^h||^2_{l^2(W)} + \sum_i |u^h_{F^i}|^2_{H^{1/2}_{00}(F^i)} \leq c(1+\log(H/h))^2|u^h|^2_{H^{1/2}(\Gamma)}.$$

This inequality follows from Lemma 5.3.3, Lemma 5.3.7, Lemma 5.3.4 and a quotient space argument.

Lemma 5.3.8 implies the upper bound.

$$\begin{aligned}
|u^h|^2_{H^{1/2}(\Gamma)} &= |\sum_i u^h_{F^i} + u^h_W|^2_{H^{1/2}(\Gamma)} \\
&\leq C(\sum_i |u^h_{F^i}|^2_{H^{1/2}(\Gamma)} + |u^h_W|^2_{H^{1/2}(\Gamma)}) \\
&\leq C(\sum_i |u^h_{F^i}|^2_{H^{1/2}_{00}(F^i)} + h||u^h||^2_{l^2(W)}).
\end{aligned}$$

∎

5.4 The New Method as an Additive Schwarz Method

The new wirebasket based method can be described in the additive Schwarz framework. This leads to a more complete understanding of the algorithm and its properties. To date, the analysis is somewhat incomplete since we have been unable to construct a partition of u^h and prove our result using Lemma 2.3.3 in the style devised by Dryja and Widlund. However, since we have already given an alternative proof, the additive Schwarz analysis is only needed to put the algorithm in perspective. It should be noted the original inspiration for this new algorithm arose from additive Schwarz analysis.

5.4.1 Two Dimensions

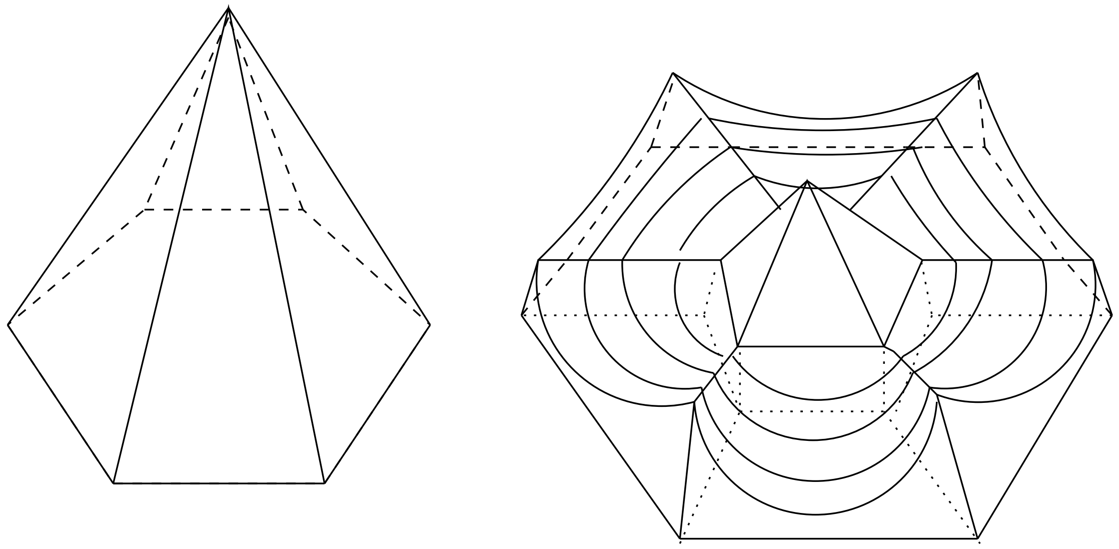
As usual, in the development of the algorithm, we will be working exclusively with the Schur complement rather then the original stiffness matrices. This does not preclude its possible application in other circumstances. In the case of the vertex based method, we work with the usual nodal basis functions of V^H and V^h. In the new wirebasket based method, we work with a different basis for our coarse space and hence a different coarse space. We will refer to this new space as $\hat{V}^H$. Let $\phi_{V_j}^{(i)}$ be the usual nodal basis function of the jth subdomain vertex for the substructure Ω_i and let $\phi_{E_l^k}^{(i)}$ be the kth nodal basis function on the lth edge of the substructure, see Figure 5.4. A basis for our new coarse space is given by

$$\psi_j^{(i)} = \phi_{V_j}^{(i)} + \frac{1}{2}(\sum_k \phi_{E_m^k}^{(i)} + \phi_{E_l^k}^{(i)}).$$

The l and m denote the two subdomain edges adjacent to the jth subdomain vertex. The complete function is obtained by a discrete harmonic extension of this function to the interior of the subdomains. We give a schematic drawing of one of these basis functions in Figure 5.3. Their support resides in a few neighboring subdomains, in fact the same subdomains where the nodal basis functions of V^H have their support. The basic structure of $\hat{V}^H$ is the same whether we work with triangular, rectangular or more general quadrilateral subdomains. This description of the space was inspired by the work of Dryja and Widlund [37].

We now outline how the new wirebasket based method is obtained in the style of an abstract additive Schwarz scheme. We use the subspaces $\hat{V}^H$ and a space for each edge. To make this clear we first look at a single substructure. We first convert the Schur complement into a modified partial hierarchical basis. This procedure is similar to that outlined in Section 3.2. The basis change performed by $T^{(i)}$ is the same as in Section 5.2.1.

$$\begin{pmatrix} I & 0 \\ T^{(i)} & I \end{pmatrix} \begin{pmatrix} S_{EE}^{(i)} & S_{VE}^{(i)} \\ S_{VE}^{(i)^T} & S_{VV}^{(i)} \end{pmatrix} \begin{pmatrix} I & T^{(i)^T} \\ 0 & I \end{pmatrix}.$$



Usual Nodal Basis Function New Nodal Basis Function

Figure 5.3: The New Basis

This gives

$$\begin{pmatrix} S_{EE}^{(i)} & \bar{S}_{VE}^{(i)} \\ \bar{S}_{VE}^{(i)^T} & \tilde{G}^{(i)} \end{pmatrix}.$$

By subassembly the global Schur complement in this new basis is represented by

$$\begin{pmatrix} S_{EE} & \bar{S}_{VE} \\ \bar{S}_{VE}^{T} & \tilde{G} \end{pmatrix}.$$

The preconditioner is then obtained by retaining the block diagonal part of this matrix with a block for each edge and the block $\tilde{G}$. The resulting matrix is then converted back to the original basis. The preconditioned operator then is

$$\begin{pmatrix} I & T^T \\ 0 & I \end{pmatrix} \begin{pmatrix} \hat{S}_{EE}^{-1} & 0 \\ 0 & \tilde{G}^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ T & I \end{pmatrix} \begin{pmatrix} S_{EE} & S_{VE} \\ S_{VE}^{T} & S_{VV} \end{pmatrix}. \qquad (5.6)$$

These transformations are for exposition only and need never be used in computations. Note that equation (5.6) is the sum of projections,

$$P_{\tilde{G}} + \sum_k P_{E^k}.$$

An implementation of this algorithm would require the calculation of $P_{\tilde{G}}$ acting on a vector. Since we do not wish to perform the calculation of $\tilde{G}$ we replace that portion of the preconditioner with an operator which, in two dimensions, is spectrally equivalent to it and for which we have a fast method of evaluation. This operator is the $\hat{G}$ defined in Section 5.2.1. We devote a large part of this section to proving that $\tilde{G}$ and $\hat{G}$, or equivalently $\tilde{G}^{(i)}$ and $\hat{G}^{(i)}$, are spectrally equivalent in two dimensions.

Theorem 5.4.1 *For scalar problems in two dimensions, with piecewise linear finite elements and a uniform triangulation, the operators $\tilde{G}^{(i)}$ and $\hat{G}^{(i)}$ are spectrally equivalent, i.e. there exists c and C independent of h and H so that*

$$cx^T\tilde{G}^{(i)}x \leq x^T\hat{G}^{(i)}x \leq Cx^T\tilde{G}^{(i)}x.$$

Proof. We only give the proof for an interior subdomain; the arguments for boundary subdomains are similar. We also restrict the proof to triangular subdomains; the proof can easily be extended for more general polygonal subdomains. We first demonstrate the result for the model problem and then show that the result can be extended to any scalar, uniformly elliptic operator on more general subdomains.

For the model problem, by simple calculations

$$\tilde{G}^{(i)} = \begin{pmatrix} 1 & -1/2 & -1/2 \\ -1/2 & 1 & -1/2 \\ -1/2 & -1/2 & 1 \end{pmatrix},$$

and

$$\hat{G}^{(i)} = \begin{pmatrix} 2/3 & -1/3 & -1/3 \\ -1/3 & 2/3 & -1/3 \\ -1/3 & -1/3 & 2/3 \end{pmatrix}.$$

These two matrices are clearly spectrally equivalent with constants independent of h and H.

We now generalize to scalar, uniformly elliptic operators, such that

$$c|u|^2_{H^1(\Omega_i)} \leq a_{\Omega_i}(u,u) \leq C|u|^2_{H^1(\Omega_i)}.$$

This inequality also holds for our finite dimensional subspace

$$c|u^h|^2_{H^1(\Omega_i)} \leq a_{\Omega_i}(u^h,u^h) \leq C|u^h|^2_{H^1(\Omega_i)}. \tag{5.7}$$

Again, we are only concerned with a subspace of this space, the space of discrete harmonic functions. Let S to be the Schur complement for our particular problem and let Q to be the Schur complement obtained by using the same triangulation for the Laplacian. It follows from equation (5.7) that

$$cx^TQx \leq x^TSx \leq Cx^TQx.$$

This is true for any basis we work with. We therefore chose to work with the modified partial hierarchical basis defined above. In the next step, we select the principal minors of the matrices S and Q associated with the vertex nodes. This results in

$$cx^T\tilde{Q}x \leq x^T\tilde{G}x \leq Cx^T\tilde{Q}x. \tag{5.8}$$

Here $\tilde{Q}$ corresponds to the definition of $\tilde{G}$ for the Laplacian. Equation (5.8) demonstrates that we need only prove our result for the Laplacian to be assured that it holds for all scalar, uniformly elliptic problems.

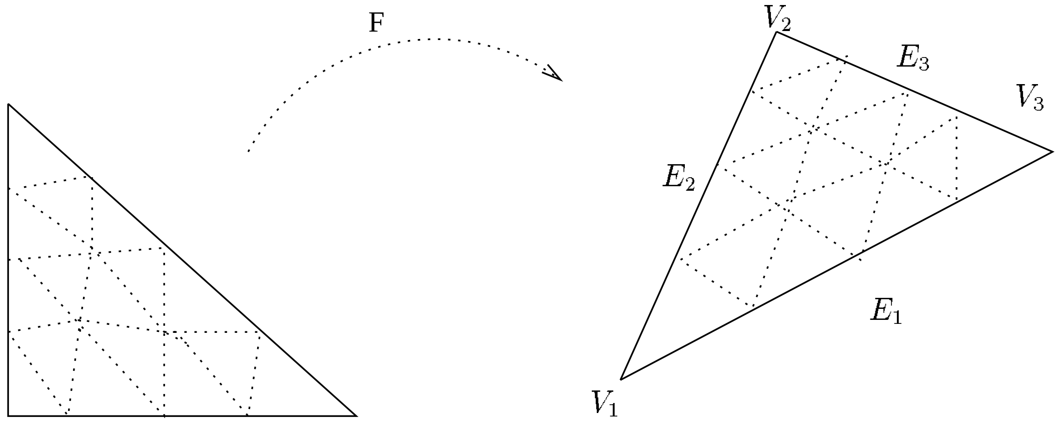
We now show how to extend the result to general triangular subdomains. We assume that there exists an affine mapping from a reference subdomain to our subdomain. In Figure 5.4 we indicate the general mapping from the reference substructure to an actual substructure. For the Laplacian we know that if the Jacobian of the mapping and its inverse are bounded away from zero, then the resulting stiffness matrix is spectrally equivalent to the stiffness matrix for the model subdomain. Their Schur complements will therefore have this same property.

We have thus demonstrated that for a scalar, uniformly elliptic operator in two dimensions on a triangular subdomain whose interior angles are bounded below, i.e. the Jacobian, and its inverse, of the affine map are bounded away from zero, $\tilde{G}$ is spectrally equivalent to $\hat{G}$. ∎

Corollary 5.4.2 *In two dimensions, the new wirebasket based method, with piecewise linear or bilinear elements, has a condition number which is bounded by* $(1 + \log(H/h))^2$ *even if* $\delta(H/h)$ *is a constant.*

Proof. We show that the upper bound is a constant, even if $\delta(H/h)$ is equal to one, using only linear algebra and the structure of the stiffness matrix. This proof is done in the style of additive Schwarz proofs. We are interested in

$$\lambda_{\max}(\hat{S}^{(i)+}S^{(i)}) =$$



Reference Substructure

General Substructure

Figure 5.4: Mapping from Reference Substructure

$$\max_{x\neq 0} \frac{\begin{pmatrix} x_V^{(i)^T} & x_E^{(i)^T}\end{pmatrix}\begin{pmatrix} S_{VV}^{(i)} & S_{VE}^{(i)} \\ S_{VE}^{(i)^T} & S_{EE}^{(i)}\end{pmatrix}\begin{pmatrix}x_V^{(i)}\\x_E^{(i)}\end{pmatrix}}{\begin{pmatrix} x_V^{(i)^T} & x_E^{(i)^T}\end{pmatrix}\begin{pmatrix} I & -T^{(i)}\\0 & I\end{pmatrix}\begin{pmatrix}\hat G^{(i)} & 0\\0&\hat S_{EE}^{(i)}\end{pmatrix}\begin{pmatrix} I & 0\\ -T^{(i)^T} & I\end{pmatrix}\begin{pmatrix}x_V^{(i)}\\x_E^{(i)}\end{pmatrix}} =$$

$$\max_{y\neq 0} \frac{\begin{pmatrix} y_V^{(i)^T} & y_E^{(i)^T}\end{pmatrix}\begin{pmatrix} I & T^{(i)}\\0 & I\end{pmatrix}\begin{pmatrix} S_{VV}^{(i)} & S_{VE}^{(i)} \\ S_{VE}^{(i)^T} & S_{EE}^{(i)}\end{pmatrix}\begin{pmatrix} I & 0\\ T^{(i)^T} & I\end{pmatrix}\begin{pmatrix}y_V^{(i)}\\y_E^{(i)}\end{pmatrix}}{\begin{pmatrix} y_V^{(i)^T} & y_E^{(i)^T}\end{pmatrix}\begin{pmatrix}\hat G^{(i)} & 0\\0&\hat S_{EE}^{(i)}\end{pmatrix}\begin{pmatrix}y_V^{(i)}\\y_E^{(i)}\end{pmatrix}} =$$

$$\max_{y\neq 0} \frac{\begin{pmatrix} y_V^{(i)^T} & y_E^{(i)^T}\end{pmatrix}\begin{pmatrix}\tilde G^{(i)} & \bar S_{VE}^{(i)}\\ \bar S_{VE}^{(i)^T} & S_{EE}^{(i)}\end{pmatrix}\begin{pmatrix}y_V^{(i)}\\y_E^{(i)}\end{pmatrix}}{\begin{pmatrix} y_V^{(i)^T} & y_E^{(i)^T}\end{pmatrix}\begin{pmatrix}\hat G^{(i)} & 0\\0&\hat S_{EE}^{(i)}\end{pmatrix}\begin{pmatrix}y_V^{(i)}\\y_E^{(i)}\end{pmatrix}} =$$

$$\max_{y\neq 0} \frac{\begin{pmatrix} y_V^{(i)^T} & y_E^{(i)^T}\end{pmatrix}\begin{pmatrix}\tilde G^{(i)} & 0\\0&\hat S_{EE}^{(i)}\end{pmatrix}\begin{pmatrix}y_V^{(i)}\\y_E^{(i)}\end{pmatrix}}{\begin{pmatrix} y_V^{(i)^T} & y_E^{(i)^T}\end{pmatrix}\begin{pmatrix}\hat G^{(i)} & 0\\0&\hat S_{EE}^{(i)}\end{pmatrix}\begin{pmatrix}y_V^{(i)}\\y_E^{(i)}\end{pmatrix}} \frac{\begin{pmatrix} y_V^{(i)^T} & y_E^{(i)^T}\end{pmatrix}\begin{pmatrix}\tilde G^{(i)} & \bar S_{VE}^{(i)}\\ \bar S_{VE}^{(i)^T} & S_{EE}^{(i)}\end{pmatrix}\begin{pmatrix}y_V^{(i)}\\y_E^{(i)}\end{pmatrix}}{\begin{pmatrix} y_V^{(i)^T} & y_E^{(i)^T}\end{pmatrix}\begin{pmatrix}\tilde G^{(i)} & 0\\0&\hat S_{EE}^{(i)}\end{pmatrix}\begin{pmatrix}y_V^{(i)}\\y_E^{(i)}\end{pmatrix}}.$$

From Theorem 5.4.1 we know that $\tilde G^{(i)}$ is spectrally equivalent to $\hat G^{(i)}$. From this and the fact that $S_{EE}^{(i)} \leq C\hat{S}_{EE}^{(i)}$, we conclude

$$\begin{aligned}
\lambda_{\max}(\hat{S}^{(i)+}S^{(i)}) &\leq C\lambda_{\max}(\begin{pmatrix} \tilde{G}^{(i)+} & 0 \\ 0 & \hat{S}_{EE}^{(i)^{-1}} \end{pmatrix} \begin{pmatrix} \tilde{G}^{(i)} & \bar{S}_{VE}^{(i)} \\ \bar{S}_{VE}^{(i)T} & S_{EE}^{(i)} \end{pmatrix}) \\
&= C\lambda_{\max}(P_{\tilde{G}}^{(i)} + \sum_E P_E^{(i)}) \\
&\leq 5C
\end{aligned}$$

This upper bound is obtained in the same manner as the upper bound for additive Schwarz methods, by noting that the preconditioned problem is the sum of a fixed number of projections, (one for the 'coarse' part, $P_{\tilde{G}}^{(i)}$, and one for each edge, $P_E^{(i)}$) each with a maximum eigenvalue of one. ∎

One of the great strengths of the additive Schwarz analysis is that it may be used when the preconditioner is obtained from a set of subspaces which do not form a direct sum of the solution space. This allows us to augment a preconditioner by including additional subspaces in regions of the solution space where the preconditioner is less effective. This is the chief trick used in Chapter 4. We therefore attempt to improve the new wirebasket based method by including the additional vertex subspaces introduced in Chapter 4. The numerical results, in a following section, suggest the conjecture.

Conjecture 5.4.1 *In two dimensions, the new wirebasket based method with the additional vertex spaces has a condition number bounded by* $C(1 + \log(H/h))$.

This result is weaker than that obtained in Chapter 4. It would be interesting to prove this result and also to obtain a result of this form for problems in three dimensions.

5.4.2 Three Dimensions

We proceed, as in the previous section, to present the new wirebasket based method as an additive Schwarz method. This section is devoted to its description in three dimensions.

Let β_j be the number of nodes on ∂F^j, the boundary of face number j. We define a basis, $\{\psi_k^{(i)}\}$, for the 'coarse' space in the following manner. For each $\phi_k^{(i)}$ on the wirebasket, there is a corresponding $\psi_k^{(i)}$ given by,

$$\psi_k^{(i)} = \phi_k^{(i)} + \sum_j (\frac{1}{\beta_j} \sum_l \phi_{k,l,j}^{(i)}).$$

The $\phi_{k,l,j}^{(i)}$ are all the nodal basis functions on the faces adjacent to the wirebasket node k. The functions $\psi_k^{(i)}$ are one at the kth node of the wirebasket, a small constant $\frac{1}{\beta_j}$ on each of the adjacent faces and zero at all other wirebasket and face nodes. They are extended as a discrete harmonic function inside the subdomains.

As in two dimensions, we now make a partial change of basis. The preconditioner is obtained by dropping the couplings between the various faces and between the faces and the wirebasket. The preconditioned problem is then the sum of projections,

$$P_{\tilde{G}} + \sum_F P_F.$$

Again, as in two dimensions, the calculation of $P_{\tilde{G}}$ acting on a vector is difficult to perform. We therefore replace $P_{\tilde{G}}$ with $Q_{\hat{G}}$. The explicit form of $P_{\tilde{G}}$ and $Q_{\hat{G}}$ can be written down using notations previously introduced.

$$Q_{\hat{G}} = \begin{pmatrix} T^T \\ I \end{pmatrix} \hat{G}^{-1} \begin{pmatrix} T & I \end{pmatrix} S$$

$$P_{\tilde{G}} = \begin{pmatrix} T^T \\ I \end{pmatrix} \tilde{G}^{-1} \begin{pmatrix} T & I \end{pmatrix} S.$$

The term

$$\begin{pmatrix} T^T \\ I \end{pmatrix} \hat{G}^{-1} \begin{pmatrix} T & I \end{pmatrix}$$

appears in equation (5.3). Note that $P_{\tilde{G}}$ is a projection in the S inner product while $Q_{\hat{G}}$ is not, though it is close to being one. Also note that the $Q_{\hat{G}}$ is scaled by the logarithmic factor $\delta(H/h)$. Such a scaling was not needed in two dimensions.

5.5 A Hierarchical Basis for the New Method

Continuing the tradition of building new domain decomposition algorithms from bits and pieces of previous methods, we use the hierarchical basis domain decomposition method, see Chapter 3, but using, for a coarse problem, the new wirebasket based method proposed in this chapter. At first glance it seems likely that the condition number should grow like $(1+\log(H/h))^4$. However, the growth is probably only proportional to $(1+\log(H/h))^2$.

In Figure 5.5 we plot the local bound on the condition number, as described in Section 2.5, for the model problem using the hierarchical domain decomposition

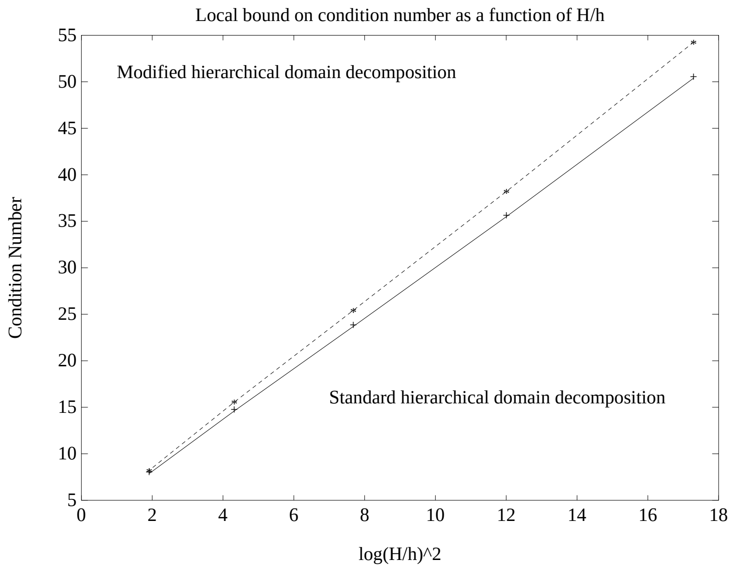


Figure 5.5: Condition Numbers for the Modified Hierarchical DD Algorithm

algorithm as introduced in Chapter 3 and its modification using the new wirebasket based method. We observe that for both methods the condition number appears to grow quadratically in the logarithm of H/h.

Conjecture 5.5.1 *In two dimensions, the hierarchical domain decomposition algorithm with a coarse problem, as borrowed from the new wirebasket based method, has a condition number which is bounded by*

$$C(1 + \log(H/h))^2.$$

Such a result might offer some insight as to how we could build a multilevel preconditioner, based on a wirebasket approach, for problems in three dimensions using techniques similar to those of Chapter 3.

5.6 Numerical Experiments in Two Dimensions

We wish first to confirm, with numerical experiments, the expected theoretical behavior of the algorithms and secondly to get a feel for the relative effectiveness of the various methods. However, we must always bear in mind that superiority in the two dimensional case need not carry over into three dimensions. The vertex based method performs the best in two dimensions but we know theoretically that in three dimensions its behavior is asymptotically worse.

The first set of experiments, reported in Table 5.1, Table 5.2 and Figures 5.6 to 5.11, are for the Laplacian on the unit square. However, due to the construction of these methods the results do not change even if the coefficients of the equation were constant on the substructures but had large jumps between substructures, see Section 2.5. The preconditioner labeled E and V has no global problem. It is block diagonal with a block for each edge and a block for each vertex and is included for comparison purposes. Also included in the tables are the local bounds as described in Section 2.5. The vertex based standard iterative substructing algorithm is indicated in the tables and graphs by V^H. We indicate in the tables the previous wirebasket method as, described by Mandel, with the label Mandel. The stopping condition in determining the iteration counts is a relative decrease of 10^{-5} in the energy norm of the error.

We also ran the same set of experiments after enriching the preconditioner by including the vertex spaces, as suggested in Section 5.4.1, see Table 5.3, Table 5.4 and Figures 5.12 to 5.17. From the graphs, we note the following behavior for the standard vertex based method, Mandel's version of the wirebasket method and the new wirebasket based method.

- All of the methods appear to have a condition number bounded independently of the number of substructures.

- Without the enriching of the vertex spaces all of the methods have condition numbers which grow like $(1 + \log(H/h))^2$.

- With the enrichment represented by the vertex spaces, the condition number of the vertex based method becomes bounded. For the new wirebasket based method and Mandel's version of the earlier wirebasket method the condition number appears to grow like $(1 + \log(H/h))$.

Preconditioner	Number of Substructures					Local
	2 by 2	4 by 4	8 by 8	12 by 12	16 by 16	Bound
			3 Nodes on Γ_{ij}			
V^H	3.11	4.92	5.46	5.37	5.40	5.67
Mandel	6.00	9.00	9.85	10.03	10.12	10.47
New Method	5.60	6.95	7.63	7.76	7.78	7.89
E and V	6.00	18.96	72.10	160.78	284	∞
I	9.77	35.26	137	307	517	∞
Unknowns on Γ	13	81	385	913	1665	
			7 Nodes on Γ_{ij}			
V^H	4.55	7.58	8.30	8.38	8.42	8.60
Mandel	8.60	12.48	13.61	13.84	14.12	14.80
New Method	10.09	12.81	14.13	14.37	14.42	14.62
E and V	8.6	25.59	95.4	212	375	∞
I	21.50	75.1	290	649	1152	∞
Unknowns on Γ	29	177	833	1969	3585	
			15 Nodes on Γ_{ij}			
V^H	6.32	10.55	11.18	11.27	11.29	12.15
Mandel	12.46	16.89	18.20	18.50	18.63	20.16
New Method	16.04	20.67	22.88	23.33	23.40	23.71
E and V	12.46	33.80	122	270	477	∞
I	45	155	599	1338	2372	∞
Unknowns on Γ	61	369	1729	4081	7425	
			31 Nodes on Γ_{ij}			
V^H	8.43	14.46	15.22	15.63	15.89	16.33
Mandel	17.02	21.76	23.14	23.43	23.57	26.08
New Method	23.73	30.86	34.27	34.96	35.03	35.52
E and V	17.02	42.53	149	327	577	∞
I	92	316	1217	2718	4700	∞
Unknowns on Γ	125	753	3521	8305	15105	

Table 5.1: Condition Numbers for Model Problem

Preconditioner	Number of Substructures				
	2 by 2	4 by 4	8 by 8	12 by 12	16 by 16
		3 Nodes on Γ_{ij}			
V^H	4	7	9	9	9
Mandel	4	10	11	13	13
New Method	4	9	12	13	12
E and V	4	10	19	27	35
I	4	14	28	40	53
Unknowns on Γ	13	81	385	913	1665
		7 Nodes on Γ_{ij}			
V^H	5	8	10	10	9
Mandel	5	11	14	17	17
New Method	5	12	15	16	15
E and V	5	10	22	31	40
I	8	22	41	59	77
Unknowns on Γ	29	177	833	1969	3585
		15 Nodes on Γ_{ij}			
V^H	5	9	10	9	9
Mandel	6	12	17	20	20
New Method	6	13	18	18	19
E and V	6	12	25	36	46
I	16	31	60	84	*
Unknowns on Γ	61	369	1729	4081	7425
		31 Nodes on Γ_{ij}			
V^H	5	9	9	9	9
Mandel	7	15	23	25	25
New Method	7	17	21	21	21
E and V	7	15	32	46	60
I	25	45	81	*	*
Unknowns on Γ	125	753	3521	8305	15105

Table 5.2: Iteration Counts for Model Problem

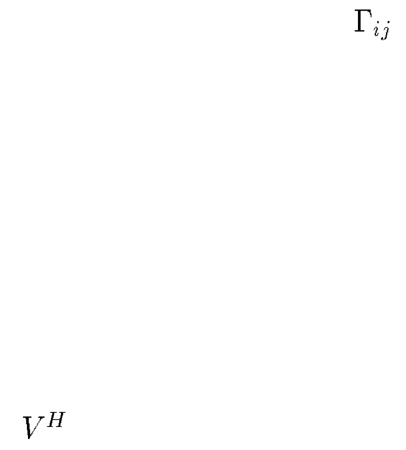


Figure 5.6: Model Problem Without Vertex Spaces

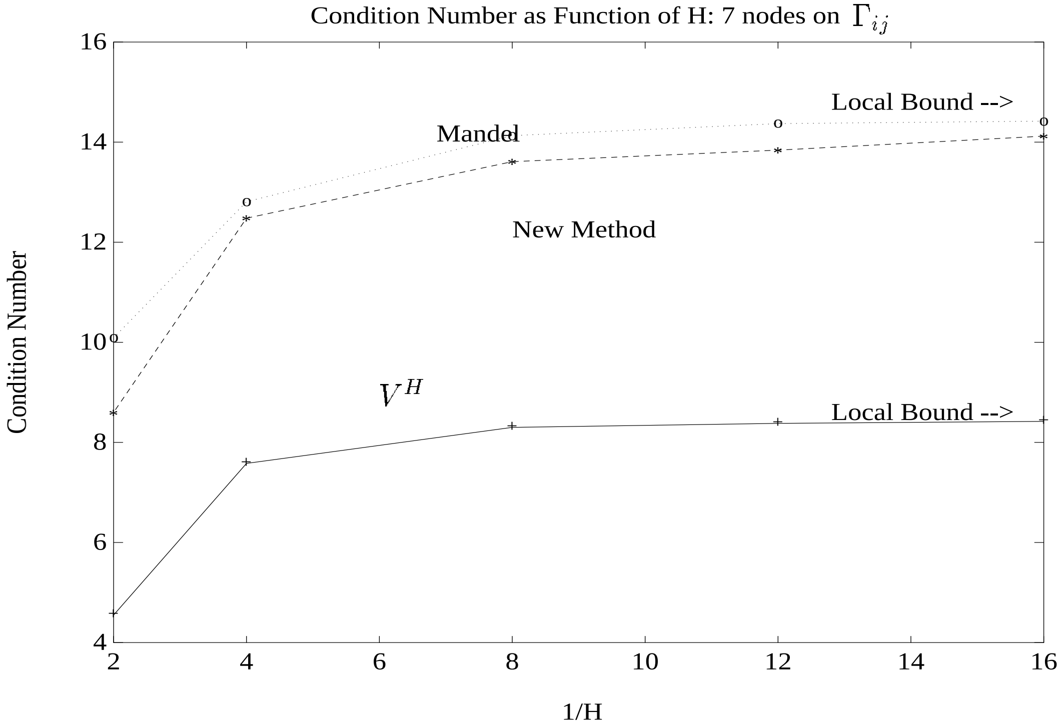


Figure 5.7: Model Problem Without Vertex Spaces

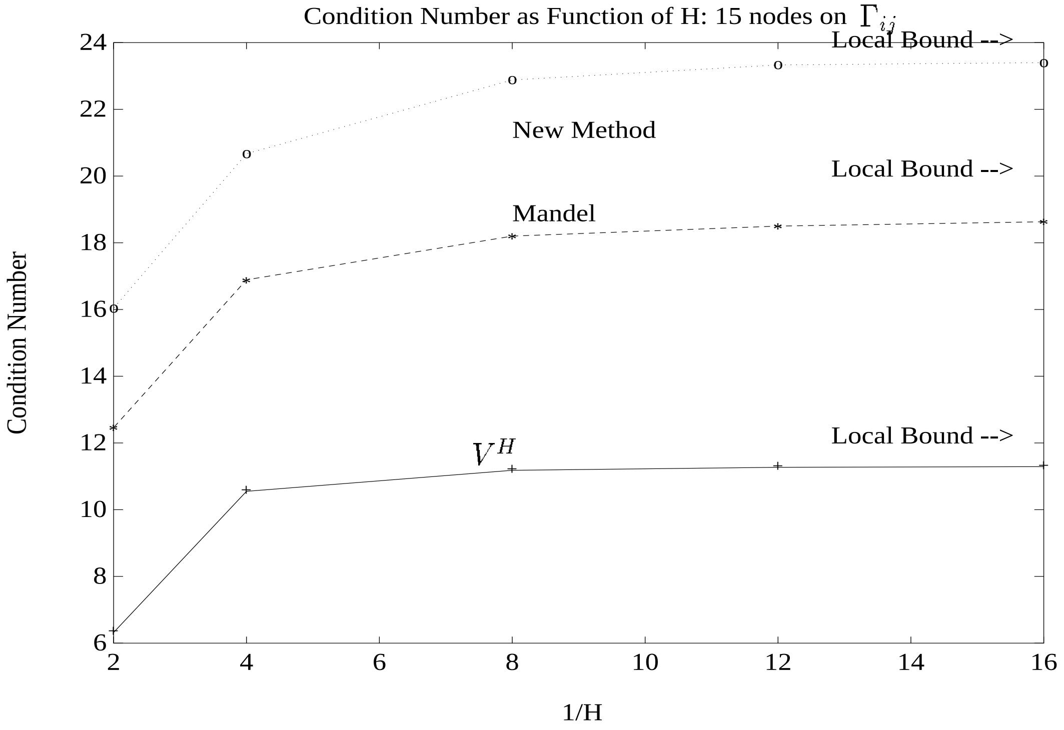


Figure 5.8: Model Problem Without Vertex Spaces

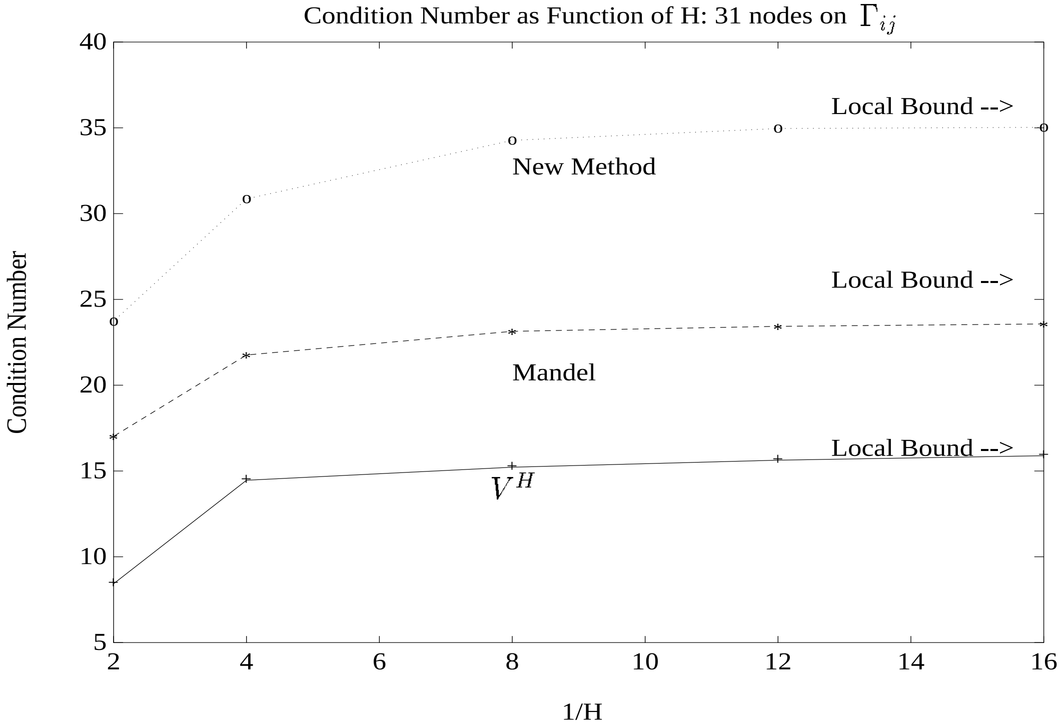


Figure 5.9: Model Problem Without Vertex Spaces

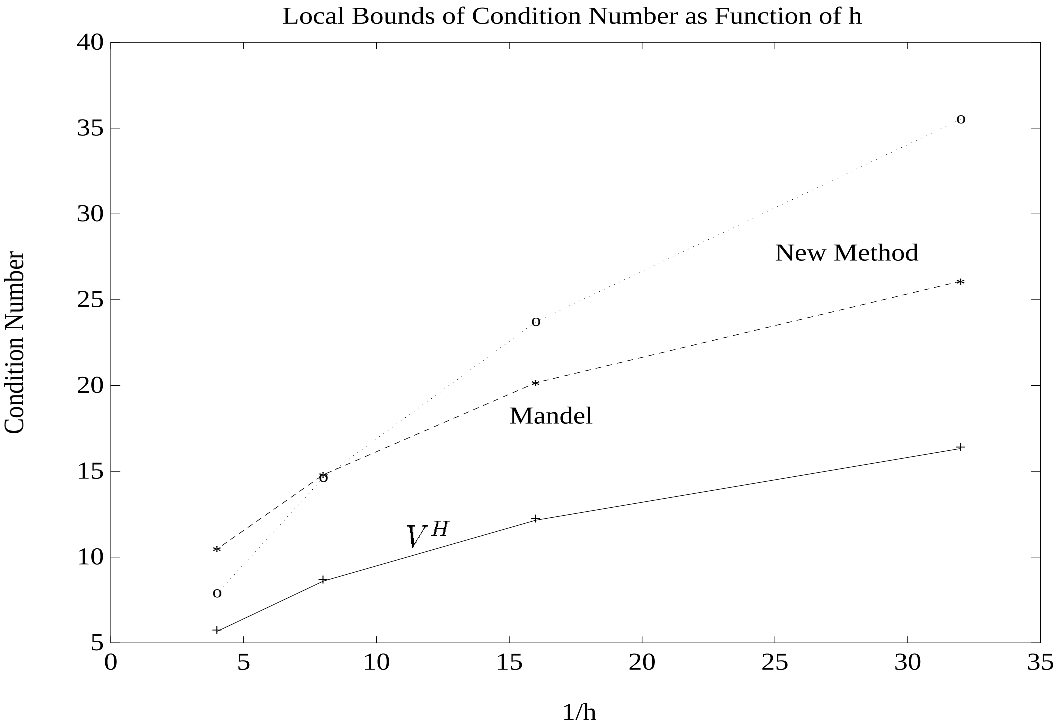


Figure 5.10: Model Problem Without Vertex Spaces

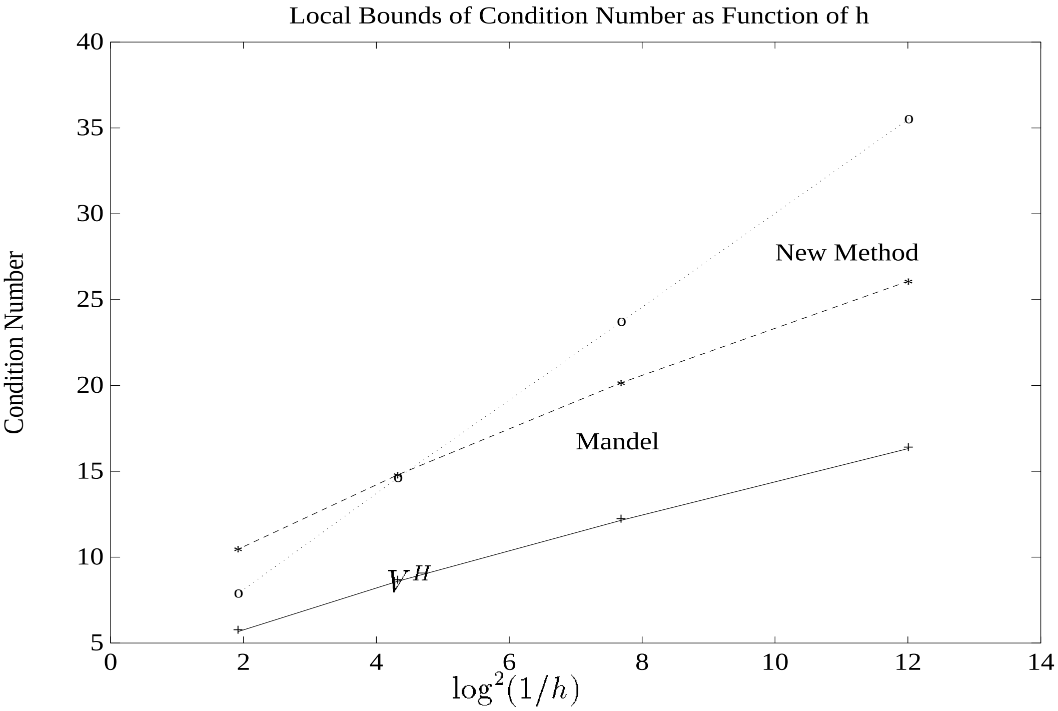


Figure 5.11: Model Problem Without Vertex Spaces

Preconditioner	Number of Substructures					Local
	2 by 2	4 by 4	8 by 8	12 by 12	16 by 16	Bound
3 Nodes on Γ_{ij}						
V^H	1.98	2.45	2.60	2.63	2.64	
Mandel	4.73	9.22	10.02	11.26	11.39	
New Method	2.83	4.40	4.81	4.89	4.92	
Unknowns on Γ	13	81	385	913	1665	
7 Nodes on Γ_{ij}						
V^H	2.09	2.55	2.68	2.70	2.70	
Mandel	5.67	11.10	13.23	13.65	13.82	
New Method	4.07	6.63	7.34	7.48	7.53	
Unknowns on Γ	29	177	833	1969	3585	
15 Nodes on Γ_{ij}						
V^H	2.15	2.82	2.78	2.80	2.81	
Mandel	6.27	12.60	15.50	16.09	16.32	
New Method	5.22	8.92	10.00	10.20	10.28	
Unknowns on Γ	61	369	1729	4081	7425	
31 Nodes on Γ_{ij}						
V^H	2.19	2.99	2.88	2.88	2.89	
Mandel	6.61	14.23	17.88	18.61	18.89	
New Method	6.30	11.22	12.77	13.06	13.16	
Unknowns on Γ	125	753	3521	8305	15105	

Table 5.3: Condition Numbers with Vertex Spaces

Preconditioner	Number of Substructures				
	2 by 2	4 by 4	8 by 8	12 by 12	16 by 16
		3 Nodes on Γ_{ij}			
V^H	4	7	7	7	7
Mandel	4	11	15	16	17
New Method	4	9	11	11	11
Unknowns on Γ	13	81	385	913	1665
		7 Nodes on Γ_{ij}			
V^H	6	7	7	7	7
Mandel	6	12	16	18	18
New Method	6	11	13	13	13
Unknowns on Γ	29	177	833	1969	3585
		15 Nodes on Γ_{ij}			
V^H	6	7	7	7	7
Mandel	7	12	17	20	20
New Method	7	11	14	15	14
Unknowns on Γ	61	369	1729	4081	7425
		31 Nodes on Γ_{ij}			
V^H	6	6	6	6	6
Mandel	7	12	18	21	22
New Method	7	11	16	16	16
Unknowns on Γ	125	753	3521	8305	15105

Table 5.4: Iteration Counts With Vertex Spaces

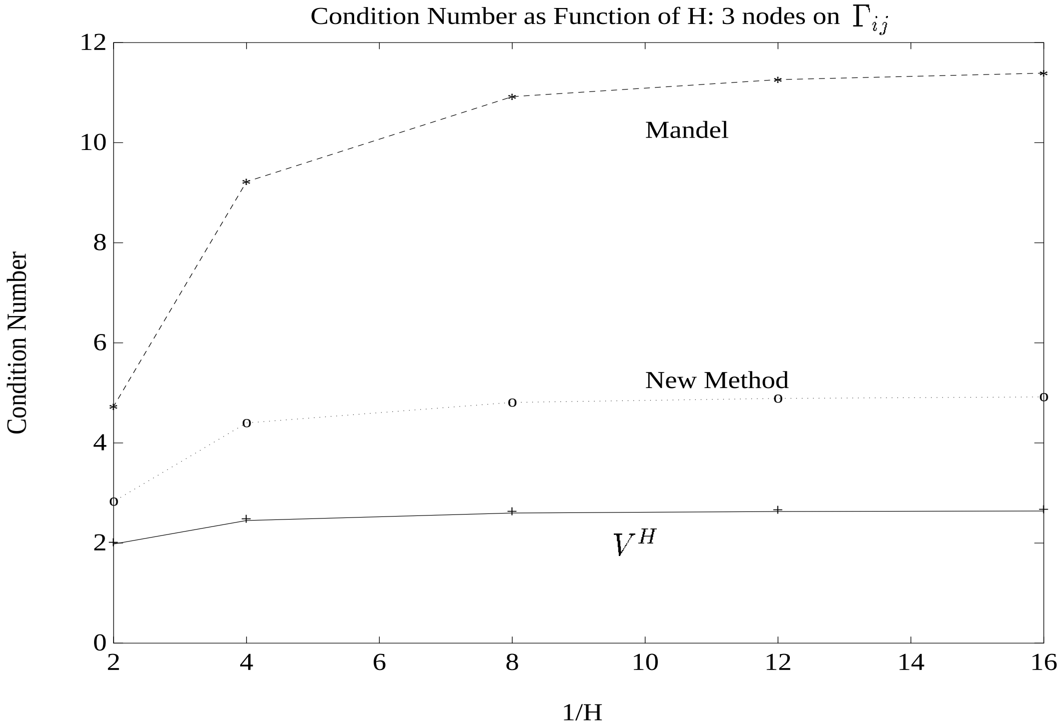


Figure 5.12: Model Problem With Vertex Spaces

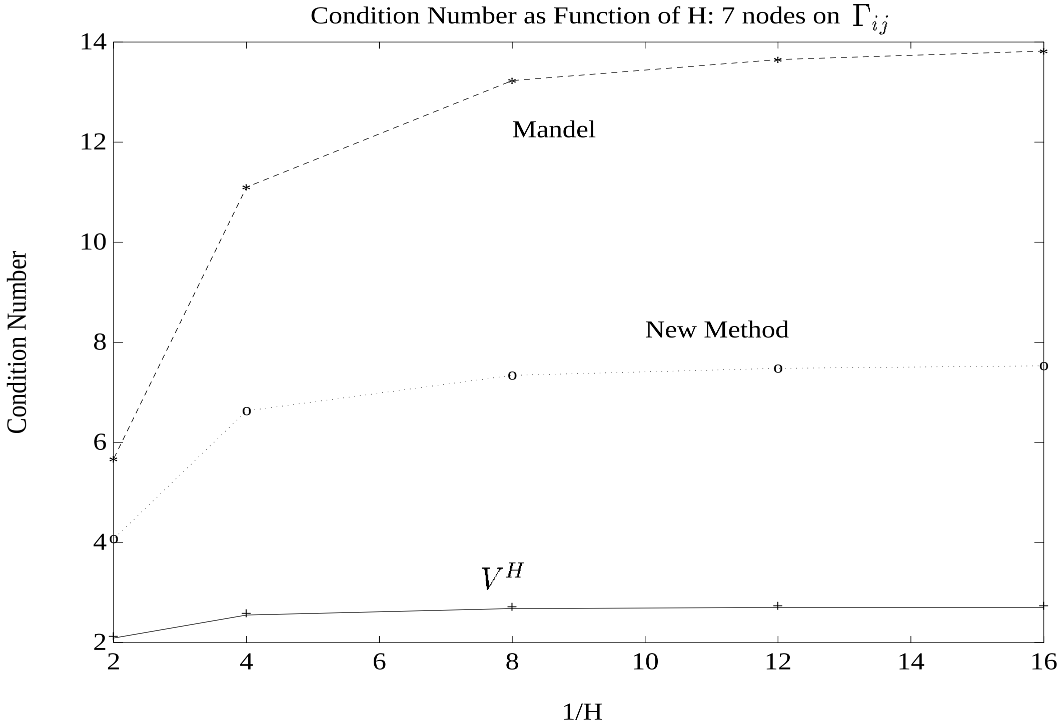


Figure 5.13: Model Problem With Vertex Spaces

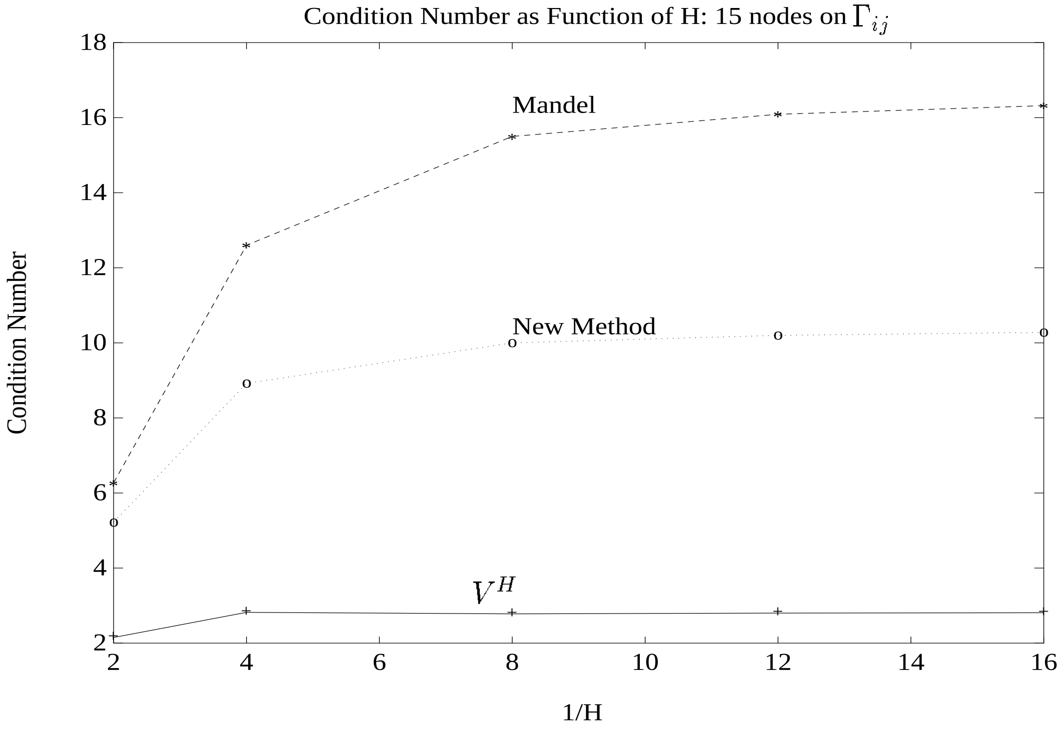


Figure 5.14: Model Problem With Vertex Spaces

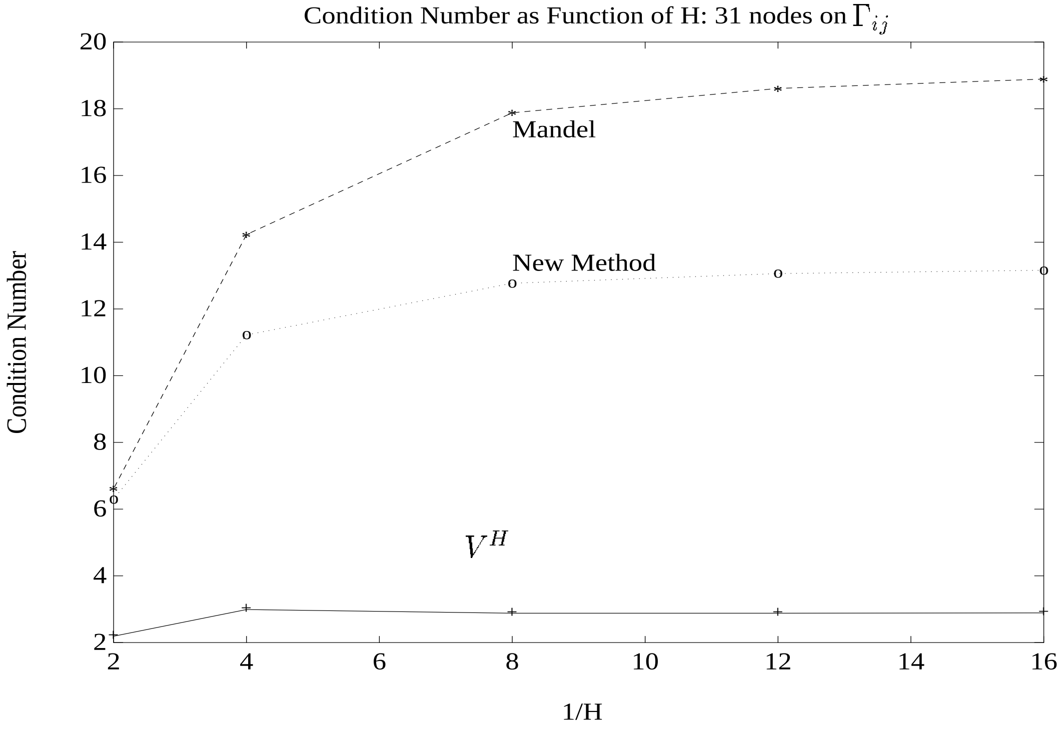


Figure 5.15: Model Problem With Vertex Spaces

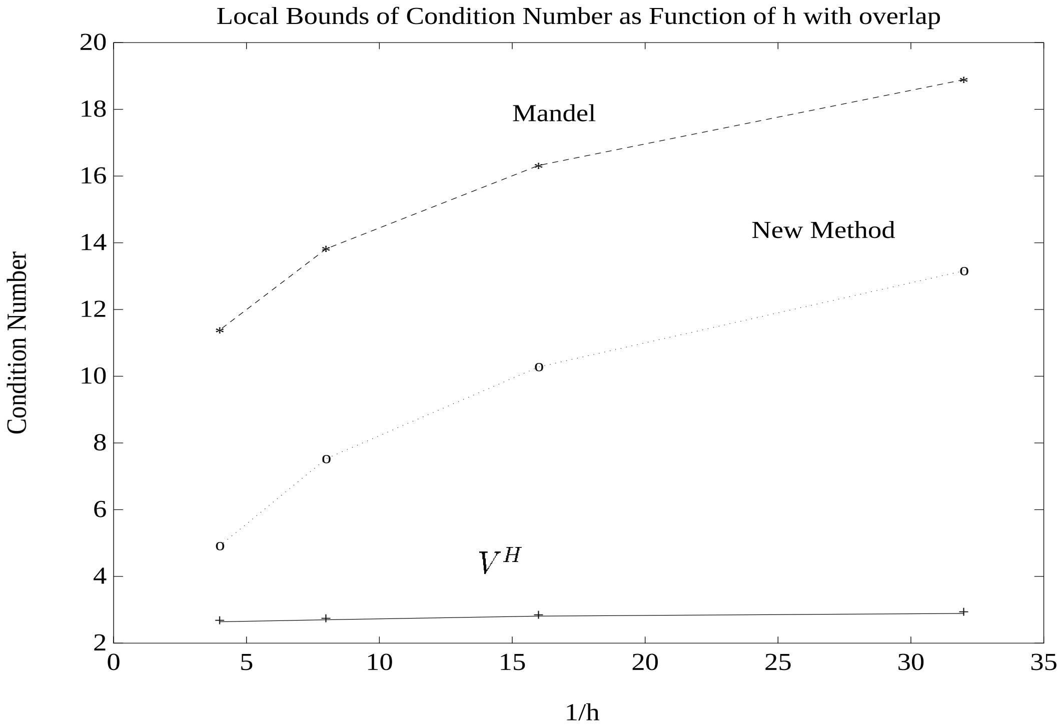


Figure 5.16: Model Problem With Vertex Spaces

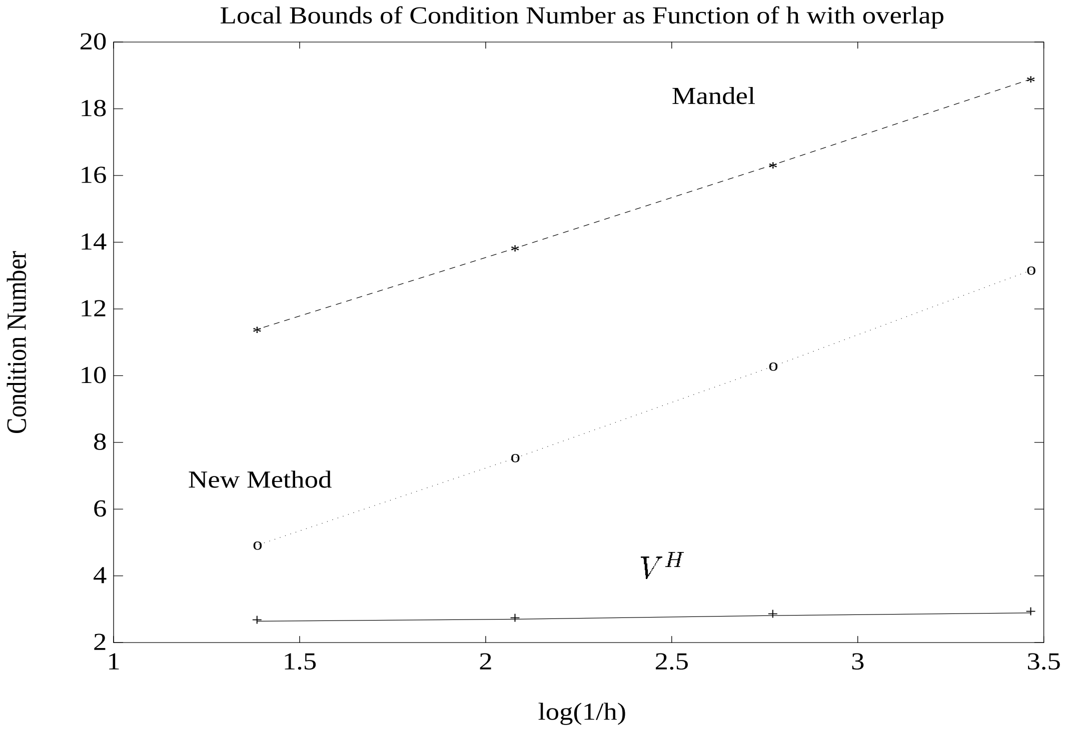


Figure 5.17: Model Problem With Vertex Spaces

Chapter 6

Operation Counts

6.1 Introduction

A standard way of judging the efficiency of various algorithms is to count the number of floating point operations (flops) needed. With the advent of parallel and vector machines these counts must be weighed very carefully with other considerations, but they are still one of the most important single components.

We will consider the case of a square domain partitioned with a regular grid of square substructures with N substructures along each edge of the domain. Each substructure will consist of a regular grid of square elements with n elements along each edge of the substructure, see Figure 6.1. We shall impose a homogeneous Dirichlet boundary condition on the domain. We define the interface between substructures by $\Gamma = \cup \partial\Omega_i \setminus \partial\Omega$. In Figure 6.1, we note that the elements form a regular $Nn - 1$ by $Nn - 1$ mesh in the domain. The vertices of the substructures form a regular $N - 1$ by $N - 1$ grid. We also note, for later use, that the total number of edges, $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$, is $2N(N - 1)$. The number of nodes on Γ is given by $2N(N-1)(n-1) + (N-1)^2$. We will assume throughout that n and N are of the same order of magnitude, otherwise the number of terms we must keep in the operation counts become unmanageable.

The usual 5-point stencil, which arises from using a regular grid of triangles and the space of piecewise linear functions, P_1, can be included in this discussion even though the elements, strictly speaking, are not square, see Figure 6.2.

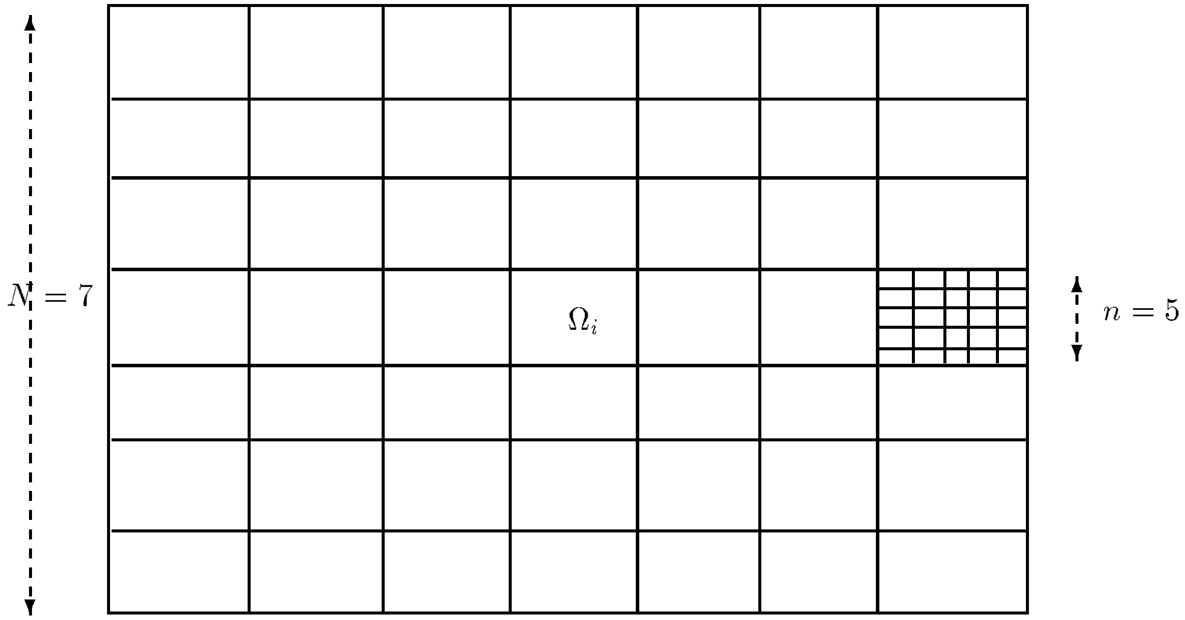


Figure 6.1: Regular Grid of Substructures

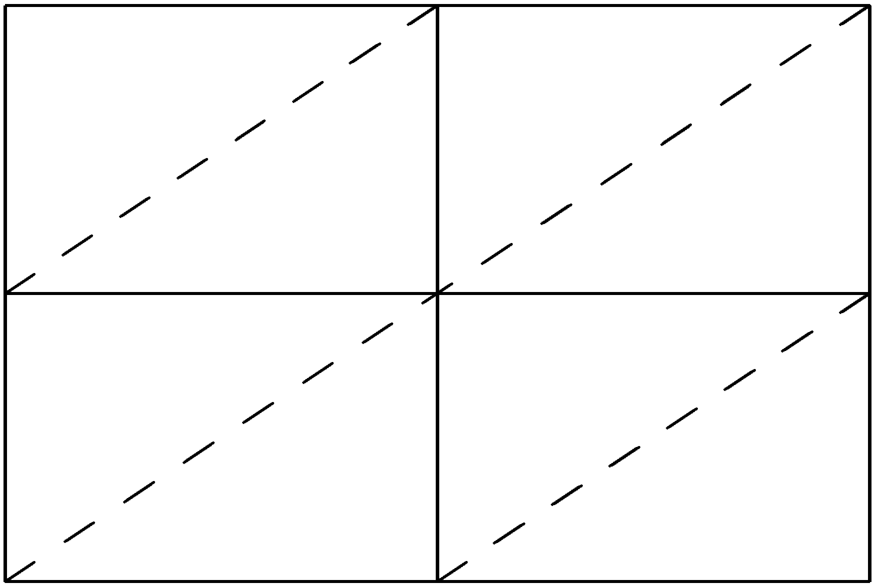


Figure 6.2: Considering Triangular Elements as a Rectangular Grid

6.2 Nested Dissection

Throughout this section we will assume nested dissection is used whenever the stiffness matrix associated with a regular grid is to be factored.

In [39], George gives the operation counts for the nested dissection of a regular finite element mesh. In counting floating point operations, to be consistent with George's study, we will only count multiplications and divides. They are given for a p by p mesh by

- to factor the stiffness matrix

$$\frac{267}{28}p^3 - 17p^2\log_2(p) + \frac{121}{4}p^2 + O(p\log_2(p))$$

- to perform a backsolve and forward solve on the resulting factorization

$$\frac{31}{2}p^2\log_2(p) - \frac{146}{3}p^2 + 48p\log_2(p) + O(p).$$

- The storage requirements for the factors are,

$$\frac{31}{4}p^2\log_2(p) - \frac{73}{3}p^2 + 24p\log_2(p) + O(p).$$

The operation counts for using nested dissection for the grid as defined in Figure 6.1 is easily calculated to be

$$\frac{267}{28}(Nn)^3 - 17(Nn)^2\log_2(Nn-1) + \frac{23}{14}(Nn)^2$$

for the factorization and

$$\frac{31}{2}(Nn)^2\log_2(Nn-1) - \frac{146}{3}(Nn)^2 + 17Nn\log_2(Nn-1)$$

for the backsolve and forward solve. The amount of storage needed is

$$\frac{31}{4}(Nn)^2\log_2(Nn-1) - \frac{73}{3}(Nn)^2 + \frac{17}{2}Nn\log_2(Nn-1).$$

These are useful numbers to compare with results for domain decomposition algorithms.

6.2.1 Iterative Substructuring with Overlap

Floating Point Operations

We now calculate the number of operations needed for an iterative substructuring algorithm with overlapping, see Chapter 4. We will assume a overlap of twenty-five percent. Experimentally this seems to be a reasonable choice though further work needs to be done to determine an optimal overlap.

1. Factor $K^{(i)} = LL^T$	$N^2(\frac{267}{28}n^3 - 17n^2\log_2(n-1) + \frac{23}{14}n^2) + O(N^2 n\log_2(n))$.
2. Form $L^{-T}(L^{-1}K_{IB}^{(i)})$	$N^2(62n^3\log_2(n-1) - \frac{584}{3}n^3) + \ O(N^2n^2\log_2(n))$.
3. $S^{(i)} = K^{(i)} - K_{IB}^{(i)^T} L^{-T}L^{-1}K_{IB}^{(i)}$	$48N^2n^2$.
4. Form $L^{-T}L^{-1}b_I^{(i)}$	$N^2(\frac{31}{2}n^2\log_2(n-1) - \frac{146}{3}n^2) + O(N^2n\log_2(n))$.
5. $b_B^{(i)} = b_B^{(i)} - K_{IB}^{(i)^T}(L^{-T}L^{-1}b_I^{(i)})$	$12N^2n$.
6. Factor Edge Spaces	$\frac{1}{3}N^2n^3 + O(N^2n^2)$.
7. Factor Vertex Spaces	$\frac{1}{6}N^2n^3 + O(N^2n^2)$.
8. Factor Coarse Problem	$\frac{267}{28}N^3 - 17N^2\log_2(N-1) + O(N^2)$.
9. Conjugate Gradient iteration	
a. Matrix Multiply	$16N^2n^2$.
b. Edge Space solves	$2N^2n^2$.
c. Vertex Space solves	N^2n^2.
d. Coarse Problem solve	$\frac{31}{2}N^2\log_2(N-1) - \frac{146}{3}N^2 + \ O(N\log_2(N))$.
e. Interpolation	$4N^2n$.
f. Inner Products	$12N^2n$.
10. $b_I^{(i)} = b_I^{(i)} - K_{IB}^{(i)}x_B^{(i)}$	$12N^2n$.
11. $x_I^{(i)} = L^{-T}L^{-1}b_I^{(i)}$	$N^2(\frac{31}{2}n^2\log_2(n-1) - \frac{146}{3}n^2) + O(N^2n\log_2(n))$.

We have shown, theoretically and numerically, in Chapter 4, that the number of iterations required to converge to a fixed precision for this algorithm is independent of both n and N, cf. [77]. The overall operation count is then given by

$$62N^2n^3\log_2(n-1) - \frac{15509}{84}N^2n^3 + \frac{267}{28}N^3 + O(N^2\log_2(N-1) + N^2n^2\log_2(N-1)).$$

If we let $N = n$ then we obtain an algorithm which is of lower order than nested dissection $O(N^2n^3\log_2(n-1))$ vs $O(N^3n^3)$. Multigrid on the other hand is still only $O(N^2n^2)$.

Storage Requirements

Outside of the CG iterations

1. $LL^T = K^{(i)}$	$\frac{31}{4}N^2n^2\log_2(n-1) - \frac{73}{3}N^2n^2 + O(N^2n\log_2(n-1))$.
2. $K_{IB}^{(i)}$	$12N^2n$.
3. Vectors	$2N^2n^2$.

Inside of the CG iterations

1. $S^{(i)}$	$8N^2n^2$.
2. Local Preconditioners	$\frac{3}{2}N^2n^2$.
3. Vectors	$12N^2n$.

6.2.2 Additive Schwarz

Floating Point Operations

Below we give the operation counts for the additive Schwarz method as proposed by Dryja and Widlund, cf. [36]. Again we use an overlap of twenty-five percent.

1. Factor $\tilde{K}^{(i)} = LL^T$	$N^2(\frac{7209}{224}n^3 - \frac{153}{4}n^2\log_2(\frac{3}{2}n-1) + \frac{1089}{16}n^2) + O(N^2n\log_2(n))$.
2. Factor Coarse Problem	$\frac{267}{28}N^3 - 17N^2\log_2(N-1) + O(N^2)$.
3. Conjugate Gradient iteration	
a. Matrix Multiply	$9N^2n^2 + O(Nn)$.
b. Subspace solves	$N^2(\frac{279}{8}n^2\log_2(\frac{3}{2}n-1) - \frac{1314}{12}n^2) + O(N^2n\log_2(n))$.
d. Coarse Problem	$\frac{31}{2}N^2\log_2(N-1) - \frac{146}{3}N^2 + 17N\log_2(N-1)$ $+ O(N)$.
e. Interpolation	$2N^2n^2 + O(Nn)$.
f. Inner Products	$6N^2n^2 + O(Nn)$.

The overall operation count for the additive Schwarz method is given by

$$\frac{7209}{224}N^2n^3 + \frac{267}{28}N^3 + (\frac{279}{8}q - \frac{153}{4})N^2n^2\log_2(\frac{3}{2}n-1) + O(N^2n^2 + N^2\log_2(N)).$$

Here q is the number of conjugate gradient iterations required plus one. Again as with iterative substructuring with overlap we know theoretically [36] and numerically [42], that the number of conjugate gradient iterations required to obtain a solution of a given precision is independent of n and N.

Storage Requirements

Inside of the CG iterations

1. $LL^T = \tilde{K}^{(i)}$	$\frac{279}{16}N^2n^2\log_2(\frac{3}{2}n-1) - \frac{657}{12}N^2n^2 + O(N^2n\log_2(n))$.
2. Vectors	$6N^2n^2$.
3. K	$\frac{9}{2}N^2n^2$.

6.2.3 Iterative Substructuring with Approximate Solvers

Floating Point Operations

One version of this algorithm, due to Bramble, Pasiack, and Schatz, is outlined in great detail in [12]. Another using a different preconditioner for the edge variables is described in Chapter 3 and in Smith and Widlund [78].

1. Factor $K^{(i)} = LL^T$	$N^2(\frac{267}{28}n^3 - 17n^2\log_2(n-1) + \frac{23}{14}n^2) + O(N^2n\log_2(n))$.
2. Factor Coarse Problem	$\frac{267}{28}N^3 - 17N^2\log_2(N-1) + O(N^2)$.
3. Conjugate Gradient iteration	
a. Matrix Multiply	
$K_{IB}^{(i)}v$	$12N^2n$.
$L^{-T}L^{-1}K_{IB}^{(i)}v$	$N^2(\frac{31}{2}n^2\log_2(n-1) - \frac{146}{3}n^2) + O(N^2n\log_2(n))$.
$K_{IB}^{(i)^T}L^{-T}L^{-1}K_{IB}^{(i)}v$	$12N^2n$.
b. Subspace solves	$2N^2n$
d. Coarse Problem	$\frac{31}{2}N^2\log_2(N-1) - \frac{146}{3}N^2 + 17N\log_2(N-1)$
e. Interpolation	$4N^2n$.
f. Inner Products	$12N^2n$.
4. $b_I^{(i)} = b_I^{(i)} - K_{IB}^{(i)}x_B^{(i)}$	$12N^2n$.

5. $x_I^{(i)} = L^{-T}L^{-1}b_I^{(i)}$ $\qquad$ $N^2(\frac{31}{2}n^2\log_2(n-1) - \frac{146}{3}n^2) + O(N^2 n \log_2(n))$.

The overall operation counts for this method is given by

$$\frac{267}{28}N^2n^3 + (q\frac{31}{2} - 17)N^2n^2\log_2(n-1) + \frac{267}{28}N^3 + O(N^2n^2 + qN^2log_2(N)).$$

This class of methods is non-optimal; we know theoretically and numerically, see Chapter 3 and [12],[78], that the number of iterations required for a fixed precision increases linearly with $\log(n)$, i.e. $q = C\log(n)$.

Storage Requirements

Outside of the CG iterations

1. Vectors	$6N^2n^2$.

Inside of the CG iterations

1. $LL^T = K^{(i)}$	$\frac{31}{4}N^2n^2\log_2(n-1)$ $-\frac{73}{3}N^2n^2 + O(N^2n\log_2(n-1))$.
2. $K_{IB}^{(i)}$	$12N^2n$.
3. $K_{BB}^{(i)}$	$8N^2n$.
4. Vectors	$12N^2n$.

6.3 Band Solvers

We again calculate the operation counts, this time assuming an underlying band solver for solving the subdomain problems and the coarse problem. The operation counts for factoring a symmetric positive definite m by m matrix with bandwidth p is given by, see, e.g., [41],

$$\frac{mp^2}{2} - \frac{p^3}{3} + \frac{3}{2}(mp - p^2).$$

The cost for a backsolve-forward solve on the factored matrix is

$$2m(p+1) - p^2.$$

6.3.1 Iterative Substructuring with Overlap

Floating Point Operations

We now calculate the number of operations needed for the iterative substructuring algorithm with overlapping introduced in Chapter 4. We will assume a overlap of twenty-five percent. Experimentally this seems to be a reasonable choice though further work needs to be done to determine an optimal overlap.

1. Factor $K^{(i)} = LL^T$	$N^2(\frac{1}{2}n^4 - \frac{5}{6}n^3 - 2n^2) + O(N^2n)$
2. Form $L^{-T}(L^{-1}K_{IB}^{(i)})$	$N^2(8n^4 - 20n^3 + 16n^2) + O(N^2n)$
3. $S^{(i)} = K^{(i)} - K_{IB}^{(i)^T}(L^{-T}L^{-1}K_{IB}^{(i)})$	$48N^2n^2$
4. Form $L^{-T}L^{-1}b_I^{(i)}$	$N^2(2n^3 - 5n^2) + O(N^2n)$
5. $b_B^{(i)} = b_B^{(i)} - K_{IB}^{(i)^T}(L^{-T}L^{-1}b_I^{(i)})$	$12N^2n$
6. Factor Edge Spaces	$\frac{1}{3}N^2n^3 + O(N^2n^2)$.
7. Factor Vertex Spaces	$\frac{1}{6}N^2n^3 + O(N^2n^2)$.
8. Factor Coarse Problem	$\frac{1}{2}N^4 - \frac{5}{6}N^3 - 2N^2 + O(N)$
9. Conjugate Gradient iteration	
a. Matrix Multiply	$16N^2n^2$.
b. Edge Space solves	$2N^2n^2$.
c. Vertex Space solves	N^2n^2.
d. Coarse Problem solve	$2N^3 - 5N^2 + O(N)$
e. Interpolation	$4N^2n$.
f. Inner Products	$12N^2n$.
10. $b_I^{(i)} = b_I^{(i)} - K_{IB}^{(i)}x_B^{(i)}$	$12N^2n$.
11. $x_I^{(i)} = L^{-T}L^{-1}b_I^{(i)}$	$N^2(2n^3 - 5n^2) + O(N^2n)$

We have shown, theoretically and numerically, in Chapter 4, that the number of iterations required to converge to a fixed precision for this algorithm is independent of both n and N, cf. [77]. The overall operation count is given by

$$\frac{17}{2}N^2n^4 - \frac{55}{3}N^2n^3 + \frac{1}{2}N^4 + (2q - \frac{5}{6})N^3 + O(N^2n^2).$$

Storage Requirements

Outside of the CG iterations

1. $LL^T = K^{(i)}$	$N^2(\frac{1}{2}n^4 - 2n^3 + 3n^2) + O(N^2n)$
2. $K_{IB}^{(i)}$	$12N^2n$.
3. Vectors	$2N^2n^2$.

Inside of the CG iterations

1. $S^{(i)}$	$8N^2n^2$.
2. Local Preconditioners	$\frac{3}{2}N^2n^2$.
3. Vectors	$12N^2n$.

6.3.2 Additive Schwarz

Floating Point Operations

Below we give the operation counts for the additive Schwarz method as proposed by Dryja and Widlund, cf. [36].

1. Factor $\tilde{K}^{(i)} = LL^T$	$N^2(\frac{81}{32}n^4 + \frac{63}{16}n^3 - \frac{27}{8}n^2) + O(N^2n)$
2. Factor Coarse Problem	$\frac{1}{2}N^4 - \frac{5}{6}N^3 - 2N^2 + O(N)$
3. Conjugate Gradient iteration	
a. Matrix Multiply	$9N^2n^2 + O(Nn)$.
b. Subspace solves	$N^2(\frac{27}{4}n^3 - \frac{9}{4}n^2)$
d. Coarse Problem solve	$2N^3 - 5N^2 + O(N)$
e. Interpolation	$N^2n^2 + O(Nn)$.
f. Inner Products	$6N^2n^2 + O(Nn)$.

The overall operation count for the additive Schwarz method is given by

$$\frac{81}{32}N^2n^4 + \frac{1}{2}N^4 + (2q - \frac{5}{6})N^3 + (\frac{63}{16} + q\frac{27}{4})N^2n^3 + O(N^2n^2q).$$

Here q is the number of conjugate gradient iterations required plus one. Again as with iterative substructuring with overlap we know theoretically [36] and numerically [42], that the number of conjugate gradient iterations required to obtain a solution of a given precision is independent of n and N.

Storage Requirements

Inside of the CG iterations

1. $LL^T = \tilde{K}^{(i)}$	$\frac{81}{32}N^2n^4$
2. Vectors	$6N^2n^2$.
3. K	$\frac{9}{2}N^2n^2$.

6.3.3 Iterative Substructuring with Approximate Solvers

Floating Point Operations

One version of this algorithm, due to Bramble, Pasciak, and Schatz, is outlined in great detail in [12]. Another using a different preconditioner for the edge variables is described in Chapter 3 and in Smith and Widlund [78].

1. Factor $K^{(i)} = LL^T$	$N^2(\frac{1}{2}n^4 - \frac{5}{6}n^3 - 2n^2) + O(N^2n)$
2. Factor Coarse Problem	$\frac{1}{2}N^4 - \frac{5}{6}N^3 - 2N^2 + O(N)$
3. Conjugate Gradient iteration	
a. Matrix Multiply	
$K_{IB}^{(i)}v$	$12N^2n$.
$L^{-T}L^{-1}K_{IB}^{(i)}v$	$N^2(2n^3 - 5n^2) + O(N^2n)$
$K_{IB}^{(i)^T}L^{-T}L^{-1}K_{IB}^{(i)}v$	$12N^2n$
b. Subspace solves	$2N^2n$
d. Coarse Problem solve	$2N^3 - 5N^2 + O(N)$
e. Interpolation	$4N^2n$.
f. Inner Products	$12N^2n$.
4. $b_I^{(i)} = b_I^{(i)} - K_{IB}^{(i)}x_B^{(i)}$	$12N^2n$.

5. $x_I^{(i)} = L^{-T}L^{-1}b_I^{(i)}$	$N^2(2n^3 - 5n^2) + O(N^2n)$

The overall operation counts for this method is given by

$$\frac{1}{2}N^2n^4 + \frac{1}{2}N^4 + (2q - \frac{5}{6})N^3 + (\frac{7}{6} + 2q)N^2n^3 + O(N^2n^2).$$

This class of methods is non-optimal. We know theoretically and numerically [12],[78] that the number of iterations required for a fixed precision increases linearly with $\log(n)$, i.e. $q = C\log(n)$.

Storage Requirements

Outside of the CG iterations

1. Vectors	$6N^2n^2$.

Inside of the CG iterations

1. $LL^T = K^{(i)}$	$N^2(\frac{1}{2}n^4 - 2n^3 + 3n^2) + O(N^2n)$
2. $K_{IB}^{(i)}$	$12N^2n$.
3. $K_{BB}^{(i)}$	$8N^2n$.
4. Vectors	$12N^2n$.

6.4 Determining the Optimal Domain Size

We wish to find a strategy for deciding on the sizes for the subdomains. We first consider attempting to minimize the operation count without any other considerations. This would, in general, be a poor way of determining the optimal size for subdomains, but it is a starting point. It is virtually impossible algebraically to determine the best subdomain size given the complexity of the operation counts, so instead we will proceed graphically. We consider the case when the grid is a 1000 by 1000 mesh and then a $10,000$ by $10,000$ mesh. We plot the operation count as a function of the size of the substructures measured by the number of mesh points along one edge. Remember that we are using for our model a square domain with square subdomains. The results are depicted in Figure 6.3 and 6.4.

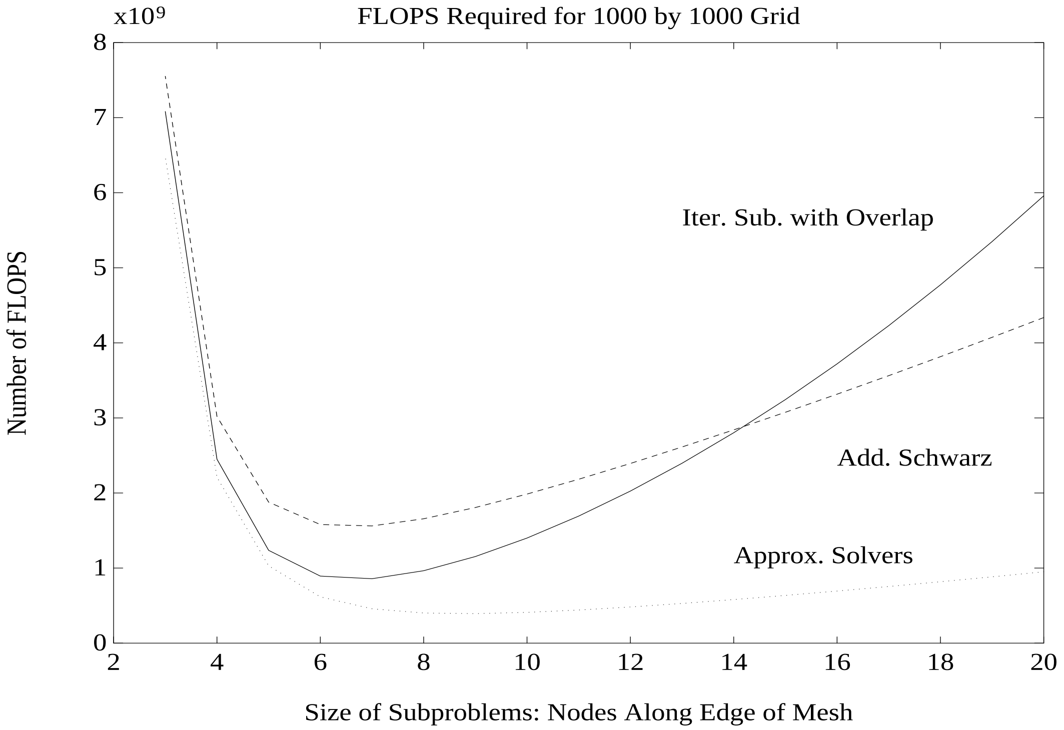


Figure 6.3:

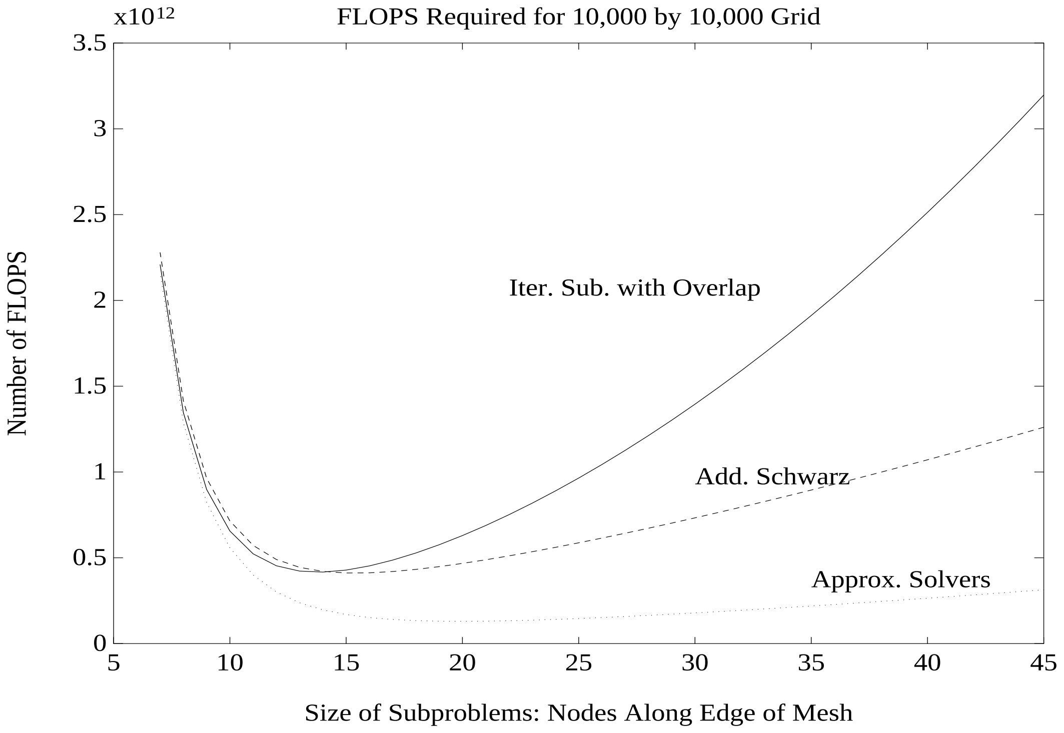


Figure 6.4:

The main conclusion is that to decrease the operation counts very small subdomains are the best. This is no surprise since the leading terms are of the form $O(N^2n^4)$. This type of analysis suggests that we should use hundreds or thousands of very small subdomains.

We also see that iterative substructuring with approximate solvers has the best properties; the operation counts are lower than the other two methods and they are much less dependent on having small subdomains. We can use much larger subdomains and still have almost optimal results.

The additive Schwarz methods of Dryja and Widlund and the iterative substructuring algorithms with approximate solvers generally require much more storage space since the iterations require all the data for the interior of the subdomains as well as for the boundaries. Therefore for very large problems, like linear elasticity problems, iterative substructuring with overlap may be preferred since it requires much less in-core or swapped storage space. No general statement can be made as to which algorithm will perform best in a variety of situations.

We note that since these calculations are somewhat imprecise, we have perturbed them by a large lower order term, $100N^2n^2$, with no effect on the qualitative behavior of the graphs. We take this as a sign that the results have some validity.

6.5 Comparison to Unpreconditioned Problem

The application of the conjugate gradient method to the original unpreconditioned linear system has a very small cost per iteration. The number of operations required for one iteration of the conjugate gradient method is about $15N^2n^2$ flops. This includes the matrix-vector multiply and inner product calculations.

For the model problem, the condition number of the unpreconditioned problem grows like $O(\frac{1}{(Nn)^2})$. Thus the number of iterations required would be $O(Nn)$. The overall operation count for the conjugate gradient method is then $O(N^3n^3)$.

For the domain decomposition methods using a sparse interior solve the leading terms of the operation counts are given below.

1. $62N^2n^3\log_2(n-1)+\frac{267}{28}N^3$ – Iterative substructuring with overlap.
2. $\frac{7209}{224}N^2n^3+\frac{267}{28}N^3$ – Additive Schwarz.
3. $\frac{267}{28}N^2n^3+\frac{267}{28}N^3$ – Iterative substructuring with approximate solves.

When band solvers are used, the leading order terms are given by,

1. $\frac{17}{2}N^2n^4 + \frac{1}{2}N^4$ – iterative substructuring with overlap,
2. $\frac{81}{32}N^2n^4 + \frac{1}{2}N^4$ – additive Schwarz,
3. $\frac{1}{2}N^2n^4 + \frac{1}{2}N^4$ – iterative substructuring with approximate solves.

The domain decomposition algorithms are constructed to be robust. The condition numbers, and hence iteration counts, will not depend strongly on the geometry, nor will they depend strongly on the form of the differential equation. This is not true of the conjugate gradient method applied to the original stiffness matrix. In that case, a differential equation resulting in a poorly conditioned stiffness matrix will cause a large number of iterations and thus a higher operation cost. In addition, since the unpreconditioned conjugate gradient method requires such a large number of iterations, reorthogonalisation may be required. This is an expensive operation.

No general conclusion can yet be drawn as to the long term superiority of the different approaches to the iterative solution of linear systems arising from finite element and finite difference discretizations of partial differential equations.

References

[1] Ivo Babuška, Alan Craig, Jan Mandel, and Juhani Pitkäranta. Efficient preconditioning for the p-version finite element method in two dimensions. Technical Report 41098, University of Colorado at Denver, October 1989. Submitted to SIAM J. Numer. Anal.

[2] Ivo Babuška, M. Griebel, and J. Pitkäranta. The problem of selecting the shape functions for a p-type finite element. *International. J. Numer. Methods Engrg.*, 18:1891 – 1908, 1989.

[3] Randolph E. Bank, Todd F. Dupont, and Harry Yserentant. The hierarchical basis multigrid method. *Numer. Math.*, 52:427 – 458, 1988.

[4] K. Bell, B. Hatlestad, O. E. Hansteen, and Per O. Araldsen. *NORSAM, a programming system for the finite element method. Users manual, Part 1, General description.* NTH, Trondheim, 1973.

[5] Petter E. Bjørstad. Multiplicative and Additive Schwarz Methods: Convergence in the 2 domain case. In Tony Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Domain Decomposition Methods*, Philadelphia, PA, 1989. SIAM.

[6] Petter E. Bjørstad and Anders Hvidsten. Iterative methods for substructured elasticity problems in structural analysis. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors, *Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1988. SIAM.

[7] Petter E. Bjørstad and Jan Mandel. On the spectra of sums of orthogonal projections with applications to parallel computing. Technical report, De-

partment of Computer Science, University of Bergen, Thormøhlensgaten 55, N-5006 Bergen , Norway, 1989. Submitted to BIT.

[8] Petter E. Bjørstad and Olof B. Widlund. Solving elliptic problems on regions partitioned into substructures. In Garrett Birkhoff and Arthur Schoenstadt, editors, *Elliptic Problem Solvers II*, pages 245–256, New York, 1984. Academic Press.

[9] Petter E. Bjørstad and Olof B. Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.*, 23(6):1093 – 1120, 1986.

[10] Petter E. Bjørstad and Olof B. Widlund. To Overlap or Not to Overlap: A Note on a Domain Decomposition Method for Elliptic Problems. *SIAM J. Sci. Stat. Comput.*, 10(5):1053 – 1061, 1989.

[11] Jean-François Bourgat, Roland Glowinski, Patrick Le Tallec, and Marina Vidrascu. Variational formulation and algorithm for trace operator in domain decomposition calculations. In Tony Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Domain Decomposition Methods*, Philadelphia, PA, 1988. SIAM.

[12] James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. The construction of preconditioners for elliptic problems by substructuring, I. *Math. Comp.*, 47(175):103– 134, 1986.

[13] James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. An iterative method for elliptic problems on regions partitioned into substructures. *Math. Comp.*, 46(173):361–369, 1986.

[14] James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. The construction of preconditioners for elliptic problems by substructuring, II. *Math. Comp.*, 49:1–16, 1987.

[15] James H. Bramble, Joseph E. Pasciak, and Alfred H. Schatz. The construction of preconditioners for elliptic problems by substructuring, IV. *Math. Comp.*, 53:1 – 24, 1989.

[16] James H. Bramble, Joseph E. Pasciak, Junping Wang, and Jinchao Xu. Convergence estimates for product iterative methods with applications to domain decomposition and multigrid. Technical report, Cornell University, 1990.

[17] James H. Bramble, Joseph E. Pasciak, and Jinchao Xu. Parallel multilevel preconditioners. Technical report, Cornell University, 1989.

[18] Xiao-Chuan Cai. Some domain decomposition algorithms for nonselfadjoint elliptic and parabolic partial differential equations. Technical Report 461, Computer Science Department, Courant Institute of Mathematical Sciences, September 1989. Courant Institute doctoral dissertation.

[19] Xiao-Chuan Cai. An additive Schwarz algorithm for nonselfadjoint elliptic equations. In Tony Chan, Roland Glowinski, , Jacques Périaux, and Olof Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*. SIAM, Philadelphia, PA, 1990.

[20] Xiao-Chuan Cai and Olof Widlund. Domain decomposition algorithms for indefinite elliptic problems. Technical Report 506, Computer Science Department, Courant Institute of Mathematical Sciences, May 1990. Accepted for publication in SIAM J. Sci. Statist. Comput.

[21] Tony F. Chan and Danny Goovaerts. Schwarz = Schur: Overlapping versus nonoverlapping domain decomposition. Technical Report CAM 88-21, Department of Mathematics, UCLA, 1988.

[22] Tony F. Chan and David F. Keyes. Interface preconditioning for domain-decomposed convection-diffusion operators. Technical report, Yale, 1989.

[23] Tony F. Chan and Diana C. Resasco. A survey of preconditioners for domain decomposition. Technical Report /DCS/RR-414, Yale University, 1985.

[24] Tony F. Chan and Diana C. Resasco. Analysis of domain decomposition preconditioners on irregular regions. In R. Vichnevetsky and R. Stepleman, editors, *Advances in Computer Methods for Partial Differential Equations*. IMACS, 1987.

[25] P. G. Ciarlet. *Lectures on three-dimensional elasticity.* Springer-Verlag, Berlin, 1983.

[26] Philippe G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North-Holland, 1978.

[27] Paul Concus, Gene H. Golub, and Dianne P. O'Leary. A generalized conjugate gradient method for the numerical solution of elliptic PDE. In James R. Bunch and Donald J. Rose, editors, *Sparse Matrix Computations*, pages 309–332, New York, 1976. Academic Press.

[28] Yann-Hervé De Roeck. A local preconditioner in a domain-decomposed method. Technical report, Centre Europée de Recherche et de Formation Avancée en Calcul Scientifique, Toulouse, France, 1989.

[29] Yann-Hervé De Roeck and Patrick Le Tallec. Analysis and test of a local domain decomposition preconditioner. In Roland Glowinski, Yuri Kuznetsov, Gérard Meurant, Jacques Périaux, and Olof Widlund, editors, *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations.* SIAM, Philadelphia, PA, 1991.

[30] Q.V. Dihn, Roland Glowinski, and Jacques Périaux. Solving elliptic problems by domain decomposition methods with applications. In Garrett Birkhoff and Arthur Schoenstadt, editors, *Elliptic Problem Solvers II*, pages 395–426, New York, 1984. Academic Press.

[31] Maksymilian Dryja. An algorithm with a capacitance matrix for a variational-difference scheme. In G. I. Marchuk, editor, *Variational-Difference Methods in Mathematical Physics*, pages 63 – 73, Novosibirsk, 1981. USSR Academy of Sciences.

[32] Maksymilian Dryja. A capacitance matrix method for Dirichlet problem on polygon region. *Numer. Math.*, 39:51–64, 1982.

[33] Maksymilian Dryja. A method of domain decomposition for 3-D finite element problems. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant,

and Jacques Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1988. SIAM.

[34] Maksymilian Dryja. An additive Schwarz algorithm for two- and three-dimensional finite element elliptic problems. In Tony Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Domain Decomposition Methods*, Philadelphia, PA, 1989. SIAM.

[35] Maksymilian Dryja and Olof B. 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.

[36] Maksymilian Dryja and Olof B. Widlund. Some domain decomposition algorithms for elliptic problems. In *Iterative Methods for Large Linear Systems*, pages 273–291, San Diego, California, 1989. Academic Press. Proceeding of the Conference on Iterative Methods for Large Linear Systems held in Austin, Texas, October 19 - 21, 1988, to celebrate the sixty-fifth birthday of David M. Young, Jr.

[37] Maksymilian Dryja and Olof B. Widlund. Towards a unified theory of domain decomposition algorithms for elliptic problems. In Tony Chan, Roland Glowinski, , Jacques Périaux, and Olof Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989*. SIAM, Philadelphia, PA, 1990.

[38] Gaetano Fichera. *Handbuch der Physik*, volume VIa/2: Mechanics of Solids II, pages 347–388. Springer-Verlag, Berlin, 1972.

[39] Alan George. Nested dissection of a regular finite element mesh. *SIAM J. Numer. Anal.*, 10:345–363, 1973.

[40] Gene Golub and D. Mayers. The use of preconditioning over irregular regions. In R. Glowinski and J. L. Lions, editors, *Computing Methods in Applied Sciences and Engineering, VI*, pages 3–14, Amsterdam, New York, Oxford, 1984.

North-Holland. Proceedings of a conference held in Versailles, France, December 12-16,1983.

[41] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins Univ. Press, 1989. Second Edition.

[42] Anne Greenbaum, Congming Li, and Zheng Chao Han. Parallelizing preconditioned conjugate gradient algorithms. *Computer Physics Communications*, 53:295–309, 1989.

[43] P. Grisvard. *Elliptic problems in nonsmooth domains.* Pitman Publishing, Boston, 1985.

[44] Morton Gurtin. *Handbuch der Physik*, volume VIa/2: Mechanics of Solids II, pages 1–273. Springer-Verlag, Berlin, 1972.

[45] F. Hartmann. *The Mathematical Foundations of Structural Mechanics.* Springer-Verlag, Berlin, 1985.

[46] Thomas R. Hughes and Robert M. Ferencz. Fully vectorized ebe preconditioners for nonlinear solid mechanics: Applications to large-scale three-dimensional continuum, shell and contact/impact problems. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1988. SIAM.

[47] Anders Hvidsten. On the parallelization of a finite element structural analysis program. Technical report, University of Bergen, Norway, 1990. Ph.D. Thesis. Computer Science Department.

[48] Fritz John. *Partial Differential Equations.* Springer-Verlag, New York, 1981. Fourth Edition.

[49] I. E. Kaporin, L. Yu. Kolotilina, and A. Yu. Yermin. Block SSOR preconditionings for high order 3D FE systems II: incomplete BSSOR preconditionings. Technical report, The Department of Numerical Mathematics and Cybernetics of the Moscow State University, Leninskije Gory, 119899, Moscow

B-234, USSR and the Leningrad Branch of V. A. Steklov Mathematical Iinstitute of the USSR Academy of Sciences, Fontanka 27, Leningrad 191011, USSR and the Department of Numerical Mathematics of the USSR Academy of Sciences, Ryleev str. 29, 119034, Moscow, USSR.

[50] David E. Keyes and William D. Gropp. A comparison of domain decomposition techniques for elliptic partial differential equations and their parallel implementation. *SIAM J. Sci. Stat. Comput.*, 8(2):s166–s202, 1987.

[51] David E. Keyes and William D. Gropp. Domain decomposition techniques for the parallel solution of nonsymmetric systems of elliptic bvps. In Tony Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Domain Decomposition Methods*, Philadelphia, PA, 1989. SIAM.

[52] David E. Keyes and William D. Gropp. Domain decomposition with local mesh refinement. Technical Report YALEU/DCS/RR-726, Yale University, August 1989.

[53] L. Yu. Kolotilina and A. Yu. Yermin. Factorized sparse approximate inverse preconditioners. Technical report, The Leningrad Branch of V. A. Steklov Mathematical Iinstitute of the USSR Academy of Sciences, Fontanka 27, Leningrad 191011, USSR and the Department of Numerical Mathematics of the USSR Academy of Sciences, Ryleev str. 29, 119034, Moscow, USSR.

[54] L. Yu. Kolotilina and A. Yu. Yermin. Block SSOR preconditionings for high order 3D FE systems. *BIT*, 29:805–823, 1989.

[55] P. Lax and A.N. Milgram. *Contributions to the Theory of Partial Differential Equations*, pages 167–190. Princeton University Press, Princeton, New Jersey, 1954. Annals of Math. Studies 33.

[56] Patrick Le Tallec, Yann-Hervé De Roeck, and Marina Vidrascu. Domain-decomposition methods for large linearly elliptic three dimensional problems. Technical Report TR/PA/90/20, Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique, Toulouse, France, 1990.

[57] Jacques Louis Lions and Enrico Magenes. *Nonhomogeneous Boundary Value Problems and Applications*, volume I. Springer, New York, Heidelberg, Berlin, 1972.

[58] Pierre Louis 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*, Philadelphia, PA, 1988. SIAM.

[59] Jan Mandel. Efficient domain decomposition preconditioning for the p-version finite element method in three dimensions. Technical report, University of Colorado at Denver, 1989.

[60] Jan Mandel. Iterative solvers by substructuring for the p-version finite element method. *Comput. Meth. Appl. Mech. Engin.*, 1989. To appear in a special issue as Proceedings of an International Conference on Spectral and High Order Methods, Como, Italy, June 1989.

[61] Jan Mandel. On block diagonal and Schur complement preconditioning. Technical report, University of Colorado at Denver, November 1989.

[62] Jan Mandel. Two-level domain decomposition preconditioning for the p-version finite element version in three dimensions. *Internat. J. Numer. Methods Engrg.*, 1989. To appear.

[63] Jan Mandel. Hierarchical preconditioning and partial orthogonalization for the p-version finite element method. In Tony F. Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989*, Philadelphia, PA, 1990. SIAM.

[64] Jan Mandel and Steve McCormick. Iterative solution of elliptic equations with refinement: The two-level case. In Tony Chan, Roland Glowinski, Jacques Périaux, and Olof Widlund, editors, *Domain Decomposition Methods*, Philadelphia, PA, 1989. SIAM.

[65] Tarek P. Mathew. Domain decomposition and iterative refinement methods for mixed finite element discretisations of elliptic problems. Technical Report 463, Computer Science Department, Courant Institute of Mathematical Sciences, September 1989. Courant Institute doctoral dissertation.

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

[67] Carlo Miranda. *Partial Differential Equations of Elliptic Type.* Springer-Verlag, New York, 1970. 2nd Revised Edition, Translated by Zane C. Motteler.

[68] Segei V. Nepomnyaschikh. *Domain Decomposition and Schwarz Methods in a Subspace for the Approximate Solution of Elliptic Boundary Value Problems.* PhD thesis, Computing Center of the Siberian Branch of the USSR Academy of Sciences, Novosibirsk, USSR, 1986.

[69] Jindřich Nečas. *Les méthodes directes en théorie des équations elliptiques.* Academia, Prague, 1967.

[70] J. A. Nitsche. On Korn's second inequality. *Analyse Numérique*, 15(3):237–248, 1981.

[71] J. T. Oden and J. N. Reddy. *An Introduction to the Mathematical Theory of Finite Elements.* John Wiley and Sons, New York, 1982.

[72] M.E.G. Ong. Hierarchical basis preconditioners for second order elliptic problems in three dimensions. Technical Report 89–3, Dept. of Applied Math. University of Washington, Seattle, 1989.

[73] James M. Ortega. *Introduction to Parallel and Vector Solution of Linear Systems.* Plenum Press, New York, 1988.

[74] J. S. Przemieniecki. *Theory of Matrix Structural Analysis.* Dover Publicatins, Inc., New York, 1985. Reprint of McGraw Hill, 1968.

[75] H. A. Schwarz. *Gesammelete Mathematische Abhandlungen*, volume 2, pages 133–143. Springer, Berlin, 1890. First published in Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, volume 15, 1870, pp. 272–286.

[76] Barry F. Smith. A domain decomposition algorithm for elliptic problems in three dimensions. Technical Report 519, Department of Computer Science, Courant Institute, October 1990.

[77] Barry F. Smith. An optimal domain decomposition preconditioner for the finite element solution of linear elasticity problems. *SIAM J. Sci. Stat. Comput.*, 1990. To appear.

[78] Barry F. Smith and Olof B. Widlund. A domain decomposition algorithm based on a change to a hierarchical basis. *SIAM J. Sci. Stat. Comput.*, 11(6), 1990. To appear.

[79] S. L. Sobolev. L'Algorithme de Schwarz dans la Théorie de l'Elasticité. *Comptes Rendus (Doklady) de l'Académie des Sciences de l'URSS*, IV((XIII) 6):243–246, 1936.

[80] Gilbert Strang. Approximation in the finite element method. *Numer. Math.*, 19:81–98, 1972.

[81] Gilbert Strang and George J. Fix. *An Analysis of the Finite Element Method.* Prentice-Hall, Englewood Cliffs, N.J., 1973.

[82] T. E. Tezduyar and J. Liou. Element-by-element and implicit-explicit finite element formulations for computational fluid dynamics. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1988. SIAM.

[83] Richard S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, 1962.

[84] Olof B. Widlund. Iterative substructuring methods: Algorithms and theory for elliptic problems in the plane. In Roland Glowinski, Gene H. Golub, Gérard A. Meurant, and Jacques Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1988. SIAM.

[85] Olof B. Widlund. On the rate of convergence of the classical Schwarz alternating method in the case of more than two subregions. Technical report,

Department of Computer Science, Courant Institute, 1989. A revised version in preparation.

[86] Jinchao Xu. Iterative methods by space decompositions and subspace correction: a unifying approach. Technical report, Penn State University, University Park, PA, 1990. Submitted to Siam Review.

[87] Harry Yserentant. Hierarchical bases give conjugate gradient type methods a multigrid speed of convergence. *Appl. Math. Comp.*, 19:347–358, 1986.

[88] Harry Yserentant. On the multi-level splitting of finite element spaces. *Numer. Math.*, 49:379–412, 1986.

[89] Harry Yserentant. Two preconditioners based on the multi-level splitting of finite element spaces. Technical report, Institut für Mathematik, Universität Dortmund, Dortmund, Germany, 1990.