Parallel Multilevel Preconditioners*

James H. Bramble†
Joseph E. Pasciak**
Jinchao Xu‡

1. Introduction

In this paper, we shall report on some techniques for the development of preconditioners for the discrete systems which arise in the approximation of solutions to elliptic boundary value problems. These techniques are analyzed in [**11**]. Here we shall only state the resulting theorems; complete proofs can be found in [**11**].

It has been demonstrated that preconditioned iteration techniques often lead to the most computationally effective algorithms for the solution of the large algebraic systems corresponding to boundary value problems in two and three dimensional Euclidean space (cf. [**3**] and the included references). The use of preconditioned iteration will become even more important on computers with parallel architecture. This paper discusses an approach for developing completely parallel multilevel preconditioners. In order to illustrate the resulting algorithms, we shall describe the simplest application of the technique to a model elliptic problem. Let Ω be a polygonal domain in R^2 and consider the problem of approximating the solution u of

$$\begin{aligned} Lu &= f \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{1.1}$$

where

$$Lu = -\sum_{i,j=1}^{2} \frac{\partial}{\partial x_i} a_{ij} \frac{\partial u}{\partial x_j} + au.$$

We assume that the matrix $\{a_{ij}(x)\}$ is symmetric and uniformly positive definite and $a(x) \geq 0$ in Ω.

*This manuscript has been authored under contract number DE-AC02-76CH00016 with the U.S. Department of Energy. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. This work was also supported in part under the National Science Foundation Grant No. DMS84-05352 and by the U.S. Army Research Office through the Mathematical Science Institute, Cornell University.

†Cornell University, Ithaca, NY 14853.

**Brookhaven National Laboratory, Upton, NY 11973.

‡Pennsylvania State University, University Park, PA 16802.

We first define a sequence of multilevel finite element spaces in the usual way. Since Ω is polygonal, we can define a 'coarse' triangulation $\tau_1 = \cup_l \tau_1^l$ where τ_1^l represents an individual triangle and τ_1 denotes the triangulation. Successively finer triangulations $\{\tau_k, k = 2, \ldots, J\}$ are defined by breaking each triangle of a coarser triangulation into four triangles by connecting the midpoints of the edges. The space $\mathcal{M}_k$ is defined to be the continuous functions defined on Ω which are piecewise linear with respect to τ_k and vanish on $\partial\Omega$. We shall be interested in developing a preconditioner for the solution of the Galerkin equations on the J'th subspace, i.e. $U \in \mathcal{M}_J$ satisfying

$$A(U,\phi) = (f,\phi), \qquad \text{for all } \phi \in \mathcal{M}_J. \tag{1.2}$$

Here $A(\cdot,\cdot)$ denotes the generalized Dirichlet integral defined by

$$A(u,v) = \sum_{i,j=1}^{2} \int_\Omega a_{ij} \frac{\partial u}{\partial x_i} \frac{\partial v}{\partial x_j}\, dx + \int_\Omega auv\, dx \tag{1.3}$$

and $(\cdot,\cdot)$ denotes the L^2 inner product on Ω.

Let $\{\phi_k^l\}$ denote the usual nodal basis for the subspace $\mathcal{M}_k$, i.e. the l'th basis function is one on the l'th node of k'th triangulation and vanishes on all others. The preconditioner $\mathcal{B}$ is defined by

$$\mathcal{B}v = \sum_{k=1}^{J} \sum_{l} (v, \phi_k^l)\phi_k^l. \tag{1.4}$$

The above preconditioner is simply a double sum, the terms of which can be computed concurrently. This results in an inherently parallel algorithm.

As is well known, the rate of convergence of an iterative method can be estimated in terms of the condition number of the preconditioned system. In [**11**], a theory for the estimation of the condition number for this type of multilevel preconditioner in terms of a number of *a priori* assumptions is given. In the above example, this theory can be used to show that the relevant condition number is at worst $O(J^2)$. Moreover, these results hold for problems in two, three and higher dimensions as well as problems with only locally quasi-uniform mesh approximation.

We note that many alternative preconditioning techniques have been proposed for such discrete systems. For example, domain decomposition preconditioners have been developed ([5],[6],[7],[8],[14], and the included references). These domain decomposition preconditioners are inherently parallel however become somewhat complex in three dimensional applications. Alternatively, multigrid [4],[9], [**15**],[**18**] and hierarchical multigrid [2],[**21**] techniques give rise to different multilevel preconditioners. The standard multigrid algorithms do not allow for completely parallel computations since the computations on a given level use results from the previous levels. Theoretical results for the usual multigrid algorithms are available, in general, for problems in any number of spatial dimensions but only for quasi-uniform mesh approximation. Good results hold for the hierarchical basis method in two dimensions with refined meshes but degenerate when applied to three dimensional problems. Finally, preconditioners based on approximate LU factorization are often proposed however a comprehensive theory is yet to be developed [**12**],[**13**], [**19**].

The outline of the remainder of the paper is as follows. A general abstract theory for the development of parallel multilevel preconditioners is discussed in Section 2. In Section 3, this theory is applied to second order elliptic boundary value problems and the serial and parallel complexity of the resulting algorithms is discussed. We apply the abstract theory to a second order problem with a locally refined mesh in Section 4. Finally, the results of numerical experiments illustrating the theory of the earlier sections are given in Section 5.

2. General theory.

In this section, we discuss a general theory for the construction of parallel multilevel preconditioners. This theory is presented in an abstract setting to most clearly illustrate the relevant analytic assumptions. The development of this class of preconditioners is based on a certain orthogonal decomposition of the approximation space. The parallel multilevel preconditioners are then abstractly defined in terms of this decomposition by the replacement of orthogonal projections by more computationally efficient operators. Applications to second order elliptic boundary value problems are given in Sections 3 and 4.

We start with the basic abstract framework. We assume that we are given a nested sequence of finite dimensional spaces,

$$\mathcal{M}_1 \subset \mathcal{M}_2 \subset \ldots \subset \mathcal{M}_J \equiv \mathcal{M}, \quad J \geq 2. \tag{2.1}$$

The space $\mathcal{M}$ and hence all of its subspaces are equipped with two inner products $(\cdot,\cdot)$ and $A(\cdot,\cdot)$. The first part of this section will consider properties of a certain orthogonal decomposition of $\mathcal{M}$ with respect to the inner product $(\cdot,\cdot)$ and the sequence of spaces (2.1). We shall investigate the spectral properties of these spaces with respect to the form $A(\cdot,\cdot)$ since, ultimately, we are interested in computing the solution to the Galerkin equations: Given $f \in \mathcal{M}$, find $U \in \mathcal{M}$ satisfying

$$A(U,v) = (f,v) \qquad \text{for all } v \in \mathcal{M}. \tag{2.2}$$

We shall use the following notation in the development. For each $k = 1,\ldots,J$, we introduce the following operators:

(1) The projection $P_k : \mathcal{M} \longrightarrow \mathcal{M}_k$ is defined for $u \in \mathcal{M}$ by

$$A(P_k u, v) = A(u,v), \qquad \text{for all } v \in \mathcal{M}_k.$$

(2) The projection $Q_k : \mathcal{M} \longrightarrow \mathcal{M}_k$ is defined for $u \in \mathcal{M}$ by

$$(Q_k u, v) = (u,v), \qquad \text{for all } v \in \mathcal{M}_k.$$

(3) The operator $A_k : \mathcal{M}_k \longrightarrow \mathcal{M}_k$ is defined for $u \in \mathcal{M}_k$ by

$$(A_k u, v) = A(u,v), \qquad \text{for all } v \in \mathcal{M}_k.$$

We shall also denote $A = A_J$ and define

$$\mathcal{O}_k = \{\phi | \phi = (Q_k - Q_{k-1})\psi\,,\, \psi \in \mathcal{M}\},$$

where $Q_0 = 0$. We shall discuss the spectral properties of A with respect to the decomposition

$$\mathcal{M} = \mathcal{O}_1 + \cdots + \mathcal{O}_J. \tag{2.3}$$

It follows from the above definitions that

$$\begin{aligned} Q_k A &= A_k P_k \\ Q_k Q_l &= Q_l Q_k = Q_l \ \text{ for } \ l \le k. \end{aligned} \tag{2.4}$$

From the second equation of (2.4), it follows that

$$(Q_k - Q_{k-1})(Q_l - Q_{l-1}) = 0$$

if $k \neq l$ and hence the decomposition (2.3) is orthogonal, i.e. $(u, v) = 0$ whenever $u \in \mathcal{O}_l$, $v \in \mathcal{O}_k$ with $l \neq k$.

We consider first the operator

$$B = \sum_{k=1}^{J} \lambda_k^{-1}(Q_k - Q_{k-1}), \tag{2.5}$$

where λ_k denotes the spectral radius of A_k. Clearly, B is symmetric and positive definite and

$$A(BAv, v) = \sum_{k=1}^{J} \lambda_k^{-1} \left\|(Q_k - Q_{k-1})Av\right\|^2, \tag{2.6}$$

where $\|\cdot\|^2 = (\cdot,\cdot)$. Note that B is block diagonal with respect to the decomposition (2.3) and each diagonal block is a multiple of the identity matrix.

The operator B may be thought of as an "approximate inverse" for A. Thus, we shall state theorems estimating the condition number $K(BA)$ of BA. We note that $K(BA) \le c_1/c_0$ for any positive constants c_0, c_1 satisfying

$$c_0 A(v, v) \le A(BAv, v) \le c_1 A(v, v), \qquad \text{for all } v \in \mathcal{M}. \tag{2.7}$$

REMARK 2.1: The form of the operator B can be motivated by the spectral decomposition of the operator A. Indeed, for a special example, namely, $\mathcal{M}_k$ the space spanned by the eigenvectors corresponding to the smallest k distinct eigenvalues of A, the operator B defined by (2.5) is in fact equal to A^{-1}.

It is straightforward to show that (cf. [11])

$$A(BAv, v) \le JA(v, v), \qquad \text{for all } v \in \mathcal{M}. \tag{2.8}$$

The lower estimate of (2.7) will require some additional hypotheses concerning the spaces $\mathcal{M}_k$. We first consider the following assumptions on the operators Q_k: For $k = 1, \ldots, J$, there exists a constant $C_1 > 0$ such that

$$\left\|(I - Q_{k-1})v\right\|^2 \le C_1 \lambda_k^{-1} A(v, v), \qquad \text{for all } v \in \mathcal{M}. \tag{A.1}$$

We have the following theorem and corollaries.

Theorem 1. *Assume that (A.1) holds. Then*

$$C_1^{-1}J^{-1}A(v,v) \le A(BAv,v) \le JA(v,v), \qquad \textit{for all } v \in \mathcal{M}. \tag{2.9}$$

Corollary 1. *For any real s,*

$$B^s = \sum_{k=1}^{J} \lambda_k^{-s}(Q_k - Q_{k-1}). \tag{2.10}$$

Moreover, for any $s \in [0,1]$,

$$J^{-s}(A^s v,v) \le (B^{-s}v,v) \le (C_1 J)^s (A^s v,v), \qquad \textit{for all } v \in \mathcal{M}. \tag{2.11}$$

We have included Corollary 1 for the purpose of future applications which will not be described in this paper. In particular, it will be used for the development of preconditioners for certain boundary operators which arise in domain decomposition techniques for second order boundary value problems [**10**].

In the next corollary, we consider the case of the sum of two operators. Let $\hat{A}(\cdot,\cdot)$ be another symmetric positive definite form and let $\hat{A}$, $\{\hat{A}_k\}$ and $\{\hat{\lambda}_k\}$ be defined analogously in terms of $\hat{A}(\cdot,\cdot)$. Consider the operator $\bar{B} : \mathcal{M} \mapsto \mathcal{M}$ defined by

$$\bar{B} = \sum_{k=1}^{J} (\lambda_k + \hat{\lambda}_k)^{-1}(Q_k - Q_{k-1}).$$

Theorem 1 immediately implies the following corollary.

Corollary 2. *Assume that (A.1) holds for both A and $\hat{A}$. Then,*

$$J^{-1}((A+\hat{A})v,v) \le (\bar{B}^{-1}v,v) \le C_1 J((A+\hat{A})v,v), \qquad \textit{for all } v \in \mathcal{M}.$$

The most natural application of the above corollary is to the discrete systems which arise in parabolic time stepping algorithms. At each time level, a function $U^n \in \mathcal{M}$ satisfying

$$(I + \tau A)U^n = F^n,$$

with known $F^n \in \mathcal{M}$ must be computed. Here τ is a positive number which is related to the time step size. We shall not consider further application of Corollary 2 in this paper.

We next apply the above results to analyze parallel multilevel preconditioners for A. An operator $\mathcal{B} : \mathcal{M} \mapsto \mathcal{M}$ is a good preconditioner for A if it satisfies:

(1) The action of $\mathcal{B}$ on vectors of $\mathcal{M}$ is economical to compute.
(2) The condition number $K(\mathcal{B}A)$ of the preconditioned system is not too large.

Item (1) above guarantees that the cost per iteration in a preconditioned scheme using $\mathcal{B}$ for solving (2.2) will not be unreasonable. Item (2) guarantees that the number of iterations in a preconditioned scheme will not be too large. Note that by Theorem 1, B satisfies (2). B may in fact satisfy (1) in many applications but generally it is desirable to avoid evaluating the action of Q_k. Hence we shall develop more computationally effective algorithms by modifying (2.5).

To get a computationally effective preconditioner, we write (2.5) in the form

$$B = \sum_{k=1}^{J-1}(\lambda_k^{-1} - \lambda_{k+1}^{-1})Q_k + \lambda_J^{-1}I.$$

Notice that if $\{\lambda_k\}_{k=1}^{J}$ satisfies the growth condition $\lambda_{k+1} \geq \sigma\lambda_k$, for $\sigma > 1$ then the operator

$$\hat{B} = \sum_{k=1}^{J} \lambda_k^{-1}Q_k \tag{2.12}$$

satisfies

$$(1 - \sigma^{-1})(\hat{B}u, u) \leq (Bu, u) \leq (\hat{B}u, u), \qquad \text{for all } u \in \mathcal{M}.$$

We consider a slightly more general operator defined by replacing $\lambda_k^{-1}I$ in (2.12) with a symmetric positive definite operator $R_k : \mathcal{M}_k \mapsto \mathcal{M}_k$, i.e.

$$\mathcal{B} = \sum_{k=1}^{J} R_kQ_k. \tag{2.13}$$

Clearly, $\mathcal{B}$ is symmetric and positive definite on $\mathcal{M}$. The cost of evaluating the action of the preconditioner $\mathcal{B}$ on a vector in $\mathcal{M}$ will be discussed in later sections but will obviously depend on an appropriate choice of R_k.

For our subsequent development, we shall need to make the following assumption concerning the operator R_k. We assume that

$$C_2\frac{\|u\|^2}{\lambda_k} \leq (R_ku, u) \leq C_3(A_k^{-1}u, u), \qquad \text{for all } u \in \mathcal{M}_k, \tag{A.2}$$

where C_2 and C_3 are positive constants not depending on J. Clearly the choice $R_k = \lambda_k^{-1}I$ corresponding to (2.12) satisfies (A.2).

The preconditioner (2.13) can be thought of as a parallel version of a V-cycle multigrid algorithm. The operator R_k plays the role of a smoothing procedure. The major difference between (2.13) and the V-cycle multigrid scheme is that the smoothing on every level of (2.13) is applied to the original fine grid residual. In contrast, the multigrid V-cycle applies the smoothing to the residual computed using the corrections from the previously visited grid. Obviously, the different terms in (2.13) can be computed in parallel while, in contrast, computations on a given grid level in a standard multigrid algorithm must wait for the results from previous levels. The connection between (2.13) and the multigrid V-cycle will be more fully discussed in Section 3. However, it is not surprising that assumptions which are equivalent to (A.2) have been made in the analysis of the usual serial multigrid algorithms [4],[9],[16],[17].

REMARK 2.2: A particularly interesting choice of R_k can be motivated as follows. As noted above, $R_k = \lambda_k^{-1} I$ satisfies (A.2). Let $\{\psi_k^l\}$ be an orthonormal basis for $\mathcal{M}_k$. Then

$$\lambda_k^{-1} u = \lambda_k^{-1} \sum_l (u, \psi_k^l) \psi_k^l, \qquad \text{for all } u \in \mathcal{M}_k. \tag{2.14}$$

In practice, an orthonormal basis for $\mathcal{M}_k$ is seldom available. However, for finite element applications with quasi-uniform grids, the right hand side of (2.14) with normalized nodal basis functions $\{\bar{\psi}_k^l\}$ defines an R_k satisfying (A.2) (see Section 3). Moreover, we note that for $u \in \mathcal{M}$,

$$R_k Q_k u = \lambda_k^{-1} \sum_l (u, \bar{\psi}_k^l) \bar{\psi}_k^l$$

and hence $R_k Q_k$ is computable without the solution of gram matrix systems. This will be discussed in more detail in Section 3.

With $\mathcal{B}$ defined in (2.13), we have the following corollary.

Corollary 3. *Under assumptions (A.1) and (A.2),*

$$C_1^{-1} C_2 J^{-1} A(v, v) \le A(\mathcal{B} A v, v) \le C_3 J A(v, v), \qquad \textit{for all } v \in \mathcal{M}. \tag{2.15}$$

We next provide an alternative hypothesis for a lower estimate in (2.15). This is the so called "regularity and approximation" assumption often used in multigrid analysis (cf. [4],[15],[18]). We assume that for a fixed $\alpha \in (0, 1]$, there exists a positive constant C_4 not depending on $k = 1, \dots, J$ satisfying

$$A((I - P_{k-1})v, v) \le (C_4 \lambda_k^{-1} \|A_k v\|^2)^\alpha A(v, v)^{1-\alpha}, \qquad \text{for all } v \in \mathcal{M}_k, \tag{A.3}$$

where $P_0 = 0$. In finite element applications, the above assumption is usually proved by using elliptic regularity for the continuous problem and the approximation properties of the space $\mathcal{M}_{k-1}$ [1],[4]. In such applications, assumption (A.3) may be stronger than (A.1), e.g. when $\alpha = 1$, (A.3) implies (A.1).

Theorem 2. *Assume that (A.2) and (A.3) hold. Then*

$$C_2 C_4^{-1} J^{1-1/\alpha} A(v, v) \le A(\mathcal{B} A v, v) \le C_3 J A(v, v), \qquad \textit{for all } v \in \mathcal{M}.$$

Remark 2.3: Included in (A.1) and (A.3) is the implicit assumption that C_1 and C_4 are greater than or equal to $K(A_1)$. In finite element applications, $K(A_1)$ will not be large if the grid size of the coarsest grid is of unit size. However, if a good preconditioner R_j is available for any finer grid, i.e. R_j satisfies in addition,

$$(R_j^{-1} u, u) \le C_5 (A_j u, u), \tag{2.16}$$

then it suffices to use

$$\mathcal{B} = \sum_{k=j}^{J} R_k Q_k.$$

In such applications, (A.1) or (A.3) need only be satisfied for $k > j$. Note that $R_j = A_j^{-1}$ will be computationally economical provided that the j'th grid size is relatively small. Many alternative choices are possible.

3. The quasi-uniform finite element application.

In this section, we shall illustrate the application of the abstract theory and algorithms discussed in the previous section to a second order elliptic boundary value problem approximated using finite element functions on a quasi-uniform mesh. We note that the hypotheses of the previous section are satisfied. We also consider the computational complexity of the resulting algorithm in both serial and parallel computing applications. For brevity, we consider only the most basic finite element applications. Many other applications are possible including examples of elliptic problems in higher dimensions.

Let $\mathcal{M}_1 \subset \cdots \subset \mathcal{M}_J \equiv \mathcal{M}$ be the finite element spaces defined in the introduction subsequent to (1.1), $A(\cdot,\cdot)$ be the generalized Dirichlet form defined in (1.3) and $(\cdot,\cdot)$ be the L^2 inner product on Ω.

We will apply the results stated in Section 2 to Problem (1.2) with the above sequence of spaces. Let h_k denote the size of the k'th triangulation. It easily follows that there are constants c_0 and c_1, not depending on k and satisfying

$$c_0 h_k^{-2} \le \lambda_k \le c_1 h_k^{-2}. \tag{3.1}$$

Inequality (A.1) with $k > 2$ is well known. For $k = 1$, we have that

$$\|v\|^2 \le \Lambda^{-1} A(v,v), \qquad \text{for all } v \in \mathcal{M}$$

where Λ is the smallest eigenvalue of A and is obviously bounded from away from zero (independently of J). We shall suppose in this application that $\mathcal{M}_1$ is such that h_1 is proportional to the diameter of Ω so that $C_1 \ge \lambda_1/\Lambda$ which is not large.

We next consider the operator R_k motivated by Remark 2.2, i.e.

$$R_k v = \sum_l (v, \phi_k^l)\phi_k^l, \text{ for } v \in \mathcal{M}_k, \tag{3.2}$$

where the sum is taken over all nodes of τ_k. As observed in Remark 2.2, the action of $R_k Q_k$ can be computed without explicitly computing Q_k. Moreover, using R_k defined by (3.2) in (2.13) leads to the preconditioner of (1.4) It is shown in [11]that (A.2) holds for this R_k.

For this problem, (A.3) will always be satisfied for some $\alpha \in (0,1]$, (cf. [1],[4]). The size of α depends on the elliptic regularity of Problem (1.1). Thus, in the case when Ω is a convex polygonal domain and the coefficients defining L are smooth, (A.3) holds with $\alpha = 1$ and we conclude from Theorem 2 that

$$K(\mathcal{B}A) \leq cJ.$$

In the case of a so called crack problem (with smooth coefficients), the largest interior angle is 2π and the regularity of (1.1) is such that (A.3) does not hold for $\alpha \geq 1/2$. Hence Corollary 3 yields the better estimate and shows that

$$K(\mathcal{B}A) \leq CJ^2.$$

Remark 3.1: It is possible to apply the theory of Section 2 to elliptic problems in three or more dimensions. Many examples are possible but we consider only the simplest. In three dimensions, we let the coarse mesh be a union of equally sized cubes. Finer meshes are obtained by breaking each cube of a coarser mesh into eight smaller cubes in the obvious way. The subspaces $\{\mathcal{M}_k\}$ are defined to be the functions on Ω which are continuous and piecewise trilinear with respect to the k'th mesh and vanish on $\partial\Omega$. The nodes of these spaces are the vertices of the cubes defining the mesh. We may take

$$\mathcal{B}u = \sum_{k=1}^{J} h_k^{-1} \sum_{l} (u, \phi_k^l)\phi_k^l, \tag{3.3}$$

where $\{\phi_k^l\}$ denotes the set of nodal basis functions. We emphasize here again that all the terms in (3.3) are independent and hence may be computed concurrently.

Remark 3.2: Assumption (A.1) is often easier to verify than (A.3). For example, we consider the two dimensional problem (1.1) when the coefficients of the operator L are discontinuous. If the jumps in the coefficients are only along the lines of the coarse mesh, then it is possible to prove that (A.1) holds with $C_1 \leq CJ$ where the constant C depends on the local variation of the coefficients of L on the coarse grid triangles but not on the magnitude of the jumps across triangles [**20**]. This leads to a conditioning result of the form

$$K(\mathcal{B}A) \leq CJ^3.$$

The dependence of constant C_4 (in (A.3)) on the size of the jumps is a much more difficult question since it requires the knowledge of the dependence of the elliptic regularity constants on such jumps.

In the remainder of this section, we consider computational issues involved in implementing the above algorithm in serial and parallel computing architectures. However, before proceeding, we make the following observation. Even though we have defined $\mathcal{B}$ as an operator on $\mathcal{M}$, in a preconditioned iterative scheme we are only required to compute $\mathcal{B}v$ given the data $W_J^l = (v, \phi^l{}_J)$. This is because when $v = A_J\theta$, we always compute $\{(A_J\theta, \phi^l{}_J) = A(\theta, \phi^l{}_J)\}$ and hence avoid the solution of the gram matrix problem required for the computation of $A_J\theta$.

We first consider the serial version of the algorithm. Let $v \in \mathcal{M}$ be given and define $W_k^l = (v, \phi_k^l)$. Let W_k denote the vector with entries $(W_k)_l = W_k^l$. We need to compute the action of $\mathcal{B}v$ given W_J. We define W_{k-1} from W_k in a recursive manner. Note that each basis function in $\mathcal{M}_{k-1}$ can be written as a local linear combination of basis functions for $\mathcal{M}_k$. Thus, each value of W_{k-1}^l can be written as a local linear combination of values of W_k. Moreover, the work involved in computing W_{k-1} from W_k is proportional to the number of unknowns in $\mathcal{M}_{k-1}$. Consequently, the work involved in computing the vectors $\{W_k\}$, $k = 1, \ldots, J$ bounded by a constant times the number of unknowns in $\mathcal{M}$. Once the vectors $\{W_k\}$ are known, we are left to compute the representation of $\mathcal{B}v$ in the basis for $\mathcal{M}$. To do this, we compute the representation of

$$\mathcal{B}_m v \equiv \sum_{k=1}^{m} \sum_{l} (v, \phi_k^l) \phi_k^l,$$

in the basis for $\mathcal{M}_m$, for $m = 1, \ldots, J$. The result at $m = J$ is of course the basis representation for $\mathcal{B}v$. For $m = 1$, the representation is already given by W_1. The representation of $\mathcal{B}_m v$, for $m > 1$ is calculated from that of $\mathcal{B}_{m-1}v$ by interpolating the $\mathcal{B}_{m-1}v$ results (i.e. expanding them in terms of the m'th basis) and adding the m'th level contribution from W_m. The work of calculating the representation of $\mathcal{B}_m v$ given that for $\mathcal{B}_{m-1}v$ is on the order of the number of unknowns in $\mathcal{M}_m$ and thus the total work for this algorithm is bounded by a constant times the number of unknowns on the finest grid.

Remark 3.3: The serial implementation of the operator $\mathcal{B}$ is closely related to the multigrid V-cycle algorithm. The step of computing W_{k-1} from W_k is nothing more that the step which "transfers the residuals" from grid level k to $k-1$ in a multigrid V-cycle algorithm. However, the multigrid algorithm requires extra computation since it must smooth and then compute new residuals on the k'th level before transferring. The second step in the serial algorithm for $\mathcal{B}$ is also duplicated in the "coarser to finer interpolation" step in the multigrid V-cycle algorithm. The symmetric multigrid V-cycle requires extra computation since it requires additional smoothing on each grid level. Thus the serial $\mathcal{B}$ algorithm, in terms of complexity, is similar to a multigrid V-cycle algorithm without smoothing.

We next consider parallel implementation of the preconditioner $\mathcal{B}$. The execution of (1.4) can obviously be made parallel in many ways by breaking up the terms into various numbers of parallel tasks. The optimal splitting of the sum is clearly dependent on characteristics of the individual parallel computer, for example, memory management considerations, task initialization overhead, the number of parallel processors, etc. We note, however, the simplicity of the form of (1.4) allows for almost complete freedom for parallel splitting.

It is of theoretical interest to consider the algorithm on a shared memory machine with an unlimited number of processors. As above, the implementation $\mathcal{B}v$ involves two steps, the calculation of the coefficients W_k^l and the computation of the representation of $\mathcal{B}v$ in the basis for $\mathcal{M}$. Each coefficient can be computed independently and involves a linear combination (not necessarily local) of the values of W_J. With enough processors, a linear combination of m numbers can be computed in $\log_2(m)$ time. Hence the coefficient vectors $\{W_k\}$ can be computed in $\log_2(N)$ time where N is the dimension of $\mathcal{M}$. Each coefficient of $\mathcal{B}v$ involves a linear combination of $M_n J$ contributions from the J grid levels (here M_n is the maximum number of neighbors for any given level. Thus, computation of $\mathcal{B}v$ can in done in time bounded by CJ.

4. A local refinement application.

In this section, we shall discuss the application of the parallel multilevel algorithm to the finite element equations corresponding to a problem with mesh refinement. Such mesh refinements are necessary for accurate modeling of problems with various type of singular behavior. For simplicity, we shall make no attempt at generality. Instead, we shall illustrate the technique by considering an example from which many obvious generalizations are possible. For this example, the domain Ω will be the unit square and we shall approximate the solution to (1.1). The form $A(\cdot,\cdot)$ and the inner product $(\cdot,\cdot)$ will be as in Section 3. The sequence of grids which we shall consider will be progressively more refined as we approach the corner (1,1). Such a mesh would be effective if, for example, the function f in (1.1) behaved like a δ function distribution at the point (1,1).

To define the mesh, we first start with a sequence of subspaces $\mathcal{M}_1,\ldots,\mathcal{M}_j$ defined using uniform grids of size $h_k = 2^{-k}$, $k = 1,\ldots,j$ as described in the quasi-uniform case (See Section 1). The $j+1$'st triangulation is then defined by refining only those triangles in the upper quarter, $[1/2,1]\times[1/2,1]$. Similarly, the $j+2$'nd triangulation is defined by refining only those triangles in the $j+1$'st grid which are in the region $[3/4,1]\times[3/4,1]$, etc. Accordingly, we shall denote $\Omega_k = (1-2^{j-k},1)\times(1-2^{j-k},1)$ and define the spaces $\mathcal{M}_k$ for $k = j+1,\ldots,J$ to be the continuous functions on Ω which are piecewise linear with respect to the k'th grid. Note that this introduces slave nodes into the computation, i.e. the vertices of the triangles on the boundary of the k'th refinement region which are not nodes for the $k-1$'st subspace. These nodes are slaves since the values of functions on these nodes are determined by the values of neighboring nodes and the continuity condition on the subspace. Thus, they do not represent degrees of freedom in the subspace. It is shown in [11], that (A.1) is satisfied for this sequence of subspaces.

We next define a sequence of operators $\{R_k\}$ satisfying (A.2). For $k \le j$, R_k is given by (3.2). Let $\{x_k^l\}$ denote the nodes of the k'th grid and let $\{\phi_k^l\}$ denote the corresponding nodal basis functions. For each node x_k^l with $k > j$ we define

$$h_{kl} = \begin{cases} h_k & \text{if } x_k^l \in \bar{\Omega}_k, \\ h_m & \text{if } x_k^l \in \bar{\Omega}_m/\bar{\Omega}_{m+1},\ j \le m < k. \end{cases}$$

Note that if $x_k^l \in \bar{\Omega}_k/\bar{\Omega}_{k+1}$ then x_k^l is a node for each finer subspace and gets assigned the same value h_k. We then define

$$R_k u = h_k^2 \sum_l h_{kl}^{-2}(u,\phi_k^l)\phi_k^l. \tag{4.1}$$

We can apply Corollary 3 to show that $K(\mathcal{B}A) \le CJ^2$ where $\mathcal{B}$ is defined by (2.13) with R_k and $\mathcal{M}_k$ as above. For this application, we have not been able to prove the regularity and approximation assumption (A.3).

For the purpose of implementation, it is more efficient to reorder the terms defining $\mathcal{B}$. For $k = j, \dots , J$ let $\mathcal{N}_k$ be the nodes of $\mathcal{M}_k$ in $\bar{\Omega}_k$ and for $k < J$ let $\mathcal{N}_k^1$ be the nodes of $\mathcal{M}_k$ in $\bar{\Omega}_k/\bar{\Omega}_{k+1}$. For a function $u \in \mathcal{M}$, it is not difficult to see by induction on J that

$$\begin{aligned}\mathcal{B}u = \sum_{k=1}^{j-1} R_k Q_k u + \sum_{x_J^l \in \mathcal{N}_J} (u, \phi_J^l)\phi_J^l \\ + \sum_{k=j}^{J-1}\Bigg[\sum_{x_k^l \in \mathcal{N}_k^1} \gamma_k^J (u, \phi_k^l)\phi_k^l + \sum_{x_k^l \in \mathcal{N}_k/\mathcal{N}_k^1} (u, \phi_k^l)\phi_k^l \Bigg]\end{aligned} \qquad (4.2)$$

where $\gamma_k^J = h_k^{-2} \sum_{m=k}^{J} h_m^2$. Note that the R_k terms in the first sum of (4.2) involves the same sums which appear in the uniform case of Section 3. In addition, the calculation corresponding to the k'th mesh in (4.2) for $k = j, \dots , J$ only involves nodal basis functions on $\bar{\Omega}_k$.

Finally we define a simpler preconditioner $\hat{\mathcal{B}}$ by replacing γ_k^J by one in (4.2), i.e.

$$\hat{\mathcal{B}}u = \sum_{k=1}^{j} \sum_{l} (u, \phi_k^l)\phi_k^l + \sum_{k=j+1}^{J} \sum_{x_k^l \in \mathcal{N}_k} (u, \phi_k^l)\phi_k^l. \qquad (4.3)$$

Note that in (4.3), the k'th refinement grid only adds a sum over the nodes in $\bar{\Omega}_k$. We note that for $u \in \mathcal{M}$, by (4.2)

$$\begin{aligned}(\mathcal{B}u, u) = \sum_{k=1}^{j-1} \sum_{l} (u, \phi_k^l)^2 + \sum_{x_J^l \in \mathcal{N}_J} (u, \phi_J^l)^2 \\ + \sum_{k=j}^{J-1}\Bigg[\sum_{x_k^l \in \mathcal{N}_k^1} \gamma_k^J (u, \phi_k^l)^2 + \sum_{x_k^l \in \mathcal{N}_k/\mathcal{N}_k^1} (u, \phi_k^l)^2 \Bigg]\end{aligned}$$

with an analogous expression for $\hat{\mathcal{B}}$. Clearly, $1 \le \gamma_k^J \le 4/3$ from which it follows that

$$(\hat{\mathcal{B}}u, u) \le (\mathcal{B}u, u) \le \frac{4}{3}(\hat{\mathcal{B}}u, u), \qquad \text{for all } u \in \mathcal{M}.$$

From the discussion in Section 3, it is clear that the first sum in (4.3) is a preconditioner for the problem on $\mathcal{M}_j$, i.e. the finest uniform grid. As we shall see, this sum can also be replaced by any uniform preconditioner for A_j without adversely effecting the asymptotic behavior of the overall condition number. Indeed, let the operator R_j be a preconditioner for A_j (satisfying (2.16) and the second inequality of (A.2)) and define for $u \in \mathcal{M}$,

$$\hat{\mathcal{B}}u = R_j Q_j u + \sum_{k=j+1}^{J} \sum_{x_k^l \in \mathcal{N}_k} (u, \phi_k^l)\phi_k^l. \qquad (4.4)$$

Note that by Remark 2.3, the operator

$$\tilde{B}u = \sum_{k=j}^{J} R_k Q_k u$$

satisfies $K(\tilde{B}A) \leq C(J-j)^2$. It is shown in [11]that $\tilde{B}$ is uniformly equivalent to $\hat{B}$. Thus, $K(\hat{B}A) \leq C(J-j)^2$.

Remark 4.1: Clearly, we could generalize this example to include much more general refinements for problems in R^2 as well as higher dimensional space. Note that the refinement only changes the preconditioner $\hat{\mathcal{B}}$ (resp. $\hat{B}$) by adding additional terms in (4.3) (resp. (4.4)) involving nodes from the refinement region. Thus, this approach is well suited to dynamic adaptive refinement techniques. New refinement regions add terms to the sum whereas the "de-refinement" of existing regions only takes away terms from the sum. The operator $\hat{B}$ is even more useful in this context since it allows the easy inclusion of this refinement preconditioner into existing large scale uniform grid codes. Preconditioners for the uniform grid already available in the existing code can be used, supplemented with additional routines implementing the terms due to the refinement.

5. Numerical results

In this section, we provide the results of numerical examples illustrating the theory discussed in the earlier sections. To demonstrate the performance of the proposed algorithms, we shall provide numerical results for a two dimensional problem with full elliptic regularity and one with less than full elliptic regularity, a two dimensional example with a geometric mesh refinement and a three dimensional example. In all of the reported results, the experimentally observed behavior of the condition number of the preconditioned system was in agreement with the theorems presented earlier. In the first example, we also compare the results of the new method with those obtained using the hierarchical preconditioning method [21] and a classical V-cycle multigrid preconditioner[4].

For our first example, we consider Problem (1.1) when $L = -\Delta \equiv -\partial^2/\partial x_1^2 - \partial^2/\partial x_2^2$ and Ω is the unit square. This example satisfies the regularity and approximation assumption (A.3) for $\alpha = 1$ as well as (A.1).

We will use a finite element discretization of (1.1) and develop a sequence of grids in a standard way. To define the coarsest grid, we start by breaking the square into four smaller squares of side length 1/2 and then dividing each smaller square into two triangles by connecting the lower left hand corner with the upper right hand corner. Subsequently finer grids are developed as in the introduction, i.e., by dividing each triangle into the four triangles formed by the edges of the original triangle and the lines connecting the centers of these edges. The space $\mathcal{M}_i$ is defined to be the set of continuous functions on Ω which are piecewise linear on the i'th triangulation and vanish on $\partial\Omega$.

We shall compare three preconditioners for (1.2). The first preconditioner $\mathcal{B}$ is defined by the multilevel algorithm (2.13) with R_k given by (3.2) and fits into the framework considered in Section 3. For comparison, we also provide results for the hierarchical preconditioner B_H [21] and a preconditioner B_M defined by a standard symmetric V-cycle of multigrid [4]. The multigrid algorithm uses one sweep of Jacobi smoothing whenever a grid level is visited and hence results in two smoothing steps on each grid for each evaluation of the preconditioner. The multigrid uses $h_0 = 1/4$ for the coarsest grid while both the hierarchical and the parallel multilevel algorithms uses $h_0 = 1/2$.

Table 5.1 gives the condition numbers K of the preconditioned systems $B_H A$, $\mathcal{B}A$ and $B_M A$ corresponding, respectively, to the hierarchical preconditioner, the preconditioner defined by (2.13), and the V-cycle multigrid preconditioner. We note that for these examples, a preconditioned conjugate gradient algorithm using the new preconditioner would be expected to take twice as many iterations as the corresponding algorithm using the V-cycle of multigrid. However, even in a serial implementation, the multigrid algorithm involves substantially more computational effort per step. The new method outperforms the hierarchical preconditioner.

Table 5.1

Condition numbers when Ω is the square.

h_J	$K(B_H A)$	$K(\mathcal{B}A)$	$K(B_M A)$
1/16	19	7.0	2.3
1/32	31	8.1	2.4
1/64	43	9.0	2.4
1/128	58	9.8	2.4

This test problem illustrates an example where all three methods work reasonably well. However, we note that $\mathcal{B}$ is preferred over standard multigrid when the parallel aspects of the algorithm are important. In addition, $\mathcal{B}$ generalizes to higher dimensional problems without convergence rate deterioration (see Table 5.5) and hence would be preferred to the hierarchical method in three dimensional computations.

We next consider the above preconditioners on a problem with less than full elliptic regularity. We again consider (1.1) with L given by the Laplacian and Ω equal to the "slit domain", i.e. Ω is the set of points in the interior of the unit square excluding the line $\{(1/2, y) | y \in [1/2, 1)\}$. This example does not satisfy the *a priori* estimates used in the proof of the regularity and approximation assumption (A.3) for $\alpha \geq 1/2$. However, assumption (A.1) is satisfied.

Table 5.2 gives the condition numbers K of the preconditioned systems $B_H A$, $\mathcal{B}A$ and $B_M A$ corresponding, respectively, to the hierarchical preconditioner, the preconditioner defined by (2.13), and the V-cycle multigrid preconditioner. The results are in general agreement with the theoretical estimates

$$K(B_H A) \leq C \ln^2(1/h_J),$$
$$K(\mathcal{B}A) \leq C \ln^2(1/h_J),$$

for the respective methods.

Table 5.2

Condition numbers when Ω is the slit domain.

h_J	$K(B_H A)$	$K(\mathcal{B}A)$	$K(B_M A)$
1/16	14.6	7.9	2.6
1/32	25.17	10.0	2.9
1/64	38.2	12.6	3.1
1/128	53.8	14.9	3.4

We next provide numerical results for the refinement example of Section 4. We once again, consider the solution of (1.1) with L the Laplacian and Ω the unit square. The sequence of spaces $\mathcal{M}_1 \subset \cdots \subset \mathcal{M}_J$ are as developed in Section 4 and provide results for the preconditioner $\hat{\mathcal{B}}$ defined by (4.3). As noted in Section 4, some such refinement would be necessary if, for example, the function f had a δ-function behavior at the point (1,1). Table 5.3 gives the condition number of the preconditioned system $\hat{\mathcal{B}}A$ as a function of the mesh size of the uniform grid h_j and the number of refinement levels l. The size of the finest triangle can be computed by dividing the uniform mesh size by 2^l. In all of the runs, the coarsest grid level corresponded to $h_0 = 1/2$. The numerical results seem to indicate that an increase in the number of uniform levels has a greater effect on the condition number than an increase in the number of refinement levels.

Table 5.3

Condition numbers for the refinement example.

h_j	$l = 1$	$l = 2$	$l = 3$	$l = 4$
1/8	6.3	6.5	6.7	6.9
1/16	7.7	7.9	8.05	8.1
1/32	8.8	9.0	9.1	9.2
1/64	9.6	9.7	9.8	9.9

We next present results for the refinement operator defined by (4.4). The problem and sequence of subspaces are as just described but only the subspaces $\mathcal{M}_k$, $k \geq j$ are used. In (4.4), we use a multigrid preconditioner (cf. [4]) scaled by 4 to define R_j, the operator on the finest uniform grid. The scaling was introduced to balance the size of the two terms in (4.4). Table 5.4 gives the condition number of the preconditioned system $\hat{B}A$ as a function of the mesh size of the uniform grid h_j and the number of refinement levels l.

Table 5.4

Condition numbers for $\hat{B}A$ using multigrid preconditioning on level j.

h_j	$l=1$	$l=2$	$l=3$	$l=4$
1/8	4.3	6.0	6.4	6.6
1/16	4.7	6.7	7.6	8.1
1/32	4.9	7.0	8.4	9.2
1/64	5.0	7.1	8.5	9.6

As a final example, we illustrate the preconditioning technique on a three dimensional problem. We consider a Galerkin approximation to the Laplace equation

$$\begin{aligned} -\Delta u &= f \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{5.1}$$

where $\Delta = \partial^2/\partial x^2 + \partial^2/\partial y^2 + \partial^2/\partial z^2$ and Ω is the unit cube. We define the coarse mesh by dividing Ω into eight smaller cubes of size $h_0 = 1/2$. Successively finer meshes are formed by dividing each cube of a coarser mesh into eight smaller cubes. The finite element space $\mathcal{M}_k$ is defined to be the set of continuous functions on Ω which are trilinear with respect to the k'th mesh and vanish on $\partial\Omega$.

Table 5.5 gives the condition number K of the preconditioned system $\mathcal{B}A$ where $\mathcal{B}$ is defined by (3.3). This example satisfies full elliptic regularity and the regularity and approximation assumption (A.3) holds with $\alpha = 1$. Thus, the theory predicts only a logarithmic growth in the condition number which is in agreement with the reported results. Note the finite element spaces are of rather large dimension, in fact, the $h_J = 1/64$ example has over a quarter of a million unknowns.

Table 5.5

Condition numbers for the three dimensional example.

h_j	$K(\mathcal{B}A)$
1/8	4.1
1/16	5.2
1/32	6.0
1/64	6.6

References

1 R.E. Bank and T.F. Dupont, *An optimal order process for solving elliptic finite element equations*, Math. Comp. **36** (1981), 35–51.

2 R.E. Bank, T.F. Dupont, and H. Yserentant, *The hierarchical basis multigrid method*, Num. Math. **52** (1988), 427–458.

3 G. Birkhoff and A. Schoenstadt, eds, "Elliptic Problem Solvers II," Academic Press, New York, 1984.

4 J.H. Bramble and J.E. Pasciak, *New convergence estimates for multigrid algorithms*, Math. Comp. **49** (1987), 311–329.

5 J.H. Bramble, J.E. Pasciak and A.H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comp. **47** (1986), 103–134.

6 J.H. Bramble, J.E. Pasciak and A.H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, II*, Math. Comp. **49** (1987), 1–16.

7 J.H. Bramble, J.E. Pasciak and A.H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, III*, Math. Comp. **51** (1988), 415–430.

8 J.H. Bramble, J.E. Pasciak and A.H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, IV*, Math. Comp. (to appear).

9 J.H. Bramble, J.E. Pasciak and J. Xu, *The analysis of multigrid algorithms with non-nested spaces or non-inherited quadratic forms*, Math. Comp., (submitted).

10 J.H. Bramble, J.E. Pasciak and J. Xu, *A multilevel preconditioner for domain decomposition boundary systems*, (in preparation).

11 J.H. Bramble, J.E. Pasciak and J. Xu, *Parallel multilevel preconditioners*, Math. Comp., (submitted).

12 P. Concus, G.H. Golub and G. Meurant, *Block preconditioning for the conjugate gradient method*, SIAM J. Sci. Stat. Comput. **6** (1985), 220–252.

13 Dupont, T., Kendall, R.P., and Rachford, H.H.,, *An approximate factorization procedure for solving self-adjoint elliptic difference equations*, SIAM J. Numer. Anal. **5** (1968), 559–573.

14 R. Glowinski, G.H. Golub, G.A. Meurant, J. Périaux, eds, "Proceedings, 1'st Inter. Conf. on Domain Decomposition Methods," SIAM, Philadelphia, 1988.

15 Hackbusch, W., "Multi-Grid Methods and Applications," Springer-Verlag, New York, 1985.

16 S. F. McCormick, *Multigrid Methods for Variational Problems: Further Results*, SIAM J. Numer. Anal. **21** (1984), 255 - 263.

17 S. F. McCormick, *Multigrid Methods for Variational Problems: General Theory for the V-Cycle*, SIAM J. Numer. Anal. **22** (1985), 634–643.

18 J. Mandel, S. McCormick and R. Bank, *Variational multigrid theory*, in "Multigrid Methods," Ed. S. McCormick, SIAM, Philadelphia, Penn., pp. 131–178.

19 J.A. Meyerink and H.A. van der Vorst, *Iterative methods for the solution of linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp. **31** (1977), 148–162.

20 J. Xu, "Theory of Multilevel Methods," Cornell Univ. (Thesis), 1988.

21 H. Yserentant, *On the multi-level splitting of finite element spaces*, Numerische Mathematik **49** (1986), 379–412.