SIAM REVIEW
Vol. 34, No. 4, pp. 581–613, December 1992

© 1992 Society for Industrial and Applied Mathematics
003

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

ITERATIVE METHODS BY SPACE DECOMPOSITION AND SUBSPACE CORRECTION*

JINCHAO XU†

Abstract. The main purpose of this paper is to give a systematic introduction to a number of iterative methods for symmetric positive definite problems. Based on results and ideas from various existing works on iterative methods, a unified theory for a diverse group of iterative algorithms, such as Jacobi and Gauss–Seidel iterations, diagonal preconditioning, domain decomposition methods, multigrid methods, multilevel nodal basis preconditioners and hierarchical basis methods, is presented. By using the notions of space decomposition and subspace correction, all these algorithms are classified into two groups, namely *parallel subspace correction* (PSC) and *successive subspace correction* (SSC) methods. These two types of algorithms are similar in nature to the familiar Jacobi and Gauss–Seidel methods, respectively.

A feature of this framework is that a quite general abstract convergence theory can be established. In order to apply the abstract theory to a particular problem, it is only necessary to specify a decomposition of the underlying space and the corresponding subspace solvers. For example, subspaces arising from the domain decomposition method are associated with subdomains whereas with the multigrid method subspaces are provided by multiple "coarser" grids. By estimating only two parameters, optimal convergence estimations for a given algorithm can be obtained as a direct consequence of the abstract theory.

Key words. domain decomposition, Gauss–Seidel, finite elements, hierarchical basis, Jacobi, multigrid, Schwarz, space decomposition, strengthened Cauchy–Schwarz inequalities, subspace correction

AMS(MOS) subject classifications. 65M60, 65N15, 65N30

1. Introduction.

In this paper, we shall discuss iterative algorithms to approximate the solution of a linear equation

$$Au = f, \tag{1.1}$$

where A is a symmetric positive definite (SPD) operator on a finite-dimensional vector space $\mathcal{V}$. There exist a large group of algorithms for solving the above problem. Classic examples are Gauss–Seidel, Jacobi iterations, and diagonal preconditioning techniques; more contemporary algorithms include multigrid and domain decomposition methods. The goal of this paper is to present these algorithms in a unified framework.

The central idea behind our framework is simple. A single step linear iterative method that uses an old approximation, u^{old}, of the solution u of (1.1), to produce a new approximation, u^{new}, usually consists of three steps:

(1) Form $r^{\text{old}} = f - Au^{\text{old}}$;

(2) Solve $Ae = r^{\text{old}}$ approximately: $\hat{e} = Br^{\text{old}}$ with $B \approx A^{-1}$;

(3) Update $u^{\text{new}} = u^{\text{old}} + \hat{e}$.

Clearly the choice of B (an approximate inverse of A) is the core of this type of algorithm. The point of our theory is to chose B by solving appropriate subspace problems. The subspaces are provided by a decomposition of $\mathcal{V}$:

$$\mathcal{V} = \sum_{i=1}^{J} \mathcal{V}_i.$$

Here $\mathcal{V}_i$ are subspaces of $\mathcal{V}$. Assume that $A_i : \mathcal{V}_i \mapsto \mathcal{V}_i$ is a restriction operator of A on $\mathcal{V}_i$ (see (3.2)); then the above three steps can be carried out on each subspace $\mathcal{V}_i$ with

*Received by the editors October 9, 1990; accepted for publication (in revised form) April 7, 1992. This work was supported by National Science Foundation grant DMS8805311-04.

†Department of Mathematics, The Pennsylvania State University, University Park, Pennsylvania 16802 (xu@math.psu.edu).

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$B = R_i \approx A_i^{-1}$ for $i = 1, 2, \ldots, J$. We shall demonstrate how an iterative algorithm can be obtained by repeatedly using the above three steps together with a decomposition of the space $\mathcal{V}$ with proper subspace solvers.

The general theory developed in this paper is based on numerous papers on multigrid and domain decomposition methods (of the most relevant are Bramble, Pasciak, Wang and Xu [13], [14], and Xu [48]). At the end of each section, we shall give brief comments on the literature that is *directly* related to this work. However, no attempt is made to give a complete survey on the vast literature in this active research field (the author must apologize in advance for the possible omission of references to works which may be closely relevant to this paper).

Our major concern in this paper is in the theoretical aspect of the algorithms and little attention is paid on their implementation. Algorithms for nonsymmetric or indefinite problems will not be discussed, but all the algorithms in this paper for symmetric positive definite problems can be applied to solving certain nonsymmetric and indefinite problems. In this direction, we refer to Cai and Widlund [20], Xu and Cai [54], Xu [48], [50], [51], and Bramble, Leyk, and Pasciak [9].

The remainder of the paper is organized as follows. In §2, we give a brief discussion of self-adjoint operators and the conjugate gradient method. In §3, we set up a general framework for linear iterative methods for symmetric positive definite problems. An abstract convergence theory for the algorithms in the framework of §3 is established in §4. As a preparation for applications of our general theory, §5 introduces a model finite element method. Sections 6 and 7 are devoted to multilevel and domain decomposition methods, respectively. Finally the Appendix contains a proof for a multigrid estimate.

2. Preliminaries. In this paper, two classes of iterative methods for the solution of (1.1) will be discussed: *the preconditioned conjugate gradient* method and the linear iterative method of the form

$$u^{k+1} = u^k + B(f - Au^k), \qquad k = 0, 1, 2, \ldots. \tag{2.1}$$

Some basic algebraic facts concerning the above methods will be discussed in this section. The first subsection is devoted to discussions of basic properties of symmetric positive definite operators (A and B will both be such operators) and preconditioned conjugate gradient methods. The second subsection is concerned with the relationship between the construction of a preconditioner and the operator that appears in (2.1).

We shall first introduce some notation. $\mathcal{V}$ is a linear vector space, $L(\mathcal{V})$ is the space of all linear operators from $\mathcal{V}$ to itself; for a given linear operator A on $\mathcal{V}$, $\sigma(A)$ is the spectrum of A, $\rho(A)$ its spectral radius, $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ its minimum and maximum eigenvalues, respectively; $(\cdot, \cdot)$ denotes a given inner product on $\mathcal{V}$; its induced norm is denoted by $\|\cdot\|$. The letter C or c, with or without subscript, denote generic constants that may not be the same at different occurrences. To avoid writing these constants repeatedly, we shall use the notation $\lesssim, \gtrsim$ and $\eqsim$. When we write

$$x_1 \lesssim y_1, \quad x_2 \gtrsim y_2, \quad \text{and} \quad x_3 \eqsim y_3,$$

then there exist constants C_1, c_2, c_3, and C_3 such that

$$x_1 \leq C_1 y_1, \quad x_2 \geq c_2 y_2 \quad \text{and} \quad c_3 x_3 \leq y_3 \leq C_3 x_3.$$

Consequently, $x_1 \lesssim 1$ means that $x_1 \leq C$ for some constant C; $x_1 \gtrsim 1$ and $x_1 \eqsim 1$ are understood similarly.

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

2.1. Self-adjoint operators and conjugate gradient methods. Let us first recall the definition of self-adjointness. Given $A \in L(\mathcal{V})$, the adjoint of A with respect to $(\cdot,\cdot)$, denoted by A^t, is the operator satisfying $(Au, v) = (u, A^t v)$ for all $u, v \in \mathcal{V}$. We say that A is self-adjoint with respect to $(\cdot,\cdot)$ if $A^t = A$ and A is symmetric positive definite if $(Av, v) > 0$ for $v \in \mathcal{V} \setminus \{0\}$, and A is nonnegative if $(Av, v) \geq 0$ for $v \in \mathcal{V}$.

An $n \times n$ matrix is self-adjoint with respect to the usual Euclidean inner product in $\mathbb{R}^n$ if and only if it is symmetric. Nevertheless a nonsymmetric matrix can still be self-adjoint with certain inner products. For example, if A and B are two symmetric positive definite matrices, although BA is not symmetric in general, it is self-adjoint with respect to $(\cdot,\cdot)_A$.

We shall have many occasions to use the following Cauchy–Schwarz inequality

$$(Ru, v) \leq (Ru, u)^{\frac{1}{2}} (Rv, v)^{\frac{1}{2}} \quad \forall u, v \in \mathcal{V},$$

where R is any nonnegative self-adjoint operator (with respect to $(\cdot,\cdot)$) on $\mathcal{V}$.

It is well known that if $A \in L(\mathcal{V})$ is SPD with respect to $(\cdot,\cdot)$, then all of its eigenvalues are positive and

$$\lambda_{\min}(A) = \min_{v \in \mathcal{V} \setminus \{0\}} \frac{(Av, v)}{\|v\|^2}, \qquad \lambda_{\max}(A) = \max_{v \in \mathcal{V} \setminus \{0\}} \frac{(Av, v)}{\|v\|^2}. \tag{2.2}$$

Furthermore, $(A\cdot,\cdot)$ defines another inner product on $\mathcal{V}$, denoted by $(\cdot,\cdot)_A$, and its induced norm is denoted by $\|\cdot\|_A$.

Throughout the paper, we shall assume the operator A in (1.1) is always SPD with certain inner product. For the equation (1.1), one of the most remarkable algorithms is the so-called conjugate gradient (CG) method (cf. [27]). Let u^k be the kth conjugate gradient iteration with the initial guess u^0. It is well known that

$$\|u - u_k\|_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|u - u_0\|_A,$$

where $\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$ is the condition number of A. This estimate shows that a smaller $\kappa(A)$ results in a faster convergence, hence the convergence may be improved by reducing the condition number. To this end, we introduce another SPD operator B on $\mathcal{V}$, self-adjoint with respect to the same inner product as A, and consider the equivalent equation

$$BAu = Bf. \tag{2.3}$$

As we pointed out earlier, BA is still SPD. Hence the CG method can be applied, and the resultant algorithm is known as *preconditioned conjugate gradient* method. The operator B here is known as a *preconditioner*. A good preconditioner should have the properties that the action of B is easy to compute and that $\kappa(BA)$ is smaller than $\kappa(A)$. For the product operator BA, there are two obvious ways to choose the inner products in the algorithm, namely $(A\cdot,\cdot)$ or $(B^{-1}\cdot,\cdot)$. In general, if the action of A is easier to compute than that of B^{-1}, we choose $(A\cdot,\cdot)$, otherwise $(B^{-1}\cdot,\cdot)$. Sometimes B takes the form $B = SS^t$ for some nonsingular operator S. Since $\kappa(BA) = \kappa(S^t AS)$, it is more convenient to apply the CG method to $(S^t AS)v = S^t f$ and to recover the solution u by $u = Sv$. In this case $S^t AS$ is self-adjoint with respect to the $(\cdot,\cdot)$.

The following result is often useful in the estimate of the condition number.

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

LEMMA 2.1. *Assume that A and B are both* SPD *with respect to $(\cdot,\cdot)$ and μ_0 and μ_1 are two positive constants. The following, which hold for all $v \in \mathcal{V}$, are equivalent:*

$$\mu_0(Av,v) \le (ABAv,v) \le \mu_1(Av,v),$$
$$\mu_0(Bv,v) \le (BABv,v) \le \mu_1(Bv,v),$$
$$\mu_1^{-1}(Av,v) \le (B^{-1}v,v) \le \mu_0^{-1}(Av,v),$$
$$\mu_1^{-1}(Bv,v) \le (A^{-1}v,v) \le \mu_0^{-1}(Bv,v).$$

If any of the above inequalities hold, then $\kappa(BA) \le \mu_1/\mu_0$.

The proof of this lemma is straightforward by using (2.2) and the observation that BA and $(AB)^{-1}$ are self-adjoint with respect to $(A\cdot,\cdot)$ and AB and $(BA)^{-1}$ are self-adjoint with respect to $(B\cdot,\cdot)$.

2.2. Linear iteration and preconditioning. We note that the core of the iterate scheme (2.1) is the operator B. Observe that if $B = A^{-1}$, one iteration gives the exact solution. In general, B may be regarded as an approximate inverse of A. We shall call B an *iterator* of the operator A. Note that a sufficient condition for the convergence of scheme (2.1) is

$$\rho = \|I - BA\|_A < 1. \tag{2.4}$$

In this case $\|u - u^k\|_A \le \rho^k \|u - u^0\|_A \to 0$ (as $k \to \infty$). We shall call the above defined ρ the convergence rate of (2.1). Note that the condition (2.4) is also necessary for convergence if B is self-adjoint with respect to $(\cdot,\cdot)$.

It is well known that the linear iterations and preconditioners are closed related. First of all, a symmetric iterative scheme gives rise to a preconditioner.

PROPOSITION 2.2. *Assume that B is symmetric with respect to the inner product $(\cdot,\cdot)$. If* (2.4) *holds, then B is* SPD *and*

$$\kappa(BA) \le \frac{1+\rho}{1-\rho}.$$

Notice that the convergence rate of the scheme (2.1) is ρ, but if we use B as a preconditioner for A, the PCG method converges at a faster rate since

$$\delta = \frac{\sqrt{\kappa(BA)}-1}{\sqrt{\kappa(BA)}+1} \le \frac{\sqrt{\frac{1+\rho}{1-\rho}}-1}{\sqrt{\frac{1+\rho}{1-\rho}}+1} = \frac{1-\sqrt{1-\rho^2}}{\rho} < \rho.$$

We conclude that for any symmetric linear iterative scheme (2.1), a preconditioner for A can be found and the convergence rate of (2.1) can be accelerated by using the PCG method.

Any preconditioner of A can also be used to construct a linear iterative scheme.

PROPOSITION 2.3. *Assume that B is a preconditioner of A. Then the following linear iteration*

$$u^{k+1} = u^k + \omega B(f - Au^k)$$

is convergent for $\omega \in (0,\ 2/\rho(BA))$, and the optimal convergence rate is attained when $\omega = 2(\lambda_{\min}(BA) + \lambda_{\max}(BA))^{-1}$, which results in an error reduction per iteration of $(\kappa(BA) - 1)/(\kappa(BA) + 1)$.

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Bibliographic comments. The CG method was proposed by Hestenes and Stiefel in [31]. The discussion of this method can be found in many textbooks such as Golub and Van Loan [27] and Hageman and Young [30]. For a history and related literature, see Golub and O'Leary [28].

3. Subspace correction methods based on space decompositions. A general framework for linear iterative methods will be presented in this section. We shall introduce notions of space decomposition and subspace correction. By decomposing the whole space into a sum of subspaces, an iterative algorithm can be obtained by correcting residues in these subspaces. Some well-known algorithms such as Jacobi and Gauss–Seidel iterations will be derived from the framework as special examples for illustration.

3.1. Space decomposition and subspace equations. A decomposition of $\mathcal{V}$ consists of a number of subspaces $\mathcal{V}_i \subset \mathcal{V}$ (for $1 \le i \le J$) such that

$$\mathcal{V} = \sum_{i=1}^{J} \mathcal{V}_i. \tag{3.1}$$

Thus, for each $v \in \mathcal{V}$, there exist $v_i \in \mathcal{V}_i$ $(1 \le i \le J)$ such that $v = \sum_{i=1}^{J} v_i$. This representation of v may not be unique in general.

For each i, we define $Q_i, P_i : \mathcal{V} \mapsto \mathcal{V}_i$ and $A_i : \mathcal{V}_i \mapsto \mathcal{V}_i$ by

$$(Q_i u, v_i) = (u, v_i), \quad (P_i u, v_i)_A = (u, v_i)_A, \quad u \in \mathcal{V}, v_i \in \mathcal{V}_i,$$

and

$$(A_i u_i, v_i) = (A u_i, v_i), \qquad u_i, v_i \in \mathcal{V}_i. \tag{3.2}$$

Q_i and P_i are both orthogonal projections and A_i is the restriction of A on $\mathcal{V}_i$ and is SPD. It follows from the definition that

$$A_i P_i = Q_i A. \tag{3.3}$$

This identity is of fundamental importance and will be used frequently in this paper. A consequence of it is that, if u is the solution of (1.1), then

$$A_i u_i = f_i \tag{3.4}$$

with $u_i = P_i u$ and $f_i = Q_i f$. This equation is the restriction of (1.1) to $\mathcal{V}_i$.

The subspace equation (3.4) will be in general solved approximately. To describe this, we introduce, for each i, another SPD operator $R_i : \mathcal{V}_i \mapsto \mathcal{V}_i$ that represents an approximate inverse of A_i in certain sense. Thus an approximate solution of (3.4) may be given by $\hat{u}_i = R_i f_i$.

Example 3.1. Consider the space $\mathcal{V} = R^n$ and the simplest decomposition:

$$\mathbb{R}^n = \sum_{i=1}^{n} \operatorname{span}\{e^i\}, \tag{3.5}$$

where e^i is the ith column of the identity matrix. For an SPD matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$

$$A_i = a_{ii}, \qquad Q_i y = y_i e^i,$$

where y_i the ith component of $y \in \mathbb{R}^n$.

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

3.2. PSC: Parallel subspace correction methods. This type of algorithm is similar to Jacobi method.

Basic idea. Let u^{old} be a given approximation of the solution u of (1.1). The accuracy of this approximation can be measured by the residual: $r^{\text{old}} = f - Au^{\text{old}}$. If $r^{\text{old}} = 0$ or very small, we are done. Otherwise, we consider the residual equation:

$$Ae = r^{\text{old}}.$$

Obviously $u = u^{\text{old}} + e$ is the solution of (1.1). Instead we solve the restricted equation to each subspace $\mathcal{V}_i$

$$A_i e_i = Q_i r^{\text{old}}.$$

As we are only seeking for a correction, we only need to solve this equation approximately using the subspace solver R_i described earlier

$$\hat{e}_i = R_i Q_i r^{\text{old}}.$$

An update of the approximation of u is obtained by

$$u^{\text{new}} = u^{\text{old}} + \sum_{i=1}^{J} \hat{e}_i$$

which can be written as

$$u^{\text{new}} = u^{\text{old}} + B(f - Au^{\text{old}}),$$

where

$$B = \sum_{i=1}^{J} R_i Q_i. \tag{3.6}$$

We therefore have the following algorithm.

ALGORITHM 3.1. *Given $u_0 \in \mathcal{V}$, apply the scheme* (2.1) *with B given by* (3.6).

Example 3.2. With $\mathcal{V} = \mathbb{R}^n$ and the decomposition given by (3.5), the corresponding Algorithm 3.1 is just the Jacobi iterative method.

It is well known that the Jacobi method is not convergent for all SPD problems; hence Algorithm 3.1 is not always convergent. However, the preconditioner obtained from this algorithm is of great importance.

LEMMA 3.1. *The operator B given by* (3.6) *is* SPD.

Proof. The symmetry of B follows from the symmetry of R_i. Now, for any $v \in \mathcal{V}$, we have

$$(Bv, v) = \sum_{i=1}^{J} (R_i Q_i v, Q_i v) \geq 0.$$

If $(Bv, v) = 0$, we then have $Q_i v = 0$ for all i. Let $v_i \in \mathcal{V}_i$ be such that $v = \sum_i v_i$, then

$$(v, v) = \sum_i (v, v_i) = \sum_i (Q_i v, v_i) = 0.$$

Therefore $v = 0$ and B is positive definite. □

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Preconditioners. By Proposition 2.2 and Lemma 3.1, B can be used as a preconditioner.

ALGORITHM 3.2. *Apply the* CG *method to equation* (2.3), *with* B *defined by* (3.6) *as a preconditioner.*

Example 3.3. The preconditioner B corresponding to Example 3.1 is

$$B = \operatorname{diag}(a_{11}^{-1}, a_{22}^{-1}, \ldots, a_{nn}^{-1}),$$

which is the well-known diagonal preconditioner for the SPD matrix A.

3.3. SSC: Successive subspace correction methods. This type of algorithm is similar to the Gauss–Seidel method.

Basic algorithm. To improve the PSC method that makes simultaneous correction, we here make the correction in one subspace at a time by using the most updated approximation of u. More precisely, starting from $v^0 = u^{\text{old}}$ and correcting its residual in $\mathcal{V}_1$ gives

$$v^1 = v^0 + R_1 Q_1 (f - A v^0).$$

By correcting the new approximation v^1 in the next space $\mathcal{V}_2$, we get

$$v^2 = v^1 + R_2 Q_2 (f - A v^1).$$

Proceeding this way successively for all $\mathcal{V}_i$ leads to the following algorithm.

ALGORITHM 3.3. *Let* $u^0 \in \mathcal{V}$ *be given, and assume that* $u^k \in \mathcal{V}$ *has been obtained. Then* u^{k+1} *is defined by*

$$u^{k+i/J} = u^{k+(i-1)/J} + R_i Q_i (f - A u^{k+(i-1)/J})$$

for $i = 1, \ldots, J$.

Example 3.4. For the decomposition (3.5), Algorithm 3.3 is the Gauss–Seidel iteration.

Example 3.5. More generally, decompose $\mathbb{R}^n$ as

$$\mathbb{R}^n = \sum_{i=0}^{J} \operatorname{span}\{e^{l_i}, e^{l_i+1}, \ldots, e^{l_{i+1}-1}\},$$

where $1 = l_0 < l_1 < \cdots < l_{J+1} = n + 1$. Then Algorithms 3.1, 3.2, and 3.3 are the block Jacobi method, block diagonal preconditioner, and block Gauss–Seidel method, respectively.

Error equations. Let $T_i = R_i Q_i A$. By (3.3), $T_i = R_i A_i P_i$. Note that $T_i : \mathcal{V} \mapsto \mathcal{V}_i$ is symmetric with respect to $(\cdot, \cdot)_A$ and nonnegative and that $T_i = P_i$ if $R_i = A_i^{-1}$.

If u is the exact solution of (1.1), then $f = Au$. By definition,

$$u - u^{k+i/J} = (I - T_i)(u - u^{k+(i-1)/J}), \qquad i = 1, \ldots, J.$$

A successive application of this identity yields

$$u - u^{k+1} = E_J (u - u^k), \tag{3.7}$$

where

$$E_J = (I - T_J)(I - T_{J-1}) \ldots (I - T_1). \tag{3.8}$$

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

There is also a symmetrized version of Algorithm 3.3.

ALGORITHM 3.4. *Let $u^0 \in \mathcal{V}$ be given and assume that $u^k \in \mathcal{V}$ has been obtained. Then u^{k+1} is defined by*

$$u^{k+i/(2J)} = u^{k+(i-1)/(2J)} + R_i Q_i (f - A u^{k+(i-1)/(2J)})$$

for $i = 1, 2, \ldots, J$, and

$$u^{k+(J+i)/(2J)} = u^{k+(J+i-1)/(2J)} + R_{J-i+1} Q_{J-i+1} (f - A u^{k+(J+i-1)/(2J)})$$

for $i = 1, 2, \ldots, J$.

The advantage of the symmetrized algorithm is that it can be used as a preconditioner. In fact, Algorithm 3.4 can be formulated in the form of (2.1) with operator B defined as follows: For $f \in \mathcal{V}$, let $Bf = u^1$ with u^1 obtained by Algorithm 3.4 applied to (1.1) with $u^0 = 0$.

For Algorithm 3.4, we have $u - u^{k+1} = E_J^s (u - u^k)$ with

$$E_J^s = (I - T_1)(I - T_2) \cdots (I - T_J)(I - T_J)(I - T_{J-1}) \cdots (I - T_1).$$

Remark 3.1. We observe that $E_J^s = E_J^* E_J$ (E_J^* is the adjoint of E_J with respect to $(\cdot, \cdot)_A$), and E_J^s is symmetric with respect to $(\cdot, \cdot)_A$. Therefore, $\|E_J^s\|_A = \|E_J\|_A^2$, and thus there is no qualitative difference in the convergence properties of Algorithms 3.3 and 3.4.

Let us introduce a relaxation method similar to Young's SOR method.

ALGORITHM 3.5. *Let $u^0 \in \mathcal{V}$ be given, and assume that $u^k \in \mathcal{V}$ has been obtained. Then u^{k+1} is defined by*

$$u^{k+i/J} = u^{k+(i-1)/J} + \omega R_i Q_i (f - A u^{k+(i-1)/J})$$

for $i = 1, \ldots, J$.

With $\mathcal{V} = \mathbb{R}^n$ and the decomposition given by (3.5), the above algorithm is the SOR method. As in the SOR method, a proper choice of ω can result in an improvement of the convergence rate, but it is not easy to find an optimal ω in general. The above algorithm is essentially the same as Algorithm 3.3 since we can absorb the relaxation parameter ω into the definition of R_i.

3.4. Multilevel methods. Multilevel algorithms are based on a nested sequence of subspaces:

$$\mathcal{M}_1 \subset \mathcal{M}_2 \subset \cdots \subset \mathcal{M}_J = \mathcal{V}. \tag{3.9}$$

Corresponding to these spaces, we define $\hat{Q}_k, \hat{P}_k : \mathcal{M}_J \mapsto \mathcal{M}_k$ as the orthogonal projections with respect to $(\cdot, \cdot)$ and $(\cdot, \cdot)_A$, respectively, and define $\hat{A}_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ by $(\hat{A}_k u_k, v_k) = (u_k, v_k)_A$ for $u_k, v_k \in \mathcal{M}_k$.

In a multilevel algorithm, an iterator of A is obtained by constructing iterators $\hat{B}_k$ of $\hat{A}_k$ for all k recursively. A basic algorithm is as follows.

ALGORITHM 3.6. *Let $\hat{B}_1 = \hat{A}_1^{-1}$ and assume that $\hat{B}_{k-1} : \mathcal{M}_{k-1} \mapsto \mathcal{M}_{k-1}$ has been defined; then for $g \in \mathcal{M}_k$, $\hat{B}_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ is defined as follows:*

Step 1. $v^1 = \hat{B}_{k-1} \hat{Q}_{k-1} g$;

Step 2. $\hat{B}_k g = v^1 + \hat{R}_k (g - \hat{A}_k v^1)$.

In the definition of this algorithm, we often call Step 1 correction (on the coarse space) and Step 2 the smoothing. The operator R_k is often called a smoother. This

operator should have the property that $\hat{R}_k(g - \hat{A}_k v^1)$ has small components on high frequencies of A_k, which are the eigenvectors with respect to the large eigenvalues. The main idea in this type of algorithm is to use the coarser spaces to correct lower frequencies, which are the eigenvectors with respect to smaller eigenvalues, and then to use R_k to damp out the high frequencies.

It is straightforward to show that

$$\hat{E}_k \equiv I - \hat{B}_k \hat{A}_k = (I - \hat{R}_k \hat{A}_k)(I - \hat{B}_{k-1} \hat{A}_{k-1} \hat{P}_{k-1}), \tag{3.10}$$

or

$$\hat{E}_k = (I - \hat{R}_k \hat{A}_k)(I - \hat{P}_{k-1} + \hat{E}_{k-1} \hat{P}_{k-1}).$$

Although this type of identity has been used effectively in previous work on the convergence properties of multigrid methods, cf. Maitre and Musy [34], Bramble and Pasciak [10] and Xu [49], it will not be used in this paper.

Relationship with SSC *methods.* Although Algorithm 3.6 and the SSC algorithm look quite different, they are closely related.

To understand the relationship between the multilevel and SSC algorithms, we look a recurrence relation for $I - \hat{B}_k \hat{A}_k \hat{P}_k$ (instead of $I - \hat{B}_k \hat{A}_k$). Multiplying both sides of (3.10) by $\hat{P}_k$ and using the fact that $\hat{P}_{k-1} \hat{P}_k = \hat{P}_{k-1}$, we get

$$I - \hat{B}_k \hat{A}_k \hat{P}_k = (I - \hat{R}_k \hat{A}_k \hat{P}_k)(I - \hat{B}_{k-1} \hat{A}_{k-1} \hat{P}_{k-1}).$$

Since $\hat{P}_J = I$, a successive application of the above identity yields

$$I - \hat{B}_J \hat{A}_J = (I - \hat{T}_J)(I - \hat{T}_{J-1}) \cdots (I - \hat{T}_1), \tag{3.11}$$

where

$$\hat{T}_1 = \hat{P}_1, \quad \hat{T}_i = \hat{R}_i \hat{A}_i \hat{P}_i, \quad i = 2, 3, \ldots, J.$$

The identity (3.11) establishes the following proposition.

PROPOSITION 3.2. *The multilevel Algorithm* 3.6 *is equivalent to the* SSC *Algorithm* 3.3 *with* $\mathcal{V}_i = \mathcal{M}_i$.

Now we look at these algorithms from an opposite viewpoint. Suppose we are given a space decomposition of $\mathcal{V}$ as in (3.1). Then the most naive way to construct a nested sequence of multilevel subspaces is

$$\mathcal{M}_k = \sum_{i=1}^{k} \mathcal{V}_i, \qquad k = 1, 2, \ldots, J, \tag{3.12}$$

which obviously satisfies (3.9).

PROPOSITION 3.3. *Algorithm* 3.3 *is equivalent to the multilevel Algorithm* 3.6 *with the multilevel subspaces defined by* (3.12) *and the smoothing operator given by*

$$\hat{R}_k = R_k Q_k.$$

Proof. As $\mathcal{V}_k \subset \mathcal{M}_k$, we have $Q_k \hat{Q}_k = Q_k$. Hence

$$\hat{T}_k = \hat{R}_k \hat{Q}_k \hat{A} = R_k Q_k \hat{Q}_k A = R_k Q_k A = T_k.$$

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

This shows that $E_J = I - \hat{B}_J \hat{A}_J$ and the two algorithms are equivalent. □

Let us now give a couple of variants of Algorithm 3.3.

ALGORITHM 3.7. *Let* $B_1 = \hat{A}_1^{-1}$ *and assume that* $B_{k-1} : \mathcal{M}_{k-1} \mapsto \mathcal{M}_{k-1}$ *has been defined; then for* $g \in \mathcal{M}_k$, $B_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ *is defined by*

Step 1. $v^1 = \hat{R}_k g$;

Step 2. $B_k g = v^1 + B_{k-1}\hat{Q}_{k-1}(g - \hat{A}_k v^1)$.

Comparing this algorithm with Algorithm 3.6, we see that the difference lies in the order of smoothing and correction. If we write down the corresponding residual operator as for Algorithm 3.6, we immediately obtain the following.

PROPOSITION 3.4. *The residual operator on* $\mathcal{M}_k$ *of Algorithm* 3.7 *is the adjoint of that of Algorithm* 3.6.

We now combine Algorithms 3.6 with 3.7 to obtain the V-cycle algorithm.

ALGORITHM 3.8. *Let* $\hat{B}_1 = \hat{A}_1^{-1}$ *and assume that* $\hat{B}_{k-1} : \mathcal{M}_{k-1} \mapsto \mathcal{M}_{k-1}$ *has been defined; then for* $g \in \mathcal{M}_k$, $\hat{B}_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ *is defined by*

Step 1. $v^1 = \hat{R}_k g$;

Step 2. $v^2 = v^1 + \hat{B}_{k-1}\hat{Q}_{k-1}(g - \hat{A}_k v^1)$;

Step 3. $\hat{B}_k g = \hat{R}_k \hat{Q}_k v^2$.

The relationship between Algorithms 3.6, 3.7, and 3.8 is described in Proposition 3.5.

PROPOSITION 3.5. *The residual operator on* $\mathcal{M}_k$ *of Algorithm* 3.8 *is self-adjoint and it is the product of the residual operator of Algorithm* 3.6 *with that of Algorithm* 3.7. *Algorithm* 3.8 *is equivalent to* SSC *Algorithm* 3.4 *with the multilevel spaces defined by* (3.12) *and* $\hat{R}_k = R_k Q_k$.

Remark 3.2. Because of Propositions 3.4 and 3.5, the convergence estimates of Algorithm 3.7 and 3.8 are direct consequences of those of Algorithm 3.6. Therefore our later convergence analysis will be carried out only for Algorithm 3.6, which also is equivalent to Algorithm 3.3.

Bibliographic comments. A classic way of formulating an iterative method is by matrix splitting; we refer to Varga [46]. The multilevel algorithm (for finite difference equations) was developed in the sixties by Fedorenko [26] and Bakhvalov [3]. Extensive research on this method has been done since the seventies (cf. Brandt [19]). Among the vast multigrid literature, we refer to the book by Hackbusch [29] and the book edited by McCormick [36] (which contains a list of over six hundred papers on multigrid). For a rather complete theoretical analysis of multilevel algorithms, we refer to the author's thesis [49]. The formulation of the multigrid algorithm in terms of operators $\hat{B}_k$ was first introduced by Bramble and Pasciak [10].

The connection of the multigrid algorithm with the SSC type algorithm has been discussed by McCormick and Ruge [39]. Algorithm 3.3 was formulated by Bramble, Pasciak, Wang, and Xu [13] and Proposition 3.8 can also be found there. The result in Proposition 3.3 seems new and it particularly reveals that the so-called FAC method (cf. [38]) is equivalent to the classic multigrid algorithm with smoothing done only in the refined region (cf. [53]).

4. Convergence theory. The purpose of this section is to establish an abstract theory for algorithms described in previous sections. For reasons mentioned in Remarks 3.1 and 3.2, it suffices to study Algorithms 3.2 and 3.3. Two fundamental theorems will be presented.

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

For Algorithm 3.2, we need to estimate the condition number of

$$T = BA = \sum_{i=1}^{J} T_i,$$

where B is defined by (3.6) and $T_i = R_i A_i P_i$.

For Algorithm 3.3, we need to establish the contraction property: there exists a constant $0 < \delta < 1$ such that

$$\|E_J\|_A \leq \delta \quad \text{with} \quad \|E_J\|_A = \sup_{v \in \mathcal{V}} \frac{\|E_J v\|_A}{\|v\|_A},$$

where E_J is given by (3.8), or equivalently,

$$\|v\|_A^2 \leq \frac{1}{1-\delta^2}(\|v\|_A^2 - \|E_J v\|_A^2) \quad \forall v \in \mathcal{V}. \tag{4.1}$$

Applying this estimate to (3.7) yields $\|u - u^k\|_A \leq \delta^k \|u - u^0\|_A$.

The estimates of $\kappa(BA)$ and $\|E_J\|_A$ are mainly in terms of two parameters, K_0 and K_1, defined as follows.

1. For any $v \in \mathcal{V}$, there exists a decomposition $v = \sum_{i=1}^{J} v_i$ for $v_i \in \mathcal{V}_i$ such that

$$\sum_{i=1}^{J} (R_i^{-1} v_i, v_i) \leq K_0 (Av, v). \tag{4.2}$$

2. For any $S \subset \{1, 2, \ldots J\} \times \{1, 2, \ldots J\}$ and $u_i, v_i \in \mathcal{V}$ for $i = 1, 2, \ldots, J$,

$$\sum_{(i,j) \in S} (T_i u_i, T_j v_j)_A \leq K_1 \left(\sum_{i=1}^{J} (T_i u_i, u_i)_A \right)^{\frac{1}{2}} \left(\sum_{j=1}^{J} (T_j v_j, v_j)_A \right)^{\frac{1}{2}}. \tag{4.3}$$

4.1. Fundamental theorems. Our first fundamental theorem is an estimate of the condition number of BA.

THEOREM 4.1 (Fundamental Theorem I). *Assume that B is the* SSC *preconditioner given by* (3.6); *then*

$$\kappa(BA) \leq K_0 K_1.$$

Proof. We prove that

$$\lambda_{\max}(BA) \leq K_1, \tag{4.4}$$

and

$$\lambda_{\min}(BA) \geq K_0^{-1}. \tag{4.5}$$

It follows directly from the definition of K_1 that

$$\|Tv\|_A^2 = \sum_{i,j=1}^{J} (T_i v, T_j v)_A \leq K_1 (Tv, v)_A \leq K_1 \|Tv\|_A \|v\|_A,$$

which implies (4.4).

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

If $v = \sum_{i=1}^{J} v_i$ is a decomposition that satisfies (4.2), then

$$(v,v)_A = \sum_{i=1}^{J}(v_i,v)_A = \sum_{i=1}^{J}(v_i,P_iv)_A,$$

and by the Cauchy–Schwarz inequality,

$$\sum_{i=1}^{J}(v_i,P_iv)_A = \sum_{i=1}^{J}(v_i,A_iP_iv) \le \sum_{i=1}^{J}(R_i^{-1}v_i,v_i)^{\frac{1}{2}}(R_iA_iP_iv,v)_A^{\frac{1}{2}}$$

$$\le \left(\sum_{i=1}^{J}(R_i^{-1}v_i,v_i)\right)^{\frac{1}{2}}\left(\sum_{i=1}^{J}(T_iv,v)_A\right)^{\frac{1}{2}} \le \sqrt{K_0}\|v\|_A(Tv,v)_A^{\frac{1}{2}}.$$

Consequently,

$$\|v\|_A^2 \le K_0(Tv,v)_A,$$

which implies (4.5). □

As a consequence of (4.4), we have

COROLLARY 4.2. *A sufficient condition for Algorithm* 3.1 *to be convergent is that*

$$K_1 < 2.$$

Remark 4.3. If K_0 is the smallest constant satisfying the inequality (4.2), then

$$\lambda_{\min}(BA) = K_0^{-1}.$$

In fact, for the trivial decomposition $v = \sum_{i=1}^{J} v_i$ with $v_i = T_iT^{-1}v$,

$$K_0 \le \max_{v\in\mathcal{V}} \frac{\sum_{i=1}^{J}(R_i^{-1}T_iT^{-1}v, T_iT^{-1}v)}{\|v\|_A^2} = \max_{v\in\mathcal{V}} \frac{(T^{-1}v,v)_A}{(v,v)_A} = (\lambda_{\min}(BA))^{-1}.$$

This together with (4.5) justifies our claim.

To present our next theorem, let us first prove a very simple but important lemma.

LEMMA 4.3. *Denote, for* $1 \le i \le J$, $E_i = (I-T_i)(I-T_{i-1})\dots(I-T_1)$ *and* $E_0 = I$. *Then*

$$I - E_i = \sum_{j=1}^{i} T_jE_{j-1}, \tag{4.6}$$

$$(2-\omega_1)\sum_{i=1}^{J}(T_iE_{i-1}v, E_{i-1}v)_A \le \|v\|_A^2 - \|E_Jv\|_A^2 \quad \forall v \in \mathcal{V}. \tag{4.7}$$

Proof. Equation (4.6) follows immediately from the trivial identity $E_{i-1} - E_i = T_iE_{i-1}$. From this identity, we further deduce that

$$\begin{aligned}\|E_{i-1}v\|_A^2 - \|E_iv\|_A^2 &= \|T_iE_{i-1}v\|_A^2 + 2(T_iE_{i-1}v, E_iv)_A \\ &= (T_iE_{i-1}v, T_iE_{i-1}v)_A + 2(T_i(I-T_i)E_{i-1}v, E_{i-1}v)_A \\ &= ((2I-T_i)T_iE_{i-1}v, E_{i-1}v)_A \ge (2-\omega_1)(T_iE_{i-1}v, E_{i-1}v)_A.\end{aligned}$$

Summing up these inequalities with respect to i gives (4.7). □

Now we are in a position to present our second fundamental theorem.

THEOREM 4.4 (Fundamental Theorem II). *For Algorithm* 3.3,

$$\|E_J\|_A^2 \le 1 - \frac{2-\omega_1}{K_0(1+K_1)^2}, \tag{4.8}$$

where $\omega_1 = \max_i^\rho(R_iA_i)$.

Proof. In view of (4.1), (4.5), and (4.7), it suffices to show that

$$\sum_{i=1}^{J}(T_iv,v)_A \le (1+K_1)^2\sum_{i=1}^{J}(T_iE_{i-1}v,E_{i-1}v)_A \quad \forall v\in\mathcal{V}. \tag{4.9}$$

By (4.6)

$$\begin{aligned}(T_iv,v)_A &= (T_iv,E_{i-1}v)_A + (T_iv,(I-E_{i-1})v)_A\\ &= (T_iv,E_{i-1}v)_A + \sum_{j=1}^{i-1}(T_iv,T_jE_{j-1}v)_A.\end{aligned}$$

Applying the Cauchy–Schwarz inequality gives

$$\sum_{i=1}^{J}(T_iv,E_{i-1}v)_A \le \left(\sum_{i=1}^{J}(T_iv,v)_A\right)^{\frac12}\left(\sum_{i=1}^{J}(T_iE_{i-1}v,E_{i-1}v)_A\right)^{\frac12}.$$

By the definition of K_1 in (4.3), we have

$$\sum_{i=1}^{J}\sum_{j=1}^{i-1}(T_iv,T_jE_{j-1}v)_A \le K_1\left(\sum_{i=1}^{J}(T_iv,v)_A\right)^{\frac12}\left(\sum_{j=1}^{J}(T_jE_{j-1}v,E_{j-1}v)_A\right)^{\frac12}.$$

Combining these three formulae then leads to (4.9). □

This theorem shows that the SSC algorithm converges as long as $\omega_1 < 2$. The condition that $\omega_1 < 2$ is reminiscent of the restriction on the relaxation parameter in the SOR method. Since SOR (or block SOR) is a special SSC method, Theorem 4.4 gives another proof of its convergence for any symmetric positive definite system (and more details for this application will be given later).

4.2. On the estimate of K_0 and K_1. Our fundamental theorems depend only on three parameters: ω_1, K_0 and K_1. Obviously there is little we can say about ω_1. We shall now discuss techniques for estimating K_0 and K_1.

We first state a simple result for the estimate of K_0.

LEMMA 4.5. *Assume that, for any* $v\in\mathcal{V}$, *there is a decomposition* $v=\sum_{i=1}^{J}v_i$ *with* $v_i\in\mathcal{V}_i$ *satisfying*

$$\sum_{i=1}^{J}(v_i,v_i)_A \le C_0(v,v)_A, \quad \text{or} \quad \sum_{i=1}^{J}\lambda_i(v_i,v_i) \le \hat{C}_0(v,v)_A,$$

where $\lambda_i=\rho(A_i)$; *then* $K_0\le C_0/\omega_0$ *or* $K_0\le\hat{C}_0/\hat{\omega}_0$ *where*

$$\omega_0 = \min_{1\le i\le J}\lambda_{\min}(R_iA_i) \quad \text{and} \quad \hat{\omega}_0 = \min_{1\le i\le J}(\lambda_i\lambda_{\min}(R_i)).$$

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

The proof of the above lemma is straightforward.

We now turn to the estimate of K_1. For this purpose, we introduce a nonnegative symmetric matrix $\mathcal{E} = (\epsilon_{ij}) \in \mathbb{R}^{J\times J}$ where ϵ_{ij} is the smallest constant satisfying

$$(T_i u, T_j v)_A \le \omega_1 \epsilon_{ij} (T_i u, u)_A^{\frac{1}{2}} (T_j v, v)_A^{\frac{1}{2}} \quad \forall u, v \in \mathcal{V}. \tag{4.10}$$

Clearly $\epsilon_{ij} \le 1$ and, $\epsilon_{ij} = 0$ if $P_i P_j = 0$. If $\epsilon_{ij} < 1$, the above inequality is often known as the *strengthened Cauchy–Schwarz inequality*.

LEMMA 4.6.

$$K_1 \le \omega_1 \rho(\mathcal{E}).$$

If $\epsilon_{ij} \lesssim \gamma^{|i-j|}$ *for some* $\gamma \in (0,1)$, *then* $\rho(\mathcal{E}) \lesssim (1-\gamma)^{-1}$ *and* $K_1 \lesssim \omega_1 (1-\gamma)^{-1}$. *In general,* $\rho(\mathcal{E}) \le J$ *and* $K_1 \le \omega_1 J$.

Proof. Since $\mathcal{E}$ is symmetric, the estimate $K_1 \le \omega_1 \rho(\mathcal{E})$ follows by definition. The other estimates follow from the inequality $\rho(\mathcal{E}) \le \max_{1\le j\le J} \sum_{i=1}^{J} \epsilon_{ij}$. □

We shall now estimate K_1 in terms of $\mathcal{E}$ in a more precise fashion. To this end, we define, for a given subset $\mathcal{J}_0 \subset \{1, \ldots, J\}$,

$$\gamma_0 = |\mathcal{J}_0|, \qquad \sigma_0 = \max_{j\notin\mathcal{J}_0} \sum_{i\notin\mathcal{J}_0} \epsilon_{ij}.$$

Here $|\mathcal{J}_0|$ denotes the number of elements in $\mathcal{J}_0$.

LEMMA 4.7.

$$K_1 \le \omega_1 (\gamma_0 + \sigma_0).$$

Proof. Let

$$S_{11} = S \cap (\mathcal{J}_0 \times \mathcal{J}_0), \qquad S_{12} = S \cap (\mathcal{J}_0 \times \mathcal{J}_0^c),$$

$$S_{21} = S \cap (\mathcal{J}_0^c \times \mathcal{J}_0), \qquad S_{22} = S \cap (\mathcal{J}_0^c \times \mathcal{J}_0^c).$$

Using (4.10), it is elementary to show that

$$\left(\sum_{(i,j)\in S_{11}} (T_i u_i, T_j v_j)_A \right)^2 \le \omega_1^2 \gamma_0^2 \sum_{i\in\mathcal{J}_0} (T_i u_i, u_i)_A \sum_{j\in\mathcal{J}_0} (T_j v_j, v_j)_A, \tag{4.11}$$

$$\left(\sum_{(i,j)\in S_{22}} (T_i u_i, T_j v_j)_A \right)^2 \le \omega_1^2 \sigma_0^2 \sum_{i\in\mathcal{J}_0^c} (T_i u_i, u_i)_A \sum_{j\in\mathcal{J}_0^c} (T_j v_j, v_j)_A. \tag{4.12}$$

Now, for any $i \in \mathcal{J}_0$, let $\mathcal{J}_0^c(i) = \{j \in \mathcal{J}_0^c : (i,j) \in S_{12}\}$. Then

$$\left(\sum_{(i,j)\in S_{12}} (T_i u_i, T_j v_j)_A \right)^2 = \left(\sum_{i\in\mathcal{J}_0} \left(T_i u_i, \sum_{j\in\mathcal{J}_0^c(i)} T_j v_j \right)_A \right)^2$$

$$\le \gamma_0 \sum_{i\in\mathcal{J}_0} \|T_i u_i\|_A^2 \left\| \sum_{j\in\mathcal{J}_0(i)} T_j v_j \right\|_A^2 \le \omega_1 \gamma_0 \sum_{i\in\mathcal{J}_0} (T_i u_i, u_i)_A \left\| \sum_{j\in\mathcal{J}_0(i)} T_j v_j \right\|_A^2.$$

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Similar to (4.12), for each i, we have

$$\left\| \sum_{j \in \mathcal{J}_0^c(i)} T_j v_j \right\|_A^2 \leq \omega_1 \sigma_0 \sum_{j \in \mathcal{J}_0^c} (T_j v_j, v_j)_A.$$

Consequently, we have shown that

$$\left(\sum_{(i,j) \in S_{12}} (T_i u_i, T_j v_j)_A \right)^2 \leq \omega_1^2 \gamma_0 \sigma_0 \sum_{i \in \mathcal{J}_0} (T_i u_i, u_i)_A \sum_{j \in \mathcal{J}_0^c} (T_j v_j, v_j)_A.$$

Similarly

$$\left(\sum_{(i,j) \in S_{21}} (T_i u_i, T_j v_j)_A \right)^2 \leq \omega_1^2 \gamma_0 \sigma_0 \sum_{i \in \mathcal{J}_0^c} (T_i u_i, u_i)_A \sum_{j \in \mathcal{J}_0} (T_j v_j, v_j)_A.$$

As $S = S_{11} \cup S_{12} \cup S_{21} \cup S_{22}$, the desired estimate follows by some elementary manipulations. □

4.3. A quantitative estimate for Gauss–Seidel iteration. As a demonstration of the abstract theory developed in this section, we shall now apply Theorem 6.10 to get a quantitative estimate of the convergence factor for Gauss–Seidel method for general symmetric positive definite algebraic systems.

Let $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ be a given SPD matrix. By Example 3.3, the Gauss–Seidel iteration for the system $Ax = b$ is just Algorithm 3.3 with respect to the decomposition (3.5). It is straightforward to see that the corresponding constant K_0 is given by

$$K_0 = \rho(DA^{-1}) = 1/\lambda_{\min}(D^{-1}A),$$

where D is the diagonal matrix of A and the corresponding matrix $\mathcal{E}$ is given by

$$\epsilon_{ij} = \frac{|a_{ij}|}{\sqrt{a_{ii} a_{jj}}}.$$

If we denote $|A| = (|a_{ij}|)$, we then have

$$K_1 \leq \rho(\mathcal{E}) = \lambda_{\max}(D^{-1}|A|).$$

An application of the Fundamental Theorem II then gives an estimate of the convergence factor, ρ, of Gauss–Seidel method as follows:

$$\rho^2 \leq 1 - \frac{\lambda_{\min}(D^{-1}A)}{(1 + \lambda_{\max}(D^{-1}|A|))^2}. \tag{4.13}$$

A more careful analysis by following the proof of Theorem 4.3 yields a slightly sharper estimate:

$$\rho^2 \leq 1 - \frac{\lambda_{\min}(D^{-1}A)}{(1 + \sqrt{\rho(D^{-1}LD^{-1}U)})^2} = 1 - \frac{\lambda_{\min}(D^{-1}A)}{(1 + \|D^{-1/2}LD^{-1/2}\|_{l_2})^2},$$

where L and U are the lower and upper triangular parts of A, respectively.

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

4.4. Algorithms for linear algebraic systems. All algorithms in the preceding section have been presented in terms of projections and operators in abstract vector spaces. In this section, we shall translate all these algorithms into explicit algebraic forms.

Assume that $\mathcal{V}$ and $\mathcal{W}$ are two vector spaces and $A \in L(\mathcal{V}, \mathcal{W})$. By convention, the matrix representation of A with respect to a basis $(\phi_1, \ldots, \phi_n)$ of $\mathcal{V}$ and a basis $(\psi_1, \ldots, \psi_m)$ of $\mathcal{W}$ is the matrix $\tilde{A} \in \mathbb{R}^{m \times n}$ satisfying

$$(A\phi_1, \ldots, A\phi_n) = (\psi_1, \ldots, \psi_m)\tilde{A}.$$

Given any $v \in \mathcal{V}$, there exists a unique $\nu = (\nu_i) \in \mathbb{R}^n$ such that

$$v = \sum_{i=1}^{n} \nu_i \phi_i.$$

The vector ν can be regarded as the matrix representation of v, denoted by $\nu = \tilde{v}$.

By definition, we have, for any two operators A, B and a vector v

$$\widetilde{AB} = \tilde{A}\tilde{B} \quad \text{and} \quad \widetilde{Av} = \tilde{A}\tilde{v}. \tag{4.14}$$

Under the basis (ϕ_k), we define the following two matrices

$$\mathcal{M} = ((\phi_i, \phi_j))_{n \times n} \quad \text{and} \quad \mathcal{A} = ((A\phi_i, \phi_j))_{n \times n}.$$

We shall call $\mathcal{M}$ and $\mathcal{A}$ to be the mass matrix and stiffness matrix, respectively. It can be easily shown that

$$\mathcal{A} = \mathcal{M}\tilde{A}.$$

and that $\mathcal{M}$ is the matrix representation of the operator defined by

$$Rv = \sum_{i=1}^{n} (v, \phi_i)\phi_i \quad \forall v \in \mathcal{V}. \tag{4.15}$$

Under a given basis (ϕ_k), equation (1.1) can be transformed to an algebraic system

$$\mathcal{A}\mu = \eta. \tag{4.16}$$

Similar to (2.1), a linear iterative method for (4.16) can be written as

$$\mu^{k+1} = \mu^k + \mathcal{B}(\eta - \mathcal{A}\mu^k), \qquad k = 0, 1, 2, \ldots, \tag{4.17}$$

where $\mathcal{B} \in \mathbb{R}^{n \times n}$ is an iterator of the matrix $\mathcal{A}$.

PROPOSITION 4.8. *Assume that* $\tilde{u} = \mu, \tilde{f} = \beta$, *and* $\eta = \mathcal{M}\beta$. *Then* u *is the solution of* (1.1) *if and only if* μ *is the solution of* (4.16). *The linear iterations* (2.1) *and* (4.17) *are equivalent if and only if* $\tilde{B} = \mathcal{B}\mathcal{M}$. *In this case* $\kappa(\mathcal{B}\mathcal{A}) = \kappa(BA)$.

In the following, we shall call $\mathcal{B}$ the algebraic representation of B.

Using the property of the operator defined by (4.15), we can show Proposition 4.9.

PROPOSITION 4.9. *The scheme* (2.1) *represents the Richardson iteration for* (4.16) *if* B *is given by*

$$Bv = \omega\rho(\mathcal{A})^{-1} \sum_{i=1}^{n} (v, \phi_i)\phi_i \quad \forall v \in \mathcal{V},$$

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

and it represents the damped Jacobi iteration if B is given by

$$Bv = \omega \sum_{i=1}^{n} a_{ii}^{-1}(v, \phi_i)\phi_i \quad \forall v \in \mathcal{V}.$$

The Gauss–Seidel method is a little more complicated. There is no compact form for the expression of B.

PROPOSITION 4.10. *The iteration* (2.1) *represents the symmetric Gauss–Seidel iteration for* (4.16) *if B is defined as follows: for any $v \in \mathcal{V}$, $Bv = w^{2n}$, where*

$$w^i = w^{i-1} + (A\phi_i, \phi_i)^{-1}(v - Aw^{i-1}, \phi_i)\phi_i$$

for $i = 1, 2, \ldots, n$ with $w^0 = 0$, and

$$w^{n+j} = w^{n+j-1} + (A\phi_{n-j+1}, \phi_{n-j+1})^{-1}(v - Aw^{n+j-1}, \phi_{n-j+1})\phi_{n-j+1}$$

for $j = 1, 2, \ldots, n$. Furthermore, $\widetilde{B} = (\mathcal{D} - \mathcal{U})^{-1}\mathcal{D}(\mathcal{D} - \mathcal{L})^{-1}\mathcal{M}$.

We are now in a position to derive the algebraic representation of the PSC preconditioner and SSC iterative algorithm. For each k, we assume that $(\phi_1^k, \ldots, \phi_{n_k}^k)$ is a basis of $\mathcal{V}_k$. Since $\mathcal{V}_k \subset \mathcal{V}$, there exists a unique matrix $\mathcal{I}_k \in \mathbb{R}^{n \times n_k}$ such that

$$(\phi_1^k, \ldots, \phi_{n_k}^k) = (\phi_1, \ldots, \phi_n)\mathcal{I}_k.$$

Assume that $I_k : \mathcal{V}_k \mapsto \mathcal{V}$ is the inclusion operator; then $\tilde{I}_k = \mathcal{I}_k$. The matrix $\mathcal{I}_k$ will play a key role in the algebraic formulations of our algorithms.

It is easy to see that the matrix representation of the projection Q_k is given by

$$\tilde{Q}_k = \mathcal{M}_k^{-1}\mathcal{I}_k^t\mathcal{M}, \tag{4.18}$$

where $\mathcal{M}_k$ is the mass matrix under the basis (ϕ_i^k).

To derive the algebraic representation of the preconditioner (3.6), we rewrite it in a slightly different form:

$$B = \sum_{k=1}^{J} I_k R_k Q_k.$$

Applying (4.14) and (4.18) gives

$$\widetilde{B} = \sum_{k=1}^{J} \tilde{I}_k \tilde{R}_k \tilde{Q}_k = \sum_{k=1}^{J} \mathcal{I}_k(\mathcal{R}_k\mathcal{M}_k)(\mathcal{M}_k^{-1}\mathcal{I}_k^t\mathcal{M}) = \mathcal{B}\mathcal{M}.$$

Here $\mathcal{R}_k$ is the algebraic representation of R_k and

$$\mathcal{B} = \sum_{k=1}^{J} \mathcal{I}_k\mathcal{R}_k\mathcal{I}_k^t. \tag{4.19}$$

Different choices of R_k yield the following three main different preconditioners:

$$\mathcal{B} = \begin{cases} \sum_{k=1}^{J} \rho(\mathcal{A}_k)^{-1}\mathcal{I}_k\mathcal{I}_k^t & \text{Richardson;} \\ \sum_{k=1}^{J} \mathcal{I}_k\mathcal{D}_k^{-1}\mathcal{I}_k^t & \text{Jacobi;} \\ \sum_{k=1}^{J} \mathcal{I}_k\mathcal{G}_k\mathcal{I}_k^t & \text{Gauss–Seidel.} \end{cases}$$

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Here $\mathcal{G}_k = (\mathcal{D}_k - \mathcal{U}_k)^{-1}\mathcal{D}_k(\mathcal{D}_k - \mathcal{L}_k)^{-1}$, $\mathcal{A}_k = \mathcal{D}_k - \mathcal{L}_k - \mathcal{U}_k$, $\mathcal{D}_k$ is the diagonal of $\mathcal{A}_k$, $-\mathcal{L}_k$ and $-\mathcal{U}_k$ are, respectively, the lower and upper triangular parts of $\mathcal{A}_k$.

Following Proposition 4.8, we get Proposition 4.11.

PROPOSITION 4.11. *The* PSC *preconditioner for the stiffness matrix $\mathcal{A}$ is given by* (4.19) *and* $\kappa(\mathcal{B}\mathcal{A}) = \kappa(BA)$.

Similarly, we can derive the algebraic representation of Algorithm 3.3 for solving (4.16).

ALGORITHM 4.1. *$\mu^0 \in \mathbb{R}^n$ is given. Assume that $\mu^k \in \mathbb{R}^n$ is obtained. Then μ^{k+1} is defined by*

$$\mu^{k+i/J} = \mu^{k+(i-1)/J} + \mathcal{I}_i\mathcal{R}_i\mathcal{I}_i^t(\eta - \mathcal{A}\mu^{k+(i-1)/J})$$

for $i = 1, \ldots, J$.

Bibliographic comments. The estimate (4.5) originates from a result in Lions [33] who proved that $\lambda_{\min}(BA) \geq K_0^{-1}$ in the special case that $R_i^{-1} = A_i$ and $J = 2$. An extension of the result to general J is contained in Dryja and Widlund [22].

The theory presented in §4.1 stems from [13], [14], but is given in an improved form. The introduction of the parameter K_1 greatly simplifies the theory.

Lemma 4.7 is inspired by a similar result in [14].

5. Finite element equations. In the following sections, we shall give some examples of iterative methods for the solution of discretized partial differential equations to demonstrate our unified theory developed in previous sections. This section is devoted to some basic properties of finite element spaces and finite element equations.

We consider the boundary-value problem:

$$\begin{aligned} -\nabla \cdot a\nabla U &= F \quad \text{in } \Omega, \\ U &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{5.1}$$

where $\Omega \subset \mathbb{R}^d$ is a polyhedral domain and a is a smooth function on $\bar{\Omega}$ with a positive lower bound.

Let $H^1(\Omega)$ be the standard Sobolev space consisting of square integrable functions with square integrable (weak) derivatives of first order, and $H_0^1(\Omega)$ the subspace of $H^1(\Omega)$ consisting of functions that vanish on $\partial\Omega$. Then $U \in H_0^1(\Omega)$ is the solution of (5.1) if and only if

$$a(U, \chi) = (F, \chi) \quad \forall \chi \in H_0^1(\Omega), \tag{5.2}$$

where

$$a(U, \chi) = \int_\Omega a\nabla U \cdot \nabla \chi dx, \qquad (F, \chi) = \int_\Omega F\chi dx.$$

Assume that Ω is triangulated with $\Omega = \cup_i \tau_i$, where τ_i's are nonoverlapping simplexes of size h, with $h \in (0, 1]$ and quasi uniform; i.e., there exist constants C_0 and C_1 not depending on h such that each simplex τ_i is contained in (contains) a ball of radius $C_1 h$ (respectively, $C_0 h$). Define

$$\mathcal{V} = \{v \in H_0^1(\Omega) : v|_\tau \in \mathcal{P}_1(\tau_i)\ \forall \tau_i\},$$

where $\mathcal{P}_1$ is the space of linear polynomials.

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

We shall now mention some properties of the finite element space. For any $v \in \mathcal{V}$, we have

$$\|v\|_{L^\infty(\Omega)} \lesssim h^{-d/p}\|v\|_{L^p(\Omega)}, \quad p \geq 1, \tag{5.3}$$

$$\|v\|_{H^1(\Omega)} \lesssim h^{-1}\|v\|, \tag{5.4}$$

$$\|v\|_{L^\infty(\Omega)} \lesssim c_d(h)\|v\|_{H^1(\Omega)}, \tag{5.5}$$

where $c_1(h) = 1, c_2(h) = |\log h|^{\frac{1}{2}}$ and $c_d(h) = h(2-d)/2$ for $d \geq 3$. The *inverse* inequalities (5.3) and (5.4) can be found, for example, in Ciarlet [21] and a proof of the discrete Sobolev inequality (5.5) can be found in Bramble and Xu [17].

Defining the L^2 projection $Q_h : L^2(\Omega) \mapsto \mathcal{V}$ by

$$(Q_h v, \chi) = (v, \chi) \quad \forall v \in L^2(\Omega), \chi \in \mathcal{V},$$

we have

$$\|v - Q_h v\| + h\|Q_h v\|_{H^1(\Omega)} \lesssim Ch\|v\|_{H^1(\Omega)}. \tag{5.6}$$

This estimate is well known; we refer to [17], [49], [50] for a rigorous proof and related results.

The finite element approximation to the solution of (4.1) is the function $u \in \mathcal{V}$ satisfying

$$a(u, v) = (F, v) \quad \forall v \in \mathcal{V}. \tag{5.7}$$

Define a linear operator $A : \mathcal{V} \mapsto \mathcal{V}$ by

$$(Au, v) = a(u, v), \qquad u, v \in \mathcal{V}.$$

The equation (5.7) is then equivalent to (1.1) with $f = Q_h F$. It is easy to see that $\rho(A) \eqsim h^{-2}$. The space $\mathcal{V}$ has a natural (nodal) basis $\{\phi_i\}_{i=1}^n$ ($n = \dim\mathcal{V}$) satisfying

$$\phi_i(x_l) = \delta_{il} \quad \forall i, l = 1, \ldots, n,$$

where $\{x_l : l = 1, \ldots, n\}$ is the set of all interior nodal points of $\mathcal{V}$. By means of these nodal basis functions, the solution of (5.7) is reduced to solving an algebraic system (4.16) with $\mathcal{A} = ((a\nabla\phi_i, \nabla\phi_l))_{n\times n}$ and $\eta = ((f, \phi_i)_{n\times 1})$.

It is well known that

$$h^d|\nu|^2 \lesssim \nu^t\mathcal{A}\nu \lesssim h^{d-2}|\nu|^2 \quad \text{and} \quad h^d|\nu|^2 \lesssim \nu^t\mathcal{M}\nu \lesssim h^d|\nu|^2 \quad \forall \nu \in \mathbb{R}^n. \tag{5.8}$$

Hence $\kappa(\mathcal{A}) \lesssim h^{-2}$ and $\kappa(\mathcal{M}) \lesssim 1$.

Now we define $R : \mathcal{V} \mapsto \mathcal{V}$ by

$$Rv = \omega h^{2-d}\sum_{i=1}^n (v, \phi_i)\phi_i \quad \text{or} \quad Rv = \omega\sum_{i=1}^n (a\nabla\phi_i, \nabla\phi_i)^{-1}(v, \phi_i)\phi_i. \tag{5.9}$$

Then we have, with $\lambda_h = \rho(A)$,

$$(Rv, v) = \lambda_h^{-1}(v, v). \tag{5.10}$$

In fact, using the techniques in §4.4, it is easy to see that this is equivalent to $\nu^t\mathcal{M}^2\nu \eqsim h^d\nu^t\mathcal{M}\nu$, which is a direct consequence of the second estimate of (5.8).

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

A similar argument shows that, for any given $\omega_1 \in (0,2)$, there exists a constant $c_0 > 0$ such that if $0 < \omega < c_0$

$$\lambda_h^{-1}(v,v) \lesssim (Rv,v) \leq \omega_1 \lambda_h^{-1}(v,v) \quad \forall v \in \mathcal{V}. \tag{5.11}$$

The operator R given by (5.9) corresponds to Richardson iteration or damped Jacobi iteration. Similar result also holds for Gauss–Seidel iteration.

Bibliographic comments. For general introduction to finite element methods, see Ciarlet [21]. A presentation of finite element spaces for multigrid analysis can be found in Xu [49]. The discrete Sobolev inequality given in (5.5) for $d = 2$ has appeared in many papers; the earliest reference appears to be Bramble [8].

6. Multilevel methods. From the space decomposition point of view, a multigrid algorithm is built upon the subspaces that are defined on a nested sequence of triangulations.

We assume that the triangulation $\mathcal{T}$ is constructed by a successive refinement process. More precisely, $\mathcal{T} = \mathcal{T}_J$ for some $J > 1$, and $\mathcal{T}_k$ for $k \leq J$ are a nested sequence of quasi-uniform triangulations; i.e., $\mathcal{T}_k$ consist of simplexes $\mathcal{T}_k = \{\tau_k^i\}$ of size h_k such that $\Omega = \cup_i \tau_k^i$ for which the quasi-uniformity constants are independent of k (cf. [21]) and τ_{k-1}^l is a union of simplexes of $\{\tau_k^i\}$. We further assume that there is a constant $\gamma < 1$, independent of k, such that h_k is proportional to γ^{2k}.

As an example, in the two-dimensional case, a finer grid is obtained by connecting the midpoints of the edges of the triangles of the coarser grid, with $\mathcal{T}_1$ being the given coarsest initial triangulation, which is quasi uniform. In this example, $\gamma = 1/\sqrt{2}$.

Corresponding to each triangulation $\mathcal{T}_k$, a finite element space $\mathcal{M}_k$ can be defined by

$$\mathcal{M}_k = \{v \in H_0^1(\Omega) : v|_\tau \in \mathcal{P}_1(\tau) \quad \forall \tau \in \mathcal{T}_k\}.$$

Obviously, the inclusion relation in (3.9) is satisfied for these spaces.

We assume that $h = h_J$ is sufficiently small and h_1 is of unit size. Note that $J = O(|\log h|)$. For each k, we define the interpolant $I_k : C(\bar{\Omega}) \mapsto \mathcal{M}_k$ by

$$(I_k u)(x) = u(x) \quad \forall x \in \mathcal{N}_k.$$

Here $\mathcal{N}_k$ is the set of all nodes in $\mathcal{T}_k$. By the discrete Sobolev inequalities (5.5), it can be easily shown that

$$\|(I_k - I_{k-1})v\|^2 + h_k^2 \|I_k v\|_A^2 \lesssim c_d(k) h_k^2 \|v\|_A^2, \quad v \in \mathcal{V}, \tag{6.1}$$

where $c_d(k) = 1, J-k$ and $2^{(d-2)(J-k)}$ for $d = 1, 2$ and $d \geq 3$, respectively.

6.1. Strengthened Cauchy–Schwarz inequalities. These types of inequalities were used as assumptions in §4 (see (4.10)). Here we shall establish them for multilevel spaces.

LEMMA 6.1. *Let $i \leq j$; then*

$$a(u,v) \lesssim \gamma^{j-i} h_j^{-1} \|u\|_A \|v\| \quad \forall u \in \mathcal{M}_i, v \in \mathcal{M}_j.$$

Here, we recall, that $\gamma \in (0,1)$ is a constant such that $h_i = \gamma^{2i}$.

Proof. It suffices to show that for any $K \in \mathcal{T}_i$,

$$\int_K a \nabla u \cdot \nabla v \lesssim \gamma^{j-i} h_j^{-1} \|u\|_{H^1(K)} \|v\|_{L^2(K)}, \tag{6.2}$$

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

since, by the Cauchy–Schwarz inequality, this implies

$$a(u,v) = \sum_{K\in\mathcal{T}_i}\int_K a\nabla u\cdot\nabla v \lesssim \gamma^{j-i}h_j^{-1}\sum_{K\in\mathcal{T}_i}\|u\|_{H^1(K)}\|v\|_{L^2(K)}$$

$$\lesssim \gamma^{j-i}h_j^{-1}\left(\sum_{K\in\mathcal{T}_i}\|u\|^2_{H^1(K)}\right)^{\frac12}\left(\sum_{K\in\mathcal{T}_i}\|v\|^2_{L^2(K)}\right)^{\frac12} = \gamma^{j-i}h_j^{-1}\|u\|_A\|v\|.$$

Given $v\in\mathcal{M}_j$, let $v_1\in\mathcal{V}$ be the unique function that vanishes on ∂K and equals to v at the interior nodes in K. Set $v_0 = v - v_1$; then $a(u,v) = a(u,v_1) + a(u,v_0)$. Since $\Delta u = 0$ on K, an application of Green's formula then gives

$$\begin{aligned}\int_K a\nabla u\cdot\nabla v_1 &= -\int_K(\nabla a\cdot\nabla u)v_1 \lesssim \|u\|_{H^1(K)}\|v_1\|_{L^2(K)}\\ &\lesssim \gamma^{j-i}h_j^{-1}\|u\|_{H^1(K)}\|v\|_{L^2(K)}.\end{aligned}$$

Let $T = \{\tau\in\mathcal{T}_j : \bar\tau\cap\partial K\neq\emptyset\}$; then supp $v_0\subset\bar T$ and $|T| = (\frac{h_i}{h_j})^{d-1}h_j^d = h_i^{d-1}h_j$. Since ∇u is constant on K, we have

$$\|\nabla u\|_{L^2(T)} = \frac{|T|^{\frac12}}{|K|^{\frac12}}\|\nabla u\|_{L^2(K)} \lesssim \left(\frac{h_i^{d-1}h_j}{h_i^d}\right)^{\frac12}\|\nabla u\|_{L^2(K)} \lesssim \gamma^{|i-j|}\|u\|_{H^1(K)}.$$

For the contributions from v_0, we have

$$\|\nabla v_0\|^2_{L^2(T)} \lesssim h_j^{d-2}\sum_{x\in\mathcal{N}_j\cap\partial K} v_0^2(x) = h_j^{d-2}\sum_{x\in\mathcal{N}_j\cap\partial K} v^2(x) \lesssim h_j^{-2}\|v\|^2_{L^2(K)}.$$

The estimate (6.2) follows. □

LEMMA 6.2. *Let $\mathcal{V}_i = (I_i - I_{i-1})\mathcal{V}$ or $\mathcal{V}_i = (Q_i - Q_{i-1})\mathcal{V}$; then*

$$a(u,v) \lesssim \gamma^{|i-j|}\|u\|_A\|v\|_A \quad \forall u\in\mathcal{V}_i, v\in\mathcal{V}_j. \tag{6.3}$$

Proof. By (6.1) or (5.6), we have

$$\|v\| \lesssim h_i\|v\|_A \quad \forall v\in\mathcal{V}.$$

The result then follows directly from Lemma 6.1. □

LEMMA 6.3. *Assume that $T_k = R_kA_kP_k$ and that $R_k : \mathcal{M}_k\mapsto\mathcal{M}_k$ satisfies*

$$\|R_kA_kv\|^2 \lesssim \lambda_k^{-1}(A_kv,v) \quad \forall v\in\mathcal{M}_k, \tag{6.4}$$

where $\lambda_k = \rho(A_k)$. Then, if $i\le j$

$$(u_i, T_jv)_A \lesssim \gamma^{j-i}\|u_i\|_A\|v\|_A \quad \forall u_i\in\mathcal{V}_i, v\in\mathcal{V}.$$

In general, for $1\le i,j\le J$,

$$(T_iu, T_jv)_A \lesssim \gamma^{|i-j|/2}(T_iu,u)^{\frac12}(T_jv,v)^{\frac12} \quad \forall u,v\in\mathcal{V}.$$

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Proof. An application of Lemma 6.1 yields

$$(u_i, T_j v)_A \lesssim \gamma^{j-i} h_j^{-1} \|u_i\|_A \|T_j v\|.$$

By (6.4)

$$\|T_j v\| = \|R_j A_j P_j v\| \lesssim h_j \|A_j^{\frac{1}{2}} P_j v\| \lesssim h_j \|v\|_A.$$

This proves the first inequality.

The second inequality follows from the Cauchy–Schwarz inequality and the inequality just proved:

$$\begin{aligned}(T_i u, T_j v)_A &\le (T_j v, v)_A^{\frac{1}{2}} (T_j T_i u, T_i u)_A^{\frac{1}{2}} \\ &\lesssim \gamma^{(j-i)/2} (T_j v, v)_A^{\frac{1}{2}} \|T_i u\|_A \lesssim \gamma^{(j-i)/2} (T_i u, u)_A^{\frac{1}{2}} (T_j v, v)_A^{\frac{1}{2}}. \qquad \square\end{aligned}$$

6.2. Hierarchical basis methods. With the multilevel spaces $\mathcal{M}_k$ $(1 \le k \le J)$ defined earlier, the hierarchical basis method is based on the decomposition of $\mathcal{V}$ into subspaces given by

$$\mathcal{V}_k = (I_k - I_{k-1})\mathcal{M} = (I - I_{k-1})\mathcal{M}_k \quad \text{for } k = 1, 2, \ldots, J. \tag{6.5}$$

Here $I_0 = 0$ and $I_k : \mathcal{M} \mapsto \mathcal{M}_k$ is the nodal value interpolant. It is not hard to see that (3.1) holds and is a direct sum. In fact, for any $v \in \mathcal{V}$, we have the following unique decomposition

$$v = \sum_{k=1}^{J} v_k \quad \text{with } v_k = (I_k - I_{k-1})v. \tag{6.6}$$

With the subspaces $\mathcal{V}_k$ given by (6.5), the operators A_k are all well conditioned. In fact, by (6.1) and (5.4), we can see that $h_k^{-2}\|v\| \eqsim a(v, v)$ for all $v \in \mathcal{V}_k$.

We assume that R_k satisfies

$$\lambda_k^{-2} \|v\|_A^2 \lesssim (R_k A_k v, v)_A \le \omega_1 (v, v)_A \quad \forall v \in \mathcal{V}_k. \tag{6.7}$$

The Richardson, Jacobi, and Gauss–Seidel iterations all satisfy this inequality.

LEMMA 6.4.

$$K_0 \lesssim c_d \quad \textit{and} \quad K_1 \lesssim 1,$$

where $c_1 = 1, c_2 = J^2$ *and* $c_d = 2^{(d-2)J}$ *for* $d \ge 3$.

Proof. For $v \in \mathcal{V}$, it follows from (6.6) and (6.1) that

$$\sum_{k=1}^{J} h_k^{-2} \|v_k\|^2 \lesssim c_d \|v\|_A^2.$$

This gives the estimate of K_0. The estimate of K_1 follows from Lemma 6.2 and Lemma 4.6. $\square$

Hierarchical basis preconditioner. By using Theorem 4.1 and Lemma 4.5, we obtain the following.

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

THEOREM 6.5. *Assume that R_k satisfies* (6.7); *then the* PSC *preconditioner* (3.6), *with $\mathcal{V}_k$ given by* (6.5), *satisfies*

$$\kappa(BA) \lesssim \begin{cases} 1 & \text{if } d = 1; \\ |\log h|^2 & \text{if } d = 2; \\ h^{2-d} & \text{if } d \geq 3. \end{cases} \tag{6.8}$$

In view of (4.19), the algebraic representation of the PSC preconditioner is

$$\mathcal{H} = \sum_{k=1}^{J} \mathcal{S}_k \mathcal{R}_k \mathcal{S}_k^t, \tag{6.9}$$

where $\mathcal{S}_k \in \mathbb{R}^{n \times (n_k - n_{k-1})}$ is the representation matrix of the nodal basis $\{\phi_i^k\}$ in $\mathcal{M}_k$, with $x_i^k \in \mathcal{N}_k \setminus \mathcal{N}_{k-1}$, in terms of the nodal basis $\{\phi^i\}$ of $\mathcal{M}$. We note that these nodal basis functions $\{\phi_i^k : x_i^k \in \mathcal{N}_k \setminus \mathcal{N}_{k-1}, k = 1, \ldots, J\}$ form a basis of $\mathcal{M}$, known as the *hierarchical basis.*

The well-known hierarchical basis preconditioner is a special case of (6.9) with $\mathcal{R}_k$ given by the Richardson iteration: $\mathcal{R}_k = h_k^{2-d}\mathcal{I}$. In this case

$$\mathcal{H} = \sum_{k=1}^{J} h_k^{2-d} \mathcal{S}_k \mathcal{S}_k^t.$$

For $d = 2$, we have $\mathcal{H} = \mathcal{S}\mathcal{S}^t$ with $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_J)$. Note that $\kappa(\mathcal{H}\mathcal{A}) = \kappa(\tilde{\mathcal{A}})$, where $\tilde{\mathcal{A}} = \mathcal{S}^t \mathcal{A} \mathcal{S}$ is just the stiffness matrix under the hierarchical basis (cf. Yserentant [56]).

The estimate of (6.8) shows that the hierarchical basis preconditioner is not very efficient when $d \geq 3$. A preconditioner that is optimal in any dimensions is presented in the next subsection.

Hierarchical iterative methods. For the SSC iterative method, we apply the Fundamental Theorem II with Lemma 6.4 and get

THEOREM 6.6. *Algorithm* 3.3 *with the subspaces $\mathcal{V}_k$ given by* (6.5) *satisfies*

$$\|E_J\|_A^2 \leq 1 - \frac{2 - \omega_1}{C c_d}$$

provided that R_k's satisfy (6.7) *with $\omega_1 < 2$.*

Compared with the usual multigrid method, the smoothing in the SSC hierarchical basis method is carried out only on the set of new nodes $\mathcal{N}_k \setminus \mathcal{N}_{k-1}$ on each subspace $\mathcal{M}_k$. The method proposed by Bank, Dupont, and Yserentant [5] can be viewed as such an algorithm with R_k given by an appropriate Gauss–Seidel iterations. Numerical examples in [5] show that the SSC algorithm converges much faster than the corresponding SSC algorithm.

6.3. Multigrid algorithms. Let $\mathcal{M}_k$ $(k = 1, \ldots, J)$ be the multilevel finite element spaces defined as in the preceding section. Again let $\mathcal{V} = \mathcal{M}_J$, but set $\mathcal{V}_k = \mathcal{M}_k$. In this case, the decomposition (3.1) is trivial.

We observe that, with $\mathcal{V}_k = \mathcal{M}_k$, there are redundant overlappings in the decomposition (3.1). The point is that these overlappings can be used advantageously to choose the subspace solvers in a simple fashion. Roughly speaking, the subspace solvers need

only to take care of those "non-overlapped parts" (which correspond to the so-called *high frequencies*). Technically, the assumption on R_k is

$$\lambda_k^{-2}\|A_k v\|^2 \lesssim \|R_k A_k v\|^2 \leq \omega_1 \lambda_k^{-1}(A_k v, v) \quad \forall v \in \mathcal{M}_k. \tag{6.10}$$

Note that (6.10) is a consequence of (5.11) and it implies

$$\rho(R_k) \gtrsim \lambda_k^{-1} \quad \text{and} \quad \rho(R_k A_k) \leq \omega_1. \tag{6.11}$$

As we have demonstrated in §5, all Richardson, damped Jacobi, and symmetric Gauss–Seidel iterations satisfy this assumption.

LEMMA 6.7. *For all* $v \in \mathcal{V}$

$$\sum_{k=1}^{J} \|(Q_k - Q_{k-1})\|^2_{H^1(\Omega)} \eqsim \|v\|^2_{H^1(\Omega)}.$$

A proof of this lemma can be found in the Appendix.

LEMMA 6.8. *Under the assumption* (6.10)

$$K_0 \lesssim 1 \quad \textit{and} \quad K_1 \lesssim 1.$$

Proof. By taking $v_i = (Q_i - Q_{i-1})v$ and applying Lemma 4.5, Lemma 6.7, and (6.11),

$$K_0 \lesssim 1.$$

The estimate for K_1 follows from Lemmas 4.6 and 6.3. □

Multilevel nodal basis preconditioners. Combining Lemma 6.8 and the first fundamental theorem (Theorem 4.1), we obtain the following.

THEOREM 6.9. *Assume that* R_k*'s satisfy* (6.10); *then the preconditioner* (3.6) *satisfies*

$$\kappa(BA) \lesssim 1.$$

Again, in view of (4.19), the algebraic representation of preconditioner (4.19) is

$$B = \sum_{k=1}^{J} \mathcal{I}_k \mathcal{R}_k \mathcal{I}_k^t, \tag{6.12}$$

where $\mathcal{I}_k \in \mathbb{R}^{n \times n_k}$ is the representation matrix of the nodal basis $\{\phi_i^k\}$ in $\mathcal{M}_k$ in terms of the nodal basis $\{\phi_i\}$ of $\mathcal{M}$, i.e.,

$$\Phi^k = \Phi \mathcal{I}_k$$

with $\Phi^k = (\phi_1^k, \ldots, \phi_{n_k}^k)$ and $\Phi = (\phi_1, \ldots, \phi_n)$.

Note that if we define $\mathcal{I}_k^{k+1} \in \mathbb{R}^{n_{k+1} \times n_k}$ such that

$$\Phi^k = \Phi^{k+1} \mathcal{I}_k^{k+1},$$

Then

$$\mathcal{I}_k = \mathcal{I}_{J-1}^J \cdots \mathcal{I}_{k+1}^{k+2} \mathcal{I}_k^{k+1}.$$

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

This identity is very useful for the efficient implementation of (6.12); cf. Xu and Qin [54]. If $\mathcal{R}_k$ are given by the Richardson iteration, we have

$$\mathcal{B} = \sum_{k=1}^{J} h_k^{2-d} \mathcal{I}_k \mathcal{I}_k^t. \tag{6.13}$$

From (6.12) or (6.13), we see that the preconditioner depends entirely on the transformation between the nodal bases on multilevel spaces. For this reason, we shall call such kinds of preconditioners *multilevel nodal basis preconditioners*.

Remark 6.1. Observing that S_k in (6.9) is a submatrix of $\mathcal{I}_k$ given in (6.12), we then have

$$(H\alpha, \alpha) \leq (B\alpha, \alpha) \quad \forall \alpha \in \mathbb{R}^n.$$

In view of the above inequality, if we take

$$\hat{H} = \sum_{k=1}^{J-1} h_k^{2-d} \mathcal{S}_k \mathcal{S}_k^t + I,$$

we obtain

$$(H\alpha, \alpha) \leq (\hat{H}\alpha, \alpha) \leq (B\alpha, \alpha) \quad \forall \alpha \in \mathbb{R}^n.$$

Even though $\hat{H}$ appears to be a very slight variation of H, numerical experiments have shown a great improvement over H for $d = 2$. We refer to [55] for the numerical results.

Multigrid iterative algorithms. We now consider Algorithm 3.3 with the multilevel subspace described as in the previous subsection. The following result follows directly from the Fundamental Theorem II (Theorem 4.4).

THEOREM 6.10. *Assume that the R_k's satisfy* (6.10) *with $\omega_1 < 2$; then the Algorithm* 3.3 *satisfies*

$$\|E_J\|_A^2 \leq 1 - \frac{2 - \omega_1}{C}.$$

The algebraic representation of Algorithm 3.3 is just Algorithm 4.1 with $\mathcal{I}_k$ defined in the previous subsection. As we know that the Algorithm 3.6 is mathematically equivalent to Algorithm 3.3, but their implementations are different.

To be more specific in the algebraic representation of Algorithm 3.6, we use the symmetric Gauss–Seidel method as the smoother.

ALGORITHM 6.1 (Algebraic representation of Algorithm 3.6). *Let $\mathcal{B}_1 = \mathcal{A}_1^{-1}$. Assume that $\mathcal{B}_{k-1} \in \mathbb{R}^{n_{k-1} \times n_{k-1}}$ is defined; then for $\eta \in \mathbb{R}^{n_k}$, $\mathcal{B}_k \in \mathbb{R}^{n_k \times n_k}$ is defined as follows*:

Step 1. $\nu^1 = \mathcal{B}_{k-1}(\mathcal{I}_{k-1}^k)^t \eta$;

Step 2. $\mathcal{B}_k \eta = \mathcal{I}_{k-1}^k \nu^1 + (\mathcal{D}_k - \mathcal{U}_k)^{-1} \mathcal{D}_k (\mathcal{D}_k - \mathcal{L}_k)^{-1} (\eta - \mathcal{A}_k \mathcal{I}_{k-1}^k \nu^1)$.

Let $\mathcal{B} = \mathcal{B}_J$, where the action of $\mathcal{B}_J$ is computed by Algorithm 6.1, the corresponding Algorithm 4.17 is mathematically equivalent to Algorithm 4.1. However, the operation counts are slightly different in these two algorithms.

Remark 6.2. The connection between the equations in terms of operators in finite element spaces and the equations in terms of matrices have not been well clarified before. Our theory shows that the so-called discrete inner products (often used in early papers

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

for the analysis of multigrid methods) are not necessary either for theoretical analysis or for practical implementation.

In a multigrid algorithm, prolongation and restriction are used to transfer the data between a coarser space and a finer space. In our formulation, the prolongation is the natural inclusion and the restriction is just the L^2 projection. Algebraically, they are just $\mathcal{I}_{k-1}^k$ and $(I_{k-1}^k)^t$. By choosing the prolongation and restriction this way and avoiding the use of discrete inner products, the theory of the multigrid method is much simplified.

Remark 6.3. An important feature of Theorem 6.10 is that its proof does not necessarily depend on much elliptic regularity of the differential equation (5.1). This makes it possible to analyze the multigrid algorithm for complicated problems.

For example, if there are large discontinuity jumps in the coefficient a in (5.1), the multigrid convergence could deteriorate because of these jumps. Nevertheless, this possible deterioration can be removed by introducing appropriate scaling as in the following scaled multilevel nodal basis preconditioner

$$\mathcal{B} = \sum_{l=1}^{J} h_k^{2-d} \mathcal{I}_k \mathcal{H}_k^{-1} \mathcal{I}_k^t,$$

where $\mathcal{H}_k = \operatorname{diag}(a_1^k, a_2^k, \ldots, a_{n_k}^k)$ and a_m^k is some proper arithmetic average of the jumps in a. It can be proved that $\kappa(\mathcal{BA}) \leq C|\log h|^\beta$ (independent of the jumps in a!) for some $\beta > 0$ (for $d = 2$ and in some cases for $d = 3$). Similar results also hold for V-cycle multigrid method. The proof of such results can be obtained by introducing certain weighted L^2 inner products and using the theory in this paper. For the discussion of the related problems, we refer to [17], [49], [50].

Another example is on the multigrid algorithms for locally refined meshes. In finite element discretizations, meshes are often locally refined to achieve better accuracy. The specially designed multigrid algorithms (cf. Brandt [19]) for such problems can be proven to be optimal by using the theory in this paper. For the discussion of this matter, we refer to [11], [13], [53].

Bibliographic comments. The strengthened Cauchy–Schwarz inequality appeared in Bank and Dupont [4]. It was established for the hierarchical basis preconditioner by Yserentant in [56] for two dimensions. The proof of Lemma 6.1 is obtained by modifying a proof of Yserentant [56].

The idea in hierarchical basis preconditioner for two-dimensional problems can be traced back to Bank and Dupont [4] for a two level method. The general multilevel hierarchical basis preconditioner was developed by Yserentant in [55]. Similar analysis in higher dimensions has been carried out by Ong [40] and Oswald [41]. A study of this method from an algebraic point of view is given by Axelsson and Vassilevski in [1], [2]. A new version of the algorithm is developed by Bank, Dupont, and Yserentant in [5]. The multilevel nodal basis preconditioner was first announced by Bramble and Xu in [18] and published in Bramble, Pasciak, and Xu [15] (see also Xu [49]). Uniform lower bound for $\lambda_{\min}(BA)$ is obtained in [15], [49] in some special cases. The uniform upper bound for $\lambda_{\max}(BA)$ has been proved by Zhang [58]. The optimal estimate in general case is obtained by Oswald [42], [43]. Lemma 6.7 can be found in [15], [49] in a slightly weak form and is implicitly contained in Oswald [43] (where the proof is based on Besov spaces). There exist several proofs of Lemma 6.7, cf. Bramble and Pasciak [11], Bornemann and Yserentant [7] and the Appendix of this paper.

The relationship between hierarchical basis and multilevel nodal basis preconditioners was first observed by Xu [49] and also discussed in [55]. For a study along these lines, see also Yserentant in [56].

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Kuo, Chan, and Tong [32] developed some multilevel filtering preconditioners which are quite similar to the multilevel nodal basis preconditioners. For a comparison of these algorithms (including hierarchical basis method), we refer to Tong, Chan, and Kuo [46].

For some special cases, Bramble, Pasciak, Wang, and Xu [13] observed that the discrete L^2 inner product in the multigrid algorithm is unnecessary. The conclusion in this paper is general.

Analysis of multigrid algorithms without use of elliptic regularity has been carried out by Bramble et al. [13], but the estimates of this paper are better and optimal. Similar results have also been obtained by Bramble and Pasciak [11].

7. Domain decomposition methods. The finite element space $\mathcal{V}$ is defined on a triangulation of the domain Ω, hence a finite element space restricted to a subdomain of Ω can naturally be regarded as a subspace of $\mathcal{V}$. Therefore a decomposition of the domain then naturally leads to a decomposition of the finite element space. This is the main viewpoint on the algorithms described in this section.

7.1. Preliminaries. We start by assuming that we are given a set of overlapping subdomains $\{\Omega_i\}_{i=1}^J$ of Ω whose boundaries align with the mesh triangulation defining $\mathcal{V}$. One way of defining the subdomains and the associated partition is by starting with disjoint open sets $\{\Omega_i^0\}_{i=1}^J$ with $\bar{\Omega} = \cup_{i=1}^J \bar{\Omega}_i^0$ and $\{\Omega_i^0\}_{i=1}^J$ quasi-uniform of size h_0. The subdomain Ω_i is defined to be a mesh subdomain containing Ω_i^0 with the distance from $\partial\Omega_i \cap \Omega$ to Ω_i^0 greater than or equal to ch_0 for some prescribed constant c.

Based on these subdomains, the subspaces $\mathcal{V}_i$ $(1 \le i \le J)$ are defined by

$$\mathcal{V}_i = \{v \in \mathcal{V} : v(x) = 0 \ \forall x \in \Omega \setminus \Omega_i\}.$$

If the number of subdomains J is too large, the above subspaces are not sufficient to produce an optimal algorithms. In regard to this consideration, we introduce a coarse finite element subspace $\mathcal{V}_0 \subset \mathcal{V}$ defined from a quasi-uniform triangulation of Ω of size h_0.

LEMMA 7.1. *For the subspaces $\mathcal{V}_i (0 \le i \le J)$, we have*

$$\mathcal{V} = \sum_{i=0}^{J} \mathcal{V}_i. \tag{7.1}$$

Furthermore there is a constant C_0 that is independent of h, h_0 or J, such that for any $v \in \mathcal{V}$, there are $v_i \in \mathcal{V}_i$ that satisfy $v = \sum_{i=0}^J v_i$ and

$$\sum_{i=0}^{J} a(v_i, v_i) \le C_0 a(v, v). \tag{7.2}$$

Proof. The main ingredient of the proof is a partition of unity, $\{\theta_i\}_{i=1}^J$, defined on Ω satisfying $\sum_{i=1}^J \theta_i = 1$ and, for $i = 1, \ldots, J$,

$$\text{supp}\theta_i \subset \Omega_i \cup \partial\Omega, \quad 0 \le \theta_i \le 1, \quad \|\nabla\theta_i\|_{\infty,\Omega_i} \le Ch_0^{-1}.$$

Here $\|\cdot\|_{\infty,D}$ denotes the L^∞ norm of a function defined on a subdomain D.

The construction of such a partition of unity is standard. A partition $v = \sum_{i=0}^J v_i$ for $v_i \in \mathcal{V}_i$ can then be obtained with

$$v_0 = Q_0 v, \quad v_i = I_h(\theta_i(v - Q_0 v)), \quad i = 1, \ldots, J,$$

where I_h is the nodal value interpolant on $\mathcal{V}$.

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

For this decomposition, we prove that (7.2) holds. For any $\tau \in \mathcal{T}_h$, note that

$$\|\theta_i - \bar{\theta}_{i,\tau}\|_{L^\infty(\tau)} \lesssim h\|\nabla\theta_i\|_{L^\infty(\tau)} \lesssim \frac{h}{h_0}.$$

Let $w = v - Q_0 v$; by the inverse inequality (5.4),

$$\begin{aligned} |v_i|_{H^1(\tau)} &\leq |\bar{\theta}_{i,\tau} w|_{H^1(\tau)} + |I_h(\theta_i - \bar{\theta}_{i,\tau})w|_{H^1(\tau)} \\ &\lesssim |w|_{H^1(\tau)} + h^{-1}\|I_h(\theta_i - \bar{\theta}_{i,\tau})w\|_{L^2(\tau)}. \end{aligned}$$

It can easily be shown that

$$\|I_h(\theta_i - \bar{\theta}_{i,\tau})w\|_{L^2(\tau)} \lesssim \frac{h}{h_0}\|w\|_{L^2(\tau)}.$$

Consequently

$$|v_i|^2_{H^1(\tau)} \lesssim |w|^2_{H^1(\tau)} + \frac{1}{h_0^2}\|w\|^2_{L^2(\tau)}.$$

Summing over all $\tau \in \mathcal{T}_h \cap \Omega_i$ gives

$$|v_i|^2_{H^1(\Omega)} = |v_i|^2_{H^1(\Omega_i)} \lesssim |w|^2_{H^1(\Omega_i)} + \frac{1}{h_0^2}\|w\|^2_{L^2(\Omega_i)},$$

and

$$\begin{aligned} \sum_{i=1}^J a(v_i, v_i) &\lesssim \sum_{i=1}^J |v_i|^2_{H^1(\Omega_i)} \lesssim \sum_{i=1}^J \left(|w|^2_{H^1(\Omega_i)} + \frac{1}{h_0^2}\|w\|^2_{L^2(\Omega_i)} \right) \\ &\lesssim \left(|v - Q_0 v|^2_{H^1(\Omega)} + \|v - Q_0 v\|^2_{L^2(\Omega)} \right) \lesssim \|v\|^2_{H^1(\Omega)}. \end{aligned}$$

For $i = 0$, we apply (5.6) and get

$$\|v_0\|_{H^1(\Omega)} \lesssim \|v\|_{H^1(\Omega)}.$$

The desired result then follows. □

LEMMA 7.2.

$$K_0 \leq C_0/\omega_0 \quad \textit{and} \quad K_1 \leq C.$$

Proof. The first estimate follows directly from Lemmas 4.5 and 7.1. To prove the second estimate, we define

$$Z_i = \{1 \leq j \leq J : \Omega_i \cap \Omega_j \neq \emptyset\}.$$

By the construction of the domain decomposition, there exists a fixed integer n_0 such that

$$|Z_i| \leq n_0 \quad \forall 1 \leq i \leq J.$$

Note that if $P_i P_j \neq 0$ or $P_j P_i \neq 0$, then $j \in Z_i$. An application of Lemma 4.7 (with $\mathcal{I}_0 = \{0\}$) gives $K_1 \leq \omega_1(1 + n_0)$. □

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

7.2. Domain decomposition methods with overlappings. By Theorem 4.1 and Lemma 7.2, we get the following theorem.

THEOREM 7.3. *The* SSC *preconditioner B given by* (3.6) *associated with the decomposition* (7.1) *satisfies*

$$\kappa(BA) \lesssim \frac{\omega_1}{\omega_0}.$$

The proof of the above theorem follows from Theorem 4.1 and Lemma 7.2.

Combining Theorem 4.4 with Lemma 7.2, we also obtain the following.

THEOREM 7.4. *The Algorithm* 3.3 *associated with the decomposition* (7.1) *satisfies*

$$\|E_J\|_A^2 \leq 1 - \frac{\omega_0(2-\omega_1)}{C}, \tag{7.3}$$

where C is a constant independent of the number of subdomains J and the mesh size h.

We note that in our theory, the subdomain problems do not have to be solved exactly and only the spectrum of the inexact solvers matters. In the estimate (7.3), ω_0 should not be too small and ω_1 should stay away from 2. It is easy to see that one iteration of a V-cycle multigrid on each subdomain always satisfies this requirement.

As for the implementation of these domain decomposition methods, the algebraic formulations (4.19) and Algorithm 4.1 can be used. For example, if exact solvers are used in subspace, the PSC preconditioner is

$$\mathcal{B} = \sum_{i=0}^{J} \mathcal{I}_i \mathcal{A}_i^{-1} \mathcal{I}_i^t.$$

Here $\mathcal{I}_i \in \mathbb{R}^{n \times n_i}$ is defined by

$$(\phi_1^i, \ldots, \phi_{n_i}^i) = (\phi_1, \ldots, \phi_n)\mathcal{I}_i,$$

where $(\phi_1^i, \ldots, \phi_{n_i}^i)$ is the nodal basis of $\mathcal{V}_i$ and $(\phi_1, \ldots, \phi_n)$ is the nodal basis of $\mathcal{V}$. Note that if $i \neq 0$, the entries of matrix $\mathcal{I}_i$ consists of 1 and zero, since $\{\phi_1^i, \ldots, \phi_{n_i}^i\}$ is a subset of $\{\phi_1, \ldots, \phi_n\}$.

7.3. Multigrid methods viewed as domain decomposition methods. As we observed earlier (see Propositions 3.2 and 3.3), multigrid and domain decomposition methods fit in the same mathematical framework. We shall now demonstrate some deeper relationship between these two algorithms.

Let $\mathcal{T}_J$ be the finest triangulation in the multilevel structure described earlier with nodes $\{x_i\}_{i=1}^{n_J}$. With such a triangulation, a natural domain decomposition is

$$\bar{\Omega} = \bar{\Omega}_0^h \bigcup \bigcup_{i=1}^{n_J} \operatorname{supp} \phi_i,$$

where ϕ_i is the nodal basis function in $\mathcal{M}_J$ associate with the node x_i and Ω_0^h (maybe empty) is the region where all functions in $\mathcal{M}_J$ vanish.

As in Example 3 of §3, the corresponding decomposition method without the coarser space is exactly the Gauss–Seidel method (see also [13]), which as we know is not very efficient (its convergence rate implied by (4.13) is known to be $1 - O(h^2)$). The more interesting case is when a coarser space is introduced. The choice of such a coarse space

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

is clear here, namely $\mathcal{M}_{J-1}$. There remains to choose a solver for $\mathcal{M}_{J-1}$. To do this, we may repeat the above process by using the space $\mathcal{M}_{J-2}$ as a "coarser" space with the supports of the nodal basis function in $\mathcal{M}_{J-1}$ as a domain decomposition. We continue in this way until we reach a coarse space $\mathcal{M}_1$ where a direct solver can be used. As a result, a multilevel algorithm based on domain decomposition is obtained.

PROPOSITION 7.5. *The recursively defined domain decomposition method described above is just a multigrid algorithm with the Gauss–Seidel method as a smoother.*

This conclusion follows from our earlier discussion. Another obvious conclusion is that the corresponding additive preconditioner of the above multilevel domain decompositions is just the multilevel nodal basis preconditioner (6.12) choosing $\mathcal{R}_k$ as Gauss–Seidel iterations. Such an additive multilevel domain decomposition method was also discussed by Dryja and Widlund [25] and Zhang [58]; but its close relationship with the preconditioner (6.12) was not addressed before.

Bibliographic comments. The mathematical ideas of domain decomposition methods can be traced back to Schwarz [44]. In the context of domain decomposition methods, the PSC preconditioner is also known as the additive Schwarz method, whereas the SSC Algorithm 3.3 is known as the multiplicative Schwarz method. The additive method was studied by Dryja and Widlund [23] and Matsokin and Nepomnyaschikh [36]. The proof of Lemma 7.1 is contained in [23]. For multiplicative methods, Lions [34] studied this method in a variational setting and established the uniform convergence in the two subdomain case. The convergence in multiple subdomain case has been studied by Widlund and his student Mathew in [35] for some special cases. General optimal estimates were established by Bramble et al. [14]. The derivation of the multigrid algorithm from the domain decomposition method appears to be new.

Another important class of domain decomposition method is often known as the *iterative substructuring* method. This type of algorithm uses the non-overlapping domain decomposition and also fits into our theory. In this direction, we refer to Bramble, Pasciak, and Schatz [12], Dryja and Widlund [24], and Smith [45].

For other domain decomposition like algorithms, we refer to Bank and Rose [6] for the marching algorithms.

Appendix: An equivalent norm in H^1. The purpose of this appendix is to present a proof for Lemma 6.7.

The main ingredients in the analysis are the fractional Sobolev spaces

$$H_0^{m+\sigma}(\Omega) \qquad (m \geq 0, 0 < \sigma < 1)$$

defined by the completion of $C_0^\infty(\Omega)$ in the following norm:

$$\|v\|_{H^{m+\sigma}(\Omega)} = \left(\|v\|^2_{H^m(\Omega)} + |v|^2_{H^{m+\sigma}(\Omega)}\right)^{\frac{1}{2}},$$

where

$$|v|^2_{H^{m+\sigma}(\Omega)} = \sum_{|\alpha|=m} \int_\Omega \int_\Omega \frac{|D^\alpha v(x) - D^\alpha v(y)|^2}{|x-y|^{d+2\sigma}} dx\, dy.$$

The following inverse inequality (cf. [16], [49]) holds:

$$\|v\|_{H^{1+\sigma}(\Omega)} \lesssim h_k^{-\sigma} \|v\|_{H^1(\Omega)}, \quad \|v\|_{H^s(\Omega)} \lesssim h_k^{-s} \|v\| \quad \forall v \in \mathcal{M}_k, \tag{8.1}$$

where $\sigma \in (0, \frac{1}{2})$ and $s \in [0, 1]$.

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Let $P_k : H_0^1(\Omega) \mapsto \mathcal{M}_k$ be the H^1 projection defined by

$$(\nabla P_k u, \nabla v_k) = (\nabla u, \nabla v_k) \quad \forall u \in H_0^1(\Omega), v_k \in \mathcal{M}_k.$$

It follows from the standard finite element approximation theory that

$$\|v - P_k v\|_{H^{1-\alpha}(\Omega)} \lesssim h_k^\alpha \|v\|_{H^1(\Omega)} \quad \forall v \in H_0^1(\Omega) \tag{8.2}$$

for some constant $\alpha \in (0, 1]$ (depending on the domain Ω).

By the definition of Q_k and the estimate (5.6), we have

$$\|Q_k v\| \le \|v\| \quad \forall v \in L^2(\Omega), \quad \|Q_k v\|_{H^1(\Omega)} \lesssim \|v\|_{H^1(\Omega)} \quad \forall v \in H_0^1(\Omega).$$

By interpolation, we have (for $\sigma \in (0, \frac{1}{2})$)

$$\|Q_k v\|_{H^\sigma(\Omega)} \lesssim \|v\|_{H^\sigma(\Omega)} \quad \forall v \in H_0^1(\Omega). \tag{8.3}$$

Lemma 6.7 is obviously a consequence of

PROPOSITION 8.6. *Let*

$$\|v\|_M^2 = \sum_{k=1}^{\infty} \|(Q_k - Q_{k-1})\|_{H^1(\Omega)}^2.$$

Then $\|v\|_M$ *is an equivalent norm in* H_0^1, *namely*,

$$\|v\|_M \eqsim \|v\|_{H^1(\Omega)} \quad \forall v \in H_0^1(\Omega).$$

Proof. Let $\tilde{Q}_k = Q_k - Q_{k-1}$ and $v_i = (P_i - P_{i-1})v$. It follows from (8.1), (8.3) and (8.2) that

$$\|\tilde{Q}_k v_i\|_{H^1(\Omega)}^2 \lesssim \lambda_k^\alpha \|\tilde{Q}_k v_i\|_{H^{1-\alpha}(\Omega)}^2 \lesssim \lambda_k^\alpha \|v_i\|_{H^{1-\alpha}(\Omega)}^2 \lesssim \lambda_k^\alpha h_i^{2\alpha} \|v_i\|_{H^1(\Omega)}^2.$$

Let $i \wedge j = \min(i, j)$, we have

$$\begin{aligned}
\|v\|_M^2 &= \sum_{k=1}^{\infty} \sum_{i,j=k}^{\infty} (\nabla \tilde{Q}_k v_i, \nabla \tilde{Q}_k v_j) = \sum_{i,j=1}^{\infty} \sum_{k=1}^{i\wedge j} (\nabla \tilde{Q}_k v_i, \nabla \tilde{Q}_k v_j) \\
&\lesssim \sum_{i,j=1}^{\infty} \sum_{k=1}^{i\wedge j} \lambda_k^\alpha h_i^\alpha h_j^\alpha \|v_i\|_{H^1(\Omega)} \|v_j\|_{H^1(\Omega)} \lesssim \sum_{i,j=1}^{\infty} \lambda_{i\wedge j}^\alpha h_i^\alpha h_j^\alpha \|v_i\|_{H^1(\Omega)} \|v_j\|_{H^1(\Omega)} \\
&\lesssim \sum_{i,j=1}^{\infty} \gamma^{\alpha|i-j|} \|v_i\|_{H^1(\Omega)} \|v_j\|_{H^1(\Omega)} \lesssim \sum_{i=1}^{\infty} \|v_i\|_{H^1(\Omega)}^2 = \|v\|_{H^1(\Omega)}^2.
\end{aligned}$$

To prove the other inequality, we use the strengthened Cauchy–Schwarz inequality and obtain (Lemma 6.1)

$$\begin{aligned}
\|v\|_{H^1(\Omega)}^2 &= \sum_{i,j=1}^{\infty} (\nabla \tilde{Q}_i v, \nabla \tilde{Q}_j v) \\
&\lesssim \sum_{i,j=1}^{\infty} \gamma^{|i-j|} \|\tilde{Q}_i v\|_{H^1(\Omega)} \|\tilde{Q}_j v\|_{H^1(\Omega)} \lesssim \sum_{i=1}^{\infty} \|\tilde{Q}_i v\|_{H^1(\Omega)}^2.
\end{aligned}$$

□

Bibliographic comments. The idea of using H^1-projections together with certain elliptic regularity was first contained in the proof of Lemma 10.3 of [49] but the argument here is much sharper. Our proof here resembles a proof in a recent paper by Bramble and Pasciak [11].

Acknowledgment. The author wishes to thank Professors James Bramble and Olof Widlund for various valuable comments on the paper.

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

REFERENCES

[1] O. Axelsson and P. Vassilevski, *Algebraic multilevel preconditioning methods*, I, Numer. Math., 56 (1989), pp. 157–177.

[2] ———, *Algebraic multilevel preconditioning methods*, II, Numer. Math., 57 (1990), pp. 1569–1590.

[3] N. Bakhvalov, *On the convergence of a relaxation method with natural constraints on the elliptic operator*, Zh. Vychisl. Mat. i Mat. Fiz., (1966).

[4] R. Bank and T. DuPont, *Analysis of a two-level scheme for solving finite element equations*, Report CNA–159, Center for Numerical Analysis, University of Texas at Austin, 1980.

[5] R. Bank, T. DuPont, and H. Yserentant, *The hierarchical basis multigrid method*, Numer. Math., 52 (1988), pp. 427–458.

[6] R. Bank and D. Rose, *Marching algorithms for elliptic boundary value problems.* I: *the constant coefficient case*, SIAM J. Numer. Anal., 14 (1977), pp. 792–829.

[7] F. Bornemann and H. Yserentant, *A basic norm equivalence for the theory of multilevel methods*, preprint, 1992.

[8] J. Bramble, *A second order finite difference analogue of the first biharmonic boundary value problem*, Numer. Math., 9 (1966), pp. 236–249.

[9] J. Bramble, Z. Leyk, and J. Pasciak, *Iterative schemes for nonsymmetric and indefinite elliptic bounday value problems*, preprint.

[10] J. Bramble and J. Pasciak, *New convergence estimates for multigrid algorithms*, Math. Comp., 49 (1987), pp. 311–329.

[11] ———, *New estimates for multilevel algorithms including* V*-cycle*, preprint, 1991.

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

[13] J. Bramble, J. Pasciak, J. Wang, and J. Xu, *Convergence estimates for multigrid algorithms without regularity assumptions*, Math. Comp., 57 (1991), pp. 23–45.

[14] ———, *Convergence estimate for product iterative methods with applications to domain decomposition and multigrid*, Math. Comp., 57 (1991), pp. 1–21.

[15] J. Bramble, J. Pasciak, and J. Xu, *Parallel multilevel preconditioners*, Math. Comp., 55 (1990), pp. 1–21.

[16] ———, *The analysis of multigrid algorithms with non-imbedded spaces or non-inherited quadratic forms*, Math. Comp., 55 (1991), pp. 1–34.

[17] J. Bramble and J. Xu, *Some estimates for a weighted L^2 projection*, Math. Comp., 56 (1991), pp. 463–476.

[18] ———, *A new multigrid preconditioner*, MSI Workshop on Practical Iterative Methods for Large Scale Computations, Minneapolis, MN, October 1991.

[19] A. Brandt, *Multilevel adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–309.

[20] X.-C. Cai and O. Widlund, *Domain decomposition algorithms for indefinite elliptic problems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 243–258.

[21] P. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, New York, 1978.

[22] M. Dryja and O. Widlund, *An additive variant of the Schwarz alternating method for the case of many subregions*, Tech. report, Courant Institute of Mathematical Sciences, New York, NY, 1987.

[23] ———, *Some domain decomposition algorithms for elliptic problems*, Tech. report, Courant Institute of Mathematical Sciences, New York, NY, 1989.

[24] ———, *Towards a unified theory of domain decomposition algorithms for elliptic problems*, in Domain Decomposition Methods for Partial Differential Equations, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990.

[25] ———, *Multilevel additive methods for elliptic finite element problems*, Courant Institute of Mathematical Sciences, New York, NY, preprint, 1990.

[26] R. Fedorenko, *A relaxing method for solving elliptic difference equations*, USSR Comput. Math. and Math. Phys., 1 (1961), pp. 1092–1096.

[27] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.

[28] G. Golub and D. O'Leary, *Some history of the conjugate gradient and Lanczos algorithms*: 1948–1976, SIAM Rev., 31 (1989), pp. 50–102.

[29] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, Heidelberg, 1985.

[30] L. HAGEMAN AND D. YOUNG, *Applied Iterative Methods*, Academic Press, New York, 1981.
[31] M. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, Nat. Bur. Std. J. Res., 49 (1952), pp. 409–436.
[32] J. KUO, T. CHAN, AND C. TONG, *Multilevel Filtering Elliptic Preconditioners*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 403–429.
[33] P. LIONS, *On the Schwarz alternating method*, in Proc. of the First International Symposium of Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. Golub, G. Meurant, and J. Periaux, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1987, pp. 1–42.
[34] J. MAITRE AND F. MUSY, *Multigrid methods: convergence theory in a variational framework*, SIAM J. Numer. Anal., 4 (1984), pp. 657–671.
[35] T. MATHEW, *Domain decomposition and iterative refinement methods for mixed finite element discretizations of elliptic problems*, Ph.D. thesis, Tech. report 463, New York University, New York, NY, 1989.
[36] A. M. MATSOKIN AND S. V. NEPOMNYASCHIKH, *A Schwarz method in a subspace*, Soviet Math., 29 (1985), pp. 78–84.
[37] S. MCCORMICK, ED., *Multigrid Methods*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1988.
[38] S. MCCORMICK, *Multilevel Adaptive Methods for Partial Differential Equations*, Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1989.
[39] S. MCCORMICK AND J. RUGE, *Unigrid for multigrid simulation*, Math. Comp., 41 (1986), pp. 43–62.
[40] E. ONG, *Hierarchical basis preconditioners for second order elliptic problems in three dimensions*, Ph.D. thesis, CAM Report 89-31, Dept. of Math., UCLA, Los Angeles, CA, 1990.
[41] P. OSWALD, *On estimates for hierarchic basis representations of finite element functions*, Tech. report N/89/16, Sektion Mathematik, Friedrich-Schiller Universität Jena, Jena, Germany, 1989.
[42] ———, *On function spaces related to finite element approximation theory*, Z. Anal. Anwendungen, 9 (1990), pp. 43–64.
[43] ———, *On discrete norm estimates related to multilevel preconditioners in the finite element method*, Proceeding of International conference on theory of functions, Varna, 1991, to appear.
[44] H. SCHWARZ, *Gesammelte Mathematische Abhandlungen* #2, Springer, Berlin, 1890.
[45] B. SMITH, *Domain decomposition algorithms for the partial differential equations of linear elasticity*, Ph.D. thesis, Tech. report 517, Dept. of Computer Science, Courant Institute, New York, NY, 1990.
[46] C. TONG, T. CHAN, AND J. KUO, *Multilevel filtering elliptic preconditioners: extensions to more general elliptic problems*, Dept. of Math., UCLA, Los Angeles, CA, 1990.
[47] R. VARGA, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.
[48] J. XU, *Two-grid finite element discretization for nonlinear elliptic equations*, SIAM J. Numer. Anal., submitted.
[49] ———, *Theory of Multilevel Methods*, Ph.D. thesis, Cornell University, Ithaca, NY, Rep AM-48, Pennsylvania State University, University Park, PA, 1989.
[50] ———, *Counter examples concerning a weighted L^2 projection*, Math. Comp., 57 (1991), pp. 563–568.
[51] ———, *Iterative methods by* SPD *and small subspace solvers for nonsymmetric or indefinite problems*, Proc. of the Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.
[52] ———, *A new class of iterative methods for nonselfadjoint or indefinite problems*, SIAM J. Numer. Anal., 29 (1992), pp. 303–319.
[53] ———, *Iterative methods by space decomposition and subspace correction: A unifying approach*, AM-67, Pennsylvania State University, University Park, PA, August 1990.
[54] J. XU AND X. CAI, *A preconditioned* GMRES *method for nonsymmetric or indefinite problems*, Math. Comp., (1992), to appear.
[55] J. XU AND J. QIN, *On some multilevel preconditioners*, AM-87, Pennsylvania State University, University Park, PA, August 1991.
[56] H. YSERENTANT, *On the multi-level splitting of finite element spaces*, Numer. Math., 49 (1986), pp. 379–412.
[57] ———, *Two preconditioners based on the multilevel splitting of finite element spaces*, Numer. Math., 58 (1990), pp. 163–184.
[58] X. ZHANG, *Multilevel Additive Schwarz Methods*, preprint.

Downloaded 06/22/13 to 128.135.12.127. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php