# AN INTRODUCTION TO MULTIGRID CONVERGENCE THEORY *

JINCHAO XU †

**Abstract.** An introduction is given in this paper to the basic idea and the convergence theory of multigrid methods. Brief discussions are first given to some basic properties of some elementary linear iterative methods such as Jacobi and Gauss-Seidel iterations and preconditioned conjugate gradient methods, and then more detailed discussions are given to a general framework of subspace correction method that can be applied to, among many other things, multigrid methods. A framework of auxiliary space method is also briefly presented for the construction of preconditioners. The multigrid method is introduced with a model elliptic boundary value problem of second order. Convergence estimates are obtained for basic multigrid methods such as backslash (\) cycle, V-cycle and W-cycle. Two different approaches are used in the convergence analysis. The first approach is the more traditional one that makes crucial use of elliptic regularity, while the second approach is based on the subspace correction framework that very weakly depends on the elliptic regularity. The first approach gives more precise estimates for simpler problems, while the second approach can be applied to more complex problems such as locally refined meshes and interface problems with large discontinuous jumps. As some more advanced topics, a general framework is briefly described on multigrid methods for nonnested multilevel subspaces and varying bilinear forms, and an optimal multigrid peconditioning technique is given for general unstructured grids using the auxiliary space framework. In addition to the aforementioned theoretical analysis, some discussions are also given to the implementation of some basic multigrid algorithms.

**AMS(MOS) subject classifications.** 65N22, 65F10, 65N30, 65N55.

* Lecture notes prepared for "Winter School on Iterative Methods in Scientitic Computing and Their Applications", December 14-20, 1995. Collection of notes for this winter school will be published by Springer.

† Department of Mathematics, Penn State University. Email: xu@math.psu.edu. The author is partially supported by NSF DMS94-03915-1 through Penn State and by NSF ASC-92-01266 and ONR-N00014-92-J-1890 through UCLA. This work was carried out while the author was visiting Department of Mathematics of UCLA. Special thanks go to the Applied Math Group at UCLA and especially to Professor Tony Chan.

**Contents**

**1. Introduction.** Multigrid methods are among the most efficient modern techniques for solving large scale algebraic systems arising from the discretization of partial differential equations. In this paper, we shall give a introduction to these methods and their convergence properties by considering their applications to a model elliptic boundary value problem of second order.

Multigrid methods have been most efficiently used in solving the linear algebraic system arising from the finite element discretizations of partial differential equations. The theory of the methods is an elegant combination of linear algebra, theory of finite element approximation and of partial differential equations. In this paper, we shall explore all these three aspects of the multigrid theory. We shall devote § 2, § 3 and § 4 to the technical materials for the theory of multigrid methods.

§ 2 is on the basic linear iterative methods and preconditioning concepts. Many elementary iterative methods, such as Jacobi and Gauss-Seidel iterations, are often the major components in a multigrid procedure, and also a multigrid method is often used in conjunction with a preconditioned conjugate gradient method. Therefore the materials in § 2 are fundamental to our multigrid algorithms and theory.

§ 3 is on an algebraic framework of subspace correction method (following Xu [32]) that can be in general used for construction and analysis of linear iterative methods. This framework will be a main technical tool in the analysis of multigrid methods in § 5.

The most technical materials in this paper are perhaps those in §4 for finite element approximation theory. In this section, some basic materials in finite elements are reviewed and some approximation results concerning multiple level of finite element spaces are presented. Some of these results depend crucially on the regularity theory for elliptic boundary value problems.

The core of this paper is §5 in which many major multigrid algorithms are introduced and analyzed. An attempt is made to explain the basic ideas behind multigrid methods and also to describe the implementation issues. But the major concern here is to present the multigrid convergence theory. The multigrid methods are analyzed with two different approaches. The first approach is the more traditional one which makes a crucial use of regularity theory of partial differential equations. The second approach is the subspace correction framework in §3.

The multigrid algorithms and their convergence analysis presented in §5 are only for the case that the underlying multilevel spaces are nested in the sense that the coarse spaces are subspaces of finer spaces. To give readers an idea on how multigrid methods can be applied to more complicated situations, we devote §6 to a general framework of nonnested multigrid methods which can be applied to cases like unstructured grids and nonconforming elements, and §7 to a special technical for construction optimal multigrid preconditioning technique for unstrutured grid using the framework of auxiliary space method.

Multigrid methods have been extensively studied in a vast literature by researchers in many different areas, a short article like this can only give a glimpse of small part of the whole subject. For further details, we refer to the tutorial book Briggs [15], research monographes Hackbusch [21, 22], McCormick [24], Wesseling [30], Bramble [5], review articles Xu [32] and Yserentant [37].

For convenience, following [32], the symbols $\lesssim, \gtrsim$ and $\eqsim$ will be used in this paper. That $x_1 \lesssim y_1, x_2 \gtrsim y_2$ and $x_3 \eqsim y_3$, mean that $x_1 \leq C_1 y_1, \quad x_2 \geq c_2 y_2$ and $c_3 x_3 \leq y_3 \leq C_3 x_3$ for some constants $C_1, c_2, c_3$ and $C_3$ that are independent of mesh parameters.

**2. Iterative and Preconditioning Methods.** Assume $\mathcal{V}$ is a finite dimensional vector space. The goal of this section is to study iterative methods and preconditioning techniques for solving the following kind of equation:

$$Au = f. \tag{2.1}$$

Here $A : \mathcal{V} \mapsto \mathcal{V}$ is an SPD linear operator over $\mathcal{V}$ and $f \in \mathcal{V}$ is given.

**2-a. Elementary linear iterative methods.** A single step linear iterative method which uses an old approximation, $u^{old}$, of the solution $u$ of (2.1), to produce a new approximation, $u^{new}$, usually consists of three steps:

1. Form $r^{old} = f - Au^{old}$;
2. Solve $Ae = r^{old}$ approximately: $\hat{e} = Br^{old}$;
3. Update $u^{new} = u^{old} + \hat{e}$,

where $B$ is a linear operator on $\mathcal{V}$ and can be thought of as an approximate inverse of $A$.

As a result, we have the following iterative algorithm.

**(2.2) Algorithm.** *Given* $u^0 \in \mathcal{V}$,

$$u^{k+1} = u^k + B(f - Au^k), \quad k = 0, 1, 2, \cdots.$$

The core of the above iterate scheme is the operator $B$. Notice that if $B = A^{-1}$, after one iteration, $u^1$ is then the exact solution. $B$ will be called an iterator of $A$.

We say that an iterative scheme like (2.2) converges if $\lim_{k\to\infty} u_k = u$ for any $u_0 \in \mathcal{V}$. Assume that $u$ and $u^k$ are solutions of (2.1) and (2.2) respectively, then

$$u - u^k = (I - BA)^k(u - u_0).$$

Therefore the iterative scheme (2.2) converges iff $\rho(I - BA) < 1$.

**(2.3) Symmetrization.** Sometimes it is more desirable that the iterator $B$ is symmetric. If $B$ is not symmetric, there is a natural way to symmetrize it. Consider the following iteration

$$\begin{aligned} u^{k+1/2} &= u^k + B(f - Au^k) \\ u^{k+1} &= u^{k+1/2} + B^t(f - Au^{k+1/2}) \end{aligned}$$

where "t" denotes the adjoint operator with respect to $(\cdot, \cdot)$. Eliminating the intermediate $u^{k+1/2}$ gives

$$u - u^{k+1} = (I - B^tA)(I - BA)(u - u^k)$$

or

$$u^{k+1} = u^k + \bar{B}(f - Au^k) \tag{2.4}$$

where, with "$*$" denoting the adjoint operator with respect to $(\cdot, \cdot)_A$,

$$\bar{B} = (I - (I - BA)^*(I - BA))A^{-1} = B^t + B - B^tAB \tag{2.5}$$

or

$$I - \bar{B}A = (I - BA)^*(I - BA). \tag{2.6}$$

Obviously $\bar{B}$ is symmetric and will be called the *symmetrization* of iterator $B$. The following identities obviously hold

$$(\bar{B}Av, v)_A = ((2I - BA)v, BAv)_A \quad \forall\, v \in \mathcal{V}. \tag{2.7}$$

and

$$||v||_A^2 - ||(I - BA)v||_A^2 = (\bar{B}Av, v)_A \quad \forall\, v \in \mathcal{V}. \tag{2.8}$$

A simple consequence of (2.8) is that

$$\lambda_{\max}(\bar{B}A) \leq 1.$$

**(2.9) Theorem.** *The following are equivalent:*

*1. The symmetrized scheme* (2.4) *is convergent.*

*2. The operator* $\bar{B}$ *given by* (2.5) *is SPD.*

*3. The matrix* $B^{-t} + B^{-1} - A$ *is SPD*

*4. There exists a constant* $\omega_1 \in (0, 2)$ *such that any one of the following is satisfied for any* $v \in \mathcal{V}$*:*

$$(BAv, BAv)_A \leq \omega_1 (BAv, v)_A; \tag{2.10}$$

$$(Av, v) \leq \omega_1 (B^{-1}v, v); \tag{2.11}$$

$$(\frac{2}{\omega_1} - 1)(Av, v) \leq ((B^{-1} + B^{-t} - A)v, v); \tag{2.12}$$

$$(2 - \omega_1)(Bv, v) \leq (\bar{B}v, v). \tag{2.13}$$

*Furthermore, the scheme 2.2 converges if (and only if, when* $B$ *is symmetric) its symmetrized scheme 2.4 converges.*

The above results can be proved easily by definition. We further notice that

$$(2 - \omega_1)B \leq \bar{B} \leq 2B. \tag{2.14}$$

**(2.15) Richardson iterative methods.** Richardson iteration is perhaps the simplest iterative method which correspond to (2.2) with $B = \frac{\omega}{\rho(A)} I$. Namely,

$$u^{k+1} = u^k + \frac{\omega}{\rho(A)}(f - Au^k), \quad k = 0, 1, 2, \cdots, \tag{2.16}$$

One can imagine that Richardson method is not very efficient method, but it is theoretically a very important one. One of the most important property of this method is its "smoothing property" that will be discussed now.

Let $A\phi_i = \lambda_i \phi_i$ with $\lambda_1 < \lambda_2 \leq \ldots \lambda_n$, $(\phi_i, \phi_j) = \delta_{ij}$, and $u - u^0 = \sum \alpha_i \phi_i$, then

$$u - u^k = \sum_i \alpha_i (1 - \omega\lambda_i/\lambda_h)^k \phi_i.$$

For a fixed $\omega \in (0, 2)$, it is clear that $(1 - \omega\lambda_i/\lambda_h)^k$ converges to zero very fast as $k \to \infty$ if $\lambda_i$ is close to $\lambda_h$. This exactly means that the high frequency modes in the error get damped out very quickly.

An iterative method (2.2) is said to be Richardson like if there exists an $\omega \in (0,2)$ such that

$$\|(I - BA)v\|_A \le \|(I - \frac{\omega}{\rho_A}A)v\|_A \quad \forall\, v \in \mathcal{V}. \tag{2.17}$$

**(2.18) Lemma.** *For the iterative method* (2.2)*, the followings are equivalent*

1. *The inequality* (2.17) *satisfies with* $\omega = C_0^{-1}$.
2. $(C_0\rho_A)^{-1}\|v\|^2 \le (\bar{B}v, v) \quad \forall\, v \in \mathcal{V}$.
3. $(C_0\rho_A)^{-1}\|Av\|^2 \le \|v\|_A^2 - \|(I - BA)v\|_A^2 \quad \forall\, v \in \mathcal{V}$.

**2-b. Jacobi and Gauss-Seidel Methods.** Assume $\mathcal{V} = \mathbb{R}^n$ and $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ is the usual SPD matrix. We write $A = D - L - U$ with $D$ being the diagonal of $A$ and $-L$ and $-U$ the lower and upper triangular part of $A$ respectively. The easiest approximate inverse of $A$ are perhaps

$$B = D^{-1} \quad \text{or} \quad B = (D - L)^{-1}.$$

As we shall see that these two choices of $B$ result in the well-known Jacobi and Gauss-Seidel methods. More generally, we have the following choice of $B$ that result in various different iterative methods:

$$B = \begin{cases} \omega & \text{Richardson;} \\ D^{-1} & \text{Jacobi;} \\ \omega D^{-1} & \text{Damped Jacobi;} \\ (D - L)^{-1} & \text{Gauss-Seidel;} \\ \omega(D - \omega L)^{-1} & \text{SOR.} \end{cases} \tag{2.19}$$

The symmetrization of the aforementioned Gauss-Seidel method is called the symmetric Gauss-Seideil method.

**(2.20) Theorem.** *Assume* $A$ *is SPD. Then*

- *Richardson method converges iff* $0 < \omega < 2/\rho(A)$;
- *Jacobi method converges iff* $2D - A$ *is SPD;*
- *Damped Jacobi method converges iff* $0 < \omega < 2/\rho(D^{-1}A)$;
- *Gauss-Seidel method always converges;*
- *SOR method converges iff* $0 < \omega < 2$.

The proof of the above results follow directly from Theorem 2.9 by (2.19) to compute $B^{-t} + B^{-1} - A$. For example, for SOR method, $B^{-t} + B^{-1} - A = \frac{2-\omega}{\omega}D$

**2-c. Alternative formulations of iterative schemes.** 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 $(\phi_1, \cdots, \phi_n)$ of $\mathcal{V}$ and a basis $(\psi_1, \cdots, \psi_m)$ of $\mathcal{W}$ is the matrix $\widetilde{A} \in \mathbb{R}^{m\times n}$ satisfying

$$(A\phi_1, \cdots, A\phi_n) = (\psi_1 \cdots, \psi_m)\widetilde{A}.$$

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 $\nu$ can be regarded as the matrix representation of $v$, denoted by $\nu = \widetilde{v}$.

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

$$\widetilde{AB} = \widetilde{A}\widetilde{B} \quad \text{and} \quad \widetilde{Av} = \widetilde{A}\widetilde{v}. \tag{2.21}$$

Under the basis $(\phi_k)$, we define the so-called mass matrix and stiffness matrix as follows

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

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

$$Rv = \sum_{i=1}^{n}(v, \phi_i)\phi_i, \quad \forall\, v \in \mathcal{V}. \tag{2.22}$$

Under a given basis $(\phi_k)$, the equation (2.1) can be transformed to an algebraic system

$$\mathcal{A}\mu = \eta. \tag{2.23}$$

Similar to (2.2), a linear iterative method for (2.23) can be written as

$$\mu^{k+1} = \mu^k + \mathcal{B}(\eta - \mathcal{A}\mu^k), \quad k = 0, 1, 2, \cdots, \tag{2.24}$$

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

**(2.25) Proposition.** *Assume that $\tilde{u} = \mu, \tilde{f} = \beta$ and $\eta = \mathcal{M}\beta$. Then $u$ is the solution of* (2.1) *if and only if $\mu$ is the solution of* (2.23). *The linear iterations* (2.2) *and* (2.24) *are equivalent if and only if $\widetilde{B} = \mathcal{B}\mathcal{M}$. 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 (2.22), we can show the following simple result.

**(2.26) Proposition.** *The scheme* (2.2) *represents the Richardson iteration for the equation* (2.23) *if $B$ is given by*

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

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

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

**2-d. Preconditioned conjugate gradient method.** The well-known conjugate gradient method is the basis of all the preconditioning techniques to be studied in this paper. The preconditioned conjugate gradient (PCG) method can be viewed as a conjugate gradient method applied to the preconditioned system:

$$BAu = Bf. \tag{2.27}$$

Here $B : V \mapsto V$ is another SPD operator and known as a preconditioner for $A$. Note that $BA$ is symmetric with respect to the inner product $(B^{-1}\cdot, \cdot)$. One version of this algorithm is as follows: *Given $u_0$; $r_0 = f - Au_0$; $p_0 = Br_0$; For $k = 1, 2, \ldots,$*

$$u_k = u_{k-1} + \alpha_k p_{k-1},\ r_k = r_{k-1} - \alpha_k A p_{k-1},\ p_k = Br_k + \beta_k p_{k-1},$$
$$\alpha_k = (Br_{k-1}, r_{k-1})/(Ap_{k-1}, p_{k-1}),\ \beta_k = (Br_k, r_k)/(Br_{k-1}, r_{k-1}).$$

It is well-known that

$$\|u - u_k\|_A \leq 2\Big(\frac{\sqrt{\kappa(BA)} - 1}{\sqrt{\kappa(BA)} + 1}\Big)^k \|u - u_0\|_A, \tag{2.28}$$

which implies that PCG converges faster with smaller condition number $\kappa(BA)$.

Observing the formulae in the PCG method and the convergence estimate (2.28), one sees that the efficiency of a PCG method depends on two main factors: the action of $B$ and the size of $\kappa(BA)$. Hence, a good preconditioner should have the properties that the action of $B$ is relatively easy to compute and that $\kappa(BA)$ is relatively small (at least smaller than $\kappa(A)$).

**3. Iterative methods by subspace correction.** Following Xu (1992) (see also Bramble-Pasciak-Wang-Xu [8, 7]), a general framework of constructing linear iterative methods and/or preconditioners can be obtained by the concept of *space decomposition* and *subspace correction*. This framework will be presented here from a purely algebraic point of view. Some simple examples are given for illustration and more important applications are given in the later sections for multigrid methods. This framework can also be applied directly to domain decomposition methods.

The presentation here more or less follows Xu [32]. The main modification is that the subspace solvers here may not be symmetric. For related topics, we refer to Bramble [5].

**3-a. Preliminaries.** A decomposition of a vector space $\mathcal{V}$ consists of a number of subspaces $\mathcal{V}_i \subset \mathcal{V}$ (for $0 \leq i \leq J$) such that

$$\mathcal{V} = \sum_{i=0}^{J} \mathcal{V}_i. \tag{3.1}$$

This means that, for each $v \in \mathcal{V}$, there exist $v_i \in \mathcal{V}_i$ $(0 \leq i \leq J)$ such that $v = \sum_{i=0}^{J} v_i$. This representation of $v$ may not be unique in general, namely (3.1) is not necessarily a direct sum.

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

$$(Q_i u, v_i) = (u, v_i), \quad (P_i u, v_i)_A = (u, v_i)_A, \quad u \in \mathcal{V}, v_i \in \mathcal{V}_i, \tag{3.2}$$

and

$$(A_i u_i, v_i) = (A u_i, v_i), \quad u_i, v_i \in \mathcal{V}_i. \tag{3.3}$$

$Q_i$ 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

$$A_i P_i = Q_i A. \tag{3.4}$$

This identity is of fundamental importance and will be used frequently in this chapter. A consequence of it is that, if $u$ is the solution of (2.1), then

$$A_i u_i = f_i \tag{3.5}$$

with $u_i = P_i u$ and $f_i = Q_i f$. This equation may be regarded as the restriction of (2.1) to $\mathcal{V}_i$.

We note that the solution $u_i$ of (3.5) is the best approximation of the solution $u$ (2.1) in the subspace $\mathcal{V}_i$ in the sense that

$$J(u_i) = \min_{v \in \mathcal{V}_i} J(v), \quad \text{with } J(v) = \frac{1}{2}(Av, v) - (f, v)$$

and

$$||u - u_i||_A = \min_{v \in \mathcal{V}_i} ||u - v||_A.$$

The subspace equation (3.5) will be in general solved approximately. To describe this, we introduce, for each $i$, another non-singular 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 (3.5) may be given by $\hat{u}_i = R_i f_i$.

**(3.6) Example.** Consider the space $\mathcal{V} = R^n$ and the simplest decomposition:

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

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

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

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

**3-b. Basic algorithms.** From the viewpoint of subspace correction, most linear iterative methods can be classified into two major algorithms, namely the *parallel subspace correction* (PSC) method and the *successive subspace correction* method (SSC).

*PSC: Parallel subspace correction.* This type of algorithm is similar to Jacobi method. The idea is to correct the residue equation on each subspace in parallel.

Let $u^{old}$ be a given approximation of the solution $u$ of (2.1). The accuracy of this approximation can be measured by the residual: $r^{old} = f - Au^{old}$. If $r^{old} = 0$ or very small, we are done. Otherwise, we consider the residual equation:

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

Obviously $u = u^{old} + e$ is the solution of (2.1). Instead we solve the restricted equation to each subspace $\mathcal{V}_i$

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

It should be helpful to note that the solution $e_i$ is the best possible correction $u^{old}$ in the subspace $\mathcal{V}_i$ in the sense that

$$J(u^{old} + e_i) = \min_{e \in \mathcal{V}_i} J(u^{old} + e), \quad \text{with } J(v) = \frac{1}{2}(Av, v) - (f, v)$$

and

$$||u - (u^{old} + e_i)||_A = \min_{e \in \mathcal{V}_i} ||u - (u^{old} + e)||_A.$$

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

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

An update of the approximation of $u$ is obtained by

$$u^{new} = u^{old} + \sum_{i=0}^{J} \hat{e}_i$$

which can be written as

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

where

$$B = \sum_{i=0}^{J} R_i Q_i. \tag{3.7}$$

We have therefore

**(3.8) Algorithm.** *Given $u_0 \in \mathcal{V}$, apply the iterative scheme* (2.2) *with $B$ given in* (3.7).

**(3.9) Example.** With $\mathcal{V} = \mathbb{R}^n$ and the decomposition given by 3-a, the corresponding (3.8) is just the Jacobi iterative method.

It is well-known that the Jacobi method is not convergent for all SPD problems (see Theorem 2.20, hence (3.8) is not always convergent. However the preconditioner obtained from this algorithm is of great importance. We note that The operator $B$ given by (3.7) is SPD if each $R_i : \mathcal{V}_i \to \mathcal{V}_i$ is SPD.

**(3.10) Algorithm.** *Apply the CG method to equation* (2.1)*, with $B$ defined by* (3.7) *as a preconditioner.*

**(3.11) Example.** The preconditioner $B$ corresponding to 3-a is

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

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

*SSC: Successive subspace correction.* This type of algorithm is similar to the Gauss-Seidel method.

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 $v^{-1} = u^{old}$ and correcting its residue in $\mathcal{V}_0$ gives

$$v^0 = v^{-1} + R_0 Q_0 (f - Av^{-1}).$$

By correcting the new approximation $v^1$ in the next space $\mathcal{V}_1$, we get

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

Proceeding this way successively for all $\mathcal{V}_i$ leads to

**(3.12) Algorithm.** *Given $u^0 \in \mathcal{V}$.*

*for* $k = 0, 1, \ldots$ *till convergence*
  $v \leftarrow u^k$
  *for* $i = 0 : J \quad v \leftarrow v + R_i Q_i (f - Av)$ *endfor*
  $u^{k+1} \leftarrow v$.
*endfor*

**(3.13) Example.** Corresponding to decomposition in Example 3-a, the Algorithm 3.12 is the Gauss-Seidel iteration.

**(3.14) Example.** More generally, decompose $\mathbb{R}^n$ as

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

where $1 = l_0 < l_1 < \cdots < l_{J+1} = n + 1$. Then (3.8), (3.10) and (3.12) are the block Jacobi method, block diagonal preconditioner and block Gauss-Seidel method respectively.

Let $T_i = R_i Q_i A$. By (3.4), $T_i = R_i A_i P_i$. 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 (2.1), then $f = Au$. Let $v^i$ be the $i-th$ iterate (with $v^0 = u^k$) from Algorithm 3.12, we have by definition

$$u - v^{i+1} = (I - T_i)(u - v^i), \quad i = 0, \cdots, J.$$

A successive application of this identity yields

$$u - u^{k+1} = E_J (u - u^k), \tag{3.15}$$

where

$$E_J = (I - T_J)(I - T_{J-1}) \cdots (I - T_1)(I - T_0). \tag{3.16}$$

**(3.17)** Remark. It is interesting to look at the operator $E_J$ in the special case that $R_i = \omega A_i^{-1}$ for all $i$. The corresponding SSC iteration is a generalization of the classic SOR method. In this case, we have

$$E_J = (I - \omega P_J)(I - \omega P_{J-1}) \cdots (I - \omega P_1)(I - \omega P_0).$$

One trivial fact is that $E_J$ is invertible when $\omega \neq 1$. Following an argument by Nicolaides [28] for the SOR method, let us take a look at the special case $\omega = 2$. Since, obviously, $(I - 2P_i)^{-1} = I - 2P_i$ for each $i$, we conclude that $E_J^{-1} = E_J^*$ where $*$ is the adjoint with respect to the inner product $(\cdot, \cdot)_A$. This means that $E_J$ is an orthogonal operator and, in particular, $||E_J||_A = 1$. As a consequence, the SSC iteration can not converge when $\omega = 2$. In fact, as we shall see in Proposition 3.42 below, in this special case, that the SSC method converges if and only if $0 < \omega < 2$.

The symmetrization of Algorithm (3.12) can also be implemented as follows.

**(3.18) Algorithm.** *Given* $u^0 \in \mathcal{V}$, $v \leftarrow u^0$
  *for* $k = 0, 1, \ldots$ *till convergence*
    *for* $i = 0 : J$ *and* $i = J : -1 : 0 \quad v \leftarrow v + R_i Q_i (f - Av)$ *endfor*
  *endfor*

The advantage of the symmetrized algorithm is that it can be used as a preconditioner. In fact, (3.18) can be formulated in the form of (2.2) with operator $B$ defined as follows: For $f \in \mathcal{V}$, let $Bf = u^1$ with $u^1$ obtained by (3.18) applied to (2.1) with $u^0 = 0$.

**(3.19) Colorization and parallelization of SSC iteration.** Associated with a given partition (3.1), a coloring of the set $\mathcal{J} = \{0, 1, 2, \ldots, J\}$ is a disjoint decomposition:

$$\mathcal{J} = \bigcup_{t=1}^{J_c} \mathcal{J}(t)$$

such that

$$P_i P_j = 0 \quad \text{for any} \quad i, j \in \mathcal{N}(t), i \neq j (1 \leq t \leq J_c).$$

We say that $i, j$ have the same color if they both belong to some $\mathcal{J}(t)$.

The important property of the coloring is that the SSC iteration can be carried out in parallel in each color.

**(3.20) Algorithm** (Colored SSC). *Given* $u^0 \in \mathcal{V}$, $v \leftarrow u^0$
*for* $k = 0, 1, \ldots$ *till convergence*
*for* $t = 1 : J_c \quad v \leftarrow v + \sum_{i \in \mathcal{J}(t)} R_i Q_i (f - Av)$ *endfor*
*endfor*

We note that the terms under the sum in the above algorithm can be evaluated in parallel (for each $t$, namely within the same color).

**3-c. Convergence theory.** The purpose of this section is to establish an abstract theory for algorithms described in previous sections.

In view of Theorem 2.9, it suffices to study (3.10) and (3.12). Two fundamental theorems will be presented.

For the preconditioner (3.10), we need to estimate the condition number of

$$T = BA = \sum_{i=0}^{J} T_i,$$

where $B$ is defined by (3.7) and $T_i = R_i A_i P_i$.

It is interesting to note the following special case:

$$BA = \sum_{i=0}^{J} P_i, \text{ if } R_i = A^{-1}.$$

For (3.12), 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 (3.16). Applying this estimate to (3.15) yields $||u - u^k||_A \leq \delta^k ||u - u^0||_A$.

**Important parameters.** The convergence theory here is to be built upon several parameters associated with the space decomposition and subspace solvers.

*Parameter* $\omega_1$. The first constant, named $\omega_1$, is the smallest constant satisfying

$$(T_i v, T_i v)_A \leq \omega_1 (T_i v, v)_A \quad \forall\, v \in \mathcal{V}, 0 \leq i \leq J. \tag{3.21}$$

or equivalently

$$(v_i, A_i v_i) \leq \omega_1 (R_i^{-1} v_i, v_i) \quad \forall\, v \in \mathcal{V}, 0 \leq i \leq J. \tag{3.22}$$

We assume that $R_i$ is chosen in such a way that $\omega_1$ is well-defined. If all $R_i$ are SPD, then $\omega_1$ is obviously well defined and in fact

$$\omega_1 = \max_{0 \leq i \leq J} \rho(R_i A_i) = \max_{0 \leq i \leq J} \rho(T_i).$$

The constant $\omega_1$ is, in most cases, very easy to estimate and its bounded-ness often comes as an assumption. For example, while all the subspace solvers are exact, namely $R_i = A_i^{-1}$, then $\omega_1 = 1$. As we shall see late, the convergence of an SSC method is assured if the following condition holds:

$$\omega_1 < 2.$$

This condition is equivalent to saying that the symmetrized schemes for all $R_i$ are convergent schemes (see Theorem 2.9) and in particular the iterative schemes given by all $R_i$ are convergent schemes.

*Parameter* $K_0$ *and* $\bar{K}_0$. The parameter $K_0$ to be introduced now plays the most crucial role in most applications and it is also most difficult to estimate in applications. It measures the correlation between space decomposition and the choice of subspace solvers. We define

$$K_0 = \sup_{\|v\|_A = 1} \inf_{v_i \in \mathcal{V}_i, \sum v_i = v} \sum_i (R_i^{-1} v_i, v_i).$$

and

$$\bar{K}_0 = \sup_{\|v\|_A = 1} \inf_{v_i \in \mathcal{V}_i, \sum v_i = v} \sum_i (\bar{R}_i^{-1} v_i, v_i).$$

In other words, for any $v \in \mathcal{V}$, there exists a decomposition $v = \sum_{i=0}^{J} v_i$ for $v_i \in \mathcal{V}_i$ such that

$$\sum_{i=0}^{J} (R_i^{-1} v_i, v_i) \leq K_0 (Av, v). \tag{3.23}$$

**(3.24) Lemma.** *Assume, for any* $v \in \mathcal{V}$, *there is a decomposition* $v = \sum_{i=0}^{J} v_i$ *with* $v_i \in \mathcal{V}_i$ *satisfying*

$$\sum_{i=0}^{J} (v_i, v_i)_A \leq C_0 (v, v)_A, \tag{3.25}$$

*then*

$$\bar{K}_0 \leq \frac{C_0}{\bar{\omega}_0}, \quad \textit{with} \quad \bar{\omega}_0 = \min_{0 \leq i \leq J} \lambda_{\min}(\bar{R}_i A_i)$$

*and, if all $R_i$ are SPD,*

$$K_0 \le \frac{C_0}{\omega_0} \quad \textit{with} \quad \textit{and} \quad \omega_0 = \min_{0\le i\le J} \lambda_{\min}(R_i A_i).$$

The above lemma is most useful in domain decomposition applications. A good upper bound of $K_0$ relies on a good lower bound of $\omega_0$, which means that each subspace solver $R_i$ should resolve the whole range of the spectrum of $A_i$. In another word, the subspace problems should be very well solved or preconditioned.

The constant $C_0$ in (3.25) only depends on the partition (decomposition) of the space and it is sometimes called *partition constant.*

**(3.26) Lemma.** *Assume, for any $v \in \mathcal{V}$, there is a decomposition $v = \sum_{i=0}^{J} v_i$ with $v_i \in \mathcal{V}_i$ satisfying*

$$\sum_{i=0}^{J} \rho(A_i)(v_i, v_i) \le \hat{C}_0 (v, v)_A,$$

*then then*

$$K_0 \le \frac{\hat{C}_0}{\check{\omega}_0} \quad \textit{with} \quad \check{\omega}_0 = \min_{0\le i\le J} (\lambda_{\min}(\bar{R}_i)\rho(A_i)),$$

*and, if all $R_i$ are SPD,*

$$K_0 \le \frac{\hat{C}_0}{\hat{\omega}_0} \quad \textit{with} \quad \textit{and} \quad \hat{\omega}_0 = \min_{0\le i\le J} (\lambda_{\min}(R_i)\rho(A_i)).$$

The above lemma is most useful in multigrid applications. A good upper bound of $K_0$ relies on a good lower bound of $\hat{\omega}_0$, which means that each subspace solver $R_i$ only needs to resolve the "upper" range of the spectrum of $A_i$. In another word, each subspace solver $R_i$ should be spectrally equivalent to $(\rho(A_i))^{-1}$.

*Parameter $K_1$ and $\bar{K}_1$.* This parameter measures the interaction among subspaces together with the subspaces solvers.

If each $R_i$ is SPD, we define $\epsilon_{ij} \in (0, 1]$, for $j < i$,

$$\epsilon_{ij}^2 = \rho(P_j T_i P_j)/\omega_1 \quad \text{and} \quad \epsilon_{ji} = \epsilon_{ij}, \quad \epsilon_{ii} = 1. \tag{3.27}$$

And we define $\bar{\epsilon}_{ij} \in (0, 1]$, for $j < i$,

$$\bar{\epsilon}_{ij}^2 = \rho(P_j \bar{T}_i P_j) \quad \text{and} \quad \epsilon_{ji} = \epsilon_{ij}, \quad \epsilon_{ii} = 1, \tag{3.28}$$

where with $\bar{R}_i$ being the symmetrization of $R_i$ (see § 2-a),

$$\bar{T}_i = \bar{R}_i A_i P_i. \tag{3.29}$$

Note that, for each $i \ge j$, $\epsilon_{ij}$ and $\bar{\epsilon}_{ij}$ are the smallest numbers satisfying

$$(T_i v_j, v_j)_A \le \omega_1 \epsilon_{ij}^2 (v_j, v_j)_A, \quad (\bar{T}_i v_j, v_j)_A \le \bar{\epsilon}_{ij}^2 (v_j, v_j)_A \quad \forall\, v_j \in \mathcal{V}_j.$$

**(3.30) Lemma.** *If each $R_i$ is SPD, then*

$$(T_i u, T_j v)_A \le \omega_1 \epsilon_{ij} (T_i u, u)_A^{\frac{1}{2}} (T_j v, v)_A^{\frac{1}{2}} \quad \forall\, u, v \in \mathcal{V}; \tag{3.31}$$

*Proof.* Without loss of generality, we may assume that $i \geq j$. It follows from Cauchy-Schwarz inequality that

$$\begin{aligned}(T_i u, T_j v)_A &\leq (T_i u, u)_A^{\frac{1}{2}} (T_i T_j v, T_j v)_A^{\frac{1}{2}} \\ &\leq \sqrt{\omega_1}\epsilon_{ij} (T_i u, u)_A^{\frac{1}{2}} (T_j v, T_j v)_A^{\frac{1}{2}} \\ &\leq \omega_1 \epsilon_{ij} (T_i u, u)_A^{\frac{1}{2}} (T_j v, v)_A^{\frac{1}{2}}.\end{aligned}$$

□

**(3.32)** Remark. Clearly $\epsilon_{ij} \leq 1$ and, $\epsilon_{ij} = 0$ if $P_i P_j = 0$. If $\epsilon_{ij} < 1$, the inequality (3.31) is often known as the *strengthened Cauchy-Schwarz inequality.*

**(3.33) Definition.**

$$K_1 = \min_{\mathcal{J}_0 \subset \{0:J\}} \quad \Big( |\mathcal{J}_0| + \max_{i \in \mathcal{J}_0^c} \sum_{j \in \mathcal{J}_0^c} \epsilon_{ij} \Big).$$

*and*

$$\bar{K}_1 = \min_{\mathcal{J}_0 \subset \{0:J\}} \quad \Big( |\mathcal{J}_0| + \max_{i \in \mathcal{J}_0^c} \sum_{j \in \mathcal{J}_0^c} \bar{\epsilon}_{ij} \Big).$$

Roughly speaking, $K_1$ is bounded if the matrix $(\epsilon_{ij})$ is sparse except for a few rows and columns.

**(3.34) Lemma.** *The parameter $K1$ admits the following estimates:*

1. $K_1 \leq J + 1$.
2. $K_1 \leq 1 + \rho((\epsilon_{ij})_{i,j=1:J}) \leq 1 + \max_{1 \leq i \leq J} \sum_{j=1}^{n} \epsilon_{ij}$.
3. *If $\epsilon_{ij} \lesssim \gamma^{|i-j|}$ or $\bar{\epsilon}_{ij} \lesssim \gamma^{|i-j|}$ for some $\gamma \in (0,1)$, then $\hat{K}_1 \lesssim \frac{1}{1-\gamma}$ or $\bar{K}_1 \lesssim \frac{1}{1-\gamma}$.*

**(3.35) Lemma.**

$$\sum_{i>j} (\bar{T}_i u_i, T_j v_j)_A \leq (\bar{K}_1 - 1) \Big( \sum_{i=0}^{J} (\bar{T}_i u_i, u_i)_A \Big)^{1/2} \Big( \sum_{j=0}^{J} (T_j v_j, T_j v_j)_A \Big)^{1/2}$$

*If each $R_i$ is SPD, then*

$$\sum_{i>j} (T_i u_i, T_j v_j)_A \leq \omega_1 (K_1 - 1) \Big( \sum_{i=0}^{J} (T_i u_i, u_i)_A \Big)^{1/2} \Big( \sum_{j=0}^{J} (T_j v_j, v_j)_A \Big)^{1/2}$$

*If each $R_i$ is SPD, then for any $S \subset \{0:J\} \times \{0:J\}$,*

$$\sum_{i,j \in S} (T_i u_i, T_j v_j)_A \leq \omega_1 K_1 \Big( \sum_{i=0}^{J} (T_i u_i, u_i)_A \Big)^{1/2} \Big( \sum_{j=0}^{J} (T_j v_j, v_j)_A \Big)^{1/2}$$

**Convergence theory.** With the parameters $\omega_1, K_0$ and $K_1$ introduced above, the convergence estimates for PSC and SSC methods can be neatly presented. The analysis for PSC preconditioner is relatively easy whereas the analysis for SSC iteration is less straightforward.

We first give a lower bound for the spectrum of the PSC preconditioner.

**(3.36) Lemma.** *Assume that all $R_i$ are SPD. The PSC preconditioner $B$ given by* (3.7) *satisfies*

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

*Proof.* If $v = \sum_{i=0}^{J} v_i$ is a decomposition that satisfies (3.23), then

$$(v,v)_A = \sum_{i=0}^{J}(v_i, v)_A = \sum_{i=0}^{J}(v_i, P_i v)_A,$$

and by the Cauchy-Schwarz inequality

$$\begin{aligned}\sum_{i=0}^{J}(v_i, P_i v)_A = \sum_{i=0}^{J}(v_i, A_i P_i v) &\le \sum_{i=0}^{J}(R_i^{-1}v_i, v_i)^{\frac{1}{2}}(R_i A_i P_i v, v)_A^{\frac{1}{2}} \\ &\le (\sum_{i=0}^{J}(R_i^{-1}v_i, v_i))^{\frac{1}{2}}(\sum_{i=0}^{J}(T_i v, v)_A)^{\frac{1}{2}} \le \sqrt{K_0}\|v\|_A (Tv, v)_A^{\frac{1}{2}}.\end{aligned}$$

Consequently

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

This implies that $\lambda_{\min}(BA) \ge K_o^{-1}$.

Now for $v = \sum_{i=0}^{J} v_i$ with $v_i = T_i T^{-1} v$, we have

$$\begin{aligned}K_0 &\le \max_{v\in\mathcal{V}} \frac{\sum_{i=0}^{J}(R_i^{-1} T_i T^{-1} v, T_i T^{-1} v)}{\|v\|_A^2} \\ &= \max_{v\in\mathcal{V}} \frac{(T^{-1}v, v)_A}{(v,v)_A} = (\lambda_{\min}(BA))^{-1}.\end{aligned}$$

The desired estimate then follows. □

**(3.37) Theorem.** *Assume all $R_i$ are SPDE. The PSC preconditioner $B$ given by* (3.7) *satisfies*

$$\lambda_{\min}(BA) = K_0^{-1} \quad \textit{and} \quad \lambda_{\max}(BA) \le \omega_1 K_1,$$

*and*

$$\kappa(BA) \le \omega_1 K_0 K_1.$$

*And in view of Lemmas 3.24 and 3.26,*

$$\kappa(BA) \le \frac{\omega_1}{\omega_0} C_0 K_1, \quad \kappa(BA) \le \frac{\omega_1}{\hat{\omega}_0} \hat{C}_0 K_1,$$

*Proof.* By Lemma 3.35,

$$||Tv||_A^2 = \sum_{i,j=0}^{J} (T_i v, T_j v)_A \le K_1 (Tv, v)_A \le K_1 ||Tv||_A ||v||_A,$$

which implies that $\lambda_{\max}(BA) \le K_1$. □

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

**(3.38) Lemma.** *Denote $E_{-1} = I$ and for $0 \le i \le J$,*

$$E_i = (I - T_i)(I - T_{i-1}) \cdots (I - T_1)(I - T_0).$$

*Then*

$$I - E_i = \sum_{j=0}^{i} T_j E_{j-1}, \tag{3.39}$$

*and for any $v \in \mathcal{V}$,*

$$||v||_A^2 - ||E_J v||_A^2 = \sum_{i=0}^{J} (\bar{T}_i E_{i-1} v, E_{i-1} v)_A \tag{3.40}$$

*where $\bar{T}_i$ is given by* (3.29).

*Furthermore if each $R_i$ is symmetric then*

$$||v||_A^2 - ||E_J v||_A^2 \ge (2 - \omega_1) \sum_{i=0}^{J} (T_i E_{i-1} v, E_{i-1} v)_A \tag{3.41}$$

*Proof.* The identity (3.39) follows immediately from the trivial identity $E_{i-1} - E_i = T_i E_{i-1}$. Similar to (2.8) and (2.7), we have

$$||E_{i-1} v||_A^2 - ||E_i v||_A^2 = ((2I - T_i) E_{i-1} v, T_i E_{i-1} v)_A = (\bar{T}_i E_{i-1} v, E_{i-1} v)_A.$$

Summing up these inequalities with respect to $i$ gives (3.40). The estimate (3.41) follows by combining (3.40) and (2.14). □

Again let us take a look at the special case that $R_i = \omega A_i^{-1}$ for each $i$. In this case, we have

$$||v||_A^2 - ||E_J v||_A^2 = \omega(2 - \omega) \sum_{i=0}^{J} ||P_i E_{i-1} v||_A^2.$$

This identity implies immediately that a necessary condition for the convergence for the related SSS method is that $0 < \omega < 2$. In fact, like in SOR method, it is not hard to see that this condition is also sufficient for the convergence (see Corollary 3.49 below). Thus, we have the following simple generalization of a classic result for the SOR method (see also Remark 3-b).

**(3.42) Proposition.** *The SSC method with $R_i = \omega A_i^{-1}$ for each $i$ converges if and only if $0 < \omega_1 < 2$.*

**(3.43) Lemma.** *Assume that $\omega_1 < 2$. If each $R_i$ is SPD, then*

$$\sum_{i=0}^{J}(T_i v, v)_A \le (1+K_1)^2 \sum_{i=0}^{J}(T_i E_{i-1}v, E_{i-1}v)_A \quad \forall\, v \in \mathcal{V}, \tag{3.44}$$

*and in general*

$$\sum_{i=0}^{J}(\bar{T}_i v, v)_A \le \left(1+\sqrt{\frac{\omega_1}{2-\omega_1}}(\bar{K}_1 - 1)\right)^2 \sum_{j=0}^{J}(\bar{T}_j E_{j-1}v, E_{j-1}v)_A \tag{3.45}$$

*Proof.* By (3.39)

$$\begin{aligned}(T_i v, v)_A &= (T_i v, E_{i-1}v)_A + (T_i v, (I - E_{i-1})v)_A \\ &= (T_i v, E_{i-1}v)_A + \sum_{j=0}^{i-1}(T_i v, T_j E_{j-1}v)_A.\end{aligned}$$

Applying the Cauchy-Schwarz inequality gives,

$$\sum_{i=0}^{J}(T_i v, E_{i-1}v)_A \le \left(\sum_{i=0}^{J}(T_i v, v)_A\right)^{\frac{1}{2}} \left(\sum_{i=0}^{J}(T_i E_{i-1}v, E_{i-1}v)_A\right)^{\frac{1}{2}},$$

and, by Lemma 3.35,

$$\begin{aligned}&\sum_{i=0}^{J}\sum_{j=0}^{i-1}(T_i v, T_j E_{j-1}v)_A \\ &\qquad \le \omega_1(K_1 - 1)\left(\sum_{i=0}^{J}(T_i v, v)_A\right)^{\frac{1}{2}} \left(\sum_{j=0}^{J}(T_j E_{j-1}v, E_{j-1}v)_A\right)^{\frac{1}{2}}.\end{aligned}$$

Combining these three formulae then leads to (3.44) and hence completes the proof for (3.47).

With arguments similar to the above (essentially by replacing $T_i$ by $\bar{T}_i$ in the above proof), it is easy to obtain that

$$\begin{aligned}\sum_{i=0}^{J}(\bar{T}_i v, v)_A \le &\left(\sum_{i=0}^{J}(\bar{T}_i v, v)_A\right)^{1/2} \left(\sum_{j=0}^{J}(\bar{T}_j E_{j-1}v, E_{j-1}v)_A\right)^{1/2} \\ &+(\bar{K}_1 - 1)\left(\sum_{i=0}^{J}(\bar{T}_i v, v)_A\right)^{\frac{1}{2}} \left(\sum_{j=0}^{J}(T_j E_{j-1}v, T_j E_{j-1}v)_A\right)^{\frac{1}{2}}.\end{aligned}$$

After canceling the common factor and using the following inequalities (see (2.11) and (2.10)):

$$(T_j w, w) \le (2-\omega_1)^{-1}(\bar{T}_j w, w), \quad (T_j w, T_j w) \le \omega_1(2-\omega_1)^{-1}(\bar{T}_j w, w),$$

The estimate (3.45) then follows easily. □

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

**(3.46) Theorem.** *Assume that $\omega_1 < 2$. If each $R_i$ is SPD, then the iterator $E_J$ (given by (3.16)) for the Algorithm 3.12 satisfies*

$$||E_J||_A^2 \le 1 - \frac{2-\omega_1}{K_0(1+\omega_1(K_1-1))^2}; \tag{3.47}$$

*and, in general,*

$$||E_J||_A^2 \le 1 - \frac{2-\omega_1}{\bar{K}_0(\sqrt{2-\omega_1}+\sqrt{\omega_1}(\bar{K}_1-1))^2} \tag{3.48}$$

*Proof.* The estimate in (3.47) is obviously equivalent to

$$||v||_A^2 \le \frac{K_0(1+K_1)^2}{2-\omega_1}(||v||_A^2 - ||E_J v||_A^2) \quad \forall\, v \in \mathcal{V}.$$

Estimate (3.47) then follows by combining (3.44) with (3.40) and (2.11).

The second estimate (3.48) then follows by combining (3.45) with the fact that $\lambda_{\min}(\sum_i \bar{T}_i) = \bar{K}_0^{-1}$ (similar to Lemma 3.36). □

As a direct consequence of the above theorem, we have the following simple result.

**(3.49) Corollary.** *A sufficient condition for the convergence of the SSC method is that*

$$\omega_1 < 2 \tag{3.50}$$

The condition (3.50) is also necessary in some sense, see Proposition 3.42.

**(3.51)** Remark. Note that the convergence estimate in Theorem 3.46 is independent of the order of how (3.12) is proceeded. Namely, if we shuffle the order in the decomposition (3.1), the corresponding estimate in (3.46) remains unchanged.

**(3.52) Theorem.** *Under the assumptions in Lemma 3.24,*

$$\kappa(BA) \le \frac{\omega_1}{\omega_0} C_0 K_1$$

*and*

$$||E_J||_A^2 \le \begin{cases} 1 - \dfrac{(2-\omega_1)\omega_0}{C_0(1+\omega_1(K_1-1))^2} & \text{if each } R_i \text{ is SPD} \\ 1 - \dfrac{\bar{\omega}_0}{C_0(\sqrt{2-\omega_1}+\sqrt{\omega_1}(K_1-1))^2} & \text{otherwise.} \end{cases}$$

**(3.53) Theorem.** *Under the assumptions in Lemma 3.26,*

$$\kappa(BA) \le \frac{\omega_1}{\hat{\omega}_0} \hat{C}_0\ K_1$$

*and*

$$||E_J||_A^2 \le \begin{cases} 1 - \dfrac{\hat{\omega}_0}{\hat{C}_0(1+\omega_1(K_1-1))^2} & \text{if each } R_i \text{ is SPD} \\ 1 - \dfrac{(2-\omega_1)\tilde{\omega}_0}{\hat{C}_0\sqrt{2-\omega_1}+\sqrt{\omega_1}(\bar{K}_1-1))^2} & \text{otherwise.} \end{cases}$$

**3-d. Matrix representations of PSC and SSC methods.** The PSC and SSC have been presented above in terms of projections and operators in abstract vector spaces. We shall now translate all these algorithms into explicit algebraic forms by using the simple techniques in § 2-c.

For each $k$, let $\mathcal{I}_k \in \mathbb{R}^{n\times n_k}$ be the matrix representation of the natural inclusion $I_k : \mathcal{V}_k \mapsto \mathcal{V}$; To derive the algebraic representation of the preconditioner (3.7), we rewrite it in a slightly different form

$$B = \sum_{k=0}^{J} I_k R_k Q_k.$$

Applying (2.21) and the easily verifiable identity $\widetilde{Q}_k = \mathcal{M}_k^{-1}\mathcal{I}_k^t\mathcal{M}$ gives

$$\widetilde{B} = \sum_{k=0}^{J} \widetilde{I}_k \widetilde{R}_k \widetilde{Q}_k = \sum_{k=0}^{J} \mathcal{I}_k(\mathcal{R}_k\mathcal{M}_k)(\mathcal{M}_k^{-1}\mathcal{I}_k^t\mathcal{M}) = \mathcal{B}\mathcal{M}.$$

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

$$\mathcal{B} = \sum_{k=0}^{J} \mathcal{I}_k\mathcal{R}_k\mathcal{I}_k^t. \tag{3.54}$$

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

$$\mathcal{B} = \begin{cases} \displaystyle\sum_{k=0}^{J} \rho(\mathcal{A}_k)^{-1}\mathcal{I}_k\mathcal{I}_k^t & \text{Richardson;} \\ \displaystyle\sum_{k=0}^{J} \mathcal{I}_k\mathcal{D}_k^{-1}\mathcal{I}_k^t & \text{Jacobi;} \\ \displaystyle\sum_{k=0}^{J} \mathcal{I}_k\mathcal{G}_k\mathcal{I}_k^t & \text{Gauss-Seidel.} \end{cases}$$

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 (2.25), we get

**(3.55) Proposition.** *The PSC preconditioner for the stiffness matrix $\mathcal{A}$ is given by* (3.54) *and* $\kappa(\mathcal{B}\mathcal{A}) = \kappa(BA)$.

Similarly, we can derive the algebraic representation of (3.12) for solving (2.23).

**(3.56) Algorithm.** *$\mu^0 \in \mathbb{R}^n$ is given. Assume that $\mu^k \in \mathbb{R}^n$ is obtained. Then $\mu^{k+1}$ 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 $i = 0 : J$.*

**4. Finite element approximations.** In the following sections, we shall introduce the multigrid methods. Our presentations will be confined on a second order elliptic model problem with the linear finite element discretization.

This section is devoted to some basic properties of finite element spaces that will be used for the analysis of multigrid algorithms.

**4-a. A model problem and finite element discretization.** We consider the boundary-value problem:

$$\begin{aligned} -\nabla \cdot a\nabla U &= F \quad \text{in } \Omega, \\ U &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{4.1}$$

where $\Omega \subset \mathbb{R}^d$ is a polyhedral domain and $a$ is a smooth function (or piecewise smooth) 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 (4.1) if and only if

$$a(U,\chi) = (F,\chi) \quad \forall \chi \in H_0^1(\Omega), \tag{4.2}$$

where

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

Introduce the fractional order Sobolev spaces

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

defined by the completion of smooth functions in the following norm:

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

where

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

It is well-known that there exists a constant $\alpha \in (0,1]$ such that

$$||U||_{1+\alpha} \leq C||F||_{\alpha-1}, \tag{4.3}$$

for the solution $U$ of (4.2), where $C$ is a constant depending on the domain $\Omega$ and the coefficient $a(x)$.

Assume that $\Omega$ is triangulated with $\Omega = \cup_i \tau_i$, where $\tau_i$'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 $\tau_i$ is contained in (contains) a ball of radius $C_1 h$ (respectively $C_0 h$). Define

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

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

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

$$||v||_{0,\infty,\Omega} \lesssim h^{-d/p}||v||_{L^p(\Omega)}, p \geq 1, \tag{4.4}$$

$$||v||_{1,\Omega} \lesssim h^{-1}||v||, \tag{4.5}$$

$$||v||_{1+\sigma,\Omega} \lesssim h^{-\sigma}||v||_{H^1(\Omega)} \quad \sigma \in (0,\frac{1}{2}), \tag{4.6}$$

$$||v||_{s,\Omega} \lesssim h^{-s}||v||_{0,\Omega} \quad s,t \in [0,1],\ t \leq s, \tag{4.7}$$

$$||v||_{L^\infty(\Omega)} \lesssim c_d(h)||v||_{H^1(\Omega)}, \tag{4.8}$$

where $c_1(h) = 1, c_2(h) = |\log h|^{\frac{1}{2}}$ and $c_d(h) = h^{\frac{2-d}{2}}$ for $d \geq 3$. The *inverse* inequalities (4.4) and (4.5) can be found, for example, in Ciarlet [19] and a proof of the discrete Sobolev inequality (4.8) can be found in Bramble and Xu [11]. A proof of (4.6) and (4.7) may be found in Bramble, Pasciak and Xu [10] and Xu [31].

**(4.9) Theorem.** *Assume that $P_h : H_0^1(\Omega) \mapsto \mathcal{V}$ is the Galerkin projection with respect to $a(\cdot,\cdot)$, then*

$$\|(I - P_h)u\|_{1-\alpha} \lesssim h^\alpha \|u\|_1 \quad \forall\, u \in H_0^1(\Omega), \tag{4.10}$$

*and*

$$\|(I - P_h)u\|_1 \lesssim h^s \|u\|_{1+s}, \quad \forall\, u \in H_0^1(\Omega) \cap H^{1+s}(\Omega),\ 0 \leq s \leq \alpha \tag{4.11}$$

*where $\alpha$ is as in* (4.3).

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

$$\|v - Q_h v\| + h\|Q_h v\|_{H^1(\Omega)} \lesssim Ch\|v\|_{H^1(\Omega)}. \tag{4.12}$$

This estimate is well-know, we refer to [31, 11] for a rigorous proof and related results.

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

$$\|Q_h v\|_{H^\sigma(\Omega)} \lesssim \|v\|_{H^\sigma(\Omega)} \quad \forall\, v \in H_0^1(\Omega). \tag{4.13}$$

and

$$\|v - Q_h v\|_{1-\alpha} \lesssim h^\alpha \|v\|_1 \quad \forall\, v \in H_0^1(\Omega). \tag{4.14}$$

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

$$a(u, v) = (F, v) \quad \forall v \in \mathcal{V}. \tag{4.15}$$

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

$$(Au, v) = a(u, v), \quad u, v \in \mathcal{V}. \tag{4.16}$$

The equation (4.15) is then equivalent to (2.1) with $f = Q_h F$. 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 (4.15) is reduced to solving an algebraic system (2.23) 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

$$h^d|\nu|^2 \lesssim \nu^t \mathcal{A}\nu \lesssim h^{d-2}|\nu|^2 \quad \text{and} \quad h^d|\nu|^2 \lesssim \nu^t \mathcal{M}\nu \lesssim h^d|\nu|^2 \quad \forall\, \nu \in \mathbb{R}^n. \tag{4.17}$$

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

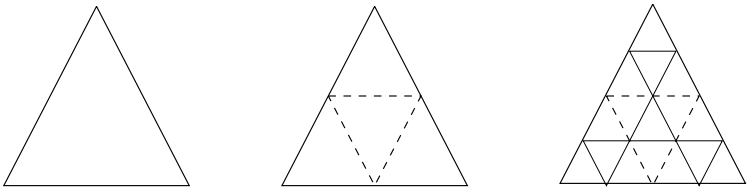

FIG. 1. *A typical multilevel grids*

**4-b. Finite element spaces on multiple levels.** This section is to study the interaction between finite element spaces with different scales. We assume that $\Omega$ has been triangulated with a nested sequence of quasi-uniform triangulations $\mathcal{T}_k = \{\tau_k^i\}$ of size $h$ for $k = 0, \ldots, j$ where the quasi-uniformity constants are independent of $k$. These triangulations should be nested in the sense that any triangle $\tau_{k-1}^l$ can be written as a union of triangles of $\{\tau_k^i\}$ (see Figure 1. We further assume that there is a constant $\eta > 1$, independent of $k$, such that

$$h_k \eqsim \eta^{-k}.$$

Associated with each $\mathcal{T}_k$, a finite element space $\mathcal{M}_k \subset H_0^1(\Omega)$ can be defined. One has

$$\mathcal{M}_1 \subset \mathcal{M}_2 \subset \ldots \subset \mathcal{M}_k \subset \ldots \subset \mathcal{M}_J. \tag{4.18}$$

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 $\mathcal{T}_k$.

Let $Q_k, P_k : H_0^1(\Omega) \mapsto \mathcal{M}_k$ be the $L^2$ and $H^1$ projection defined by

$$(Q_k u, v_k) = (u, v_k), \quad (\nabla P_k u, \nabla v_k) = (\nabla u, \nabla v_k) \quad \forall\, u \in H_0^1(\Omega), v_k \in \mathcal{M}_k. \tag{4.19}$$

**(4.20) Lemma.** *Let $R_k$ be any one of $I_k, Q_k$ or $P_k$. Then*

1. $R_i R_j = R_{i \wedge j}$.
2. $(R_i - R_{i-1})(R_j - R_{j-1}) = 0$ *if* $i \neq j$.
3. $(R_k - R_{k-1})^2 = R_k - R_{k-1} = (I - R_{k-1})R_k$.

**(4.21) Lemma.** *lm:Ik*

$$\|(I_k - I_{k-1})v\|^2 + h_k^2 \|I_k v\|_A^2 \lesssim c_d(k) h_k^2 \|v\|_A^2, \quad v \in \mathcal{V}, \tag{4.22}$$

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

**4-c. Regularity and approximation property.** Associated with each $\mathcal{M}_k$, we define, as in (3.3), $A_k : \mathcal{M}_k \mapsto \mathcal{M}_k$. The following result is instrumental in multigrid analysis.

**(4.23) Theorem.** *Assume $\alpha$ is as in* (4.3). *Then*

$$A((I - P_{k-1})u, u) \lesssim (\lambda_k^{-1} \|A_k u\|^2)^{\alpha} A(u, u)^{1-\alpha} \quad \forall\, u \in \mathcal{M}_k. \tag{4.24}$$

*Proof.* Let $u \in \mathcal{M}_k$. Applying Cauchy-Schwarz's inequality and the following norm equivalence (See Bank and Dupont [2])

$$||A_k^{s/2} v|| \eqsim ||v||_s \quad \forall\, H^s(\Omega) \cap H_0^1(\Omega) \quad s \in [0,1].$$

we deduce that

$$\begin{aligned} A((I-P_{k-1})u,u) &\leq (A_k^{\frac{1+\alpha}{2}} u, A_k^{\frac{1-\alpha}{2}}(I-P_{k-1})u) \\ &\leq ||A_k^{\frac{1+\alpha}{2}} u|| ||A_k^{\frac{1-\alpha}{2}}(I-P_{k-1})u|| \\ &\leq ||A_k^{\frac{1+\alpha}{2}} u|| ||(I-P_{k-1})u||_{1-\alpha}. \end{aligned}$$

By Hölder's inequality,

$$||A_h^{\frac{1+\alpha}{2}} u|| \leq \left(A(u,u)^{1-\alpha} ||A_h u||^{2\alpha}\right)^{1/2}. \tag{4.25}$$

Note that $\lambda_k \lesssim h_k^{-2}$, the theorem the follows by combining above inequalities with Theorem 4.9. ∎

**4-d. Strengthened Cauchy-Schwarz inequalities.** These types of inequalities were used as assumptions in Section 4 (see (3.31)). Here we shall establish them for multilevel spaces.

**(4.26) Lemma.** *Let $i \geq j$; then*

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

*Here, we recall, that $\gamma \in (0,1)$ is a constant such that $h_j \eqsim \gamma^{2j}$.*

*Proof.* Given $K \in \mathcal{T}_j$, it follows from Green's identity that

$$\begin{aligned} \int_K a\nabla u \cdot \nabla v &= \int_K \nabla a \cdot \nabla u v + \int_{\partial K} a \frac{\partial u}{\partial n} v \\ &\leq ||u||_{1,K} ||v|| + ||\nabla u||_{0,\partial K} ||v||_{0,\partial K} \\ &\lesssim ||u||_{1,K} ||v|| + (h_j^{-1/2} ||\nabla u||_{0,K})(h_i^{-1/2} ||v||_{0,K}) \\ &\lesssim (h_j h_i)^{-1/2} ||\nabla u||_{0,K} ||v||_{0,K} \\ &\lesssim \gamma^{i-j} h_i^{-1} ||\nabla u||_{0,K} ||v||_{0,K}. \end{aligned}$$

A repeated applications of Cauchy-Schwarz inequality yield

$$\begin{aligned} a(u,v) = \sum_{K \in \mathcal{T}_j} \int_K a\nabla u \cdot \nabla v &\lesssim \gamma^{i-j} h_j^{-1} \sum_{K \in \mathcal{T}_j} ||u||_{H^1(K)} ||v||_{L^2(K)} \\ &\lesssim \gamma^{i-j} h_j^{-1} \left( \sum_{K \in \mathcal{T}_j} ||u||^2_{H^1(K)} \right)^{\frac{1}{2}} \left( \sum_{K \in \mathcal{T}_j} ||v||^2_{L^2(K)} \right)^{\frac{1}{2}} = \gamma^{i-j} h_j^{-1} ||u||_A ||v||. \end{aligned}$$

∎

The inequality in the previous lemma is a generalization of the strengthen Cauchy inequality for hierarchical basis functions in Yserentant [36]. Our proof is similar in nature to that in [36], but appears to be a little shorter and more straightforward.

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

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

*Proof.* By (4.12), we have

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

The result then follows directly from Lemma 4.26. □

**(4.29) Lemma.** *Assume that* $T_k = R_k A_k P_k$ *and that* $R_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ *satisfies*

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

*where* $\lambda_k = \rho(A_k)$*. Then, for* $0 \le i, j \le J$

$$(T_i u, T_j v)_A \lesssim \gamma^{\frac{|i-j|}{2}}(T_i u, u)_A^{\frac{1}{2}}(T_j v, v)_A^{\frac{1}{2}} \quad \forall\, u, v \in \mathcal{V}.$$

*Proof.* If $i \le j$, an application of Lemma 4.26 yields

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

By the assumption on $R_k$,

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

Consequently

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

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

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

□

**4-e. An equivalent norm using multigrid splitting.** If nested multilevel finite element spaces $\mathcal{M}_k$ are allowed to get refined in an infinite way, namely $k \to \infty$, then the Sobolev space $H_0^1$ can be characterized by these finite element spaces in a very elegant way. We shall give such a characterization.

**(4.30) Theorem.** *For all* $v \in H_0^1(\Omega)$*,*

$$||v||_1^2 \eqsim \sum_{k=0}^{\infty}||(Q_k - Q_{k-1})v||_1^2 \eqsim \sum_{k=0}^{\infty} h_k^{-2}||(Q_k - Q_{k-1})v||.$$

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

$$\begin{aligned} ||\tilde{Q}_k v_i||_1^2 &\lesssim h_k^{-2\alpha} ||\tilde{Q}_k v_i||_{1-\alpha}^2 \quad \text{(by inverse inequality (4.7))} \\ &\lesssim h_k^{-2\alpha} ||v_i||_{1-\alpha}^2 \quad \text{(by (4.13))} \\ &\lesssim h_k^{-2\alpha} h_i^{2\alpha} ||v_i||_1^2 \quad \text{(by (4.14)).} \end{aligned}$$

Note that $v = \sum_i v_i$ Let $i \wedge j = \min(i, j)$, we have

$$\begin{aligned} &\sum_{k=0}^{\infty} ||(Q_k - Q_{k-1})v||_1^2 \\ &= \sum_{k=0}^{\infty} \sum_{i,j=k}^{\infty} (\nabla \tilde{Q}_k v_i, \nabla \tilde{Q}_k v_j) \quad (\text{since } \tilde{Q}_k v_i = 0 \text{ if } i < k) \\ &= \sum_{i,j=1}^{\infty} \sum_{k=0}^{i \wedge j} (\nabla \tilde{Q}_k v_i, \nabla \tilde{Q}_k v_j) \quad \text{(change the order of sum: Fubini theorem)} \\ &\lesssim \sum_{i,j=1}^{\infty} \sum_{k=0}^{i \wedge j} h_k^{-2\alpha} h_i^{\alpha} h_j^{\alpha} ||v_i||_1 ||v_j||_1 \lesssim \sum_{i,j=1}^{\infty} h_{i \wedge j}^{-2\alpha} h_i^{\alpha} h_j^{\alpha} ||v_i||_1 ||v_j||_1 \\ &\lesssim \sum_{i,j=1}^{\infty} \eta^{\alpha |i-j|} ||v_i||_1 ||v_j||_1 \lesssim \sum_{i=1}^{\infty} ||v_i||_1^2 = ||v||_1^2. \end{aligned}$$

To prove the other inequality, we use the strengthened Cauchy-Schwarz inequality and obtain (Lemma 4.26)

$$||v||_1^2 = \sum_{i,j=1}^{\infty} (\nabla \tilde{Q}_i v, \nabla \tilde{Q}_j v) \lesssim \sum_{i,j=1}^{\infty} \gamma^{|i-j|} ||\tilde{Q}_i v||_1 ||\tilde{Q}_j v||_1 \lesssim \sum_{i=1}^{\infty} ||\tilde{Q}_i v||_1^2.$$

□

**(4.31) Theorem.** *For all* $v \in H_0^1(\Omega)$,

$$||v||_1^2 \eqsim \sum_{k=0}^{\infty} h_k^{-2} ||(I - Q_{k-1})v||^2.$$

*Proof.* By previous theorem, we obviously have

$$\sum_{k=0}^{\infty} h_k^{-2} ||(I - Q_{k-1})v|| \geq \sum_{k=0}^{\infty} h_k^{-2} ||(Q_k - Q_{k-1})v|| \gtrsim ||v||_1^2.$$

The proof for the other direction of inequality is identical to that of the previous theorem except using $\tilde{Q}_k = I - Q_{k-1}$ instead of $Q_k - Q_{k-1}$. □

**(4.32) Theorem.** *For all* $v \in \tilde{H}_0^s(\Omega) (-1 \leq s \leq 1)$,

$$||v||_s^2 \eqsim \sum_{k=0}^{\infty} ||(Q_k - Q_{k-1})v||_s^2 \eqsim \sum_{k=0}^{\infty} h_k^{-2s} ||(Q_k - Q_{k-1})v||.$$

*Proof.* Set $B = \sum_{k=0}^{\infty} h_k^{-2}(Q_k - Q_{k-1})$. We then have $||v||^2 = (B^0 v, v)$ and, by previous theorem, $||v||_1^2 \eqsim (Bv, v)$. An application of operator interpolation then gives that $||v||_s^2 = (B^s v, v)$ which implies the desired result. □

**(4.33)** Remark. The above theorem is also valid for $-3/2 \le s \le 3/2$.

**(4.34)** Remark. A relevant interesting identity is as follows:

$$||v||_1^2 \eqsim \sum_{k=0}^{\infty} h_k^2 ||A_k P_k||^2 \quad \forall\, v \in H_0^1(\Omega).$$

**5. Multigrid methods.** This section is devoted to multigrid methods and their convergence properties. The following topics will be studied: classic multigrid iterative methods, BPX preconditioners, hierarchical basis methods, methods for locally refined meshes and full multigrid principle.

**5-a. Analysis for smoothers.** The most crucial step in developing a multigrid solver is the design of a relaxation scheme. A relaxation scheme is also the most problem-dependent part of a multigrid solver as most other parts (such as prolongation and restriction operators) are usually quite standard. The role of relaxation is not to reduce the overall error, but to smooth it out (namely damp out the non-smooth or high frequency components) so that it can be well approximated by functions on a coarser grid.

The smoother will be analyzed by three approaches in this section. The first approach is through numerical experiments, which would give an intuitive idea on the numerical behavior of a smoother. The second approach is Brandt's local mode analysis. This approach, using local Fourier analysis, can give a good insight on the role of a smoother. The third approach is to build technical machineries for the convergence analysis of multigrid methods.

**A model problem and some numerical examples.** Consider the Poisson equation with homogeneous Dirichlet condition on unit square discretized with uniform triangulation, the discretized equation can be expressed as

$$4u_{ij} - (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) = b_{i,j}, \quad 1 \le i, j \le n. \tag{5.1}$$

The damped Jacobi and Gauss-Seidel methods are among the most popular relaxation schemes for this problem. The damped Jacobi (or Richardson) iteration can be written as

$$4\tilde{u}_{ij} = \omega(\bar{u}_{i+1,j} + \bar{u}_{i-1,j} + \bar{u}_{i,j+1} + \bar{u}_{i,j-1}) + b_{i,j} \tag{5.2}$$

where $\tilde{u}_{ij}$ denote the new value of $u$ while $\bar{u}_{ij}$ denote the old value of $u$, and the (point) Gauss-Seidel iteration (with lexigraphical order on nodal points, from left to right and bottom to top):

$$4\tilde{u}_{ij} = (\bar{u}_{i+1,j} + \tilde{u}_{i-1,j} + \bar{u}_{i,j+1} + \tilde{u}_{i,j-1}) + b_{i,j}. \tag{5.3}$$

The Gauss-Seidel method has a good smoothing property. Let us illustrate this by a simple numerical example. Consider the equation (5.1) with an initial residual $u - u^0$ shown on the left plot of Figure 2. The initial residual apparently contains a

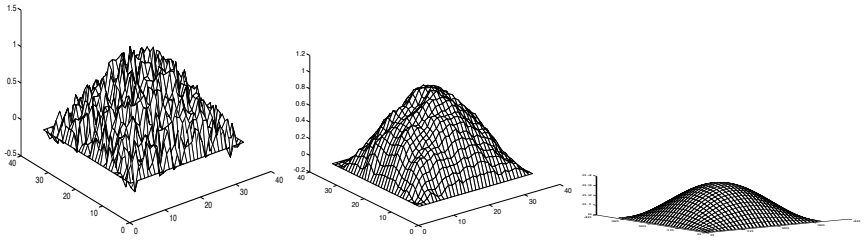

FIG. 2. *Residual after* 0, 2 *and* 100 *iterations, respectively, with* 961 *unknowns.*

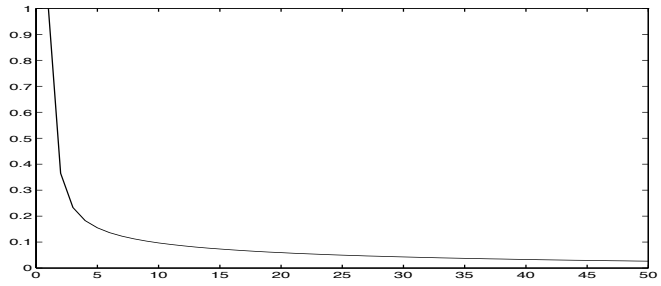


FIG. 3. *Convergence history of Gauss-Seidel method within 50 iterations*

lot of oscillations. The middle plot in Figure 2 is the residual after 2 Gauss-Seidel iterations. As we see the error components are smoothed out very quickly with only two Gauss-Seidel iterations although the global errors are still very large. The right plot in Figure 2 is the residual after one hundred iterations and as we see the error is still quite big.

*Basic ideas in a multigrid strategy.* The above numerical examples show that, high frequency errors, which involve local variations in the solution, are well annihilated by simple relaxation methods such as Gauss-Seidel iterations. Low-frequency or more global errors are much more insensitive to the application of simple relaxation methods. In fact, as shown in Figure 3, the convergence rate of the Gauss-Seidel iteration consists of a rather rapid initial residual reduction phase, which gradually develops into a much slower residual reduction phase, corresponding to a situation where all high-frequency errors have been damped down and low-frequency errors dominate. A multigrid methodology capitalizes on the this rapid initial high-frequency errors associated with an initial solution on the fine grid, using a simple relaxation scheme such as Gauss-Seidel iteration. Therefore, the solution is transfered to a coarse grid. On this grid, the low-frequency errors of the fine grid manifest themselves as high frequency errors, and are thus damped out efficiently using the same relaxation scheme. The coarse grid corrections computed in this manner are interpolated back to the fine grid in order to update the solution. This procedure can be applied recursively on a sequence of coarser and coarser grids, where each grid-level is responsible for eliminating a particular frequency bandwidth of errors.

Multigrid strategies may be applied to any existing relaxation technique. The success of the overall solution strategy depends on a close matching between the bandwidth of errors which can be efficiently smoothed on a given grid using the

particular chosen relaxation strategy, with a careful construction of a sequence of coarse grids, in order to represent the entire error frequency range.

**Brandt's local mode analysis.** The local mode analysis of Brandt [13] is a general effective tool to analyze and predict the performance of a multigrid solver and in particular the performance of a smoother. This method is based on the fact that a relaxation process is often a local process in which the information propagates just few mesh-sizes per sweep. Therefore, one can assume the problem to be in an unbounded domain, with constant (frozen) coefficients, in which case the algebraic error can be expanded in terms of Fourier series.

The local mode analysis for smoother can in fact be applied in a rigorous fashion to the model problem discussed in previous section. Let us first recall the discrete Fourier theory. For clarity, we confine our discussion in two dimensional case. The discrete Fourier transform theory says that every discrete function $u : I_n \mapsto \mathbb{R}$, with $I_n = \{(i,j) : 0 \leq i,j \leq n\}$, can be written as

$$u_{i,j} = \sum_{\theta \in \Theta_n} c_\theta \psi_{i,j}(\theta), \quad \psi_{i,j}(\theta) = e^{\mathbf{i}(i\theta_1 + j\theta_2)}, \quad \mathbf{i} = \sqrt{-1}, \quad \theta = (\theta_1, \theta_2).$$

where

$$c_\theta = \frac{1}{(n+1)^2} \sum_{(k,l) \in I_n} u_{k,l} \psi_{k,l}(-\theta).$$

and

$$\Theta_n = \{\frac{2\pi}{n+1}(k,l) \quad -m \leq k,l \leq m+p\},$$

where $p = 1, m = (n+1)/2$ for odd $n$ and $p = 0, m = n/2+1$ for even $n$.

We now first use the discrete Fourier transform to analyze the damped Jacobi method. Let $\tilde{\epsilon}_{i,j} = u_{i,j} - \tilde{u}_{i,j}$ and $\bar{\epsilon}_{i,j} = u_{i,j} - \bar{u}_{i,j}$. It is easy to see that

$$\tilde{\epsilon}_{ij} = \bar{\epsilon}_{i,j} - \frac{\omega}{4}(4\bar{\epsilon}_{ij} - (\bar{u}_{i+1,j} + \bar{u}_{i-1,j} + \bar{u}_{i,j+1} + \bar{u}_{i,j-1})). \tag{5.4}$$

We write

$$\tilde{\epsilon}_{i,j} = \sum_{\theta \in \Theta_n} \tilde{c}_\theta \psi_{i,j}(\theta) \tag{5.5}$$

and

$$\bar{\epsilon}_{i,j} = \sum_{\theta \in \Theta_n} \bar{c}_\theta \psi_{i,j}(\theta). \tag{5.6}$$

Substituting the above expressions into (5.4) and comparing the coefficients of each $\psi_{ij}(\theta)$, we obtain that

$$\lambda(\theta) = 1 - \omega(1 - \frac{\cos\theta_1 + \cos\theta_2}{2}). \tag{5.7}$$

where

$$\lambda(\theta) \equiv \frac{\tilde{c}(\theta)}{\bar{c}(\theta)} \tag{5.8}$$

is called the amplification factor of the local mode $\psi_{i,j}(\theta)$.

The smoothing factor introduced by Brandt is the following quantity

$$\bar{\rho} = \sup\{|\lambda(\theta)|, \pi/2 \le |\theta_k| \le \pi, k = 1, 2\}. \tag{5.9}$$

Roughly speaking, the smoothing factor $\bar{\rho}$ is the maximal amplification factor corresponding to those high frequency local modes that oscillate within $2h$ range (and hence can not be resolved by coarse grid of size $2h$).

For the damped Jacobi method, it is easy to see that

$$\bar{\rho} = \max\{|1 - 2\omega|, |1 - \omega/2|, |1 - 3\omega/2|\}.$$

The optimal $\omega$ that minimizes the smoothing factor is

$$\omega = 4/5, \quad \bar{\rho} = 3/5.$$

For $\omega = 1$ we have $\bar{\rho} = 1$. This means that the undamped Jacobi method for this model problem, although convergent as an iterative method by itself, should not be used as a smoother.

We next exam the smoothing property of the Gauss-Seidel iteration. Unlike the Jacobi method, Gauss-Seidel method depends on the ordering of the unknown. The most natural ordering is perhaps the lexicographic order which was used in the numerical examples given earlier and the corresponding Gauss-Seidel method reads

$$\tilde{\epsilon}_{ij} = \frac{1}{4}(\bar{\epsilon}_{i+1,j} + \tilde{\epsilon}_{i-1,j} + \bar{\epsilon}_{i,j+1} + \tilde{\epsilon}_{i,j-1}). \tag{5.10}$$

Again using the Fourier transform (5.5) and (5.6), we obtain the local amplification factor as follows:

$$\lambda(\theta) = \frac{e^{\mathbf{i}\theta_1} + e^{\mathbf{i}\theta_1}}{4 - e^{-\mathbf{i}\theta_1} - e^{-\mathbf{i}\theta_2}}.$$

It is elementary to see that

$$\bar{\rho} = |\lambda(\pi/2, \cos^{-1}(4/5))| = 1/2.$$

This means that Gauss-Seidel method is a better smoother than the damped Jacobi method.

A more interesting ordering for the Gauss-Seidel method is the so-called red-black ordering. In this particular example, we say two grid points belong to the same color (see $(3-b)$) if and only then they are not neighbors (in either horizontal or vertical direction). It is easy to see that the uniform grid in our example can be grouped into two colors, often called red color and black color. The red-black ordering is to first order all the nodes in one color and then order the other points in another color. (The actual ordering within the same color is not crucial).

The smoothing factor for the Gauss-Seidel method with red-black ordering can not be obtained as easily as the lexigraphical ordering, but it can indeed be proved that

$$\bar{\rho} = 1/4.$$

This means that the Gauss-Seidel method with red-black ordering is a better smoother than the one with the lexigraphical ordering. Furthermore red-black Gauss-Seidel has much better parallel feature (see $(3-b)$).

**General smoother analysis.** We shall now develop some technical results concerning the smoothing property of the Gauss-Seidel method. We choose to study Gauss-Seidel method since it is one of the better smoothers for our model problems and also it is less obvious to analyse. The analysis for other smoother is relatively simple (see the analysis for Richardson in (2.16)).

**(5.11) Lemma.** *For the stiffness matrix* $\mathcal{A} = \mathcal{D} - \mathcal{L} - \mathcal{U}$

$$||(\mathcal{D} - \mathcal{L})\xi||_2 \eqsim h^{d-2}||\xi||_2 \quad \forall\, \xi \in \mathbb{R}^N.$$

*Proof.* Because of the sparsity, it is trivial to prove that

$$||(\mathcal{D} - \mathcal{L})\xi||_2 \lesssim h^{d-2}||\xi||_2.$$

Now it follows that

$$\begin{aligned} h^{2-d}(\xi,\xi) &\lesssim \frac{1}{2}(\mathcal{D}\xi,\xi) \leq \frac{1}{2}((\mathcal{A}+\mathcal{D})\xi,\xi) \\ &= ((\mathcal{D}-\mathcal{L})\xi,\xi) \leq ||(\mathcal{D}-\mathcal{L})\xi||_2||\xi||_2. \end{aligned}$$

This completes the proof. □

The following result is an operator interpretation of the algebraic result given in Lemma 5.11, which means that the Gauss-Seidel is basically like Richardson iteration.

**(5.12) Lemma.** *Assume that* $R : \mathcal{V} \mapsto \mathcal{V}$ *represents the iterator for symmetric Gauss-Seidel iteration. Then*

$$R \eqsim h^2 \eqsim \lambda_h^{-1}.$$

*Proof.* By definition, the matrix representation of $R$ is

$$\tilde{R} = (\mathcal{D} - \mathcal{U})^{-1}\mathcal{D}(\mathcal{D} - \mathcal{L})^{-1}\mathcal{M}.$$

Given $v \in \mathcal{V}$, let $\nu = \tilde{v}$, then it is easy to see that

$$(Rv, v) = ||\mathcal{D}^{\frac{1}{2}}(\mathcal{D} - \mathcal{L})^{-1}\mathcal{M}\nu||_2^2$$

Thus it is equivalent to showing that

$$||\mathcal{D}^{\frac{1}{2}}(\mathcal{D} - \mathcal{L})^{-1}\mathcal{M}\nu||_2^2 \eqsim h^2(\mathcal{M}\nu, \nu).$$

Making a change of variable $\xi = (\mathcal{D} - \mathcal{L})^{-1}\mathcal{M}\nu$ and using the fact that $\mathcal{M}^{-1} \eqsim h^{-d}$, the above relation can be reduced to

$$||(\mathcal{D} - \mathcal{L})\xi||_2^2 \eqsim h^{d-2}(\mathcal{D}\xi,\xi) \eqsim h^{2(d-2)}||\xi||_2^2,$$

which was given by Lemma 5.11 □

**(5.13) Lemma.** *For the stiffness matrix* $\mathcal{A} = \mathcal{D} - \mathcal{L} - \mathcal{U}$

$$(\mathcal{A}\xi,\xi) \leq \frac{2}{1+k_0}((\mathcal{D} - \mathcal{L})\xi,\xi) \quad \forall\, \xi \in \mathbb{R}^N,$$

*where $k_0$ is the maximal number of nonzero row entries of $A$.*

*If $R : \mathcal{V} \mapsto \mathcal{V}$ represents the iterator for Gauss-Seidel iteration. Then*

$$(Av, v) \leq \frac{2}{1+k_0}(R^{-1}v, v) \quad \forall\, v \in V_h.$$

*Proof.* It is easy to see that the desired estimate is equivalent to the following:

$$(\mathcal{A}\xi, \xi) \leq k_0(\mathcal{D}\xi, \xi) \quad \forall\, \xi \in \mathbb{R}^N$$

which can be obtained by a simple of application of Cauchy-Schwarz inequality. □

**5-b. A basic multigrid cycle: backslash (\) cycle.** Although, as we shall see, multigrid methods have many variants, there is one particular multigrid algorithm which can be viewed a basic multigrid cycle. This algorithm is sometimes called the backslash (\) cycle (we shall explain below why this algorithm is given this name).

We shall first present this method from a more classic point of view. This more classic approach makes it easier to introduce many different kinds of classic multigrid methods and also make it possible to use more classic approach to analyze the convergence of multigrid methods.

A multigrid process can be viewed as defining a sequence of operators $B_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ which are approximate inverse of $A_k$ in the sense that $||I - B_k A_k||_A$ is bounded away from 1. A typical way of defining such a sequence of operators is the following *backslash* cycle multigrid procedure.

**(5.14) Algorithm.** *For $k = 0$, define $B_0 = A_0^{-1}$. Assume that $B_{k-1} : \mathcal{M}_{k-1} \mapsto \mathcal{M}_{k-1}$ is defined. We shall now define $B_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ which is an iterator for the equation of the form*

$$A_k v = g.$$

1. *Fine grid smoothing: For $v^0 = 0$ and $l = 1, 2, \cdots, m$*

$$v^l = v^{l-1} + R_k(g - A_k v^{l-1})$$

2. *Coarse grid correction: $e_{k-1} \in \mathcal{M}_{k-1}$ is the approximate solution of the residual equation $A_{k-1}e = Q_{k-1}(g - Av^m)$ by the iterator $B_{k-1}$:*

$$e_{k-1} = B_{k-1}Q_{k-1}(g - Av^m).$$

*Define*

$$B_k g = v^m + e_{k-1}.$$

After the first step, the residual $v - v^m$ is small on high frequencies. In another word, $v - v^m$ is smoother (see the middle plot in Figure 2) and hence it can be very well approximated by a coarse space $\mathcal{M}_{k-1}$. The second step in the above algorithm plays role of correcting the low frequencies by the coarser space $\mathcal{M}_{k-1}$ and the coarse grid solver $B_{k-1}$ given by induction.

With the above defined $B_k$, we may consider the following simple iteration

$$u^{k+1} = u^k + B_J(f - Au^k) \tag{5.15}$$

There are many different ways to make use of $B_k$, which will be discussed late.

Before we now study its convergence, we now discuss briefly the algebraic version of the above algorithm.

Let $\Phi^k = (\phi_1^k, \cdots, \phi_{n_k}^k)$ be the nodal basis vector for the space $\mathcal{M}_k$, we define the so-called prolongation matrix $\mathcal{I}_k^{k+1} \in \mathbb{R}^{n_{k+1} \times n_k}$ as follows

$$\Phi^k = \Phi^{k+1} \mathcal{I}_k^{k+1}, \tag{5.16}$$

**(5.17) Algorithm** (MATRIX VERSION). *Let $\mathcal{B}_0 = \mathcal{A}_0^{-1}$. 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:*

1. *Fine grid smoothing: For $\nu^0 = 0$ and $l = 1, 2, \cdots, m$*

$$\nu^l = \nu^{l-1} + \mathcal{R}_k(\eta - \mathcal{A}_k \nu^{l-1})$$

2. *Coarse grid correction: $\varepsilon_{k-1} \in \mathbb{R}^{n_{k-1}}$ is the approximate solution of the residual equation $\mathcal{A}_{k-1}\varepsilon = (\mathcal{I}_{k-1}^k)^t(\eta - \mathcal{A}_k \nu^m)$ by using $\mathcal{B}_{k-1}$*

$$\varepsilon_{k-1} = \mathcal{B}_{k-1}(\mathcal{I}_{k-1}^k)^t(\eta - \mathcal{A}_k \nu^m).$$

*Define*

$$\mathcal{B}_k \eta = \nu^m + \mathcal{I}_{k-1}^k \varepsilon_{k-1}.$$

The above algorithm is given in recurrence. But it can also be easily implemented in a non-recursive fashion. For such type of implementation, we refer to Algorithm 5.24

**5-c. A convergence analysis using full elliptic regularity.** With the assumption of full elliptic regularity (namely $\alpha = 1$ in (4.3), see also Theorem 4.23), a very sharp convergence estimate can be obtained in a very simple and elegant fashion.

We shall assume that the smoothers $R_k$ are SPD and satisfies

$$\frac{c_0}{\lambda_k}(v, v) \leq (R_k v, v) \leq (A_k^{-1} v, v) \quad \forall\, v \in \mathcal{M}_k. \tag{5.18}$$

We would like to remark that the above assumptions on $R_k$ can be much weakened and, for example, $R_k$ do not need to by symmetric (in this case, assumptions need to be made on the symmetrization of $R_k$, see, e.g., § 5-e).

If the regularity estimate (4.3) holds with $\alpha = 1$, then there exists a positive constant $c_1$ independent of mesh parameters such that (see Theorem 4.23)

$$\|(I - P_{k-1})v\|_A^2 \leq c_1 \lambda_k^{-1} \|A_k v\|^2 \quad \forall\, v \in \mathcal{M}_k. \tag{5.19}$$

The next technical result shows that any function smoothened by local relaxation can be well approximated by a coarser grid.

**(5.20) Lemma.**

$$\|(I - P_{k-1})K_k^m v\|_A^2 \leq \frac{c_1}{2mc_0}(\|v\|_A^2 - \|K_k^m v\|_A^2).$$

*Proof.*

$$\begin{aligned} ||(I-P_{k-1})K_k^m v||_A^2 &\le c_1\lambda_k^{-1}||A_kK_k^m v||^2 \\ &= \frac{c_1}{c_0}(R_kA_kK_k^m v, A_kK_k^m v) = \frac{c_1}{c_0}((I-K_k)K_k^{2m}v, v)_A \\ &\le \frac{c_1}{2mc_0}(||v||_A^2 - ||K_k^m v||_A^2). \end{aligned}$$

The proof is completed by using the following elementary inequality:

$$((I-K_k)K_k^{2m}v, v)_A \le \frac{1}{2m}\sum_{j=0}^{2m-1}((I-K_k)K_k^j v, v)_A \tag{5.21}$$

$$\le \frac{1}{2m}(||v||_A^2 - ||K_k^m v||_A^2).$$

□

**(5.22) Theorem.** *For the Algorithm 5.14, we have*

$$||I - B_kA_k||_A^2 \le \frac{c_1}{2mc_0 + c_1}, \quad 1 \le k \le J.$$

*Proof.* By definition of Algorithm 5.14, we have

$$I - B_kA_k = (I - P_{k-1}B_{k-1}A_{k-1})(I - R_kA_k)^m$$

and, thus, for all $v \in \mathcal{M}_k$

$$\begin{aligned} &||(I-B_kA_k)v||_A^2 \\ &= ||(I-P_{k-1})K_k^m v||_A^2 + ||(I-B_{k-1}A_{k-1})P_{k-1}K_k^m v)||_A. \end{aligned}$$

Let $\delta = c_1/(2mc_0 + c_1)$. We shall prove the above estimate by induction. First of all it is obviously true for $k = 0$. Assume it holds for $k-1$. In the case of $k$, we have from the above identity that

$$\begin{aligned} ||(I-B_kA_k)v||_A^2 &\le ||(I-P_{k-1})K_k^m v||_A^2 + \delta||P_{k-1}K_k^m v||_A^2 \\ &\le (1-\delta)||(I-P_{k-1})K_k^m v||_A^2 + \delta||K_k^m v||_A^2 \\ &\le (1-\delta)\frac{c_1}{2mc_0}(||v||_A^2 - ||K_k^m v||_A^2) + \delta||K_k^m v||_A^2 \\ &\le \delta||v||_A^2. \end{aligned}$$

□

**5-d. V-cycle and W-cycle.** Two important variants of the above backslash cycle are the so-called V-cycle and W-cycle.

A V-cycle algorithm is obtained from the backslash cycle by performing more smoothings after the coarse grid corrections. Such an algorithm, roughly speaking, is like a backslash (\) cycle plus a slash (/) (a reversed backslash) cycle. The detailed algorithm is given as follows.

**(5.23) Algorithm.** *For $k = 0$, define $B_0 = A_0^{-1}$. Assume that $B_{k-1} : \mathcal{M}_{k-1} \mapsto \mathcal{M}_{k-1}$ is defined. We shall now define $B_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ which is an iterator for the equation of the form*

$$A_k v = g.$$

1. *pre-smoothing: For* $v^0 = 0$ *and* $l = 1, 2, \cdots, m$

$$v^l = v^{l-1} + R_k(g - A_k v^{l-1})$$

2. *Coarse grid correction:* $e_{k-1} \in \mathcal{M}_{k-1}$ *is the approximate solution of the residual equation* $A_{k-1}e = Q_{k-1}(g - Av^m)$ *by the iterator* $B_{k-1}$*:*

$$e_{k-1} = B_{k-1}Q_{k-1}(g - Av^m).$$

3. *post-smoothing: For* $v^{m+1} = v^m + e_{k-1}$ *and* $l = m+2, 2, \cdots, 2m$

$$v^l = v^{l-1} + R_k(g - A_k v^{l-1})$$

*A non-recursive implementation.* The recursive formulation of the above algorithm makes it a little less straightforward to code sometimes. A non-recursive version of the algorithm is given below in terms of matrices and vectors.

**(5.24) Algorithm** (V-CYCLE COMPUTATION OF $\mathcal{B}\beta$).

$\beta_J = \beta$;
*for* $l = J : 1$,
  $\alpha_l = \mathcal{R}_l\beta_l, \beta_{l-1} = (\mathcal{I}_{l-1}^l)^t(\beta_l - \mathcal{A}_l\alpha_l)$;
*endfor*
*for* $l = 2 : J$,
  $\alpha_l = \mathcal{R}_l^t(\alpha_l + \mathcal{I}_{l-1}^l\alpha_{l-1})$;
*endfor*
$\mathcal{B}\beta = \alpha_J$.

The reason why this algorithm is called V-cycle is quite clear with the above implementation. The algorithm starts on the finest level and traverses all the grids, one at a time, until it reaches the coarsest grid. Then it traverses all the grids until it reaches the finest level.

It is easy to see that the operators $B_k$ defined by the above V-cycle algorithm satisfy

$$I - B_kA_k = (I - B_k^{(\backslash)}A_k)^*(I - B_k^{(\backslash)}A_k)$$

where $B_k^{(\backslash)}$ correspond the operator defined by the backslash cycle Algorithm 5.14 and $*$ is the adjoint operator with respect to the $A-$inner product. Consequently,

$$||I - B_kA_k||_A = ||I - B_k^{(\backslash)}A_k||_A^2.$$

This means that the convergence of the V-cycle is a consequence of the convergence of backslash cycle.

The W-cycle, roughly speaking, is like a V-cycle plus another V-cycle. In a V-cycle iteration, the coarse grid correction is only performed once, while in a W-cycle, the coarse grid correction is performed twice.

**(5.25) Algorithm.** *For* $k = 0$*, define* $B_0 = A_0^{-1}$*. Assume that* $B_{k-1} : \mathcal{M}_{k-1} \mapsto \mathcal{M}_{k-1}$ *is defined. We shall now define* $B_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ *which is an iterator for the equation of the form*

$$A_k v = g.$$

1. *pre-smoothing: For* $v^0 = 0$ *and* $l = 1, 2, \cdots, m$

$$v^l = v^{l-1} + R_k(g - A_k v^{l-1})$$

2. *Coarse grid correction:* $e_{k-1} \in \mathcal{M}_{k-1}$ *is the approximate solution of the residual equation* $A_{k-1}e = Q_{k-1}(g - Av^m)$ *by applying iterator* $B_{k-1}$ *twice,* $e_{k-1} = w^2$, *with*

$$w^j = w^{j-1} + B_{k-1}(Q_{k-1}(g - Av^m) - A_{k-1}w^{j-1}, \quad j = 0, 1, 2, \quad w^0 = 0.$$

3. *post-smoothing: For* $v^{m+1} = v^m + e_{k-1}$ *and* $l = m+2, 2, \cdots, 2m$

$$v^l = v^{l-1} + R_k(g - A_k v^{l-1}).$$

Again it is also easy to see that the convergence of the W-cycle is an easy consequence of the convergence of V-cycle. But it is often the case that W-cycle is easier to analyze. We shall now give an optimal estimate for the convergence of W-cycle based on the elliptic regularity assumption (4.3) (which implies that (4.24) holds).

**(5.26) Theorem.** *Under the elliptic regularity assumption* (4.3) *for some* $\alpha \in (0, 1]$, *the W-cycle iteration admits the following estimate*

$$\|I - B_k A_k\|_A^2 \le \frac{M}{m^\alpha + M} \tag{5.27}$$

*for some constant* $M$ *that is independent of mesh parameters.*

*Proof.* Let $\delta = \frac{M}{m^\alpha + M}$. The estimate (5.27) will be proved by induction. As there is nothing to prove for $k = 0$, we assume that (5.27) is valid for $k-1$. By definition, the following recurrence relation holds for any $v \in \mathcal{M}_k$.

$$A((I - B_k A_k)v, v) = A((I - P_{k-1})\tilde{v}, \tilde{v}) + A((I - B_{k-1}A_{k-1})P_{k-1}\tilde{v}, P_{k-1}\tilde{v})$$

where $\tilde{v} = K_k^m v$.

$$\begin{aligned}
&\|(I - B_k A_k)v\|_A^2 \\
&\quad \le \|(I - P_{k-1})\tilde{v}\|_A^2 + \delta\|P_{k-1}\tilde{v}\|_A^2 \quad \text{(by induction)} \\
&\quad \le (1 - \delta^2)\|(I - P_{k-1})\tilde{v}\|_A^2 + \delta^2\|\tilde{v}\|_A^2 \\
&\quad \le (1 - \delta^2)(c_1\lambda_k^{-1}\|A_k\tilde{v}\|^2)^\alpha \|\tilde{v}\|_A^{2(1-\alpha)} + \delta^2\|\tilde{v}\|_A^2 \quad \text{(by (4.24))} \\
&\quad \le (1 - \delta^2)\left[\frac{c_1}{2mc_0}(\|v\|_A^2 - \|\tilde{v}\|_A^2)\right]^\alpha \|\tilde{v}\|_A^{2(1-\alpha)} + \delta^2\|\tilde{v}\|_A^2. \quad \text{(by (5.21))}
\end{aligned}$$

Note that $\|\tilde{v}\|_A^2/\|v\|_A^2 \in [0, 1]$, the desired estimate then easily follows if we can prove the following elementary inequality

$$(1 - \delta^2)\left[\frac{c_1}{2mc_0}(1 - t)\right]^\alpha t^{1-\alpha} + \delta^2 \le \delta \quad t \in [0, 1]. \tag{5.28}$$

By Hölder inequality, for any $\eta > 0$, we have

$$(1 - t)^\alpha t^{1-\alpha} \le \alpha\eta(1 - t) + (1 - \alpha)\eta^{\frac{\alpha}{\alpha-1}}t.$$

Thus (5.28) is a consequence of the following inequality

$$(1 - \delta^2)\left[\frac{c_1}{2mc_0}\right]^\alpha \alpha\eta(1 - t) + ((1 - \alpha)\eta^{\frac{\alpha}{\alpha-1}} + \delta^2)t \le \delta \quad t \in [0, 1].$$

The choice of $\eta$ is made to minimize the above left hand side, name to equalize the coefficients of $1 - t$ and $t$. The proof may be completed with some more elementary manipulations by choosing sufficient large $M$ in the expression of $\delta$. □

**5-e. Subspace correction interpretation.** We shall now discuss the multigrid method from the subspace correction point of view. Let $\mathcal{M}_k$ $(k = 0, 1, \cdots, J)$ be the multilevel finite element spaces defined as in the preceding section. Again let $\mathcal{V} = \mathcal{M}_J$, but set $\mathcal{V}_k = \mathcal{M}_{J-k}$. In this case, the decomposition (3.1) is trivial.

We observe that, with the above choice of subspaces $\mathcal{V}_k$, there are redundant overlappings in the decomposition (3.1). The point is that these overlappings can be used advantageously in the choice of 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*). As we know, the methods like Gauss-Seidel method discussed earlier just satisfies such requirements.

With the above ingredients, the successive subspace correction method Algorithm (3.12) can be stated as follows.

**(5.29) Algorithm.** *Given* $u^0 \in \mathcal{V}$.
*for* $k = 0, 1, \ldots$ *till convergence*
$v \leftarrow u^k$
*for* $i = J : -1 : 0 \quad v \leftarrow v + R_i Q_i (f - Av)$ *endfor*
$u^{k+1} \leftarrow v$.
*endfor*

**(5.30) Lemma.** *For the Algorithm 5.14 with* $m = 1$, *we have*

$$I - B_J A_J = (I - T_0)(I - T_1) \cdots (I - T_J) \tag{5.31}$$

*where*

$$T_0 = P_0, T_k = R_k A_k P_k, \quad 1 \le k \le J.$$

*Hence, with such defined operators* $B_k$, *the iteration* (5.15) *is mathematically equivalent to Algorithm 5.29.*

*Furthermore if* $B_J^m$ *is obtained from Algorithm 5.14 for general* $m \ge 1$, *then*

$$||I - B_J^m A_J||_A \le ||I - B_J A_J||_A.$$

Based on the above lemma, different proofs may be obtained on the convergence of the backslash cycle multigrid method. In particular, the framework given in § 3 can be applied. This new analysis does not depend crucially on the elliptic regularity and hence can be applied more easily to more complicated situations such as the problems with large discontinuous jump coefficients and locally refined meshes (see § $(5 - i)$) .

We now consider the more general case in which we do not assume any elliptic regularity for the underlying partial differential equations. (4.1).

We assume that all the smoothers $R_k$ satisfy

$$(R_k v, v) \le \omega_1 (A_k^{-1} v, v) \quad \forall\, v \in \mathcal{M}_k. \tag{5.32}$$

and, if $R_k$ are all symmetric

$$\frac{c_0}{\lambda_k}(v, v) \le (R_k v, v) \le \frac{c_1}{\lambda_k}(v, v) \quad \forall\, v \in \mathcal{M}_k. \tag{5.33}$$

or, in general

$$\frac{c_0}{\lambda_k}(v, v) \le (\bar{R}_k v, v) \le \frac{c_1}{\lambda_k}(v, v) \quad \forall\, v \in \mathcal{M}_k. \tag{5.34}$$

where $\bar{R}_k$ is the symmetrization of $R_k$ (see § 2-a).

By Lemma 5.13, the Gauss-Seidel methods satisfy the above assumptions.

The idea is to use the general framework of § 3. According to the theory there, we need to estimate two basic parameters, namely $\bar{K}_0$ and $\bar{K}_1$.

By Theorem 4.30 and (5.33), there exists a constant $C_0$ independent of mesh parameters such that

$$K_0 \leq C_0. \tag{5.35}$$

**(5.36) Lemma.** *The $\bar{\epsilon}_{ij}$ defined in* (3.27) *satisfy, for some $\gamma \in (0,1)$ independent of mesh parameters,*

$$\bar{\epsilon}_{ij} \lesssim \gamma^{|i-j|/2}. \tag{5.37}$$

*Proof.* Let $i \geq j$. It follows from Lemma 4.26 and (5.33) that

$$(\bar{T}_i v_j, v_j)_A \lesssim \gamma^{i-j} h_i^{-1} ||v_j||_A ||\bar{T}_i v_j|| \lesssim \gamma^{i-j} (v_j, v_j)_A \quad \forall\, v_j \in \mathcal{V}_j.$$

The desired estimate then follows from the definition of $\bar{\epsilon}_{ij}$. □

By definition, we conclude from above lemma that there exists a constant $C_1$ independent of mesh parameters such that

$$\bar{K}_1 \leq C_1. \tag{5.38}$$

With the above results, the following result follows directly from Theorem 3.46.

**(5.39) Theorem.** *Assume that the smoothers $R_k$ satisfy* (5.33) *and* (5.32) *with $\omega_1 < 2$, then the backslash cycle Algorithm 5.14 or Algorithm* (5.29) *satisfy*

$$||I - B_J A_J||_A^2 \leq 1 - \frac{2 - \omega_1}{C}$$

*for some positive constant $C$ independent of mesh parameters.*

*Another convergence analysis using full elliptic regularity.* If the full elliptic regularity is valid, however, a more straightforward proof can also be obtained in this new framework. We shall first present such a proof.

**(5.40) Theorem.** *For the iteration* (5.15) *with $B_J$ given by Algorithm 5.14 with one smoothing on each level, then*

$$||I - B_J A_J||_A^2 \leq 1 - \frac{c_0}{c_1}.$$

*Proof.* Denote $E_{-1} = I$ and for $0 \leq i \leq J$,

$$E_i = (I - T_i)(I - T_{i-1}) \cdots (I - T_1)(I - T_0).$$

Note that $E_J = (I - B_J A_J)^*$. It follows that, denoting $\tilde{P}_i = P_i - P_{i-1}$,

$$
\begin{aligned}
||v||_A^2 &= \sum_{i=0}^{J}(\tilde{P}_i v, v)_A \\
&= \sum_{i=0}^{J}(\tilde{P}_i v, E_{i-1}v)_A \quad (\text{since } (I-E_{i-1})v \in \mathcal{M}_{i-1}) \\
&= \sum_{i=0}^{J}(\tilde{P}_i v, A_i P_i E_{i-1}v) \\
&\leq \sqrt{c_1}\sum_{i=0}^{J}\lambda_i^{-1/2}||\tilde{P}_i v||_A ||A_i P_i E_{i-1}v|| \quad (\text{by (5.19)}) \\
&\leq \sqrt{\frac{c_1}{c_0}}\sum_{i=0}^{J}||\tilde{P}_i v||_A (R_i A_i P_i E_{i-1}v, A_i P_i E_{i-1}v)^{1/2} \quad (\text{by (5.18)}) \\
&\leq \sqrt{\frac{c_1}{c_0}}\sum_{i=0}^{J}||\tilde{P}_i v||_A (T_i E_{i-1}v, E_{i-1}v)_A^{1/2} \\
&\leq \sqrt{\frac{c_1}{c_0}}\Big(\sum_{i=0}^{J}||\tilde{P}_i v||_A^2\Big)^{1/2}\Big(\sum_{i=0}^{J}(T_i E_{i-1}v, E_{i-1}v)_A\Big)^{1/2} \\
&\leq \sqrt{\frac{c_1}{c_0}}\Big(\sum_{i=0}^{J}||\tilde{P}_i v||_A^2\Big)^{1/2}\Big(\sum_{i=0}^{J}(T_i E_{i-1}v, E_{i-1}v)_A\Big)^{1/2} \\
&\leq \sqrt{\frac{c_1}{c_0}}||v||_A\Big(||v||_A^2 - ||E_J v||_A^2\Big)^{1/2}.
\end{aligned}
$$

Consequently,

$$||E_J v||_A^2 \leq (1-\frac{c_0}{c_1})||v||_A^2 \quad \forall\, v \in \mathcal{V}.$$

The desired estimate then follows easily. □

**5-f. Full multigrid cycle.** We shall now describe a more efficient multigrid technique, called full multigrid cycle, originally proposed by Brandt.

On each level of finite element space $\mathcal{V}_k$, there corresponds to a finite element approximation $u_k \in \mathcal{V}_k$ such that

$$a(u^{(k)}, v) = (F, v) \quad \forall v \in \mathcal{V}. \tag{5.41}$$

Similar to (4.11), the best error estimate in $H^1$ norm is

$$||U - u^{(k)}||_1 = O(h_k^{\alpha}). \tag{5.42}$$

If $\mu^{(k)} \in \mathbb{R}^{n_k}$ is the nodal value vector of $u^{(k)}$, then

$$A_k \mu^{(k)} = b^{(k)}. \tag{5.43}$$

where $b^{(k)} = ((F, \phi_i^k))$. It can be proved that, with $\mathcal{I}_k^{k+1}$ given by (5.16)

$$b^{(k)} = (\mathcal{I}_k^{k+1})^t b^{(k+1)} \quad \text{with } b^{(J)} = b.$$

The full multigrid method is based on the following two observations:

1. $u^{(k-1)} \in \mathcal{V}_{k-1} \subset \mathcal{V}_k$ is close to $u^{(k)} \in \mathcal{V}_k$ and hence can be used as an initial guess for an iterative scheme for solving $u^{(k)}$.
2. Each $u^{(k)}$ can be solved within its truncation error shown in (5.42) by a multigrid iterative scheme.

**(5.44) Algorithm.** $\hat{\mu}^{(1)} \leftarrow \mathcal{A}_1^{-1} b^{(1)}$. *For* $k = 2 : J$

1. $\hat{\mu}^{(k)} \leftarrow \mathcal{I}_{k-1}^k \hat{\mu}^{(k-1)}$,
2. *Iterate* $\hat{\mu}^{(k)} \leftarrow \hat{\mu}^{(k)} + \mathcal{B}_k(b^{(k)} - \mathcal{A}_k \hat{\mu}^{(k)})$ *for* $m$ *times.*

The most important fact on the full multigrid method is that it has an optimal computational complexity $O(N)$ to compute the solution within truncation error.

**(5.45) Proposition.** *Assume that that* $C_0$ *is a positive constant satisfying (for all* $k$*)*

$$||\mu^{(k)} - \mu^{(k-1)}||_{\mathcal{A}} \leq C_0 h_k^{\alpha}.$$

*Then*

$$||\mu^{(k)} - \hat{\mu}^k||_{\mathcal{A}} \leq h_k^{\alpha}$$

*if*

$$m \geq \frac{\log(2^{\alpha} + C_0)}{|\log \delta|}.$$

*Proof.* By definition, $||\mu^{(1)} - \hat{\mu}^1||_{\mathcal{A}} = 0$. Now assume that

$$||\mu^{(k-1)} - \hat{\mu}^{k-1}||_{\mathcal{A}} \leq h_{k-1}^{\alpha}.$$

Then

$$\begin{aligned} ||\mu^{(k)} - \hat{\mu}^k||_{\mathcal{A}} &\leq \delta^m ||\mu^{(k)} - \hat{\mu}^{k-1}||_{\mathcal{A}} \\ &\leq \delta^m ||\mu^{(k)} - \mu^{k-1}||_{\mathcal{A}} + \delta^m ||\mu^{(k-1)} - \hat{\mu}^{k-1}||_{\mathcal{A}} \\ &\leq \delta^m (C_0 + 2^{\alpha}) h_k^{\alpha} \\ &\leq h_k^{\alpha} \end{aligned}$$

□

**5-g. BPX multigrid preconditioners.** We shall now describe a parallelized version of the multigrid method studied earlier. This method was first proposed in Bramble-Pasciak-Xu [9] and Xu [31], and is now often known as the BPX preconditioner in the literature.

**Basic algorithm and theory.** There are different ways of deriving the BPX preconditioners. The method was originally resulted from an attempt to parallelizing the classic multigrid method. With the current multigrid theoretical technology, the derivation of this method is not so difficult. We shall first derive this preconditioner based on Theorem 4.30 and then, in the next subsection, study the method using the framework of subspace correction.

By Theorem 4.30, we have

$$(Av, v) \eqsim \sum_{k=0}^{J} h_k^{-2} ||(Q_k - Q_{k-1})v||^2 = (\hat{A}v, v) \quad v \in \mathcal{V}.$$

where

$$\hat{A} = \sum_k h_k^{-2}(Q_k - Q_{k-1}).$$

Using Lemma (4.20), we can show that

$$\hat{A}^{-1} = \sum_k h_k^2(Q_k - Q_{k-1}).$$

Using the fact that $h_k = 2h_{k+1}$, we deduce that

$$\begin{aligned}(\hat{A}^{-1}v, v) &= \sum_{k=0}^{J} h_k^2((Q_k - Q_{k-1})v, v) = \sum_{k=0}^{J} h_k^2(Q_k v, v) - \sum_{k=0}^{J-1} h_{k+1}^2(Q_k v, v) \\ &\eqsim h_J^2(v, v) + \sum_{k=0}^{J-1} h_k^2(Q_k v, v) \eqsim \sum_{k=0}^{J} h_k^2(Q_k v, v) = (\tilde{B}v, v)\end{aligned}$$

where

$$\tilde{B} = \sum_{k=0}^{J} h_k^2 Q_k.$$

If $R_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ is an SPD operator satisfying

$$(R_k v_k, v_k) \eqsim h_k^2(v_k, v_k) \quad \forall\, v_k \in \mathcal{M}_k \tag{5.46}$$

then, for

$$B = \sum_{k=0}^{J} R_k Q_k \tag{5.47}$$

we have

$$(Bv, v) \eqsim (\tilde{B}v, v) \eqsim (A^{-1}v, v)$$

namely

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

**(5.48) Theorem.** *Assume that $R_k$'s satisfy* (5.46)*; then the preconditioner* (3.7) *satisfies*

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

We note that all the relaxation methods mentioned earlier, such as Richardson, Jacobi and symmetric Gauss-Seidel satisfy (5.46).

**Subspace correction approach.** In § 5-e, the slash cycle multigrid method is interpreted as a successive subspace correction method. Correspondingly, the BPX preconditioner can be interpreted as the relevant PSC (parallel subspace correction) preconditioner. It is possible to use the abstract theory in § 3 to derive some estimates for the BPX preconditioner somewhat more refined than that in Theorem 5.48, we leave the details to the interested readers. In § 5-i, we shall use this approach to analyze the BPX preconditioner for locally refined meshes.

*Implementation.* Again, in view of (3.54), the algebraic representation of the preconditioner given by (3.54) is

$$\mathcal{B} = \sum_{k=0}^{J} \mathcal{I}_k \mathcal{R}_k \mathcal{I}_k^t, \tag{5.49}$$

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

Let $\mathcal{I}_k^{k+1} \in \mathbb{R}^{n_{k+1} \times n_k}$ be as defined in (5.16), then

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

This identity is very useful for the efficient implementation of (5.49) on both serial and parallel fashions.

If $\mathcal{R}_k$ are given by the Richardson iteration, we have

$$\mathcal{B} = \sum_{k=0}^{J} h_k^{2-d} \mathcal{I}_k \mathcal{I}_k^t. \tag{5.50}$$

From (5.49) or (5.50), we see that the preconditioner depends entirely on the transformation between the nodal bases on multilevel spaces.

Let, for $1 \leq l \leq J$,

$$\mathcal{B}_l = \sum_{k=0}^{l} \mathcal{I}_k^l \mathcal{R}_k (\mathcal{I}_k^l)^t.$$

By definition $\mathcal{B} = \mathcal{B}_J$ and

$$\mathcal{B}_l = \mathcal{R}_l + \mathcal{I}_{l-1}^l \mathcal{B}_{l-1} (\mathcal{I}_{l-1}^l)^t.$$

We shall use the above recurrence relation to compute the action of $\mathcal{B}$. Assume that $m_l$ is the number of operations that are needed to compute the action $\mathcal{B}_l \alpha_l$ for $\alpha_l \in \mathbb{R}^{n_l}$. By the identity

$$\mathcal{B}_l \alpha_l = \mathcal{R}_l \alpha_l + \mathcal{I}_{l-1}^l [B_{l-1} (\mathcal{I}_{l-1}^l)^t \alpha_l]$$

we get, for some constant $c_0 > 0$,

$$m_l \leq m_{l-1} + c_0 n_l$$

from which we conclude that

$$m_J \leq m_1 + c_0 \sum_{l=2}^{J} n_l \leq c_1 n$$

for some positive constant $c_1$. This means that the action of $B_J$ can be carried out within $O(n)$ operations.

**(5.51) Algorithm** (COMPUTATION OF $\mathcal{B}\alpha$).
$\alpha_J = \alpha$;
***for*** $l = J : 1$,
  $\alpha_{l-1} = (\mathcal{I}_{l-1}^l)^t \alpha_l$;
***end***

$\beta_0 = \mathcal{R}_0 \alpha_0$;
*for* $l = 1 : J$,
  $\beta_l = \mathcal{R}_l \alpha_l + \mathcal{I}_{l-1}^l \beta_{l-1}$;
*end*
$\mathcal{B}\alpha = \beta_J$.

As it is discussed above, the number of operations needed in the above algorithm is $O(n)$. We also note that all the vectors $\alpha_l$ for $1 \leq k \leq J$ need to be stored, but the whole storage space for these vectors is also only $O(n)$.

**5-h. Hierarchical basis methods.** Assume that we are given a nested sequence of multigrid subspace of $H_0^1(\Omega)$

$$\mathcal{M}_1 \subset \mathcal{M}_2 \subset \ldots \subset \mathcal{M}_k \subset \ldots \subset \mathcal{M}_J.$$

as described in §sc:nestedMG. The so-called *hierarchical basis* refers to the special set of nodal basis functions

$$\{\phi_i^k : x_i^k \in \mathcal{N}_k \setminus \mathcal{N}_{k-1}, k = 0, \cdots, J\}. \tag{5.52}$$

It is easy to see that this set of functions does form a basis of $\mathcal{M}$. For $d \neq 2$, it is often more convenient to use the scaled HB as follows

$$\{h_k^{2-d} \phi_i^k : x_i^k \in \mathcal{N}_k \setminus \mathcal{N}_{k-1}, k = 0, \cdots, J\}. \tag{5.53}$$

With a proper ordering, we shall denote the scaled HB by $\{\psi_i, i = 1 : N\}$.

The HB in multiple dimensions is formally a direct generalization of the HB in one dimensional case. But the property for the corresponding stiffness matrix in multiple dimensions is not all as clear as in one dimension where the stiffness matrix is an identity matrix in some special case. In this section, we shall show that at least in two dimensions, hierarchical basis is still very useful.

The hierarchical basis in two dimensions was first analyzed by Yserentant in his pioneering paper [36]. The work of Yserentant was apparently motivated by the famous unpublished technical report of Bank and Dupont [1]. Incidently these three authors got together wrote another important paper Bank-Dupont-Yserentant [3] on a Gauss-Seidel (or multiplicative) variant of the hierarchical basis method. The presentation of the materials in this section is of course mostly based on the aforementioned papers, and moreover it also adopts the view of subspace correction from Xu [32] (see also Xu [31].

**Preliminaries.** We shall now discuss multigrid subspaces that are directly related to the HB. Consider the part of the HB functions on level $k$ as follows

$$\{\phi_i^k : x_i^k \in \mathcal{N}_k \setminus \mathcal{N}_{k-1}\}. \tag{5.54}$$

It is easy to see that the above set of functions spans to the following subspace

$$\mathcal{V}_k = (I_k - I_{k-1})\mathcal{M} = (I - I_{k-1})\mathcal{M}_k, \quad \text{for } k = 0 : J. \tag{5.55}$$

Here, we recall, $I_{-1} = 0$ and $I_k : \mathcal{M} \mapsto \mathcal{M}_k$ is the nodal value interpolant. The above subspaces obviously give rise to a direct sum decomposition of the space $\mathcal{M}$ as follows

$$\mathcal{V} = \oplus_{k=0}^J \mathcal{V}_k.$$

In fact, for any $v \in \mathcal{V}$, we have the following unique decomposition

$$v = \sum_{k=0}^{J} v_k \quad \text{with} \quad v_k = (I_k - I_{k-1})v.$$

With the subspaces $\mathcal{V}_k$ given by (5.55), the operators $A_k$ are all well conditioned. In fact, by (4.22) and (4.5), we can see that

$$A(v,v) \eqsim h_k^{-2}\|v\| \quad \forall\, v \in \mathcal{V}_k.$$

As a result, the subspace equations can be effectively solved by elementary iterative methods such as Richardson, Jacobi and Gauss-Seidel methods.

**Stiffness matrix in terms of hierarchical basis.** The easiest way of understanding the HB is perhaps, like in one dimension, through the study of the property of the corresponding stiffness matrix. As one may expect, the condition number of the HB stiffness matrix should be smaller than the NB stiffness matrix. This is indeed the case in two and three dimensions.

**(5.56) Theorem.** *Assume that $\hat{\mathcal{A}}$ is the stiffness matrix under the scaled hierarchical basis, then*

$$\kappa(\hat{\mathcal{A}}) \lesssim \kappa_d(h) \tag{5.57}$$

*where*

$$\kappa_d(h) \lesssim \begin{cases} 1 & \text{if } d = 1; \\ |\log h|^2 & \text{if } d = 2; \\ h^{2-d} & \text{if } d \geq 3. \end{cases} \tag{5.58}$$

In fact, the estimates given in the above theorem can be proven to be sharp. The most interesting case is obviously $d = 2$ for which $\kappa(\hat{\mathcal{A}}) \lesssim |\log h|^2$. Compared with the conditioning of the stiffness matrix under the NB, this is a great improvement. It is also in the case $d = 2$ that the HB is most useful. Indeed for $d = 3$, the $\kappa(\hat{\mathcal{A}}) = O(h^{-1})$ is also one magnitude smaller than the condition number of the NB stiffness matrix, but such an improvement is not attractive as we shall see that a much better approach (such as BPX preconditioner) is available. There is no doubt that, as far as $\kappa(\hat{\mathcal{A}})$ is concerned, the HB is of no use for $d \geq 4$.

*Proof.* Given $\alpha \in \mathbb{R}^N$, set $v = \sum_{i=1}^{N} \alpha_i \psi_i$. We can write

$$v = \sum_k v_k \quad \text{with} \quad v_k = (I_k - I_{k-1})v = \sum_{x_i^k \in \mathcal{N}_k \setminus \mathcal{N}_{k-1}} \alpha_i^k \phi_i^k.$$

It follows that

$$|v_k|_1^2 \eqsim \sum_{x_i^k \in \mathcal{N}_k \setminus \mathcal{N}_{k-1}} v_k^2(x_i^k) = \sum_{x_i^k \in \mathcal{N}_k \setminus \mathcal{N}_{k-1}} (\alpha_i^k)^2 = |\alpha|^2.$$

Thus

$$\begin{aligned} \alpha^t \hat{\mathcal{A}} \alpha &= a(v,v) = \sum_{kl} a(v_k, v_l) \\ &\lesssim \sum_{kl} \gamma^{|k-l|} |v_k|_1 |v_l|_1 \quad \text{(by Lemma 4.27)} \\ &\lesssim \sum_k |v_k|_1^2 \eqsim |\alpha|^2. \end{aligned}$$

This implies that $\lambda_{\max}(\hat{\mathcal{A}}) \lesssim 1$. On the other hand

$$\begin{aligned} |\alpha|^2 &\eqsim \sum_k |v_k|_1^2 = \sum_k |(I_k - I_{k-1})v|_1^2 \\ &\lesssim \sum_k (J-k+1)|v|_1^2 \quad \text{(by (4.22) )} \\ &\lesssim \kappa_d(h)|v|_1^2 \lesssim \kappa_d(h)\alpha^t \hat{\mathcal{A}}\alpha \end{aligned}$$

This proves that $\lambda_{\min}(\hat{\mathcal{A}}) \gtrsim \kappa_d(h)^{-1}$. □

The above proof is essentially the same as in Yserentant [36] (and see also Ong [29] for $d = 3$).

**Subspace correction approach and general case.** Following Xu [32], we shall now study the HB method from the viewpoint of space decomposition and subspace correction.

In view of (3.54), the algebraic representation of the PSC preconditioner is

$$\mathcal{H} = \sum_{k=0}^{J} \mathcal{S}_k \mathcal{R}_k \mathcal{S}_k^t, \tag{5.59}$$

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 $\mathcal{M}$.

*A special case.* If $\mathcal{R}_k$ is given by the Richardson iteration: $\mathcal{R}_k = h_k^{2-d}\mathcal{I}$, we have

$$\mathcal{H} = \sum_{k=0}^{J} h_k^{2-d} \mathcal{S}_k \mathcal{S}_k^t = \hat{S}\hat{S}^t.$$

where

$$\mathcal{S} = (h_1^{1-d/2}\mathcal{S}_1, h_2^{1-d/2}\mathcal{S}_2, \cdots, h_J^{1-d/2}\mathcal{S}_J)$$

is the representation matrix of the HB in terms of NB. Obviously the HB stiffness matrix and NB stiffness matrix are related by $\hat{\mathcal{A}} = \mathcal{S}^t\mathcal{A}\mathcal{S}$. Therefore,

$$\kappa(\mathcal{H}\mathcal{A}) = \kappa(\hat{\mathcal{A}}),$$

and, as a result of Theorem 5.56,

$$\kappa(\mathcal{H}\mathcal{A}) \eqsim \kappa_d(h). \tag{5.60}$$

The above estimate apparently also holds for the more general $\mathcal{H}$ when $\mathcal{R}_k$ is given by either Jacobi or symmetric Gauss-Seidel since either of this iteration satisfies the following spectrally equivalence

$$\mathcal{R}_k \eqsim h_k^{2-d}. \tag{5.61}$$

In fact, the estimate (5.60) also follows easily from the general theory for the PSC preconditioner. To see this,

For the SSC iterative method, it is more convenient to choose $\mathcal{R}_k$ to be the symmetric Gauss-Seidel as the other two methods need to be properly scaled to assume that $\omega_1 \in (0, 2)$. The resulting algorithm is

**(5.62) Algorithm.** *$\mu^0 \in \mathbb{R}^n$ is given. Assume that $\mu^k \in \mathbb{R}^n$ is obtained. Then $\mu^{l+1}$ is defined by*

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

*for $k = 0 : J$.*

It can be proved late that

$$||\mu - \mu_l||_{\mathcal{A}} \leq \left(1 - \frac{c}{\kappa_d(h)}\right)^l ||\mu - \mu_0||_{\mathcal{A}}. \tag{5.63}$$

**Convergence analysis. (5.64) Lemma.** *We assume that $R_k$ is either Richardson, or Jacobi or symmetric Gauss-Seidel iteration, then*

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

*where $\omega_1$ is a constant and for symmetric Gauss-Seidel, $\omega_1 = 1$.*

*Proof.* The proof for Richardson or Jacobi method is straightforward. The proof for the symmetric Gauss-Seidel method is almost identical to that of Lemma 5.12 and the detail is left to the reader. □

**(5.65) Lemma.**

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

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

*Proof.* For $v \in \mathcal{V}$, it follows from (4.22) that

$$\sum_{k=0}^{J} h_k^{-2} ||v_k||^2 \lesssim c_d ||v||_A^2.$$

This gives the estimate of $K_0$. The estimate of $K_1$ follows from Lemma 4.27. □

By using Theorem 3.37 and Lemma 3.26, we obtain

**(5.66) Theorem.** *Assume that $R_k$ satisfies* (5.64)*; then the PSC preconditioner* (3.7)*, with $\mathcal{V}_k$ given by* (5.55)*, satisfies*

For the SSC iterative method, we apply Theorem 3.46 with Lemma 5.65 and get

**(5.67) Theorem.** *The Algorithm 3.12 with the subspaces $\mathcal{V}_k$ given by* (5.55) *satisfies*

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

*provided that $R_k$'s satisfy* (5.64) *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 [3] can be viewed as such an algorithm with $R_k$ given by an appropriate Gauss-Seidel iterations. Numerical examples in [3] show that the SSC algorithm converges much faster than the corresponding SSC algorithm.

**Relation with BPX preconditioners.** Observing that $S_k$ in (5.59) is a submatrix of $\mathcal{I}_k$ given in (5.49), we then have

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

In view of the above inequality, if we take

$$\hat{\mathcal{H}} = \sum_{k=0}^{J-1} h_k^{2-d} \mathcal{S}_k \mathcal{S}_k^t + I,$$

we obtain

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

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 Xu and Qin [35] for the numerical results.

**5-i. Locally refined grids.** In practical computations, finite element grids are often locally refined (by using some error estimators or other adaptive strategies). In this subsection, we shall describe optimal multigrid procedures for adaptive grids. Our presentation here is based on [9], [7] and [6].

With appropriate rearrangement and grouping, we may assume that the mesh refinement can be done in the following fashion. We first start with the original domain $\Omega$ which is also denoted by $\Omega_0$. We introduce a relatively coarse and quasiuniform triangulation of $\Omega_0$ with a mesh size $h_0$ and denote the corresponding finite element space by $\mathcal{V}_0 \subset H_0^1(\Omega_0)$. Let $\Omega_1$ be a subregion where we wish to increase the resolution and we do so by subdividing the elements of the first triangulation and get a new triangulation of $\Omega_1$ with mesh size $h_1$ in $\Omega_1$ and introduce an additional finite element space $\mathcal{V}_1 \subset H_0^1(\Omega_1)$. We repeat this process and finally get a collection of subdomains $\Omega_i$ together with the corresponding finite element spaces $\mathcal{V}_i$ defining on a triangulation of mesh size $h_i$ for $i = 1, 2, \cdots, J$ for some integer $J > 1$. Throughout, we have

$$\Omega_i \subset \Omega_{i-1}, \qquad \mathcal{V}_{i-1} \cap H_0^1(\Omega_i) \subset \mathcal{V}_i \subset H_0^1(\Omega_i), \quad i = 1, 2, \cdots, J.$$

The finite element space on the repeatedly refined mesh can be written as

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

The only restrictions on the mesh domains $\{\Omega_k\}$ are that $\partial\Omega_k$ for $k \geq 1$ consists of edges of mesh triangles in the mesh $\mathcal{T}_{k-1}$ and that there is at least one edge from $\mathcal{T}_{k-1}$ contained in $\Omega_k$.

Let $A : \mathcal{V} \mapsto \mathcal{V}$ be as defined by (4.16). Operators $A_i : \mathcal{V}_i \mapsto \mathcal{V}_i$ and $Q_i, P_i : \mathcal{V} \mapsto \mathcal{V}_i$ can be defined similarly as before. If we choose $R_i : \mathcal{V}_i \mapsto \mathcal{V}_i$ to be some appropriate approximate solvers of $A_i$'s, we then have all the ingredients to define our PSC and SSC algorithms. In this setting, the coarse space $\mathcal{M}_0$ may not be very coarse, therefore we assume that, for the PSC type of algorithm, the first subspace solver $R_0$ is SPD and satisfies

$$(R_0^{-1} v, v) \eqsim (A_0 v, v) \quad \forall\, v \in \mathcal{M}_0, \tag{5.68}$$

and for the SSC type of algorithm, we assume that $R_0$ satisfies

$$||(I - R_0 A_0)||_A \leq \delta_0 \tag{5.69}$$

for some $\delta_1 \in (0,1)$ independent of mesh parameters.

As for the other subspace solvers $R_k$ for $k > 1$, for clarity, we assume that $R_k$ is given by a Gauss-Seidel iteration or properly damped Jacobi iteration. But apparently other reasonable solvers can also be adopted.

We would like to remark that the corresponding PSC and SSC methods in this setting can be viewed as a "nested" multigrid method associated with the multilevel spaces given by

$$\mathcal{M}_k = \sum_{i=0}^{k} \mathcal{V}_i, \quad 0 \leq k \leq J$$

but with special coarse space solvers $R_k$ only defined on the subspace $\mathcal{V}_k$, namely the smoothings are only carried in the refined regions.

To analyze the corresponding PSC and SSC methods, we introduce, for each $k$, an auxiliary finite element space $\hat{\mathcal{M}}_k$ which is defined on a quasi-uniform triangulation with mesh size $h_k$ and satisfies $\mathcal{M}_k \subset \hat{\mathcal{M}}_k$ and $\hat{\mathcal{M}}_0 \subset \cdots \subset \hat{\mathcal{M}}_J$. It is easy to see that $\hat{\mathcal{M}}_k$ can be well defined. Corresponding to the space $\hat{\mathcal{M}}_k$, let $\hat{Q}_k : \mathcal{V} \mapsto \hat{\mathcal{M}}_k$ be the $L^2$ projection.

The following result (from [7]) plays a crucial role in our analysis for the algorithms discussed in this section.

**(5.70) Lemma.** *Assume that $h_{k-1}/h_k \leq C$. There exists a sequence of linear operators $\Pi_k : \mathcal{V} \mapsto \mathcal{M}_k$ for $k = 0, 1, 2, \cdots, J$ with $\Pi_J = I$ such that, for any $v \in \mathcal{V}$, $(\Pi_k - \Pi_{k-1})v \in \mathcal{V}_k$ and*

$$||(I - \Pi_k)v|| \lesssim ||(I - \hat{Q}_k)v|| \tag{5.71}$$

*and*

$$||\Pi_k v||_A \lesssim ||v||_A. \tag{5.72}$$

*Proof.* The linear operator $\Pi_k$ is then defined, for $v \in \mathcal{V}$, by $\Pi_k v = w$, where $w$ is the unique function in $\mathcal{M}_k$ satisfying

$$w = \begin{cases} \hat{Q}_k v & \text{at the nodes of } \mathcal{M}_k \text{ in the interior of } \Omega_{k+1}, \\ v & \text{at the remaining nodes of } \mathcal{M}_k. \end{cases}$$

By this definition, it is clear that $(\Pi_k - \Pi_{k-1})v \in \mathcal{V}_k$. To establish (5.72), we first note that

$$||\hat{Q}_k v - w||^2 \leq C h_k^2 \Sigma'(\hat{Q}_k v(x_i^k) - v(x_i^k))^2 \leq C||(I - \hat{Q}_k)v||^2,$$

where the sum $\sum'$ is taken over the nodes $x_i^k$ of $\mathcal{M}_k$ on $\partial\Omega_{k+1}$. Combining the above estimate with (4.12) yields

$$||(I - \Pi_k)v|| \leq ||(I - \hat{Q}_k)v|| + ||\hat{Q}_k v - w|| \leq ||(I - \hat{Q}_k)v||$$

This proves part of the estimate (5.71). The rest of (5.72) can be estimated similarly by using $||\Pi_k v||_A \leq ||(I-\hat{Q}_k)v||_A + ||v-w||_A$. This completes the proof. □

**(5.73) Lemma.** *Let $\Pi_k : \mathcal{V} \mapsto \mathcal{M}_k$ be the operator from in the previous lemma and $\Pi_{-1} = 0$, then, there exists a constant $c_0$ independent of mesh parameters such that*

$$\sum_{k=0}^{J} \lambda_k ||(\Pi_k - \Pi_{k-1})v||_A^2 \leq c_0 ||v||_A^2 \quad \forall\, v \in \mathcal{V}. \tag{5.74}$$

*Proof.* By Lemma 5.70, we have, for $k \geq 1$,

$$\begin{aligned} ||(\Pi_k - \Pi_{k-1})v||_A^2 &\leq 2(||(I-\Pi_k)v||_A^2 + ||(I-\Pi_{k-1})v||_A^2) \\ &\leq 2(||(I-\hat{Q}_{k-1})v||_A^2 + ||(I-\hat{Q}_{k-1})v||_A^2) \end{aligned}$$

Thus, combining (5.72),

$$\sum_{k=0}^{J} \lambda_k ||(\Pi_k - \Pi_{k-1})v||_A^2 \lesssim ||v||_A^2 + \sum_{k=0}^{J} \lambda_k ||(I-\hat{Q}_k)v||_A^2.$$

The desired estimate then follows by using Theorem 4.31. □

**PSC version.** Let us first consider the preconditioner corresponding to the PSC algorithm:

$$B = \sum_{k=0}^{J} R_k Q_k. \tag{5.75}$$

Thanks to Lemma 5.73, similar to the quasiuniform case, we can use our general framework in § 3 to obtain the following theorem.

**(5.76) Theorem.** *If $R_0$ satisfies* (5.68) *and $R_k$ are given by Jacobi or symmetric Gauss-Seidel, then the PSC preconditioner* (5.75) *yields a uniformly bounded condition number*

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

As a special example, we may choose $R_k$ as follows:

$$R_k v = h_k^{2-d} \sum_{x_i^k \in \mathcal{N}_k} (v, \phi_i^k)\phi_i^k \tag{5.77}$$

as we know this corresponds to Richardson iteration which is equivalent to Jacobi iteration. For such a choice, we obtain that

$$Bv = R_0 v + \sum_{k=1}^{J} h_k^{2-d} \sum_{x_i^k \in \mathcal{N}_k} (v, \phi_i^k)\phi_i^k. \tag{5.78}$$

We notice that (5.78) is exactly the preconditioner given in [9] for locally refined meshes.

The hierarchical basis type algorithms for these composite grids can be obtained by the decomposition with $\mathcal{V}_i = (I_i - I_{i-1})\mathcal{V}$ (here $I_i : \mathcal{V} \mapsto \mathcal{M}_i$ is the nodal value interpolation operator). It is easy to see that the SSC type preconditioner is

$$Hv = R_0 v + \sum_{k=1}^{J} h_k^{2-d} \sum_{x_i^k \in \mathcal{N}_k \setminus \mathcal{N}_{k-1}} (v, \phi_i^k)\phi_i^k. \tag{5.79}$$

This preconditioner is equivalent to what is given in [36] for refined meshes. We further point out the corresponding algorithm in [3] for the refined meshes is the SSC algorithm by choosing $R_k$ to be some appropriate Gauss-Seidel iteration.

**SSC version.** The SSC version correspond to multigrid algorithms with smoothings done in the refined regions. Similarly we have the following convergence theorem.

**(5.80) Theorem.** *If $R_0$ satisfies* (5.69) *and $R_k$ are given by properly damped Jacobi or Gauss-Seidel, then the corresponding SSC iteration yields a uniform contraction:*

$$||I - BA||_A \leq \delta$$

*for some $\delta \in [\delta_0, 1)$ independent of mesh parameters.*

*Additional bibliographic comments.* The multilevel algorithm for finite element or finite difference equations was first developed in early sixties by the Russian mathematician Fedorenko [20]. In the early seventies, Brandt [12] brought this method to the attention of western countries and extensive research have been done on this method since then. It has become one of the most popular and also most powerful iterative methods in nowadays.

Multilevel method for composite grids can be traced back to Brandt [12] or composite grid in McCormick [25] (see also the references therein). The finite element space on so refined mesh is $\mathcal{V} = \sum_{i=0}^{J} \mathcal{V}_i$. PSC and SSC methods can be naturally obtained with Gauss-Seidel iteration as subspace solvers. The SSC iteration corresponds to a multigrid method with smoothings carried out only on the refined region discussed in Brandt [12]. The PSC preconditioner was first considered in Bramble-Pasciak-Xu [9].

The aforementioned refined grids may not give the minimal degree of freedoms from approximation point of view, but it is a computationally efficient approach that has the desirable structure for multigrid applications. If a more traditional graded meshes are used on each level, proper nested subspaces are then hard to come by and the corresponding nonnested multigrid methods are more complicated (cf. Zhang [38]).

**6. Multigrid Methods for nonnested subspaces and varying bilinear forms.** Theories presented in previous sections are based on the assumptions that the multilevel subspaces are nested in the sense the the coarse spaces are subspaces of the finer spaces and that the bilinear forms on each level are all the same. In this section, we shall present a more general theory that do not depend on these assumptions. Such a theory has been successfully applied to many situations and some example of applications will be briefly mentioned near the end of this section.

Assume we are given a Hilbert space $H$ and a hierarchy of real finite dimensional subspaces of $H$

$$\mathcal{M}_0, \mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_J$$

which are related by the so-called prolongation operators $I_k : \mathcal{M}_{k-1} \mapsto \mathcal{M}_k$.

In addition, let $A_k(\cdot,\cdot)$ and $(\cdot,\cdot)_k$ be symmetric positive definite bilinear forms on $\mathcal{M}_k$ . We shall develop multigrid algorithms for the solution of the following problem: Given $f \in \mathcal{M}_J$, find $u \in \mathcal{M}_J$ satisfying

$$A_J(u,\phi) = (f,\phi)_J \quad \forall \phi \in \mathcal{M}_J.$$

To define the multigrid algorithms, we need to define some auxiliary operators. For $k = 0,\ldots,J$, the operator $A_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ is defined by

$$(A_k w,\phi)_k = A_k(w,\phi) \quad \forall w,\phi \in \mathcal{M}_k.$$

Clearly the operator $A_k$ is symmetric positive definite (in both the $A_k(\cdot,\cdot)$ and $(\cdot,\cdot)_k$ inner products). In terms of the prolongation operator $I_k$, we have operators $I_k^t : \mathcal{M}_k \mapsto \mathcal{M}_{k-1}$ and $I_k^* : \mathcal{M}_k \mapsto \mathcal{M}_{k-1}$ defined by

$$(6.1)\quad (I_k^t w,\phi)_{k-1} = (w, I_k\phi)_k A_{k-1}(I_k^* w,\phi) = A_k(w, I_k\phi) \quad \forall w \in \mathcal{M}_k, \phi \in \mathcal{M}_{k-1}.$$

In other words, $I_k^t$ are *iks* are the adjoint of $I_k$ with the inner products $(\cdot,\cdot)_k$ and $A_k(\cdot,\cdot)$ respectively. $I_k^t$ is often called *restriction* operator, which is another main ingredient of any multigrid algorithm.

Another important component of the multigrid algorithm is the *smoothing*, which will be represented by a sequence of linear operators $R_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ for $1 \le k \le j$ to define the smoothing process. These operators may be symmetric or nonsymmetric with respect to the inner product $(\cdot,\cdot)_k$. If $R_k$ is not symmetric, then we denote by $R_k^t$ its adjoint and set

$$R_k^{(l)} = \begin{cases} R_k & \text{if } l \text{ is odd;} \\ \\ R_k^t & \text{if } l \text{ is even.} \end{cases}$$

With the framework and notation given above, we are now in a position to define a multigrid algorithm, which will be characterized in terms of a sequence of recursively defined operators $B_k : \mathcal{M}_k \mapsto \mathcal{M}_k$. In the following, $p, m_k$ are given positive integers and $\lambda_k$ is either equal to $\rho(A_k)$ or an upper bound of $\rho(A_k)$ such that $\lambda_k \eqsim \rho(A_k)$.

**Algorithm S**

***Step 1*** $B_0 = A_0^{-1}$.

***Step 2*** *Assume $B_{k-1}$ is defined. Then $B_k$ is defined, for $g \in \mathcal{M}_k (A_k w = g)$, as follows:*

1. *Pre-smoothing on $\mathcal{M}_k$:*

$$\begin{aligned} &w^0 = 0 \\ &w^l = w^{l-1} + R_k^{(l+m_k)}(g - A_k w^{l-1}) \\ &l = 1,2,\cdots,m_k. \end{aligned}$$

2. *Correction on $\mathcal{M}_{k-1}$: $w^{m_k+1} = w^{m_k} + I_k q^p$ where $q^p \in \mathcal{M}_{k-1}$ is defined as follows, with $r_k = g - A_k w^{m_k}$,*

$$\begin{aligned} &q^0 = 0 \\ &q^l = q^{l-1} + B_{k-1}(I_k^t r_k - A_{k-1} q^{l-1}), \\ &l = 1,2,\cdots,p. \end{aligned}$$

*3. Post-smoothing on $\mathcal{M}_k$:*

$$w^l = w^{l-1} + R_k^{(l+m_k+1)}(g - A_k w^{l-1})$$
$$l = m_k + 2, \cdots, 2m_k + 1.$$

***Define:*** $B_k g = w^{2m_k+1}$.

**(6.2)** Remark. Ordinarily, the above multigrid methods can be made more general. For example, In Step 1, $B_1$ may be defined by an iterative method which solves the equation approximately on $\mathcal{M}_1$. Another generalization is that the number of pre- and post-smoothings are not necessarily the same. Nevertheless we are not going to consider these more general cases here. However in some circumstances it seems crucial to our theory that the number of pre- and post smoothings should be the same, which will guarantee that the multigrid operators $B_k$ are also symmetric(see Lemma 6.6). This is reasonable and important from many viewpoints. The most natural reason would be because the original problems are symmetric themselves.

A great advantage in setting out the algorithm by means of the operators $B_k$ is that we have a very simple recurrence relation for the "residue"operator $E_k \stackrel{def}{=} I - B_k A_k$ as given in the following lemma.

**(6.3) Lemma.** *Let $E_k = I - B_k A_k$ and $K_k = I - R_k A_k$. Then*

$$E_k = (\tilde{K}_k^{m_k})^* \left((I - I_k I_k^*) + I_k E_{k-1}^p I_k^*\right) \tilde{K}_k^{m_k}. \tag{6.4}$$

*Furthermore, for any $u, v \in \mathcal{M}_k$*

$$A_k(E_k u, v) = A_k((I - I_k I_k^*)\tilde{u}, \tilde{v}) + A_{k-1}(E_{k-1}^p I_k^* \tilde{u}, I_k^* \tilde{v}), \tag{6.5}$$

*where $\tilde{u} = \tilde{K}_k^{m_k} u$ and*

$$\tilde{K}_k^{m_k} = \begin{cases} (K_k^* K_k)^{\frac{m-1}{2}} K_k^*, & \text{if } m \text{ is odd;} \\ (K_k^* K_k)^{\frac{m}{2}}, & \text{if } m \text{ is even.} \end{cases}$$

The verification of the above lemma is straightforward by the definition of the algorithm. The next thing we want to address is that the **Algorithm S** defines a symmetric operator. More specifically, we have

**(6.6) Lemma.** *$B_k$ is symmetric with respect to $(\cdot,\cdot)_k$ and $E_k$ is symmetric with respect to $A_k(\cdot,\cdot)$.*

We observe that in the algorithm stated above, $p$ and $m_k$ are free parameters. With different parameters, we will take account of the three types of the algorithms named in the following:

**(6.7) Definition.** *The **Algorithm S** is known as the*

1. *V-cycle if $p = 1$ and $m_k = m \geq 1$ for $k = 0, \cdots, J$,*
2. *W-cycle if $p = 2$ and $m_k = m \geq 1$ for $k = 0, \cdots, J$,*
3. *Variable V-cycle if $p = 1$ and $\gamma_0 m_k \leq m_{k-1} \leq \gamma_1 m_k$ for $k = 0, \cdots, J$, where $\gamma_0$ and $\gamma_1$ are constants greater than 1.*

The subsequent multigrid convergence theory is largely based on two basic assumptions, one is concerned with the smoothing operators $R_k$ and the other is relevant to the "regularity" of the underlying problem and the approximation property of the

multilevel spaces. More refined convergence estimates depend on more assumptions on prolongation operators and such assumptions will be stated during the presentation of the convergence theory.

The assumption on the smoothing operator $R_k$ which concern only one level of space is the same as in the nested case.

**(A0)** $\qquad \|v\|^2 \leq C_0 \lambda_k(\bar{R}_k v, v)$

where $\bar{R}_k$ is the symmetrization of $R_k$ (see § 2-a).

A direct consequence of **(A0)** is

**(A0′)** $\qquad \rho(K_k) < 1,$

where $\rho(\cdot)$ denotes the spectral radius. This assumption will be used in place of **(A0)** to get some more general (but weaker) results.

The second assumption, usually called "regularity and approximation assumption", is that there exists a constant $\beta \in (0,1]$ and a constant $C_1$ such that

**(A1)** $\quad |A_k((I - I_k I_k^*)v, v)| \leq C_1(\lambda_k{}^{-1}\|A_k v\|_k{}^2)^{\beta} A_k(v,v)^{1-\beta} \quad \forall v \in \mathcal{M}_k.$

This is the most crucial assumption in the multigrid theory to be presented in this section. It relates the bilinear forms, different levels of spaces and prolongation operators. In the case of elliptic boundary value problems, its verification is strongly tied to the regularity property of the underlying partial differential equation.

It is not hard to see that **(A1)** implies that

(6.8) $\qquad A_k(I_k v, I_k v) \leq \tilde{C}_1 A_{k-1}(v,v), \quad \forall\, v \in \mathcal{M}_{k-1}.$

**(6.9) Theorem.** *Under the assumptions **(A0)** and **(A1)**, the Algorithm S has the following convergence properties (with a positive constant $M$ depending only on $C_0, C_1$ and $\beta$):*

*1. W-cycle converges uniformly for sufficiently many smoothings:*

$$\|\!|E_k|\!\|_k \leq \frac{M}{m^{\beta/2}} \text{ if } m^{\beta/2} \geq 2M.$$

*2. Variable V-cycle converges uniformly for sufficiently large $m_J$:*

$$\|\!|E_k|\!\|_k \leq \frac{c_0}{m^{\beta/2}} \text{ if } m_J^{\beta/2} \geq c_0.$$

*3. Variable V-cycle gives a uniform preconditioner with any fixed number of smoothings on the finest grid:*

$$\kappa(B_J A_J) \leq \frac{(M + m^{\beta})^2}{m^{2\beta}}.$$

*4. If the following condition holds*

***(A2)*** $\qquad A_k(I_k v, I_k v) \leq A_{k-1}(v,v) \quad \forall v \in \mathcal{M}_{k-1}.$

$$A_k(I_k v, I_k v) \leq A_{k-1}(v,v)$$

*Then, the W-cycle and VV-cycle converge uniformly and V-cycle converges nearly uniformly with any given number of smoothings:*

$$\|\!|E_k|\!\|_k \leq \begin{cases} \frac{M}{m_k^\beta + M} & \textit{for variable V-cycle} \\ \frac{M^\beta}{(m+M)^\beta} & \textit{for W-cycle} \\ \frac{k^{1/\beta-1}M}{m^\beta + k^{1/\beta-1}M} & \textit{for V-cycle} \end{cases}$$

*5. If the following condition holds*

$$A_k(I_k v, I_k v) \leq 2A_{k-1}(v, v),$$

*then the W-cycle converges uniformly with any given number of smoothings.*

The proof of the above theorem is not very complicated and may be found in [9] and [31]

**Applications.** The above theory has found many applications. First of all, the development of the theory was motivated to applications to multigrid methods for unstructured grids. Unstructured grids refer to grids that do not have a natural multilevel structures. Most of the grids generated by traditional grid generators may fall into such a categories. In this application, one has to coarsen the given grid to obtain a sequence of often nonnested multilevel coarse grids. There have been many coarsening techniques available in two dimensions, see Chan and Smith [16], Bank and Xu [4] and others.

Another important application is to multigrid methods for nonconforming finite elements (especially to fourth order problems). Nonconforming elements often give rise to nonnested multigrid subspaces (even on nested multilevel grids). A major concern in this application, is the choice of prolongation operators which are often obtained by using proper averaging techniques. In most applications, estimates like (A2) or (A3) are hard to be satisfied and hence only W-cycle or variable V-cycle can be proved to be convergent with sufficently many smoothings, or variable V-cyle gives rise to an optimal preconditioner. One interesting exception of the above phenomenon is the work by Chen and Oswald [17] and Chen [18] where they have proved that (A2) or (A3) can be satisfied for some nonconforming $P_1$ elements in some special situations. For this application and related subjects, we refer to Brenner [14] (and the references cited there),

**7. The auxiliary space method and an optimal preconiditioner for unstructured grids.** In this section, an abstract framework of *auxiliary space method* is proposed and, as an application, an optimal multigrid technique is developed for general unstructured grids. The auxiliary space method is a (nonnested) two level preconditioning technique based on a simple relaxation scheme (smoother) and an auxiliary space (that may be roughly understood as a nonnested coarser space). An optimal multigrid preconditioner is then obtained for a discretized partial differential operator defined on an unstructured grid by using an auxiliary space defined on a more structured grid in which a further *nested* multigrid method can be naturally applied. This new technique make it possible to apply multigrid methods to general unstructured grids without too much more programming effort than traditional solution methods.

The materials in this section are taken from Xu [33].

**7-a. The auxiliary space method.** The auxiliary space method is a general preconditioning approach based on a relaxation scheme and an auxiliary space. As mentioned in the introduction, this method can be interpreted in various ways but it may be best understood as a two level nonnested multigrid preconditioner. A detailed description of this approach will be described in details below and two theorems will be given for estimating the condition number of the preconditioned system.

Assume that a linear inner product space $\mathcal{V}$ is given together with a linear operator $A : \mathcal{V} \to \mathcal{V}$ that is SPD (symmetric, positive and definite) with respect to an inner product $(\cdot, \cdot)$. Consider the linear equation (2.1). The main ingredient in the new preconditioning technique is another auxiliary linear inner product space $\mathcal{V}_0$ together with an operator $A_0 : \mathcal{V}_0 \mapsto \mathcal{V}_0$ that is SPD with respect to an inner product $[\cdot, \cdot]$ on $\mathcal{V}_0$. This space, in most applications, may be viewed as certain approximation for $\mathcal{V}$. The space $\mathcal{V}_0$ needs not be a subspace of $\mathcal{V}$ in general, but it should be, in some sense, simpler than $\mathcal{V}$. The operator $A_0$ may be viewed as certain representation or approximation of $A$ in the space $\mathcal{V}_0$ and $A_0$ is assumed to be preconditioned by another SPD operator $B_0 : \mathcal{V}_0 \mapsto \mathcal{V}_0$. In another word, the auxiliary space $\mathcal{V}_0$ is chosen in such a way that the equation given by $A_0$ can be more easily solved than (2.1).

The auxiliary space $\mathcal{V}_0$ is linked with the original space $\mathcal{V}$ by an operator $\Pi : \mathcal{V}_0 \mapsto \mathcal{V}$. If $\mathcal{V}_0$ is viewed as a "coarse" space, $\Pi$ plays a role of prolongation in multigrid method. The "restriction" operator is given by its adjoint $\Pi^t : \mathcal{V} \mapsto \mathcal{V}_0$ defined by

$$[\Pi^t v, w] = (v, \Pi w) \quad v \in \mathcal{V}, w \in \mathcal{V}_0.$$

Another ingredient is an SPD operator $R : \mathcal{V} \mapsto \mathcal{V}$. The role of $R$ is to resolve what can not be resolved, in preconditioning $A$, by the aforementioned space $\mathcal{V}_0$ and the operators defined on $\mathcal{V}_0$. By the multigrid terminology, $R$ is like a smoother. In most applications, $R$ is given by a simple relaxation scheme such as Jacobi and Gauss-Seidel method.

With the ingredients described above, the proposed preconditioner is as follows.

$$B = R + \Pi B_0 \Pi^t. \tag{7.1}$$

In the special case that $\mathcal{V}_0 \subset \mathcal{V}$ and $[\cdot, \cdot] = (\cdot, \cdot)$, $\Pi$ can obviously be given by the natural inclusion operator and as a result $\Pi^t$ is nothing but the orthogonal projection $Q_0 : \mathcal{V} \mapsto \mathcal{V}_0$. In this case, the preconditioner (7.1) is reduced to

$$B = R + B_0 Q_0$$

which is the two-level special case of the general nested multilevel preconditioner in Bramble-Pasciak-Xu [9].

By definition, for any $u, v \in \mathcal{V}_h$,

$$(BAu, v)_A = (RAu, v)_A + [B_0 A_0 \Pi^* u, \Pi^* v]_{A_0} \tag{7.2}$$

where $(\cdot, \cdot)_A = (A\cdot, \cdot)$, $[\cdot, \cdot]_{A_0} = [A_0 \cdot, \cdot]$ and $\Pi^* = A_0^{-1} \Pi^t A$ satisfying

$$[\Pi^* v, w]_{A_0} = (v, \Pi w)_A \quad v \in \mathcal{V}, w \in \mathcal{V}_0.$$

Denote $\rho_A = \rho(A)$, the spectral radius of $A$. Also denote by $||\cdot||$ the norm induced by either $(\cdot, \cdot)$ or $[\cdot, \cdot]$. The main abstract result in this section is stated below.

**(7.3) Theorem.** *Assume that there are some nonnegative constants $\alpha_0, \alpha_1$, $\lambda_0$, $\lambda_1$, and $\beta_1$ such that, for all $v \in \mathcal{V}$ and $w \in \mathcal{V}_0$,*

$$\alpha_0 \rho_A^{-1}(v, v) \leq (Rv, v) \leq \alpha_1 \rho_A^{-1}(v, v), \tag{7.4}$$

*(7.5)* $$\lambda_0[w,w]_{A_0} \leq [B_0A_0w,w]_{A_0} \leq \lambda_1[w,w]_{A_0},$$

*(7.6)* $$\|\Pi w\|_A^2 \leq \beta_1\|w\|_{A_0}^2,$$

*and furthermore, assume that there exists a linear operator* $P : \mathcal{V} \mapsto \mathcal{V}_0$ *and positive constants* $\beta_0$ *and* $\gamma_0$ *such that,*

*(7.7)* $$\|Pv\|_{A_0}^2 \leq \beta_0^{-1}\|v\|_A^2$$

*and*

*(7.8)* $$\|v - \Pi Pv\|^2 \leq \gamma_0^{-1}\rho_A^{-1}\|v\|_A^2.$$

*Then the preconditioner given by* (7.1) *satisfies*

*(7.9)* $$\kappa(BA) \leq (\alpha_1 + \beta_1\lambda_1)((\alpha_0\gamma_0)^{-1} + (\beta_0\lambda_0)^{-1}).$$

*In particular, if* $P$ *is a right inverse of* $\Pi$ *namely* $\Pi Pv =$ *for* $v \in \mathcal{V}$*, then*

*(7.10)* $$\kappa((\Pi B_0\Pi^t)A) \leq \frac{\beta_1}{\beta_0}\frac{\lambda_1}{\lambda_0}.$$

*Proof.* One first notes that (7.6) is equivalent to

$$\|\Pi^* v\|_{A_0}^2 \leq \beta_1\|v\|_A^2 \quad \forall\, v \in \mathcal{V}.$$

By (7.2) and the assumptions, one has

$$(BAv,v)_A \leq \alpha_1\rho_A^{-1}\|Av\|^2 + \lambda_1\|\Pi^* v\|_{A_0}^2 \leq (\alpha_1 + \lambda_1\beta_1)\|v\|_A^2.$$

This means that $\lambda_{\max}(BA) \leq \alpha_1 + \beta_1\lambda_1$.

It follows that

$$\begin{aligned}
(v,v)_A &= (v - \Pi Pv, v)_A + [Pv, \Pi^* v]_{A_0} \\
&\leq \|v - \Pi Pv\|\|Av\| + \|Pv\|_{A_0}\|\Pi^* v\|_{A_0} \\
&\leq (\gamma_0\rho_A)^{-1/2}\|v\|_A\alpha_0^{-1/2}\rho_A^{1/2}(RAv,Av)^{1/2} \quad \text{(by (7.8) and (7.4))} \\
&\quad +\beta_0^{-1/2}\|v\|_A\lambda_0^{-1/2}[B_0A_0\Pi^* v, \Pi^* v]_{A_0}^{1/2} \quad \text{(by (7.7) and (7.5))} \\
&\leq ((\alpha_0\gamma_0)^{-1} + (\beta_0\lambda_0)^{-1})^{1/2}\|v\|_A(BAv,v)_A^{1/2} \quad \text{(by (7.2)).}
\end{aligned}$$

This implies that $\lambda_{\min}(BA) \geq ((\alpha_0\gamma_0)^{-1} + (\beta_0\lambda_0)^{-1})^{-1}$. The estimate (7.9) then follows.

In the particular case that $P$ is a right inverse of $\Pi$, one may take $\gamma_0 = \infty$ and $R = 0$, the proof of the corresponding estimate is then transparent. □

The last estimate for a special case in the above theorem corresponds to the *fictitious space lemma* of Nepomnyaschikh [26, 27]. A necessary condition for $\Pi$ to have a right inverse is that $\Pi : \mathcal{V}_0 \mapsto \mathcal{V}$ is surjective. As a consequence, $\dim\mathcal{V}_0 \geq \dim\mathcal{V}$, namely that $\mathcal{V}_0$ has to be at least as rich as $\mathcal{V}$. Furthermore the construction of $\Pi$ also needs more caution. The introduction of an additional smoother (or relaxation operator) $R$ greatly relaxes the constraints on the choice $\mathcal{V}_0$ and $\Pi$ in the fictitious space approach and, hence the resulting preconditioner is potentially more flexible and more robust.

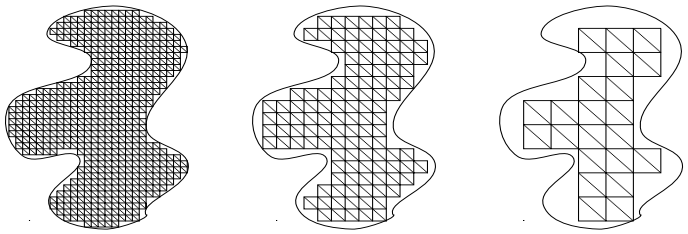


FIG. 4. *An example of an auxiliary grid together with two nested coarser grids.*

**7-b. A special technique for unstructured grids.** This subsection is to construct optimal multigrid preconditioners for finite element matrix from unstructured grids. The term "unstructured grids" here loosely mean those grids that do not possess natural or convenient multilevel structures. The main idea is to choose an auxiliary space from a rather structured grid in which a natural nested multilevel structure is available. This idea was briefly discussed in Xu [34].

In this section, the model problem in §3.1 will be used and the conforming piecewise linear finite element spaces over quasi-uniform triangulations $\mathcal{T}_h$ will be considered. Let $\Omega_h$ be the mesh domain determined by $\mathcal{T}_h$, namely

$$\bar{\Omega}_h = \cup_{\tau \in \mathcal{T}_h} \bar{\tau}.$$

To avoid unimportant technical difficulty, it is assumed, for all feasible $h$, that

$$\Omega_h \subset \Omega.$$

The finite element space $\mathcal{V}_h$ consists of continuous piecewise (with respect to $\mathcal{T}_h$) linear functions that vanish on $\bar{\Omega} \setminus \Omega_h$.

**A structured auxiliary finite element space.** Given a uniform square partition of the whole space with mesh size

$$h_0 \equiv 2^{-J} \eqsim h$$

for some integer $J \eqsim |\log h|$, let $\mathcal{T}_0$ be the union of squares ($d = 2$) or cubes ($d = 3$) that are contained in $\Omega$ (see Figure 1(a)). Let $\Omega_0$ denote the mesh domain determined by $\mathcal{T}_0$, namely

$$\bar{\Omega}_0 = \cup_{\tau \in \mathcal{T}_0} \bar{\tau}.$$

By construction, $\Omega_0 \subset \Omega$ and

$$\max_{x \in \bar{\Omega} \setminus \Omega_0} \operatorname{dist}(x, \partial\Omega) \lesssim h.$$

The set of interior nodes of $\mathcal{T}_0$ will be denoted by $\mathcal{N}_0 = \{x_j^0 : 1 \le j \le n_0\}$

An auxiliary finite element space $\mathcal{V}_0$ will be defined to be a space of continuous piecewise bilinear ($d = 2$) or trilinear ($d = 3$) functions that vanish on $\bar{\Omega} \setminus \Omega_0$. Alternatively, if $\mathcal{T}_0$ is a uniform triangulation (see Figure 1(a)) consisting of triangles ($d = 2$) or tetrahedrals ($d = 3$) from the aforementioned squares ($d = 2$) or cubes ($d = 3$), $\mathcal{V}_0$ may be defined to be a finite element space consisting of continuous piecewise linear functions that vanish on $\bar{\Omega} \setminus \Omega_0$.

The finite element space $\mathcal{V}_0$ will be used as the auxiliary space for the space $\mathcal{V}_h$.

Associated with the spaces $\mathcal{V}_h$ and $\mathcal{V}_0$, the operators: $A_h : \mathcal{V}_h \mapsto \mathcal{V}_h$ and $A_0 : \mathcal{V}_0 \mapsto \mathcal{V}_0$ are defined by

$$(A_h u, v) = a(u, v) \quad \forall\, u, v \in \mathcal{V}_h, \quad (A_0 u_0, v_0) = a(u_0, v_0) \quad \forall\, u_0, v_0 \in \mathcal{V}_0.$$

The operators between $\mathcal{V}_h$ and $\mathcal{V}_0$ will be the standard nodal value interpolants: $\Pi_h : \mathcal{V}_0 \mapsto \mathcal{V}_h$ and $\Pi_0 : \mathcal{V}_h \mapsto \mathcal{V}_0$. The approximation and stability properties of these two operators will be addressed in the following two lemmas.

**(7.11) Lemma.** *For all $v_0 \in \mathcal{V}_0$,*

$$||v_0 - \Pi_h v_0|| \lesssim h||v_0||_1, \quad ||\Pi_h v_0||_1 \lesssim ||v_0||_1. \tag{7.12}$$

**(7.13) Lemma.** *For all $v \in \mathcal{V}_h$,*

$$||v - \Pi_0 v|| \lesssim h||v||_1 \text{ and } ||\Pi_0 v||_1 \lesssim ||v||_1. \tag{7.14}$$

The proof of the above two lemmas can be found in Xu [33].

**An optimal multigrid preconditioner.** By the abstract approach in §2 and the auxiliary space $\mathcal{V}_0$, a preconditioner $B_h$ for $A_h$ can be obtained as follows

$$B_h = R_h + \Pi_h B_0 \Pi_h^t, \tag{7.15}$$

where $B_0 : \mathcal{V}_0 \mapsto \mathcal{V}_0$ is a given SPD preconditioner of $A_0$ and $R_h : \mathcal{V}_h \mapsto \mathcal{V}_h$ is an SPD smoother for $A$ satisfying

$$(R_h v, v) \eqsim h^2(v, v).$$

By Theorem 7.3 and the estimates from the previous subsections, there exist some positive constants $\alpha, \beta, \gamma$ that are independent of $h$ such that

$$\kappa(B_h A_h) \leq (\alpha + \beta\lambda_{\max}(B_0 A_0))(\gamma + \lambda_{\min}^{-1}(B_0 A_0)).$$

The preconditioner $B_0$ may be obtained by a further multigrid method. Such type of multigrid method has been discussed by Kornhuber and Yserentant [23]. Setting $\hat{\mathcal{T}}_J = \mathcal{T}_0$ (with $h_J = h_0 = 2^{-J}$) and $\hat{\mathcal{V}}_J = \mathcal{V}_0$, triangulations $\hat{\mathcal{T}}_k$ $(1 \leq k \leq J)$ (with $h_k = 2^{-k}$) and their corresponding spaces $\hat{\mathcal{V}}_k$ can be defined similarly. As shown in Figure 1, if Figure 1(a) corresponds to $\hat{\mathcal{V}}_J = \mathcal{V}_0$, then Figure 1(b) and Figure 1 (c) correspond to $\hat{\mathcal{V}}_{J-1}$ and $\hat{\mathcal{V}}_{J-2}$ respectively. Evidently

$$\hat{\mathcal{V}}_1 \subset \hat{\mathcal{V}}_2 \subset \ldots \subset \hat{\mathcal{V}}_J.$$

Consequently, a BPX preconditioner $B_0$ may be obtained so that [1] $\kappa(B_0 A_0) \lesssim 1$ and hence $\kappa(B_h A_h) \lesssim 1$, or a hierarchical basis preconditioner $B_0$ may be obtained so that $\kappa(B_0 A_0) \lesssim |\log h|^2$ and hence $\kappa(BA) \lesssim |\log h|^2$.

[1] Although the corresponding estimate was not optimal in [23] for their more general considerations, but it can be easily proved in the current context by the technique in [32].

Attention now is turned to the issues such as implementation and complexity of the aforementioned preconditioner. Let $\mathcal{A}_h$ and $\mathcal{A}_0$ be the stiffness matrices corresponding to the finite element spaces $\mathcal{V}_h$ and $\mathcal{V}_0$ respectively. Let $\mathcal{I}$ be the matrix representation of the interpolation $\Pi_h$, namely

$$(\Pi_h\psi_1, \ldots, \Pi_h\psi_{n_0}) = (\phi_1, \ldots, \phi_{n_h})\mathcal{I},$$

where $\{\psi_i : i = 1 : n_0\}$ and $\{\phi_j : j = 1 : n_h\}$ are the nodal bases for $\mathcal{V}_h$ and $\mathcal{V}_0$ respectively.

The precondition matrix for the stiffness matrix $\mathcal{A}$ can be written as (cf. [32])

$$\mathcal{B}_h = \mathcal{R}_h + \mathcal{I}\mathcal{B}_0\mathcal{I}^t$$

where $R_h$ represents Richardson, Jacobi or Gauss-Seidel iteration.

By definition, $\mathcal{I} = (\alpha_{ij}) \in \mathbb{R}^{n_h \times n_0}$ with $\alpha_{ij} = \psi_j(x_i^h)$. Obviously $\mathcal{I}$ is a sparse matrix with $O(n_h)$ nonzeros. The evaluation of $\alpha_{ij}$ depends on the location of $x_i^h$ relative to the partition $\mathcal{T}_0$. Because of the regularity of $\mathcal{T}_0$, each $x_i^h$ can be located in $\mathcal{T}_0$ with $O(1)$ operations by, for example, comparing the magnitude of the coordinates of $x_i^h$. Therefore $\mathcal{I}$ and $\mathcal{I}^t$ can be both obtained with $O(n_h)$ operations. For a more direct way of computing the action of $\mathcal{I}$, for example, if $\xi \in \mathbb{R}^{n_0}$ and $\eta = \mathcal{I}\xi$, then $\eta_i = w(x_i^h)$ with $w = \sum_{j=1}^{n_0} \xi_j\psi_j$. Again $w(x_i^h)$ can be easily obtained as long as the location of $x_i^h$ is known in $\mathcal{T}_0$.

In summary, when $B_0$ is given by BPX preconditioner, the resulting preconditioner which may be called *BPX preconditioner for unstructured grids* has the following features: 1. one action of $B$ requires only $O(n_h)$ operations; 2. the condition number of $BA$ is uniformly bounded independent of $h$ in both two and three dimensions; 3. it can be applied to unstructured grids.

REMARK. In practical computations, the auxiliary grid $\mathcal{T}_0$ can be more flexible than given above. For example, one does not have to take the elements that are exactly contained in $\Omega$.

REMARK. For simplicity, the details for unstructured grids are only given for Dirichlet boundary value problems. Applications to more general cases are also possible. For Neumann boundary condition, for example, it is not sufficient that the auxiliary grid only consists of those regular elements that are contained in $\Omega$. Instead, the auxiliary grid consists of all those regular elements that *intersect* with $\Omega$. The application of the technique to locally refined meshes is a little more complicated. Again the idea is to use a structured refined mesh to define the auxiliary space. Locally refined meshes with multilevel structures were discussed in Brandt [12], McCormick [25] (see also the references therein), Bramble-Pasciak-Xu [9] and Bramble-Pasciak [6].

*Some remarks.* The main spirit of this section is that, with the help of an additional smoother, a quite rough auxiliary space can be used to construct an optimal preconditioner for a discretized partial differential operator. For an elliptic partial differential equation of order $2m$, for example, the auxiliary space $\mathcal{V}_0$ for a given finite element space $\mathcal{V}$ defined on a grid of size $h$ needs only to satisfy the following approximation property:

$$\inf_{w \in \mathcal{V}_0} ||v - w||_0 \lesssim h^m ||v||_{m,h} \quad \forall\, v \in \mathcal{V} \tag{7.16}$$

where $||\cdot||_0$ is the $L^2$ norm and $||\cdot||_{m,h}$ is a (discretized) $H^m$ norm.

The approximation property (7.16) is a very weak one and is certainly much weaker than the approximation property that $\mathcal{V}$ (as any reasonable finite element space) should have. One important point to address is that the role of $\mathcal{V}_0$ or the role of a coarse grid in a multigrid algorithm is to resolve the spectrum of the discretized differential operator and there is no reason that $\mathcal{V}_0$ should be comparable with $\mathcal{V}$ as far as their approximation properties are concerned. Roughly speaking, the spectral property of a discretized partial differential operator is mainly determined by the original partial differential operator rather than the underlying discretization space. Hence in order to capture the spectrum of a discretized operator, the auxiliary space only need to have an approximation property like (7.16) that is related to the order of the original differential operator but not, in certain sense, strongly related to the discretization space ($\mathcal{V}$).

The weak approximation property (7.16) makes it possible to use a simple and structured auxiliary space for preconditioning a complicated and unstructured problem. As the main application of this general idea, the main concrete conclusion of this section is that a finite element space defined on an unstructured grid can be well preconditioned by combining a simple relaxation scheme and a structured grid. As a consequence, it is possible to solve a finite element equation on a general unstructured grid by a multigrid approach with an optimal computational complexity.

## REFERENCES


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

[2] R. E. Bank and T. Dupont. An optimal order process for solving elliptic finite element equations. *Math. Comp.*, 36:35–51, 1981.

[3] R. E. Bank, T. Dupont, and H. Yserentant. The hierarchical basis multigrid method. *Numer. Math.*, 52:427–458, 1988.

[4] R.E. Bank and J. Xu. An algorithm for coarsening unstructured meshes. *Numer. Math.*, 1995. (to appear).

[5] J. H. Bramble. *Multigrid Methods*, volume 294 of *Pitman Research Notes in Mathematical Sciences*. Longman Scientific & Technical, Essex, England, 1993.

[6] J. H. Bramble and J. E. Pasciak. New estimates for multigrid algorithms including the V–cycle. *Math. Comp.*, 60:447–471, 1993.

[7] J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu. Convergence estimates for multigrip algorithms without regularity assumptions. *Math. Comp.*, 57:23–45, 1991.

[8] J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu. Convergence estimates for product iterative methods with applications to domain decomposition. *Math. Comp.*, 57:1–21, 1991.

[9] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comp.*, 55:1–22, 1990.

[10] J. H. Bramble, J. E. Pasciak, and J. Xu. The analysis of multigrid algorithms with nonnested spaces or noninherited quadratic forms. *Math. Comp.*, 56:1–34, 1991.

[11] J. H. Bramble and J. Xu. Some estimates for a weighted $l^2$ projection. *Math. Comp.*, 56:463–476, 1991.

[12] A. Brandt. Multi–level adaptive solutions to boundary–value problems. *Math. Comp.*, 31:333–390, 1977.

[13] A. Brandt. *Multigrid techniques: 1984 guide with applications to fluid dynamics*. GMD–Studien Nr. 85. Gesellschaft für Mathematik und Datenverarbeitung, St. Augustin, 1984.

[14] S. C. Brenner. Multigrid methods for nonconforming finite elements. In J. Mandel and S. F. McCormick, editors, *Preliminary Proc. of the 4th Copper Mountain Conference on Multigrid Methods*, volume 1, pages 135–149, Denver, 1989. Computational Mathematics Group, Univ. of Colorado.

[15] W. L. Briggs. *A Multigrid Tutorial*. SIAM Books, Philadelphia, 1987.

[16] T. F. Chan and Barry Smith. Domain decomposition and multigrid methods for elliptic prob-

lems on unstructured meshes. In David Keyes and Jinchao Xu, editors, *Domain Decomposition Methods in Science and Engineering, Proceedings of the Seventh International Conference on Domain Decomposition, October 27-30, 1993, The Pennsylvania State University*. American Mathematical Society, Providence, 1994.

[17] Z. Chen. Multigrid and multilevel methods for nonconforming rotated $q_1$ elements. preprint, 1996.

[18] Z. Chen. On the convergence of nonconforming multigrid methods for second-order elliptic problems. preprint, 1996.

[19] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North–Holland, Amsterdam, New York, 1978.

[20] R. P. Fedorenko. A relaxation method for solving elliptic difference equations. *Z. Vycisl. Mat. i. Mat. Fiz.*, 1:922–927, 1961. Also in U.S.S.R. Comput. Math. and Math. Phys., 1 (1962), pp. 1092–1096.

[21] W. Hackbusch. *Multigrid Methods and Applications*, volume 4 of *Computational Mathematics*. Springer–Verlag, Berlin, 1985.

[22] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*. Springer–Verlag, Berlin, 1993.

[23] R. Kornhuber and H. Yserentant. Multilevel methods for elliptic problems on domains not resolved by the coarse grid. In *Domain Decomposition Methods in Scientific and Engineering Computing: Proceedings of the Seventh International Conference on Domain Decomposition*, volume 180 of *Contemporary Mathematics*, pages 49–60, Providence, Rhode Island, 1994. American Mathematical Society.

[24] S. F. McCormick. *Multigrid Methods*, volume 3 of *Frontiers in Applied Mathematics*. SIAM Books, Philadelphia, 1987.

[25] S. F. McCormick. *Multilevel Adaptive Methods for Partial Differential Equations*, volume 6 of *Frontiers in Applied Mathematics*. SIAM Books, Philadelphia, 1989.

[26] S. V. Nepomnyaschikh. Mesh theorems of traces, normalizations of function traces and their inversion. *Sov. J. Numer. Anal. Math. Modeling*, 6:151–168, 1991.

[27] S. V. Nepomnyaschikh. Decomposition and fictitious domains methods for elliptic boundary value problems. In D. E. Keyes, T. F. Chan, G. A. Meurant, J. S. Scroggs, and R. G. Voigt, editors, *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 62–72, Philadelphia, 1992. SIAM.

[28] R. A. Nicolaides. On a geometrical aspect of sor and the theory of consistent ordering for positive definite matrices. *Numer. Math.*, 23:99–104, 1974.

[29] M. E. G. Ong. The 3D linear hierarchical basis preconditioner and its shared memory parallel implementation. In *Vector and Parallel Computing: issues in applied research and development*, pages 273–283. Ellis Horwood Ltd. (J. Wiley & Sons), Chichester, 1989.

[30] P. Wesseling. *An Introduction to Multigrid Methods*. John Wiley & Sons, Chichester, 1992.

[31] J. Xu. *Theory of Multilevel Methods*. PhD thesis, Cornell University, 1989.

[32] J. Xu. Iterative methods by space decomposition and subspace correction: A unifying approach. *SIAM Review*, 34:581–613, 1992.

[33] J. Xu. An auxiliary space preconditioning technique with application to unstructured grids. *Numer. Math.*, 1995. (to appear).

[34] J. Xu. Multigrid and domain decomposition methods. In J. Xu W. Cai, J. Shu and Z. Shi, editors, *Numerical Analysis in Applied Sciences*. Science Press, 1995.

[35] J. Xu and J. Qin. Some remarks on a multigrid preconditioner. *SIAM J. Sci. Comput.*, 15:172–184, 1994.

[36] H. Yserentant. On the multi–level splitting of finite element spaces. *Numer. Math.*, 49:379–412, 1986.

[37] H. Yserentant. Old and new convergence proofs for multigrid methods. *Acta Numerica*, 1992.

[38] S. Zhang. Optimal order non–nested multigrid methods for solving finite element equations I-III. *Math. Comp.*, 1990,1991,1994.