AN INTRODUCTION TO MULTILEVEL METHODS

JINCHAO XU *

Abstract. An introduction is given in this paper to the basic idea and the convergence theory of multilevel methods including overlapping domain decomposition methods and multigrid methods. Brief discussions are first given of 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 devoted to a general framework of subspace correction methods that can be applied to, among many other things, multilevel methods. A framework of auxiliary space methods is also briefly presented for the construction of preconditioners. The overlapping domain decomposition method and multigrid method are introduced with a model elliptic boundary value problem of second order. Convergence estimates are obtained for an overlapping domain decomposition method and especially for basic multigrid methods such as backslash (\) cycle, V-cycle and W-cycle. Two different approaches are used in the convergence analysis for multigrid methods. 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 preconditioning technique is given for general unstructured grids using the auxiliary space framework. For nonsymmetric and/or indefinite problems, some two grid techniques are discussed. In addition to the aforementioned theoretical analysis, some discussions are also devoted to the implementation of some basic multigrid algorithms.

* Department of Mathematics, Penn State University, USA. Email: xu@math.psu.edu. This author is partially supported by NSF DMS94-03915-1 through Penn State and by NSF ASC-92-01266

1. Introduction. Multilevel 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 an introduction to these methods and their convergence properties by considering their applications to a model elliptic boundary value problem of second order.

Multilevel methods have been most efficiently used in solving the linear algebraic systems 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 multilevel theory. We shall devote §2, §4 and §5 to the technical materials for the theory of multilevel 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 multilevel procedure, and also a multilevel method is often used in conjunction with a preconditioned conjugate gradient method. Therefore the materials in §2 are fundamental to our multilevel algorithms and theory.

§3 is a self-contained introduction to the very basic ideas for classic multigrid methods by using the simplest possible one dimensional model problem.

§4 is on an algebraic framework of subspace correction methods (following Xu [33]) that can be used in general for construction and analysis of linear iterative methods. This framework will be a main technical tool in the analysis of domain decomposition methods in §6 and multigrid methods in §7.

The most technical materials in this paper are perhaps those in §5 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.

Overlapping domain decomposition methods are presented and analysed in §6. In particular, it is demonstrated that a recursive application of nonoverlapping domain decomposition (corresponding to smallest possible subdomains) will naturally lead to a typical multigrid method.

The core of this paper is §7 in which many major multigrid algorithms are introduced and analysed. 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 analysed with two different approaches. The first approach is the more traditional one which makes crucial use of regularity theory of partial differential equations. The second approach is the subspace correction framework in §4.

The multigrid algorithms and their convergence analysis presented in §7 are only for the case that the underlying multilevel spaces are nested in the sense that the coarse spaces are subspaces of finer spaces. §8 is devoted to giving readers an idea of how multigrid methods can be applied to more complicated situations, where a general framework of nonnested multigrid methods which can be applied to cases like unstructured grids and nonconforming elements is presented, and a special technique is given for constructing optimal multigrid preconditioning techniques for unstructured grids using the framework of auxiliary space methods.

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 a small part of the whole subject. For further details, we refer to the research monographs

of Hackbusch [20, 21], McCormick [23], Wesseling [31] and Bramble [5], and to the review articles of Xu [33] and Yserentant [44].

For convenience, following [33], 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 (symmetric positive definite) linear operator over $\mathcal{V}$ and $f \in \mathcal{V}$ is given.

2.1. 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 $r^{\text{old}} = f - Au^{\text{old}}$;
2. solve $Ae = r^{\text{old}}$ approximately: $\hat{e} = Br^{\text{old}}$;
3. update $u^{new} = u^{\text{old}} + \hat{e}$;

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.

ALGORITHM 2.2. *Given* $u^0 \in \mathcal{V}$,

$$u^{k+1} = u^k + B(f - Au^k), \quad k = 0, 1, 2, \cdots. \tag{2.3}$$

The core of the above iterative 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 (2.3) 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 (2.3) respectively. Then

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

Therefore the iterative scheme (2.3) converges iff $\rho(I - BA) < 1$.

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

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

or

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

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

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

or

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

Obviously $\bar{B}$ is symmetric and will be called the *symmetrization* of the 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

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

THEOREM 2.1. *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* $v \in \mathcal{V}$*:*

$$(BAv, BAv)_A \leq \omega_1 (BAv, v)_A; \tag{2.9}$$

$$(Av, v) \leq \omega_1 (B^{-1}v, v); \tag{2.10}$$

$$\left(\frac{2}{\omega_1} - 1\right)(Av, v) \leq ((B^{-1} + B^{-t} - A)v, v); \tag{2.11}$$

$$(2 - \omega_1)(Bv, v) \leq (\bar{B}v, v). \tag{2.12}$$

Furthermore, the scheme (2.3) 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

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

Richardson iterative methods. Richardson iteration is perhaps the simplest iterative method which corresponds to (2.3) with $B = \frac{\omega}{\rho(A)} I$. Namely,

$$u^{k+1} = u^k + \frac{\omega}{\rho(A)}(f - Au^k), \quad k = 0, 1, 2, \cdots, \tag{2.14}$$

One can imagine that the Richardson method is not very efficient, but it is theoretically very important. One of the most important properties 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

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

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

An iterative method (2.3) is said to be Richardson-like if there exists an $\omega \in (0, 2)$ such that

$$\|(I - BA)v\|_A \leq \|(I - \frac{\omega}{\rho(A)} A)v\|_A \quad \forall\, v \in \mathcal{V}. \tag{2.15}$$

LEMMA 2.2. *For the iterative method (2.3), the following are equivalent.*

1. *The inequality* (2.15) *is satisfied with* $\omega = C_0^{-1}$.
2. $(C_0 \rho(A))^{-1} \|v\|^2 \leq (\bar{B}v, v) \quad \forall\, v \in \mathcal{V}$.
3. $(C_0 \rho(A))^{-1} \|Av\|^2 \leq \|v\|_A^2 - \|(I - BA)v\|_A^2 \quad \forall\, v \in \mathcal{V}$.

2.2. 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 parts of A respectively. The easiest approximate inverses of A are perhaps

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

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

$$B = \begin{cases} \omega & \text{Richardson;} \\ D^{-1} & \text{Jacobi;} \\ \omega D^{-1} & \text{Damped Jacobi;} \\ (D - L)^{-1} & \text{Gauss–Seidel;} \\ \omega (D - \omega L)^{-1} & \text{SOR.} \end{cases} \tag{2.16}$$

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

THEOREM 2.3. *Assume A is SPD. Then*

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

The proof of the above results follow directly from Theorem 2.1 by (2.16) to compute $B^{-t} + B^{-1} - A$. For example, for the SOR method, $B^{-t} + B^{-1} - A = \frac{2-\omega}{\omega} D$.

2.3. 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 ν 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

$$\widetilde{AB} = \widetilde{A}\widetilde{B} \quad \text{and} \quad \widetilde{Av} = \widetilde{A}\widetilde{v}. \tag{2.17}$$

Under the basis (ϕ_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 easily be shown that

$$\mathcal{A} = \mathcal{M}\widetilde{A},$$

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

$$Rv = \sum_{i=1}^n (v, \phi_i)\phi_i, \quad \forall\, v \in \mathcal{V}. \tag{2.18}$$

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

$$\mathcal{A}\mu = \eta. \tag{2.19}$$

Similarly to (2.3), a linear iterative method for (2.19) can be written as

$$\mu^{k+1} = \mu^k + \mathcal{B}(\eta - \mathcal{A}\mu^k), \quad k = 0, 1, 2, \cdots, \tag{2.20}$$

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

PROPOSITION 2.4. *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 μ is the solution of* (2.19). *The linear iterations (2.3) and* (2.20) *are equivalent if and only if $\widetilde{B} = \mathcal{B}\mathcal{M}$. In this case the condition numbers are related by $\kappa(\mathcal{B}\mathcal{A}) = \kappa(BA)$.*

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

Using the properties of the operator defined by (2.18), we can prove the following simple result.

PROPOSITION 2.5. *The scheme (2.3) represents the Richardson iteration for the equation* (2.19) *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

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

2.4. 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:

$$BAu = Bf. \tag{2.21}$$

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

$$\|u - u_k\|_A \leq 2\Big(\frac{\sqrt{\kappa(BA)} - 1}{\sqrt{\kappa(BA)} + 1}\Big)^k \|u - u_0\|_A, \tag{2.22}$$

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

Observing the formulae in the PCG method and the convergence estimate (2.22), 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. Multigrid methods for a one dimensional problem. In this section, we give a self-contained introduction to multigrid ideas for a one dimensional problem. The motivation here is to present the very basic ideas of classic multigrid methods using the simplest possible model.

3.1. Basic multigrid ideas. In this subsection, we discuss some multigrid ideas by studying a simple two point boundary value problem and its finite difference discretizations. The presentation is informal but hopefully informative. Numerical examples are used to illustrate the main ideas.

Two point boundary value problem and finite difference discretization. We consider the following simplest possible elliptic boundary problem:

$$-u'' = f, \quad x \in (0,1) \qquad u(0) = u(1) = 0. \tag{3.1}$$

We use a simple central finite difference method to discretize the above equation. To do that, for any integer N, we consider a uniform grid, denoted by $\mathcal{T}_h$, of the interval $[0, 1]$ as follows:

$$0 = x_0 < x_1 < \cdots < x_{N+1} = 1, \quad x_j = \frac{j}{N+1} (j = 0 : N+1). \tag{3.2}$$

This partition consists of uniform subintervals of the length $h = \frac{1}{N+1}$.

For a smooth solution u, we have

$$u''(x_j) \approx \frac{u(x_{j-1}) - 2u(x_j) + u(x_{j+1})}{h^2}.$$

Therefore, if $\mu = (\mu_i) \in R^N$ satisfies

$$\frac{-\mu_{j-1} + 2\mu_j - \mu_{j+1}}{h^2} = f(x_j), \quad 1 \le j \le N, \quad \mu_0 = \mu_{N+1} = 0,$$

then

$$\mu_j \approx u(x_j).$$

It is natural to expect that this approximation gets more accurate as N gets larger.

The finite difference discretization gives rise to a discrete set of approximation (μ_j). It is often convenient to interpolant these discrete values into some function that is defined everywhere on the interval $[0, 1]$. One natural way is the *linear interpolation*. The linear interpolation of the vector (μ_j) is a piecewise linear function, u_h, with respect to the partition (3.2), such that

$$u_h(x_j) = \mu_j, \quad 0 \le j \le N+1, \quad \mu_0 = \mu_{N+1} = 0.$$

Figure 1 shows the plots of the exact solution of (3.1) with $f(x) = 1$ and linearly interpolated u_h from its finite difference approximation with $N = 5$ and $N = 7$ and $N = 15$ respectively. As we see the approximation property of the finite difference discretization is reasonable.

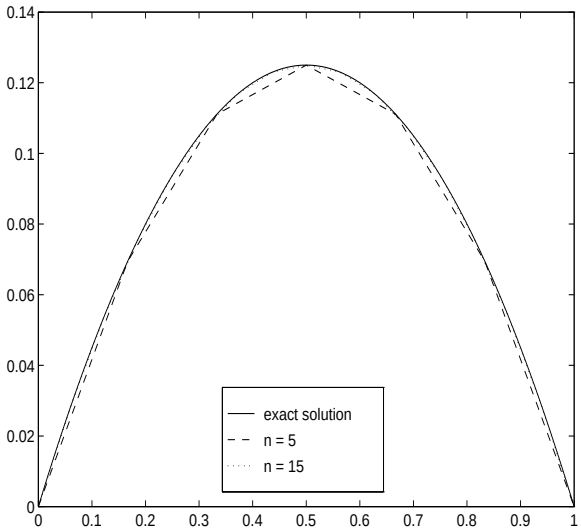

FIG. 1. *An exact solution and two approximate solutions*

Algebraic system for the discrete solution. Richardson iteration. Our main concern here is on how to effectively find the discrete solution, namely to solve the following linear algebraic system for μ:

$$A\mu = \beta. \tag{3.3}$$

where

$$A = \frac{1}{h}\text{diag}(-1, 2, -1), \beta_j = hf(x_j).$$

Here $\mathrm{diag}(-1, 2, -1)$ denotes the tridiagonal matrix with main diagonal entries being 2 and both sub-diagonals being -1.

One probably wonder why we choose to keep a scaling factor of $1/h$ in the above definition of stiffness matrix. The main reason to do so is to make our notation be consistent with that from the finite element method we will discuss more late. Although this additional constant factor may look a little awkward, it should not add much complication.

Of course, this particular tridiagonal system can be easily solved by many methods such as banded Gaussian elimination. But for illustrational purpose, we shall instead consider to use iterative methods.

One very simple iterative method for (3.3) is the following Richardson method

$$\mu^l = \mu^{l-1} + \sigma^{-1}(b - A\mu^{l-1}),$$

or, for $j = 1 : N$,

$$\mu_j^l = \mu_j^{l-1} + \sigma^{-1}\left(\beta_j - \frac{-\mu_{j-1}^{l-1} + 2\mu_j^{l-1} - \mu_{j-1}^{l-1}}{h}\right),$$

where $\sigma > 0$ is a positive parameter. It is not so difficult to properly choose σ so that the above iterative scheme converges, namely for any initial guess μ^0, the sequence (μ^l) generated by the above iteration converges to the exact solution μ of (3.3). In fact, a necessary and sufficient condition for the convergence is the following

$$\sigma > 2/\rho(A).$$

Here $\rho(A)$ is the spectral radius of A. It is easy to see that (for example, $4/h$ is an upper bound of its raw sums).

$$\rho(A) < 4/h.$$

Therefore it is reasonable to make the following choice:

$$\sigma = 4/h$$

and the resulting algorithm is

$$\mu^l = \mu^{l-1} + \frac{h}{4}(b - A\mu^{l-1}). \tag{3.4}$$

In the rest of this subsection, unless otherwise noted, we shall choose σ as above for simplicity.

On Figure 2 the convergence history plot of the above Richardson iterative method for typical application is shown. As we see, this iterative scheme converges very slowly.

Our main goal is to find a way to speed up such kind of rather slowly convergent iterative scheme. To do that, we need to study its convergent property in more microscopic level. First of all, let us now take a careful look at the convergence history picture and make the following observation:

> *Observation 1.* The scheme converges rather fast in the very beginning but then slows down after a few step.

To further understand this phenomenon, let us plot the detailed pictures of the error functions in the first few iterations. After a careful look at these pictures, we have the following observation:

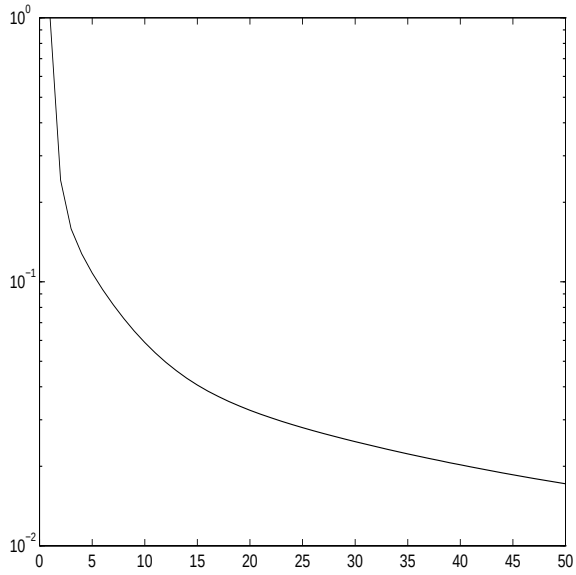

FIG. 2. *A picture on the Richardson method convergence history*

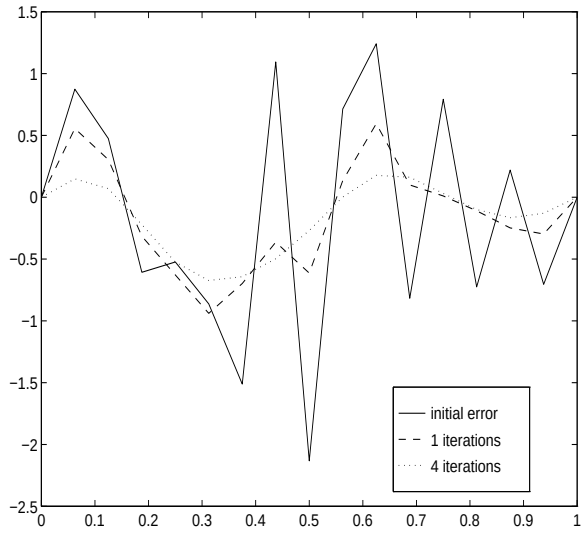

FIG. 3. *The smoothing effect of the Richardson method*

Observation 2. The scheme not only converges fast in the first few steps, but also smooth out the error function very quickly.

In other words, the error function becomes a much smoother function after a few such simple iterations. This property of the iterative scheme is naturally called *a smoothing property* and an iterative scheme having this smoothing property is called a *smoother*.

The above two observations, especially the second one, concern the most important property of the simple Richardson method that we can take advantage to get a

much faster algorithm.

Coarse grid correction and a two-grid method. As we discussed above, although the Richardson iteration does not converges very fast, it does quickly smooth out the rough component in the error. We shall now explore how to speed up the convergence. The main idea is to use the fact that a relatively smooth function can be well approximated on a relatively coarse grid. Now, since the error is smooth after a few Richardson iterations, we should be able to correct the residual effectively using a coarser grid.

Let μ^m denote the approximate solution obtained after m-th iteration of the Richardson method and $r = b - A\mu^m$. We consider the residual equation as follows:

$$A_h \epsilon^h = \gamma^h. \tag{3.5}$$

Here $A_h = A$ and $\gamma^h = \gamma$ and we use the superscript h to refer that the above equation is on the grid of size h. Here $\epsilon = \mu - \mu^m$. Since ϵ is smooth, ϵ should be approximated well on a grid coarser than the original grid. For simplicity, we assume that N is an odd integer and we take the coarse grid, denoted by $\mathcal{T}_{2h}$ to be a grid of size $2h$ with the following grid points

$$x_i^{2h} = x_{2i}, i = 1 : N_{2h} \equiv (N+1)/2. \tag{3.6}$$

We would like to restrict the equation (3.5) to the above coarse grid and then solve the coarse grid equation. To do this, we first need to find a reasonable way for obtaining the restriction of (3.5) to the coarse grid. To this end, we can imagine that (3.5) is the linear system corresponding to a finite different discretization of a two point boundary value problem:

$$-\epsilon'' = g, \quad x \in (0,1) \qquad E(0) = E(1) = 0. \tag{3.7}$$

Here g is some function satisfying

$$\gamma_i = hg(x_i). \tag{3.8}$$

Now, we use finite difference to discretize (3.7) on the coarse grid $\mathcal{T}_{2h}$ and obtain the corresponding system as follows:

$$A_{2h} \epsilon^{2h} = \gamma^{2h} \tag{3.9}$$

where $A_{2h} = (2h)^{-1}\text{diag}\ (-1, 2, -1) \in R^{N_{2h} \times N_{2h}}$ is the stiffness matrix from the coarse grid $\mathcal{T}_{2h}$ and

$$\gamma_i^{2h} = 2hg(x_i^{2h}). \tag{3.10}$$

Comparing (3.8) with (3.10) and using (3.6), we obtain the following relation for the components of the restricted vector γ^{2h}:

$$\gamma_i^{2h} = 2\gamma_{2i}^h. \tag{3.11}$$

The equation (3.9) with (3.11) is a desired restricted equation of (3.5) to the coarse grid $\mathcal{T}_{2h}$.

We would like to mention the following slightly different restriction is often used instead of (3.11):

$$\gamma_i^{2h} = \frac{1}{2}\gamma_{2i-1}^h + \gamma_{2i}^h + \frac{1}{2}\gamma_{2i+1}^h \tag{3.12}$$

which is apparently very close to (3.11) since, for a smooth residual, $\gamma_{2i-1}^h \approx \gamma_{2i+1}^h \approx \gamma_{2i}^h$. We can imagine that using an averaged value should be better than a single value. We shall see in the next subsection, this is the most natural restriction in the finite element setting. This restriction also has a more interesting relation with the prolongation matrix that we will introduce in a moment.

The relation (3.11) or (3.12) defines a so-called restriction matrix $I_h^{2h} : R^{N_h} \mapsto R^{N_{2h}}$ such that

$$\gamma_{2h} = I_h^{2h}\gamma^h.$$

The equation (3.9) is half the size of (3.5). Let us solve this equation exactly and obtain:

$$\epsilon^{2h} = A_{2h}^{-1}\gamma^{2h}.$$

This ϵ^{2h} is the coarse grid correction of μ^m. To add this correction to μ^m, we need to think ϵ^{2h} to be the nodal value vector of a piecewise linear function defined on $\mathcal{T}_{2h}$. Apparently this piecewise linear function can be evaluated at the grid points of $\mathcal{T}_h$ as follows:

$$\tilde{\epsilon}_{2i}^h = \epsilon_i^{2h}, \tilde{\epsilon}_{2i+1}^h = \frac{1}{2}(\epsilon_i^{2h} + \epsilon_{i+1}^{2h}), \quad i = 1 : N_{2h}. \tag{3.13}$$

The above relation defines a so-called prolongation matrix $I_{2h}^h : R^{N_{2h}} \mapsto R^{N_h}$ such that

$$\tilde{\epsilon}^h = I_{2h}^h\epsilon^{2h}.$$

It is easy to see this prolongation matrix is related to the restriction matrix defined by (3.12) by the following identity:

$$I_{2h}^h = [I_h^{2h}]^t$$

where t is the transposition. Because of this relationship, the restriction given by (3.12) is particularly interesting.

With this prolongation matrix, the approximation μ^m can be updated as follows

$$\mu^m + I_{2h}^h\epsilon^{2h}.$$

ALGORITHM 3.14 (A TWO GRID METHOD). *Given μ^0, $\mu \leftarrow \mu^0$*

1. *Fine grid smoothing: applying m times Richardson iterations to obtain μ^m.*
2. *Coarse grid correction: solving the residual equation restricted on the coarse grid $\mathcal{T}_{2h}$ to obtain*

$$\epsilon^{2h} = A_{2h}^{-1}(I_h^{2h}\gamma^h).$$

3. *Update: $\mu \leftarrow \mu^m + I_{2h}^h\epsilon^{2h}$.*
4. *Stop if convergent or continue from step 1.*

Let us now give a numerical example to show how well the above two-grid method works.

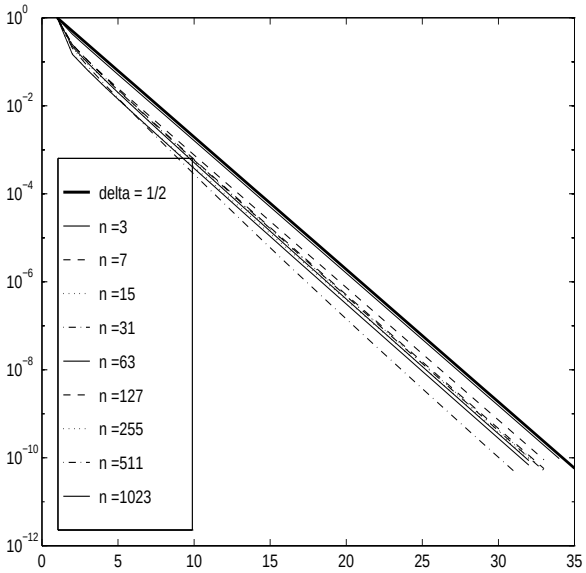

FIG. 4. *Convergence history of the two level method for different number of unknowns (different values of h respectively). The thick line corresponds to damping factor* 1/2. *Only one smoothing step is applied in the algorithm.*

A different formulation and possible multigrid extension. We would like to present the above two-grid method in a slightly different but equivalent fashion. Let us first only consider the above algorithm in the following special case:

$$\mu^0 = 0, \text{with step 4 removed}$$

This means nothing but one sweep of Algorithm 3.14 with zero initial guess. Recall we are trying to solve the system $A_h\xi^h = \beta^h$. Given the right hand side $\beta^h \in R^{N_h}$, with the aforementioned one sweep of two-grid iteration, we obtain an approximate solution which we denote by $\tilde{\xi}^h \in R^{N_h}$. This procedure defines the following transformation:

$$\beta^h \mapsto \tilde{\xi}^h.$$

It is not hard to see that the above transformation is in fact linear and hence it corresponds to a matrix $B_h \in R^{N_h \times N_h}$. Namely

$$\tilde{\xi}^h = B_h\beta^h.$$

Since $\tilde{\xi}^h$ is meant to be an approximation of the solution of the equation $A_h\xi^h = \beta^h$, B_h can then thought as an approximate inverse of A_h.

For clarity, let us reiterate the detailed definition of B_h as follows:

ALGORITHM 3.15 (AN ALGORITHM DEFINING THE ACTION OF B_h). *Let* $\beta^h \in R^{N_h}$ *be given.*

1. *Fine grid smoothing: with* $\mu^0 = 0$,
$$\mu^l = \mu^{l-1} + \sigma^{-1}(b - A\mu^{l-1}),\ l = 1 : m.$$
2. *Coarse grid correction: Form the residual* $\gamma^h = \beta^h - A_h\mu^m$ *and solve the coarse grid residual equation* $A_{2h}\epsilon = I_h^{2h}\gamma^h$ *exactly*
$$\epsilon^{2h} = A_{2h}^{-1}(I_h^{2h}\gamma^h).$$

3. Define: $B_h\beta^h = \mu^m + I_{2h}^h \epsilon^{2h}$.

With the B_h defined above, it is clear that Algorithm 3.15 is equivalent to the following iteration:

$$\mu \leftarrow \mu^0, \text{ iterate } \mu \leftarrow \mu + B_h(\beta^h - A_h\mu) \text{ till convergence.}$$

In the above two-grid algorithm, an exact coarse grid solver on $\mathcal{T}_{2h}$ is used. If the grid $\mathcal{T}_{2h}$ is not very coarse, such an exact solver can still be very costly. To improve the efficiency, we need to replace this exact solver by some appropriate approximate solver. The idea is to repeat the same two-grid method on this equation on $\mathcal{T}_{2h}$ by using a further coarse grid, say $\mathcal{T}_{4h}$. We continue this process till we reach a very coarse mesh for which an exact solver can be easily used. The resulting algorithm is a typical multigrid algorithm which we shall discuss in more details in the next section.

Multilevel coarse grid corrections and a multigrid method. To describe a multigrid algorithm, we first need to have a multiple level of grids, say $\mathcal{T}_k$ with $k = 1 : J$ and $\mathcal{T}_J = \mathcal{T}_h$ being the finest mesh. There are many ways to obtain multiple level of grids and one simple definition of the grid points in $\mathcal{T}_k$ is as follows:

$$x_i^k = \frac{i}{2^k}, \quad i = 1, 2, \cdots, n_k, k = 1, 2, \cdots, J$$

where $N_k = 2^k - 1$. Note that $\mathcal{T}_k$ can be viewed as being obtained by adding midpoints of the subintervals in $\mathcal{T}_{k-1}$. For each k the set of above nodes will be denoted by $\mathcal{N}_k$.

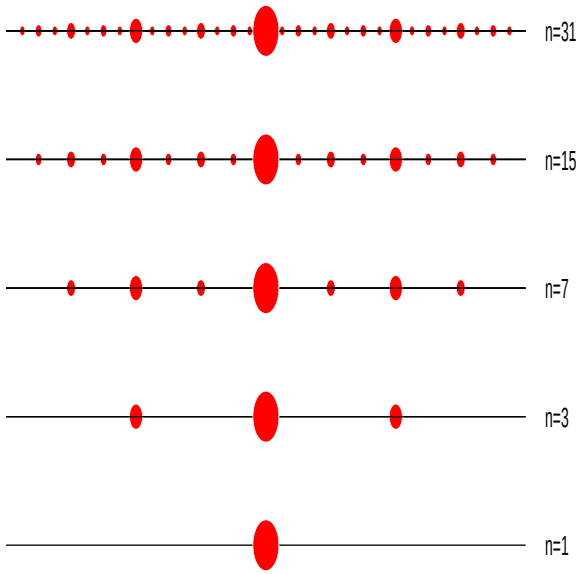

FIG. 5. *Multiple grids in one dimension*

With our previous experiences in two-grid method, the description of a multigrid method is not very difficult. In fact, the multiple level grids are treated by treating each two consecutive grids, say $\mathcal{T}_k$ versus $\mathcal{T}_{k-1}$. If we think $\mathcal{T}_k$ versus $\mathcal{T}_{k-1}$ like $\mathcal{T}_h$ and $\mathcal{T}_{2h}$, then there is not much new in the multigrid setting.

Let us give some details the definition of the restriction and prolongation matrices. The restriction matrix $I_k^{k-1} : R^{n_k} \mapsto R^{n_{k-1}}$ can be defined, as in (3.12), by

$$\gamma^{k-1} = I_k^{k-1}\gamma^k : \ \gamma_i^{k-1} = \frac{1}{2}\gamma_{2i-1}^k + \gamma_{2i}^k + \frac{1}{2}\gamma_{2i+1}^k. \tag{3.16}$$

The prolongation matrix $I_{k-1}^k : R^{n_{k-1}} \mapsto R^{n_k}$ can be defined as in (3.13) by

$$\epsilon^k = I_{k-1}^k \epsilon^{k-1} : \ \epsilon_{2i}^k = \epsilon_i^{k-1}, \epsilon_{2i+1}^k = \frac{1}{2}(\epsilon_i^{k-1} + \epsilon_{i+1}^{k-1}), \ i = 1 : N_{k-1}. \tag{3.17}$$

With the restriction and prolongation matrices in hands, we can now present a multilevel version of the earlier two-grid algorithm. As mentioned before, the idea is to repeat this two grid process for the coarse grid by using an even coarser grid. The resulting algorithm is just a desired multigrid algorithm. It is not hard to see that this kind of algorithm can be defined recursively.

Like the matrix B_h introduced in Algorithm 3.15, we would like to present the multigrid method by defining a sequence of matrices $B_k : R^{n_k} \mapsto R^{n_k}$ which is an approximate of A_k.

ALGORITHM 3.18 (A MULTIGRID ALGORITHM). *Define $B_1 = A_1^{-1}$. Assume $B_{k-1} : R^{n_{k-1}} \mapsto R^{n_{k-1}}$ is defined. To define $B_k : R^{n_k} \mapsto R^{n_k}$, we consider to approximately solve the equation $A_k\mu = \beta^k$ for $\beta^k \in R^{n_k}$.*

1. *Fine grid smoothing: with $\mu^0 = 0$,*

$$\mu^l = \mu^{l-1} + \sigma_k^{-1}(\beta^k - A_k\mu^{l-1}), \ l = 1 : m.$$

2. *Coarse grid correction: form the residual $\gamma^k = \beta^k - A_k\mu^m$ and solve the coarse grid residual equation $A_{k-1}\epsilon = I_k^{k-1}\gamma^k$ by using B_{k-1}*

$$\epsilon^{k-1} = B_{k-1}I_k^{k-1}\gamma^k.$$

3. *Define: $B_k\beta^k = \mu^m + I_{k-1}^k\epsilon^{k-1}$.*

With the B_k defined above, a typical multigrid iteration can be obtained.

ALGORITHM 3.19.

$$\mu \leftarrow \mu^0, \ \textit{iterate } \mu \leftarrow \mu + B_J(\beta^J - A_J\mu) \textit{ till convergence},$$

where the action of B_J is computed by Algorithm 3.18.

Let us now give a numerical example to demonstrate the efficiency of the above multigrid method.

As we can see from the numerical examples, the multigrid algorithm converges very fast. In fact, its convergence rate is independent of the grid size. We will prove in the next subsection that there exists a constant $\delta \in (0,1)$ independent of h such that

$$\|\mu - \mu^k\|_{A_J} \leq \delta^k \|\mu - \mu^0\|_{A_J}$$

where μ^k is the $k-th$ iteration by the multigrid Algorithm 3.18.

Summary. In this subsection, we have tried to explain the main ideas of multigrid methodology by using a very simple two point boundary value problem with the simple finite difference discretization. Despite its simplicity, this model reflects most of the main features of more general and more complicated elliptic boundary value problems.

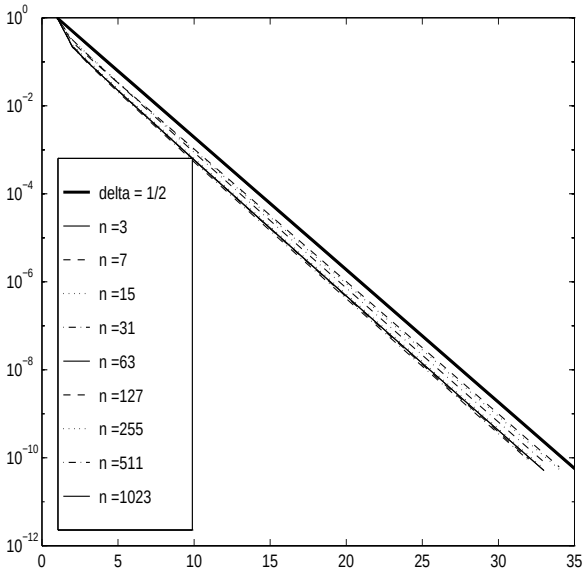

FIG. 6. *Convergence history of multilevel method for different number of unknowns (different values of h respectively). The thick line corresponds to damping factor* $1/2$. *Only one smoothing step is applied in the algorithm.*

Roughly speaking, a multigrid method is based on two important observations. First, a local relaxation method such as Richardson iteration can damp out non-smooth part of the error and after a few Richardson iterations, the residual become a relatively smooth vector. Secondly, a relatively smooth vector can be well approximated by a relatively coarser grid. Combining the above two observations, a local relaxation iterative scheme can then be speeded up by correcting the residual on coarse grid. Applying this idea recursively with a multiple level of grids, an efficient multigrid method is then obtained.

3.2. Why multigrid works?. In this subsection we give some simple mathematical analysis of some aspects of the multigrid method introduced in the previous subsection. This analysis should be helpful to have a better understanding why the multigrid idea works.

Why Richardson iteration has smoothing property?. Because of the extraordinary importance of this smoothing property, we shall now try to give some simple theoretical analysis. To do this, we make use of the eigenvalues and eigenvectors of the matrix A.

We recall that λ is an eigenvalue of A and and $\xi \in R^N \setminus \{0\}$ is a corresponding eigenvector if

$$A\xi = \lambda\xi.$$

Because of the special structure of A, all the N eigenvalues, λ_k, and the corresponding eigenvectors, $\xi^k = (\xi_j^k)$, of A can be obtained, for $1 \le k \le N$, as follows:

$$\lambda_k = \frac{4}{h}\sin^2\frac{k\pi}{2(N+1)}, \xi_j^k = \sin\frac{kj\pi}{N+1}(1 \le j \le N).$$

Indeed, the relation $A\xi^k = \lambda_k \xi^k$ can be verified by following elementary trigonometric identities:

$$-\sin\frac{k(j-1)\pi}{N+1} + 2\sin\frac{kj\pi}{N+1} - \sin\frac{k(j+1)\pi}{N+1} = 4\sin^2\frac{k\pi}{2(N+1)}\sin\frac{kj\pi}{N+1}$$

Actually, it is not very difficult to derive these formula directly (see appendix A).

To understand the behavior of these eigenvectors, let us take $N = 6$ and plot the linear interpolations of all these 6 eigenvectors. We immediately observe that each vector ξ^k corresponds to a given frequency, and larger k corresponds to higher frequency. As λ_k is increasing with respect to k, we can then say that a larger eigenvalue of A corresponds to a higher frequency eigenvector. From a numerical point of view, we say a relatively low frequency vector is relatively smoother whereas a high frequency vector is nonsmooth.

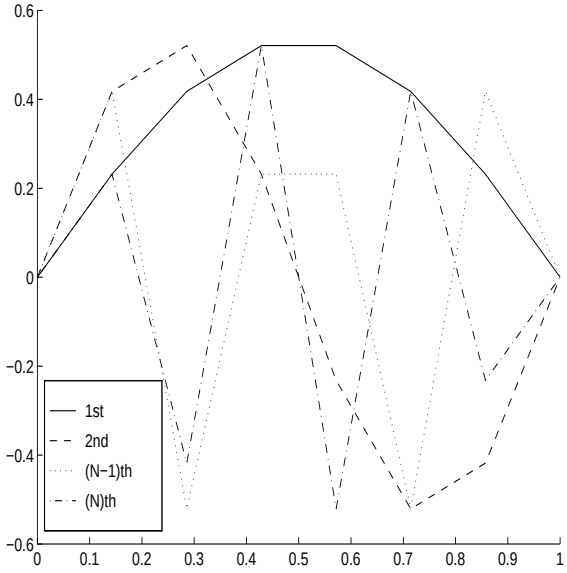

FIG. 7. *The eigenvectors*

We note that the set of eigenvectors $\xi^k = (\xi_j^k)$ forms an orthogonal basis of R^N. (This fact can be checked directly, or it also follows from the fact that the matrix A is symmetric and has N distinctive eigenvalues.) Therefore, any $\xi \in R^N$ can be expanded in terms of these eigenvectors:

$$\xi = \sum_{k=1}^{N} \alpha_k \xi^k.$$

This type of expansion is often called discrete Fourier expansion. The smoothness of the vector ξ has a lot to do with the relative size of the coefficients α_k. To see this numerically, let us again take $N = 4$ and consider the following two vectors

$$\xi = \sum_{k=1}^{4} 2^{1-k}\xi^k, \quad \eta = \sum_{k=1}^{4} 2^{k-4}\xi^k.$$

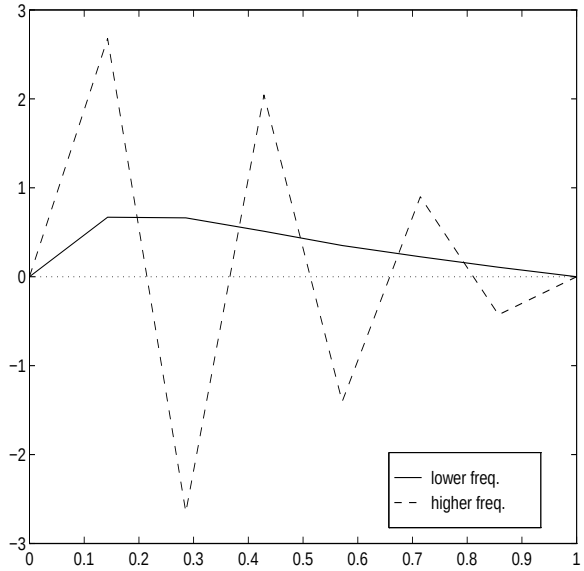

FIG. 8. *Plots of ξ and η. ξ-solid line; η dashed line*

The first vector ξ has larger coefficients in front of lower frequencies whereas the second vector η has larger coefficients in front of higher frequencies. From Figure 8, it is easy to see how the smoothness of a vector depends on the relative size of its Fourier coefficients. We conclude that, in general, a vector with relatively small Fourier coefficients in front of the higher frequencies is relatively smooth and conversely, a vector with relatively large Fourier coefficients in front of the higher frequencies is relatively rough or nonsmooth.

Fourier analysis for the Richardson method. Our earlier numerical experiments indicate that the Richardson method has a smoothing property. Based on our understanding of the relation between the smoothness and the size of Fourier coefficients, we can imagine that this smoothing property can be analyzed using the discrete Fourier expansion.

Let μ be the exact solution of (3.3) and μ^l the result of $l-th$ iteration from the damped Jacobi (3.4). Then

$$\mu - \mu^l = (1 - \sigma^{-1}A)(\mu - \mu^{l-1}) = \ldots = (1 - \sigma^{-1}A)^l(\mu - \mu^0).$$

Consider the Fourier expansion of the initial error:

$$\mu - \mu^0 = \sum_{k=1}^{N} \alpha_k \xi^k.$$

Then

$$\mu - \mu^m = \sum_{k=1}^{N} (1 - \lambda_k/\sigma)^m \xi^k = \sum_{k=1}^{N} \alpha_k^m \xi^k$$

where

$$\alpha_k^m = \left(1 - \sin^2 \frac{k\pi}{2(N+1)}\right)^m \alpha_k$$

Note that

$$\alpha_k^m = \alpha_k \sin^{2m} \frac{N-k+1}{N+1}\frac{\pi}{2} = \alpha_k \left(\frac{N-k+1}{N+1}\frac{\pi}{2}\right)^{2m}$$

which, for k close to N, approaches to 0 very rapidly when $m \to \infty$. This means that high frequency components get damped very quickly.

This simple analysis clearly justifies the smoothing property that has been observed by numerical experiments.

Other smoothers. You may ask if it is just lucky to choose $\sigma = 4/h$ so that the corresponding Richardson method has this smoothing property. There is certain truth to this, since a good choice of damping parameter is not always easy to come by. But it is not so difficult to convince yourself – by the same analysis – that this smoothing property still preserves qualitatively for other choice of σ as long as

$$\sigma \geq \frac{2+c_0}{h}.$$

where c_0 is constant independent on N.

Note that the above range of σ excludes the case that $\sigma = 2/h$ which corresponds to the so-called ordinary Jacobi method:

$$\text{For } j = 1:N,\ \mu_j^l = \mu_j^{l-1} + \frac{h}{2}\left(\beta_j - \frac{-\mu_{j-1}^{l-1} + 2\mu_j^{l-1} - \mu_{j-1}^{l-1}}{h}\right).$$

This Jacobi method, however, does not have a smoothing property. To see this, with m Jacobi iteration, the Fourier coefficient in front of, for example, the highest frequency is as follows:

$$\begin{aligned} \alpha_N^m &= \left(1 - 2\sin^2\frac{N\pi}{2(N+1)}\right)^m \alpha_N = \cos^m\frac{N\pi}{N+1}\alpha_N \\ &\sim (-1)^m\left(1 - \frac{\pi^2}{(N+1)^2}\right)^m \alpha_N \end{aligned}$$

which, if $\alpha_N \neq 0$, can not practically get very small for large N.

A parameter-free and yet more efficient smoother is the so-called Gauss-Seidel iteration. This method modifies the Jacobi method in such a way that makes use of the most updated values of each component:

$$\text{For } j = 1:N,\ \mu_j^l = \mu_j^{l-1} + \frac{h}{2}\left(\beta_j - \frac{-\mu_{j-1}^{l} + 2\mu_j^{l-1} - \mu_{j-1}^{l-1}}{h}\right).$$

Comparing it with Jacobi method, the difference lies in the term μ_{j-1}^l in which the superscript l indicates the most updated value of μ_j is used.

It is not hard to imagine that this method has a better convergence property than the Jacobi method. It is also very easy to be convinced by numerical experiments that this is actually a better smoother than the Richardson method, although it is a little bit more complicated to analyze theoretically.

An intuitive discussion. Both the Richardson Jacobi and Gauss-Seidel methods are oftentimes called *local relaxation* methods. This name refers to the fact that what both of these algorithms do are just trying to correct the residual vector locally at one nodal point at a time (recall that $\mu_j \approx u(x_j)$). This local relaxation procedure is then effective to the error components that are local in nature. Incidently, the nonsmooth or high frequency component which oscillates across one or few grid points have a strong local feature. Therefore, it is not surprising the both Richardson and Gauss-Seidel iteration can damp out these nonsmooth components more easily. These methods are very inefficient for relatively smoother components in the error since a smoother function is more globally related in nature.

The finite element method. The finite element method for the approximation of (3.1) is based on an equivalent variational formulation of (3.1). To derive such a formulation, let us introduce the following functional space:

$$\mathcal{V} = \{v : \ v \text{ is continuous and piecewise differentiable}, v(0) = v(1) = 0\}.$$

Multiplying any function $v \in \mathcal{V}$ on both hand sides of (3.1) and integrating by parts, we get

$$\int_0^1 f v dx = -\int_0^1 u'' v dx = \int_0^1 u' v' dx \tag{3.20}$$

Denote

$$A(u, v) = \int_0^1 u' v' dx.$$

We then have

$$A(u, v) = (f, v) \quad \forall\, v \in \mathcal{V}. \tag{3.21}$$

Like the finite difference discretization, a finite element discretization is also based on a partition of the interval $(0, 1)$. Given any positive integer N, we consider the partition of the interval $(0, 1)$ given by (3.2). Associated with the partition $\mathcal{T}_h$, we define a linear finite element space is defined as follows

$$\mathcal{V}_h = \{v : \ v \text{ is continuous and piecewise linear w.r.t. } \mathcal{T}_h,\ v(0) = v(1) = 0\}.$$

Namely $\mathcal{V}_h$ consists of piecewise linear continuous functions. An example function of $\mathcal{V}_h$ is shown in the Figure 9.

Apparently $\mathcal{V}_h$ is a linear vector space of dimension N and each internal nodal point corresponds to a degree of freedom. $\mathcal{V}_h$ has a natural basis, known as nodal basis, which is given by, for $i = 1, 2, \cdots, N$

$$\phi_i(x) = \begin{cases} \frac{x - x_{i-1}}{h}, & x \in [x_{i-1}, x_i]; \\ \frac{x_{i+1} - x}{h}, & x \in [x_i, x_{i+1}]; \\ 0 & \text{elsewhere.} \end{cases}$$

Note that $\phi_i(x_j) = \delta_{ij}$ and for any $v \in \mathcal{V}_h$,

$$v(x) = \sum_{i=1}^{N} v(x_i)\phi_i(x).$$

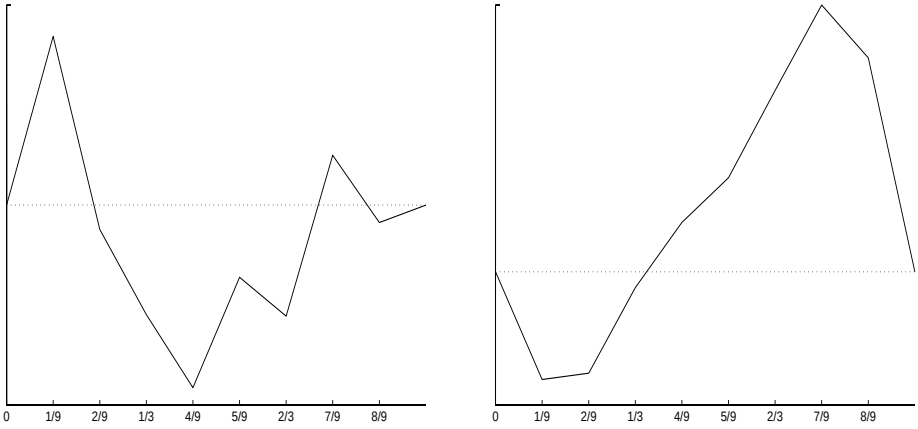

FIG. 9. *Typical finite element functions.*

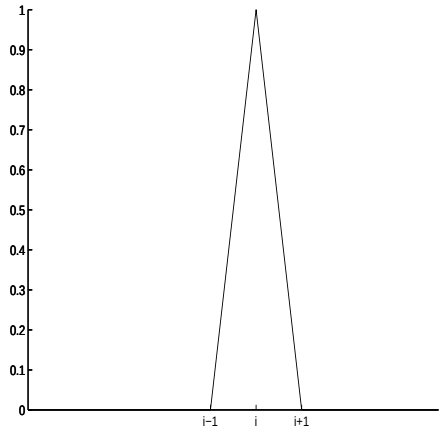

FIG. 10. *Finite element basis function* ϕ_i.

The finite element approximation of (3.3) is defined as follows

$$\left\{ \begin{array}{c} \text{Find } u_h \in \mathcal{V}_h, \quad \text{such that} \\ A(u_h, v_h) = (f, v_h) \quad \forall\ v_h \in \mathcal{V}_h. \end{array} \right. \tag{3.22}$$

We write

$$u_h(x) = \sum_{i=1}^{N} \mu_i \phi_i(x) \text{ with } \mu_i = u_h(x_i).$$

The equation (3.22) is then equivalent to

$$\sum_{i=1}^{N} \mu_i A(\phi_i, \phi_j) = (f, \phi_j), \quad j = 1, 2, \cdots, N$$

or

$$A\mu = b. \tag{3.23}$$

Here $A = (A(\phi_i, \phi_j))_{N\times N}$ and $b = ((f, \phi_j))_{1\times N}$.

The matrix A is known as stiffness matrix under the nodal basis (ϕ_i).

A direct calculation shows that

$$\begin{aligned}\int_0^1 \phi'_{j-1}\phi'_j &= \int_0^1 \phi'_j\phi'_{j+1} = -\frac{1}{h}, \quad 1 < j < N \\ \int_0^1 (\phi'_j)^2 &= \frac{2}{h}, \quad 1 \le j \le N.\end{aligned}$$

Therefore, the finite element equation is reduced to

$$\frac{-\mu_{j-1} + 2\mu_j - \mu_{j+1}}{h} = f_j, \quad 1 \le j \le N, \tag{3.24}$$

where $f_j = \int_0^1 f\phi_j$.

We conclude that the stiffness matrix of the linear finite element approximation on a uniform grid, after proper scaling, is the same as the stiffness matrix of the finite difference scheme.

In this case, the stiffness matrix scaled by h is tridiagonal

$$A = \text{diag}(-1, 2, -1).$$

Nested multilevel finite element subspaces. One convenient thing in finite element setting is that we can naturally talk about subspaces of piecewise linear functions. Associated with the multilevel grids $\mathcal{T}_k$ defined in §3.1, we can naturally define the following multilevel subspaces:

$$V_k = \{v : \ v \text{ is continuous and piecewise linear w.r.t. } \mathcal{T}_k,\ v(0) = v(1) = 0\}.$$

Since a piecewise linear function with respect to a coarse grid is naturally a piecewise linear function with respect to a finer grid, we have the following nested relation:

$$V_1 \subset V_2 \subset \ldots \subset V_J.$$

Let us now discuss how to restrict a fine grid finite element equation to a coarse grid finite element equation. Consider the following linear algebraic system on level k:

$$A_k\epsilon^k = \gamma^k$$

This equation is equivalent to the following equation in variational form:

$$A(\epsilon^k, v_k) = (g, v_k) \quad \forall\, v_k \in V_k. \tag{3.25}$$

where

$$\epsilon^k(x) = \sum_{i=1}^{n_k} \epsilon_i\phi_i^k(x)$$

and g is the unique function in V_k satisfying

$$\gamma_i^k = (g, \phi_i^k).$$

The most natural restriction of the variational problem on V_{k-1} is the following

$$A(\epsilon^{k-1}, v_{k-1}) = (g, v_{k-1}) \quad \forall\, v_{k-1} \in V_{k-1}.$$

The algebraic system resulting from this is as follows:

$$A_{k-1}\epsilon^{k-1} = \gamma^{k-1}$$

where

$$\gamma_i^{k-1} = (g, \phi_i^{k-1}).$$

Using the following obvious relation

$$\phi_i^{k-1} = \frac{1}{2}\phi_{2i-1}^k + \phi_{2i}^k + \frac{1}{2}\phi_{2i+1}^k,$$

we then get

$$\gamma_i^{k-1} = \frac{1}{2}\gamma_{2i-1}^k + \gamma_{2i}^k + \frac{1}{2}\gamma_{2i+1}^k.$$

This is the exact relationship that defines the restriction matrix.

Multigrid analysis. In order to fully justify the efficiency of the multigrid method, we will now give a complete proof on the uniform convergence of the multigrid method that we discussed. In general, multigrid convergence analysis is not at all easy. But since we are now dealing with a very special one dimensional problem, it is possible for us to give a rather simple analysis.

The proof we shall present only consists of simple algebraic manipulations. Thus the proof is quite easy to follow logically. But we would like to point out that these algebraic arguments are mostly motivated from analytic considerations. With m throughout this section stays we denote the number of smoothing steps in Algorithm 3.18

Error equation. Let μ be the exact solution. Then $\beta^k = A_k\mu$ and, from the fine grid smoothing step, we have

$$\mu - \mu^l = (I_k - \sigma_k^{-1}A_k)(\mu - \mu^{l-1}) = \ldots = (I_k - \sigma_k^{-1}A_k)^l\mu.$$

where $I_k \in R^{n_k \times n_k}$ is the identity matrix. By the definition of B_k

$$\begin{aligned}
(I_k - B_kA_k)\mu &= \mu - B_k\beta^k = \mu - \mu^m - I_{k-1}^k\epsilon^{k-1} \\
&= (\mu - \mu^m) - I_{k-1}^k B_{k-1} I_k^{k-1} A_k(\mu - \mu^m) \\
&= (I_k - I_{k-1}^k B_{k-1} I_k^{k-1} A_k)(\mu - \mu^m) \\
&= (I_k - I_{k-1}^k B_{k-1} I_k^{k-1} A_k)(I_k - \sigma_k^{-1}A_k)^m\mu
\end{aligned}$$

Since the above relation holds for any μ, we arrive at the following matrix identity:

$$I_k - B_kA_k = (I_k - I_{k-1}^k B_{k-1} I_k^{k-1} A_k)K_k^m$$

where $K_k = I_k - \sigma_k^{-1}A_k$.

We shall proceed to manipulate the above error equation into a more convenient format. Let us first note that the following basic identity holds:

$$A_{k-1} = I_k^{k-1} A_k I_{k-1}^k.$$

This identity can of course be verified by writing out the matrix multiplication. Another way to check it is to use finite element definition of A_{k-1} and then write the coarse grid finite element basis function in terms of fine grid basis function. Since at least one of these two verifications is straightforward, we leave the details to the readers.

Let us write

$$I_k - B_k A_k = (I_k - I_{k-1}^k P_{k-1} + I_{k-1}^k (I_{k-1} - B_{k-1} A_{k-1}) P_{k-1}) K_k^m.$$

where

$$P_{k-1} = A_{k-1}^{-1} I_k^{k-1} A_k.$$

Note that $P_{k-1} : R^{n_k} \mapsto R^{n_{k-1}}$ and it is in some sense a projection matrix with respect to the energy inner project. In fact, the following relation holds for $\mu_k \in R^{n_k}$ and $\nu_{k-1} \in R^{n_{k-1}}$

$$(3.26) \quad (\mu_k, I_{k-1}^k \nu_{k-1})_{A_k} = (P_{k-1}\mu_k, \nu_{k-1})_{A_{k-1}} = (I_{k-1}^k P_{k-1}\mu_k, I_{k-1}^k \nu_{k-1})_{A_k}.$$

Using the above identities, we derive the following identity:

$$\|(I_k - B_k A_k)\mu\|_{A_k}^2 = \|(I_k - I_{k-1}^k P_{k-1})\tilde{\mu}\|_{A_k}^2 + \|(I_{k-1} - B_{k-1}A_{k-1})P_{k-1}\tilde{\mu}\|_{A_{k-1}}^2$$

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

The following lemma concerns an approximation result which shows how a fine grid vector can be approximated by a coarse grid vector.

LEMMA 3.1. *For all $\nu \in R^{n_k}$ the following inequality holds:*

$$\|(I_k - I_{k-1}^k P_{k-1})\nu\|_{A_k}^2 \leq \frac{1}{2} h_k \|A_k \nu\|^2.$$

Proof. We first claim that the projection matrix P_{k-1} can be characterized by the following relation

$$(3.27) \qquad (P_{k-1}\nu)_{2i} = \nu_{2i}, (P_{k-1}\nu)_{2i-1} = \frac{1}{2}(\nu_{2i-2} + \nu_{2i}), i = 1 : N_{k-1}.$$

These relations follow from the equality:

$$I_k^{k-1} A_k = A_{k-1} P_{k-1}.$$

which can be directly verified. The proof is then terminated by the next chain of simple relations:

$$\begin{aligned}
\|(I_k - I_{k-1}^k P_{k-1})\nu\|_{A_k}^2 &= \frac{2}{h_k} \sum_{i=1}^{N_{k-1}} \left(\nu_{2i-1} - \frac{1}{2}(\nu_{2i-2} + \nu_{2i})\right)^2 \\
&= \frac{1}{2h_k} \sum_{i=1}^{N_{k-1}} \left(2\nu_{2i-1} - (\nu_{2i-2} + \nu_{2i})\right)^2 \\
&\leq \frac{1}{2h_k} \sum_{j=1}^{N_k} \left(2\nu_j - (\nu_{j-1} + \nu_{j+1})\right)^2 \\
&= \frac{h_k}{2} \|A_k \nu\|^2.
\end{aligned}$$

□ We would like to point out that the relation (5.18) is true only for the very special equation (3.1). For more general equation, this relation is no longer valid, but the estimate in the lemma still holds (with a different constant in place of $\frac{1}{2}$ in the right hand side of the inequality) and the proof is a little bit more complicate.

The following lemma gives a quantitative estimate on how a fine grid vector smoothed m times by local relaxation can be approximated by a coarse grid vector. It is worth nothing that same norms are used in both hands of the estimate.

LEMMA 3.2. *Let $\nu \in R^{n_k}$ and $K_k = I_k - \sigma_k^{-1} A_k$. Then the following inequality holds:*

$$\|(I - I_{k-1}^k P_{k-1}) K_k^m \nu\|_{A_k}^2 \leq \frac{1}{m}(\|\nu\|_{A_k}^2 - \|K_k^m \nu\|_{A_k}^2).$$

Proof. Note that $\sigma(K_k) \subset (0,1)$ and $(K_k \cdot, \cdot)_A = (\cdot, K_k \cdot)_A$. Taking into account these observations we get:

$$\begin{aligned}
\|(I - I_{k-1}^k P_{k-1}) K_k^m \nu\|_{A_k}^2 &\leq \frac{h_k}{2} \|A_k K_k^m \nu\|^2 \\
&= 2(\sigma_k^{-1} A_k K_k^m \nu, A_k K_k^m \nu) = 2((I - K_k) K_k^{2m} \nu, \nu)_{A_k} \\
&\leq \frac{1}{m} \sum_{i=0}^{2m-1} ((I - K_k) K_k^i \nu, \nu)_{A_k} \\
&= \frac{1}{m}(((I - K_k^{2m})\nu, \nu)_{A_k}
\end{aligned}$$

□

THEOREM 3.3. *Let the operator B_k be defined as in algorithm 3.18. Then there exists a real number $\delta \in (0,1)$ depending only on the number of smoothing steps m, such that the following inequality holds*

$$\|(I - B_k A_k)\nu\|_{A_k}^2 \leq \frac{1}{m+1} \|\nu\|_{A_k}^2 \quad \forall\, \nu \in R^{N_k},$$

for every $k \in \{1, 2, \ldots, J\}$.

Proof. The proof is done by induction. For $k = 1$ it trivially follows from the choice $B_1 = A_1^{-1}$. Let us assume that this inequality is true for $k-1$. Then taking $\tilde{\nu}_m = K_k^m \nu$ we have:

$$\begin{aligned}
\|(I - B_k A_k)\nu\|_A^2 &\leq \|(I - P_{k-1}) v_m\|_{A_k}^2 + \delta \|P_{k-1} \tilde{\nu}\|_{A_{k-1}}^2 \\
&= (1 - \frac{1}{m+1}) \|(I - P_{k-1})\tilde{\nu}\|_{A_k}^2 + \frac{1}{m+1} \|\tilde{\nu}\|_{A_k}^2 \\
&\leq \frac{m}{m+1} \frac{1}{m} (\|\nu\|_A^2 - \|\tilde{\nu}\|_A^2) + \frac{1}{m+1} \|\tilde{\nu}\|_{A_k}^2 \\
&= \frac{1}{m+1} \|\nu\|_{A_k}^2
\end{aligned}$$

□

(3.28) Remark. Looking back at the convergence history plots Figure 4 and Figure 6 we observe that the numerical experiments show that the estimate obtained in the preceeding theorem is very sharp.

4. Iterative methods by subspace correction. Following Xu [33] (see also Bramble et al. [8, 7]), a general framework for 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 [33]. The main modification is that the subspace solvers here may not be symmetric. For related topics, we refer to Bramble [5].

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

$$\mathcal{V} = \sum_{i=0}^{J} \mathcal{V}_i. \tag{4.1}$$

This means that, for each $v \in \mathcal{V}$, there exist $v_i \in \mathcal{V}_i$ $(0 \le i \le J)$ such that $v = \sum_{i=0}^{J} v_i$. This representation of v may not be unique in general, namely (4.1) 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

$$(Q_i u, v_i) = (u, v_i), \quad (P_i u, v_i)_A = (u, v_i)_A, \quad u \in \mathcal{V}, v_i \in \mathcal{V}_i, \tag{4.2}$$

and

$$(A_i u_i, v_i) = (A u_i, v_i), \quad u_i, v_i \in \mathcal{V}_i. \tag{4.3}$$

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

$$A_i P_i = Q_i A. \tag{4.4}$$

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

$$A_i u_i = f_i \tag{4.5}$$

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 (4.5) 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

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

In general the subspace equation (4.5) will be solved approximately. To describe this, we introduce, for each i, another nonsingular operator $R_i : \mathcal{V}_i \mapsto \mathcal{V}_i$ that represents an

approximate inverse of A_i in a certain sense. Thus an approximate solution of (4.5) may be given by $\hat{u}_i = R_i f_i$.

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

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

where e^i is the i-th column of the identity matrix. For an 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 is the i-th component of $y \in \mathbb{R}^n$.

4.2. 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 the 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: $r^{\text{old}} = f - Au^{\text{old}}$. If $r^{\text{old}} = 0$ or is very small, we are done. Otherwise, we consider the residual equation:

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

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

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

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

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

and

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

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

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

An update of the approximation of u is obtained by

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

which can be written as

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

where

$$B = \sum_{i=0}^{J} R_i Q_i. \tag{4.7}$$

We have therefore

ALGORITHM 4.8. *Given $u_0 \in \mathcal{V}$, apply the iterative scheme (2.3) with B given in* (4.7).

(4.9) Example. With $\mathcal{V} = \mathbb{R}^n$ and the decomposition given by Example 4.1, the corresponding Algorithm 4.8 is just the Jacobi iterative method.

It is well known that the Jacobi method is not convergent for all SPD problems (see Theorem 2.3) hence Algorithm 4.8 is not always convergent. However the preconditioner obtained from this algorithm is of great importance. We note that the operator B given by (4.7) is SPD if each $R_i : \mathcal{V}_i \to \mathcal{V}_i$ is SPD.

ALGORITHM 4.10. *Apply the CG method to equation* (2.1), *with B defined by* (4.7) *as a preconditioner.*

(4.11) Example. The preconditioner B corresponding to Example 4.1 is

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

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 make the correction here in one subspace at a time by using the most updated approximation of u. More precisely, starting from $v^{-1} = u^{\text{old}}$ 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

ALGORITHM 4.12. *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

(4.13) Example. Corresponding to decomposition in Example 4.1, the Algorithm 4.12 is the Gauss–Seidel iteration.

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

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

where $1 = l_0 < l_1 < \cdots < l_{J+1} = n+1$. Then Algorithms 4.8, 4.10 and 4.12 are the block Jacobi method, block diagonal preconditioner and block Gauss–Seidel methods respectively.

Let $T_i = R_i Q_i A$. By (4.4), $T_i = R_i A_i P_i$. 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 4.12. 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

$$u - u^{k+1} = E_J(u - u^k), \tag{4.15}$$

where

$$E_J = (I - T_J)(I - T_{J-1}) \cdots (I - T_1)(I - T_0). \tag{4.16}$$

(4.17) 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 [27] 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 cannot converge when $\omega = 2$. In fact, as we shall see in Proposition 4.9 below, in this special case, the SSC method converges if and only if $0 < \omega < 2$.

The symmetrization of Algorithm 4.12 can also be implemented as follows.

ALGORITHM 4.18. *Given* $u^0 \in \mathcal{V}$, $v \leftarrow u^0$
for $k = 0, 1, \ldots$ till convergence
  for $i = 0 : J$ and $i = J : -1 : 0$ $v \leftarrow v + R_i Q_i (f - Av)$ endfor
endfor

The advantage of the symmetrized algorithm is that it can be used as a preconditioner. In fact, Algorithm 4.18 can be formulated in the form of (2.3) with operator B defined as follows: for $f \in \mathcal{V}$, let $Bf = u^1$ with u^1 obtained by Algorithm 4.18 applied to (2.1) with $u^0 = 0$.

(4.19) Colorization and parallelization of SSC iteration.

DEFINITION 4.20. *Associated with a given partition* (4.1), *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

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

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.

ALGORITHM 4.21 (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).

4.3. Convergence theory. The purpose of this section is to establish an abstract theory for algorithms described in previous sections.

In view of Theorem 2.1, it suffices to study Algorithms 4.10 and 4.12. Two fundamental theorems will be presented.

For the preconditioner of Algorithm 4.10, we need to estimate the condition number of

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

where B is defined by (4.7) 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 Algorithm 4.12, 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 (4.16). Applying this estimate to (4.15) yields

$$\|u - u^k\|_A \leq \delta^k \|u - u^0\|_A.$$

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

Parameter ω_1. The first constant, named ω_1, is the smallest constant satisfying

$$(T_i v, T_i v)_A \leq \omega_1 (T_i v, v)_A \quad \forall\, v \in \mathcal{V},\ 0 \leq i \leq J, \tag{4.22}$$

or equivalently

$$(v_i, A_i v_i) \leq \omega_1 (R_i^{-1} v_i, v_i) \quad \forall\, v \in \mathcal{V},\ 0 \leq i \leq J. \tag{4.23}$$

We assume that R_i is chosen in such a way that ω_1 is well defined. If all R_i are SPD, then ω_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 ω_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 later, 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 2.1) and in particular the iterative schemes given by all R_i are convergent schemes.

Parameters K_0 and $\bar{K}_0$. The parameter K_0 to be introduced now plays the most crucial rôle 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

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

and

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

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

$$\sum_{i=0}^J (R_i^{-1} v_i, v_i) \le K_0 (Av, v). \tag{4.24}$$

LEMMA 4.1. *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*

$$\sum_{i=0}^J (v_i, v_i)_A \le C_0 (v, v)_A; \tag{4.25}$$

then

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

and, if all R_i are SPD,

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

The above lemma is most useful in domain decomposition applications. A good upper bound for K_0 relies on a good lower bound for ω_0, which means that each subspace solver R_i should resolve the whole range of the spectrum of A_i. In other words, the subspace problems should be very well solved or preconditioned.

The constant C_0 in (4.25) only depends on the partition (decomposition) of the space and it is sometimes called the *partition constant*.

LEMMA 4.2. *Assume, for any $v \in \mathcal{V}$, that there is a decomposition $v = \sum_{i=0}^{J} v_i$ with $v_i \in \mathcal{V}_i$ satisfying*

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

then

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

and, if all R_i are SPD,

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

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 other words, each subspace solver R_i should be spectrally equivalent to $(\rho(A_i))^{-1}$.

Parameters K_1 and $\bar{K}_1$. These parameters measure the interaction among subspaces together with the subspace solvers.

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

$$\epsilon_{ij}^2 = \rho(P_j T_i P_j)/\omega_1 \quad \text{and} \quad \epsilon_{ji} = \epsilon_{ij}, \quad \epsilon_{ii} = 1. \tag{4.26}$$

We also define $\bar{\epsilon}_{ij} \in (0, 1]$, for $j < i$, by

$$\bar{\epsilon}_{ij}^2 = \rho(P_j \bar{T}_i P_j) \quad \text{and} \quad \epsilon_{ji} = \epsilon_{ij}, \quad \epsilon_{ii} = 1, \tag{4.27}$$

where with $\bar{R}_i$ being the symmetrization of R_i (see eqn. (2.5)),

$$\bar{T}_i = \bar{R}_i A_i P_i. \tag{4.28}$$

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

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

LEMMA 4.3. *If each R_i is SPD, then*

$$(T_i u, T_j v)_A \leq \omega_1 \epsilon_{ij} (T_i u, u)_A^{\frac{1}{2}} (T_j v, v)_A^{\frac{1}{2}} \quad \forall\, u, v \in \mathcal{V}; \tag{4.29}$$

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

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

□

(4.30) Remark. Clearly $\epsilon_{ij} \le 1$ and $\epsilon_{ij} = 0$ if $P_i P_j = 0$. If $\epsilon_{ij} < 1$, the inequality (4.29) is often known as the *strengthened Cauchy–Schwarz inequality.*

DEFINITION 4.31.

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

and

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

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

LEMMA 4.4. *The parameter K_1 admits the following estimates:*

1. $K_1 \le J + 1$;
2. $K_1 \le 1 + \rho((\epsilon_{ij})_{i,j=1:J}) \le 1 + \max\limits_{1 \le i \le J} \sum\limits_{j=1}^{n} \epsilon_{ij}$;
3. *if $\epsilon_{ij} \lesssim \gamma^{|i-j|}$ or $\bar{\epsilon}_{ij} \lesssim \gamma^{|i-j|}$ for some $\gamma \in (0,1)$, then*

$$\hat{K}_1 \lesssim \frac{1}{1-\gamma} \quad \text{or} \quad \bar{K}_1 \lesssim \frac{1}{1-\gamma}.$$

LEMMA 4.5.

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

If each R_i is SPD, then

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

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

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

4.3.2. Convergence theory. With the parameters ω_1, K_0 and K_1 introduced above, the convergence estimates for the PSC and SSC methods can be neatly presented. The analysis for the 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.

LEMMA 4.6. *Assume that all R_i are SPD. The PSC preconditioner B given by* (4.7) *satisfies*

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

Proof. If $v = \sum_{i=0}^{J} v_i$ is a decomposition that satisfies (4.24), 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

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

Consequently

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

This implies that $\lambda_{\min}(BA) \ge K_0^{-1}$.

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. □

(4.32) Remark. It is easy to see from the above proof that the following slightly more general identity also holds:

$$(B^{-1} v, v) = \inf_{\substack{v_i \in \mathcal{V}_i \\ \sum v_i = v}} \sum_i (R_i^{-1} v_i, v_i) \tag{4.33}$$

which apparently implies Lemma 4.6.

THEOREM 4.7. *Assume all R_i are SPDE. The PSC preconditioner B given by* (4.7) *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 4.1 and 4.2,

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

Proof. By Lemma 4.5,

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

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

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

LEMMA 4.8. *Denote* $E_{-1} = I$ *and for* $0 \leq i \leq J$,

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

Then

$$I - E_i = \sum_{j=0}^{i} T_j E_{j-1}, \tag{4.34}$$

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

$$\|v\|_A^2 - \|E_J v\|_A^2 = \sum_{i=0}^{J} (\bar{T}_i E_{i-1} v, E_{i-1} v)_A \tag{4.35}$$

where $\bar{T}_i$ *is given by* (4.28). *Furthermore if each* R_i *is symmetric then*

$$\|v\|_A^2 - \|E_J v\|_A^2 \geq (2 - \omega_1) \sum_{i=0}^{J} (T_i E_{i-1} v, E_{i-1} v)_A. \tag{4.36}$$

Proof. The identity (4.34) follows immediately from the trivial identity $E_{i-1} - E_i = T_i E_{i-1}$. Similarly 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 (4.35). The estimate (4.36) follows by combining (4.35) and (2.13). □

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 of the related SSS method is that $0 < \omega < 2$. In fact, as for the SOR method, it is not hard to see that this condition is also sufficient for the convergence (see Corollary 4.12 below). Thus, we have the following simple generalization of a classic result for the SOR method (see also Remark 4.2).

PROPOSITION 4.9. *The SSC method with* $R_i = \omega A_i^{-1}$ *for each* i *converges if and only if* $0 < \omega_1 < 2$.

LEMMA 4.10. *Assume that* $\omega_1 < 2$. *If each* R_i *is SPD, then*

$$\sum_{i=0}^{J} (T_i v, v)_A \leq (1 + K_1)^2 \sum_{i=0}^{J} (T_i E_{i-1} v, E_{i-1} v)_A \quad \forall\, v \in \mathcal{V}, \tag{4.37}$$

and in general

$$\sum_{i=0}^{J} (\bar{T}_i v, v)_A \leq \left(1 + \sqrt{\frac{\omega_1}{2 - \omega_1}} (\bar{K}_1 - 1)\right)^2 \sum_{j=0}^{J} (\bar{T}_j E_{j-1} v, E_{j-1} v)_A. \tag{4.38}$$

Proof. By (4.34)

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

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 4.5,

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

Combining these three formulae then leads to (4.37) and hence completes the proof for (4.38).

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 cancelling the common factor and using the following inequalities (see (2.10) and (2.9)):

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

the estimate (4.38) then follows easily. □

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

THEOREM 4.11. *Assume that $\omega_1 < 2$. If each R_i is SPD, then the iterator E_J (given by (4.16)) for the Algorithm 4.12 satisfies*

$$\|E_J\|_A^2 \le 1 - \frac{2 - \omega_1}{K_0 (1 + \omega_1 (K_1 - 1))^2}; \tag{4.39}$$

and, in general,

$$\|E_J\|_A^2 \le 1 - \frac{2 - \omega_1}{\bar{K}_0 (\sqrt{2 - \omega_1} + \sqrt{\omega_1} (\bar{K}_1 - 1))^2} \tag{4.40}$$

Proof. The estimate in (4.39) is obviously equivalent to

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

Estimate (4.39) then follows by combining (4.37) with (4.35) and (2.10).

The second estimate (4.40) then follows by combining (4.38) with the fact that $\lambda_{\min}(\sum_i \bar{T}_i) = \bar{K}_0^{-1}$ (similar to Lemma 4.6). ∎

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

COROLLARY 4.12. *A sufficient condition for the convergence of the SSC method is that*

$$\omega_1 < 2. \tag{4.41}$$

The condition (4.41) is also necessary in some sense; see Proposition 4.9.

(4.42) Remark. Note that the convergence estimate in Theorem 4.11 is independent of the order in which Algorithm 4.12 is executed. Namely, if we shuffle the order in the decomposition (4.1), the corresponding estimate in Theorem 4.11 remains unchanged.

THEOREM 4.13. *Under the assumptions in Lemma 4.1,*

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

and

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

THEOREM 4.14. *Under the assumptions in Lemma 4.2,*

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

and

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

4.4. 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 §2.3.

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 (4.7), we rewrite it in a slightly different form

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

Applying (2.17) 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

$$\mathcal{B} = \sum_{k=0}^{J} \mathcal{I}_k\mathcal{R}_k\mathcal{I}_k^t. \tag{4.43}$$

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 (2.4), we get

PROPOSITION 4.15. *The PSC preconditioner for the stiffness matrix $\mathcal{A}$ is given by* (4.43) *and* $\kappa(\mathcal{B}\mathcal{A}) = \kappa(BA)$.

Similarly, we can derive the algebraic representation of Algorithm 4.12 for solving (2.19).

ALGORITHM 4.44. *$\mu^0 \in \mathbb{R}^n$ is given. Assume that $\mu^k \in \mathbb{R}^n$ is obtained. Then μ^{k+1} is defined by*

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

for $i = 0 : J$.

5. Finite element approximations. In the following sections, we shall introduce the multigrid methods. Our presentations will be confined to 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.

5.1. A model problem and finite element discretization. We consider the boundary value problem:

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

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 (5.1) if and only if

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

where

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

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:

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

where

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

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

$$\|U\|_{H^{1+\alpha}(\Omega)} \leq C\|F\|_{H^{\alpha-1}(\Omega)}, \tag{5.3}$$

for the solution U of (5.2), where C is a constant depending on the domain Ω and the coefficient $a(x)$.

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

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

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

$$\|v\|_{L^\infty(\Omega)} \lesssim h^{-d/p}\|v\|_{L^p(\Omega)}, p \geq 1, \tag{5.4}$$

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

$$\|v\|_{H^{1+\sigma}(\Omega)} \lesssim h^{-\sigma}\|v\|_{H^{H^1(\Omega)}(\Omega)} \quad \sigma \in (0, \frac{1}{2}), \tag{5.6}$$

$$\|v\|_{H^s(\Omega)} \lesssim h^{t-s}\|v\|_{H^t(\Omega)} \quad s,t \in [0,1],\ t \leq s, \tag{5.7}$$

$$\|v\|_{L^\infty(\Omega)} \lesssim c_d(h)\|v\|_{H^1(\Omega)}, \tag{5.8}$$

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 (5.4) and (5.5) can be found, for example, in Ciarlet [18] and a proof of the discrete Sobolev inequality (5.8) can be found in Bramble and Xu [11]. A proof of (5.6) and (5.7) may be found in Bramble, Pasciak and Xu [10] and Xu [32].

THEOREM 5.1. *Assume that $P_h : H_0^1(\Omega) \mapsto \mathcal{V}$ is the Galerkin projection with respect to $A(\cdot,\cdot)$. Then*

$$\|(I - P_h)u\|_{H^{1-\alpha}(\Omega)} \lesssim h^\alpha \|u\|_{H^1(\Omega)} \quad \forall\, u \in H_0^1(\Omega), \tag{5.9}$$

and

$$\|(I - P_h)u\|_{H^1(\Omega)} \lesssim h^s \|u\|_{H^{1+s}(\Omega)}, \quad \forall\, u \in H_0^1(\Omega) \cap H^{1+s}(\Omega),\ 0 \leq s \leq \alpha \tag{5.10}$$

where α is as in (5.3).

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

$$\|v - Q_h v\| + h\|c - Q_h v\|_{H^1(\Omega)} \lesssim Ch\|v\|_{H^1(\Omega)}. \tag{5.11}$$

This estimate is well known; we refer to [32, 11] for a rigorous proof and related results.

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

$$\|Q_h v\|_{H^\sigma(\Omega)} \lesssim \|v\|_{H^\sigma(\Omega)} \quad \forall\, v \in H_0^1(\Omega). \tag{5.12}$$

and

$$\|v - Q_h v\|_{H^{1-\alpha}(\Omega)} \lesssim h^\alpha \|v\|_{H^1(\Omega)} \quad \forall\, v \in H_0^1(\Omega). \tag{5.13}$$

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

$$A(u, v) = (F, v) \quad \forall v \in \mathcal{V}. \tag{5.14}$$

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

$$(Au, v) = A(u, v), \quad u, v \in \mathcal{V}. \tag{5.15}$$

The equation (5.14) 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 (5.14) is reduced to solving an algebraic system (2.19) 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, for all $\nu \in \mathbb{R}^n$,

$$h^d|\nu|^2 \lesssim \nu^t \mathcal{A}\nu \lesssim h^{d-2}|\nu|^2 \quad \text{and} \quad h^d|\nu|^2 \lesssim \nu^t \mathcal{M}\nu \lesssim h^d|\nu|^2. \tag{5.16}$$

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

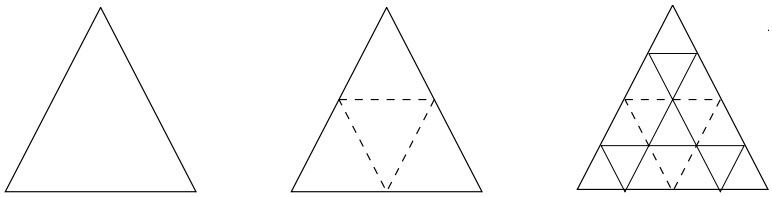

FIG. 11. *Typical multilevel grids.*

5.2. Finite element spaces on multiple levels. This section is to study the interaction between finite element spaces with different scales. We assume that Ω 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 τ_{k-1}^l can be written as a union of triangles of $\{\tau_k^i\}$ (see Fig. 11). 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

$$\mathcal{M}_0 \subset \mathcal{M}_1 \subset \ldots \subset \mathcal{M}_k \subset \ldots \subset \mathcal{M}_J. \tag{5.17}$$

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, for all $u \in H_0^1(\Omega), v_k \in \mathcal{M}_k$, by

$$(Q_k u, v_k) = (u, v_k), \quad (\nabla P_k u, \nabla v_k) = (\nabla u, \nabla v_k). \tag{5.18}$$

LEMMA 5.2. *Let R_k be any one of I_k, Q_k or P_k. Then*

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

LEMMA 5.3.

$$\|(I_k - I_{k-1})v\|^2 + h_k^2 \|I_k v\|_A^2 \lesssim c_d(k) h_k^2 \|v\|_A^2, \quad v \in \mathcal{V}, \tag{5.19}$$

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

5.3. Regularity and approximation property. Associated with each $\mathcal{M}_k$, we define, as in (4.3), $A_k : \mathcal{M}_k \mapsto \mathcal{M}_k$. The following result is instrumental in multigrid analysis.

THEOREM 5.4. *Assume α is as in* (5.3). *Then*

$$A((I - P_{k-1})u, u) \lesssim (\lambda_k^{-1} \|A_k u\|^2)^{\alpha} A(u, u)^{1-\alpha} \quad \forall\, u \in \mathcal{M}_k. \tag{5.20}$$

Proof. Let $u \in \mathcal{M}_k$. Applying the Cauchy–Schwarz 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

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

By Hölder's inequality,

$$\|A_k^{\frac{1+\alpha}{2}} u\| \leq \left(A(u,u)^{1-\alpha} \|A_k u\|^{2\alpha}\right)^{1/2}, \tag{5.21}$$

and, since $\lambda_k \lesssim h_k^{-2}$, Theorem 5.1 gives

$$\begin{aligned} \|(I-P_{k-1})u\|_{H^{1-\alpha}(\Omega)} &\lesssim h_k^{\alpha} \|(I-P_{k-1})u\|_{H^1(\Omega)} \\ &\eqsim \lambda_k^{-\frac{\alpha}{2}} A((I-P_{k-1})u, (I-P_{k-1})u)^{\frac{1}{2}} \\ &\eqsim \lambda_k^{-\frac{\alpha}{2}} A((I-P_{k-1})u, u)^{\frac{1}{2}}. \end{aligned}$$

The theorem follows by combining these inequalities. □

5.4. Strengthened Cauchy–Schwarz inequalities. These types of inequalities were used as assumptions in §4 (see eqn. (4.29)). Here we shall establish them for multilevel spaces.

LEMMA 5.5. *Let $i \geq j$; then*

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

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

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

A repeated application of the Cauchy–Schwarz inequality yields

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

□ The inequality in the previous lemma is a generalization of the strengthened Cauchy inequality for hierarchical basis functions in Yserentant [43]. Our proof is similar in nature to that in [43], but appears to be a little shorter and more straightforward.

LEMMA 5.6. *Let* $\mathcal{V}_i = (I_i - I_{i-1})\mathcal{V}$ *or* $\mathcal{V}_i = (Q_i - Q_{i-1})\mathcal{V}$*; then*

$$A(u,v) \lesssim \gamma^{|i-j|}\|u\|_A\|v\|_A \quad \forall\, u \in \mathcal{V}_i, v \in \mathcal{V}_j. \tag{5.22}$$

Proof. By (5.11), we have

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

The result then follows directly from Lemma 5.5. □

LEMMA 5.7. *Assume that* $T_k = R_kA_kP_k$ *and that* $R_k : \mathcal{M}_k \mapsto \mathcal{M}_k$ *satisfies*

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

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

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

Proof. If $i \le j$, an application of Lemma 5.5 yields

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

By the assumption on R_k,

$$\|T_jv\| = \|R_jA_jP_jv\| \lesssim h_j\|A_j^{\frac{1}{2}}P_jv\| \lesssim h_j\|v\|_A.$$

Consequently

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

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

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

□

5.5. An equivalent norm using multigrid splitting. If nested multilevel finite element spaces $\mathcal{M}_k$ are allowed to be 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.

THEOREM 5.8. *For all* $v \in H_0^1(\Omega)$*,*

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

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

$$\begin{aligned}
\|\tilde{Q}_k v_i\|^2_{H^1(\Omega)} &\lesssim h_k^{-2\alpha}\|\tilde{Q}_k v_i\|^2_{H^{1-\alpha}(\Omega)} \quad \text{(by inverse inequality (5.7))} \\
&\lesssim h_k^{-2\alpha}\|v_i\|^2_{H^{1-\alpha}(\Omega)} \quad \text{(by (5.12))} \\
&\lesssim h_k^{-2\alpha}h_i^{2\alpha}\|v_i\|^2_{H^1(\Omega)} \quad \text{(by (5.13)).}
\end{aligned}$$

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

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

To prove the other inequality, we use the strengthened Cauchy–Schwarz inequality and obtain (Lemma 5.5)

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

□

THEOREM 5.9. *For all* $v \in H^1_0(\Omega)$,

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

Proof. By the previous theorem, we obviously have

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

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 $Q_k - Q_{k-1}$. □

THEOREM 5.10. *For all $v \in \tilde{H}_0^s(\Omega)$ for $-1 \le s \le 1$, we have* [1]

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

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 the previous theorem, $\|v\|^2_{H^1(\Omega)} \eqsim (Bv, v)$. An application of operator interpolation then gives that $\|v\|^2_{H^s(\Omega)} = (B^s v, v)$ which implies the desired result. □

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

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

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

6. Overlapping domain decomposition methods. We shall now discuss our first major algorithm which is based on a domain decomposition with overlappings. As the finite element space $\mathcal{V}$ is defined on a triangulation of the domain Ω, a finite element space restricted to a subdomain of Ω can naturally be regarded as a subspace of $\mathcal{V}$ and hence a decomposition of the domain naturally leads to a decomposition of the finite element space. This is the main viewpoint for the algorithms described in this section.

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

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

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

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

LEMMA 6.1. *For the subspaces $\mathcal{V}_i$ $(0 \le i \le J)$, we have*

$$\mathcal{V} = \sum_{i=0}^{J} \mathcal{V}_i. \tag{6.1}$$

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

$$\sum_{i=0}^{J} A(v_i, v_i) \le C_0 A(v, v). \tag{6.2}$$

[1] $\tilde{H}_0^s(\Omega)$ is the interpolation space $[L_2(\Omega), H_0^1(\Omega)]_s$.

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

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

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

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

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

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

For this decomposition, we prove that (6.2) holds. For any $\tau \in \mathcal{T}_h$, note that

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

where $\bar{\theta}_{i,\tau}$ denotes the average of θ_i over τ. Let $w = v - Q_0 v$; by the inverse inequality (5.5),

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

It can easily be shown that

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

Consequently

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

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

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

and

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

For $i = 0$, we apply (5.11) and get

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

The desired result then follows. □

LEMMA 6.2. *The parameters K_0 and K_1 of subsection 4.3.1 satisfy*

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

Proof. The first estimate follows directly from Lemmas 4.2 and 6.1. To prove the second estimate, we define

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

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

$$|Z_i| \le n_0, \quad \forall\, 1 \le i \le J.$$

Note that if $P_i P_j \neq 0$ or $P_j P_i \neq 0$, then $j \in Z_i$. It therefore follows that $K_1 \le 1 + n_0$. □

(6.3) Remark. A slightly more careful analysis would give that the constant in (6.2) depends on the overlapping size, say δ_0, in the following way

$$C_0 = O(\frac{h_0}{\delta_0}).$$

6.2. Domain decomposition methods with overlappings. By Theorem 4.7 and Lemma 6.2, we get

THEOREM 6.3. *The SSC preconditioner B given by* (4.7) *associated with the decomposition* (6.1) *satisfies*

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

The proof of the above theorem follows from Theorem 4.7 and Lemma 6.2.

Combing Theorem 4.11 with Lemma 6.2, we also obtain

THEOREM 6.4. *The Algorithm 4.12 associated with the decomposition* (6.1) *satisfies*

$$\|E_J\|_A^2 \le 1 - \frac{\omega_0(2-\omega_1)}{C}, \tag{6.4}$$

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

We note that in our theory, the subdomain problems do not have to be solved exactly and only the spectrum of the inexact solvers matters. In the estimate (6.4), ω_0 should not be too small and ω_1 should stay away from 2. It is easy to see that one iteration of a V-cycle multigrid on each subdomain always satisfies this requirement.

As for the implementation of these domain decomposition methods, the algebraic formulations (4.43) and Algorithm 4.44 can be used. For example, if exact solvers are used in each subspace, the PSC preconditioner is

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

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

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

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

6.3. A recursive application of domain decomposition. As we shall see later, multigrid and domain decomposition methods can be viewed in the same mathematical framework. In fact, as we shall now demonstrate, a typical multigrid method may be derived by a recursive application of the overlapping domain decomposition method discussed above.

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

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

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

In view of our earlier discussions, the corresponding decomposition method without the coarser space is exactly the Gauss–Seidel method, which as we know is not very efficient (its convergence rate is easily shown to be $1 - O(h^2)$). The more interesting case is when a coarser space is introduced. The choice of such a coarse space is clear here, namely a subspace defined on a triangulation with characteristic mesh size being the same as the subdomains' size which is $2h$, therefore the coarse space $\mathcal{M}_{J-1}$ is the most natural candidate. But the space $\mathcal{M}_{J-1}$ is in general still very large and an efficient solver for this space is needed. The most natural way to obtain such a solver for $\mathcal{M}_{J-1}$ is simply to repeat the above process by using the space $\mathcal{M}_{J-2}$ as a 'coarser' space with the supports of the nodal basis function in $\mathcal{M}_{J-1}$ as a domain decomposition. We continue in this way until we reach a coarse space $\mathcal{M}_0$ where a direct solver can be used. As a result, a multilevel algorithm based on domain decomposition is obtained. In fact, this recursively defined domain decomposition method is just a multigrid algorithm with the Gauss–Seidel method as a smoother. Multigrid methods will be the main topic in the rest of this paper.

7. 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 the full multigrid principle.

7.1. 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 rôle 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 analysed by three approaches in this section. The first approach is through numerical experiments, which would give an intuitive idea of 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 rôle of a smoother. The third approach is to build technical machinery for the convergence analysis of multigrid methods.

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

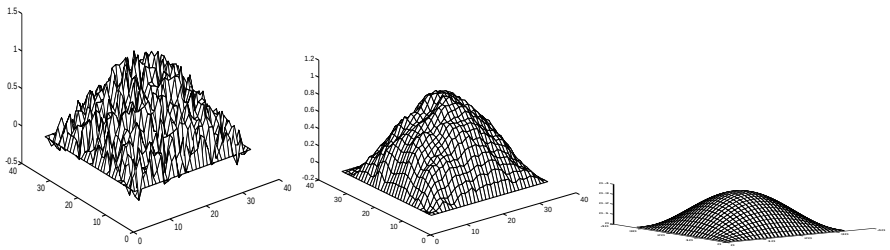

FIG. 12. *Residual after* 0, 2 *and* 100 *iterations, respectively, with* 961 *unknowns.*

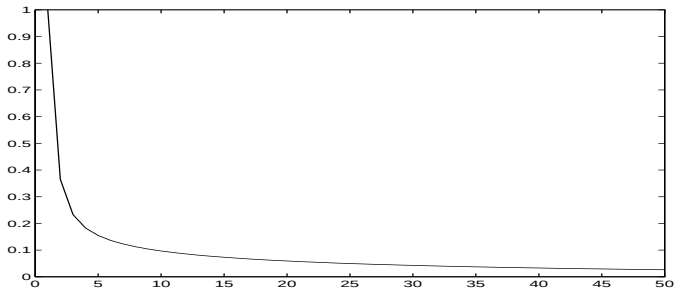

FIG. 13. *Convergence history of Gauss–Seidel method within 50 iterations.*

$$4u_{ij} - (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) = b_{i,j}, \quad 1 \le i,j \le n. \tag{7.1}$$

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

$$4\tilde{u}_{ij} = \omega(\bar{u}_{i+1,j} + \bar{u}_{i-1,j} + \bar{u}_{i,j+1} + \bar{u}_{i,j-1}) + b_{i,j} \tag{7.2}$$

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

$$4\tilde{u}_{ij} = (\bar{u}_{i+1,j} + \tilde{u}_{i-1,j} + \bar{u}_{i,j+1} + \tilde{u}_{i,j-1}) + b_{i,j}. \tag{7.3}$$

The Gauss–Seidel method has a good smoothing property. Let us illustrate this by a simple numerical example. Consider eqn. (7.1) with an initial residual $u - u^0$ shown on the left plot of Fig. 12. The initial residual apparently contains a lot of oscillations. The middle plot in Fig. 12 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 Fig. 12 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 Fig. 13, 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 this rapid initial reduction of 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 transferred 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.

7.1.2. Brandt's local mode analysis. The local mode analysis of Brandt [13] is an effective general 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 a 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 a smoother can in fact be applied in a rigorous fashion to the model problem discussed in the previous section. Let us first recall the discrete Fourier theory. For clarity, we confine our discussion to the two dimensional case. The discrete Fourier transform theory says that every discrete function $u : I_n \mapsto \mathbb{R}$, with $I_n = \{(i,j) : 0 \le i,j \le n\}$, 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

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

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

$$\tilde{\epsilon}_{ij} = \bar{\epsilon}_{i,j} - \frac{\omega}{4}(4\bar{\epsilon}_{ij} - (\bar{\epsilon}_{i+1,j} + \bar{\epsilon}_{i-1,j} + \bar{\epsilon}_{i,j+1} + \bar{\epsilon}_{i,j-1})). \tag{7.4}$$

We write

$$\tilde{c}_{i,j} = \sum_{\theta \in \Theta_n} \tilde{c}_\theta \psi_{i,j}(\theta) \tag{7.5}$$

and

$$\bar{c}_{i,j} = \sum_{\theta \in \Theta_n} \bar{c}_\theta \psi_{i,j}(\theta). \tag{7.6}$$

Substituting the above expressions into (7.4) and comparing the coefficients of each $\psi_{ij}(\theta)$, we obtain that

$$\lambda(\theta) = 1 - \omega \left(1 - \frac{\cos\theta_1 + \cos\theta_2}{2}\right). \tag{7.7}$$

where

$$\lambda(\theta) \equiv \frac{\tilde{c}(\theta)}{\bar{c}(\theta)} \tag{7.8}$$

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

The smoothing factor introduced by Brandt is the quantity

$$\bar{\rho} = \sup\{|\lambda(\theta)|, \pi/2 \le |\theta_k| \le \pi, k = 1, 2\}. \tag{7.9}$$

Roughly speaking, the smoothing factor $\bar{\rho}$ is the maximal amplification factor corresponding to those high frequency local modes that oscillate within range $2h$ (and hence cannot be resolved by a 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 ω 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 examine the smoothing property of the Gauss–Seidel iteration. Unlike the Jacobi method, the Gauss–Seidel method depends on the ordering of the unknowns. 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

$$\tilde{c}_{ij} = \frac{1}{4}(\bar{c}_{i+1,j} + \tilde{c}_{i-1,j} + \bar{c}_{i,j+1} + \tilde{c}_{i,j-1}). \tag{7.10}$$

Again using the Fourier transform (7.5) and (7.6), 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 the 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 Definition 4.20) if and only they are not neighbors (in either the horizontal or vertical directions). It is easy to see that the uniform grid in our example can be grouped into two colors, often called red and black. 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 cannot be obtained as easily as the lexicographical 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 lexicographical ordering. Furthermore red-black Gauss–Seidel has much better parallel features (see Definition 4.20).

7.1.3. General smoother analysis. We shall now develop some technical results concerning the smoothing property of the Gauss–Seidel method. We choose to study the 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 smoothers is relatively simple (see the analysis for Richardson in (2.14)).

LEMMA 7.1. *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 7.1, which means that Gauss–Seidel is basically like a Richardson iteration.

LEMMA 7.2. *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 7.1 □

LEMMA 7.3. *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* $\mathcal{A}$.

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 application of the Cauchy–Schwarz inequality. □

7.2. A basic multigrid cycle: the backslash (\) cycle. Although, as we shall see, multigrid methods have many variants, there is one particular multigrid algorithm which can be viewed as 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 classical point of view. This more classical approach makes it easier to introduce many different kinds of classical multigrid methods and also make it possible to use more classical approaches 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 inverses 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.

ALGORITHM 7.11. *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 at high frequencies. In other words, $v - v^m$ is smoother (see the middle plot in Fig. 12) and hence it can be very well approximated by a coarse space $\mathcal{M}_{k-1}$. The second step in the above algorithm plays the rôle 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:

$$u^{k+1} = u^k + B_J(f - Au^k). \tag{7.12}$$

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

Before we 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

$$\Phi^k = \Phi^{k+1}\mathcal{I}_k^{k+1}. \tag{7.13}$$

ALGORITHM 7.14 (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 form, but it can also be easily implemented in a nonrecursive fashion. For such an implementation, we refer to Algorithm 7.19

7.3. A convergence analysis using full elliptic regularity. With the assumption of full elliptic regularity (namely $\alpha = 1$ in (5.3), see also Theorem 5.4), 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 satisfy

$$\frac{c_0}{\lambda_k}(v, v) \le (R_k v, v) \le (A_k^{-1}v, v) \quad \forall\, v \in \mathcal{M}_k. \tag{7.15}$$

We would like to remark that the above assumptions on R_k can be much weakened and, for example, the R_k do not need to be symmetric (in this case, assumptions need to be made on the symmetrizations of the R_k, see, e.g., §7.5).

If the regularity estimate (5.3) holds with $\alpha = 1$, then there exists a positive constant c_1 independent of mesh parameters such that (see Theorem 5.4)

$$\|(I - P_{k-1})v\|_A^2 \le c_1 \lambda_k^{-1} \|A_k v\|^2 \quad \forall\, v \in \mathcal{M}_k. \tag{7.16}$$

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

LEMMA 7.4.

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

Proof.

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

The proof is completed by using the following elementary inequality:

$$\begin{aligned} ((I - K_k)K_k^{2m} v, v)_A &\le \frac{1}{2m} \sum_{j=0}^{2m-1} ((I - K_k)K_k^j v, v)_A \\ &\le \frac{1}{2m}(\|v\|_A^2 - \|K_k^m v\|_A^2). \end{aligned} \tag{7.17}$$

□

THEOREM 7.5. *For the Algorithm 7.11, we have*

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

Proof. By definition of Algorithm 7.11, we have

$$I - B_k A_k = (I - P_{k-1} B_{k-1} A_{k-1})(I - R_k A_k)^m$$

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

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

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

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

□

7.4. 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.

ALGORITHM 7.18. *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 nonrecursive implementation. The recursive formulation of the above algorithm makes it a little less straightforward to code sometimes. A nonrecursive version of the algorithm is given below in terms of matrices and vectors.

ALGORITHM 7.19 (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 $l = 2 : J$,
  $\tilde{\alpha}_l = \alpha_l + \mathcal{I}_{l-1}^l\alpha_{l-1}$;
  $\alpha_l = \tilde{\alpha}_l\mathcal{R}_l^t(\beta_l - \mathcal{A}_l\tilde{\alpha}_l)$;
endfor

$\mathcal{B}\beta = \alpha_J$.

The reason why this algorithm is called a 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

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

where the $B_k^{(\backslash)}$ correspond to the operators defined by the backslash cycle (Algorithm 7.11) and $*$ is the adjoint operator with respect to the A-inner product. Consequently,

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

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

ALGORITHM 7.20. *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 $w^0 = 0$ and*

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

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 the V-cycle. But it is often the case that the W-cycle is easier to analyze. We shall now give an optimal estimate for the convergence of the W-cycle based on the elliptic regularity assumption (5.3) (which implies that (5.20) holds).

THEOREM 7.6. *Under the elliptic regularity assumption* (5.3) *for some $\alpha \in (0, 1]$, the W-cycle iteration admits the following estimate:*

$$\|I - B_k A_k\|_A^2 \leq \frac{M}{m^\alpha + M} \tag{7.21}$$

for some constant M that is independent of mesh parameters.

Proof. Let $\delta = \frac{M}{m^\alpha + M}$. The estimate (7.21) will be proved by induction. As there is nothing to prove for $k = 0$, we assume that (7.21) 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 $\tilde{v} = K_k^m v$.

$$\begin{aligned}
&\|(I - B_k A_k)v\|_A^2 \\
&\quad \leq \|(I - P_{k-1})\tilde{v}\|_A^2 + \delta\|P_{k-1}\tilde{v}\|_A^2 \quad \text{(by induction)} \\
&\quad \leq (1 - \delta^2)\|(I - P_{k-1})\tilde{v}\|_A^2 + \delta^2\|\tilde{v}\|_A^2 \\
&\quad \leq (1 - \delta^2)(c_1\lambda_k^{-1}\|A_k\tilde{v}\|^2)^\alpha \|\tilde{v}\|_A^{2(1-\alpha)} + \delta^2\|\tilde{v}\|_A^2 \quad \text{(by (5.20))} \\
&\quad \leq (1 - \delta^2)\left[\frac{c_1}{2mc_0}(\|v\|_A^2 - \|\tilde{v}\|_A^2)\right]^\alpha \|\tilde{v}\|_A^{2(1-\alpha)} + \delta^2\|\tilde{v}\|_A^2 \quad \text{(by (7.17))}.
\end{aligned}$$

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

$$(1-\delta^2)\left[\frac{c_1}{2mc_0}(1-t)\right]^{\alpha} t^{1-\alpha} + \delta^2 t \le \delta \quad t \in [0,1]. \tag{7.22}$$

By the 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 (7.22) is a consequence of the 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 η is made to minimize the above left hand side, namely to equalize the coefficients of $1-t$ and t. The proof may be completed with some more elementary manipulations by choosing sufficiently large M in the expression for δ. □

7.5. 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 (4.1) is trivial.

We observe that, with the above choice of subspaces $\mathcal{V}_k$, there are redundant overlappings in the decomposition (4.1). 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 take care of those 'nonoverlapped parts' (which correspond to the so-called *high frequencies*). As we know, methods like the Gauss–Seidel method discussed earlier just satisfy such requirements.

With the above ingredients, the successive subspace correction method Algorithm 4.12 can be stated as follows.

ALGORITHM 7.23. *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

LEMMA 7.7. *For the Algorithm 7.11 with* $m = 1$, *we have*

$$I - B_J A_J = (I - T_0)(I - T_1)\cdots(I - T_J) \tag{7.24}$$

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* (7.12) *is mathematically equivalent to Algorithm 7.23.*

Furthermore if B_J^m *is obtained from Algorithm 7.11 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 of the convergence of the backslash cycle multigrid method. In particular, the framework given in §4 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 §7.9) .

We now consider the more general case in which we do not assume any elliptic regularity for the underlying partial differential equations (5.1).

We assume that all the smoothers R_k satisfy

$$(R_k v, v) \leq \omega_1 (A_k^{-1} v, v) \quad \forall\, v \in \mathcal{M}_k. \tag{7.25}$$

and, if the R_k are all symmetric,

$$\frac{c_0}{\lambda_k}(v, v) \leq (R_k v, v) \leq \frac{c_1}{\lambda_k}(v, v) \quad \forall\, v \in \mathcal{M}_k, \tag{7.26}$$

or, in general,

$$\frac{c_0}{\lambda_k}(v, v) \leq (\bar{R}_k v, v) \leq \frac{c_1}{\lambda_k}(v, v) \quad \forall\, v \in \mathcal{M}_k, \tag{7.27}$$

where $\bar{R}_k$ is the symmetrization of R_k (see eqn. (2.5)).

By Lemma 7.3, the Gauss–Seidel methods satisfy the above assumptions.

The idea is to use the general framework of §4. According to the theory there, we need to estimate two basic parameters, namely $\bar{K}_0$ and $\bar{K}_1$.

By Theorem 5.8 and (7.26), there exists a constant C_0 independent of mesh parameters such that

$$K_0 \leq C_0. \tag{7.28}$$

LEMMA 7.8. *The $\bar{\epsilon}_{ij}$ defined in* (4.26) *satisfy, for some $\gamma \in (0, 1)$ independent of mesh parameters,*

$$\bar{\epsilon}_{ij} \lesssim \gamma^{|i-j|/2}. \tag{7.29}$$

Proof. Let $i \geq j$. It follows from Lemma 5.5 and (7.26) 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 the above lemma that there exists a constant C_1 independent of mesh parameters such that

$$\bar{K}_1 \leq C_1. \tag{7.30}$$

With the above results, the following result follows directly from Theorem 4.11.

THEOREM 7.9. *Assume that the smoothers R_k satisfy* (7.26) *and* (7.25) *with $\omega_1 < 2$; then the backslash cycle (Algorithm 7.11 or Algorithm 7.23) satisfies*

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

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.

THEOREM 7.10. *For the iteration* (7.12) *with B_J given by Algorithm 7.11 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 $\tilde{P}_i = P_i - P_{i-1}$,

$$\begin{aligned}
\|v\|_A^2 &= \sum_{i=0}^{J} (\tilde{P}_i v, v)_A \\
&= \sum_{i=0}^{J} (\tilde{P}_i v, E_{i-1} v)_A \quad (\text{since } (I - E_{i-1})v \in \mathcal{M}_{i-1}) \\
&= \sum_{i=0}^{J} (\tilde{P}_i v, A_i P_i E_{i-1} v) \\
&\leq \sqrt{c_1} \sum_{i=0}^{J} \lambda_i^{-1/2} \|\tilde{P}_i v\|_A \|A_i P_i E_{i-1} v\| \quad (\text{by (7.16)}) \\
&\leq \sqrt{\frac{c_1}{c_0}} \sum_{i=0}^{J} \|\tilde{P}_i v\|_A (R_i A_i P_i E_{i-1} v, A_i P_i E_{i-1} v)^{1/2} \quad (\text{by (7.15)}) \\
&\leq \sqrt{\frac{c_1}{c_0}} \sum_{i=0}^{J} \|\tilde{P}_i v\|_A (T_i E_{i-1} v, E_{i-1} v)_A^{1/2} \\
&\leq \sqrt{\frac{c_1}{c_0}} \left(\sum_{i=0}^{J} \|\tilde{P}_i v\|_A^2 \right)^{1/2} \left(\sum_{i=0}^{J} (T_i E_{i-1} v, E_{i-1} v)_A \right)^{1/2} \\
&\leq \sqrt{\frac{c_1}{c_0}} \left(\sum_{i=0}^{J} \|\tilde{P}_i v\|_A^2 \right)^{1/2} \left(\sum_{i=0}^{J} (T_i E_{i-1} v, E_{i-1} v)_A \right)^{1/2} \\
&\leq \sqrt{\frac{c_1}{c_0}} \|v\|_A \left(\|v\|_A^2 - \|E_J v\|_A^2 \right)^{1/2}.
\end{aligned}$$

Consequently,

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

The desired estimate then follows easily. ∎

7.6. Full multigrid cycle. We shall now describe a more efficient multigrid technique, called a full multigrid cycle, originally proposed by Brandt.

On each level of the finite element space $\mathcal{V}_k$, there is a corresponding finite element approximation $u^{(k)} \in \mathcal{V}_k$ such that

$$A(u^{(k)}, v) = (F, v) \quad \forall v \in \mathcal{V}. \tag{7.31}$$

Similarly to (5.10), the best error estimate in the H^1 norm is

$$\|U - u^{(k)}\|_1 = O(h_k^\alpha). \tag{7.32}$$

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

$$\mathcal{A}_k \mu^{(k)} = b^{(k)} \tag{7.33}$$

where $b^{(k)} = ((F, \phi_i^k))$. It can be proved that, with $\mathcal{I}_k^{k+1}$ given by (7.13),

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

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 (7.32) by a multigrid iterative scheme.

ALGORITHM 7.34. $\hat{\mu}^{(1)} \leftarrow \mathcal{A}_1^{-1} b^{(1)}$. *For* $k = 2 : J$

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

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

PROPOSITION 7.11. *Assume that that* C_0 *is a positive constant satisfying (for all* k*)*

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

Then

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

if

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

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

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

Then

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

□

7.7. BPX multigrid preconditioners. We shall now describe a parallelized version of the multigrid method studied earlier. This method was first proposed by Bramble, Pasciak and Xu [9] and by Xu [32], and is now often known as the BPX preconditioner in the literature.

7.7.1. Basic algorithm and theory. There are different ways of deriving the BPX preconditioners. The method originally resulted from an attempt to parallelize the classical 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 5.8 and then, in the next subsection, study the method using the framework of subspace correction.

By Theorem 5.8, 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 5.2, 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

$$(R_k v_k, v_k) \eqsim h_k^2(v_k, v_k) \quad \forall\, v_k \in \mathcal{M}_k \tag{7.35}$$

then, for

$$B = \sum_{k=0}^{J} R_k Q_k \tag{7.36}$$

we have

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

namely

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

THEOREM 7.12. *Assume that the R_k satisfy* (7.35)*; then the preconditioner* (4.7) *satisfies*

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

We note that all the relaxation methods mentioned earlier, such as Richardson, Jacobi and symmetric Gauss–Seidel, satisfy (7.35).

7.7.2. Subspace correction approach. In §7.5, 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 §4 to derive some estimates for the BPX preconditioner somewhat more refined than that in Theorem 7.12; we leave the details to the interested readers. In §7.9, we shall use this approach to analyze the BPX preconditioner for locally refined meshes.

Implementation. Again, in view of (4.43), the algebraic representation of the preconditioner given by (4.43) is

$$\mathcal{B} = \sum_{k=0}^{J} \mathcal{I}_k \mathcal{R}_k \mathcal{I}_k^t, \tag{7.37}$$

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.

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

Let $\mathcal{I}_k^{k+1} \in \mathbb{R}^{n_{k+1} \times n_k}$ be as defined in (7.13); 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

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

$$\mathcal{B} = \sum_{k=0}^{J} h_k^{2-d} \mathcal{I}_k \mathcal{I}_k^t. \tag{7.38}$$

From (7.37) or (7.38), we see that the preconditioner depends entirely on the transformation between the nodal bases on multilevel spaces.

For $1 \le l \le J$, let

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

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.

ALGORITHM 7.39 (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 discussed above, the number of operations needed in the above algorithm is $O(n)$. We also note that all the vectors α_l for $1 \leq k \leq J$ need to be stored, but the whole storage space for these vectors is also only $O(n)$.

7.8. Hierarchical basis methods. Assume that we are given a nested sequence of multigrid subspaces 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 §5.2. The so-called *hierarchical basis* refers to the special set of nodal basis functions

$$\{\phi_i^k : x_i^k \in \mathcal{N}_k \setminus \mathcal{N}_{k-1}, k = 0, \cdots, J\}. \tag{7.40}$$

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 (hierarchical basis) as follows

$$\{h_k^{2-d} \phi_i^k : x_i^k \in \mathcal{N}_k \setminus \mathcal{N}_{k-1}, k = 0, \cdots, J\}. \tag{7.41}$$

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

The hierarchical basis in two dimensions was first analysed by Yserentant in his pioneering paper [43]. The work of Yserentant was apparently motivated by the famous unpublished technical report of Bank and Dupont [1]. Incidently these three authors got together and wrote another important paper [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 [33] (see also Xu [32]).

7.8.1. 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

$$\{\phi_i^k : x_i^k \in \mathcal{N}_k \setminus \mathcal{N}_{k-1}\}. \tag{7.42}$$

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

$$\mathcal{V}_k = (I_k - I_{k-1})\mathcal{M} = (I - I_{k-1})\mathcal{M}_k, \quad \text{for } k = 0 : J. \tag{7.43}$$

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 (7.43), the operators A_k are all well conditioned. In fact, by (5.19) and (5.5), 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 solved effectively by elementary iterative methods such as Richardson, Jacobi and Gauss–Seidel methods.

7.8.2. 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 (normal basis) stiffness matrix. This is indeed the case in two and three dimensions.

THEOREM 7.13. *Assume that $\hat{\mathcal{A}}$ is the stiffness matrix under the scaled hierarchical basis; then*

$$\kappa(\hat{\mathcal{A}}) \lesssim \kappa_d(h) \tag{7.44}$$

where

$$\kappa_d(h) \lesssim \begin{cases} 1 & \text{if } d = 1; \\ |\log h|^2 & \text{if } d = 2; \\ h^{2-d} & \text{if } d \geq 3. \end{cases} \tag{7.45}$$

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 the 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

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

Thus

$$\begin{aligned} \alpha^t \hat{\mathcal{A}} \alpha &= A(v,v) = \sum_{k,l} A(v_k, v_l) \\ &\lesssim \sum_{k,l} \gamma^{|k-l|} |v_k|_1 |v_l|_1 \quad \text{(by Lemma 5.6)} \\ &\lesssim \sum_k |v_k|_1^2 \eqsim |\alpha|^2. \end{aligned}$$

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

$$\begin{aligned} |\alpha|^2 &\eqsim \sum_k |v_k|_1^2 = \sum_k |(I_k - I_{k-1})v|_1^2 \\ &\lesssim \sum_k (J-k+1)|v|_1^2 \quad \text{(by (5.19))} \\ &\lesssim \kappa_d(h)|v|_1^2 \lesssim \kappa_d(h)\alpha^t \hat{\mathcal{A}} \alpha \end{aligned}$$

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

The above proof is essentially the same as in Yserentant [43] (and see also Ong [28] for $d = 3$).

7.8.3. Subspace correction approach and general case. Following Xu [33], we shall now study the HB method from the viewpoint of space decomposition and subspace correction.

In view of (4.43), the algebraic representation of the PSC preconditioner is

$$\mathcal{H} = \sum_{k=0}^J \mathcal{S}_k \mathcal{R}_k \mathcal{S}_k^t, \tag{7.46}$$

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 7.13,

$$\kappa(\mathcal{H}\mathcal{A}) \eqsim \kappa_d(h). \tag{7.47}$$

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 these iterations satisfies the following spectral equivalence:

$$\mathcal{R}_k \eqsim h_k^{2-d}. \tag{7.48}$$

In fact, the estimate (7.47) also follows easily from the general theory for the PSC preconditioner.

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 as follows.

ALGORITHM 7.49. *Let $\mu^0 \in \mathbb{R}^n$ be given. Assume that $\mu^k \in \mathbb{R}^n$ is obtained. Then μ^{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 later that

$$\|\mu - \mu_l\|_{\mathcal{A}} \leq \left(1 - \frac{c}{\kappa_d(h)}\right)^l \|\mu - \mu_0\|_{\mathcal{A}}. \tag{7.50}$$

7.8.4. Convergence analysis. LEMMA 7.14. *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 ω_1 is a constant and for symmetric Gauss–Seidel, $\omega_1 = 1$.

Proof. The proof for the Richardson or Jacobi method is straightforward. The proof for the symmetric Gauss–Seidel method is almost identical to that of Lemma 7.2 and the detail is left to the reader. □

LEMMA 7.15.

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

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 (5.19) 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 5.6. □

For the SSC iterative method, we apply Theorem 4.11 with Lemma 7.15 and get

THEOREM 7.16. *The Algorithm 4.12 with the subspaces $\mathcal{V}_k$ given by* (7.43) *satisfies*

$$\|E_J\|_A^2 \le 1 - \frac{2-\omega_1}{Cc_d}$$

provided that the R_k satisfy (7.14) *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 iteration. Numerical examples in [3] show that the hierarchical basis SSC algorithm converges much faster than the corresponding SSC algorithm.

7.8.5. Relation with BPX preconditioners. Observing that S_k in (7.46) is a submatrix of $\mathcal{I}_k$ given in (7.37), we then have

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

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

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

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 [42] for the numerical results.

7.9. 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 Ω which is also denoted by Ω_0. We introduce a relatively coarse and quasi-uniform triangulation of Ω_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 Ω_1 be a subregion where we wish to increase the resolution: we do so by subdividing the elements of the first triangulation to get a new triangulation of Ω_1 with mesh size h_1 in Ω_1 and we 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 Ω_i together with the corresponding finite element spaces $\mathcal{V}_i$ defined on a triangulation of mesh size h_i for $i = 1, 2, \cdots, J$ for some integer $J > 1$. Throughout, we have

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

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 Ω_k.

Let $A : \mathcal{V} \mapsto \mathcal{V}$ be as defined by (5.15). Operators $A_i : \mathcal{V}_i \mapsto \mathcal{V}_i$ and $Q_i, P_i : \mathcal{V} \mapsto \mathcal{V}_i$ can be defined as before. If we choose $R_i : \mathcal{V}_i \mapsto \mathcal{V}_i$ to be some appropriate approximate solvers of the A_i, 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: we therefore assume that, for the PSC type algorithm, the first subspace solver R_0 is SPD and satisfies

$$(R_0^{-1}v, v) \eqsim (A_0 v, v) \quad \forall\, v \in \mathcal{M}_0, \tag{7.51}$$

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

$$\|(I - R_0 A_0)\|_A \leq \delta_0 \tag{7.52}$$

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

As for the other subspace solvers R_k for $k > 1$, we assume for clarity that R_k is given by a Gauss–Seidel iteration or properly damped Jacobi iteration. 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 out 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 rôle in our analysis for the algorithms discussed in this section.

LEMMA 7.17. *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 $v \in \mathcal{V}$, $(\Pi_k - \Pi_{k-1})v \in \mathcal{V}_k$,*

$$\|(I - \Pi_k)v\| \lesssim \|(I - \hat{Q}_k)v\| \tag{7.53}$$

and

$$\|\Pi_k v\|_A \lesssim \|v\|_A. \tag{7.54}$$

Proof. The linear operator Π_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 (7.54), 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 (5.11) 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 (7.53). The rest of (7.54) can be estimated similarly by using $\|\Pi_k v\|_A \leq \|(I - \hat{Q}_k)v\|_A + \|v - w\|_A$. This completes the proof. □

LEMMA 7.18. *Let $\Pi_k : \mathcal{V} \mapsto \mathcal{M}_k$ be the operator from the previous lemma and $\Pi_{-1} = 0$; then, there exists a constant c_0 independent of mesh parameters such that*

$$\sum_{k=0}^{J} \lambda_k \|(\Pi_k - \Pi_{k-1})v\|_A^2 \leq c_0 \|v\|_A^2 \quad \forall\, v \in \mathcal{V}. \tag{7.55}$$

Proof. By Lemma 7.17 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 with (7.54),

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

The desired estimate then follows by using Theorem 5.9. □

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

$$B = \sum_{k=0}^{J} R_k Q_k. \tag{7.56}$$

Thanks to Lemma 7.18, as for the quasi-uniform case, we can use our general framework in §4 to obtain the following theorem.

THEOREM 7.19. *If R_0 satisfies* (7.51) *and R_k are given by Jacobi or symmetric Gauss–Seidel, then the PSC preconditioner* (7.56) *yields a uniformly bounded condition number*

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

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

$$R_k v = h_k^{2-d} \sum_{x_i^k \in \mathcal{N}_k} (v, \phi_i^k)\phi_i^k. \tag{7.57}$$

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

$$Bv = R_0 v + \sum_{k=1}^{J} h_k^{2-d} \sum_{x_i^k \in \mathcal{N}_k} (v, \phi_i^k)\phi_i^k. \tag{7.58}$$

We notice that (7.58) 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

$$Hv = R_0 v + \sum_{k=1}^{J} h_k^{2-d} \sum_{x_i^k \in \mathcal{N}_k \setminus \mathcal{N}_{k-1}} (v, \phi_i^k)\phi_i^k. \tag{7.59}$$

This preconditioner is equivalent to what is given in [43] 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.

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

THEOREM 7.20. *If R_0 satisfies* (7.52) *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 the early sixties by the Russian mathematician Fedorenko [19]. In the early seventies, Brandt [12] brought this method to the attention of western countries and extensive research has been done on this method since then. Nowadays it has become one of the most popular and powerful iterative methods.

Multilevel methods for composite grids can be traced back to Brandt [12] or to composite grids in McCormick [24] (see also the references therein). The finite element space on a mesh refined in this way is $\mathcal{V} = \sum_{i=0}^{J} \mathcal{V}_i$. PSC and SSC methods can be naturally obtained with Gauss–Seidel iterations 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 and Xu [9].

The aforementioned refined grids may not give the minimal degrees of freedom from an approximation point of view, but they are a computationally efficient approach and have the desirable structure for multigrid applications. If 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 [45]).

8. Multigrid for unstructured problems. 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. This type of theory can only be effectively applied to problems with good structures (for example, conforming finite elements on structured grids as discussed above).

8.1. Nonnested subspaces and varying bilinear forms. In this section, we shall present a general multigrid theory that only depends on some weak assumptions (see Bramble, Pasciak and Xu [10], Xu [32]). Such a theory has been successfully applied to many situations and some examples 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, for all $w \in \mathcal{M}_k, \phi \in \mathcal{M}_{k-1}$, by

$$(I_k^t w, \phi)_{k-1} = (w, I_k\phi)_k \quad A_{k-1}(I_k^* w, \phi) = A_k(w, I_k\phi). \tag{8.1}$$

In other words, the I_k^t and I_k^* are the adjoints of I_k with respect to the inner products $(\cdot,\cdot)_k$ and $A_k(\cdot,\cdot)$ respectively. I_k^t is often called the *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 λ_k is either equal to $\rho(A_k)$ or is 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 for $g \in \mathcal{M}_k(A_k w = g)$, B_k is defined as follows.*

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

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

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 $r_k = g - A_k w^{m_k}$.*

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

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

(8.2) 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 8.2). 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 $E_k \stackrel{\text{def}}{=} I - B_k A_k$ as given in the following lemma.

LEMMA 8.1. *Let $E_k = I - B_k A_k$ and $K_k = I - R_k A_k$. Then*

$$E_k = (\tilde{K}_k^{m_k})^* \left((I - I_k I_k^*) + I_k E_{k-1}^p I_k^* \right) \tilde{K}_k^{m_k}. \tag{8.3}$$

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

$$A_k(E_k u, v) = A_k((I - I_k I_k^*)\tilde{u}, \tilde{v}) + A_{k-1}(E_{k-1}^p I_k^* \tilde{u}, I_k^* \tilde{v}), \tag{8.4}$$

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

LEMMA 8.2. *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 algorithm named in the following:

DEFINITION 8.5. *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 (VV-cycle) if* $p = 1$ *and* $\gamma_0 m_k \leq m_{k-1} \leq \gamma_1 m_k$ *for* $k = 0, \cdots, J$, *where* γ_0 *and* γ_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 concerns only one level of space is the same as in the nested case.

(A0) $$\|v\|^2 \leq C_0 \lambda_k(\bar{R}_k v, v)$$

where $\bar{R}_k$ is the symmetrization of R_k (see eqn. (2.5)).

A direct consequence of (**A0**) is

(A0′) $$\rho(K_k) < 1,$$

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 the 'regularity and approximation assumption', is that there exists a constant $\beta \in (0, 1]$ and a constant C_1 such that

(A1) $$|A_k((I - I_k I_k^*)v, v)| \leq C_1 (\lambda_k{}^{-1} \|A_k v\|_k{}^2)^{\beta} A_k(v, v)^{1-\beta} \quad \forall v \in \mathcal{M}_k.$$

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

(8.6) $$A_k(I_k v, I_k v) \leq \tilde{C}_1 A_{k-1}(v, v), \quad \forall\, v \in \mathcal{M}_{k-1}.$$

THEOREM 8.3. *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* β*):*

1. The W-cycle converges uniformly for sufficiently many smoothings:

$$\|E_k\|_k \leq \frac{M}{m^{\beta/2}} \quad \text{if } m^{\beta/2} \geq 2M.$$

2. The variable V-cycle converges uniformly for sufficiently large m_J*:*

$$\|E_k\|_k \leq \frac{c_0}{m^{\beta/2}} \quad \text{if } m_J^{\beta/2} \geq c_0.$$

3. The 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 conditions

(A2) $A_k(I_k v, I_k v) \le A_{k-1}(v,v) \;\; A_k(I_k v, I_k v) \le A_{k-1}(v,v)$

hold for all v in $\mathcal{M}_{k-1}$, then the W-cycle and VV-cycle converge uniformly and the V-cycle converges nearly uniformly with any given number of smoothings:

$$\|\!|E_k|\!\|_k \le \begin{cases} \frac{M}{m_k^\beta + M} & \text{for the variable V-cycle,} \\ \frac{M^\beta}{(m+M)^\beta} & \text{for the W-cycle,} \\ \frac{k^{1/\beta-1}M}{m^\beta + k^{1/\beta-1}M} & \text{for the V-cycle.} \end{cases}$$

5. If the condition

$$A_k(I_k v, I_k v) \le 2A_{k-1}(v,v)$$

holds, 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 [32].

8.1.1. Applications. The above theory has found many applications. First of all, the development of the theory was motivated by applications to multigrid methods for unstructured grids. Unstructured grids refer to grids that do not have a natural multilevel structure. Most of the grids generated by traditional grid generators may fall into such 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 [15], 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 satisfy and hence only W-cycle or variable V-cycle can be proved to be convergent with sufficiently many smoothings, or variable V-cyle gives rise to an optimal preconditioner. One interesting exception to the above phenomenon is the work by Chen and Oswald [16] and Chen [17] 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.

8.2. The auxiliary space method with application to unstructured grids. In this section, an abstract framework of *auxiliary space methods* 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 makes 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 [38].

8.2.1. 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 set out 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 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 some approximation for $\mathcal{V}$. The space $\mathcal{V}_0$ need 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 a representation of or approximation to 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 other words, 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, Π plays the rôle of prolongation in the 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 rôle of R is to resolve what cannot 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 the Jacobi and Gauss–Seidel methods.

With the ingredients described above, the proposed preconditioner is as follows:

$$B = R + \Pi B_0 \Pi^t. \tag{8.7}$$

In the special case that $\mathcal{V}_0 \subset \mathcal{V}$ and $[\cdot, \cdot] = (\cdot, \cdot)$, Π can obviously be given by the natural inclusion operator and as a result Π^t is nothing but the orthogonal projection $Q_0 : \mathcal{V} \mapsto \mathcal{V}_0$. In this case, the preconditioner (8.7) 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 and Xu [9].

By definition, for any $u, v \in \mathcal{V}_h$,

$$(BAu, v)_A = (RAu, v)_A + [B_0 A_0 \Pi^* u, \Pi^* v]_{A_0} \tag{8.8}$$

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.

THEOREM 8.4. *Assume that there are some nonnegative constants α_0, α_1, λ_0, λ_1, and β_1 such that, for all $v \in \mathcal{V}$ and $w \in \mathcal{V}_0$,*

$$\alpha_0 \rho_A^{-1} (v, v) \le (Rv, v) \le \alpha_1 \rho_A^{-1} (v, v), \tag{8.9}$$

$$\lambda_0[w,w]_{A_0} \le [B_0A_0w,w]_{A_0} \le \lambda_1[w,w]_{A_0}, \tag{8.10}$$

$$\|\Pi w\|_A^2 \le \beta_1\|w\|_{A_0}^2, \tag{8.11}$$

and furthermore, assume that there exists a linear operator $P : \mathcal{V} \mapsto \mathcal{V}_0$ *and positive constants* β_0 *and* γ_0 *such that,*

$$\|Pv\|_{A_0}^2 \le \beta_0^{-1}\|v\|_A^2 \tag{8.12}$$

and

$$\|v - \Pi Pv\|^2 \le \gamma_0^{-1}\rho_A^{-1}\|v\|_A^2. \tag{8.13}$$

Then the preconditioner given by (8.7) *satisfies*

$$\kappa(BA) \le (\alpha_1 + \beta_1\lambda_1)((\alpha_0\gamma_0)^{-1} + (\beta_0\lambda_0)^{-1}). \tag{8.14}$$

In particular, if P *is a right inverse of* Π, *namely* $\Pi Pv =$ *for* $v \in \mathcal{V}$, *then*

$$\kappa((\Pi B_0\Pi^t)A) \le \frac{\beta_1}{\beta_0}\frac{\lambda_1}{\lambda_0}. \tag{8.15}$$

Proof. One first notes that (8.11) is equivalent to

$$\|\Pi^* v\|_{A_0}^2 \le \beta_1\|v\|_A^2 \quad \forall\ v \in \mathcal{V}.$$

By (8.8) and the assumptions, one has

$$(BAv,v)_A \le \alpha_1\rho_A^{-1}\|Av\|^2 + \lambda_1\|\Pi^* v\|_{A_0}^2 \le (\alpha_1 + \lambda_1\beta_1)\|v\|_A^2.$$

This means that $\lambda_{\max}(BA) \le \alpha_1 + \beta_1\lambda_1$. It follows that

$$\begin{aligned}
(v,v)_A &= (v - \Pi Pv, v)_A + [Pv, \Pi^* v]_{A_0} \\
&\le \|v - \Pi Pv\|\|Av\| + \|Pv\|_{A_0}\|\Pi^* v\|_{A_0} \\
&\le (\gamma_0\rho_A)^{-1/2}\|v\|_A\alpha_0^{-1/2}\rho_A^{1/2}(RAv,Av)^{1/2} \\
&\qquad \text{(by (8.13) and (8.9))} \\
&\quad +\beta_0^{-1/2}\|v\|_A\lambda_0^{-1/2}[B_0A_0\Pi^* v, \Pi^* v]_{A_0}^{1/2} \\
&\qquad \text{(by (8.12) and (8.10))} \\
&\le ((\alpha_0\gamma_0)^{-1} + (\beta_0\lambda_0)^{-1})^{1/2}\|v\|_A(BAv,v)_A^{1/2} \quad \text{(by (8.8))}.
\end{aligned}$$

This implies that $\lambda_{\min}(BA) \ge ((\alpha_0\gamma_0)^{-1} + (\beta_0\lambda_0)^{-1})^{-1}$. The estimate (8.14) then follows.

In the particular case that P is a right inverse of Π, one may take $\gamma_0 = \infty$ and $R = 0$; the proof of the corresponding estimate is then clear. □

The last estimate for a special case in the above theorem corresponds to the *fictitious space lemma* of Nepomnyaschikh [25, 26]. A necessary condition for Π to have a right inverse is that $\Pi : \mathcal{V}_0 \mapsto \mathcal{V}$ is surjective. As a consequence, $\dim\mathcal{V}_0 \ge \dim\mathcal{V}$, in other words $\mathcal{V}_0$ has to be at least as rich as $\mathcal{V}$. Furthermore the construction of Π 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 Π in the fictitious space approach; hence the resulting preconditioner is potentially more flexible and more robust.

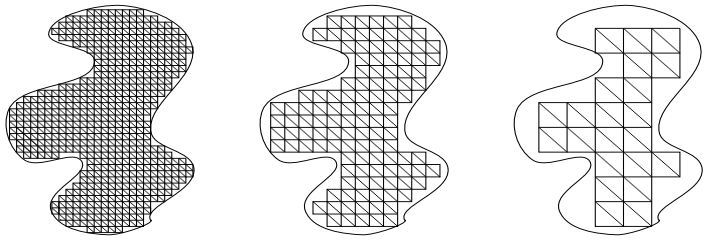

FIG. 14. *An example of an auxiliary grid together with two nested coarser grids.*

8.2.2. A special technique for unstructured grids. The purpose of this subsection is to construct optimal multigrid preconditioners for a 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 [39].

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 Ω_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 $\bar{\Omega} \setminus \Omega_h$.

8.2.3. 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 Ω (see Fig. 4(a)). Let Ω_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 Fig. 4(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.

LEMMA 8.5. *For all $v_0 \in \mathcal{V}_0$,*

$$\|v_0 - \Pi_h v_0\| \lesssim h\|v_0\|_1, \quad \|\Pi_h v_0\|_1 \lesssim \|v_0\|_1. \tag{8.16}$$

LEMMA 8.6. *For all $v \in \mathcal{V}_h$,*

$$\|v - \Pi_0 v\| \lesssim h\|v\|_1 \ \textit{and}\ \|\Pi_0 v\|_1 \lesssim \|v\|_1. \tag{8.17}$$

The proof of the above two lemmas can be found in Xu [38].

8.2.4. 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

$$B_h = R_h + \Pi_h B_0 \Pi_h^t, \tag{8.18}$$

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 8.4 and the estimates from the previous subsections, there exist some positive constants α, β, γ 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. Multigrid methods of this type have been discussed by Kornhuber and Yserentant [22]. 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 Fig. 4, if Fig. 4(a) corresponds to $\hat{\mathcal{V}}_J = \mathcal{V}_0$, then Fig. 4(b) and Fig. 4 (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 [2] $\kappa(B_0 A_0) \lesssim 1$ 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 $\kappa(BA) \lesssim |\log h|^2$.

Attention is now 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 Π_h, namely

$$(\Pi_h \psi_1, \ldots, \Pi_h \psi_{n_0}) = (\phi_1, \ldots, \phi_{n_h})\mathcal{I},$$

[2] Although the corresponding estimate was not optimal in [22] for their more general considerations, it can easily be proved to be optimal in the current context by the technique in [33].

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. [33])

$$\mathcal{B}_h = \mathcal{R}_h + \mathcal{I}\mathcal{B}_0\mathcal{I}^t$$

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 α_{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 both be 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 easily be obtained as long as the location of x_i^h is known in $\mathcal{T}_0$.

In summary, when B_0 is given by a BPX preconditioner, the resulting preconditioner which may be called the *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 independently 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 Ω.

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 conditions, for example, it is not sufficient that the auxiliary grid only consists of those regular elements that are contained in Ω. Instead, the auxiliary grid consists of all those regular elements that *intersect* with Ω. 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 [24] (see also the references therein) Bramble, Pasciak and Xu [9] and Bramble and Pasciak [6].

The main spirit of this section is that, with the help of an additional smoother, quite a 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 only needs to satisfy the following approximation property:

$$\inf_{w \in \mathcal{V}_0} \|v - w\|_0 \lesssim h^m \|v\|_{m,h} \quad \forall\, v \in \mathcal{V} \tag{8.19}$$

where $\|\cdot\|_0$ is the L^2 norm and $\|\cdot\|_{m,h}$ is a (discretized) H^m norm.

The approximation property (8.19) 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 rôle of $\mathcal{V}_0$ or the rôle 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 have an approximation property like (8.19) that is related to the order of the original differential operator but is not, in a certain sense, strongly related to the discretization space ($\mathcal{V}$).

The weak approximation property (8.19) 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.

9. Nonsymmetric and/or indefinite linear problems. In this section, we shall study a class of iterative methods for solving nonsymmetric or indefinite equations that are governed by some SPD systems. The main idea is to use two grids of different sizes in which the coarse grid is used to resolve the lower frequencies while the fine grid is used to resolve high frequencies on which the leading SPD operator really dominates and hence an SPD preconditioner can be used. The development of this type of two-grid method can be found in Xu and Cai [41], Xu [35, 34, 36, 40, 37].

9.1. Model problems. In this section, we shall discuss finite element discretizations for nonsymmetric and/or indefinite linear partial differential equations. These results, mostly well known, lay down the groundwork for the further analysis of nonlinear problems.

Linear elliptic partial differential operators. Let α, β, γ (with the ranges in $\mathbb{R}^{2\times 2}$, $\mathbb{R}^2$ and $\mathbb{R}^1$ respectively) be smooth functions on $\overline{\Omega}$ satisfying, for some positive constant α_0,

$$\xi^T \alpha(x)\xi \geq \alpha_0 |\xi|^2 \quad \forall\, \xi \in \mathbb{R}^2.$$

We shall study the following two linear operators: [3]

$$\mathcal{L}\, v = -\mathrm{div}(\alpha(x)\nabla v) \quad \text{and} \quad \hat{\mathcal{L}}\, v = \mathcal{L}\, v + \beta(x)\cdot\nabla v + \gamma(x) v. \tag{9.20}$$

Obviously $\mathcal{L} : H_0^1(\Omega) \to H^{-1}(\Omega)$ is an isomorphism. Our basic assumption is that $\hat{\mathcal{L}} : H_0^1(\Omega) \to H^{-1}(\Omega)$ is also an isomorphism. (A simple sufficient condition for this assumption to be satisfied is that $\gamma(x) \geq 0$.) An application of the open mapping theorem yields

$$\|v\|_{H^1(\Omega)} \lesssim \|\hat{\mathcal{L}} v\|_{H^{-1}(\Omega)} \quad \forall\, v \in H_0^1(\Omega). \tag{9.21}$$

It is easy to see that if $\hat{\mathcal{L}}$ satisfies the above assumption, so does its formal adjoint:

$$\hat{\mathcal{L}}^* u = -\mathrm{div}(\alpha(x)\nabla u + \beta(x) u) + \gamma(x) u.$$

Namely $\hat{\mathcal{L}}^* : H_0^1(\Omega) \to H^{-1}(\Omega)$ is also isomorphic and satisfies (9.21).

[3] Nonsymmetric and/or indefinite operators or bilinear forms will in general be denoted by symbols with 'hats'.

Corresponding to $\mathcal{L}$ and $\hat{\mathcal{L}}$, we define two bilinear forms, for $u, v \in H_0^1(\Omega)$, as follows:

$$A(u,v) = \int_\Omega \alpha(x)\nabla u \cdot \nabla v \, dx, \quad \hat{A}(u,v) = A(u,v) + \int_\Omega ((\beta \cdot \nabla u)v + \gamma(x)uv) \, dx. \tag{9.22}$$

We shall often use the following well known regularity result (using (9.21)).

LEMMA 9.1. *If $u \in H_0^1(\Omega)$ and $\hat{\mathcal{L}}u \in L^2(\Omega)$, then $u \in H^2(\Omega)$ and*

$$\|u\|_{H^2(\Omega)} \leq C\|\hat{\mathcal{L}}u\|$$

for some positive constant C depending on the coefficients of $\hat{\mathcal{L}}$ and the domain Ω.

Finite element discretizations. We assume that Ω is partitioned by a quasi-uniform triangulation $T_h = \{\tau_i\}$. By this we mean that the τ_i are simplices of size h with $h \in (0,1)$ and $\bar{\Omega} = \cup_i \bar{\tau}_i$ and there exist constants C_0 and C_1 not depending on h such that each element τ_i is contained in (contains) a ball of radius $C_1 h$ (respectively $C_0 h$).

For a given triangulation T_h, a finite element space $\mathcal{V}_h \subset \mathcal{V} \equiv H_0^1(\Omega)$ is defined by

$$\mathcal{V}_h = \{v \in C(\bar{\Omega}) : v|_\tau \in \mathcal{V}_\tau^r \quad \forall\, \tau \in T_h, v|_{\partial\Omega} = 0\},$$

where $\mathcal{V}_\tau^r$ is the space of polynomials of degree not greater than a positive integer r. For a given $v \in C(\bar{\Omega})$, $v_I \in \mathcal{V}_h$ will denote the standard nodal value interpolation of v.

It is well known (cf. [18]) that $\mathcal{V}_h$ satisfies the following approximation property

$$\inf_{\chi \in \mathcal{V}_h} \{\|v - \chi\|_{L^p(\Omega)} + h\|v - \chi\|_{W_p^1(\Omega)}\} \lesssim h^k |v|_{W_p^k(\Omega)}, \tag{9.23}$$

for all $v \in W_p^k(\Omega) \cap H_0^1(\Omega)$, $2 \leq k \leq r+1$ and $1 \leq p \leq \infty$.

Let $P_h : \mathcal{V} \longrightarrow \mathcal{V}_h$ be the standard Galerkin projection defined by

$$A(P_h v, \chi) = A(v, \chi) \quad \forall\, \chi \in \mathcal{V}_h. \tag{9.24}$$

Using Lemma 9.1 and a standard duality argument, we have

$$\|v - P_h v\| \lesssim h\|v\|_{H^1(\Omega)} \quad \forall\, v \in \mathcal{V}. \tag{9.25}$$

For the nonsymmetric and/or indefinite problems, the following lemma (based on Schatz [30]) is of fundamental importance.

LEMMA 9.2. *If $h \ll 1$, then*

$$\|v_h\|_{H^1(\Omega)} \lesssim \sup_{\varphi \in \mathcal{V}_h} \frac{\hat{A}(v_h, \varphi)}{\|\varphi\|_{H^1(\Omega)}} \quad \textit{and} \quad \|v_h\|_{H^1(\Omega)} \lesssim \sup_{\varphi \in \mathcal{V}_h} \frac{\hat{A}(\varphi, v_h)}{\|\varphi\|_{H^1(\Omega)}} \;\forall\, v_h \in \mathcal{V}_h. \tag{9.26}$$

The same results are also valid for $\varepsilon \ll 1$ if $\hat{A}$ in (9.26) is replaced by $\hat{A}_\varepsilon$ defined by

$$\hat{A}_\varepsilon(u,v) = \int_\Omega (\alpha_\epsilon(x)\nabla u \cdot \nabla v + (\beta_\epsilon \cdot \nabla u)v + \gamma_\epsilon(x)uv) \, dx$$

with the functions $\alpha_\epsilon, \beta_\epsilon, \gamma_\epsilon \in L_\infty(\Omega)$ *satisfying*

$$\|\alpha - \alpha_\varepsilon\|_{L^\infty(\Omega)} + \|\beta - \beta_\varepsilon\|_{L^\infty(\Omega)} + \|\gamma - \gamma_\varepsilon\|_{L^\infty(\Omega)} = \delta_\varepsilon$$

where $\delta_\epsilon = o(1)$ *as* $\varepsilon \to 0$.

Proof. Since $\hat{\mathcal{L}} : H_0^1(\Omega) \to H^{-1}(\Omega)$ is an isomorphism, we have

$$\|v_h\|_{H^1(\Omega)} \lesssim \sup_{w \in \mathcal{V}} \frac{\hat{A}(v_h, w)}{\|w\|_{H^1(\Omega)}}.$$

Note that, by definition and (9.25),

$$\begin{aligned}
\hat{A}(v_h, P_h w) &= \hat{A}(v_h, w) - \hat{A}(v_h, w - P_h w) \\
&= \hat{A}(v_h, w) + (A - \hat{A})(v_h, w - P_h w) \\
&\geq \hat{A}(v_h, w) - c\|v_h\|_{H^1(\Omega)}\|w - P_h w\| \\
&\geq \hat{A}(v_h, w) - c_1 h\|v_h\|_{H^1(\Omega)}\|w\|_{H^1(\Omega)}.
\end{aligned}$$

The proof of the first estimate in (9.26) then follows by using the fact that $\|P_h w\|_{H^1(\Omega)} \lesssim \|w\|_{H^1(\Omega)}$. The proof of the second estimate is similar.

For the form $\hat{A}_\epsilon(\cdot, \cdot)$, it follows from the assumption that

$$\hat{A}_\epsilon(v_h, \phi) \geq \hat{A}(v_h, \phi) - c\delta_\epsilon\|v_h\|_{H^1(\Omega)}\|\phi\|_{H^1(\Omega)}.$$

The desired result then follows easily if $\epsilon \ll 1$. □

Now, define $\hat{P}_h : \mathcal{V} \longrightarrow \mathcal{V}_h$ by

$$\hat{A}(\hat{P}_h v, \chi) = \hat{A}(v, \chi) \quad \forall\, \chi \in \mathcal{V}_h. \tag{9.27}$$

Following (9.26) and Lemma 9.1, we have

LEMMA 9.3. *If* $h \ll 1$, *then* $\hat{P}_h$ *is well defined and*

$$\|u - \hat{P}_h u\| + h\|u - \hat{P}_h u\|_{H^1(\Omega)} \lesssim h \inf_{\chi \in \mathcal{V}_h} \|u - \chi\|_{H^1(\Omega)} \quad \forall\, u \in \mathcal{V}.$$

The following results show that P_h and $\hat{P}_h$ are 'super-close' in the $H^1(\Omega)$ and $W^1_\infty(\Omega)$ norms.

LEMMA 9.4. *Assume that* P_h *and* $\hat{P}_h$ *are defined by* (9.24) *and* (9.27) *respectively; then*

$$\|P_h u - \hat{P}_h u\|_{H^1(\Omega)} \lesssim \|u - \hat{P}_h u\|.$$

Proof. By definition

$$A(P_h u - \hat{P}_h u, \chi) = (A - \hat{A})(u - \hat{P}_h u, \chi) \quad \forall\, \chi \in \mathcal{V}_h.$$

The desired estimates then follow by taking $\chi = P_h u - \hat{P}_h u$. □

We end this section by stating some basic error estimates for $\hat{P}_h$.

LEMMA 9.5. *The projection* $\hat{P}_h$ *admits the following estimate*

$$\begin{aligned}
\|u - \hat{P}_h u\| &\lesssim h^{r+1}\|u\|_{H^{r+1}(\Omega)}, \\
\|u - \hat{P}_h u\|_{H^1(\Omega)} &\lesssim h^{r+1}\|u\|_{H^{r+1}(\Omega)}.
\end{aligned}$$

9.2. Two-grid discretizations. In this section, we shall present a number of algorithms for non-SPD problems based on two finite element spaces. The idea is to reduce a non-SPD problem to an SPD problem by solving a non-SPD problem on a much smaller space.

The basic mechanisms in our approach are two quasi-uniform triangulations of Ω, T_H and T_h, with two different mesh sizes H and h ($H > h$), and the corresponding finite element spaces $\mathcal{V}_H$ and $\mathcal{V}_h$ which will be called coarse and fine space respectively. In the applications given below, we shall always assume that

$$H = O(h^\lambda), \quad \text{for some } 0 < \lambda < 1. \tag{9.28}$$

With the bilinear form $\hat{A}$ defined in (9.22), for $h \ll 1$, let $u_h \in \mathcal{V}_h$ be the unique solution of

$$\hat{A}(u_h, \chi) = (f, \chi) \quad \forall\, \chi \in \mathcal{V}_h$$

and denote the bilinear form of the lower order terms of the operator $\hat{\mathcal{L}}$ (in (9.20)) by

$$N(v, \chi) = (\hat{A} - A)(v, \chi) = (\beta \cdot \nabla v, \chi) + (\gamma v, \chi).$$

Let us now present our first two-grid algorithm.

ALGORITHM 9.29.

1. Find $u_H \in \mathcal{V}_H$ such that $\hat{A}(u_H, \varphi) = (f, \varphi) \quad \forall\, \varphi \in \mathcal{V}_H$.

2. Find $u^h \in \mathcal{V}_h$ such that $A(u^h, \chi) + N(u_H, \chi) = (f, \chi) \quad \forall\, \chi \in \mathcal{V}_h$.

We note that the linear system in the second step of the above algorithm is SPD.

THEOREM 9.6. *Assume $u^h \in \mathcal{V}_h$ is the solution obtained by Algorithm 9.29 for $H \ll 1$. Then*

$$\|u_h - u^h\|_{H^1(\Omega)} \lesssim H^{r+1} \|u\|_{H^{r+1}(\Omega)}$$

and

$$\|u - u^h\|_{H^1(\Omega)} \lesssim (h^r + H^{r+1}) \|u\|_{H^{r+1}(\Omega)}$$

provided that $u \in H^{r+1}(\Omega)$.

Proof. A direct calculation and an application of Lemma 9.5 shows that

$$\begin{aligned} A(u_h - u^h, \chi) &= -N((I - \hat{P}_H)\, u_h, \chi) \\ &\lesssim \|(I - \hat{P}_H) u_h\|\, \|\chi\|_{H^1(\Omega)} \\ &\lesssim (H \|u - u_h\|_{H^1(\Omega)} + \|(I - \hat{P}_H) u\|)\, \|\chi\|_{H^1(\Omega)} \\ &\lesssim H^{r+1}\, \|u\|_{H^{r+1}(\Omega)}\, \|\chi\|_{H^1(\Omega)}. \end{aligned}$$

The desired result then follows. □

REMARK. If $\beta(x) = 0$ and $r \geq 2$, we have

$$\|P_h u - u^h\|_{H^1(\Omega)} \lesssim \|u - u_H\|_{H^{-1}(\Omega)} \lesssim H^{r+2} \|u\|_{H^{r+2}(\Omega)}$$

and

$$\|u - u^h\|_{H^1(\Omega)} \lesssim (h^r + H^{r+2}) \|u\|_{H^{r+2}(\Omega)}.$$

Algorithm 9.29 can be applied in a successive fashion.

ALGORITHM 9.30. *Let $u_h^0 = 0$; assume that $u_h^k \in \mathcal{V}_h$ has been obtained; $u_h^{k+1} \in \mathcal{V}_h$ is defined as follows.*

1. Find $e_H \in \mathcal{V}_H$ such that $\hat{A}(e_H + u_h^k, \varphi) = (f, \varphi) \quad \forall\, \varphi \in \mathcal{V}_H$.

2. Find $u^h \in \mathcal{V}_h$ such that $A(u_h^{k+1}, \chi) + N(u_h^k + e_H, \chi) = (f, \chi) \quad \forall\, \chi \in \mathcal{V}_h$.

As is well known, most linear iterative methods for solving algebraic systems can be obtained by an appropriate matrix (or operator) splitting. For the nonsymmetric system under consideration, the most natural splitting would lead to the following iterative method

$$A(u_h^{k+1}, \chi) + N(u_h^k, \chi) = (f, \chi) \quad \forall\, \chi \in \mathcal{V}_h.$$

This iterative scheme, however, is not convergent in general. The Algorithm 9.30 may be considered as a modification of this 'natural' iterative scheme with recourse to an additional coarse space.

THEOREM 9.7. *Assume $u_h^k \in \mathcal{V}_h$ is the solution obtained by Algorithm 9.30 for $k \geq 1$; then*

$$\|u_h - u_h^k\|_{H^1(\Omega)} \lesssim H^{k+r} \|u\|_{H^{r+1}(\Omega)},$$

and

$$\|u - u_h^k\|_{H^1(\Omega)} \lesssim (h^r + H^{k+r}) \|u\|_{H^{r+1}(\Omega)}$$

Proof. By definition and Lemma 9.5,

$$\begin{aligned} A(u_h - u_h^k, \chi) &= N((I - \hat{P}_H)(u_h^{k-1} - u_h), \chi) \\ &\leq \|(I - \hat{P}_H)(u_h^{k-1} - u_h)\| \, \|\chi\|_{H^1(\Omega)} \\ &\lesssim H \, \|u_h^{k-1} - u_h\|_{H^1(\Omega)} \, \|\chi\|_{H^1(\Omega)}. \end{aligned}$$

This implies

$$\|u_h - u_h^k\|_{H^1(\Omega)} \lesssim H \|u_h - u_h^{k-1}\|_{H^1(\Omega)}.$$

Applying the above estimate successively and then using Theorem 9.6 yields

$$\|u_h - u_h^k\|_{H^1(\Omega)} \lesssim H^{k-1} \|u_h - u_h^1\|_{H^1(\Omega)} \lesssim H^{k+r} \|u\|_{H^{r+1}(\Omega)}.$$

□

Before ending this section, we present an algorithm for symmetric and indefinite problems (namely $\beta(x) = 0$ in (9.20)). This algorithm is based on the finite element space

$$\hat{\mathcal{V}}_h = (I - \hat{P}_H)\mathcal{V}_h.$$

ALGORITHM 9.31.

1. Find $u_H \in \mathcal{V}_H$ such that $\hat{A}(u_H, \varphi) = (f, \varphi) \quad \forall\, \varphi \in \mathcal{V}_H$.

2. Find $e_h \in \hat{\mathcal{V}}_h$ such that $A(e_h, \chi) = (f, \chi) \quad \forall\, \chi \in \hat{\mathcal{V}}_h$.

3. $u^h = u_H + e_h$.

We note that the system in the second step of Algorithm 9.31 is SPD. But since it is on the space $\hat{\mathcal{V}}_h$, this system may not be solved very easily. Nevertheless this algorithm is of some theoretical interest. In fact, as shown in the next theorem,

$$\|u - u^h\|_{H^1(\Omega)} \lesssim (h + H^3) \|u\|_{H^2(\Omega)}$$

if linear finite elements are used.

THEOREM 9.8. *Assume $u^h \in \mathcal{V}_h$ is obtained by Algorithm 9.31; then*

$$\|u - u^h\|_{H^1(\Omega)} \lesssim (h^r + H^{r+2})\|u\|_{H^{r+1}(\Omega)}.$$

Proof. As $\hat{A}$ is symmetric, so is $\hat{P}_H$. Thus

$$\hat{A}((I - \hat{P}_H)u, \chi) = (f, \chi) \quad \forall\, \chi \in \hat{\mathcal{V}}_h.$$

Therefore

$$\begin{aligned} A(u_h - (u_H + e_h), \chi) &= -(\gamma(u - u_H), \chi) \lesssim H\|u - u_H\|\|\chi\|_{H^1(\Omega)} \\ &\lesssim H^{r+2}\|u\|_{H^{r+1}(\Omega)}\|\chi\|_{H^1(\Omega)}. \end{aligned}$$

where we have used the fact that $\|\chi\| \lesssim H\|\chi\|_{H^1(\Omega)}$ for $\chi \in \hat{\mathcal{V}}_h$. The desired result follows by taking $\chi = u_h - (u_H + e_h) \in \hat{\mathcal{V}}_h$. ∎

9.3. Iteration and precondition. The algorithms discussed above are based on exact solvers for the SPD problems. We shall now discuss algorithms based on inexact SPD solvers. For generality and clarity, we begin our discussion in an abstract setting.

We assume that $\mathcal{V}$ is a given linear vector space equipped with an inner product $(\cdot, \cdot)$. Let $L(\mathcal{V})$ denote the space of all linear operators from $\mathcal{V}$ to itself. We are interested in solving the equation

$$\hat{A}u = f, \tag{9.32}$$

for a given $f \in \mathcal{V}$. Here $\hat{A} \in L(\mathcal{V})$ is a given invertible operator satisfying

$$\hat{A} \;=\; A + N,$$

and $A \in L(\mathcal{V})$ is SPD in the sense that

$$(Au, v) = (u, Av) \quad \forall\, u, v \in \mathcal{V} \quad \text{and} \quad (Av, v) > 0 \quad \text{if} \quad v \neq 0;$$

the perturbation operator $N \in L(\mathcal{V})$ is not SPD in general.

As A is SPD, $(\cdot, \cdot)_A = (A\cdot, \cdot)$ defines an inner product on $\mathcal{V}$ and induces a norm on $\mathcal{V}$, denoted by $\|\cdot\|_A$. Given $G \in L(\mathcal{V})$, we define its A-norm by

$$\|G\|_A = \sup_{v \in \mathcal{V}} \frac{\|Gv\|_A}{\|v\|_A}.$$

The construction of an iterative algorithm for (9.32) often amounts to the construction of a $\hat{B} \in L(\mathcal{V})$ which behaves like $\hat{A}^{-1}$. One approach is to use $\hat{B}$ to obtain a linear iterative scheme as follows

$$u^{k+1} = u^k + \hat{B}(f - \hat{A}u^k), \tag{9.33}$$

for $k = 0, 1, 2, \cdots$, and any $u^0 \in \mathcal{V}$. Obviously a sufficient condition for the convergence of scheme (9.33) is

$$\eta = \|I - \hat{B}\hat{A}\|_A < 1,$$

and in this case

$$\|u - u^k\|_A \leq \eta^k \|u\|_A.$$

Another approach is to use $\hat{B}$ as a preconditioner for (9.32) in conjunction with GMRES type methods (cf. [29]). Unlike the conjugate gradient method for SPD problems, the GMRES method may not be convergent without proper preconditioning. A preconditioner for the GMRES method is not only to speed up the convergence but more importantly to guarantee the convergence as well. More precisely, if there are two constants $\alpha_0, \alpha_1 > 0$ such that

$$(\hat{B}\hat{A}v, v)_A \geq \alpha_0 (v, v)_A, \quad \|\hat{B}\hat{A}v\|_A \leq \alpha_1 \|v\|_A, \quad \forall\, v \in \mathcal{V},$$

then the GMRES method applied to the preconditioned system

$$\hat{B}\hat{A}u = \hat{B}f$$

with the inner product $(\cdot, \cdot)_A$ converges at the rate $1 - \alpha_0^2/\alpha_1^2$ (cf. [29]).

Now we assume that a subspace $\mathcal{V}_0 \subset \mathcal{V}$ is given; we define an operator $\hat{A}_0 : \mathcal{V}_0 \mapsto \mathcal{V}_0$, and three projections $Q_0, P_0, \hat{P}_0 : \mathcal{V} \mapsto \mathcal{V}_0$ by, for all $u_0, v_0 \in \mathcal{V}_0$,

$$(\hat{A}_0 u_0, v_0) = (\hat{A}u_0, v_0),$$

and for all $u \in \mathcal{V}, v_0 \in \mathcal{V}_0$,

$$(AP_0 u, v_0) = (Au, v_0), \quad (\hat{A}\hat{P}_0 u, v_0) = (\hat{A}u, v_0), \quad (Q_0 u, v_0) = (u, v_0).$$

It is clear that $\hat{A}_0$, P_0 and Q_0 are well defined. We shall assume that $\hat{A}_0$ is invertible, which implies that $\hat{P}_0$ is also well defined. By the definitions of $\hat{P}_0, \hat{A}_0$ and Q_0,

$$\hat{A}_0 \hat{P}_0 = Q_0 \hat{A}.$$

It follows that, for a given $f \in \mathcal{V}$,

$$\hat{u}_0 = \hat{A}_0^{-1} Q_0 f \quad \text{if and only if} \quad (\hat{A}\hat{u}_0, v_0) = (f, v_0), \quad \forall\, v_0 \in \mathcal{V}_0.$$

Many estimates in this paper will be established in terms of the parameter

$$\delta_0 = \sup_{u,v \in \mathcal{V}} \frac{(N(I - \hat{P}_0)u, v)}{\|u\|_A \|v\|_A}. \tag{9.34}$$

The assumption that we shall make later is that δ_0 can be sufficiently small if the subspace $\mathcal{V}_0$ is properly chosen.

In the study of preconditioners, we need to use another parameter defined by

$$\bar{\delta} = \sup_{u,v \in \mathcal{V}} \frac{(Nu, v)}{\|u\|_A \|v\|_A}.$$

It is easy to see that

$$\|A^{-1}N\|_A \leq \bar{\delta}. \tag{9.35}$$

Observe that $\bar{\delta} = \delta_0$ if $\mathcal{V}_0 = \{0\}$. Without loss of generality, we assume that $\delta_0 \leq \bar{\delta}$.

LEMMA 9.9. *For any* $u \in \mathcal{V}$,

$$\|(\hat{P}_0 - P_0)u\|_A \le \delta_0 \|u\|_A, \quad \|u - \hat{P}_0 u\|_A \le (1+\delta_0)\|u\|_A. \tag{9.36}$$

Proof. It follows from the definitions of $\hat{P}_0$ and P_0 that

$$(A(\hat{P}_0 - P_0)u, v_0) = (N(I - \hat{P}_0)u, v_0), \quad \forall\ u \in \mathcal{V}, v_0 \in \mathcal{V}_0,$$

which, with $v_0 = (\hat{P}_0 - P_0)u$, implies the first inequality in (9.36). The second estimate obviously follows from the first one. □

Linear iterative algorithms. We now present the main algorithm proposed in Xu [35]. The algorithm depends on a given solver for A, represented by a $B \in L(\mathcal{V})$, satisfying

$$\|I - BA\|_A < 1.$$

ALGORITHM 9.37. *Given* $u^0 \in \mathcal{V}$, *assume* u^k *is defined for* $k \ge 0$; *then*

1. *solve (exactly) the equation on* $\mathcal{V}_0$:

$$\hat{A}_0 \hat{u}_0 = Q_0(f - \hat{A}u^k);$$

2. *set* $g = f - \hat{A}(u^k + \hat{u}_0)$, *for* $i = 0, 1, \cdots, p$ *and* $v^0 = 0$,

$$v^{i+1} = v^i + B(g - Av^i);$$

3. $u^{k+1} = u^k + \hat{u}_0 + v^p$.

Like in the classical multigrid method, the first step of the above algorithm plays the rôle of correction on the small subspace $\mathcal{V}_0$; the second step plays the rôle of smoothing (by the SPD operator A).

Let us derive the error equation of the above algorithm. Without loss of generality, we assume that $p = 1$. Note that $f = \hat{A}u$, and it follows that

$$\hat{u}_0 = \hat{P}_0(u - u^k) \quad \text{and} \quad v^1 = B\hat{A}(I - \hat{P}_0)(u - u^k).$$

Thus

$$u - u^{k+1} = (I - B\hat{A})(I - \hat{P}_0)(u - u^k).$$

Obviously Algorithm 9.37 is identical to (9.33) if $\hat{B}$ satisfies

$$I - \hat{B}\hat{A} = (I - B\hat{A})(I - \hat{P}_0). \tag{9.38}$$

THEOREM 9.10. *Assume that* $\hat{B}$ *is given by* (9.38); *then*

$$\|I - \hat{B}\hat{A}\|_A \le \eta,$$

where

$$\eta = \rho^p + 3\delta_0, \quad \rho = \|I - BA\|_A. \tag{9.39}$$

Consequently

$$\|u - u^k\|_A \le (\rho^p + 3\delta_0)^k \|u - u^0\|_A,$$

where the u^k are defined by Algorithm 9.37 and u is the solution of (9.32). *Therefore the Algorithm 9.37 is convergent if δ_0 is small enough to ensure that $3\delta_0 < 1 - \rho^p$.*

Proof. Without loss of generality, we assume that $p = 1$. Given $u \in \mathcal{V}$, denote $u_0 = \hat{P}_0 u$, $v = A^{-1}\hat{A}(u - u_0)$ and $w = u - u_0$. We shall first show that

$$\|w - v\|_A \le \delta_0 \|u\|_A, \quad \|v\|_A \le (1 + 2\delta_0)\|u\|_A. \tag{9.40}$$

In fact

$$\begin{aligned} \|w - v\|_A^2 &= (A(w - v), w - v) = ((A - \hat{A})(u - u_0), w - v) \\ &= -(N(u - u_0), w - v) \le \delta_0 \|u\|_A \|w - v\|_A. \end{aligned}$$

The first estimate in (9.40) then follows. To see the second estimate in (9.40), by Lemma 9.9

$$\begin{aligned} \|v\|_A^2 &= (Av, v) = (\hat{A}(u - u_0), v) = (A(u - u_0), v) + (N(u - u_0), v) \\ &\le (1 + \delta_0)\|u\|_A \|v\|_A + \delta_0 \|u\|_A \|v\|_A \le (1 + 2\delta_0)\|u\|_A \|v\|_A. \end{aligned}$$

Therefore (9.40) is justified. Thanks to (9.40), the rest of the proof is easy:

$$\begin{aligned} &\|(I - B\hat{A})(I - \hat{P}_0)u\|_A = \|w - B(Av)\|_A \\ &\le \|w - v\|_A + \|v - B(Av)\|_A \le \delta_0 \|u\|_A + \rho \|v\|_A \\ &\le (\delta_0 + \rho(1 + 2\delta_0)) \|u\|_A \le (\rho + 3\delta_0) \|u\|_A \end{aligned}$$

as desired. □

Preconditioners for GMRES type methods. Based on the theory just developed, a number of preconditioners can be derived in a straightforward fashion.

First, as a direct consequence of Theorem 2.1, we have

THEOREM 9.11. *Suppose*

$$\hat{B} = (I - B\hat{A})\hat{A}_0^{-1} Q_0 + B; \tag{9.41}$$

then, for all $v \in \mathcal{V}$,

$$(\hat{B}\hat{A}v, v)_A \ge (1 - \eta)(v, v)_A, \quad \|\hat{B}\hat{A}v\|_A \le (1 + \eta)\|v\|_A.$$

The proof of the above theorem is straightforward and hence is omitted. We shall now derive the theory developed in [41].

THEOREM 9.12. *Let*

$$\hat{B} = \omega \hat{A}_0^{-1} Q_0 + B. \tag{9.42}$$

Then, for η given by (9.39) *and for all $v \in \mathcal{V}$,*

$$(\hat{B}\hat{A}v, v)_A \ge \frac{1}{2}(1 - \eta)(v, v)_A, \quad \|\hat{B}\hat{A}v\|_A \le (\omega + 2)(1 + \bar{\delta})\|v\|_A, \tag{9.43}$$

provided that ω is sufficiently large and δ_0 is sufficiently small, e.g.

$$\omega \ge \frac{(1 + 2\bar{\delta})^2}{1 - \eta}, \quad \delta_0 \le \frac{1}{4} \frac{1 - \eta}{\omega + 1 + 2\bar{\delta}}. \tag{9.44}$$

Proof. Obviously

$$\begin{aligned}\hat{B}\hat{A} &= \omega\hat{P}_0 + B\hat{A} = (\omega - 1 + B\hat{A})\hat{P}_0 + \hat{P}_0 + B\hat{A}(I - \hat{P}_0) \\ &= (\omega - 1 + B\hat{A})P_0 + (\omega - 1 + B\hat{A})(\hat{P}_0 - P_0) + \hat{P}_0 + B\hat{A}(I - \hat{P}_0)\end{aligned}$$

By (9.35) and the fact that $\|I - BA\|_A < 1$, it is easy to show that

$$\|I - B\hat{A}\|_A \leq 1 + 2\bar{\delta}.$$

Hence, by (9.36)

$$((\omega - 1 + B\hat{A})(\hat{P}_0 - P_0)v, v)_A \leq (\omega + 1 + 2\bar{\delta})\delta_0\|v\|_A^2.$$

An application of the Cauchy–Schwarz inequality gives

$$\begin{aligned}((I - B\hat{A})P_0v, v)_A &\leq \|I - B\hat{A}\|_A\|P_0v\|_A\|v\|_A \\ &\leq \frac{1}{1-\eta}(1 + 2\bar{\delta})^2\|P_0v\|_A^2 + \frac{1-\eta}{4}\|v\|_A^2.\end{aligned}$$

Combining the above two estimates with Theorem 9.11 yields

$$(\hat{B}\hat{A}v, v)_A \geq (\omega - \frac{(1 + 2\bar{\delta})^2}{1-\eta})\|P_0v\|_A^2 + \left(\frac{3(1-\eta)}{4} - (\omega + 1 + 2\bar{\delta})\delta_0\right)\|v\|_A^2.$$

The first estimate in (9.43) then follows if (9.44) holds. The rest of the proof is straightforward. □

We are now in a position to derive the main result in [41].

THEOREM 9.13. *Assume that $\bar{B}$ is a SPD preconditioner for A and*

$$\hat{B} = \omega\hat{A}_0^{-1}Q_0 + \bar{B}. \tag{9.45}$$

Then, for all $v \in \mathcal{V}$,

$$(\hat{B}\hat{A}v, v)_A \geq \frac{\lambda_0 + \lambda_1}{4}(\frac{2\lambda_0}{\lambda_1 + \lambda_0} - 3\delta_0)A(v, v),$$

and

$$\|\hat{B}\hat{A}v\|_A \leq (\omega + 2)(1 + \bar{\delta})\frac{\lambda_0 + \lambda_1}{2}\|v\|_A,$$

provided that ω is sufficiently large and δ_0 is sufficiently small. Here

$$\lambda_0 = \lambda_{\min}(BA), \lambda_1 = \lambda_{\max}(BA).$$

Proof. Let $B = \frac{2}{\lambda_0 + \lambda_1}\bar{B}$. Then

$$\rho = \|I - BA\|_A \leq \frac{\lambda_1 - \lambda_0}{\lambda_1 + \lambda_0} < 1.$$

The desired result can be derived from Theorem 9.12. □

Subspace correction method. The algorithms we have studied above are based on a given iterative algorithm for the SPD problem. In this section, we shall discuss a special class of iterative methods for SPD problems and discuss the corresponding Algorithm 9.37 and its modifications.

Suppose that $\mathcal{V}_0$ used in the definition of Algorithm 9.37 coincides with that in the decomposition (4.1). Then, if Algorithm 9.37 is applied with Algorithm 4.12, the subspace problems on $\mathcal{V}_0$ are solved twice in each iteration, once for A_0 and once for $\hat{A}_0$. We shall remove the solver for A_0 from Algorithm 4.12 and modify Algorithm 9.37 as follows.

ALGORITHM 9.46. *Given $u^0 \in \mathcal{V}$, assume that u^k is defined for $k \geq 1$; then we define $u^{k+1} = \hat{u}^k + v^J$ where*

$$\hat{u}^k = u^k + \hat{A}_0^{-1} Q_0 (f - \hat{A} u^k)$$

and, for $i = 1, \cdots, J$,

$$v^i = v^{i-1} + R_i Q_i (g - A v^{i-1})$$

with $g = f - \hat{A}\hat{u}^k$ and $v^0 = 0$.

The error equation of the above algorithm is

$$u - u^{k+1} = (I - \tilde{B}\hat{A})(I - \hat{P}_0)(u - u^k)$$

where

$$I - \tilde{B}A = (I - T_J)(I - T_{J-1}) \cdots (I - T_1). \tag{9.47}$$

THEOREM 9.14. *Assume that $\omega_1 < 2$. Then Algorithm 9.46 converges if δ_0, given by (9.34), is sufficiently small. Furthermore the error operator $\tilde{E} = (I - \tilde{B}\hat{A})(I - \hat{P}_0)$ satisfies*

$$\|\tilde{E}\|_A \leq \eta$$

where

$$\eta = 5\delta_0 + \sqrt{1 - \frac{2 - \omega_1}{K_0(1 + \omega_1(K_1 - 1))^2}}. \tag{9.48}$$

Proof. Define B by

$$I - BA = (I - \tilde{B}A)(I - P_0). \tag{9.49}$$

Then by the estimate (4.39) in Thereom 4.11,

$$\rho = \rho(I_B A) \leq \sqrt{1 - \frac{2 - \omega_1}{K_0(1 + \omega_1(K_1 - 1))^2}}. \tag{9.50}$$

A direct manipulation yields

$$\begin{aligned}(I - \tilde{B}\hat{A})(I - \hat{P}_0) = (I - B\hat{A})(I - \hat{P}_0) \\ + \quad (I - \tilde{B}A)(P_0 - \hat{P}_0) + (B - \tilde{B})N(I - \hat{P}_0).\end{aligned}$$

Thus

$$\begin{aligned}\|(I-\tilde{B}\hat{A})(I-\hat{P}_0)u\|_A &\le \|(I-B\hat{A})(I-\hat{P}_0)u\|_A \\ &+ \|(I-\tilde{B}A)(P_0-\hat{P}_0)u\|_A + \|(B-\tilde{B})N(I-\hat{P}_0)u\|_A \\ &\equiv I_1+I_2+I_3.\end{aligned}$$

The estimate of I_1 is given by Theorem 9.10

$$I_1 \le (\rho+3\delta_0)\|u\|_A$$

where ρ is given by (9.50). By the assumption on R_i,

$$\|I-T_i\|_A \le 1,$$

which implies that $\|I-\tilde{B}A\|_A \le 1$. Hence, by (9.36),

$$I_2 \le \|(P_0-\hat{P}_0)u\|_A \le \delta_0\|u\|_A.$$

It remains to estimate I_3. We first note that, by (9.49),

$$(B-\tilde{B})A = (I-\tilde{B}A)P_0.$$

Thus

$$\|(B-\tilde{B})A\|_A = \|I-\tilde{B}A\|_A\|P_0\|_A \le 1.$$

Let 't' and '$*$' denote the transpositions with respect to the inner products $(\cdot,\cdot)$ and $(A\cdot,\cdot)$ respectively; then

$$\|(B-\tilde{B})^tA\|_A = \|[(B-\tilde{B})A]^*\|_A = \|(B-\tilde{B})A\|_A \le 1.$$

Consequently

$$\begin{aligned}\|(B-\tilde{B})N(I-\hat{P}_0)u\|_A^2 &= ((B-\tilde{B})N(I-\hat{P}_0)u, A(B-\tilde{B})N(I-\hat{P}_0)u) \\ &\le \delta_0\|u\|_A\|(B-\tilde{B})^tA(B-\tilde{B})N(I-\hat{P}_0)u\|_A \\ &\le \delta_0\|u\|_A\|(B-\tilde{B})^tA\|_A\|(B-\tilde{B})N(I-\hat{P}_0)u\|_A \\ &\le \delta_0\|u\|_A\|(B-\tilde{B})N(I-\hat{P}_0)u\|_A.\end{aligned}$$

Hence

$$I_3 = \|(B-\tilde{B})N(I-\hat{P}_0)u\|_A \le \delta_0\|u\|_A.$$

The desired estimate then follows. □

With the subspace correction methods for the SPD problem, we shall now discuss the corresponding preconditioners studied in Section 3.2.

THEOREM 9.15. *For $\tilde{B}$ given by* (9.47), *we have*

$$\hat{B} = (I-\tilde{B}\hat{A})\hat{A}_0^{-1}Q_0 + \tilde{B}. \tag{9.51}$$

Then, for η given by (9.48) *and for all $v \in \mathcal{V}$,*

$$(\hat{B}\hat{A}v,v)_A \ge (1-\eta)(v,v)_A, \quad \|\hat{B}\hat{A}v\|_A \le (1+\eta)\|v\|_A.$$

Because of Theorem 9.14, the proof of this theorem or the next one is identical to that of Theorem 9.11 or Theorem 9.12.

THEOREM 9.16. *For $\tilde{B}$ given by* (9.47), *define*

$$\hat{B} = \omega \hat{A}_0^{-1} Q_0 + \tilde{B}. \tag{9.52}$$

Then, for all $v \in \mathcal{V}$,

$$(\hat{B}\hat{A}v, v)_A \geq \frac{1}{2}(1-\eta)A(v,v),$$

and

$$\|\hat{B}\hat{A}v\|_A \leq (\omega+2)(1+\bar{\delta})\|v\|_A,$$

provided that ω is sufficiently large and δ_0 is sufficiently small. Note that preconditioner (9.52) may also be applied in the SPD case.

THEOREM 9.17. *Suppose*

$$\hat{B} = \omega \hat{A}_0^{-1} Q_0 + \sum_{i=1}^{J} R_i Q_i. \tag{9.53}$$

Then, for all $v \in \mathcal{V}$,

$$(\hat{B}\hat{A}v, v)_A \geq \frac{\lambda_0+\lambda_1}{4}\left(\frac{2\lambda_0}{\lambda_1+\lambda_0} - 4\delta_0\right)(v,v)_A,$$

and

$$\|\hat{B}\hat{A}v\|_A \leq (\omega+2)(1+\bar{\delta})\frac{\lambda_0+\lambda_1}{2}\|v\|_A,$$

provided that ω is sufficiently large and δ_0 is sufficiently small.

Proof. Using the obvious identity

$$\hat{B}\hat{A} = \hat{P}_0 - P_0 + (\omega - 1)\hat{P}_0 + \bar{B}\hat{A},$$

the desired result then follows by (9.36) and Theorem 9.13. □

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. **73**, 1–36 (1996).

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

[16] Z. Chen and P. Oswald. *Multigrid and multilevel methods for nonconforming rotated q_1 elements.* Preprint (1996).

[17] Z. Chen. *On the convergence of nonconforming multigrid methods for second-order elliptic problems.* Preprint (1996).

[18] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* North–Holland, Amsterdam, New York (1978).

[19] R. P. Fedorenko. *A relaxation method for solving elliptic difference equations.* Z. Vycisl. Mat. i. Mat. Fiz. **1**, 922–927 (1961). Also in USSR Comput. Math. and Math. Phys. **1**, 1092–1096 (1962).

[20] W. Hackbusch. *Multigrid Methods and Applications*, volume 4 of *Computational Mathematics.* Springer-Verlag, Berlin (1985).

[21] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations.* Springer-Verlag, Berlin (1993).

[22] 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.

[23] S. F. McCormick. *Multigrid Methods*, volume 3 of *Frontiers in Applied Mathematics.* SIAM Books, Philadelphia (1987).

[24] S. F. McCormick. *Multilevel Adaptive Methods for Partial Differential Equations*, volume 6 of *Frontiers in Applied Mathematics.* SIAM Books, Philadelphia (1989).

[25] S. V. Nepomnyaschikh. *Mesh theorems of traces, normalizations of function traces and their inversion.* Sov. J. Numer. Anal. Math. Modeling **6**, 151–168 (1991).

[26] 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.

[27] 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).

[28] 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).

[29] Y. Saad and M. H. Schultz. *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems.* SIAM J. Sci. Stat. Comput. **7**, 856–869 (1986).

[30] Alfred H. Schatz. *An observation concerning Ritz–Galerkin methods with indefinite bilinear forms.* Math. Comp. **28**, 959–962 (1974).

[31] P. Wesseling. *An Introduction to Multigrid Methods.* John Wiley & Sons, Chichester (1992).

[32] J. Xu. *Theory of Multilevel Methods.* PhD thesis, Cornell University (1989).

[33] J. Xu. *Iterative methods by space decomposition and subspace correction: A unifying approach.* SIAM Review **34**, 581–613 (1992).

[34] J. Xu. *Iterative methods by SPD and small subspace solvers for nonsymmetric or indefinite 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 106–118, Philadelphia (1992). SIAM.

[35] J. Xu. *A new class of iterative methods for nonselfadjoint or indefinite problems.* SIAM J. Numer. Anal. **29**, 303–319 (1992).

[36] J. Xu. *A novel two-grid method for semilinear elliptic equations.* SIAM J. Sci. Comput. **15**, 231–237 (1994).

[37] J. Xu. *Some two–grid finite element methods.* In *Domain Decomposition Methods in Science and Engineering: The Sixth International Conference on Domain Decomposition*, volume 157 of *Contemporary Mathematics*, pages 79–87, Providence, Rhode Island (1994). American Mathematical Society.

[38] J. Xu. *An auxiliary space preconditioning technique with application to unstructured grids.* To appear in Numer. Math.

[39] 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).

[40] J. Xu. *Two-grid discretization techniques for linear and nonlinear elliptic PDEs.* To appear in SIAM J. Numer. Anal.

[41] J. Xu and X.-C. Cai. *A preconditioned GMRES method for nonsymmetric or indefinite problems.* Math. Comp. *59*, 311–319 (1992).

[42] J. Xu and J. Qin. *Some remarks on a multigrid preconditioner.* SIAM J. Sci. Comput. **15**, 172–184 (1994).

[43] H. Yserentant. *On the multi–level splitting of finite element spaces.* Numer. Math. **49**, 379–412 (1986).

[44] H. Yserentant. *Old and new convergence proofs for multigrid methods.* Acta Numerica. Cambridge University Press (1992).

[45] S.Y. Zhang *Optimal order non–nested multigrid methods for solving finite element equations I–III.* Part I in Math. Comp. **55**, 23–36 (1990); Part II in Math. Comp. **55**, 439–450 (1990); Part III in Math. Comp. **64**, 23–49 (1995).