Acta Numerica (2017), pp. 591–721 © Cambridge University Press, 2017
doi:10.1017/S096249297000083 Printed in the United Kingdom

Algebraic multigrid methods

Jinchao Xu and Ludmil Zikatanov
Center for Computational Mathematics and Applications,
Department of Mathematics,
The Pennsylvania State University,
University Park, PA 16802, USA
E-mail: jinchao@psu.edu, ludmil@psu.edu

This paper provides an overview of AMG methods for solving large-scale systems of equations, such as those from discretizations of partial differential equations. AMG is often understood as the acronym of 'algebraic multigrid', but it can also be understood as 'abstract multigrid'. Indeed, we demonstrate in this paper how and why an algebraic multigrid method can be better understood at a more abstract level. In the literature, there are many different algebraic multigrid methods that have been developed from different perspectives. In this paper we try to develop a unified framework and theory that can be used to derive and analyse different algebraic multigrid methods in a coherent manner. Given a smoother R for a matrix A, such as Gauss–Seidel or Jacobi, we prove that the optimal coarse space of dimension n_c is the span of the eigenvectors corresponding to the first n_c eigenvectors $\bar{R}A$ (with $\bar{R} = R + R^T - R^T AR$). We also prove that this optimal coarse space can be obtained via a constrained trace-minimization problem for a matrix associated with $\bar{R}A$, and demonstrate that coarse spaces of most existing AMG methods can be viewed as approximate solutions of this trace-minimization problem. Furthermore, we provide a general approach to the construction of quasi-optimal coarse spaces, and we prove that under appropriate assumptions the resulting two-level AMG method for the underlying linear system converges uniformly with respect to the size of the problem, the coefficient variation and the anisotropy. Our theory applies to most existing multigrid methods, including the standard geometric multigrid method, classical AMG, energy-minimization AMG, unsmoothed and smoothed aggregation AMG and spectral AMGe.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

CONTENTS

1. Introduction

Multigrid methods are among the most efficient numerical methods for solving large-scale systems of equations – linear and non-linear alike – arising from the discretization of partial differential equations. These types of methods can be viewed as an acceleration of traditional iterative methods based on local relaxation, such as Gauss–Seidel and Jacobi methods. For linear systems arising from finite element or finite difference discretization of elliptic boundary value problems, local relaxation methods were observed to converge very rapidly for the high-frequency part of the solution. The low-frequency part of the solution, although slow to converge, corresponds to a relatively smooth part of the function that can be well approximated on a coarser grid. The main idea behind such multigrid methods is to project the error obtained after applying a few iterations of local relaxation methods onto a coarser grid. The projected error equations have two characteristics. First, the resulting system is smaller. Second, part of the slowly converging low-frequency error on a finer grid becomes relatively high-frequency on the coarser grid, and these frequencies can be further corrected via a local relaxation method, but this time on the coarse grid. By repeating this process via ever coarser grids, a multilevel iterative process is obtained. Such algorithms have been proved to have uniform convergence with nearly optimal complexity for a large class of linear algebraic systems arising from

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

the discretization of partial differential equations, especially elliptic boundary problems of second and fourth order. One main component of this type of multilevel algorithm is a hierarchy of geometric grids, typically a sequence of nested grids obtained by successive refinement. The resulting algorithms are known as geometric multigrid (GMG) methods.

Despite their extraordinary efficiency, however, GMG methods have limitations. They depend on a hierarchy of geometric grids, which is often not readily available, and it can be argued that the range of applicability of GMG methods is therefore limited.

Algebraic multigrid (AMG) methods were designed in an attempt to address such limitations. They were proposed as a means to generalize geometric multigrid methods for systems of equations that share properties with discretized PDEs, such as the Laplacian equation, but potentially have unstructured grids in the underlying discretization. The first AMG algorithm in Brandt, McCormick and Ruge (1982) was a method developed under the assumption that such a problem was being solved. The AMG algorithm was later generalized using several heuristics to extend its applicability to more general problems and matrices. As a result, a variety of AMG methods have been developed in the past three decades, and they have been successfully applied to many practical problems.

In this paper we give an overview of AMG methods from a theoretical viewpoint. AMG methods have been developed via a combination of certain theoretical considerations and heuristic arguments, and many AMG methods work, with varying efficiency, for different applications. We find it very hard to give a coherent picture of the state of the art of AMG methods by simply choosing to make a comprehensive list of existing algorithms without digging into their theoretical foundations. Unfortunately, a good theoretical understanding of how and why these methods work is still seriously lacking.

In preparing this paper, we have undertaken the task of making a thorough investigation into the design and analysis of AMG from a theoretical point of view. While there are many bits and pieces of ideas diffused in the literature, we have managed to re-examine most of the existing results and ‘re-invent the wheel’ in trying to deliver a coherent theoretical description. To do this, we have developed several tools for the design and analysis of AMG.

With very few exceptions, most AMG algorithms have targeted the solution of symmetric positive definite (SPD) systems. In this paper we choose to present our studies for a slightly larger class of problems, namely symmetric positive semi-definite (SPSD) systems. This approach is not only more inclusive, but more importantly, the SPSD class of linear systems can be viewed as more intrinsic to AMG ideas. For example, while the standard discretizations of the Laplacian operator with homogeneous Dirichlet boundary conditions results in an SPD system, the design of AMG may be

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

better understood by using local problems (defined on subdomains) with homogeneous Neumann boundary conditions, yielding SPSD sub-systems.

In short, in this paper we consider AMG techniques for solving the linear system of equations

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

where A is a given SPSD operator or sparse matrix, and the problem is posed in a vector space of large dimension. The starting point of an AMG procedure is first to choose a smoother, which is often taken to be some local relaxation iterative method such as pointwise Jacobi or Gauss–Seidel, or, more generally, overlapping Schwarz methods. The use of pointwise smoothers seems to encompass most of the efforts in the literature to construct and implement most (if not all) AMG methods. More general smoothers based on overlapping Schwarz methods are necessary for some problems, but we shall not study them in detail in this paper. In any event, any chosen smoother is expected to converge well only on certain components of the solution, which will be known as *algebraic high frequencies* with respect to the given smoother. With the smoother fixed, the main task of an AMG method is then to identify a sequence of coarse spaces that would complement this smoother well. Roughly speaking, an ideal sequence of coarse spaces is such that any vector (namely the solution to (1.1) for any f) in the finest space can be well represented by a linear combination of all the algebraic high frequencies on all coarse spaces. As a result, the AMG method would converge well for problem (1.1).

It is hard to translate the above multilevel setting into a concise mathematical statement. Instead we will focus on first answering this question for a two-level setting. For a two-level setting, the first theoretical question is as follows: ‘Given a smoother, say R, what is the optimal coarse space of given dimension such that the resulting AMG has the best convergence rate?’ This question will be thoroughly addressed in Section 5. As it turns out, the optimal coarse space will consist of the eigenvectors corresponding to the lower end of the spectrum of a matrix such as RA. While our two-level theory is theoretically pleasing, it does not offer a practical solution, as finding these eigenvectors would be too expensive. Thus, the task of the AMG design is to find a good but inexpensive approximation of this *algebraic low-frequency* eigenspace that will still result in an AMG algorithm with desirable convergence properties. We call such an approximation of an optimal coarse space a ‘quasi-optimal’ coarse space.

In the design of all these AMG algorithms, one key component is the coarsening of spaces via the graph associated with the matrix A. The two main strategies are based on independent sets and on aggregation. We will present several approaches to the construction of quasi-optimal coarse spaces. We would especially advocate two approaches that have sound

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

theoretical foundations. The first approach is outlined in Section 5 and later on in Section 11. We will first prove that the optimal coarse space from the theory in Section 5 can be characterized, in a mathematically equivalent manner, by solutions to a trace-minimization problem. In a functional setting, such a trace-minimization can be interpreted as minimization of the sum of the energy norms of a set of coarse basis functions. These precise equivalents give very clear guidance as to how some AMG methods can be constructed. One practical approach based on such a theorem is to look for energy-minimization basis functions among locally supported function classes. The resulting algorithms are known as energy-min AMG. Other AMG methods, such as classical AMG and aggregation-based AMG, can be viewed as approximations to energy-min AMG. The second approach is outlined in Section 6. The main idea is to construct a quasi-optimal coarse space by piecing together the low-end eigenspaces of some appropriately defined local operators or matrices. This approach can be used to provide quasi-optimal coarse spaces for various AMG methods, including the standard geometric multigrid method, classical AMG, energy-minimization AMG, unsmoothed and smoothed aggregation AMG, and spectral AMGe. As a simple example, this method relies on the fact that an $n_c = Jm$-dimensional low-end eigenspace of an operator can be well approximated by gluing together J-pieces of m-dimensional low-end eigenspaces, for some carefully chosen local operators. Here m is a very small integer, for example, $m = 1$ for the Laplacian operator and $m = 3$ (resp. $m = 6$) for the two-dimensional (resp. three-dimensional) linear elasticity operator. This important property of eigenspaces is closely related to Weyl's lemma on the asymptotic behaviour of eigenvalues for elliptic boundary value problems, discussed in Section 2.3.

Most AMG methods are designed in terms of the adjacency graph of the coefficient matrix of a given linear algebraic system. In Section 7 we give a brief description of graph theory and the adjacency graph of a sparse matrix. One highlight of Section 7 is the concept of M-matrix relatives. This simple tool is instrumental in the design and analysis of the classical AMG method.

One important step in the design of most AMG methods is to zero out some entries of the coefficient matrix A by using the concept of connection strength to construct a filtered matrix $\tilde{A}$, which is equivalent to omitting weakly connected edges in the adjacency graph $\mathcal{G}(A)$ to obtain $\mathcal{G}(\tilde{A})$. Several possible strength functions are introduced in Section 8 to describe the connection strength. In Section 9, the graph $\mathcal{G}(\tilde{A})$ is then coarsened by either keeping a maximal independent set (MIS) as a coarse vertex set $\mathcal{C}$ and then dropping the rest of grid, or using aggregation/agglomeration. In Section 9 we also give some technical details of the construction of a coarse space by using degrees of freedom.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

By using the aforementioned general approaches and theoretical techniques, we then motivate and present a number of AMG methods. Some of the highlights of the paper are outlined below.

We first give an overview of GMG and its relationship with AMG in Section 10. After describing some details of a typical GMG method for linear finite element matrix, we argue that the geometric information used in defining a GMG is essentially the graph information of the underlying finite element grid (without using other geometric information such as coordinates of the grid points). This is a strong indication that at least some GMG methods can be realized in a purely algebraic fashion, using only the stiffness matrix, an algebraic smoother, and the adjacency graph of the stiffness matrix. On the other hand, we prove that a GMG method can also be formally obtained via our general AMG approach, presented in Section 6. Furthermore, we use the example of AMGe in Section 10.4 to demonstrate that geometric information on the grid can be used effectively to construct a geometry-based AMG.

In Section 11 we give a detailed account of AMG methods based on energy-minimization. We first present our new theory that the optimal coarse space shown in the two-level theory in Section 5 can actually be obtained by trace-minimization (Section 11.1), after which we derive the energy-min AMG method by seeking a set of locally supported coarse basis functions for energy-minimization.

Classical AMG, as the first class of AMG algorithms studied in the literature, will be presented in Section 12. We derive and analyse this type of method using the framework in Section 6 together with the notion of M-matrix relatives introduced in Section 7.2. We further discuss how a classical AMG method can be viewed as an approximation of the energy-min AMG method.

Aggregation-based AMG will be presented in Section 13. Again we derive and analyse this method using the framework in Section 6. One remarkable feature of aggregation AMG methods is their ease at preserving multidimensional near-null spaces, such as rigid body modes in linear elasticity.

To demonstrate how an AMG method addresses possible heterogeneous properties in a given problem, we devote Section 14 to showing how classical AMG is designed to address the difficulties arising from discretized elliptic problems with strong discontinuous jumps, or anisotropy in the coefficients of the underlying PDE.

In Section 15 we outline a class of AMG methods that attempt to choose the coarse spaces in a bootstrap and adaptive fashion. This class of AMG algorithms does not fall within the theoretical frameworks presented in this paper, but they provide a practical approach to generalizing many existing AMG techniques to more general problems. Finally we make some concluding remarks in Section 16.

We conclude these introductory remarks with a brief summary of the acronyms used for the various AMG algorithms reviewed in this paper.

Aggregation-based AMG:	
– unsmoothed aggregation	UA-AMG
– smoothed aggregation	SA-AMG
Bootstrap and adaptive AMG:	
– classical	αAMG
– smoothed aggregation	αSA-AMG
– bootstrap AMG	BAMG
Element-based AMG	AMGe
Spectral AMGe	ρAMGe

2. Model problems and discretization

While AMG has found applications to a wide range of linear algebraic systems, its development has been mainly motivated by the solution of systems arising from the discretization of partial differential equations by finite element, finite difference or other numerical methods. In this section we will discuss a model of second-order elliptic boundary problems, their finite difference and finite element discretization, and relevant properties of the underlying differential operators and their discretization.

2.1. Model elliptic PDE operators

We consider the boundary value problems

$$\mathcal{L}u = -\nabla \cdot \alpha(x)\nabla u = f, \quad x \in \Omega, \tag{2.1}$$

where $\alpha : \Omega \mapsto \mathbb{R}^{d\times d}$ is an SPD matrix function satisfying

$$\alpha_0 \|\xi\|^2 \le \xi^T \alpha(x)\xi \le \alpha_1\|\xi\|^2, \quad \xi \in \mathbb{R}^d, \tag{2.2}$$

for some positive constants α_0 and α_1. Here $d = 1, 2, 3$, and $\Omega \subset \mathbb{R}^d$ is a bounded domain with boundary $\Gamma = \partial\Omega$.

A variational formulation for (2.1) is as follows: find $u \in V$ such that

$$a(u, v) = (f, v), \quad \text{for all } v \in V. \tag{2.3}$$

Here

$$a(u,v) = \int_\Omega (\alpha(x)\nabla u)\cdot\nabla v, \quad (f,v) = \int_\Omega fv,$$

and V is a Sobolev space that can be chosen to address different boundary conditions accompanying equation (2.1). One case is that of mixed

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

boundary conditions

$$\begin{aligned} u &= 0, \quad x \in \Gamma_D, \\ (\alpha\nabla u)\cdot n &= 0, \quad x \in \Gamma_N, \end{aligned} \tag{2.4}$$

where $\Gamma = \Gamma_D \cup \Gamma_N$. The pure Dirichlet problem is when $\Gamma_D = \Gamma$, while the pure Neumann problem is when $\Gamma_N = \Gamma$. We thus have V as

$$V = \begin{cases} H^1(\Omega) = \{v \in L^2(\Omega) : \partial_i v \in L^2(\Omega), i = 1 : d\}, \\ H^1_D(\Omega) = \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\}. \end{cases} \tag{2.5}$$

When we consider a pure Dirichlet problem, $\Gamma_D = \Gamma$, we denote the space by $V = H^1_0(\Omega)$. In addition, for pure Neumann boundary conditions, the following condition is added to ensure the existence of the solution to (2.3):

$$\int_\Omega f = 0. \tag{2.6}$$

One of the most commonly used model problems is when

$$\alpha(x) = 1, \quad x \in \Omega, \tag{2.7}$$

which corresponds to the Poisson equation

$$-\Delta u = f. \tag{2.8}$$

This simple problem provides a good representative model for isotropic problems.

There are two other cases that are of special interest. The first case is when α is a scalar and has discontinuous jumps such as

$$\alpha(x) = \begin{cases} \epsilon & x \in \Omega_1, \\ 1 & x \in \Omega_2. \end{cases} \tag{2.9}$$

The second case is when α is a diagonal matrix such as (for $d = 2$)

$$\alpha(x) = \begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix}, \tag{2.10}$$

which corresponds to the operator

$$-u_{xx} - \epsilon u_{yy} = f. \tag{2.11}$$

In both cases we assume that ϵ is sufficiently small to investigate the robustness of algorithms with respect to discontinuous jumps and anisotropy.

2.2. Examples of finite difference and finite element discretizations

As an illustrative example, we consider a finite difference discretization of the Poisson equation (2.8) with pure Dirichlet boundary conditions on the

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

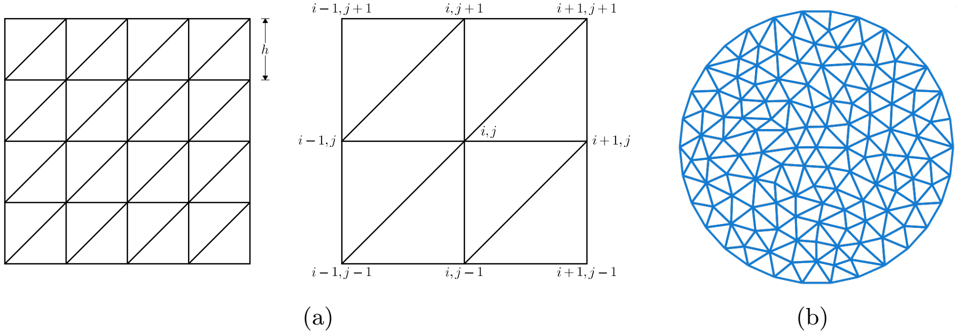


Figure 2.1. (a) Regular (uniform) triangulations for the unit square, and (b) unstructured mesh approximating the unit disk.

unit square $\Omega = (0,1) \times (0,1)$. We consider a uniform triangulation of Ω (see Figure 2.1(a)), and we set

$$(x_i, y_j) = \left(\frac{i}{n+1}, \frac{j}{n+1}\right), \quad u_{i,j} \approx u(x_i, y_j), \quad (i, j = 0, \dots, n+1).$$

We use the standard central difference approximation to the Laplacian operator

$$(-\Delta u)(x_i, y_j) \approx \frac{4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1}}{h^2}.$$

The finite difference scheme is then given by

$$4u_{i,j} - (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) = h^2 f_{i,j}, \tag{2.12}$$

where

$$f_{i,j} = f(x_i, y_j) \tag{2.13}$$

and $u_{i,j} \approx u(x_i, y_j)$. The approximations $u_{i,j}$ are found by solving a linear system. We order the points (x_i, y_j) lexicographically, and for $k = 1, \dots, n^2$ we have

$$k = (j-1)n + i, \quad x_k^h = (x_i, y_j), \quad \mu_k = u_{i,j}, \quad 1 \le i, j \le n, \tag{2.14}$$

We can then write (2.12) as

$$A\mu = b, \tag{2.15}$$

where

$$A = \text{tridiag}(-I, B, -I) \quad \text{and} \quad B = \text{tridiag}(-1, 4, -1). \tag{2.16}$$

A slightly different scheme is obtained using more of the neighbouring points of (x_i, y_j). We can build an approximation using 8 points $(x_{i\pm1}, y_{j\pm1})$ together with the 'centre' point (x_i, y_j). As a result we have the following

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

9-point finite difference scheme:

$$\begin{aligned}8\mu_{i,j}-\mu_{i-1,j}-\mu_{i+1,j}-\mu_{i,j-1}-\mu_{i,j+1}&\\ -\mu_{i-1,j-1}-\mu_{i+1,j-1}-\mu_{i-1,j+1}-\mu_{i+1,j+1}&=2h^2f_{i,j}.\end{aligned}\tag{2.17}$$

Again, if we order (x_i,y_j) lexicographically, then (2.17) is the linear system (2.15) corresponding to the 9-point finite difference discretization of the Laplace equation, where

$$A=\operatorname{tridiag}(-C,B,-C),\quad B=\operatorname{tridiag}(-1,8,-1),\ C=\operatorname{tridiag}(1,1,1).\tag{2.18}$$

We now give an example of finite element discretization. Given a triangulation $\mathcal{T}_h$ for Ω, such as that given in Figure 2.1, let $V_h\subset V$ be a finite element space consisting of piecewise linear (or higher-order) polynomials with respect to the triangulation $\mathcal{T}_h$. The finite element approximation of the variational problem (2.3) is: find $u_h\in V_h$ such that

$$a(u_h,v_h)=(f,v_h),\quad \text{for all } v_h\in V_h.\tag{2.19}$$

Assume $\{\phi_i\}_{i=1}^N$ is the nodal basis of V_h, namely $\phi_i(x_j)=\delta_{ij}$ for any nodes x_j. We write $u_h(x)=\sum_{j=1}^N\mu_j\phi_j(x)$. Equation (2.19) is then equivalent to

$$\sum_{j=1}^N\mu_ja(\phi_j,\phi_i)=(f,\phi_i),\quad j=1,2,\ldots,N,$$

which is a linear system of equations:

$$A\mu=b,\quad (A)_{ij}=a(\phi_j,\phi_i)\quad\text{and}\quad (b)_i=(f,\phi_i).\tag{2.20}$$

The matrix A is known as the stiffness matrix of the nodal basis $\{\phi_i\}_{i=1}^N$.

For $d=2$ and the special uniform triangulation shown in Figure 2.1, this stiffness matrix for the Laplacian operator turns out to be exactly the one given by (2.12). This special case is an example of the close relationship between finite difference and finite element methods.

We note that the finite element method is based on the variational formulation (2.3), whereas the finite difference method is not. In the development of the AMG method, however, the variational method is also used to derive coarse level equations for finite difference methods.

For any $T\in\mathcal{T}_h$, we define

$$\overline{h}_T=\operatorname{diam}(T),\quad h_T=|T|^{1/d},\quad \underline{h}_T=2\sup\{r>0: B(x,r)\subset T\text{ for }x\in T\}.\tag{2.21}$$

We say that the mesh $\mathcal{T}_h$ is *shape-regular* if there exists a uniformly bounded constant $\sigma\geq 1$ such that

$$\underline{h}_T\leq h_T\leq\overline{h}_T\leq\sigma\underline{h}_T,\quad\text{for all } T\in\mathcal{T}_h.\tag{2.22}$$

We call σ the *shape-regularity constant.*

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Let $h = \max_{T\in\mathcal{T}_h} \overline{h}_T$, with $\overline{h}_T$ defined as in (2.21). We say that the mesh $\mathcal{T}_h$ is *quasi-uniform* if there exists a uniformly bounded constant $C > 0$ such that

$$\frac{h}{\underline{h}_T} \leq C. \tag{2.23}$$

2.3. Spectral properties

We now discuss the spectral properties of the partial differential operator $\mathcal{L}$ given in (2.1).

We recall the well-known Courant–Fischer min-max principle (Courant and Hilbert 1924) for eigenvalues of symmetric matrices.

Theorem 2.1. Let T be an $n \times n$ symmetric matrix with respect to $(\cdot,\cdot)_*$, and let $\{\lambda_j, \zeta_j\}$ be its eigenpairs with $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Then

$$\lambda_k = \min_{\dim W = k} \max_{x\in W, x\neq 0} \frac{(Tx,x)_*}{(x,x)_*}, \tag{2.24}$$

where the minimum is achieved if

$$W = \operatorname{span}\{\zeta_j : j = 1 : k\}, \tag{2.25}$$

and

$$\lambda_k = \max_{\dim W = n-k+1} \min_{x\in W, x\neq 0} \frac{(Tx,x)_*}{(x,x)_*}, \tag{2.26}$$

where the maximum is achieved if

$$W = \operatorname{span}\{\zeta_j : j = k : n\}. \tag{2.27}$$

Next, we recall Theorem 1 in Fan (1949), which is known as the Ky–Fan trace-minimization principle.

Theorem 2.2. We assume T is symmetric with respect to $(\cdot,\cdot)_*$, and $\{\lambda_j, \zeta_j\}$ are its eigenpairs with $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Then

$$\min_{P\in\mathbb{R}^{n\times k}, P^*P=I} \operatorname{trace}(P^*TP) = \sum_{j=1}^{k} \lambda_j.$$

Furthermore, the minimum is achieved when

$$\operatorname{range}(P) = \operatorname{span}\{\zeta_j\}_{j=1}^k \quad \text{and} \quad P^*P = I.$$

Here $P^* \in \mathbb{R}^{k\times n}$ is the adjoint of P with respect to the $(\cdot,\cdot)_*$-inner product, namely

$$(P^*u, v) = (u, Pv)_*, \quad \text{for all } u \in \mathbb{R}^n,\ v \in \mathbb{R}^k.$$

Finally, following Xu (1992), we use the notation $a \lesssim b$ to represent the existence of a generic positive constant C, which is independent of salient

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

parameters, such as problem size or anisotropic ratio, and such that $a \leq Cb$. Furthermore, we write $a \eqsim b$ if and only if $a \lesssim b$ and $b \lesssim a$.

Theorem 2.3. The PDE operator $\mathcal{L}$ has a complete set of eigenfunctions (φ_k) and non-negative eigenvalues

$$0 \leq \lambda_1 \leq \lambda_2 \leq \cdots$$

such that

$$\mathcal{L}\varphi_k = \lambda_k \varphi_k, \quad k = 1, 2, 3 \ldots,$$

where

(a) $\lim_{k\to\infty} \lambda_k = \infty$,

(b) (φ_i) forms an orthonormal basis of V as well as for $L^2(\Omega)$.

Furthermore,

(c) for the pure Neumann problem, $\lambda_1 = 0$, and φ_1 is the constant function,

(d) for the pure Dirichlet problem, $\lambda_1 > 0$ is simple, and φ_1 does not change sign.

We have the well-known Weyl estimate for the asymptotic behaviour of the Laplacian operator (Weyl 1911, Weyl 1912, Reed and Simon 1978).

Lemma 2.4 (Weyl's law). Assume that Ω is 'contented', which means that Ω can be approximated by unions of cubes in $\mathbb{R}^d$; see Reed and Simon (1978, p. 271) for the exact definition. For the homogeneous Dirichlet boundary condition, the eigenvalues of the pure Laplacian operator satisfy

$$\lim_{k\to\infty} \frac{\lambda_k}{k^{2/d}} = w_\Omega, \quad \text{with } w_\Omega = \frac{(2\pi)^2}{[\omega_d \operatorname{Vol}(\Omega)]^{2/d}}, \tag{2.28}$$

where ω_d is the volume of the unit ball in $\mathbb{R}^d$, and the eigenvalues of the operator $\mathcal{L}$ given in (2.1) satisfy

$$\lambda_k \eqsim k^{2/d}, \quad \text{for all } k \geq 1. \tag{2.29}$$

Next, we extend Weyl's law to discretized PDE operators. The following theorem gives a discrete version of Weyl's law for finite element discretization. Further details of this result can be found in Xu, Zhang and Zikatanov (2016*b*).

Theorem 2.5. Let $V_h \subset H_0^1(\Omega)$ be a family of finite element spaces on a quasi-uniform mesh with $\dim V_h = N$. Consider the discretized operator of (2.1),

$$\mathcal{L}_h : V_h \mapsto V_h, \quad (\mathcal{L}_h u, v) = a(u, v), \quad \text{for all } u, v \in V_h,$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

and its eigenvalues,

$$\lambda_{h,1} \leq \lambda_{h,2} \leq \cdots \leq \lambda_{h,N}.$$

Then, for all $1 \leq k \leq N$, there exists a constant $C_w > 0$ independent of k such that we have the estimates

$$\lambda_k \leq \lambda_{h,k} \leq C_w \lambda_k \tag{2.30}$$

and

$$\lambda_{h,k} \eqsim k^{2/d}. \tag{2.31}$$

2.4. Properties of finite element matrices

The main algebraic property for the stiffness matrices given by (2.20) is that it is sparse with $O(N)$ non-zeros, symmetric positive definite (for both Dirichlet and mixed boundary conditions) and semi-definite for pure Neumann boundary conditions. Its eigenvalues satisfy a discrete Weyl's law.

For simplicity, we will only consider pure Dirichlet boundary conditions in the rest of this section.

Lemma 2.6. The stiffness matrix A given by (2.20) has the following properties.

(a) The condition number of A, defined by the ratio of the extreme eigenvalues of A,

$$\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)},$$

satisfies

$$\kappa(A) \eqsim h^{-2}.$$

Furthermore,

$$\lambda_{\min}(A) \eqsim h^2 \quad \text{and} \quad \lambda_{\max}(A) \eqsim 1.$$

(b) The discrete version of Weyl's law holds:

$$\lambda_k(A) \eqsim \left(\frac{k}{N}\right)^{2/d}.$$

We next discuss some more refined spectral properties of finite element stiffness matrices from uniform grids for the unit square domain $\Omega = (0,1) \times (0,1)$ for $d = 2$. We begin with the Poisson equation. It is easy to derive a closed-form solution of the eigenpairs of A given by (2.15), and we have

$$\lambda_{kl}(A) = 4\left(\sin^2 \frac{k\pi}{2(n+1)} + \sin^2 \frac{l\pi}{2(n+1)}\right) \tag{2.32}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

and

$$\phi_{ij}^{kl} = \sin\frac{ki\pi}{n+1}\sin\frac{lj\pi}{n+1}, \quad 1 \le i \le n,\ 1 \le j \le n. \tag{2.33}$$

Now consider the important case of anisotropic problem (2.11). We order the vertices of the triangulation lexicographically, and, as before, denote them by $\{(ih, jh)\}_{i,j=0}^{n}$. The stiffness matrix is then

$$A = \operatorname{tridiag}(-I, B, -I) \quad \text{with } B = \operatorname{tridiag}(-\epsilon, 2(1+\epsilon), -\epsilon). \tag{2.34}$$

Obviously,

$$A = I \otimes B + C \otimes I \quad \text{with } C = \operatorname{tridiag}(-1, 0, -1),$$

and it is easily verified that

$$\lambda_i(B) = 2(1+\epsilon) - 2\epsilon\cos\frac{i\pi}{(n+1)}, \quad \lambda_j(C) = -2\cos\frac{j\pi}{(n+1)}, \quad 1 \le i, j \le N.$$

which leads to the expression

$$\lambda_{ij}(A) = 4\epsilon\sin^2\frac{i\pi}{2(n+1)} + 4\sin^2\frac{j\pi}{2(n+1)}$$

for the eigenvalues, and

$$\phi_{ij}^{k\ell} = \sin\frac{ki\pi}{n+1}\sin\frac{\ell j\pi}{n+1}$$

for the corresponding eigenvectors.

3. Linear vector spaces and duals

In this paper we will mainly consider linear systems of equations of the form

$$Au = f. \tag{3.1}$$

Here

$$A : V \mapsto V', \tag{3.2}$$

$$f \in V',$$

where V is a finite-dimensional linear vector space and V' is the dual of V. If we use the notation $\langle\cdot,\cdot\rangle$ to denote the pairing between V' and V, we can write (3.1) in a variational form: find $u \in V$ such that

$$a(u, v) = \langle f, v\rangle, \quad \text{for all } v \in V, \tag{3.3}$$

where

$$a(u, v) = \langle Au, v\rangle. \tag{3.4}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

3.1. Dual and inner product

For convenience of exposition, we will assume that V is equipped with an inner product $(\cdot,\cdot)$. By the Riesz representation theorem, for any $f \in V'$, there is a unique $u \in V$ such that

$$(u,v) = \langle f, v\rangle, \quad \text{for all } v \in V. \tag{3.5}$$

It is via the Riesz representation theorem that we identify the dual space V' with V. In the rest of this paper, for convenience, we will always assume that $V' = V$ for any finite-dimensional vector space V. As a result, we have

$$V'' = (V')' = V' = V. \tag{3.6}$$

Thanks to the identification $V' = V$ via (3.5), the identities in (3.6) are clear.

We would like to point out that, in an abstract discussion of all iterative methods for problem (3.1), it suffices to use the abstract dual pairing $\langle\cdot,\cdot\rangle$ without having to introduce an inner product $(\cdot,\cdot)$ on V. However, we find that using an inner product is convenient for exposition, as we shall see later. We further point out that we will not use an inner product to identify $V' = V$ for any infinite-dimensional vector space in this paper.

If $\{\phi_i\}_{i=1}^N$ is a basis of V, we will always choose a basis $\{\psi_i\}_{i=1}^N$ of V' that is dual to the basis of V, namely

$$(\psi_j, \phi_i) = \delta_{ij}, \quad 1 \le i,\ j \le N. \tag{3.7}$$

Such a dual basis will be used only for theoretical considerations and not for the actual implementation of any algorithms.

We will consider two kinds of linear vector spaces. The first kind is $V = \mathbb{R}^n$ and the inner product is just the dot product

$$(u,v)_{\ell^2} = \sum_{i=1}^n u_i v_i, \quad \text{for all } u = (u_i),\ v = (v_i) \in \mathbb{R}^n,$$

A canonical basis of $\mathbb{R}^N$ is formed by the column vectors of the identity matrix, $\{e_i\}_{i=1}^N$. It is easy to see that the dual basis of $(\mathbb{R}^N)' = \mathbb{R}^N$ is just the original basis $\{e_i\}_{i=1}^N$ itself.

The second kind is a finite-dimensional functional subspace of $L^2(\Omega)$ for a given domain $\Omega \subset \mathbb{R}^d$ $(1 \le d \le 3)$, equipped with the L^2-inner product

$$(u,v) = \int_\Omega u(x)v(x).$$

One commonly used linear vector space is a finite element space V_h, and the nodal basis functions $\{\phi_i\}_{i=1}^N$ are often used as a basis. In this case, the dual basis functions $\{\psi_i\}_{i=1}^N$ of V' are no longer the original nodal basis functions but rather a set of functions (which are usually globally supported) that

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

satisfy (3.7). This set of dual basis functions is usually needed to derive the matrix representation of operators between various spaces and their duals, but they are not needed for the actual implementation of relevant algorithms.

For a linear operator

$$L : V \mapsto V, \tag{3.8}$$

its adjoint,

$$L' : V \mapsto V, \tag{3.9}$$

is defined as follows:

$$(L'u, v) = (u, Lv), \quad u, v \in V. \tag{3.10}$$

Since V plays the role of both V and its dual V', (3.8) and (3.9) can have four different meanings:

(i) if $L : V \mapsto V$, then $L' : V' \mapsto V'$,

(ii) if $L : V \mapsto V'$, then $L' : V \mapsto V'$,

(iii) if $L : V' \mapsto V$, then $L' : V' \mapsto V$,

(iv) if $L : V' \mapsto V'$, then $L' : V \mapsto V$.

Thanks to the identification we made between V' and V via (3.5), definition (3.10) is applicable to all four cases.

If $V = \mathbb{R}^n$ and $(u, v) = (u, v)_{\ell^2}$, then $L' = L^T$, namely the matrix transpose. We say that an operator $A : V \mapsto V'$ is symmetric positive definite (SPD) if

$$A' = A, \quad (Av, v) > 0, \quad \text{for all } v \in V \setminus \{0\}.$$

When A is SPD, it defines another inner product $(\cdot, \cdot)_A$ on V,

$$(u, v)_A = (Au, v), \quad u, v \in V,$$

and a corresponding norm,

$$\|v\|_A = (v, v)_A^{1/2}, \quad v \in V.$$

We use the superscript '*' for the adjoint operator with respect to $(\cdot, \cdot)_A$, that is,

$$(Bu, v)_A = (u, B^* v)_A.$$

It is easy to see that

$$(BA)^* = B'A, \tag{3.11}$$

and $(BA)^* = BA$ if and only if $B' = B$.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

3.2. Matrix representation

Let $V_c \subset V$ be a subspace and consider the inclusion operator $\imath_c : V_c \mapsto V$. Assume that $\{\phi_i^c\}_{i=1}^{n_c}$ and $\{\phi_i\}_{i=1}^{n}$ are basis functions of V_c and V respectively, and the matrix representation of $\imath_c$ is the matrix

$$P : \mathbb{R}^{n_c} \mapsto \mathbb{R}^n \text{ satisfying } (\phi_1^c, \ldots, \phi_{n_c}^c) = (\phi_1, \ldots, \phi_n)P. \tag{3.12}$$

Equation (3.12) is shorthand for the expansion of the basis in V_c via the basis in V:

$$\phi_k^c = \sum_{j=1}^{n} p_{jk}\phi_j, \quad P = (p_{jk}), \quad k = 1, \ldots, n_c, \; j = 1, \ldots, n. \tag{3.13}$$

What is the matrix representation of $\imath_c' : V' \mapsto V_c'$? Although we have $V' = V$ and $V_c' = V_c$, we need to use dual bases $\{\psi_i^c\}_{i=1}^{n_c} \subset V_c'$ and $\{\psi_i\}_{i=1}^{n} \subset V'$ respectively. With respect to these dual bases, the matrix representation of $\imath_c'$ is simply P^T (the transpose of P), since it is easy to verify that

$$(\imath_c'\psi_1, \ldots, \imath_c'\psi_n) = (\psi_1^c, \ldots, \psi_{n_c}^c)P^T.$$

Now consider a linear operator

$$A : V \mapsto V. \tag{3.14}$$

There are two different ways to get a matrix representation of A, because V plays two roles here: V is V itself, and $V = V'$. For the first case, we use the same basis $\{\phi_i\}$ for V as the domain of A and V as the range of A. In this case, the matrix representation of A is the matrix

$$\hat{A} \in \mathbb{R}^{n \times n} \text{ satisfying } (A\phi_1, \ldots, A\phi_n) = (\phi_1, \ldots, \phi_n)\hat{A}. \tag{3.15}$$

In the second case, we use the basis $\{\phi_i\}$ for V as the domain space of A, but use the dual basis $\{\psi_i\}$ for $V' = V$ as the range space of A. In this case, the matrix representation of A is the matrix

$$\tilde{A} \in \mathbb{R}^{n \times n} \text{ satisfying } (A\phi_1, \ldots, A\phi_n) = (\psi_1, \ldots, \psi_n)\tilde{A}. \tag{3.16}$$

It is easy to see that

$$\tilde{A} = ((A\phi_j, \phi_i)) \tag{3.17}$$

and

$$\tilde{A} = M\hat{A}, \quad M = ((\phi_j, \phi_i)). \tag{3.18}$$

The matrix $\tilde{A}$ in (3.17) is often called the stiffness matrix of A and the matrix M in (3.18) is called the mass matrix.

In the early multigrid literature, a discrete inner product equivalent to the L^2-inner product was often introduced for finite element spaces so that the corresponding mass matrix becomes diagonal. However, if we view the

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

underlying finite element operator (3.14) in a slightly different way,

$$A : V \mapsto V', \tag{3.19}$$

we will then see easily that the introduction of the discrete L^2-inner product is not necessary.

If $V = \mathbb{R}^n$ and we choose the canonical basis $\{e_i\}$ for V, we would not encounter the mass matrix problem as in the functional space case, since in this case $\{e_i\}$ is also the dual basis of V'. This is certainly convenient, but the convenience tends to mask some subtle but important difference between various vectors and matrices in a given problem and the objects (functions) that they represent.

Given a matrix $A \in \mathbb{R}^{n\times n}$, we can view it as either

$$A : \mathbb{R}^n \mapsto \mathbb{R}^n \tag{3.20}$$

or

$$A : \mathbb{R}^n \mapsto (\mathbb{R}^n)'. \tag{3.21}$$

As it turns out, when A is obtained from the discretization of partial differential equations, (3.21) is more informative than (3.20). Hence we write a matrix equation

$$Ax = b. \tag{3.22}$$

It is sometimes helpful to regard x and b as living in two 'different' spaces,

$$x \in \mathbb{R}^n \quad \text{and} \quad b \in (\mathbb{R}^n)'. \tag{3.23}$$

3.3. Eigenvalues and eigenvectors

Let us briefly discuss eigenvalues and eigenvectors for the symmetric operator $T : V \mapsto V$. If (λ, ϕ) is an eigenpair of T,

$$T\phi = \lambda\phi,$$

then it is easy to see that $(\lambda, \tilde{\phi})$ is an eigenpair of the matrix representation $\tilde{T}$ of T:

$$\tilde{T}\tilde{\phi} = \lambda\tilde{\phi}.$$

Here $\tilde{\phi} \in \mathbb{R}^n$ is the vector representation of ϕ:

$$\phi = (\phi_1, \dots, \phi_n)\tilde{\phi}.$$

For an operator A defined in (3.14), we need to be cautious when talking about eigenvalues of A. Although we identify $V' = V$ via (3.5), V' and V play two different roles, so A is essentially a mapping between two 'different' spaces V and V', and the spectrum of A should be defined carefully. However, if we consider a symmetric operator

$$R : V' \mapsto V, \tag{3.24}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

then $RA : V \mapsto V$ is an operator that is symmetric with respect to the A-inner product. In this case, if (λ, ϕ) is an eigenpair of RA, then $(\lambda, \tilde{\phi})$ is an eigenpair of $\tilde{R}\tilde{A}$ (*i.e.* equal to the matrix representation of RA).

We consider the trivial identification operator

$$\jmath : V' \mapsto V \text{ such that } \jmath\psi_i = \psi_i, \text{ for all } i,$$

namely $\jmath v = v$ for all $v \in V' = V$. It is easy to see that the matrix representation of $\jmath$ is the inverse of the mass matrix $M = ((\phi_j, \phi_i))$, namely

$$\tilde{\jmath} = M^{-1}.$$

Using this identification operator, the operator $\jmath A : V \mapsto V$ is a symmetric operator from V to V. We can then talk about its spectrum. For example, if (λ, ϕ) is an eigenpair of $\jmath A$, then $(\lambda, \tilde{\phi})$ satisfies

$$\tilde{A}\tilde{\phi} = \lambda M \tilde{\phi}.$$

This is often the generalized eigenvalue problem appearing in finite element analysis.

Although, for all $v \in V$, $\jmath A v = Av$ because of the identification introduced above, $\jmath A$ and A are, strictly speaking, two different operators: they have two different ranges and their matrix representations are different.

The discussions above, although simple, may seem a little confusing at first glance, but an unambiguous understanding and clarification of these concepts and the underlying subtleties will be helpful for the presentation of algebraic multigrid methods in the rest of this paper. For more detailed discussions on relevant topics, we refer to Xu (2016).

3.4. Bibliographical notes

For a general reading on the basic linear algebra materials used here, we refer to Halmos (1974) and Xu (1992, 2016). In particular, for a more detailed discussion related to dual spaces and matrix representations, we refer to Xu (1992, 2016).

4. Basic iterative methods

We now consider linear iterative methods for solving (2.20). We will focus on two of the most commonly used algorithms, namely the Jacobi and Gauss–Seidel methods. Let us first give a brief introduction to linear iterative methods in a more general setting. Recall the basic problem under consideration. Given a finite-dimensional vector space V equipped with an inner product $(\cdot, \cdot)$, we consider

$$Au = f, \tag{4.1}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

where $A : V \mapsto V'$ is symmetric positive definite (SPD) and V' is the dual of V. As mentioned in Section 3, we will identify $V' = V$ via an inner product $(\cdot, \cdot)$.

4.1. Basic iterative methods

A general linear iterative method for solving (4.1) can be written as follows. Given $u^0 \in V$, we define

$$u^m = u^{m-1} + B(f - Au^{m-1}), \quad m = 1, 2, \ldots, \tag{4.2}$$

where $B : V' \mapsto V$ is a linear operator which can be thought of as an approximate inverse of A.

Sometimes it is more desirable for the iterator B to be symmetric. If B is not symmetric, there is a natural way to symmetrize it. Consider the following iteration:

$$\begin{cases} u^{m-1/2} = u^{m-1} + B(f - Au^{m-1}), \\ \quad\;\; u^m = u^{m-1/2} + B'(f - Au^{m-1/2}). \end{cases} \tag{4.3}$$

The symmetrized iteration (4.3) can be written as

$$u^m = u^{m-1} + \bar{B}(f - Au^{m-1}), \quad m = 1, 2, \ldots, \tag{4.4}$$

where

$$\bar{B} = B' + B - B'AB, \tag{4.5}$$

which satisfies

$$I - \bar{B}A = (I - BA)^*(I - BA). \tag{4.6}$$

Obviously, $\rho(I - \bar{B}A) < 0 \Longleftrightarrow \bar{B} > 0 \Longleftrightarrow G \equiv (B')^{-1} + B^{-1} - A > 0$.

Theorem 4.1. The following results hold.

(a) Equation (4.3) converges if and only if $G > 0$, which implies that (4.2) converges. Furthermore,

$$\|I - BA\|_A^2 = \lambda_{\max}(I - \bar{B}A) = 1 - \left(\sup_{\|v\|_A = 1} (\bar{B}^{-1}v, v) \right)^{-1}. \tag{4.7}$$

(b) If $B' = B$, $G > 0$ if and only if (4.2) converges and, with $\eta = \lambda_{\min}(G)$,

$$\frac{2\eta}{\eta + 1}(Bv, v) \le (\bar{B}v, v) \le 2(Bv, v), \quad v \in V. \tag{4.8}$$

Remark 4.2. As an exercise, the reader may generalize Theorem 4.1 to the case when A is SPSD.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Algorithm 1 Modified Jacobi method

```
For $i = 1 : n$,    $x_i^m \leftarrow x_i^{m-1} + \omega a_{ii}^{-1}\Big(b_i - \sum_{j=1}^{n} a_{ij}x_j^{m-1}\Big)$.
```

Algorithm 2 Modified Gauss–Seidel method

```
For $i = 1 : n$,    $x_i^m \leftarrow x_i^{m-1} + \omega a_{ii}^{-1}\Big(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{m} - \sum_{j=i}^{n} a_{ij}x_j^{m-1}\Big)$.
```

4.2. Jacobi and Gauss–Seidel methods

For $A = (a_{ij}) \in \mathbb{R}^{n\times n}$, we write

$$A = D + L + U,$$

where D is the diagonal of A, and L and U are the strict lower and upper triangular parts of A respectively.

Given $\omega > 0$, the (modified) Jacobi method can be written as (4.2) with

$$B = \omega D^{-1} = (\omega^{-1}D)^{-1},$$

as shown in Algorithm 1. The (modified) Gauss–Seidel method can be written as (4.2) with

$$B = (\omega^{-1}D + L)^{-1},$$

as shown in Algorithm 2.

The next result, which follows easily from Theorem 4.1, is well known.

Theorem 4.3. The modified Jacobi method converges if and only

$$0 < \omega < \frac{2}{\rho(D^{-1}A)}, \tag{4.9}$$

and the modified Gauss–Seidel method converges if and only if

$$0 < \omega < 2. \tag{4.10}$$

In practice, it is often easy to choose ω to satisfy (4.9) so that the modified Jacobi method is guaranteed to converge. In the rest of this paper, we may always assume that such a choice of ω is made. For the Gauss–Seidel method, we will always choose $\omega = 1$ (optimal successive over-relaxation is not generally used in multigrid methods). The Jacobi and Gauss–Seidel methods together with their convergence theory can be extended to block matrices in a straightforward fashion.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

4.3. The subspace correction method

We consider a sequence of spaces $V_1, \dots, V_J$. These spaces, which will be known as *auxiliary spaces*, are not necessarily subspaces of V, but each of them is related to the original space V by a linear operator

$$\Pi_k : V_k \mapsto V. \tag{4.11}$$

Our very basic assumption is that the following decomposition holds:

$$V = \sum_{i=1}^{J} \Pi_i V_i. \tag{4.12}$$

This means that for any $v \in V$ there exists $v_i \in V_i$ (which may not be unique) such that

$$v = \sum_{i=1}^{J} \Pi_i v_i. \tag{4.13}$$

Furthermore, we assume that each V_i is equipped with an energy inner product $a_i(\cdot,\cdot)$. We define

$$A_i : V_i \mapsto V_i'$$

by

$$(A_i u_i, v_i) = a_i(u_i, v_i), \quad u_i,\ v_i \in V_i.$$

Let $\Pi_i' : V' \mapsto V_i'$ be the adjoint of Π_i:

$$(\Pi_i' f, v_i) = (f, \Pi_i v_i), \quad f \in V',\ v_i \in V_i.$$

Let $P_i = \Pi_i^* : V \mapsto V_i$ be the adjoint of Π_i with respect to the A-inner products:

$$(P_i u, v_i)_{A_i} = (u, \Pi_i v_i)_A, \quad u \in V,\ v_i \in V_i.$$

The following identity holds:

$$\Pi_i' A = A_i P_i. \tag{4.14}$$

If u is the solution of (4.1), by (4.14), we have

$$A_i u_i = f_i, \tag{4.15}$$

where

$$u_i = P_i u, \quad f_i = \Pi_i' f.$$

This equation may be regarded as the restriction of (4.1) to V_i. We assume that each such A_i has an approximate inverse or preconditioner:

$$R_i : V_i' \mapsto V_i. \tag{4.16}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Algorithm 3 Successive subspace correction method

```
Given $u^0 \in V$, for any $m = 1, 2, \ldots,$
1: $v \leftarrow u^{m-1}$
2: $v \leftarrow v + \Pi_i R_i \Pi_i'(f - Av)$, $i = 1, 2, \ldots, J$,
3: $u^m \leftarrow v$
```

The *parallel subspace correction* (PSC for short) method is (4.2) with $B = B_{\text{psc}}$ given by

$$B_{\text{psc}} = \sum_{i=1}^{J} \Pi_i R_i \Pi_i'. \tag{4.17}$$

The *successive subspace correction* (SSC for short) method is defined in Algorithm 3.

The algorithm is equivalent to (4.2) with $B = B_{\text{ssc}}$ given by

$$I - B_{\text{ssc}} A = (I - T_J)(I - T_{J-1}) \cdots (I - T_1), \tag{4.18}$$

where

$$T_i = \Pi_i R_i \Pi_i' A = \Pi_i R_i A_i P_i. \tag{4.19}$$

Theorem 4.4. Assume that all R_k are SPD. Then

$$(B_{\text{psc}}^{-1} v, v) = \min_{\sum_i \Pi_i v_i = v} \sum_{k=1}^{J} (R_k^{-1} v_k, v_k), \tag{4.20}$$

with the unique minimizer given by

$$v_k^* = R_k \Pi_k' B_{\text{psc}}^{-1} v. \tag{4.21}$$

Theorem 4.5. Under the assumptions given above, we obtain the identity

$$\begin{aligned} \|I - B_{\text{ssc}} A\|_A^2 &= \|(I - T_J)(I - T_{J-1}) \cdots (I - T_1)\|_A^2 \\ &= 1 - \frac{1}{1 + c_0} \qquad (4.22) \\ &= 1 - \frac{1}{c_1}. \qquad (4.23) \end{aligned}$$

Here

$$c_0 = \sup_{\|v\|_A = 1} c_0(v), \quad c_1 = \sup_{\|v\|_A = 1} c_1(v) = 1 + c_0,$$

and, with

$$w_i = (I - T_i^{-1}) \Pi_i v_i + \sum_{j=i+1}^{J} \Pi_j v_j,$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

we obtain

$$c_0(v) = \inf_{\sum_i \Pi_i v_i = v} \sum_{i=1}^{J} (T_i \overline{T}_i^{-1} T_i^* w_i, w_i)_A, \tag{4.24}$$

and

$$\begin{aligned} c_1(v) &= (\overline{B}_{\rm ssc}^{-1} v, v) \\ &= \inf_{\sum_i \Pi_i v_i = v} (\overline{T}_i^{-1}(\overline{T}_i T_i^{-1} \Pi_i v_i + T_i^* w_i), (\overline{T}_i T_i^{-1} \Pi_i v_i + T_i^* w_i))_A. \end{aligned} \tag{4.25}$$

In particular, if $R_i = A_i^{-1}$, then

$$c_0(v) = \inf_{\sum_i \Pi_i v_i = v} \sum_{i=1}^{J} \biggl\| P_i \sum_{j=i+1}^{J} \Pi_j v_j \biggr\|_{A_i}^2 \tag{4.26}$$

and

$$c_1(v) = \inf_{\sum_i \Pi_i v_i = v} \sum_{i=1}^{J} \biggl\| P_i \sum_{j=i}^{J} \Pi_j v_j \biggr\|_{A_i}^2. \tag{4.27}$$

Lemma 4.6. If $R_k = A_k^{-1}$ for all k, and V_k are subspaces of V, then

$$\frac{1}{4}(B_{\rm psc}^{-1} v, v) \leq (\bar{B}_{\rm ssc}^{-1} v, v) \leq c^*(B_{\rm psc}^{-1} v, v), \quad v \in V, \tag{4.28}$$

where

$$c^* = \max_{1 \leq k \leq M} [N(k)]^2 \text{ and } N(k) = \bigl\{ j \in \{1, \dots, J\} : V_j \cap V_k \neq \{0\} \bigr\}.$$

Proof. Given $v = \sum_{i=1}^{J} v_i$, with $v_i \in V_i$, it follows that

$$\begin{aligned} \|v\|_A^2 &= \sum_{k,j=1}^{J} (v_k, v_j)_A = \sum_{k=1}^{J} (v_k, v_k)_A + 2 \sum_{j>k}^{J} (v_k, v_j)_A \\ &= -\sum_{k=1}^{J} (v_k, v_k)_A + 2 \sum_{j \geq k}^{J} (v_k, v_j)_A. \end{aligned}$$

Thus

$$\begin{aligned} \sum_{k=1}^{J} \|v_k\|_A^2 &\leq 2 \sum_{k=1}^{J} \biggl(v_k, \sum_{j=k}^{J} v_j \biggr)_A = 2 \sum_{k=1}^{J} \biggl(v_k, P_k \sum_{j=k}^{J} v_j \biggr)_A \\ &\leq 2 \biggl(\sum_{k=1}^{J} \|P_k \sum_{j=k}^{J} v_j\|_A^2 \biggr)^{1/2} \biggl(\sum_{k=1}^{J} \|v_k\|_A^2 \biggr)^{1/2}. \end{aligned}$$

Consequently,

$$\sum_{k=1}^{J} \|v_k\|_A^2 \le 4 \sum_{k=1}^{J} \left\| P_k \sum_{j=k}^{J} v_j \right\|_A^2 .$$

By (4.20), (4.27) and (4.25), we have

$$(B_{\rm psc}^{-1} v, v) \le 4c_1(v) = 4(\bar{B}_{\rm ssc}^{-1} v, v).$$

The upper bound also follows easily. From $\|P_k\|_A = 1$ and the Schwarz inequality, we obtain

$$\begin{aligned}\sum_{k=1}^{J} \left\| P_k \sum_{j=k}^{J} v_j \right\|_A^2 &= \sum_{k=1}^{J} \left\| P_k \sum_{j\in N(k);j\ge k} v_j \right\|_A^2 \\ &\le \sum_{k=1}^{J} \left\| \sum_{j\in N(k);j\ge k} v_j \right\|_A^2 \le \sum_{k=1}^{J} N(k) \sum_{j\in N(k);j\ge k} \|v_j\|_A^2 \\ &\le \sqrt{c^*} \sum_{k=1}^{J} \sum_{j\in N(k);j\ge k} \|v_j\|_A^2 \le c^* \sum_{k=1}^{J} \|v_k\|_A^2 .\end{aligned}$$

The proof is concluded by taking the infimum over all decompositions on both sides and applying (4.27). □

Remark 4.7. We would like to point that the estimate in (4.28) holds for anisotropic and jump coefficient problems, and the constant c^* only depends on the 'topology' of the overlaps between the subspaces and does not depend on other ingredients and properties.

The Jacobi and Gauss–Seidel method can be interpreted as PSC and SSC based on the decomposition

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

with exact subspace solves such that

$$B_{\rm psc} = D^{-1}, \quad B_{\rm ssc} = (D+L)^{-1}, \quad \bar{B}_{\rm ssc} = (D+U)^{-1} D (D+L)^{-1}.$$

By Theorem 4.5,

$$c_0 = \sup_{\|v\|_A=1} (D^{-1}Uv, Uv), \quad c_1 = \sup_{\|v\|_A=1} (D^{-1}(D+U)v, (D+U)v).$$

By Lemma 4.6,

$$\frac{1}{4}(Dv, v) \le (D^{-1}(D+U)v, (D+U)v) \le c_*(Dv, v), \quad v \in \mathbb{R}^n. \tag{4.29}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

We note that

$$c_1 = \sup_{v \in V} \frac{(\bar{B}_{\mathrm{ssc}}^{-1} v, v)}{\|v\|_A^2} \le \sigma \sup_{v \in V} \frac{\|v\|_D^2}{\|v\|_A^2},$$

where

$$\sigma = \sup_{v \in V} \frac{\|v\|^2_{\bar{B}_{\mathrm{ssc}}^{-1}}}{(Dv, v)}. \tag{4.30}$$

In the above presentation, most results are for SPD problems. We would like to point out that all of these results can be extended to a more general class of problems, namely symmetric positive semi-definite problems. When A is a matrix, we further assume that all the diagonals of A are non-zero and hence positive. When the subspace correction method is used for a more general symmetric positive semi-definite operator A, we further assume that each A_i is SPD. We shall use the acronym SPSD to denote matrices or operators that satisfy the aforementioned properties.

4.4. Bibliographical notes

The general notion of subspace corrections by means of space decompositions was described by Xu (1992), based on Bramble, Pasciak, Wang and Xu (1991*b*, 1991*c*). It is an abstract point of view encompassing the theory and practice of a large class of iterative algorithms such as multigrid and domain decomposition methods. In the past two decades a great deal of effort has been put into the investigation of theoretical and practical issues related to these methods. General results, applicable in many cases, in the theory of additive and multiplicative methods in Hilbert space can be found in Xu and Zikatanov (2002). For a literature review and basic results we refer the reader to the monographs and survey articles by Hackbusch (1985), Bramble (1993), Vassilevski (2008), Xu (1989, 1997), Xu and Zou (1998), Yserentant (1993), Toselli and Widlund (2005), Griebel and Oswald (1995) and Smith, Bjørstad and Gropp (1996). For detailed studies of classical iterative methods, we refer to the monographs by Young (1971), Hackbusch (1994), Varga (2000) and Saad (2003).

Note that in this section we have considered SPSD matrices, and according to Lee, Wu, Xu and Zikatanov (2007, 2008) and Ayuso de Dios, Brezzi, Marini, Xu and Zikatanov (2014), all the results in this section are valid for SPSD problems with semi-norms. Relations between the auxiliary space method and subspace correction methods is drawn in Chen (2011). In the classical multigrid literature (Brandt *et al.* 1982, Brandt, McCormick and Ruge 1985, Ruge and Stüben 1987, Trottenberg, Oosterlee and Schüller 2001), the notions of algebraically smooth (low) frequencies and algebraic high frequencies play an important role. They are also instrumental in the design of new AMG methods. As indicated by the convergence

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

estimates, for a given smoother, the desirable coarse spaces should capture or well approximate the lower end of the spectrum of the relaxed matrix $\bar{R}A$ or $D^{-1}A$. This is usually referred to as *near-null space* (Treister and Yavneh 2015, Lai and Olson 2011, Xu 2009, Brezina *et al.* 2006).

5. Abstract multigrid methods and two-level theory

In this section we will present algebraic multigrid methods in an abstract setting. The acronym AMG for 'algebraic multigrid' can also be used to stand for 'abstract multigrid'.

Our focus will be on two-level methods. In view of algorithmic design, the extension of two-level to multilevel is straightforward: a general multilevel V-cycle algorithm can be obtained by recursively applying a two-level algorithm. However, the extension of a two-level convergence theory to a multilevel case can be highly non-trivial.

We will only consider SPSD problems as described in Section 4. As is done in most of the literature, the design principle of an AMG is to optimize the choice of coarse spaces with a given smoother. The most commonly used smoothers are the Gauss–Seidel method and the (modified or scaled) Jacobi method. As these smoothers are convergent as iterative methods, the resulting AMG method is always convergent. The task of our AMG convergence theory is to make sure that this convergence is also fast. In particular, for systems arising from the discretization of partial differential equations, we hope that our AMG method converges uniformly with respect to the size of the problem and/or some crucial parameters from the underlying PDEs. We sometimes speak of this as 'uniformly convergent' or 'uniform convergence'.

As it turns out, we are often able to establish such uniform convergence for two-level AMG, but very rarely can we extend a uniform convergence result to the multilevel case. For second-order elliptic boundary value problems, multilevel convergence is very well understood for geometric multigrid methods. However, a rigorous multilevel convergence theory for AMG without using geometric information is still very much an open problem.

We will mainly focus on two-level convergence theory for AMG methods in this section and also in the rest of the paper.

5.1. *A two-level method*

A two-level method typically consists of the following components:

(i) a smoother, $R : V' \mapsto V$,

(ii) a coarse space V_c, which may or may not be a subspace of V, linked to V via a prolongation operator,

$$P : V_c \mapsto V,$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

(iii) a coarse space solver, $B_c : V_c' \mapsto V_c$.

In the discussion below we need the inner product

$$(u,v)_{\bar{R}^{-1}} = (\overline{T}^{-1}u, v)_A = (\bar{R}^{-1}u, v), \quad \overline{T} = \overline{R}A \tag{5.1}$$

and the accompanying norm $\|\cdot\|_{\bar{R}^{-1}}$. Here we recall that the definition of $\bar{R}$ is analogous to that in (4.5).

We always assume that $\bar{R}$ is SPD, and hence the smoother R is always convergent. Furthermore,

$$\|v\|_A^2 \le \|v\|_{\bar{R}^{-1}}^2. \tag{5.2}$$

The restriction of (4.1) is then

$$A_c u_c = f_c, \tag{5.3}$$

where

$$A_c = P'AP, \quad f_c = P'f.$$

The coarse space solver B_c is often chosen to be the exact solver, namely $B_c = A_c^{-1}$, for analysis, but in a multilevel setting, B_c is recursively defined and it is an approximation to A_c^{-1}. We distinguish two different cases in choosing B_c:

$$\text{it is an } \textit{exact} \text{ two-level method if } B_c = A_c^{-1}, \tag{5.4}$$

$$\text{it is an } \textit{inexact} \text{ two-level method if } B_c \ne A_c^{-1}. \tag{5.5}$$

When A is semi-definite, we use N to denote the kernel of A and we always assume that $N \subset V_c$. Let

$$W := N^{\perp} \quad \text{and} \quad W_c := V_c \cap W, \tag{5.6}$$

where the orthogonality is understood with respect to the $(\cdot,\cdot)_{\bar{R}^{-1}}$-inner product. Let $Q_1 : V \mapsto W$ be the orthogonal projection with respect to the $(\cdot,\cdot)_{\bar{R}^{-1}}$-inner product

$$(Q_1 v, w)_{\bar{R}^{-1}} = (v, w)_{\bar{R}^{-1}}, \quad \text{for all } v \in V, w \in W. \tag{5.7}$$

Here A_c is semi-definite on V_c but invertible on W_c. We denote the restriction of A_c on W_c by $\hat{A}_c$, and define the pseudo-inverse of A_c by

$$A_c^{\dagger} := Q_1' \hat{A}_c^{-1} Q_1. \tag{5.8}$$

With a slight abuse of notation, we will still use A_c^{-1} to denote the pseudo-inverse of A_c, namely

$$A_c^{-1} = A_c^{\dagger}.$$

We will use similar notation for the pseudo-inverse of other relevant singular operators and matrices in the rest of the paper.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Algorithm 4 Two-level multigrid method

```
Given $g \in V'$, the action $Bg$ is defined via the following two steps.
1: Coarse grid correction: $w = PB_cP'g$.
2: Post-smoothing: $Bg := w + R(g - Aw)$.
```

We choose to define an AMG algorithm in terms of an operator $B : V' \mapsto V$, which can be considered as an approximate inverse or a preconditioner of A. A typical two-level MG method is shown in Algorithm 4.

In the rest of this section, we take $B_c = A_c^{-1}$.

There are usually two different (and mathematically equivalent) ways to choose V_c, V, P and R. The first, known as the operator version, is such that

$$V_c \subset V.$$

In this case $P = \imath_c$, where

$$\imath_c : V_c \mapsto V \tag{5.9}$$

is the natural inclusion of V_c into V. In the application to finite element discretization for second-order elliptic boundary value problems, V_c and V are just the finite element subspaces of $H^1(\Omega)$. This type of notation is convenient for analysis. However, this is not an algorithm that can be directly used for implementation.

The second, known as the matrix version, is such that

$$V_c = \mathbb{R}^{n_c} \quad \text{and} \quad V = \mathbb{R}^n,$$

and

$$P : \mathbb{R}^{n_c} \mapsto \mathbb{R}^n \tag{5.10}$$

is the prolongation matrix.

These two different notations are related via the use of basis functions of $\{\phi_i^c\}_{i=1}^{n_c} \subset V_c$ and $\{\phi_i\}_{i=1}^{n} \subset V$. As noted earlier, the prolongation matrix P given in (3.12)–(3.13) is simply the matrix representation of $\imath_c$ given in (5.9), and we have

$$(\phi_i^c, \dots, \phi_{n_c}^c) = (\phi_i, \dots, \phi_n)P. \tag{5.11}$$

The following observation is clear.

Observation 5.1. Finding a coarse space $V_c \subset V$ is equivalent to finding a prolongation matrix P in (5.10).

Lemma 5.2. *The error propagation operator for two-level AMG operator $E = I - BA$ is*

$$E = (I - RA)(I - \Pi_c), \tag{5.12}$$

where $\Pi_c = \imath_c A_c^{-1} \imath_c' A$, which is the $(\cdot,\cdot)_A$ orthogonal projection on V_c, in matrix notation $\Pi_c = PA_c^{-1}P^T A$.

5.2. An optimal two-level AMG theory

The aim of the AMG method is to balance the interplay between smoother R and the coarse space V_c. Most existing AMG methods first fix a smoother, which is often given by the Jacobi or Gauss–Seidel methods (or their combinations and variations), and then optimize the choice of coarse spaces. This is the approach we will mainly discuss in this paper. However, we also comment on a different approach by first fixing the coarse space and then trying to optimize the choice of the smoother. It is also possible to try to make an optimal choice of smoother and coarse space simultaneously, but we will not address this approach here.

Let $Q_c : V \mapsto V_c$ be the orthogonal projection with respect to the $(\cdot,\cdot)_{\bar{R}^{-1}}$-inner product

$$(Q_c u, v_c)_{\bar{R}^{-1}} = (u, v_c)_{\bar{R}^{-1}}, \quad \text{for all } v_c \in V_c. \tag{5.13}$$

By the definition of W and W_c, we have that $(\cdot,\cdot)_A$ is an inner product on W, $\|\cdot\|_A$ is a norm on W, and the projection $\Pi_c : V \mapsto W_c$ is well defined:

$$(\Pi_c u, v_c)_A = (u, v_c)_A, \quad \text{for all } u \in V,\ v_c \in W_c. \tag{5.14}$$

The two-level convergence rate is obtained in the following theorem.

Theorem 5.3. Assume that $N \subset V_c$. The convergence rate of an exact two-level AMG is given by

$$\|E\|_A^2 = 1 - \frac{1}{K(V_c)}, \tag{5.15}$$

where

$$K(V_c) = \max_{v\in W} \frac{\|(I-Q_c)v\|_{\bar{R}^{-1}}^2}{\|v\|_A^2} = \max_{v\in W} \min_{v_c\in W_c} \frac{\|v - v_c\|_{\bar{R}^{-1}}^2}{\|v\|_A^2}. \tag{5.16}$$

Proof. We note that

$$\|(I-T)v\|_A^2 = ((I-\bar{T})v, v)_A, \quad \text{for all } v \in V.$$

Then we have

$$\begin{aligned}
\|E\|_A^2 &= \max_{w\in W} \frac{\|(I-T)(I-\Pi_c)w\|_A^2}{\|w\|_A^2} \\
&= \max_{w\in W} \frac{((I-\overline{T})(I-\Pi_c)w, (I-\Pi_c)w)_A}{\|w\|_A^2} \\
&= 1 - \min_{w\in W} \frac{(\overline{T}(I-\Pi_c)w, (I-\Pi_c)w)_A}{\|w\|_A^2}
\end{aligned}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

$$
\begin{aligned}
&= 1 - \min_{w\in W} \frac{(Q_1\overline{T}(I-\Pi_c)w,(I-\Pi_c)w)_A}{\|(I-\Pi_c)w\|_A^2 + \|\Pi_c w\|_A^2}\\
&= 1 - \min_{v\in W_c^{\perp_A}} \frac{(Q_1\overline{T}v,v)_A}{\|v\|_A^2}\\
&= 1 - \min_{v\in W_c^{\perp_A}} \frac{((I-\Pi_c)Q_1\overline{T}v,v)_A}{\|v\|_A^2}\\
&= 1 - \lambda_{\min}(X),
\end{aligned}
$$

where

$$X = (I-\Pi_c)Q_1\overline{T} : W_c^{\perp_A} \mapsto W_c^{\perp_A},$$

and it is easy to see that X is self-adjoint with respect to $(\cdot,\cdot)_A$.

One key observation is that the inverse of X can be explicitly written as

$$Z = (Q_1\overline{T})^{-1}(I-Q_c),$$

since by definition, we have, for any $u,v\in V$,

$$
\begin{aligned}
(\Pi_c Zu,v)_A &= ((Q_1\overline{T})^{-1}(I-Q_c)u,\Pi_c v)_A = (\overline{T}(Q_1\overline{T})^{-1}(I-Q_c)u,\Pi_c v)_{\bar{R}^{-1}}\\
&= (Q_1\overline{T}(Q_1\overline{T})^{-1}(I-Q_c)u,\Pi_c v)_{\bar{R}^{-1}} = ((I-Q_c)u,\Pi_c v)_{\bar{R}^{-1}} = 0,
\end{aligned}
$$

which implies $\Pi_c Z = 0$. Thus we have

$$Z : W_c^{\perp_A} \mapsto W_c^{\perp_A},$$

and furthermore

$$XZ = (I-\Pi_c)(I-Q_c) = I - \Pi_c = I \text{ on } W_c^{\perp_A}.$$

Consequently

$$\lambda_{\min}(X) = \frac{1}{\lambda_{\max}(Z)}.$$

Finally,

$$
\begin{aligned}
\lambda_{\max}(Z) &= \max_{v\in W_c^{\perp_A}} \frac{((Q_1\overline{T})^{-1}(I-Q_c)v,v)_A}{(v,v)_A}\\
&= \max_{v\in W_c^{\perp_A}} \frac{(\overline{T}(Q_1\overline{T})^{-1}(I-Q_c)v,v)_{\bar{R}^{-1}}}{(v,v)_A}\\
&= \max_{v\in W_c^{\perp_A}} \frac{(Q_1\overline{T}(Q_1\overline{T})^{-1}(I-Q_c)v,v)_{\bar{R}^{-1}}}{(v,v)_A}\\
&= \max_{v\in W_c^{\perp_A}} \frac{((I-Q_c)v,v)_{\bar{R}^{-1}}}{(v,v)_A}\\
&= \max_{v\in W_c^{\perp_A}} \frac{\|(I-Q_c)v\|_{\bar{R}^{-1}}^2}{(v,v)_A} = K(V_c).
\end{aligned}
$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

The last identity holds because $I - Q_c = (I - Q_c)(I - \Pi_c)$, and we can then take the maximum over all $v \in W$. This completes the proof. $\square$

Theorem 5.3 can be stated as follows using the matrix representation introduced in Section 3.

Theorem 5.4. Assume that $P \in \mathbb{R}^{n \times n_c}$ and $N(\tilde{A}) \subset \text{Range}(P)$. The convergence rate of an exact two-level AMG is given by

$$\|E\|_A^2 = 1 - \frac{1}{\tilde{K}(P)}, \tag{5.17}$$

where

$$\tilde{K}(P) = \max_{v \in \mathbb{R}^n} \min_{v_c \in \mathbb{R}^{n_c}} \frac{\|v - Pv_c\|^2_{\tilde{\bar{R}}^{-1}}}{\|v\|^2_{\tilde{A}}}. \tag{5.18}$$

Remark 5.5. The result of Theorem 5.4 can be viewed as follows. We note that $\overline{T}^{-1}(I - Q_c)$ is a self-adjoint operator with respect to the A-inner product. Hence, we immediately have

$$K(V_c) = \|\overline{T}^{-1}(I - Q_c)\|_A = \|(\overline{R}A)^{-1}(I - Q_c)\|_A.$$

In the case of two subspaces, we have the following theorem giving the precise convergence rate of the corresponding SSC method.

Theorem 5.6. Let $\{\mu_j, \zeta_j\}_{j=1}^n$ be the eigenpairs of $\bar{T} = \bar{R}A$, and assume that $\{\zeta_j\}$ are orthogonal with respect to $(\cdot,\cdot)_{\bar{R}^{-1}}$. The convergence rate $\|E(V_c)\|_A$ is minimal for the coarse space

$$V_c^{\text{opt}} = \text{span}\{\zeta_j\}_{j=1}^{n_c} \in \operatorname*{arg\,min}_{\dim V_c = n_c, N \subset V_c} K(V_c). \tag{5.19}$$

In this case,

$$\|E\|_A^2 = 1 - \mu_{n_c+1}. \tag{5.20}$$

Proof. By Theorem 5.3, we just need to maximize $1/(K(V_c))$. For any $v \in V_c^\perp$, where $\perp$ is with respect to the $\bar{R}^{-1}$-inner product, we have

$$\min_{v_c \in V_c} \|v - v_c\|^2_{\bar{R}^{-1}} = \|v\|^2_{\bar{R}^{-1}}. \tag{5.21}$$

Then it follows that

$$\frac{1}{K(V_c)} = \min_{v \in V} \max_{v_c \in V_c} \frac{\|v\|_A^2}{\|v - v_c\|^2_{\bar{R}^{-1}}} \leq \min_{v \in V_c^\perp} \max_{v_c \in V_c} \frac{\|v\|_A^2}{\|v - v_c\|^2_{\bar{R}^{-1}}} = \min_{v \in V_c^\perp} \frac{\|v\|_A^2}{\|v\|^2_{\bar{R}^{-1}}}.$$

By the min-max principle (Theorem 2.1), we have

$$\max_{\dim V_c = n_c} \frac{1}{K(V_c)} \leq \max_{\dim V_c = n_c} \min_{v \in V_c^\perp} \frac{\|v\|_A^2}{\|v\|^2_{\bar{R}^{-1}}} = \mu_{n_c+1}.$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

On the other hand, if we choose $V_c^{\mathrm{opt}} = \operatorname{span}\{\zeta_j\}_{j=1}^{n_c}$, it is easy to compute that $K(V_c^{\mathrm{opt}}) = 1/(\mu_{n_c+1})$. So we have

$$\max_{\dim V_c = n_c} \frac{1}{K(V_c)} = \mu_{n_c+1},$$

with optimal coarse space

$$V_c^{\mathrm{opt}} = \operatorname{span}\{\zeta_j\}_{j=1}^{n_c}. \qquad \square$$

Using the matrix representation introduced in Section 3, we state the matrix version of Theorem 5.6 below. For simplicity, with abuse of notation, we still use A to denote the matrix representation of operator A.

Theorem 5.7. Let $\{\mu_j, \zeta_j\}_{j=1}^{n}$ be the eigenpairs of $\bar{T} = \bar{R}A$. Let us also assume that $\{\zeta_j\}$ are orthogonal with respect to $(\cdot,\cdot)_{\bar{R}^{-1}}$. The convergence rate $\|E(P)\|_A$ is minimal for P such that

$$\operatorname{Range}(P) = \operatorname{Range}(P^{\mathrm{opt}}), \tag{5.22}$$

where

$$P^{\mathrm{opt}} = (\zeta_1, \dots, \zeta_{n_c}). \tag{5.23}$$

In this case,

$$\|E\|_A^2 = 1 - \mu_{n_c+1}. \tag{5.24}$$

The following theorem is important in motivating most AMG algorithms.

Theorem 5.8. Given $\eta > 0$, let $\mathcal{X}_\eta$ be defined as

$$\mathcal{X}_\eta = \bigl\{P \in \mathbb{R}^{n\times n_c} : (Pv, Pv)_{\bar{R}^{-1}} \geq \eta(v,v), \quad v \in \mathbb{R}^{n_c}\bigr\}. \tag{5.25}$$

Then, with P^{opt} given by (5.23), we have $P \in \arg\min_{Q\in\mathcal{X}_\eta} \operatorname{trace}(Q^T A Q)$ if

$$P \in \mathcal{X}_\eta \quad \text{and} \quad \operatorname{Range}(P) = \operatorname{Range}(P^{\mathrm{opt}})).$$

Since the eigenvalues of $\bar{R}A$ are expensive to compute, the practical value of Theorem 5.6 is limited. However, it provides useful guidance in the design of practical AMG methods.

For finite element discretizations we can use Weyl's law combined with Theorem 5.6 to prove an estimate for the convergence rate of a two-grid method with an optimal coarse space.

Corollary 5.9. Let the assumptions of the discrete Weyl's law (Theorem 2.5) hold, and let the smoother $\bar{R}$ be spectrally equivalent to the diagonal of the stiffness matrix A. Let $\gamma > 0$ be such that $\gamma n \leq n_c < n$. Then, for the optimal coarse space, we have the estimate

$$\mu_{n_c+1} \geq \delta_0 \quad \text{and} \quad \|E\|_A^2 \leq 1 - \delta_0,$$

where $\delta_0 \in (0,1)$ only depends on γ and the constants γ_0 and γ_1 in (2.31).

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Proof. By the assumptions in Theorem 2.5 and the fact that $\bar{R}$ is spectrally equivalent to the diagonal of A, we have

$$|w|_1^2 \eqsim \|w\|_A^2, \quad \|w\|_0^2 \eqsim h^2 \|w\|_{\bar{R}^{-1}}^2. \tag{5.26}$$

Further, Lemma 2.4 and Theorem 2.5 then show that

$$\mu_{n_c+1}(\bar{R}A) = \min_{\substack{W \subset V_h \\ \dim W = k}} \max_{\substack{w \in W \\ \|w\|_{\bar{R}^{-1}} \neq 0}} \frac{\|w\|_A^2}{\|w\|_{\bar{R}^{-1}}^2} \eqsim h^2 \lambda_{n_c+1} = O\left(\left(\frac{n_c h^d}{\mathrm{Vol}(\Omega)}\right)^{2/d}\right). \tag{5.27}$$

The desired result follows immediately from Theorem 5.6 because $\mathrm{Vol}(\Omega) \eqsim h^d n$ and $\gamma n \leq n_c < n$, which gives $\mu_{n_c+1} \eqsim 1$. □

Remark 5.10. Since the coarse space which minimizes the convergence rate is also the coarse space which minimizes $K(V_c)$, as a corollary we have the inequality

$$K(V_c) = \frac{1}{1 - \|E\|_A^2} \geq \frac{1}{\mu_{n_c+1}},$$

or

$$\|E\|_A^2 \geq 1 - \mu_{n_c+1}.$$

Theorem 5.3 provides an explicit estimate for the convergence of a two-level method in terms of $K(V_c)$. For a given method, a smaller bound on $K(V_c)$ means a faster convergence rate. In particular, the two-level AMG method is uniformly convergent if $K(V_c)$ is uniformly bounded with respect to mesh parameters.

5.3. Quasi-optimal theories

We now look at the necessary and sufficient condition for uniform convergence of a two-level method as proved in Section 5.2 (see Theorem 5.3):

$$\min_{v_c \in V_c} \|v - v_c\|_{\bar{R}^{-1}} \leq K(V_c) \|v\|_A^2. \tag{5.28}$$

Here $K(V_c)$ is the smallest constant for which (5.28) holds for all $v \in V$. The space V_c which minimizes $K(V_c)$ is V_c^{opt}. A similar argument was used in the proof of Theorem 5.6 in Section 5.2. Here we generalize the result to semi-definite A.

For a given smoother R, one basic strategy in the design of AMG is to find a coarse space such that $K(V_c)$ is as small as possible in practice. There are many cases, however, in which the operator $\bar{R}^{-1}$ in the definition of $K(V_c)$ is difficult to work with.

One commonly used approach is to replace $\bar{R}^{-1}$ with a simpler but spectrally equivalent SPD operator. More specifically, we assume that $D : V \mapsto$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

V' is an SPD operator such that

$$c_D(Dv,v) \le (\bar{R}^{-1}v,v) \le c^D(Dv,v), \quad \text{for all } v \in V. \tag{5.29}$$

That is,

$$c_D\|v\|_D^2 \le \|v\|_{\bar{R}^{-1}}^2 \le c^D\|v\|_D^2, \quad \text{for all } v \in V, \tag{5.30}$$

where

$$(u,v)_D = (Du,v), \quad \|v\|_D^2 = (v,v)_D.$$

Examples of such equivalent norms for Schwarz smoothers are given in (4.20) and (4.27). As a rule, the norm defined by $\bar{R}$ corresponding to the symmetric Gauss–Seidel method, that is, R defined by the pointwise Gauss–Seidel method, can be replaced with the norm defined by the diagonal of A (*i.e.*, by the Jacobi method, which, while not always convergent as a relaxation, provides an equivalent norm).

In terms of this operator D, we introduce the quantity

$$K(V_c,D) = \max_v \frac{\|v - Q_D v\|_D^2}{\|v\|_A^2} = \max_v \min_{v_c \in V_c} \frac{\|v - v_c\|_D^2}{\|v\|_A^2}, \tag{5.31}$$

where $Q_D : V \mapsto V_c$ is the $(u,v)_D$-orthogonal projection.

By (5.16), (5.31) and (5.30), we have

$$c_D K(V_c,D) \le K(V_c) \le c^D K(V_c,D). \tag{5.32}$$

Theorem 5.11. The two-level algorithm satisfies

$$1 - \frac{1}{c_D K(V_c,D)} \le \|E\|_A^2 \le 1 - \frac{1}{c^D K(V_c,D)} \le 1 - \frac{1}{c^D C}, \tag{5.33}$$

where C is any upper bound of $K(V_c,D)$, namely

$$\min_{w \in V_c} \|v - w\|_D^2 \le C\|v\|_A^2, \quad \text{for all } v \in V. \tag{5.34}$$

The proof of Theorem 5.11 is straightforward and indicates that, if c_D and c^D are 'uniform' constants, the convergence rate of the two-level method is 'uniformly' dictated by the quantity $K(V_c,D)$.

We say that V_c is quasi-optimal if the following inequality holds:

$$\min_{w \in V_c} \|v - w\|_D^2 \le \gamma\mu_{n_c+1}^{-1}\|v\|_A^2, \quad \text{for all } v \in V, \tag{5.35}$$

with a constant $\gamma > 0$ independent of the size of the problem.

The construction of an approximation to the optimal coarse space V_c^{opt} that is used in most AMG algorithms relies on two operators A_M and D_M which satisfy

$$c_1\|v\|_D^2 \le \|v\|_{D_M}^2, \quad \|v\|_{A_M}^2 \le c_2\|v\|_A^2, \quad \text{for all } v \in V, \tag{5.36}$$

with constants c_1 and c_2 independent of the problem size. Here, on the

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

right-hand side, we have a seminorm $\|\cdot\|_{A_M}$, because sometimes A_M is only semi-definite. We point out that here A_M and D_M are analogues of $\underset{\sim}{A}_W$ and $\underset{\sim}{D}$ defined in (6.6) and (6.7), respectively, in the general framework of Section 6; the assumptions in (5.36) are analogous to Assumption 6.4 which we made in the general AMG framework in Section 6.

Theorem 5.12. If D_M and A_M satisfy (5.36), and V_c is a coarse space such that

$$\min_{w\in V_c} \|v-w\|_{D_M}^2 \le \gamma\mu_{n_c+1}^{-1}\|v\|_{A_M}^2, \quad \text{for all } v\in V, \tag{5.37}$$

then the following hold.

(a) We have

$$\min_{w\in V_c} \|v-w\|_{\overline{R}^{-1}}^2 \le \frac{c^D}{c_D}\frac{c_2}{c_1}\gamma\mu_{n_c+1}^{-1}\|v\|_A^2, \quad \text{for all } v\in V. \tag{5.38}$$

(b) The corresponding two-level AMG algorithm satisfies

$$\|I-BA\|_A^2 \le 1-\frac{c_D}{c^D}\frac{c_1}{c_2}\frac{1}{\gamma}\,\mu_{n_c+1}. \tag{5.39}$$

5.4. Algebraically high and low frequencies

In geometric MG, algebraically smooth error is also smooth in the usual geometric sense. However, in AMG settings, smooth error can be geometrically non-smooth. In order to make this distinction, we use the term *algebraically smooth error* when we refer to the error in the AMG setting that is not damped (eliminated) by the smoother R. In general, good interpretation of the algebraically smooth error leads to an efficient and robust AMG algorithm. Careful characterization of the algebraically smooth error is needed, since in this case we can try to construct a coarser level which captures these error components well.

Here is a more formal definition of an algebraically smooth error.

Definition 5.13. Let $R : V \mapsto V$ be a smoothing operator such that its symmetrization $\overline{R} = R + R^T - R^TAR$ is positive definite. Given $\varepsilon \in (0,1)$, we say that the vector v is algebraically ε-smooth (or v is an *ε-algebraic low frequency*) with respect to A if

$$\|v\|_A^2 \le \varepsilon\|v\|_{\overline{R}^{-1}}^2. \tag{5.40}$$

The set of algebraically smooth vectors will be denoted by

$$\mathcal{L}_\varepsilon = \{v : \|v\|_A^2 \le \varepsilon\|v\|_{\overline{R}^{-1}}^2\}. \tag{5.41}$$

We point out that this is a set of vectors (a ball, or rather an ellipsoid) and not a linear vector space in general. It is then clear that the elements of this set need to be approximated well by elements from the coarse space.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

The rationale of Definition 5.13 can be seen from the following simple result (it is always true that $(\bar{R}Av, v) \le \|v\|_A^2$).

Lemma 5.14. Any vector $v \in V$ that satisfies

$$(\overline{R}A\boldsymbol{v}, \boldsymbol{v})_A \le \varepsilon \|v\|_A^2 \tag{5.42}$$

is ε-algebraically smooth.

Proof. By the Schwarz inequality for the inner product defined by $\overline{R}^{-1}$ and (5.42), we have

$$\|\boldsymbol{v}\|_A^2 = (\overline{R}A\boldsymbol{v}, \overline{R}^{-1}\boldsymbol{v}) \le (\overline{R}A\boldsymbol{v}, A\boldsymbol{v})^{1/2}(\overline{R}^{-1}\boldsymbol{v}, \boldsymbol{v})^{1/2} \le \sqrt{\varepsilon}(\overline{R}^{-1}\boldsymbol{v}, \boldsymbol{v})^{1/2}\|\boldsymbol{v}\|_A.$$

□

We can easily show that this definition is equivalent to saying that the algebraically smooth error components are the components for which the smoother converges slowly. Indeed, inequality (5.42) is clearly equivalent to $((I - \overline{R}A)v, v)_A \ge (1-\varepsilon)(v, v)_A$, namely

$$\frac{\|S\boldsymbol{v}\|_A^2}{\|\boldsymbol{v}\|_A^2} \ge 1 - \varepsilon, \quad S = I - T \quad \text{and} \quad T = RA. \tag{5.43}$$

The property (5.43) is often referred to as the *smoothing property*.

Remark 5.15. In the classical multigrid literature, an algebraically smooth error is defined as $\boldsymbol{e} \in V$ such that

$$\|\boldsymbol{e}\|_{AD^{-1}A}^2 \le \varepsilon \|\boldsymbol{e}\|_A^2, \tag{5.44}$$

for a small and positive parameter ε which implies

$$\|\boldsymbol{e}\|_A^2 \le \|\boldsymbol{e}\|_D \|\boldsymbol{e}\|_{AD^{-1}A} \le \sqrt{\varepsilon}\|\boldsymbol{e}\|_D\|\boldsymbol{e}\|_A. \tag{5.45}$$

That is,

$$\|\boldsymbol{e}\|_A^2 \le \varepsilon \|\boldsymbol{e}\|_D^2. \tag{5.46}$$

As is clearly seen from Definition 5.13, Lemma 5.14 implies (5.45) with $\overline{R} \approx D^{-1}$, where D is the diagonal of A.

Thanks to (7.9), we have the following result.

Lemma 5.16. If e is algebraically smooth, that is, e satisfies (5.46), then

$$\|\boldsymbol{e}\|_{\tilde{A}}^2 \lesssim \varepsilon \|\boldsymbol{e}\|_{\tilde{D}}^2. \tag{5.47}$$

Namely, e is also algebraically smooth with respect to $\tilde{A}$, the M-matrix relative of A.

On the other hand, note that by the definition of $\|\cdot\|_{\bar{R}^{-1}}$, we always have $\|v\|_{\bar{R}^{-1}} \ge C\|v\|_A$, with constant C independent of the parameters of

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

interest. Drawing from analogy with the geometric multigrid method, we introduce the notion of algebraic high frequency as follows.

Definition 5.17. Given $\delta \in (0,1]$, we call $v \in V$ a *δ-algebraic high frequency* if

$$\|v\|_A^2 \geq \delta \|v\|_{\bar{R}^{-1}}^2.$$

The set of algebraically high-frequency vectors will be denoted by

$$\mathcal{H}_\delta = \big\{ v : \|v\|_A^2 \geq \delta \|v\|_{\bar{R}^{-1}}^2 \big\}. \tag{5.48}$$

The concept of algebraic high and low frequencies will be used in the two-level AMG theory in Section 5.3, and also in the design of classical AMG in Section 12.1.

Lemma 5.18. Let (ϕ_i, μ_i) be all the eigenpairs for $\bar{R}A$, namely $\bar{R}A\phi_i = \mu_i \phi_i$. Then

$$\operatorname{span}\{\phi_i : \mu_i \leq \varepsilon\} \subset \mathcal{L}_\varepsilon$$

and

$$\operatorname{span}\{\phi_i : \mu_i \geq \delta\} \subset \mathcal{H}_\delta.$$

We now introduce the notion of *near-null space.*

Definition 5.19 (near-null space). For sufficiently small $\varepsilon \in (0,1)$, we shall say that $\operatorname{span}\{\phi_i : \mu_i \leq \varepsilon\}$ is an ε-near-null space of $\bar{R}A$.

5.5. Smoothing properties of Jacobi and Gauss–Seidel methods

The essence of multigrid methods is that simple iterative methods such as Jacobi and Gauss–Seidel methods have a special property known as the *smoothing property.* As an illustration, we apply the Gauss–Seidel method to

$$A\mu = b,$$

with A given by (2.16) for isotropic problems and (2.34) for anisotropic problems, respectively. We first choose μ randomly (as shown in Figure 5.1) and then compute $A\mu$ for both (2.16) and (2.34) to compute right-hand sides $b = A\mu$ respectively. We then apply the Gauss–Seidel method to both equations with initial guess $\mu^0 = 0$ (see Figure 5.2).

Note that, for A given by (2.34) when $\epsilon \ll 1$, we have

$$\lambda_{11} < \lambda_{21} < \cdots < \lambda_{N1} < \lambda_{ij}, \quad i \geq 1, j \geq 2.$$

The corresponding eigenfunctions, which can be viewed as 'algebraic low frequencies', can be highly oscillatory in the x-direction.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

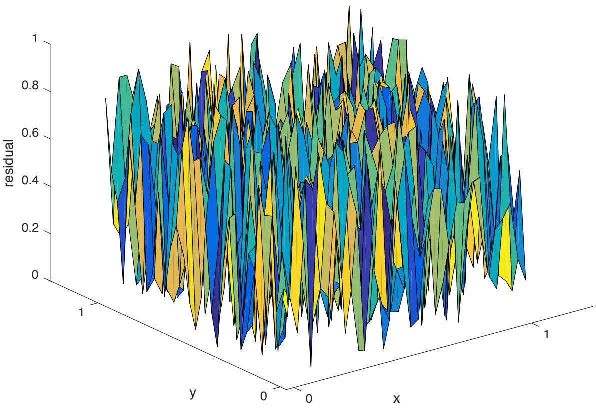


Figure 5.1. Initial error for both (2.16) and (2.34).

As an illustration of the difference between algebraic high/low frequencies and geometric high/low frequencies, we consider the linear system given by (2.34) for anisotropic problems. Clearly, A can be written as

$$A = \epsilon I \otimes M + M \otimes I \quad \text{with } M = \text{tridiag}(-1, 2, -1).$$

We define the vector $\mu \in R^N$ as

$$\mu = x \otimes y, \quad \text{with } x = \mathbf{1}_n, \quad \text{and} \quad y = (1\ 0\ 1\ 0 \cdots 1\ 0\ 1)^T \in R^n.$$

Then it is easy to compute that

$$Mx = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad \text{and} \quad My = \begin{pmatrix} 2 \\ -2 \\ 2 \\ \vdots \\ -2 \\ 2 \end{pmatrix}.$$

We have

$$A\mu = \epsilon(I \otimes M)(x \otimes y) + (M \otimes I)(x \otimes y) = \epsilon(x \otimes My) + (Mx \otimes y)$$

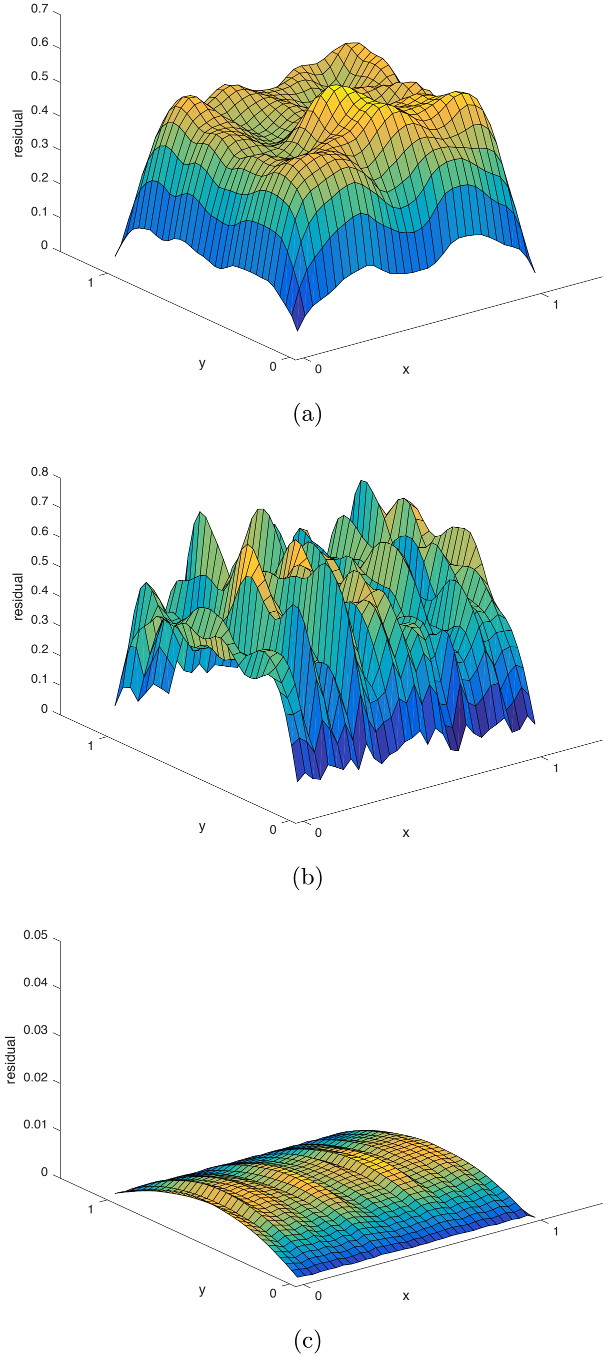
and

$$\begin{aligned} \|\mu\|_A^2 &= \mu^T A\mu = \epsilon(x \otimes y)^T (x \otimes My) + (x \otimes y)^T (Mx \otimes y) \\ &= \epsilon(x^T x) \otimes (y^T My) + (x^T Mx) \otimes (y^T y) = \epsilon n(n+1) + n + 1. \end{aligned}$$

Letting D be the diagonal of A, we then have

$$\|\mu\|_D^2 = 2(1+\epsilon)\mu^T \mu = (1+\epsilon)n(n+1).$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press



(a)

(b)

(c)

Figure 5.2. (a) Error after applying five Gauss–Seidel iterations to (2.16). (b) Error after applying five Gauss–Seidel iterations to (2.34). (c) Error after applying one block Gauss–Seidel iteration to (2.34).

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

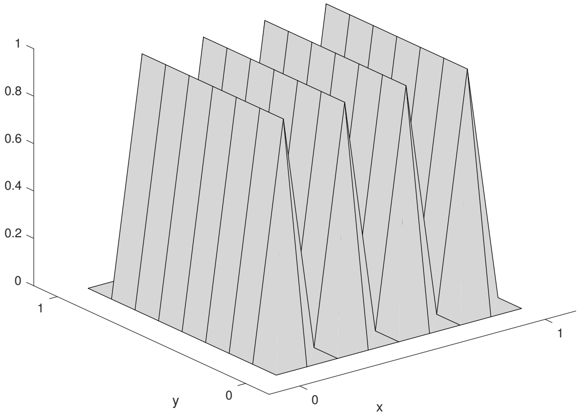


Figure 5.3. Graph of function defined in (5.49). This function is highly oscillatory in the x-direction.

This shows that

$$\frac{\|\mu\|_A^2}{\|\mu\|_D^2} = \frac{\epsilon}{1+\epsilon} + \frac{1}{(1+\epsilon)(n+1)},$$

which implies that μ is an algebraic low frequency if ϵ is sufficiently small.

On the other hand, if we denote the nodal basis functions corresponding to the uniform finite element mesh by $\{\phi_{ij} : 1 \le i, j \le n\}$, that is, ϕ_{ij} is a piecewise linear function such that

$$\phi_{ij}(kh, lh) = \delta_{ik}\delta_{jl},$$

then we define

$$\mathbf{\Phi} = (\phi_{11}, \phi_{12}, \ldots, \phi_{1n}, \phi_{21}, \ldots, \phi_{nn}).$$

If we consider the finite element function corresponding to μ, namely, the function defined by

$$u = \mathbf{\Phi}\mu = \sum_{i=1}^{(n+1)/2} \sum_{j=1}^{n} \phi_{2i-1,j}, \tag{5.49}$$

then, in the geometric point of view, this function is highly oscillatory in the x-direction, which is a geometric high frequency (see Figure 5.3).

5.6. Convergence theory in view of algebraic high and low frequencies

We next present a convergence theory based on algebraic high and low frequencies. We first prove the following lemma.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Lemma 5.20. Let $V_c \subset V$ be such that we obtain the 'stable decomposition'

$$V = V_c + V_{hf}$$

for some $V_{hf} \subset V$ which consists of δ-algebraic high frequencies (see Definition 5.17). Namely, for any $v \in V$, there exists $v_c \in V_c$ and $v_{hf} \in V_{hf}$ such that

$$v = v_c + v_{hf}, \quad \|v_{hf}\|_A^2 \le c_1 \|v\|_A^2.$$

Then the corresponding two-level AMG satisfies

$$\|E\|_A \le 1 - \frac{\delta}{c_1}.$$

Proof. It follows that

$$\inf_{w_c \in V_c} \|v - w_c\|_{\bar{R}^{-1}}^2 \le \|v_{hf}\|_{\bar{R}^{-1}}^2 \le \delta^{-1} \|v_{hf}\|_A^2 \le \frac{c_1}{\delta} \|v\|_A^2.$$

As a result,

$$K(V_c) \le \frac{c_1}{\delta},$$

and finally we have

$$\|E\|_A = 1 - \frac{1}{K(V_c)} \le 1 - \frac{\delta}{c_1}. \qquad \square$$

Corollary 5.21. If V_{hf} consists of δ-algebraic high frequencies, then for the coarse space V_c given by

$$V_c = \mathrm{Range}(I - P_{hf}), \tag{5.50}$$

where $P_{hf} : V \mapsto V_{hf}$ is the A-orthogonal projection, we obtain

$$\|E\|_A \le 1 - \delta.$$

5.7. Bibliographical notes

One of the first results on two-level convergence of AMG methods can be found in papers by Brandt *et al.* (1982) and Ruge and Stüben (1987). There has been much research on extending the MG theory to algebraic settings (Maitre and Musy 1983, Bank and Douglas 1985, Mandel 1988) and the algebraic variational approach to the two-level MG theory (McCormick 1985, McCormick 1984, McCormick and Ruge 1982).

For two-grid convergence, sharper results, including two-sided bounds, are given by Zikatanov (2008) and also considered by Falgout and Vassilevski (2004) and Falgout, Vassilevski and Zikatanov (2005). These two-level results are more or less a direct consequence of the abstract theory provided by Bramble, Pasciak, Wang and Xu (1991*c*), Xu (1992) and Xu and Zikatanov

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

(2002). A survey of these and other related results can be found in a recent article by MacLachlan and Olson (2014). The two approaches included in this section were recently developed by Xu, Zhang and Zikatanov (2016*a*, 2016*c*).

Theorem 5.3 can be found in Zikatanov (2008), and can be viewed as a consequence of the XZ identity (Xu and Zikatanov 2002) in the special case of two subspaces from the general framework of the subspace correction method. The original proof of this theorem in Zikatanov (2008) was based on the XZ identity. The proof here is new and more direct.

Multilevel results are difficult to establish in general algebraic settings, and most of them are either based on unrealistic assumptions or they use geometric grids to prove convergence. We refer to Vaněk, Mandel and Brezina (1996) and Brezina and Vassilevski (2011) for results in this direction. Rigorous multilevel results for finite element equations can be derived using the auxiliary space framework, which was developed by Xu (1996) for quasi-uniform meshes. More recently, Chen, Nochetto and Xu (2012) showed that multilevel convergence results for adaptively refined grids are optimal. A multilevel convergence result on shape-regular grids using AMG based on quad-tree coarsening (in two dimensions) and oct-tree coarsening (in three dimensions) is shown in Grasedyck, Wang and Xu (2015).

Finally, we point out that the notation used in parts of this section originates in Bank and Dupont (1980), Bramble and Pasciak (1987) and Bramble, Pasciak and Xu (1990), and is convenient for the analysis, especially when finite element equations are considered.

6. A general approach to constructing coarse spaces

In this section we describe an abstract framework for constructing coarse spaces by using the notion of space decomposition and subspace corrections.

Let us first introduce some technical results that will be used later as analytic tools.

Lemma 6.1. Let $\underset{\sim}{V}$ and V be two vector spaces and let $\Pi : \underset{\sim}{V} \mapsto V$ be a surjective map. Let $\underset{\sim}{B} : \underset{\sim}{V}' \mapsto \underset{\sim}{V}$ be an SPD operator. Then $B := \Pi \underset{\sim}{B} \Pi'$ is also SPD. Furthermore,

$$(B^{-1}v, v) = \min_{\Pi \underset{\sim}{v} = v} \langle \underset{\sim}{B}^{-1} \underset{\sim}{v}, \underset{\sim}{v} \rangle, \tag{6.1}$$

with the unique minimizer given by

$$\underset{\sim}{v}^* = \underset{\sim}{B} \Pi' B^{-1} v. \tag{6.2}$$

Lemma 6.2. Assume the following two conditions are satisfied for Π.

(a) For all $\underset{\sim}{v} \in \underset{\sim}{V}$,

$$\|\Pi \underset{\sim}{v}\|_A \le \tilde{\mu}_1 \|\underset{\sim}{v}\|_{\underset{\sim}{B}^{-1}}. \tag{6.3}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

(b) For any $v \in V$, there exists $\underset{\sim}{v} \in \underset{\sim}{V}$ such that $\Pi \underset{\sim}{v} = v$ and

$$\|\underset{\sim}{v}\|_{\underset{\sim}{B}^{-1}} \le \tilde{\mu}_0 \|v\|_A. \tag{6.4}$$

Then

$$\kappa(BA) \le \left(\frac{\tilde{\mu}_1}{\tilde{\mu}_0}\right)^2.$$

A direct consequence of Lemma 6.2 is the following result.

Theorem 6.3 (fictitious space lemma). Assume the following two conditions are satisfied for Π. First,

$$\|\Pi \underset{\sim}{v}\|_A \le \mu_1 \|\underset{\sim}{v}\|_{\underset{\sim}{A}}, \quad \text{for all } \underset{\sim}{v} \in \underset{\sim}{V}$$

Second, for any $v \in V$, there exists $\underset{\sim}{v} \in \underset{\sim}{V}$ such that $\Pi \underset{\sim}{v} = v$ and

$$\|\underset{\sim}{v}\|_{\underset{\sim}{A}} \le \mu_0 \|v\|_A.$$

Then $\kappa(\Pi) \le \mu_1/\mu_0$ and, under the assumptions of Lemma 6.1,

$$\kappa(BA) \le \left(\frac{\mu_1}{\mu_0}\right)^2 \kappa(\underset{\sim}{B}\underset{\sim}{A}).$$

We assume there exists a sequence of spaces $V_1, V_2, \dots, V_J$, which are not necessarily subspaces of V, but each of them is related to the original space V by a linear operator

$$\Pi_j : V_j \mapsto V. \tag{6.5}$$

We assume that V can be written as a sum of subspaces, and (4.12) and (4.13) hold.

Let

$$\underset{\sim}{W} = V_1 \times V_2 \times \cdots \times V_J,$$

with the inner product

$$(\underset{\sim}{u}, \underset{\sim}{v}) = \sum_{i=1}^{J} (u_i, v_i),$$

where $\underset{\sim}{u} = (u_1, \dots, u_J)^T$ and $\underset{\sim}{v} = (v_1, \dots, v_J)^T$. More generally, for $\underset{\sim}{f} = (f_1, \dots, f_J)^T \in \underset{\sim}{V}'$ with $f_i \in V_i'$, we can define

$$(\underset{\sim}{f}, \underset{\sim}{v}) = \sum_{i=1}^{J} (f_i, v_i).$$

We now define $\Pi_W : \underset{\sim}{W} \mapsto V$ by

$$\Pi_W \underset{\sim}{u} = \sum_{i=1}^{J} \Pi_i u_i, \quad \text{for all } \underset{\sim}{u} = (u_1, \dots, u_J)^T \in \underset{\sim}{W}.$$

Formally, we can write

$$\Pi_W = (\Pi_1, \ldots, \Pi_J) \quad \text{and} \quad \Pi'_W = \begin{pmatrix} \Pi'_1 \\ \vdots \\ \Pi'_J \end{pmatrix}.$$

We assume there is an operator $A_j : V_j \mapsto V'_j$ which is symmetric positive semi-definite for each j, and define $\underset{\sim}{A}_W : \underset{\sim}{W} \mapsto \underset{\sim}{W}'$ as follows:

$$\underset{\sim}{A}_W := \operatorname{diag}(A_1, A_2, \ldots, A_J). \tag{6.6}$$

For each j, we assume there is a symmetric positive definite operator $D_j : V_j \mapsto V'_j$, and define $\underset{\sim}{D} : \underset{\sim}{W} \mapsto \underset{\sim}{W}'$ as follows:

$$\underset{\sim}{D} := \operatorname{diag}(D_1, D_2, \ldots, D_J). \tag{6.7}$$

We associate a coarse space V_j^c, $V_j^c \subset V_j$, with each of the spaces V_j, and consider the corresponding orthogonal projection $Q_j : V_j \mapsto V_j^c$ with respect to $(\cdot,\cdot)_{D_j}$. We define $\underset{\sim}{Q} : \underset{\sim}{W} \mapsto \underset{\sim}{W}'$ by

$$\underset{\sim}{Q} := \operatorname{diag}(Q_1, Q_2, \ldots, Q_J). \tag{6.8}$$

Assumption 6.4.

(a) The following inequality holds for all $\underset{\sim}{w} \in \underset{\sim}{W}$:

$$\|\Pi_W \underset{\sim}{w}\|_D^2 \le C_{p,2} \|\underset{\sim}{w}\|_{\underset{\sim}{D}}^2, \tag{6.9}$$

for some positive constant $C_{p,2}$.

(b) For each $w \in V$, there exists a $\underset{\sim}{w} \in \underset{\sim}{W}$ such that $w = \Pi_W \underset{\sim}{w}$ and

$$\|\underset{\sim}{w}\|_{\underset{\sim}{A}_W}^2 \le C_{p,1} \|w\|_A^2, \tag{6.10}$$

with a positive constant $C_{p,1}$ independent of w.

(c) For all j,

$$N(A_j) \subset V_j^c. \tag{6.11}$$

Remark 6.5. Assumption 6.4 implies that

$$w \in N(A) \Rightarrow \underset{\sim}{w} \in N(A_1) \times \cdots \times N(A_J).$$

We define the global coarse space V_c by

$$V_c := \sum_{j=1}^{J} \Pi_j V_j^c. \tag{6.12}$$

Further, for each coarse space V_j^c, we define

$$\mu_j^{-1}(V_j^c) := \max_{v_j \in V_j} \min_{v_j^c \in V_j^c} \frac{\|v_j - v_j^c\|_{D_j}^2}{\|v_j\|_{A_j}^2} \tag{6.13}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

and

$$\mu_c = \min_{1\le j\le J} \mu_j(V_j^c), \tag{6.14}$$

which is finite, thanks to Assumption 6.4(c) (*i.e.* (6.11)).

By the two-level convergence theory, if D_j provides a convergent smoother, then $(1-\mu_j(V_j^c))$ is the convergence rate for the two-level AMG method for V_j with coarse space V_j^c. The next theorem gives an estimate for the convergence of the two-level method in terms of the constants from Assumption 6.4 and μ_c.

Theorem 6.6. If Assumption 6.4 holds, then for each $v \in V$ we have the error estimate

$$\min_{v_c\in V_c} \|v - v_c\|_D^2 \le C_{p,1}C_{p,2}\mu_c^{-1}\|v\|_A^2. \tag{6.15}$$

Proof. By Assumption 6.4, for each $v \in V$, there exists $\underset{\sim}{v} \in \underset{\sim}{V}$ such that

$$v = \Pi_W \underset{\sim}{v}, \tag{6.16}$$

and (6.10) is satisfied.

By the definition of μ_c, we have

$$\|\underset{\sim}{v} - \underset{\sim}{Q}\underset{\sim}{v}\|_{\underset{\sim}{D}}^2 \le \mu_c^{-1}\|\underset{\sim}{v}\|_{\underset{\sim}{A}_W}^2. \tag{6.17}$$

We let $v_c = \Pi_W \underset{\sim}{Q}\underset{\sim}{v}$. Then $v_c \in V_c$, and by Assumption 6.4 we have

$$\begin{aligned}\|v - v_c\|_D^2 = \|\Pi_W(\underset{\sim}{v} - \underset{\sim}{Q}\underset{\sim}{v})\|_D^2 &\le C_{p,2}\|\underset{\sim}{v} - \underset{\sim}{Q}\underset{\sim}{v}\|_{\underset{\sim}{D}}^2\\ &\le C_{p,2}\mu_c^{-1}\|\underset{\sim}{v}\|_{\underset{\sim}{A}_W}^2 \le C_{p,1}C_{p,2}\mu_c^{-1}\|v\|_A^2.\end{aligned}$$ □

We define another product space

$$\underset{\sim}{V} := V_c \times V_1 \times V_2 \times \cdots \times V_J, \tag{6.18}$$

and we set $\Pi_c : V_c \mapsto V$ to be the natural inclusion from V_c to V. Then we define $\Pi : \underset{\sim}{V} \mapsto V$ by

$$\Pi := (\Pi_c\ \Pi_1\ \Pi_2\ \cdots\ \Pi_J) \tag{6.19}$$

and $\underset{\sim}{A} : \underset{\sim}{V} \mapsto \underset{\sim}{V}'$ by

$$\underset{\sim}{A}: = \begin{pmatrix} A_c & & & \\ & A_1 & & \\ & & \ddots & \\ & & & A_J \end{pmatrix}, \tag{6.20}$$

where $A_c : V_c \mapsto V_c'$ is given by

$$A_c := \Pi_c' APi_c \tag{6.21}$$

and $\underset{\sim}{B} : \underset{\sim}{V} \mapsto \underset{\sim}{V}'$ is given by

$$\underset{\sim}{B} := \begin{pmatrix} A_c^{-1} & & & & \\ & D_1^{-1} & & & \\ & & D_2^{-1} & & \\ & & & \ddots & \\ & & & & D_J^{-1} \end{pmatrix}. \tag{6.22}$$

We introduce the additive preconditioner $\widehat{B}$,

$$\widehat{B} := \Pi \underset{\sim}{B} \Pi' = \Pi_c A_c^{-1} \Pi_c' + \sum_{j=1}^{J} \Pi_j D_j^{-1} \Pi_j', \tag{6.23}$$

and we have the following results.

Lemma 6.7. If Assumption 6.4 holds, then for any $v \in V$ there exists $\underset{\sim}{v} \in \underset{\sim}{V}$ such that (6.4) holds, that is,

$$\|\underset{\sim}{v}\|_{\underset{\sim}{B}^{-1}} \leq \tilde{\mu}_0 \|v\|_A,$$

where $\tilde{\mu}_0$ is a constant depending on $C_{p,1}$, $C_{p,2}$, μ_c and c^D.

Lemma 6.8. If Assumption 6.9 holds, then (6.3) holds with constant $\tilde{\mu}_1$ depending on $C_{p,2}$ and c^D.

By directly applying Lemma 6.2, we immediately have the following result.

Theorem 6.9. If Assumption 6.4 holds, then

$$\kappa(\widehat{B}A) \leq \left(\frac{\tilde{\mu}_1}{\tilde{\mu}_0}\right)^2. \tag{6.24}$$

The following two-level convergence result is an application of the convergence theorem (Theorem 5.3) with the error estimate in Theorem 6.6.

Theorem 6.10. If Assumption 6.4 holds, then the two-level AMG method with coarse space defined in (6.12) converges with a rate

$$\|E\|_A^2 \leq 1 - \frac{\mu_c}{C_{p,1} C_{p,2} c^D}.$$

6.1. Bibliographical notes

Matsokin and Nepomnyashchikh (1985) first proved the fictitious space lemma. The work of Xu (1996) on the auxiliary space method is related. An additive version of Lemma 6.1 can be found in Xu and Zikatanov (2002), and the most general case (including multiplicative preconditioners) is found in Xu (2016).

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

In later sections we show how the general theory used here can be applied to various AMG algorithms, for example classical AMG (Brandt *et al.* 1982, Ruge and Stüben 1987), smoothed aggregation AMG (Míka and Vaněk 1992*a*, 1992*b*), spectral AMGe (Chartier *et al.* 2003, Efendiev, Galvis and Vassilevski 2011), and other algorithms.

Many of the works in the domain decomposition (DD) literature also use techniques for defining coarse spaces, which, to a large extent, have similar aims to the coarse space constructions for AMG outlined in this section. We refer to Toselli and Widlund (2005), Widlund (1994, 2009), Dohrmann, Klawonn and Widlund (2008), Spillane *et al.* (2014) and the references therein for more details on using local eigenspaces for constructing coarse spaces in DD methods.

7. Graphs and sparse matrices

In this section we give a brief introduction to the basic graph theory that is often used for sparse matrices and also for the study of AMG.

7.1. *Sparse matrix and its adjacency graph*

An *undirected graph* (or simply a *graph*) $\mathcal{G}$ is a pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite set of points called *vertices*, and $\mathcal{E}$ is a finite set of pairs of vertices, known as *edges*. We often write $\mathcal{V} = \{1, \ldots, n\}$ for some fixed n. We will not consider directed graphs in this paper because the graphs corresponding to the symmetric sparse matrices are undirected.

An edge $e \in \mathcal{E}$ is an unordered pair (j, k), where $j, k \in \mathcal{V}$. The vertices j and k are said to be *adjacent* if $(j, k) \in \mathcal{E}$. A *path* from a vertex j to a vertex k is a sequence $(j_0, j_1, j_2, \ldots, j_l)$ of vertices where $j_0 = j$, $j_l = k$, and $(j_i, j_{i+1}) \in \mathcal{E}$ for all $i = 0, 1, \ldots, l-1$. A vertex j is *connected* to a vertex k if there is a path from j to k. $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *connected* if every pair of vertices is connected by a path; otherwise it is said to be *disconnected*. A graph $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$ is called a *subgraph* of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if $\mathcal{V}_0 \subset \mathcal{V}$ and $\mathcal{E}_0 \subset \mathcal{E}$.

The *neighbourhood* $N(i)$ is the set of vertices adjacent to the vertex i. The *degree* or *valency* of a vertex is the number of edges that connect to it. These are defined as

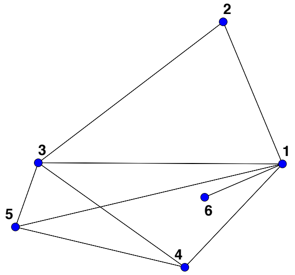
$$N(i) = \{j : (i, j) \in \mathcal{E}\}, \quad d_i = |\{j : (i, j) \in \mathcal{E}\}|. \tag{7.1}$$

A *path* connecting two vertices i and j is a sequence of edges $(k_0, k_1), (k_1, k_2), \ldots, (k_{m-1}, k_m)$ in $\mathcal{E}$ such that $k_0 = i$ and $k_m = j$. The length of the path is the number of edges in it. The *distance* between two vertices i and j is the length of the shortest path connecting i and j, and we denote it by $\operatorname{dist}(i, j)$. If i, j are not connected, then $\operatorname{dist}(i, j) = \infty$. The diameter of a graph is the largest distance between two vertices, that is, $\operatorname{diam}(\mathcal{G}) = \max_{(i,j)\in\mathcal{E}} \operatorname{dist}(i, j)$. An *independent set* is a set of vertices in which no two

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

$$A = \begin{pmatrix} * & * & * & * & * & * \\ * & * & * & 0 & 0 & 0 \\ * & * & * & * & * & 0 \\ * & 0 & * & * & * & 0 \\ * & 0 & * & * & * & 0 \\ * & 0 & 0 & 0 & 0 & * \end{pmatrix}$$

(a)



(b)

Figure 7.1. (a) A sparse symmetric matrix, and (b) its associated graph.

are adjacent. A *maximal independent set* is an independent set such that adding any other vertex to the set forces the set to contain an edge.

Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$, the *adjacency graph* of A is an undirected graph, denoted by $\mathcal{G}(A)$, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{1, 2, \ldots, n\}$. The edges $\mathcal{E}$ are defined as

$$\mathcal{E} = \{(j, k) : a_{jk} \neq 0\}.$$

A matrix A is called *irreducible* if its adjacency graph $\mathcal{G}(A) = (\mathcal{V}, \mathcal{E})$ is connected. Otherwise, A is called *reducible.*

An example of a symmetric matrix is shown in Figure 7.1(a) and the corresponding graph is shown in Figure 7.1(b). Pictorial representation of a graph is often not available, and a graph can be drawn in different ways with different coordinates of the vertices. As a general rule, sparse matrices do not provide any geometric information for the underlying graph and only the combinatorial or topological properties of $\mathcal{G}(A)$.

Given

$$S \subset \{1, \ldots, n\} \times \{1, \ldots, m\},$$

we define

$$\mathbb{R}_S^{n\times m} = \big\{X = (x_{ij}) \in \mathbb{R}^{n\times m} : x_{ij} = 0 \text{ if } (i, j) \notin S\big\}. \tag{7.2}$$

We say that X has sparsity pattern given by S if and only if $X \in \mathbb{R}_S^{n\times m}$.

Often, the sparsity pattern of a matrix is determined in advance and the set S is determined by a given matrix. For $Y \in \mathbb{R}^{n\times m}$, we let

$$\mathrm{S}(Y) = \big\{(i, j) : y_{ij} \neq 0\big\}.$$

We now consider the graphs associated with finite element or finite difference stiffness matrices. In the case of finite elements, this is the space of finite element functions V_h, and for finite difference discretizations this is the space of *mesh-functions* V_h, which can be identified with $\mathbb{R}^N$.

We assume that we have a finite-dimensional space V_h^N which we use to discretize the Neumann problem. We also have the finite element space for the Dirichlet (or mixed boundary condition) problem, and we assume that the following inclusions hold: $V_h = V_h^D \subset V_h^N$. Equivalently, we have a subspace of the variables which vanish on V_h: for example, the values of the finite element or finite difference solution at the nodes on the Dirichlet boundary vanish.

Let $A^{\mathcal{N}}$ be the matrix corresponding to the finite element or finite difference discretization of the model second-order elliptic equation with Neumann boundary conditions. Clearly we have the identity

$$(A^{\mathcal{N}}u, v) = \sum_{e\in\mathcal{E}} \omega_e \delta_e u \delta_e v. \tag{7.3}$$

Here, the sum is over all edges $\mathcal{E}$ of the graph $\mathcal{G}(A^{\mathcal{N}}) = (\mathcal{V}, \mathcal{E})$, $\delta_e v = v_i - v_j$, if $\mathcal{E} \ni e = (i, j)$, $i < j$. Also, $\omega_e = -(A^{\mathcal{N}})_{ij}$ are the off-diagonal entries of $A^{\mathcal{N}}$. Note that since we consider the Neumann problem, the bilinear form defined by $A^{\mathcal{N}}$ vanishes for u (resp. v) such that $u_i = 1$ (resp. $v_i = 1$) for all i. For both 5-point and 9-point stencils we have that $\omega_e = 1$ for all e.

We consider the stiffness matrix $A^{\mathcal{N}}$, corresponding to the model problem (2.1) with Neumann boundary conditions on a bounded domain $\Omega \subset \mathbb{R}^d$, that is, we have the boundary condition

$$\alpha \nabla u \cdot \boldsymbol{n} = 0 \quad \text{on } \partial\Omega, \tag{7.4}$$

where $\boldsymbol{n}$ is the unit normal vector to $\partial\Omega$ pointing outward. It is easy to derive the stiffness matrices corresponding to the Dirichlet or mixed boundary condition problem: we just restrict the bilinear form defined by $A^{\mathcal{N}}$ to the subspace given by

$$(Au, v) = \sum_{e\in\mathcal{E}} \omega_e \delta_e u \delta_e v, \quad u_j = v_j = 0 \quad x_j \in \Gamma_D. \tag{7.5}$$

Remark 7.1. Similar relations between differential problems with natural (Neumann) and essential (Dirichlet) boundary conditions are seen not only for the model problem considered here, but also for problems on $H(\text{curl})$, $H(\text{div})$, linear elasticity and others.

7.2. M-matrix relatives of finite element stiffness matrices

A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is called an *M-matrix* if it satisfies the following three properties:

$$a_{ii} > 0 \text{ for } i = 1, \dots, n, \tag{7.6}$$

$$a_{ij} \leq 0 \text{ for } i \neq j, \ i, \ j = 1, \dots, n, \tag{7.7}$$

$$A \text{ is semi-definite.} \tag{7.8}$$

As the first step in creating a space hierarchy, most AMG algorithms for $Au = f$ with positive semi-definite A use a simple filtering of the entries

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

of A, and construct an M-matrix which is then used to define crucial AMG components.

Definition 7.2 (M-matrix relative). We call a matrix $\widetilde{A}$ an M*-matrix relative* of A if $\widetilde{A}$ is an M-matrix and satisfies the inequalities

$$(v,v)_{\widetilde{A}} \lesssim (v,v)_A \quad \text{and} \quad (v,v)_D \lesssim (v,v)_{\widetilde{D}}, \quad \text{for all } v \in V, \tag{7.9}$$

where $\widetilde{D}$ and D are the diagonals of $\widetilde{A}$ and A respectively.

A few remarks are in order.

(i) We have used the term M-matrix to denote semi-definite matrices, and we are aware that this is not the precise definition. It is, however, much more convenient to refer to M-matrices, and we decided to relax the definition here in the hope that the inaccuracy would pay off by appealing to the reader.

(ii) We point out that the restricted M-matrix relatives are instrumental in the definition of coarse spaces and also in the convergence rate estimates. This is clearly seen later in Section 5.2 where we present the unified two-level theory for AMG.

(iii) Often, the case is that the one-sided inequality in (7.9) is in fact a spectral equivalence.

By definition, we have the following simple but important result.

Lemma 7.3. Let A_+ be an M-matrix relative of A and let D and D_+ be the diagonal matrices of A and A_+, respectively. If $V_c \subset V$ is a subspace, then the estimate

$$\|u - u_c\|_D^2 \lesssim \|u\|_A^2 \tag{7.10}$$

holds for some $u_c \in V_c$, given the estimate

$$\|u - u_c\|_{D_+}^2 \lesssim \|u\|_{A_+}^2. \tag{7.11}$$

This result means that we only need to work on the M-matrix relative of A in order to get the estimate (7.10).

In this section we show how to construct the M-matrix relative to the matrix resulting from a finite element discretization of the model problem (2.1) with linear elements. We first consider an isotropic problem with Neumann boundary condition (7.4) and isotropic $\alpha = a(x)I$. Construction of M-matrix relatives in the case of anisotropic tensor $\alpha(x)$ in (2.1) is postponed to Section 14.2.

In the rest of this section we make the following assumptions on the coefficient and the geometry of Ω.

- The domain $\Omega \subset \mathbb{R}^d$ is partitioned into simplices $\Omega = \cup_{T\in\mathcal{T}_h} T$.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

- The coefficient $a(x)$ is a scalar-valued function, and its discontinuities are aligned with the partition $\mathcal{T}_h$.
- We consider the Neumann problem, and hence the bilinear form (2.3) is

$$\int_\Omega a(x)\nabla v\cdot\nabla u = \sum_{(i,j)\in\mathcal{E}}(-a_{ij})\delta_e u\delta_e v = \sum_{e\in\mathcal{E}}\omega_e\delta_e u\delta_e v. \tag{7.12}$$

- It is well known that the off-diagonal entries of the stiffness matrix A are given by

$$\begin{aligned}\omega_e &= -(\phi_j,\phi_i)_A = \sum_{T\supset e}\omega_{e,T},\\ \omega_{e,T} &= \frac{1}{d(d-1)}\overline{a}_T|\kappa_{e,T}|\cot\alpha_{e,T}, \quad \overline{a}_T = \frac{1}{|T|}\int_T a(x)\,\mathrm{d}x.\end{aligned}$$

  Here, $e=(i,j)$ is a fixed edge with end-points x_i and x_j, $T\supset e$ is the set of all elements containing e, $|\kappa_{e,T}|$ is the volume of the $(d-2)$-dimensional simplex opposite to e in T, and $\alpha_{e,T}$ is the dihedral angle between the two faces in T not containing e.

- Let $\mathcal{E}$ denote the set of edges in the graph defined by the triangulation and let $\mathcal{E}^-$ be the set of edges where $a_{ij}\geq 0$, $i\neq j$. The set complementary to $\mathcal{E}^-$ is $\mathcal{E}^+=\mathcal{E}\setminus\mathcal{E}^-$. Then, with $\omega_e=-a_{ij}$ and $\delta_e u=(u_i-u_j)$, $e=(i,j)$, we have

$$\int_\Omega a(x)\nabla v\cdot\nabla u = \sum_{e\in\mathcal{E}^+}\omega_e\delta_e u\delta_e v - \sum_{e\in\mathcal{E}^-}|\omega_e|\delta_e u\delta_e v. \tag{7.13}$$

- We also assume that the partitioning is such that the constant function is the only function in the null space of the bilinear form (7.12). This is, of course, the case when Ω is connected (which is true, as Ω is a domain).

The non-zero off-diagonal entries of A may have either positive or negative sign, and usually $\mathcal{E}^-\neq\emptyset$. The next theorem shows that the stiffness matrix A defined via the bilinear form (7.12) is spectrally equivalent to the matrix A_+ defined as

$$(A_+u,v) = \sum_{e\in\mathcal{E}^+}\omega_e(u_i-u_j)(v_i-v_j). \tag{7.14}$$

Thus, we can ignore any positive off-diagonal entries in A, or equivalently we may drop all ω_e for $e\in\mathcal{E}^-$. Indeed, A_+ is obtained from A by adding to the diagonal all positive off-diagonal elements and setting the corresponding off-diagonal elements to zero. This is a stronger result that we will need later, because it gives spectral equivalence with the M-matrix relative of A_+.

Theorem 7.4. Assume that A is the stiffness matrix corresponding to linear finite element discretization of (2.1) with boundary conditions given

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

by (7.4). Then A_+ is an M-matrix relative of A which is spectrally equivalent to A. The constants of equivalence depend only on the shape-regularity of the mesh. Moreover, the graph corresponding to A_+ is connected.

A simple corollary, which we will use later to prove estimates for the convergence rate, is as follows.

Corollary 7.5. Assume that A is the stiffness matrix for piecewise linear discretization of equation (7.12) and A_+ is the M-matrix relative defined in Theorem 7.4. Then the diagonal D of A and the diagonal D_+ of A_+ are spectrally equivalent.

Proof. For the diagonal elements of A and A_+ we have

$$[D]_j = (\phi_j, \phi_j)_A \eqsim (\phi_j, \phi_j)_{A_+} = [D_+]_j.$$

The equivalences written above follow directly from Lemma 7.4. □

Corollary 7.5 together with Lemma 7.3 provides a theoretical foundation for using M-matrix relatives to design AMG for finite element matrices.

7.3. Bibliographical notes

We have introduced some standard notions from graph theory. For the reader interested in more detailed descriptions, we refer to the classical textbooks by Diestel (2010) and Gibbons (1985) as a general introduction to graph theory, and to Saad (2003) and Varga (2000) for considerations linking graphs, sparse matrices and iterative methods.

Our results on M-matrix relatives are related to the some of the works on preconditioning by Z-matrices and L-matrices (Kraus and Schicho 2006, Kraus 2008). They are implicitly used in most of the AMG literature (Ruge and Stüben 1987) where the classical connection strength definition gives an M-matrix. We point out that the M-matrix property and the existence of an M-matrix relative is often not sufficient to achieve even a two-level uniform convergence of AMG. A typical example is a matrix which has been rescaled and the constant is no longer in the kernel of the discrete operator. In this case, the standard AMG application may fail, and the near kernel needs to be recovered by different means, such as the adaptive AMG processes considered in Section 15 and the references given therein.

8. Connection strength

A central task in AMG is to obtain an appropriate coarse space or prolongation. This process is known as a coarsening process. In a geometric grid or, more generally, the adjacency graph of the stiffness matrix, we need to identify vertices to be deleted from the graph. We need the coarsened graph to still provide a good approximation for algebraically low frequencies.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

8.1. Basic idea and strength function

If a subset of vertices on which an algebraically smooth vector, say v, changes very slowly, we only need to keep one degree of freedom to represent v in this subset. In other words, we can either aggregate this subset together or keep one vertex and delete the rest of the vertices in this subset. We say that the vertices in this subset are strongly connected to each other. Connection strength is a concept introduced to identify strongly connected pairs of vertices. Roughly speaking, we say that i and j are strongly connected if $v_i \approx v_j$.

We imagine coarsening the graph in two stages: the first step is to remove some edges and the second step is to remove some vertices. The second step is the goal. Let us examine how the second step is carried out: we either (i) aggregate some neighbouring vertices together or (ii) pick a maximal independent set, denoted by $\mathcal{C}$, in the filtered graph and then remove all the remaining vertices. Using the argument above, (i) each aggregate should only consist of strongly connected vertices, or (ii) each of the deleted vertices should be strongly connected to some point in $\mathcal{C}$. To guarantee either of these two situations, we then have to remove all the weakly connected edges in the first step.

Let us further use some heuristic arguments to motivate how the connection strength should be defined. Let v be an algebraically smooth (5.40), namely

$$\|v\|_A^2 \le \epsilon \|v\|_{\bar{R}^{-1}}^2.$$

Let $u = v/\|v\|_{\bar{R}^{-1}}$. We then have

$$(Au, u) \le \epsilon \Rightarrow \sum_{e=(i,j)\in\mathcal{E}} (-a_{ij})(u_i - u_j)^2 \le \epsilon. \tag{8.1}$$

Thanks to Lemma 5.16, we can assume that A is an M-matrix, namely $-a_{ij} = |a_{ij}|$. From (8.1), we have the following observations.

(i) A larger $|a_{ij}|$ means a smaller $(u_i - u_j)^2$.

(ii) An algebraically smooth error varies more slowly in the direction where $|a_{ij}|$ is larger.

This observation leads to the following definition of connection strength. Given a threshold $\theta > 0$, we say that the vertex j of the adjacency graph is *θ-strongly connected* to vertex i if

$$-a_{ij} \ge \theta \max_{k\neq i} -a_{ik}. \tag{8.2}$$

Note that by the definition above we may have j strongly connected to i while i is not strongly connected to j. As a result, the adjacency graph corresponding to the matrix of strong connections may not be symmetric.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

However, our theoretical framework is given in terms of the symmetrized operator $\bar{R}$, regardless of whether the original smoother R is symmetric. This is due to the fact that we used the energy norm, namely the A-norm, to measure the convergence rate, and the resulting convergence rate is given in terms of $\bar{R}$. This is the best convergence theory we have, and we will use this theory to study AMG algorithms. As a result, we will only consider strength functions that are symmetric. A strength function

$$s_c : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}_+ \tag{8.3}$$

associated with an SPSD matrix is symmetric if $s_c(i,j) = s_c(j,i)$.

Given a threshold $\theta > 0$, we say that i and j are θ-strongly connected if

$$s_c(i,j) \geq \theta.$$

We then define the strength matrix:

$$S = \sum_{s_c(i,j) \geq \theta} e_i e_j^T. \tag{8.4}$$

Note that S is a Boolean matrix with entries equal to 0 or 1 depending on the connection strength.

Given the non-overlapping decomposition

$$\mathcal{V} = \bigcup_{i=1}^{m} \mathcal{A}_i = \bigcup_{\tilde{\mathcal{A}} \in \mathcal{V}_{\mathcal{A}}} \tilde{\mathcal{A}}, \quad \mathcal{V}_{\mathcal{A}} = (\mathcal{A}_1, \ldots, \mathcal{A}_m), \tag{8.5}$$

we extend the definition of strength function to

$$s_c : \mathcal{V}_{\mathcal{A}} \times \mathcal{V}_{\mathcal{A}} \mapsto \mathbb{R}_+, \tag{8.6}$$

and we assume that s_c is symmetric.

Given a threshold $\theta > 0$, we say that $\mathcal{A}_i$ and $\mathcal{A}_j$ are θ-strongly connected to each other if

$$s_c(\mathcal{A}_i, \mathcal{A}_j) \geq \theta.$$

An example of such a strength function is given in (8.13).

In the AMG literature, a number of heuristics have been proposed for identifying strong connections, particularly when considering discretizations of anisotropic equations. In general, connection strength is a notion that is difficult to address theoretically or to relate to the convergence rate of an algorithm. We refer to the classical papers and monographs mentioned in Section 9.6 for further discussions on related issues. Current trends in AMG development aim to re-evaluate the role of the classical definition of connection strength.

We would finally like to comment that connection strength is used to define the sparsity of P, and it is crucial, for example, in the proof of the convergence of the two-level method for discretizations of elliptic equations

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

with jump coefficients. Choosing the ‘right’ sparsity of P is crucial, as a denser P would lead to a better approximation from a coarser space, and a sparser P would lead to a less expensive algorithm.

In the rest of the section, we discuss different definitions of connection strength:

(i) classical AMG,

(ii) lean AMG,

(iii) local-optimization based AMG.

8.2. Classical AMG

With the above motivation, we define the strength function as follows:

$$\begin{aligned} s_c(i,j) &= \frac{-a_{ij}}{\min\bigl(\max_{k\neq i}(-a_{ik}), \max_{k\neq j}(-a_{jk})\bigr)} \\ &= \frac{a_{ij}}{\max\bigl(\min_{k\neq i} a_{ik}, \min_{k\neq j} a_{jk}\bigr)}. \end{aligned} \tag{8.7}$$

Definition (8.7) is a symmetrized version of the strength function used in the classical AMG literature (see (8.2)).

The following definition is also commonly used in classical AMG algorithms:

$$s_c(i,j) = \frac{|a_{ij}|}{\min\bigl(|N(i)|^{-1}\sum_{k\neq i}|a_{ik}|, |N(j)|^{-1}\sum_{k\neq i}|a_{jk}|\bigr)}. \tag{8.8}$$

Again, this is a symmetrized version of strength functions used in the AMG literature.

Finally, we may also have the following two definitions, which are based on Cauchy–Schwarz for SPSD matrices:

$$s_1(i,j) = \frac{|a_{ij}|}{\sqrt{a_{ii}a_{jj}}} \tag{8.9}$$

and

$$s_2(i,j) = \frac{-2a_{ij}}{a_{ii}+a_{jj}}. \tag{8.10}$$

Note that definition (8.7) (which is mostly associated with classical AMG) ignores all the non-negative entries of the stiffness matrix $A = (a_{ij})$.

8.3. Local-optimization strength function

Suppose that each pair of the index $\{i,j\} \subset \{1,\ldots,n\}$ is associated with a space V_{ij}, which is not necessarily a subspace of V. Assume now that

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

we have two operators: $A_{ij} : V_{ij} \mapsto V_{ij}'$, which is symmetric positive semi-definite, and $D_{ij} : V_{ij} \mapsto V_{ij}'$, which is symmetric positive definite.

Given a number $k_{ij} < \dim V_{ij}$, we define a coarse space $V_{ij}^c \subset V_{ij}$ as

$$V_{ij}^c := \operatorname{span}\{\zeta_{ij}^{(k)}, k = 1 : k_{ij}\},$$

where $\zeta_{ij}^{(k)}$ is the eigenvector corresponding to the kth smallest eigenvalue of $D_{ij}^{-1} A_{ij}$.

We let $Q_{ij} : V_{ij} \mapsto V_{ij}^c$ denote the orthogonal projection with respect to $(\cdot,\cdot)_{D_{ij}}$. Motivated by (6.13), we define the strength function s_c as follows:

$$s_c(i,j) := \left(\sup_{v \in V_{ij}} \frac{\|(I - Q_{ij})v\|_{D_{ij}}^2}{\|v\|_{A_{ij}}^2} \right)^{-1}. \tag{8.11}$$

A special case of the above definition is introduced in aggregation-based algebraic multigrid (AGMG). Suppose now that we have a set of aggregates $\{\mathcal{A}_1, \dots, \mathcal{A}_J\}$. We fix a pair $\{i,j\} \subset \{1,\dots,J\}$, and define

$$G = \mathcal{A}_i \bigcup \mathcal{A}_j.$$

We then let V_{ij} define the restriction of V to G, that is,

$$V_{ij} := \{v|_G : v \in V\}, \tag{8.12}$$

where

$$v|_G(x) = \begin{cases} v(x) & \text{if } x \in G, \\ 0 & \text{if } x \notin G. \end{cases}$$

We use A_{ij} and D_{ij}, respectively, to denote the restriction of A and D to G. We then choose $k_{ij} = 1$ and define the local coarse space V_{ij}^c by

$$V_{ij}^C = \operatorname{span}\{\zeta_G\}, \quad \zeta_G = \zeta_{ij}^{(1)},$$

and the orthogonal projection $Q_{ij} : V_{ij} \mapsto V_{ij}^C$ with respect to $(\cdot,\cdot)_{D_{ij}}$, namely,

$$Q_{ij} v = \frac{(v, \zeta_G)_{D_{ij}}}{\|\zeta_G\|_{D_{ij}}^2} \zeta_G.$$

The strength function based on aggregation is defined as

$$s_c(i,j) := \left(\sup_{v \in V_{ij}} \frac{\|(I - Q_{ij})v\|_{D_{ij}}^2}{\|v\|_{A_{ij}}^2} \right)^{-1}. \tag{8.13}$$

Another example is choosing $V_{ij} = \mathbb{R}^2$ and

$$A_{ij} = \begin{pmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{pmatrix}, \quad D_{ij} = \begin{pmatrix} a_{ii} & 0 \\ 0 & a_{jj} \end{pmatrix}.$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

We choose $k_{ij} = 1$ and define the coarse space $V_{ij}^c \subset V_{ij}$ as

$$V_{ij}^c = \operatorname{span}\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}.$$

By direct computation, the strength function defined in (8.11) is given by

$$s_c(i,j) = \frac{1 - s_1^2}{1 - s_2}, \quad s_1 = \frac{|a_{ij}|}{\sqrt{a_{ii}a_{jj}}}, \quad s_2 = -\frac{2a_{ij}}{a_{ii} + a_{jj}}. \tag{8.14}$$

We note that $s_1 \geq s_2$ and hence

$$s_c(i,j) \leq 1 + s_2 \leq 1 + s_1. \tag{8.15}$$

We point out that the strength function given by (8.14) is obtained by using the theory in Section 6, while the other strength functions such as (8.7) are obtained by heuristic considerations.

8.4. Lean AMG

Instead of using the absolute value of matrix entries as the criterion to determine if two points are strongly coupled, lean AMG uses *affinity* to measure the connection strength, which is based on the following heuristic observation: *Given a vector v, if (i,j) is a strong connected pair of vertices, after several relaxation on v, namely*

$$v \leftarrow (I - RA)^\nu v,$$

the values of v_i and v_j should be close.

In lean AMG, we generate K test vectors. Each test vector is the result of applying ν Gauss–Seidel relaxation sweeps to $Ax = 0$, starting from randomly generated vectors $x^{(1)}, \dots, x^{(K)} \in \mathbb{R}^n$. Further, we denote

$$X_{n\times K} := \begin{pmatrix} X_1^T \\ \vdots \\ X_n^T \end{pmatrix} = (I - RA)^\nu (x^{(1)} \ \cdots \ x^{(K)}). \tag{8.16}$$

Here X_i^T is the ith row of $X_{n\times K}$. The strength function for lean AMG is then defined as

$$s_c(i,j) := \frac{|(X_i, X_j)|^2}{(X_i, X_i)(X_j, X_j)}. \tag{8.17}$$

8.5. Bibliographical notes

Classical algorithms for determining connection strength can be found in Brandt *et al.* (1982), Stüben (1983), Brandt *et al.* (1985), Ruge and Stüben (1987) and Briggs, Henson and McCormick (2000). The original measure of connection strength given by Ruge (1983, 1985), McCormick and Ruge

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

(1989) and Brandt *et al.* (1985) is non-symmetric, but for theoretical considerations, which only depend on the symmetrized smoothers, it suffices to use the slightly more restrictive but symmetric versions of connection strength.

Some extensions of these classical algorithms for defining strong connections are based on different measures for connectivity and distance such as *measure of importance* and *algebraic distance*. Details can be found in Ruge and Stüben (1987) and Trottenberg *et al.* (2001, Appendix A).

Connection strength functions have had little theoretical backing in the past. The results developed in this section, such as local Poincaré inequalities and especially the strength functions (8.15), show that such heuristics are reasonable, and their choice can be motivated by theoretical results.

For aggregation AMG, typically a symmetric connection strength function is used, as defined by Vaněk *et al.* (1996). Some recent aggregation algorithms also define connection strength based on sharp theoretical results, and use the local two-level convergence rate (8.11) as a measure (Notay 2010, Napov and Notay 2012).

For aggregations based on matching (aggregates of size 2), the 'heavy edge' matching algorithms in Karypis and Kumar (1998) correspond to a connection strength function selecting aggregates depending on edge weights in the adjacency graph. Some recent works (*e.g.* Livne and Brandt 2012) use connection strength functions based on the size of the entries in the Gram matrix formed by a set of smoothed test vectors.

9. Coarsening strategies

Once the smoother is identified, the central task of an AMG method is to identify a sequence of coarse spaces, in functional terminology, or equivalently, in an algebraic setting, to identify a sequence of prolongation matrices. This procedure is known as 'coarsening'. In this section we will discuss this coarsening procedure.

Roughly speaking, given equation (1.1), namely $Au = f$, on a vector space V, the goal is to find a subspace $V_c \subset V$ such that the solution $u_c \in V_c$ of the 'coarsened' problem

$$A_c u_c = f_c, \quad A_c = \imath_c' A \imath_c, f_c = \imath_c' f \tag{9.1}$$

would provide a good approximation to the original solution $u \in V$. More specifically, the solution of u_c of (9.1) would provide a good approximation to those 'algebraically smooth' components of the error for which the given smoother does not converge well.

9.1. *Motivations*

In some sense, 'coarsening' is done almost everywhere in numerical analysis. For example, the finite element equation (2.19) can be viewed as a coarsened equation of the original equation (2.3). In this case the finite element space V_h is a coarsened subspace of $V = H_0^1(\Omega)$.

It is therefore informative for us to examine how a finite element space is constructed in general. While there are many different ways to construct finite element spaces, mathematically speaking, the most convenient approach is via the use of 'degrees of freedom', which refers to a basis of a dual space. More specifically, in a finite element discretization, the finite element space V_h is obtained by first specifying the dual space V_h' (the so-called space of degrees of freedom, or variables). For linear finite elements, $V_h' = \{\psi_i : i = 1 : n_h\}$ is such that

$$\psi_i(v) = v(x_i^h).$$

With such variables (nodal evaluation), we then find a dual basis $\{\phi_i : i = 1 : n_h\}$ which are piecewise linear functions such that

$$\psi_i(\phi_j) = \delta_{ij}, \quad 1 \leq i, j \leq n.$$

The finite element space V_h is then defined by

$$V_h = \operatorname{span}\{\phi_i : i = 1 : n_h\}.$$

In fact, mathematically speaking, all existing finite element spaces V_h can be obtained by first constructing V_h. This is the approach taken in the classical literature on finite element methods: see Ciarlet (2002).

In a similar way to the finite element method, we will therefore focus on techniques for constructing a coarse space $V_c \subset V$ by first identifying its dual basis V_c'. Such an approach is rather abstract, but it turns out to be more intrinsic, more general, and, in fact, more commonly used in the AMG literature, albeit implicitly.

It is interesting to note that we rarely use the word 'coarsening' in the design of a geometric multigrid method. Instead, we use a 'refinement' procedure to define a sequence of nested spaces. As an example, Figure 9.1 shows a uniformly refined triangular grid used for discretization of the Poisson equation with linear finite elements.

In AMG, we do not have the luxury of this hierarchy of spaces given by a geometric refinement. Instead, we carry out a reverse-engineering of the refinement process, namely *coarsening.*

Conceivably, we could also use such a reverse-engineering process to recover geometric multigrid methods by starting from the finest geometric grid, at least for some special cases: for example, if all triangles in the triangulation shown in Figure 9.1 were acute and the graph corresponding to the mesh were also an adjacency graph of the stiffness matrix. It is clear that the coarse grid vertices, that is, the set $\mathcal{C}$ (known from the refinement), is a maximal independent set (MIS) of vertices in the graph corresponding to the refined mesh. In the simplest case, the coarse grid variables (the dual

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

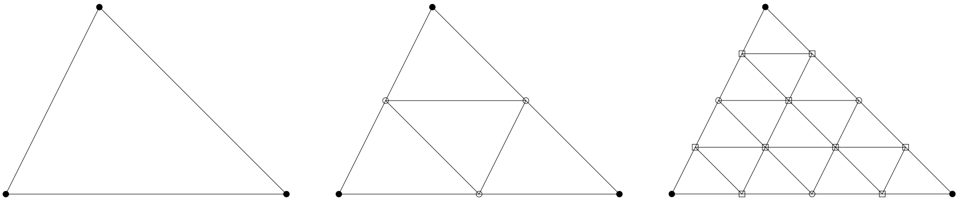


Figure 9.1. Regular refinement of a coarse grid element: •, coarsest level vertices; ○, first level of refinement; □, second level of refinement.

basis for V_c) precisely correspond to the MIS. This observation is explored later, in Section 9.3.1, when constructing algorithms in the classical AMG framework for a selection of coarse grid vertices via the MIS algorithm (Algorithm 5). For further examples of relations between geometric AMG and GMG coarsening, we refer to Section 10.2.

The above reverse-engineering for GMG gives some hint as to how a coarsening process needs to be done in AMG, but we need to study the process in a broader framework and – more importantly – we will use the degrees of freedom, *i.e.* the dual bases, to obtain coarse spaces. In the above GMG example, each grid point in the geometric grid corresponds to exactly one variable. However, this is not always the case in applications.

9.2. Basic approach

By mimicking the construction of finite element spaces as described above, given a linear algebraic system of equation $Au = f$, we adapt a coarsening strategy consisting of the following steps.

1 We consider the adjacency graph $\mathcal{G}(A)$ of the coefficient matrix A. Based on a certain strength function s_c as described in Section 8, we remove the weakly connected edge in $\mathcal{G}(A)$, that is, we drop certain entries in A to form a filtered matrix $\tilde{A}$.

2 We carry out one of the following two substeps.

 (a) *Classical AMG.* Find a maximal independent set of $\mathcal{G}(\tilde{A})$ to form the set of coarse vertices $\mathcal{C}$. Then remove the rest, namely $\mathcal{V} \setminus \mathcal{C} \equiv \mathcal{F}$.

 (b) *Aggregation AMG.* Agglomerate using some greedy algorithm: pick a point and agglomerate its neighbours and go from there.

3 We obtain a subset of variables obtained from the above steps, *i.e.* $(V')_c$.

4 We use $(V')_c$ to define a high-frequency space $V_{hf} = (V')_c^{(0)}$ by

$$V_{hf} = (V')_c^{(0)} := \{v \in V : \langle g, v\rangle = 0,\ \text{for all } g \in (V')_c\}. \tag{9.2}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

5 Find a tentative coarse space W_c such that

$$V = V_{hf} \oplus W_c. \tag{9.3}$$

6 Apply a certain postprocessing (such as smoothing) to W_c to obtain V_c:

$$V_c = SW_c.$$

7 Using V_c, or equivalently the prolongation P, we form the coarse matrix $A_c = P^T AP$.

8 We then repeat the above steps to A_c in place of A until a desirable coarsest level is reached.

9.2.1. Construction of $(V')_c$

Given $A \in \mathbb{R}^{n\times n}$ and the associated graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we proceed as follows.

(i) Form the two non-overlapping decompositions

$$\mathcal{V} = \mathcal{C} \cup \mathcal{F}, \quad \mathcal{C} = \bigcup_{i=1}^{n_c} \mathcal{A}_i.$$

(ii) Identify $(V')_c = \operatorname{span}\{N_i : i = 1 : n_c\} \subset V'$.

Here are three examples that will be discussed in detail in later sections (see Sections 11, 12 and 13).

Aggregation AMG. $\mathcal{F} = \emptyset$ and

$$N_i(v) = \langle v\rangle_{\mathcal{A}_i} \equiv \frac{1}{|\mathcal{A}_i|}\sum_{j\in\mathcal{A}_i}\psi_j(v) = \frac{1}{|\mathcal{A}_i|}\sum_{j\in\mathcal{A}_i} v_j, \quad i = 1 : n_c. \tag{9.4}$$

Classical AMG. $\mathcal{F} \neq \emptyset$ and

$$\mathcal{A}_i = \{k_i\}, \quad N_i(v) = \psi_{k_i}(v) = v_{k_i}. \tag{9.5}$$

In this case, $\mathcal{C}$ usually consists of disconnected vertices.

Energy-min AMG. $\mathcal{F} \neq \emptyset$ and $\mathcal{A}_i$ are aggregates

$$N_i(v) = \langle v\rangle_{\mathcal{A}_i} \equiv \frac{1}{|\mathcal{A}_i|}\sum_{j\in\mathcal{A}_i}\psi_j(v) = \frac{1}{|\mathcal{A}_i|}\sum_{j\in\mathcal{A}_i} v_j, \quad i = 1 : n_c. \tag{9.6}$$

9.2.2. Construction of V_c

Given coarse grid degrees of freedom $(V')_c \subset V'$, we define V_{hf} as in (9.2). The following lemma shows how to find a subspace W_c, a 'pre-coarse space', such that $V = V_{hf} \oplus W_c$.

Lemma 9.1. If $\phi_{k,c}$, $k = 1, \dots, n_c$ are elements of V such that the Gram matrix

$$G = (G_{km}) = (N_m(\phi_{k,c}))$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

is non-singular, then we have

$$V = V_{hf} \oplus W_c, \quad W_c = \mathrm{span}\{\phi_{k,c}\}_{k=1}^{n_c}.$$

Proof. We first show that $W_c \cap V_{hf} = \{0\}$. In fact, if

$$v := \sum_{k=1}^{n_c} (\tilde{v})_k \phi_{k,c} \in W_c \cap V_{hf},$$

then we have

$$0 = N_m(v) = \sum_{k=1}^{n_c} N_m(\phi_{k,c}) = \sum_{k=1}^{n_c} (\tilde{v})_k G_{km}, \quad m = 1, \dots, n_c.$$

Hence,

$$G\tilde{v} = 0,$$

and since, by assumption, G is non-singular, we must have $v = 0$. Therefore, $W_c \cap V_{hf} = \{0\}$.

Next, for any $v \in V$, we define $w_c \in W_c$ as

$$w_c = \sum_{k=1}^{n_c} (\widetilde{w}_c)_k \phi_{k,c}, \quad \widetilde{w}_c = G^{-1} \begin{pmatrix} N_1(v) \\ \vdots \\ N_{n_c}(v). \end{pmatrix}.$$

It is immediate to check that

$$N_m(v - w_c) = 0, \quad m = 1, \dots, n_c,$$

and hence $(v - w_c) \in V_{hf}$. This proves that

$$v = w_c + \underbrace{(v - w_c)}_{\in V_{hf}},$$

and completes the proof. □

Lemma 9.1 gives us a way to construct a subspace W_c such that $V = V_{hf} \oplus W_c$.

Lemma 9.2. Assume the coarse grid degrees of freedom are defined as

$$N_k := \sum_{j \in \mathcal{A}_k} \alpha_j \psi_j, \quad k = 1, \dots, n_c,$$

where $\sum_{j \in \mathcal{A}_k} \alpha_j = 1$ and $\{\psi_j\}$ is the dual basis of $\{\phi_j\}$. If $\{\phi_{k,c} : k = 1, \dots, n_c\}$ are elements of V such that the Gram matrix

$$G = (N_l(\phi_{k,c})) = I,$$

then we have

$$\phi_{k,c} = \sum_{j \in \mathcal{A}_k} \phi_j + v_{hf}, \quad v_{hf} \in V_{hf}. \tag{9.7}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Proof. We fix a k and consider the subset $W_k \subset V$ such that

$$W_k := \{v \in V : N_k(v) = 1 \text{ and } N_l(v) = 0, \text{ for all } l \neq k\}.$$

Pick any $v_1, v_2 \in W_k$. We have

$$N_l(v_1 - v_2) = N_l(v_1) - N_l(v_2) = 0, \quad \text{for all } l = 1, \dots, n_c,$$

and this shows that $(v_1 - v_2) \in V_{hf}$.

Furthermore, if we define $v_k^c = \sum_{j \in \mathcal{A}_k} \phi_j$, then

$$N_k(v_k^c) = \sum_{j \in \mathcal{A}_k} \sum_{i \in \mathcal{A}_k} \alpha_j(\psi_j, \phi_i) = \sum_{j \in \mathcal{A}_k} \alpha_j = 1,$$

and $N_l(v_k^c) = 0$ for all $l \neq k$ since $\mathcal{A}_k$ and $\mathcal{A}_l$ have no overlap. We then have $v_k^c \in W_k$, and hence

$$W_k = \{v_k^c + v_{hf} : v_{hf} \in V_{hf}\}.$$ □

Next, we describe how to construct the coarse space V_c using W_c.

Lemma 9.3. Assume that $V = V_{hf} \oplus W_c$ and $\varphi_{1,c}, \dots, \varphi_{n_c,c}$ is a basis in W_c. Then $\phi_{k,c} = S\varphi_{k,c}$, $k = 1, \dots, n_c$ are linearly independent if $S : V \mapsto V$ satisfies one of the following conditions:

(a) S maps a linear independent set in W_c into a linear independent set in V,

(b) S is invertible,

(c) $S = I - Q_{hf}$, where $Q_{hf} : V \mapsto V_{hf}$.

As a result, we have

$$V = V_{hf} \oplus V_c, \quad V_c = \operatorname{span}\{S\varphi_{k,c} : k = 1, \dots, n_c\}.$$

Proof. We only need to prove case (c), as cases (a) and (b) are trivial. If $\phi_{k,c}$ were linearly dependent, there would be a linear combination of $\{\phi_{k,c}\}_{k=1}^{n_c}$ which vanishes. Equivalently, this means that there exists $w_c \in W_c$ such that $(I - Q_{hf})w_c = 0$, which implies that $w_c \in V_{hf}$. Since $V_f \cap W_c = \{0\}$, we have that $w_c = 0$ and the only vanishing linear combination in $\operatorname{span}\{\phi_{k,c}\}_{k=1}^{n_c}$ is the trivial one, and this completes the proof. □

Remark 9.4. **(i)** For smoothed aggregation, $S = I - \omega D^{-1}A$ for some properly chosen ω so that S is non-singular, or maps a special linearly independent set of vectors to a linearly independent set of vectors (see Section 13).

(ii) For classical AMG with the ideal interpolation, $S = I - Q_{hf}$, and Q_{hf} is the A-orthogonal projection (see Section 12).

(iii) For classical AMG with the standard interpolation, $S = I - Q_{hf}$, and Q_{hf} is an approximation to the ideal interpolation (see Section 12).

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Algorithm 5 Maximal independent set method

```
1: Set $C \leftarrow \emptyset$, $i \leftarrow 1$.
2: If $i$ and all its neighbours are not visited, then set $C \leftarrow C \cup \{i\}$ and
   mark $i$ and all vertices in $N(i)$ as visited.
3: If all vertices are visited, then output $C$ and stop; else set $i \leftarrow i + 1$
   and go to 2.
```

To give a summary of the discussions above, the coarsening algorithms in AMG are methods for determining the coarse grid degrees of freedom, or coarse grid variables. Such algorithms are based on selecting degrees of freedom associated with subsets of vertices in the adjacency graph that correspond to the matrix A, or to the strength matrix S, as is done in geometric coarsening, when the hierarchy of meshes or adjacency graphs is known.

9.3. Two basic coarsening algorithms

In the next two subsections we present typical algorithms for finding the coarse grid degrees of freedom. Each such degree of freedom is associated with a vertex or a subset of a graph. Two types of algorithm are distinguished: classical AMG algorithms pick coarse grid degrees of freedom that correspond to a maximal independent set of vertices in the adjacency graph of the strength matrix, and aggregation-based AMG algorithms use a splitting of the adjacency graph of the strength matrix in connected subgraphs.

9.3.1. A maximal independent set algorithm

Here we present a simple 'greedy' maximal independent set (MIS) algorithm, which has been used in classical AMG algorithms to identify coarse grid degrees of freedom. Given the adjacency graph of the strength matrix, the simple greedy MIS algorithm is shown in Algorithm 5.

Remark 9.5. We note that the MIS algorithm recovers the geometric coarsening if the vertices are visited in such an order that coarse grid vertices are ordered first and all the connections in Figure 9.1 are strong. This is obvious, but nevertheless shows that geometric coarsening can sometimes be recovered by an algebraic algorithm.

Let us point out that for finite element stiffness matrices obtained via adaptive refinement algorithms, the hierarchy of vertices is naturally included in the refinement procedure. For regular refinement this choice of maximal independent set is illustrated in Figure 9.1.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Algorithm 6 Greedy aggregation method

Input: Graph $\mathcal{G}$ with n vertices. **Output:** $\mathcal{V} = \cup_{k=1}^{n_c} \mathcal{V}_k$, and $\mathcal{V}_k \cap \mathcal{V}_j = \emptyset$ when $k \neq j$.

1: Set $n_c \leftarrow 0$ and for $k = 1 : n$ do:

   a. If k and all its neighbours have not been visited, then (i) we set $n_c \leftarrow n_c + 1$, (ii) we label with n_c the subgraph whose vertices are k and the neighbours of k, and (iii) we mark k and all its neighbours as visited.

   b. If at least one neighbour of k has been visited, we continue the loop over the vertices.

2: Since after this procedure there might be vertices which do not belong to any aggregate (but definitely have a neighbouring aggregate), we add each such vertex to a neighbouring aggregate and we pick the one which has a minimal number of vertices in it.

3: The algorithm ends when all vertices are in a subset.

9.3.2. An aggregation algorithm

The class of algorithms known as aggregation algorithms refers to the splitting of the adjacency graph of the strength matrix into a union of connected subgraphs. Let $\{\mathcal{V}_k\}_{k=1}^{n_c}$ be a non-overlapping splitting of the set of vertices

$$\mathcal{V} = \cup_{k=1}^{n_c} \mathcal{V}_k, \quad \mathcal{V}_j \cap \mathcal{V}_k = \emptyset, \quad \text{for } j \neq k.$$

We then define

$$\mathcal{E}_k = \{(l, m) \in \mathcal{E} : l \in \mathcal{V}_k \text{ and } m \in \mathcal{V}_k\} \tag{9.8}$$

to be the set of edges associated with $\mathcal{V}_k$.

An aggregation can be done in many different ways, some very sophisticated. In general, all combinatorial graph-partitioning algorithms can be used for aggregation. We will not consider such algorithms in detail, however, but provide the basic and most important example of the greedy aggregation algorithm: see Algorithm 6. This algorithm can be applied recursively to provide a multilevel hierarchy of aggregates.

9.3.3. Aggressive coarsening

The extended strong connections and the corresponding strength operator are used to construct coarse spaces of smaller dimension. This procedure is also known as aggressive coarsening. We recall the definition of a path in the graph given in Section 7.1, and all our considerations concern the adjacency graph $\mathcal{G}(S)$ of the strength matrix $S \in \mathbb{R}^{n\times n}$ defined in (8.4). Aggressive coarsening refers to a selection of coarse grid vertices as the independent set in the adjacency graph. Specifically, we choose vertices whose distance is larger than 2 with respect to the strength operator.

Definition 9.6 (strong connection along a path). A vertex i is said to connect strongly to a vertex j along a path of length l if there exists a path $(k_0, k_1, \dots, k_l)$ in $\mathcal{G}(S)$ such that $k_0 = i$, $k_l = j$ and $s_c(k_m, k_{m+1}) \geq \theta$, $m = 0, 1, \dots, l-1$.

The next definition is related to the number of strongly path-connected vertices.

Definition 9.7 ((m, l)-strong connection). For given integers $m > 0$ and $l > 0$, a vertex i is (m, l)-strongly connected to a vertex j if and only if i strongly connects to j along at least m paths of length l (*cf.* Definition 9.6).

An aggressive coarsening algorithm generates a maximal independent set using Algorithm 5 for the graph $\mathcal{G}_{m,l} = (\mathcal{V}, \mathcal{E}_{m,l})$ with a set of vertices $\mathcal{V} = \{1, \dots, n\}$ and a set of edges $\mathcal{E}_{m,l}$ defined as

$$\mathcal{E}_{m,l} := \{(i,j) : i \text{ is } (m,l) \text{ strongly connected to } j\}. \tag{9.9}$$

As is well known (Diestel 2010), $(S^l)_{ij}$ is non-zero if and only if there is a path of length $\leq l$ between i and j. An aggressive coarsening exploits this property, and is an algorithm which selects a set of coarse grid degrees of freedom corresponding to vertices in the graph which are at graph distance larger than l. It uses the adjacency graph $\mathcal{G}(S^l)$ of S^l in place of $\mathcal{G}(S)$ in an aggregation or a maximal independent set algorithm.

As an example, let us consider aggressive coarsening with $l = 2$ and $m = 1$. The set of coarse grid degrees of freedom is obtained by applying the standard MIS algorithm twice. First we find a maximal independent set in $\mathcal{G}(S)$ and obtain a set of coarse grid degrees of freedom C (these are at graph distance at least 2). Then we apply the MIS algorithm for a second time to the graph with vertices the C-points, the edges between them being given by the strength operator corresponding to S^2.

Similarly, for aggregation, an aggressive coarsening corresponds to applying Algorithm 6 recursively, or applying it directly to the graph corresponding to S^l for a given l.

9.4. Adaptive coarsening for classical AMG

An adaptive coarsening algorithm is an algorithm which adaptively chooses the coarse grid degrees of freedom based on a given definition of strength function based on the smoother in a two-grid algorithm. One example of adaptive coarsening follows from the classical compatible relaxation introduced by Brandt. The algorithm takes as input a smoother which leaves

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

the coarse grid variables invariant and only smooths the components in the algebraic high-frequency space V_{hf}.

A typical adaptive coarsening algorithm follows the steps given below.

Step 0. Set $k = 0$ and choose $(V')_{k,c} \subset V'$, for example, using the MIS or aggregation method introduced in Section 9.3.

Step 1. Define $V_{k,f}$ as the subspace of V which is annihilated by the functionals in $(V')_{k,c}$, namely

$$V_{k,f} = \{v \in V : (g, v) = 0, \text{ for all } g \in (V')_{k,c}\}.$$

Step 2. Let $\imath_{k,f} : V_{k,f} \mapsto V$ be the natural inclusion operator, and compute an estimate $\rho_{k,f}$ of the norm of the smoother on $V_{k,f}$ using

$$\rho_{k,f} \approx \sup_{v \in V_{k,f}} \frac{\|(I - \imath_{k,f} R_{k,f} \imath'_{k,f} A)v\|_A^2}{\|v\|_A^2}.$$

Here, $R_{k,f}$ could be the restriction of the smoother R on $V_{k,f}$, or more generally, any relaxation on $V_{k,f}$.

Step 3. Given a threshold $\delta_f > 0$, if $\rho_{k,f} > \delta_f$, we set $k = k + 1$, add more functionals to $(V')_c$ and go to **Step 1**. Otherwise, we set $(V')_c = (V')_{k,c}$, and accordingly $V_f = V_{k,f}$ and stop the iteration.

In **Step 3**, if the stopping criterion is not satisfied, we need to enrich the space $(V')_c$ by extending the set. One example of doing so is introduced in compatible relaxation methods by extending the set $\mathcal{C}$ using the following procedure.

First we randomly choose a vector $v^0 \in V_{k,f}$, and form

$$v = (I - \imath_{k,f} R_{k,f} \imath'_{k,f} A)^\nu v^0 \quad \text{for some } \nu \geq 1.$$

Then, with a given threshold $\theta \in (0, 1)$, we let

$$\mathcal{C}_0^1 = \Big\{ i \in \mathcal{F} : |v_i| > \theta \max_k |v_k| \Big\}.$$

and

$$\mathcal{C}_1 = \mathcal{C}_0 \cup \mathrm{MIS}(\mathcal{C}_0^1).$$

Finally, we update $\mathcal{C} \leftarrow \mathcal{C}_1$ and proceed with the next compatible relaxation iteration.

As is clear from the algorithm outlined above, we can use any of the definitions of connection strength to obtain $(V')_{0,c}$ at **Step 0**. When $\rho_{k,f} > \delta_f$, it means that $\mathcal{C}$ obtained from the strength function s_0 is not satisfactory, namely, too coarse. This means that either the threshold for s_0 is too small or s_0 itself is not satisfactory. We could still use s_0, but with a smaller threshold to obtain a $\mathcal{C}_0'$ that is bigger than $\mathcal{C}_0$ but does not necessarily

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

contain $\mathcal{C}_0$. For example, if we initially use the (m,l)-strong connection defined in Definition 9.7, $\mathcal{C}$ can be extended by increasing the value of m or decreasing the value of l. This approach may not be computationally efficient. A more effective approach, as used in compatible relaxation, is to find candidates for additional C-points by examining the filtered matrix

$$A^{(1)} = \{a_{ij} : i, j \in \mathcal{F}_0\},$$

where $\mathcal{F}_0 = \Omega \setminus \mathcal{C}_0$, to get the set of coarse grid degrees of freedom $\mathcal{C}_0^1$ and add them to $\mathcal{C}_0$ to extend the size of $\mathcal{C}$.

9.5. AGMG coarsening: a pairwise aggregation

Aggregation-based algebraic multigrid (AGMG) uses the strength function of (8.13) to form aggregations such that the local convergence rate (see (6.13)) on each aggregate is bounded by a given threshold. The main idea of the algorithm can be explained as follows.

It first splits the index set Ω into aggregates, each of which has at most two elements, namely

$$\Omega = \bigcup_j \mathcal{A}_j^{(0)}, \quad \mathcal{A}_i^{(0)} \bigcap \mathcal{A}_j^{(0)} = \emptyset \quad \text{and } |\mathcal{A}_j^{(0)}| \leq 2. \tag{9.10}$$

This process is done by a greedy algorithm. At each step, the algorithm finds the pair $G = \{i, j\}$ for which the strength function defined in (8.14) is maximal.

Using these pairs as aggregates $\{\mathcal{A}_j^{(0)}\}$, we form the unsmoothed aggregation prolongation P, which is piecewise constant with respect to the aggregates and P has orthogonal columns.

We let $A^{(0)} := A$ and $P^{(0)} := P$. Then $A^{(1)} := (P^{(0)})^T A^{(0)} P^{(0)}$. We then apply the pairwise aggregation algorithm on $A^{(1)}$ and find larger aggregates $\{\mathcal{A}_j^{(1)}\}$. Each $\mathcal{A}_j^{(1)}$ is a union of two pairs in $\{\mathcal{A}_j^{(0)}\}$ which minimize the strength function defined in (8.13). Then we obtain $P^{(1)}$ and $A^{(2)}$. Applying this procedure recursively, we obtain the final aggregates $\mathcal{A}_j$, each of which is a union of several pairs in $\{\mathcal{A}_j^{(0)}\}$.

The pairwise aggregation strategy aims to find the aggregates on which the Poincaré constant $\mu_j(V_j^c)^{-1}$ is bounded, which is introduced in our abstract framework by (6.13). As stated in the abstract convergence theorem, bounding $\mu_j(V_j^c)^{-1}$ will bound the convergence rate of the AMG method.

9.6. Bibliographical notes

The coarsening strategies used in AMG are basically of two types: the first uses connection strength to define a 'strength' graph and then performs the greedy aggregation or MIS algorithms, and the other is based on algorithms

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

such as compatible relaxation coarsening, which uses the smoother to detect slowly converging components. These are heuristic approaches, which work well on a certain class of problems but rarely have a theoretical justification for their efficiency as AMG splitting algorithms.

Classical algorithms for selection of the coarse grid degrees of freedom are found in Brandt *et al.* (1982), Stüben (1983), Brandt *et al.* (1985), Ruge and Stüben (1987), and the MG tutorial of Briggs *et al.* (2000).

Parallel coarse-grid selection algorithms are found in Sterck, Yang and Heys (2006) and, in combination with scalable interpolation algorithms, in De Sterck, Falgout, Nolting and Yang (2008). Coarsening using information about discretization, *i.e.* AMGe, is given in Jones and Vassilevski (2001) and Brezina *et al.* (2001). Spectral AMGe coarsening is considered in detail in Chartier *et al.* (2003). Many of the 'upscaling' and related techniques in homogenization (see Efendiev, Hou and Wu 2000, Hou, Wu and Cai 1999), resemble the coarsening procedures introduced in the classical and modern AMG literature.

More sophisticated maximal independent set (MIS) algorithms for selection of coarse grid degrees of freedom, using different measures for connectivity and distance in the graph corresponding to A, are found in Ruge and Stüben (1987) and Trottenberg *et al.* (2001, Appendix A). Most of these algorithms are refinements of the greedy algorithm given in this section. For parallel versions, we refer to Luby (1986), Cleary, Falgout, Henson and Jones (1998) and Sterck *et al.* (2006) for specific details on parallel and parallel randomized MIS algorithms. Other coarsening schemes that are also suitable for parallel implementation are the coupled and decoupled coarsening schemes (Yang 2006, Henson and Yang 2002).

Regarding the aggregation coarsening methods, we refer to Vakhutinsky, Dudkin and Ryvkin (1979), Blaheta (1986) and Marek (1991) for earlier work on such methods. The greedy aggregation algorithm presented here is found in Vaněk *et al.* (1996). A special class of aggregation coarsening methods based on matching were first employed by Karypis and Kumar (1998) for fast graph partitioning, and later used in several of the AMG methods. One example is the AGMG algorithm described in Section 9.5 and found in Napov and Notay (2012) and Notay (2012). The algorithm given by Kim, Xu and Zikatanov (2003) also uses this coarsening approach. Special matching techniques which optimize matrix invariants were used by D'Ambra and Vassilevski (2013, 2014). The compatible relaxation (CR) algorithm, first introduced by Brandt (2000) and further investigated by Livne (2004), Falgout and Vassilevski (2004) and Brannick and Falgout (2010), is a device that reduces the role of connection strength to only define initial set of coarse grid degrees of freedom and then use the smoother to select additional degrees of freedom. Other coarse/fine degrees of freedom partitioning algorithms are considered in MacLachlan and Saad (2007) from

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

both classical and compatible relaxation perspectives. The aggregation algorithms, which aggregate vertices based on a local measure for two-level convergence (Notay 2010, Napov and Notay 2012, Livne and Brandt 2012), are somewhat different adaptive coarsening algorithms.

10. GMG, AMG and geometry-based AMG

Historically, the algebraic multigrid method was motivated by the geometric multigrid method. In this section we will explore the relationship between these two types of method.

10.1. Geometric multigrid method

We begin our discussion with a simple one-dimensional model problem, namely (2.1) for $d = 1, \Omega = (0, 1)$ and $\alpha \equiv 1$ with zero Dirichlet boundary condition. For any integer N, we consider a uniform grid, denoted by $\mathcal{T}_h$, of the interval $[0, 1]$ as follows:

$$0 = x_0 < x_1 < \cdots < x_{N+1} = 1, \quad x_j = \frac{j}{N+1}(j = 0 : N + 1). \tag{10.1}$$

This partition consists of uniform sub-intervals with length $h = 1/(N + 1)$, that is, $\mathcal{T}_h = \bigcup_i \{\tau_i\}$, where $\tau_i = (x_{i-1}, x_i)$ for $d = 1$. Such a uniform partition is shown in Figure 10.1.

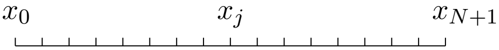


Figure 10.1. One-dimensional uniform grid.

We define a linear finite element space associated with the partition $\mathcal{T}_h$:

$$V_h = \{v : v \text{ is continuous and piecewise linear w.r.t. } \mathcal{T}_h, \, v(0) = v(1) = 0\}. \tag{10.2}$$

Let $V_h = V_h$. Recall from the previous section that the finite element approximation of our model problem is then $u_h \in V_h$, satisfying (2.19). We introduce the operator $A_h : V_h \mapsto V_h$ such that

$$(A_h v_h, w_h) = a(v_h, w_h), \quad v_h, w_h \in V_h.$$

Then the finite element solution u_h satisfies

$$A_h u_h = f_h, \tag{10.3}$$

where $f_h \in V_h$ is the L^2-projection of f: $(f_h, v_h) = (f, v_h)$, $v_h \in V_h$.

To describe a geometric multigrid algorithm, we 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.

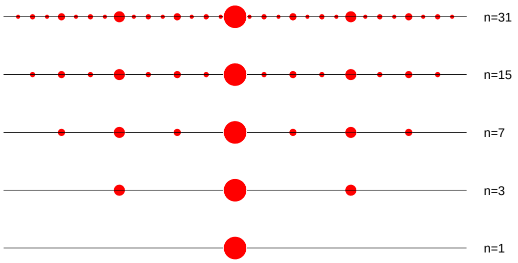


Figure 10.2. Multiple grids in one dimension.

One simple definition of the grid points in $\mathcal{T}_k$ is

$$x_i^k = \frac{i}{2^k}, \quad i = 0, 1, 2, \dots, N_k + 1, \ k = 1, 2, \dots, J,$$

where $N_k = 2^k - 1$. Note that $\mathcal{T}_k$ can be viewed as being obtained by adding midpoints of the sub-intervals in $\mathcal{T}_{k-1}$. For each k, the set of above nodes will be denoted by $\mathcal{N}_k$. This is illustrated in Figure 10.2.

For $k = 1 : J$, similar to the finite element space V_h defined as in (10.2), we define the finite element space V_k associated with the grid $\mathcal{T}_k$, to obtain a nested sequence of finite element spaces as follows:

$$V_1 \subset \cdots \subset V_k \subset \cdots \subset V_J. \tag{10.4}$$

The classical V-cycle GMG method simply applies Algorithm 4 recursively with the following setting:

(i) $\mathcal{V}_k = V_k$,

(ii) $A_k : V_k \mapsto V_k$ defined by

$$(A_k u_k, v_k) = a(u_k, v_k), \quad u_k, v_k \in V_k,$$

(iii) $\imath_{k-1}^k : V_{k-1} \mapsto V_k$, the inclusion operator,

(iv) $R_k : V_k \mapsto V_k$ corresponding to a smoother such as the Gauss–Seidel method.

Algebraic setting

Equation (10.3) may be called the operator form of the finite element equation. To get an equation in terms of vectors and matrix, we use the nodal basis functions for V_h:

$$\phi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{h} & x \in [x_{i-1}, x_i], \\ \dfrac{x_{i+1} - x}{h} & x \in [x_i, x_{i+1}], \\ 0 & \text{elsewhere}. \end{cases} \tag{10.5}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

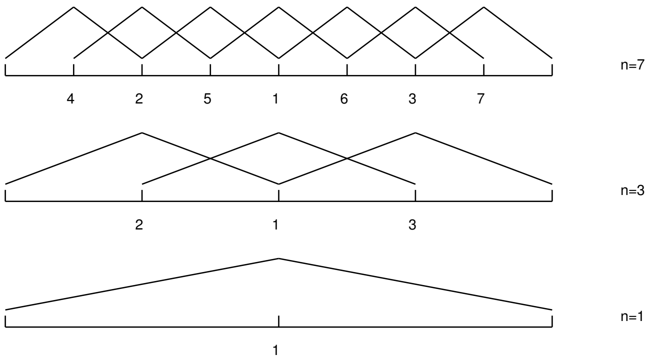


Figure 10.3. One-dimensional nodal basis functions on each level.

On each level, as in (10.5), we introduce a set of nodal basis functions $\{\phi_i^{(k)} : i = 1 : N_k\}$ for the finite element space V_k. This is illustrated in Figure 10.3.

Each $v \in V_k$ can be uniquely written as a linear combination of the basis functions:

$$v = \xi_1 \phi_1^{(k)} + \xi_2 \phi_2^{(k)} + \cdots + \xi_{N_k} \phi_{N_k}^{(k)}. \tag{10.6}$$

This gives an isomorphism from V_k to $\mathbb{R}^{N_k}$, which maps $v \in V_k$ to $\mu \in \mathbb{R}^{N_k}$, that is,

$$v = \xi_1 \phi_1^{(k)} + \xi_2 \phi_2^{(k)} + \cdots + \xi_{N_k} \phi_{N_k}^{(k)} \longrightarrow \mu = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_{N_k} \end{pmatrix}. \tag{10.7}$$

Here μ is called the *matrix representation* of v. Recall from the discussion in Section 2.2, giving a basis of V_h defined in (10.5), that (2.19) is equivalent to the linear system equations in (2.20).

We introduce the auxiliary space $V_k := \mathbb{R}^{N_k}$. The transition operator P_k^{k+1} from V_k to V_{k+1} is a matrix in $\mathbb{R}^{N_{k+1} \times N_k}$, and satisfies

$$(\phi_1^{(k)} \cdots \phi_{N_k}^{(k)}) = (\phi_1^{(k+1)} \cdots \phi_{N_{k+1}}^{(k+1)}) \imath_k^{k+1}. \tag{10.8}$$

For the special one-dimensional problem we are now considering, for $k = 1, 2, \ldots, J-1$ we have

$$\phi_j^{(k)} = \frac{1}{2} \phi_{2j-1}^{(k+1)} + \phi_{2j}^{(k+1)} + \frac{1}{2} \phi_{2j+1}^{(k+1)}. \tag{10.9}$$

The matrix that encodes this relation is

$$P_k^{k+1} = \begin{pmatrix} \frac{1}{2} & & & & \\ 1 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ & 1 & & & \\ & \frac{1}{2} & \frac{1}{2} & & \\ & & 1 & & \\ & & \frac{1}{2} & & \\ & & & \ddots & \frac{1}{2} \\ & & & & 1 \\ & & & & \frac{1}{2} \end{pmatrix}. \tag{10.10}$$

The classical V-cycle AMG method is obtained by simply applying Algorithm 4 recursively with the following setting:

(i) $V_k = \mathbb{R}^{N_k}$,

(ii) $A_k \in \mathbb{R}^{N_k \times N_k} : V_k \mapsto V_k$ defined by

$$(A_k)_{ij} = a(\phi_i^{(k)}, \phi_j^{(k)}), \quad 1 \le i, j \le N_k,$$

(iii) $P_{k-1}^k : V_{k-1} \mapsto V_k$, a matrix in $\mathbb{R}^{N_{k+1} \times N_k}$ defined by (10.8),

(iv) $R_k : V_k \mapsto V_k$ corresponding to a smoother such as the Gauss–Seidel method.

A similar multigrid algorithm can be obtained for problems in two and three dimensions as long as we have a multiple level of grids and the corresponding finite element spaces on each level.

10.2. Obtaining AMG from GMG

The first obstacle to extending GMG to AMG is the geometric information used in GMG. However, a close inspection of GMG reveals that a GMG method only depends on the following two major ingredients:

(i) the stiffness matrix corresponding to the finest grid,

(ii) the prolongation matrix on each level.

Once all the prolongation matrices are given, the stiffness matrices on all coarser levels are given by

$$A_k = P_k^T A P_k, \quad k = J-1, J-2, \ldots, 1, \quad P_k = \prod_{j=k}^{J-1} P_j^{j+1}. \tag{10.11}$$

Of course, a smoother is also needed on each level, but its definition can be considered purely algebraic for the moment.

As the stiffness matrix on the finest grid is always available in any given application, the only thing left is the prolongation matrices. We will now use the example of the linear finite element method to discuss the relationship between the prolongation matrix and geometric information. Two observations are highly relevant.

Observation 1. The prolongation matrix only depends on the natural graph associated with the underlying grid, but not on the coordinates of grid points.

Observation 2. The graph of the underlying grid is very close to the adjacency graph of the stiffness matrix.

Based on the above discussions, roughly speaking, we can essentially recover a geometric multigrid method for the stiffness matrix corresponding to the continuous linear finite element discretization of the Laplace equation by only using the algebraic and graph information provided by the stiffness matrix:

(i) form the adjacency graph $\mathcal{G}(A)$ of the stiffness matrix A,

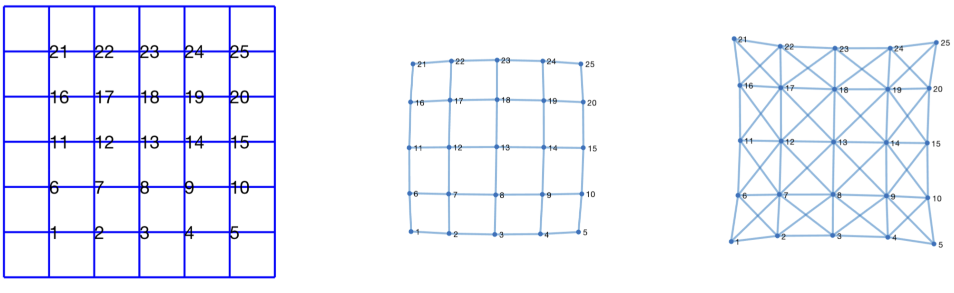
(ii) coarsen $\mathcal{G}(A)$.

As an illustrative example, let us consider the stiffness matrices corresponding to a discretization of the Laplace equation on a square domain with bilinear elements. It is well known (Ciarlet 2002) that the stiffness matrix in this case is the same as the scaled matrix for the 9-point finite difference stencil (2.18). The corresponding adjacency graph shown in Figure 10.4(c) is denser (has more edges) than the mesh graph shown in Figure 10.4(a). The set of its edges includes the diagonals of each of the squares forming the mesh. For the construction of the prolongation/interpolation matrix, we recall that the prolongation matrix gives the coefficients of the expansion of a coarse grid basis function on a grid of size $2h$ in terms of the finer grid basis on a grid of size h. Locally this matrix looks as follows:

$$[(P_{2h}^h)^T]_i = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}.$$

This matrix is often called a ‘prolongation stencil’, and it shows the coefficients in the expansion of a coarse grid basis function in a compact form. In the centre we have the coefficient 1, in front of the fine grid basis function associated with a coarse grid vertex. The rest of the entries refer to the coefficients corresponding to the fine grid basis functions in the expansion.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press



(a) (b) (c)

Figure 10.4. (a) 6 × 6 uniform grid, (b) graph of the matrix corresponding to the 5-point finite difference stencil, and (c) graph of the matrix corresponding to the 9-point finite difference stencil.

On regularly refined triangular grids we have an analogous situation. We refer to Section 9 for details of the selection of coarse grid degrees of freedom in this case. Prolongation and restriction matrices only depend on the topological structure of this graph. Similar observations led to the development of the AMG: if the geometric coordinates are unknown, different avenues for constructing coarse spaces are needed, leading to different variants of AMG algorithms.

10.3. Obtaining GMG from AMG

In this section we use the unified theory in Section 6 to obtain GMG from AMG. The main ingredients needed are the spaces V_j, operators Π_j, A_j, D_j, and coarse spaces V_j^c.

We now consider constructing a two-level geometric multigrid method for (2.1) (or the variational formulation (2.3)). Suppose we have two grids: a fine grid $\mathcal{T}_h$ and a coarse grid $\mathcal{T}_H$. On each grid we define a linear finite element space V_h and V_H with nodal basis functions $\{\phi_j^h\}$ and $\{\phi_j^H\}$, respectively, and we consider the following partition of the domain Ω:

$$\overline{\Omega} = \bigcup_{j=1}^{J} \overline{\Omega}_j, \quad \text{with } \overline{\Omega}_j = \operatorname{supp}(\phi_j^H). \tag{10.12}$$

We then define V_j as

$$V_j := \{\chi_j v : v \in V_h\}, \tag{10.13}$$

where χ_j is the characteristic function of Ω_j. We note that V_j is not a subspace of $H^1(\Omega)$. The operator $\Pi_j : V_j \mapsto V_h$ is defined as follows:

$$\Pi_j v_j := I_h(\phi_j^H v_j), \quad \text{for all } v_j \in V_j,\ j = 1, \dots, J, \tag{10.14}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

where I_h is the nodal interpolation operator on the fine grid. Here we note that $\phi_j^H v_j$ is continuous in Ω. We notice that, by definition, $\phi_j^H \chi_j = \phi_j^H$ on Ω, which implies the identities

$$\sum_{j=1}^{J} \Pi_j \chi_j v = I_h\left(\sum_{j=1}^{J} \phi_j^H \chi_j v\right) = I_h\left(\sum_{j=1}^{J} \phi_j^H v\right) = I_h(v) = v,$$

and hence

$$\sum_{j=1}^{J} \Pi_j \chi_j = \mathrm{Id}.$$

The operator $A_j : V_j \mapsto V_j'$ is the local restriction of the bilinear form $a(\cdot,\cdot)$, namely,

$$(A_j u_j, v_j) := a(u_j, v_j)_{\Omega_j} = \int_{\Omega_j} \alpha(x) \nabla u_j \cdot \nabla v_j, \quad u_j, v_j \in V_j. \tag{10.15}$$

The following estimate tells us that A_j satisfies (6.10) with decomposition $v = \sum_{j=1}^{J} \Pi_j v_j$ where $v_j = \chi_j v$:

$$\sum_{j=1}^{m_c} \|v_j\|_{A_j}^2 = \sum_{j=1}^{m_c} a(v,v)_{\Omega_j} \leq C_1 \|v\|_A^2,$$

where C_1 depends on the number of overlaps of Ω_j.

We choose $D : V \mapsto V'$ using the fine grid basis functions as follows:

$$(D\phi_i^h, \phi_j^h) := a(\phi_i^h, \phi_j^h)\delta_{ij}, \quad 1 \leq i, j \leq n, \tag{10.16}$$

and

$$(Du, v) := \sum_{j=1}^{n} a(\phi_j^h, \phi_j^h) u_j v_j, \quad u, v \in V. \tag{10.17}$$

Note that $(D\cdot,\cdot)$ is an inner product on V which induces a norm $\|\cdot\|_D$.

In the above definition, the matrix representation of D is the diagonal of the matrix representation of A. In a similar way, we can define $D_j : V_j \mapsto V_j'$ using the basis functions for V_j.

By a simple scaling argument, we obtain

$$\|v\|_0^2 \eqsim h^d \|v\|_{\ell^2}^2.$$

From (10.17), using inverse inequality, we then have

$$\|v\|_D^2 \lesssim h^{-2} \sum_{j=1}^{n} \|\phi_j^h\|_0^2 v_j^2 \eqsim h^{d-2} \|v\|_{\ell^2}^2 \eqsim h^{-2} \|v\|_0^2.$$

We also have

$$\|I_h v\|_0^2 \eqsim h^d \|I_h v\|_{\ell^2}^2 = h^d \|v\|_{l^2}^2 \eqsim \|v\|_0^2.$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Assumption 6.9 can then be verified by

$$\begin{aligned}\Big\|\sum_{j=1}^J \Pi_j v_j\Big\|_D^2 &\lesssim h^{-2}\Big\|\sum_{j=1}^J \Pi_j v_j\Big\|_0^2 = h^{-2}\Big\|I_h\Big(\sum_{j=1}^J \phi_j^H v_j\Big)\Big\|_0^2\\ &\eqsim h^{-2}\Big\|\sum_{j=1}^J \phi_j^H v_j\Big\|_0^2 = h^{-2}\sum_{i,j=1}^J \int_\Omega \phi_i^H v_i \phi_j^H v_j\\ &\le h^{-2}\sum_{i,j=1}^J \|\phi_i^H\|_\infty \|\phi_j^H\|_\infty \int_\Omega v_i v_j \le h^{-2}\sum_{i,j=1}^J \int_{\Omega_i \bigcap \Omega_j} v_i v_j\\ &\le C_o \sum_{j=1}^J h^{-2}\int_{\Omega_j} |v_j|^2 \lesssim C_o \sum_{j=1}^J \|v_j\|_{D_j}^2.\end{aligned}$$

By the definition of A_j, the kernel of A_j consists of all constant functions in V_j, and we choose V_j^c to be the one-dimensional space of constant functions on Ω_j. Then

$$\mu_j(V_j^c) = \lambda_j^{(2)}, \tag{10.18}$$

where $\lambda_j^{(2)}$ is the second smallest eigenvalue of the operator $D_j^{-1}A_j$.

The global coarse space V_c is defined as in (6.12). Note that in this case it is easy to show, by definition, that the coarse space V_c constructed by (6.12) is in fact identical to V_H, namely

$$V_c = \operatorname{span}\{\phi_j^H : j = 1, \ldots, J\}. \tag{10.19}$$

By Theorem 6.10, the convergence rate of this two-level geometric multigrid method depends on the $\min_j(\lambda_j^{(2)})$. If the Poincaré inequality is true for each V_j, namely

$$\inf_{v_c \in V_j^c} \|v - v_c\|_{D_j}^2 \le c_j \|v\|_{A_j}^2, \quad \text{for all } v \in V_j, \tag{10.20}$$

with c_j a constant, then the two-level geometric multigrid method converges uniformly.

10.4. Spectral AMGe: a geometry-based AMG

We now consider the *element*-based AMG approaches, and also define the ingredients needed to fit these methods into the general results in Section 6. The AMGe methods are less algebraic, as they assume an underlying grid and use element local stiffness matrices to define interpolation operators. In the AMGe setting, it is assumed that we know the decomposition of the $n \times n$ matrix A, that is,

$$A = \sum_{\tau \in \mathcal{T}} \tilde{A}_\tau, \tag{10.21}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

where $\mathcal{T}$ is the set of finite elements used to discretize the problem and, for each element $\tau \in \mathcal{T}$, $\tilde{A}_\tau$ is the zero-extension of the local stiffness matrix A_τ on τ (which is symmetric positive semi-definite).

To define V_j, D_j, A_j and Π_j in Section 6, corresponding to AMGe, we partition the domain Ω into disjoint subdomains, $\Omega_1, \dots, \Omega_J$. Each subdomain is an agglomeration of elements, and

$$\bar{\Omega} = \bigcup_{j=1}^{J} \bar{\Omega}_j.$$

For each subdomain Ω_j, we introduce the cutoff operator $\chi_j : V \mapsto V_j$, whose action on $v \in V$ is defined by

$$(\chi_j v)(x) := \begin{cases} v(x) & \text{if } x \in \bar{\Omega}_j, \\ 0 & \text{if } x \notin \bar{\Omega}_j. \end{cases} \tag{10.22}$$

Then we define the space V_j by

$$V_j := \chi_j V. \tag{10.23}$$

Here A_j is defined by summing up all the associated stiffness matrices for elements in Ω_j, namely

$$A_j := \sum_{\tau \subset \Omega_j} \tilde{A}_\tau. \tag{10.24}$$

Clearly, A_j is symmetric positive semi-definite. It is easy to verify that (6.10) holds. In fact, we have the following equations:

$$\sum_{j=1}^{J} \|\chi_j v\|_{A_j}^2 = \sum_{j=1}^{J} a(v,v)_{\Omega_j} = \sum_{\tau \in \mathcal{T}} a(v,v)_\tau = \|v\|_A^2. \tag{10.25}$$

If we denote the diagonal of A by D, then D_j is a diagonal matrix defined by

$$[D_j]_{ii} := \begin{cases} D_{ii} & \text{if } i \in \Omega_j, \\ 0 & \text{if } i \notin \Omega_j. \end{cases} \tag{10.26}$$

The operator Π_j is provided by diagonal matrices defined as

$$[\Pi_j]_{ii} := \begin{cases} [A_j]_{ii}/[A]_{ii} & \text{if } i \in \Omega_j, \\ 0 & \text{if } i \notin \Omega_j. \end{cases} \tag{10.27}$$

Note that $[\Pi_j]_{ii} = 1$ if i is an inner point of Ω_j. Inequality (6.9) is verified in Lemma 12.4.

On each local space V_j, spectral AMGe chooses locally the 'best' coarse space V_j^c, which is the subspace spanned by eigenvectors of $D_j^{-1}A_j$ belonging

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

to its m_j^c smallest eigenvalues. The global coarse space is then defined by (6.12).

By the abstract convergence theorem (Theorem 5.3), the convergence rate of two-level spectral AMGe depends on the minimum of each m_j^c+1 smallest eigenvalue of $D_j^{-1}A_j$ on V_j. More precisely,

$$\|E\|_A \leq 1 - \frac{\mu_c}{C_{p,1}C_{p,2}},$$

with $\mu_c = \min_{1\leq j\leq J} \mu^{(j)}_{m_j^c+1}$.

10.5. Bibliographical notes

The main ideas of the GMG method were first demonstrated in the pioneering works by Fedorenko (1961, 1964) and Bahvalov (1966). Similar ideas, using group relaxation methods, can be traced back to the works of Southwell (1940, 1946). The first description of a truly multigrid method is found in the seminal work by Brandt (1973). Further developments in multilevel methods are those of Brandt (1977) and Hackbusch (1979, 1978). These works have attracted a lot of attention from the computational mathematics and engineering communities. Advances in the convergence analysis of multigrid methods have been made by Nicolaides (1975, 1977), Bank and Dupont (1980), Braess and Hackbusch (1983), Bramble and Pasciak (1987), Bramble, Pasciak and Xu (1990, 1991*a*), Bramble, Pasciak, Wang and Xu (1991*c*) and Xu (1992).

BoxMG (Dendy 1982, 1983) is a method that uses geometrically refined grids and defines interpolation using algebraic techniques. We refer to Dendy (1982, 1983) and de Zeeuw (1990) for details, and also to MacLachlan, Moulton and Chartier (2012) for results on the equivalence between BoxMG and classical AMG.

Element-based AMG approaches are less algebraic, as they assume an underlying grid and use element stiffness matrices to define interpolation operators. Such methods include *plain* AMGe, *element-free* AMGe, *spectral* AMGe and *spectral agglomerate* AMGe, and are developed to improve AMG robustness for finite element based AMG. For results and discussions on different flavours of AMGe we refer to Jones and Vassilevski (2001), Brezina *et al.* (2001), Henson and Vassilevski (2001) and Brezina *et al.* (2006).

11. Energy-min AMG

Here we consider the energy-minimization algorithms for the construction of coarse spaces. While this is not historically the first AMG approach to coarsening, we focus on this technique first, as it can be used to motivate most other AMG algorithms.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

11.1. Energy-minimization versus trace-minimization

In the next theorem we add a constraint to Theorem 5.8 and give a relation between the optimal coarse space $V_c^{\rm opt}$ and the energy-minimization. We refer to Section 5.2 for the definition of $P^{\rm opt}$ and $\mathcal{X}_\eta$.

Theorem 11.1 (trace-minimization theorem). Given $\eta > 0$, let $\mathcal{Z}_\eta$ be defined as

$$\mathcal{Z}_\eta = \bigl\{P \in \mathbb{R}^{n\times n_c} : (Pv, Pv)_{\bar{R}^{-1}} \geq \eta(v,v),\ v \in \mathbb{R}^{n_c} \text{ and } P\mathbf{1} = \sqrt{n_c\eta}\zeta_1\bigr\}. \tag{11.1}$$

Then $P \in \arg\min_{Q\in\mathcal{Z}_\eta} \operatorname{trace}(Q^TAQ)$ if

$$P \in \mathcal{Z}_\eta \quad \text{and} \quad \operatorname{Range}(P) = \operatorname{Range}(P^{\rm opt}).$$

Let $\hat{P} = \bar{R}^{-1/2}P$ and define

$$\mathcal{Y}_\eta = \bigl\{P \in \mathbb{R}^{n\times n_c} : (Pv, Pv) \geq \eta(v,v), v \in \mathbb{R}^{n_c} \text{ and } P\mathbf{1} = \sqrt{n_c\eta}\hat{\zeta}_1\bigr\}, \tag{11.2}$$

where $\hat{\zeta}_j$ is the eigenvector corresponding to the jth smallest eigenvalue of $\bar{R}^{1/2}A\bar{R}^{1/2}$. It is clear that $\bar{R}^{1/2}A\bar{R}^{1/2}$ and $\bar{R}A$ have the same spectrum. Theorem 11.1 can be written as follows.

Theorem 11.2. Given $\eta > 0$, let $\mathcal{Y}_\eta$ be defined as in (11.2). Then,

$$P \in \arg\min_{Q\in\mathcal{Y}_\eta} \operatorname{trace}(Q^T\bar{R}^{1/2}A\bar{R}^{1/2}Q)$$

$$\text{if } P \in \mathcal{Y}_\eta \text{ and } \operatorname{Range}(P) = \operatorname{Range}(P^{\rm opt}).$$

Suppose now that we have a bilinear form $a(\cdot,\cdot)$ on V which is symmetric positive semi-definite, and an inner product $(\cdot,\cdot)_{\bar{R}^{-1}}$ on V. Here, for example, the operator $\bar{R}$ is the scaled parallel (resp. successive) subspace correction method corresponding to the splitting of V as

$$V = \sum_{i=1}^n \operatorname{span}\{\phi_i\}.$$

In practice, $\bar{R}$ can be a symmetrization of any A-norm convergent smoother on V. Here, for simplicity, we choose $\bar{R} = D^{-1}$.

We now consider a finite element space V with basis functions

$$\{\phi_j : j = 1 : n\}.$$

Let $P = (p_{ij}) \in \mathbb{R}^{n\times n_c}$ be such that

$$(\phi_1^c, \dots, \phi_{n_c}^c) = (\hat{\phi}_1, \dots, \hat{\phi}_n)P, \quad \hat{\phi}_j = \phi_j/\|\phi_j\|_A, \quad j = 1 : n. \tag{11.3}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Let $\{(\mu_j, \zeta_j)\}$ denote the eigenpairs of $\bar{R}A$, and $\hat{\zeta}_j = \zeta_j/\|\zeta_j\|_A$. We then define

$$X_\eta = \bigl\{(\phi_1^c, \dots, \phi_{n_c}^c) : (\phi_1^c, \dots, \phi_{n_c}^c) \text{ satisfies (11.3) with } P \in \mathcal{Y}_\eta\bigr\}. \tag{11.4}$$

We consider the minimization problem

$$\min_{(\phi_1^c,\dots,\phi_{n_c}^c)\in X_\eta} \sum_{j=1}^{n_c} \|\phi_j^c\|_A^2. \tag{11.5}$$

Note that

$$\begin{aligned}\sum_{j=1}^{n_c} \|\phi_j^c\|_A^2 &= \sum_{j=1}^{n_c} a(\phi_j^c, \phi_j^c) = \sum_{j=1}^{n_c} a\biggl(\sum_{k=1}^{n} p_{kj}\hat{\phi}_k, \sum_{l=1}^{n} p_{lj}\hat{\phi}_l\biggr)\\ &= \sum_{j=1}^{n_c}\sum_{k=1}^{n}\sum_{l=1}^{n} p_{kj} a(\hat{\phi}_k, \hat{\phi}_l) p_{lj} = \operatorname{trace}(P^T D^{-1/2}\tilde{A} D^{-1/2} P).\end{aligned}$$

Then Theorem 11.1 implies that

$$\operatorname{span}\{\zeta_j, 1 \le j \le n_c\} = \operatorname{span}\{\phi_j^0, 1 \le j \le n_c\}, \tag{11.6}$$

where

$$(\phi_1^0, \dots, \phi_{n_c}^0) \in \operatorname*{arg\,min}_{(\phi_1^c,\dots,\phi_{n_c}^c)\in X_\eta} \sum_{j=1}^{n_c} \|\phi_j^c\|_A^2. \tag{11.7}$$

In the next section we use the functional setting, and provide details of the design of energy-minimizing bases.

11.2. Energy-minimization basis for AMG and Schwarz methods

The discussion above motivates the computation of an energy-minimizing basis as the solution of a global optimization problem with constraint. This could be of concern regarding the efficiency of the proposed approach. As we show later in this section, however, this is not a concern because the optimization problem is well conditioned and can be solved efficiently. We also show below that the basis functions solving the energy-minimization problem are locally harmonic within each coarse grid 'element'. This property of the energy-minimizing basis suggests that various 'harmonic extension' techniques, used to define coarse spaces in multigrid methods (see Chan, Xu and Zikatanov 1998, Brezina *et al.* 2001, Jones and Vassilevski 2001) are very closely related to the energy-minimization algorithms. This property also suggests that the energy-minimizing basis may also be used for numerical homogenization for multiscale problems (see Efendiev, Hou and Wu 2000, Hou, Wu and Cai 1999).

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

We start our description with a given set of subdomains Ω_i with the property that none of the subdomains is fully contained in the union of the rest of them. More precisely, we have

$$\Omega = \bigcup_{i=1}^{n_c} \Omega_i \quad \text{and} \quad \bar{\Omega}_i \bigcap \bigg(\bigcup_{j \neq i} \Omega_j \bigg)^c \neq \emptyset, \tag{11.8}$$

where the superscript c is the standard set-complement. Equivalently, in the purely algebraic setting, when there is no function space in the background, we may choose Ω_i to be a subset of vertices of the adjacency graph corresponding to a matrix A. The aim is to construct basis functions $\{\phi_i^H\}_{i=1}^{n_c}$ that are in $\mathcal{X}_\eta$, with the following additional restrictions:

$$\operatorname{supp}(\phi_i^H) \subset \bar{\Omega}_i, \quad 1 \leq i \leq n_c.$$

We want the basis functions to have a total minimal energy among all such functions, namely $\{\phi_i^H\}_{i=1}^J$ is the minimizer of

$$\min \sum_{i=1}^{n_c} \|\psi_i\|_A^2 \quad \text{subject to } \psi_i \in V_i \text{ and } (\psi_1, \dots, \psi_{n_c}) \in \mathcal{X}_\eta. \tag{11.9}$$

Here

$$V_i = \{v \in V_h : \operatorname{supp}(v) \subset \bar{\Omega}_i\}, \quad 1 \leq i \leq n_c. \tag{11.10}$$

Thanks to (11.8), the decomposition (11.11) holds, namely

$$V = \sum_{i=1}^{n_c} V_i. \tag{11.11}$$

Remark 11.3. In AMG, the minimization problem (11.9) written in terms of the prolongation matrices is as follows: find $P \in \mathbb{R}^{n \times n_c}$ such that

$$P = \underset{Y \in \mathbb{R}^{n \times n_c}}{\arg\min}\, \mathcal{F}(Y), \quad Y \mathbf{1}_{n_c} = \mathbf{1}_n, \quad \mathcal{F}(Y) = \operatorname{trace}(Y^T A Y). \tag{11.12}$$

In terms of vectors, local support means that few non-zeros per column (or per row) are allowed in P. We note that the functions $\{\phi_i^H\}$ satisfying the properties mentioned above are linearly independent due to the second assumption in (11.8) and the constraint in (11.12). This linear independence is equivalent to assuming that P is a full rank matrix (*i.e.* $\operatorname{rank}(P) = n_c$).

By the assumption in (11.8), for each j, there exists $k \in \Omega_j$ such that $k \notin \Omega_i$ for all $i \neq j$. We define

$$\mathcal{A}_j = \{k \in \Omega_j : k \notin \Omega_i, i \neq j\}, \quad j = 1 : n_c.$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Then $\mathcal{A}_j \cap \mathcal{A}_i = \emptyset$ if $i \neq j$. We then define

$$\mathcal{C} = \bigcup_{j=1}^{n_c} \mathcal{A}_j \quad \text{and} \quad \mathcal{F} = \Omega \setminus \mathcal{C}.$$

We define $N_j \in V'$ as follows:

$$N_j(v) = \frac{1}{|\mathcal{A}_j|} \sum_{i \in \mathcal{A}_j} \psi_i(v) = \frac{1}{|\mathcal{A}_j|} \sum_{i \in \mathcal{A}_j} v_i, \quad j = 1 : n_c.$$

Clearly $\{N_j\}_{j=1}^{n_c}$ are linearly independent, and if $\operatorname{supp}(v) \subset \Omega_j$, then $N_i(v) = 0$ for all $i \neq j$.

We define $(V')_c \subset V'$ by

$$(V')_c = \operatorname{span}\{N_j : j = 1 : n_c\},$$

and $V_{hf} \subset V$ by

$$V_{hf} = \{v \in V : (g, v) = 0, \ \text{for all } g \in (V')_c\}.$$

If the basis $\{\varphi_j\}_{j=1}^{n_c}$ satisfies

$$\operatorname{supp}(\varphi_j) \subset \Omega_j \quad \text{and} \quad \sum_{j=1}^{n_c} \varphi_j = 1,$$

then we have the Gram matrix

$$G = (G_{ij}) = (N_j(\varphi_i)) = I.$$

By Lemma 9.1, we have

$$V = V_{hf} \oplus W_c, \quad W_c = \operatorname{span}\{\varphi_j : j = 1 : n_c\}.$$

Let us first introduce some notation. We define the restriction A_i of A on each subspace V_i as

$$(A_i u_i, v_i) = (A u_i, v_i), \quad \text{for all } u_i, v_i \in V_i, \ i = 1 : n_c. \tag{11.13}$$

Let $Q_i : V' \mapsto V_i'$ be a projection defined as the adjoint of the natural inclusion $V_i \subset V$:

$$\langle Q_i u', v_i \rangle = \langle u', v_i \rangle \quad \text{for all } v_i \in V_i, \ u' \in V'.$$

We now define the following PSC-type preconditioner (see (4.17)):

$$B = \sum_{i=1}^{n_c} A_i^{-1} \imath_i' = \sum_{i=1}^{n_c} \imath_i A_i^{-1} \imath_i'. \tag{11.14}$$

Thanks to (11.8), it is easy to see that the operator $T : V \mapsto V$ is an isomorphism.

We are now in a position to state and prove the first result in this section.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Theorem 11.4. The minimization problem (11.9) has a unique solution which is given by

$$\phi_i^H = A_i^{-1} Q_i B^{-1} 1, \tag{11.15}$$

satisfying

$$\operatorname{supp}(\phi_i^H) \subset \Omega_i.$$

Proof. This result actually follows directly from Theorem 4.4 with $v = 1$. However, let us give a different proof by finding the critical point of the quadratic functional

$$L = \sum_{i=1}^{n_c} \left(\frac{1}{2} \|\phi_i\|_A^2 - \langle \lambda, \phi_i \rangle \right).$$

Differentiating this functional gives

$$[\partial_{\phi_i} L] \xi_i = (A\phi_i, \xi_i) - \langle \lambda, \xi_i \rangle, \quad \xi_i \in V_i.$$

Hence the the ith component of the critical point (ϕ_i^H) is given by

$$(\phi_i, \xi_i)_A = \langle \lambda, \xi_i \rangle, \quad \text{for all } \xi_i \in V_i, \ i = 1 : n_c. \tag{11.16}$$

From the above equations we obtain that

$$\phi_i^H = A_i^{-1} Q_i \lambda.$$

Summing up leads to

$$\lambda = B^{-1} 1.$$

This gives a derivation of (11.15).

It is obvious that this unique critical point (ϕ_i^H) is indeed the unique global minimizer of (11.9) that has a convex objective functional and a convex constraint. □

We now show that the constructed basis functions are locally discrete A-harmonic. We say that a function $w \in V$ is discrete A-harmonic on a subdomain D if

$$(w, v)_A = 0, \quad \text{for all } v \in V_{h,0}(D) \equiv \{v \in V_h : \operatorname{supp}(v) \subset \bar{D}\}.$$

This property requires us to define the 'subdomains' D on which it holds. Below, we introduce such subdomains in terms of function spaces. Matrix/vector representations of the considerations below are easy to write. To define an analogue of coarse grid elements (an analogue to a finite element coarse grid), we first consider the set of all points in Ω that are interior to all the Ω_i:

$$\omega_0 = \left(\bigcup_{i=1}^{n_c} \partial \Omega_i \right)^c \bigcap \Omega.$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Given $x \in \omega_0$, define the following function with values in the subsets of $\{1, \ldots, n_c\}$, which is the set of indices of subdomains Ω_i that contain x:

$$I(x) = \{i : x \in \Omega_i\}. \tag{11.17}$$

To rule out any ambiguity, we shall assume that for any $x \in \omega_0$ the set $I(x)$ is ordered in ascending order. We then define

$$K_x = \{y \in \omega_0 : I(y) = I(x)\}. \tag{11.18}$$

That is, K_x is the intersection of all the Ω_i that contain x (see Figure 11.1).

The following simple proposition will lead us to an appropriate definition of coarse grid elements.

Proposition 11.5. For the sets K_x defined in (11.18) we have:

(a) $K_x = K_y$ if and only if $I(x) = I(y)$,

(b) either $K_x \cap K_y = \emptyset$ or $K_x = K_y$, $x \in \omega_0$, $y \in \omega_0$,

(c) there are a finite number m_H of different sets K_x, $x \in \omega_0$.

Proof. The ($\Rightarrow$) direction in (a) follows from the fact that $x \in K_x = K_y$, and hence $I(x) = I(y)$. The other direction follows from the definition of $K_{x,y}$.

To prove (b) let us assume that there exists $z \in \omega_0$, such that $z \in K_x$ and $z \in K_y$. The definition of K_x and K_y then gives that $I(x) = I(y) = I(z)$. By (a), $K_x = K_y$. This proves (b).

The conclusion (c) follows directly from (b). □

Let $\mathcal{T}_H$ denote the finite collection of m_H sets from Proposition 11.5(c). We have

$$\omega_0 = \bigcup_{x \in \omega_0} K_x = \bigcup_{K \in \mathcal{T}_H} K.$$

As it is obvious that $\bar{\omega}_0 = \bar{\Omega}$,

$$\bar{\Omega} = \bar{\omega}_0 = \overline{\bigcup_{K \in \mathcal{T}_H} K} = \bigcup_{K \in \mathcal{T}_H} \bar{K}. \tag{11.19}$$

This means that the collection of $\mathcal{T}_H$ forms a non-overlapping partition of Ω. Each element in $\mathcal{T}_H$ will be called a *coarse grid element.*

Remark 11.6. It is tempting to show how these macro-elements look on an unstructured grid, and in Figure 11.1 we have depicted three such supports together with their intersection. However, let us point out that an essential feature of the technique we present here is that the coarse elements need not be defined explicitly, and they might have a fairly complicated shape.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

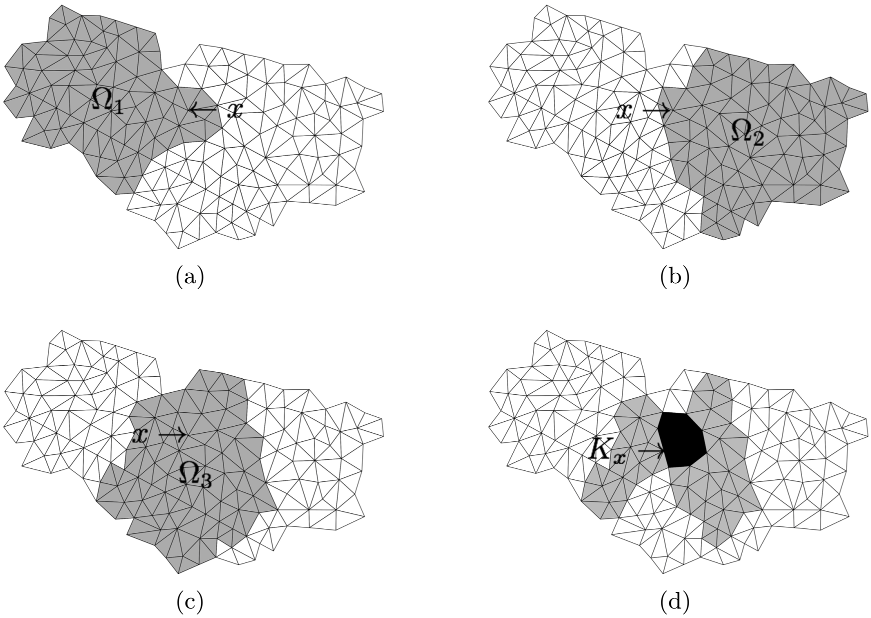


Figure 11.1. (a–c) A triangular grid and the supports of three basis functions. The intersections are plotted in (d): the darker shaded domain corresponds to a coarse element and is the intersection of all three supports; the lighter shaded domains are intersections of two supports; the white area corresponds to no intersection.

Lemma 11.7. Let $\lambda = B^{-1}1$. Assume that for each coarse element $K \in \mathcal{T}_H$ as defined above, we have $(\mathbf{1}, w_K)_A = 0$ for all w_K supported in K. Then

$$\langle \lambda, \xi \rangle = 0, \quad \text{for all } \xi \in V_{h,0}(K).$$

Proof. By definition $K = K_y$ for some $y \in \Omega$. Thus

$$V_{h,0}(K) = \bigcap_{i \in I(y)} V_i \quad \text{and} \quad \sum_{i \in I(y)} \phi_i^H(x) = 1, x \in K.$$

Thus, by (11.16), we have

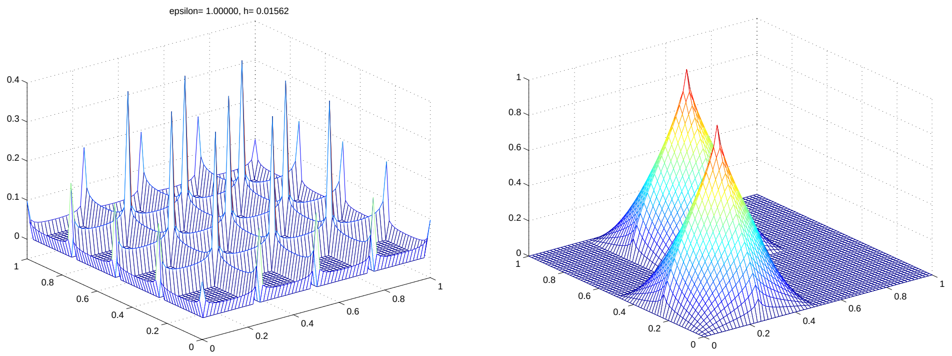
$$(\phi_i, \xi)_A = \langle \lambda, \xi \rangle, \quad \text{for all } \xi \in V_{h,0}(K),$$

and

$$\langle \lambda, \xi \rangle = \sum_{i \in I(y)} (\phi_i, \xi)_A = (\mathbf{1}, \xi)_A = 0.$$

The desired result then follows. □

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press



(a) (b)

Figure 11.2. (a) The profile of $\lambda = B^{-1}\mathbf{1}$, and (b) a typical basis function ϕ_i^H.

When the coarsening corresponds to a geometric multigrid and uniform refinement, the lemma shows that $\lambda = B^{-1}1 \in V'$ is a discrete edge δ-function with respect to the coarse elements (*i.e.* λ is supported around ∂K). Figure 11.2 shows an example profile of λ and a basis function ϕ_i^H.

Combining the above result with the identity (11.16), we immediately obtain our second main result in this section.

Theorem 11.8. Each basis function ϕ_i^H is discrete A-harmonic on each coarse element $K \in \mathcal{T}_H$, namely

$$(\phi_i^H, v)_A = 0, \quad v \in V_{h,0}(K). \tag{11.20}$$

In one space dimension ($d = 1$) the above result is rather trivial, and this has in fact already been discussed by Wan, Chan and Smith (1999/2000). In this case, the basis function (ϕ_i^H) is analogous to the generalized finite element basis function in Babuška and Osborn (1983).

The local harmonic properties in all the aforementioned literature are obtained by *construction* from *local* element boundaries. It is interesting to note that the energy-minimizing basis studied here is the result of a more *global* construction, and the local harmonic properties are a by-product of the construction.

11.3. Convergence of energy-minimization AMG

We present a proof of the two-level convergence of AMG based on energy-minimization.

We define cutoff operators χ_j on the subdomains Ω_j introduced in (11.8) as follows:

$$(\chi_j v)(x) := \begin{cases} v(x) & \text{if } x \in \bar{\Omega}_j, \\ 0 & \text{if } x \notin \bar{\Omega}_j. \end{cases}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Then we define spaces W_j by

$$W_j := \chi_j V = \chi_j v, \quad \text{for } v \in V.$$

It is easy to verify that

$$\{\chi_j \phi_i : \operatorname{supp} \phi_i \cap \Omega_j \neq \emptyset\} \tag{11.21}$$

forms a basis of W_j.

The operator $\Pi_j : W_j \mapsto V$ is defined as in (10.14), with ϕ_j^H being the solution of the minimization problem (11.9) ((11.15) in Theorem 11.4).

We then have that

$$\sum_{j=1}^{n_c} \Pi_j \chi_j = \mathrm{Id}.$$

In fact, we have

$$\sum_{j=1}^{n_c} \phi_j^H(x) = \sum_{j:x\in\Omega_j} \phi_j(x) = \sum_{j\in I(x)} \phi_j(x) = 1,$$

which implies the identity

$$\sum_{j=1}^{n_c} \Pi_j \chi_j v = I_h\biggl(\sum_{j=1}^{n_c} \phi_j^H \chi_j v\biggr) = I_h(v) = v, \text{ and hence } \sum_{j=1}^{n_c} \Pi_j \chi_j = \mathrm{Id}. \tag{11.22}$$

The operator $A_j : W_j \mapsto W_j'$ is the local restriction of the bilinear form $a(\cdot,\cdot)$ as in (10.15).

Inequality (6.10) is satisfied with the decomposition $v = \sum_{j=1}^{n_c} \Pi_j v_j$, where $v_j = \chi_j v \in W_j$:

$$\sum_{j=1}^{m_c} \|v_j\|_{A_j}^2 = \sum_{j=1}^{m_c} a(v,v)_{\Omega_j} \le C_o \|v\|_A^2,$$

where C_o depends on the number of overlaps of Ω_j.

We choose $D : V \mapsto V'$ using the fine grid basis functions as follows:

$$(D\phi_i^h, \phi_j^h) := (A\phi_i^h, \phi_j^h)\delta_{ij}, \quad 1 \le i,j \le n. \tag{11.23}$$

Notice that in the above definition the matrix representation of D is the diagonal of the matrix representation of A. In a similar way, we can define $D_j : W_j \mapsto W_j'$ using basis functions of W_j defined in (11.21), and the matrix representation of D_j is the diagonal of the matrix representation of A_j. It is well known that

$$\|v\|_D^2 \eqsim h^{-2}\|v\|_0^2, \text{ for all } v \in V, \quad \text{and} \quad \|v_j\|_{D_j}^2 \eqsim h^{-2}\|v_j\|_0^2, \text{ for all } v_j \in W_j.$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Inequality (6.9) can be verified by

$$\begin{aligned}\left\|\sum_{j=1}^{n_c}\Pi_j v_j\right\|_D^2 &\lesssim h^{-2}\int_\Omega\left(\sum_{j=1}^{n_c}\phi_j^H v_j\right)^2 = h^{-2}\sum_{i,j=1}^{n_c}\int_\Omega \phi_i^H v_i \phi_j^H v_j\\ &\le h^{-2}\sum_{i,j=1}^{n_c}\|\phi_i\|_\infty\|\phi_j\|_\infty\int_\Omega v_i v_j\\ &\le h^{-2}\Big(\max_{1\le j\le n_c}\|\phi_j\|_\infty\Big)^2\sum_{i,j=1}^{n_c}\int_{\Omega_i\bigcap\Omega_j} v_i v_j\\ &\le C_o\Big(\max_{1\le j\le n_c}\|\phi_j\|_\infty\Big)^2\sum_{j=1}^{n_c}h^{-2}\int_{\Omega_j}|v_j|^2\\ &\lesssim C_o\Big(\max_{1\le j\le n_c}\|\phi_j\|_\infty\Big)^2\sum_{j=1}^{n_c}\|v_j\|_{D_j}^2.\end{aligned}$$

We choose local coarse spaces $W_j^c \subset W_j$ to be the space of constant functions on $\bar{\Omega}_j$. Then the global coarse space V_c is defined as

$$V_c := \sum_{j=1}^{n_c}\Pi_j W_j^c.$$

In fact, for this case, V_c is the subspace spanned by $\{\phi_j^H\}$:

$$V_c = \operatorname{span}\{\phi_j^H : j = 1,\dots,n_c\}. \tag{11.24}$$

We choose the subdomain Ω_j such that the Poincaré inequality holds:

$$\inf_{v_c\in W_j^c}\|v - v_c\|_0^2 \le c d_j^2 |v|_1^2,$$

where d_j is the diameter of Ω_j, and c is a constant independent of the mesh size and the size of Ω_j. Since

$$\|v\|_{D_j}^2 \eqsim h^{-2}\|v\|_0^2, \quad \|v\|_{A_j}^2 \eqsim |v|_1^2,$$

we have

$$\inf_{v_c\in W_j^c}\|v - v_c\|_{D_j}^2 \le c\left(\frac{d_j}{h}\right)^2\|v\|_{A_j}^2,$$

Combining the above discussion with Theorem 6.10, we obtain the following result.

Theorem 11.9. The convergence rate of two-level AMG based on energy-minimization is bounded by

$$\|E\|_A \le 1 - \mu, \tag{11.25}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

where $0 < \mu < 1$ depends only on the size and overlaps of the subdomains Ω_j.

11.4. Bibliographical notes

Energy-minimization seems to encompass several algorithms: the Lagrange equations for this minimization are solved approximately in classical AMG (Section 12), while the functional is approximately minimized in smoothed aggregation (Section 13).

For energy-minimization approaches we refer to the works by Mandel, Brezina and Vaněk (1999), Wan *et al.* (1999/2000), Chan, Xu and Zikatanov (1998), Xu and Zikatanov (2004) and Brannick and Zikatanov (2006). An interesting fact is that for regularly refined grids, given the right supports, the trace (energy) minimization prolongation recovers the coarse basis very closely, although not exactly. The small discrepancies are due to effects from the boundary, and the further in graph distance the coarse grid basis function is, the closer it is to piecewise linear.

The extensive numerical experiments reported in Wan *et al.* (1999/2000), Wan (1998), Mandel *et al.* (1999) and Xu and Zikatanov (2004) show that the energy-minimizing basis leads to uniformly convergent two-grid and multigrid methods for many problems of practical interest, and especially for problems with rough coefficients. These methods also provide a framework for numerical homogenization, and are related to the homogenization methods used by Grauschopf, Griebel and Regler (1997) and the M^3 techniques of Lipnikov, Moulton and Svyatskiy (2011).

The 'smoothed aggregation' approach in the algebraic multigrid method, as well as the theoretical framework, is reported in works by Vaněk, Mandel and Brezina (Vaněk *et al.* 1996, 1998, Mandel *et al.* 1999), who draw an explicit relation between the construction of a basis for the coarse space and the 'energy' of the basis functions. As pointed out by Wan *et al.* (1999/2000) and Mandel *et al.* (1999), this can be viewed as a step (or steps) towards obtaining basis functions minimizing a quadratic (energy) functional, which we considered earlier.

Olson, Schroder and Tuminaro (2011) explore different sparsity patterns, including long-distance energy-minimizing prolongations. Schroder (2012) considers anisotropic problems.

Vassilevski and Zikatanov (2006) consider constrained energy-minimization preserving multiple vectors. In the finite element settings, when the element stiffness matrices are available, local energy-minimization provides coarse spaces which result in uniform two-level convergence (Kolev and Vassilevski 2006).

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

12. Classical AMG

A coarse space in classical AMG is always viewed as a subspace defined via a prolongation (interpolation) matrix. Its dimension, n_c, is a fraction of the dimension of the finer space. Popular interpolation schemes in classical AMG are *direct, standard* or *multipass* interpolation. The matrix representations of such interpolations are of the form $P = \binom{W}{I}$, with $W \in \mathbb{R}^{n_F \times n_C}$, and they can be viewed as *sparse* approximations to the so-called 'ideal' interpolation with $W = -A_{FF}^{-1} A_{FC}$, which is, in general, a full matrix. Here, the matrix A_{FF} is the block of the matrix corresponding to the F-points and A_{FC} is the block corresponding to the connections between the C-points and F-points. The splitting of vertices of the adjacency graph corresponding to A in subsets F and C is done using one of the coarsening strategies described in Section 9.

12.1. Coarse spaces in classical AMG

The coarsening algorithm in classical AMG uses a splitting of the set of vertices $\{1, 2, \dots, n\}$ of the graph corresponding to A into two disjoint sets

$$\mathcal{C} \cup \mathcal{F} = \{1, 2, \dots, n\}, \quad \mathcal{C} \cap \mathcal{F} = \emptyset, \tag{12.1}$$

where $\mathcal{C}$ is a maximal independent set whose independence is with respect to the graph of the strength operator S defined in Section 8.2, and the splitting is referred to as a 'C/F splitting'. A simple greedy C/F splitting algorithm is introduced in Algorithm 5, Section 9.3.1.

We assume that V is equipped with a basis $\{\phi_k\}_{k=1}^n$, and we let $n_f = |\mathcal{F}|$, $n_c = |\mathcal{C}|$ denote the cardinality of the sets forming the C/F splitting. When convenient we assume that

$$\mathcal{F} = \{1, \dots, n_f\} \quad \text{and} \quad \mathcal{C} = \{n_f + 1, \dots, n\}. \tag{12.2}$$

In this case, the high-frequency space V_{hf} as defined in (9.2) can be written as

$$V_{hf} = \operatorname{span}\{\phi_j, j \in \mathcal{F}\}. \tag{12.3}$$

Following the procedure in Section 9, we first proceed to identify a tentative coarse space W_c. One easy choice is

$$W_c = \operatorname{span}\{\phi_j, j \in \mathcal{C}\}. \tag{12.4}$$

Obviously the above tentative space satisfies (9.3), namely

$$V = V_{hf} \oplus W_c,$$

but W_c is hardly a low-frequency space. Given a basis function in W_c, ϕ_{j_k}, we filter out its high-frequency component to obtain the following coarse

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

basis function:

$$\phi_{k,c} := (I - \Pi_{hf})\phi_{j_k} = \phi_{j_k} + \sum_{j \notin \mathcal{C}} p_{jk}\phi_j. \tag{12.5}$$

Here P_{hf} is the $(\cdot,\cdot)_A$ orthogonal projection onto V_{hf}. By the definition of Π_{hf},

$$\Pi_{hf}\phi_k = \begin{cases} \phi_k & \text{if } k \in \mathcal{F}, \\ -\sum_{j\in\mathcal{F}} p_{jk}\phi_j & \text{if } k \in \mathcal{C}, \end{cases}$$

where p_{jk} satisfies

$$-\sum_{j\in\mathcal{F}} p_{jk}(\phi_j, \phi_i)_A = (\phi_k, \phi_i)_A, \quad \text{for all } i \in \mathcal{F},\ k \in \mathcal{C}. \tag{12.6}$$

In matrix notation, with the ordering given by (12.2), the matrix form of the equations in (12.6) is

$$A_{FF}W = A_{FC}, \quad \text{where } A = \begin{pmatrix} A_{FF} & A_{FC} \\ A_{FC}^T & A_{CC} \end{pmatrix}. \tag{12.7}$$

Here, $W_{jk} = p_{jk}$, $j \in \mathcal{F}$, $k \in \mathcal{C}$, where p_{jk} are the coefficients given in (12.6).

In view of Lemma 9.3, we have $S = I - \Pi_{hf}$. V_c is the span of the functions in (12.5):

$$V_c = \operatorname{span}\{\phi_{k,c}\}_{k=1}^{n_c} = \operatorname{Range}(I - P_{hf}).$$

We note that

$$(\phi_{1,c}, \dots, \phi_{n_c,c}) = (\phi_1, \dots, \phi_n)\begin{pmatrix} W \\ I \end{pmatrix}.$$

Thus the corresponding prolongation matrix is

$$P = \begin{pmatrix} W \\ I \end{pmatrix}, \quad W = -A_{FF}^{-1}A_{FC}. \tag{12.8}$$

The functions $\{\phi_{k,H}\}$ given in (12.5) form a basis of V_c similar to the geometric multigrid method for Lagrangian finite elements. We denote the prolongation matrix defined in (12.8) by P^{opt}, and refer to it below as the *ideal interpolation.*

The following result can be easily established.

Lemma 12.1. If A satisfies $A\mathbf{1} = 0$, the solution of (12.7) also satisfies the constraint in (11.12), namely, $W\mathbf{1}_{n_c} = \mathbf{1}_{n_F}$.

Next, we introduce the set of the prolongations $\mathcal{P}_C$,

$$\mathcal{P}_C = \left\{P : P = \begin{pmatrix} W \\ I \end{pmatrix}\right\}.$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

and from the definition given in (5.25) in Section 11 we recall that

$$\mathcal{P}_C \subset \eta\mathcal{X},$$

for some constant $\eta > 0$. For example, $\eta \eqsim \max_{1\le j\le n}|D_{jj}|$ if $\bar{R}^{-1} \eqsim D$, where D is the diagonal of A. Equivalently, $P \in \mathcal{P}_C$ means that the coarse grid basis is defined as

$$\phi_{k,H} = \phi_{j_k} + v_f, \quad j_k \in \mathcal{C}, \quad v_f \in V_f, \quad k = 1,\ldots,n_c.$$

Clearly, all elements of $\mathcal{P}_C$ are of full rank, and we also have that $P^{\text{opt}} \in \mathcal{P}_C$ by definition.

We have the following theorem, showing that the coarse grid matrix corresponding to $V_c^{\text{opt}} = \text{Range}(P^{\text{opt}})$ has a minimal trace.

Theorem 12.2. If we fix the set of indexes $\mathcal{C}$ and coarse grid degrees of freedom, then for P^{opt} we have

$$P^{\text{opt}} = \arg\min_{P\in\mathcal{P}_C} \text{trace}(P^T AP). \tag{12.9}$$

Furthermore, if we denote $A_c = (P^{\text{opt}})^T AP^{\text{opt}}$, then

$$\|v_c\|^2_{A_c} = \min\big\{\|v\|^2_A : \langle\phi'_{j_k}, v\rangle = v_{c,k},\ k = 1,\ldots,n_c\big\}. \tag{12.10}$$

Proof. The relation (12.9) follows from the simple identities

$$\begin{aligned}
\text{trace}(A_c) &= \sum_{k=1}^{n_c} \|\phi_{k,H}\|^2_A = \sum_{k=1}^{n_c} \|(I - P_{hf})\phi_{j_k}\|^2_A \\
&= \sum_{k=1}^{n_c} \min_{v_k\in V_{hf}} \|\phi_{j_k} + v_k\|^2_A \\
&= \min\left\{\sum_{k=1}^{n_c} \|\phi_{j_k} + v_k\|^2_A : v_k \in V_{hf},\ k = 1,\ldots,n_c\right\} \\
&= \min_{P\in\mathcal{P}_C} \text{trace}(P^T AP).
\end{aligned}$$

To prove (12.10), we note that any $v \in V$ such that $\langle\phi'_{j_k}, v\rangle = v_{c,k}$ can be written as

$$v = w_c + v_f, \quad \text{where } w_c = \sum_{k=1}^{n_c} v_{c,k}\phi_{j_k}.$$

By the definition of P_{hf}, we have $(I-P_{hf})v_f = 0$ and $v_c = (I-P_{hf})(v-v_f) = (I - P_{hf})v$. We then have

$$\|v\|^2_A = \|P_{hf}v\|^2_A + \|(I - P_{hf})v\|^2_A = \|P_{hf}v\|^2_A + \|v_c\|^2_{A_c} \ge \|v_c\|^2_{A_c}. \qquad \square$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Theorem 12.2 shows that the minimizer P satisfies equation (12.7). By Theorem 12.2 it follows that the minimizer of (11.12) and the solution to (12.7) are the same for $A\mathbf{1} = 0$.

12.2. Quasi-optimality of the ideal interpolation

The following two-level convergence result is well known: see *e.g.* MacLachlan, Manteuffel and McCormick (2006).

Theorem 12.3. For P^{opt} defined as the solution to the minimization problem (11.12), or equivalently the solution to (12.7), the two-level AMG method with prolongation P^{opt} converges with a rate $\|E\|_A \le 1-\delta$, where δ is a constant depending only on the maximum degree of the vertices in the graph corresponding to A and the threshold θ in choosing the strong connections.

Proof. According to Corollary 5.21, we only need to verify that V_{hf} consists of δ-algebraic high frequencies as defined in Definition 5.17. Clearly, from the discussion in Section 7.2, we can assume that A is an M-matrix with all connections being strong connections. We consider the graph corresponding to A and recall that the set of coarse grid degrees of freedom is a maximal independent set of vertices in this graph. By definition of connection strength, for any $j \in \{1,\dots,n\}$ and any $i \in N(j)$ we have

$$a_{jj} = \sum_{k\in N_j} -a_{jk} \le |N_j| \max_{k\in N_j}(-a_{jk}) \le -\frac{|N_j|}{\theta} a_{ji}.$$

Next, for any $j \in F$, let $k_j \in \mathcal{C}$ be such that $|a_{k_j,j}| > 0$. Such a k_j exists because the set of C-vertices is a maximal independent set. This choice is not unique, and we just fix one such index k_j for every j. Using the notation from Section 9, and the fact that $v \in V_{hf}$ vanishes at the C-vertices, we obtain

$$\begin{aligned}
\|v\|^2_{\bar{R}^{-1}} &\le c^D\|v\|_D^2 = c^D \sum_{j\in F} a_{jj}v_j^2 = c^D \sum_{j\in F} a_{jj}(v_j - v_{k_j})^2 \\
&\le c^D \sum_{j\in F} -\frac{|N_j|}{\theta} a_{j,k_j}(v_j - v_{k_j})^2 \\
&\le \max_j(|N_j|)\frac{c^D}{\theta} \sum_{(i,j)\in\mathcal{E}} -a_{ij}(v_i - v_j)^2 \\
&\le \max_j(|N_j|)\frac{c^D}{\theta}\|v\|_A^2. \qquad \square
\end{aligned}$$

We note that interpolations like P^{opt} (with the possible exception of one-dimensional problems) are not used in practice, because the prolongation

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

matrix $P^{\rm opt}$ could have a lot of fill-in and $(P^{\rm opt})^T AP^{\rm opt}$ is dense. However, it is also important to note that sparse approximations to $P^{\rm opt}$ are used in practice. Thus the energy-minimization technique for constructing coarse spaces may be viewed as a motivation for other AMG techniques used to construct approximations of $V_c^{\rm opt}$. For example, most of the known techniques approximate the minimizer of the functional $\mathcal{F}$ given in equation (11.12): (i) the prolongation matrices constructed in the classical AMG framework approximate the solution to equation (12.7) over the subset of $\mathcal{P}_C$ consisting of sparse matrices, and (ii) the energy-minimization techniques outlined in Section 11.2 and the smoothed aggregation considered in Section 13.3 minimize (approximately) the trace of the coarse grid matrix $\mathcal{F}(P)$ over a set of sparse matrices P.

12.3. Construction of prolongation matrix P

12.3.1. Prolongation

An intuitive idea for finding an approximate solution of the problem (12.7) is to use some basic iterative methods such as the Jacobi method, and then properly rescale the coefficients of W so that it satisfies

$$W\mathbf{1}_{n_c} = \mathbf{1}_{n_f} \tag{12.11}$$

and also fits into a sparsity pattern:

$$W \in \mathbb{R}_S^{n_f\times n_c}. \tag{12.12}$$

Following this idea, we construct the interpolations used in classical AMG (Trottenberg *et al.* 2001, Section A.7) – direct interpolation, standard interpolation and multipass interpolation – in a unified fashion.

Direct interpolation. Here we approximate the solution to (12.7)–(12.12) by one Jacobi iteration with initial guess $W_0 = 0$, namely,

$$W_1 = -D_{FF}^{-1}A_{FC}.$$

In order to satisfy the constraint (12.11), we rescale W_1 to obtain that

$$W = M(-D_{FF}^{-1}A_{FC}) = [\mathrm{diag}(A_{FC}\mathbf{1})]^{-1}A_{FC},$$

where M is a rescaling operator defined by

$$M(Y) = [\mathrm{diag}(Y\mathbf{1})]^{-1}Y. \tag{12.13}$$

Standard interpolation. The construction of the prolongation matrix via standard interpolation can be viewed in several different ways. Probably the most natural way is to view it as a smoothing of the direct interpolation, again followed by rescaling. Indeed, assuming that the smooth error (e_F^T, e_C^T) satisfies

$$A_{FF}e_F + A_{FC}e_C = 0,$$

we approximate this equation via

$$e_F = -D_{FF}^{-1}A_{FC}e_C + D_F^{-1}A_{FF}(D_{FF}^{-1}A_{FC})e_C \tag{12.14}$$

$$W = (I - D_{FF}^{-1}A_{FF})W^1, \quad W^1 = -D_{FF}^{-1}A_{FC}, \tag{12.15}$$

using the same notation e_F, e_C for the approximations (see Trottenberg *et al.* 2001). This is equivalent to a Jacobi smoothing iteration applied to the unscaled direct interpolation W^1. Rescaling is also needed for the standard interpolation, and the final formula is

$$W = [\mathrm{diag}((I - D_{FF}^{-1}A_{FF})W^1\mathbf{1})]^{-1}(I - D_{FF}^{-1}A_{FF})W^1. \tag{12.16}$$

The similarity to the smoothed aggregation discussed in Section 13.3 is obvious, as this is indeed a smoothing applied to W^1.

Multipass interpolation. This is an approximation to the solution of (12.7) when the C/F splitting has been constructed by means of aggressive coarsening, namely, using the (m,l)-strong connection defined in Definition 9.7. The set of F is divided into l disjoint subsets $F_1, F_2, \dots, F_l$ as follows. We first define the distance from a point j to a subset of points C:

$$\mathrm{dist}(j, C) = \min\{\mathrm{dist}(j, i) : i \in C\}.$$

Next, we set

$$F_k = \{j \in \mathcal{F} : \mathrm{dist}(j, \mathcal{C}) = k\}, \quad k = 1, 2, \dots, l, \quad l = \max\{\mathrm{dist}(j, \mathcal{C}) : j \in F\}.$$

Then A_{FF}, A_{FC} can be written as the following block matrices:

$$A_{FF} = \begin{pmatrix} A_{F_1F_1} & A_{F_1F_2} & \cdots & A_{F_1F_l} \\ A_{F_2F_1} & A_{F_2F_2} & \cdots & A_{F_2F_l} \\ \vdots & \vdots & \ddots & \vdots \\ A_{F_lF_1} & A_{F_lF_2} & \cdots & A_{F_lF_l} \end{pmatrix}, \quad A_{FC} = \begin{pmatrix} A_{F_1C} \\ A_{F_2C} \\ \vdots \\ A_{F_lC} \end{pmatrix}. \tag{12.17}$$

We can also write W block-wise as $W^T = (W_{F_1}^T, W_{F_2}^T, \dots, W_{F_l}^T)$. Then (12.7) takes the form

$$\begin{cases} A_{F_1F_1}W_{F_1} + A_{F_1F_2}W_{F_2} + \cdots + A_{F_1F_l}W_{F_l} + A_{F_1C} = 0, \\ A_{F_2F_1}W_{F_1} + A_{F_2F_2}W_{F_2} + \cdots + A_{F_2F_l}W_{F_l} + A_{F_2C} = 0, \\ \vdots \\ A_{F_lF_1}W_{F_1} + A_{F_lF_2}W_{F_2} + \cdots + A_{F_lF_l}W_{F_l} + A_{F_lC} = 0. \end{cases} \tag{12.18}$$

To define the entries of W_{F_k} via multipass interpolation, we use the following steps.

1 For $k = 1$, use direct interpolation to approximate W_{F_1}. More precisely, we write the first equation in (12.18) as

$$A_{F_1F_1}W_{F_1} + \hat{A}_{F_1C} = 0,$$

with $\hat{A}_{F_1C} = A_{F_1F_2}W_{F_2} + \cdots + A_{F_1F_l}W_{F_l} + A_{F_1C}$. Then we apply direct interpolation and write W_{F_1} as a function of W_{F_j}, $j > 1$.

2 While $k < l$:

(a) Write the $(k+1)$st equation in (12.18) substituting the expressions for W_{F_m}, $m < (k+1)$ obtained from the previous steps. The $(k+1)$th equation then has the form

$$\hat{A}_{F_{k+1}F_{k+1}}W_{F_{k+1}} + \hat{A}_{F_{k+1}F_{k+2}}W_{F_{k+2}} + \cdots \hat{A}_{F_{k+1}F_l}W_{F_l} + \hat{A}_{F_{k+1}C} = 0. \tag{12.19}$$

(b) Apply direct interpolation to (12.19) to write $W_{F_{k+1}}$ as a function of W_{F_j}, $j > (k+1)$.

(c) Set $k \leftarrow k+1$.

From the derivation and the definitions above, we have the following set inclusions describing the sparsity patterns of the prolongations defined earlier:

$$\mathrm{S}(P) \subset \mathrm{S}\begin{pmatrix} A_{FC} \\ I \end{pmatrix} \qquad \text{(direct interpolation (12.3.1))},$$

$$\mathrm{S}(P) = \mathrm{S}\begin{pmatrix} \widehat{A}_{FC} \\ I \end{pmatrix} \qquad \text{(standard interpolation (12.16))},$$

$$\mathrm{S}(P) = \mathrm{S}\begin{pmatrix} \widehat{A}_{F_1C} \\ \widehat{A}_{F_2C} \\ \vdots \\ \widehat{A}_{F_lC} \\ I \end{pmatrix} \qquad \text{(multipass interpolation).}$$

12.3.2. Interpolation preserving a given vector

From the above discussions, it is important that the prolongation P preserves some vectors which can represent algebraic smooth errors. Both direct and standard interpolation use a diagonal scaling to make sure that the prolongation preserves constant vectors, which is the kernel of scalar elliptic operators. To generalize this idea, here we introduce αAMG interpolation, which constructs the prolongation matrix by choosing an initial guess that preserves the near-null component $v^{(1)}$, which may not necessarily be constant vectors.

We construct $P \in \mathcal{P}_C$ such that it 'preserves' a given vector v, namely,

$$v_F = Wv_C.$$

To do this, we first pick an initial guess W^0 for W (or P),

$$W^0 = D_v^{-1}A_{FC}, \tag{12.20}$$

where D_v is a diagonal matrix such that the following identity holds:

$$v_F = D_v^{-1} A_{FC} v_C.$$

It is easy to derive that the explicit formula for diagonal entries of D_v is

$$d_{kk} = \frac{\sum_{j\in C} a_{kj} v_j^{(1)}}{v_k^{(1)}}.$$

Then the W in this 'vector-preserving' interpolation is obtained by applying one Jacobi iteration for the linear problem (12.7) with initial guess W^0:

$$W = D^{-1} A_{FC} + D_{FF}^{-1}(-A_{FC} - A_{FF} D^{-1} A_{FC}).$$

A fully detailed description of the construction of a prototype vector v and coarse space interpolating this vector exactly, using αAMG (the classical AMG version of adaptive AMG), can be found in Brezina *et al.* (2006). The ideas, however, were outlined earlier in Brandt *et al.* (1982).

12.4. Classical AMG within the abstract AMG framework

Classical AMG falls within the abstract theory developed in Sections 5 and 6. To do this, we first consider an *M-matrix relative* of A using the adjacency graph corresponding to a strength function. We then use a maximal independent set algorithm to identify C, the set of coarse points, to form a C/F splitting. We further split the set of indices $\Omega = \{1, 2, \ldots, n\}$ into subsets $\Omega_1, \Omega_2, \ldots, \Omega_J$ so that

$$\Omega = \bigcup_{j=1}^{J} \Omega_j. \tag{12.21}$$

Then, for each $j \in C$, we define

$$\Omega_j := \{j\} \bigcup F_j^s, \quad j = 1, \ldots, J. \tag{12.22}$$

where $F_j^s := \mathcal{F} \bigcap s_j$ and s_j is the set of strong neighbours of j. This depends on the definition of connection strength. For example, in the direct interpolation we introduced in the previous section, we simply use the strength connection defined in (8.8); in the standard interpolation we use the (m, l)-strong connection defined in Definition 9.7, with $m = 1$ and $l = 2$.

For each Ω_j we let

$$\Omega_j = \{m_1, m_2, \ldots, m_{n_j}\}, \tag{12.23}$$

and let $n_j := |\Omega_j|$, namely, n_j is the cardinality of Ω_j. In accordance with the notation in Section 6, we then define

$$V_j := \mathbb{R}^{n_j}, \tag{12.24}$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

and the associated operator $\Pi_j : V_j \mapsto V$,

$$(\Pi_j v)_i = \begin{cases} p_{m_k,k} v_k & \text{if } i = m_k, \\ 0 & \text{if } i \notin \Omega_j, \end{cases} \tag{12.25}$$

where $p_{m_k,k}$ are given weights. As all the constructions below will be based on the M-matrix relative of A, without loss of generality, we use A to denote the M-matrix relative.

Following Section 6, we introduce the operator $\chi_j : V \mapsto V_j$,

$$(\chi_j v)_i := v_{m_i}, \tag{12.26}$$

which takes as argument a vector v and returns only the portion of it with indices in Ω_j. That is, $\chi_j v$ is a vector in $\mathbb{R}^{n_j}$. It is immediate to verify that

$$\sum_{j=1}^{J} \Pi_j \chi_j = I.$$

To estimate the convergence rate using the theory in Section 5, we need to verify all the items in Assumptions 6.4. To do this, we choose a decomposition $v = \sum_{j=1}^{J} \Pi_j v_j$ with $v_j = \chi_j v$. We further define $C_{p,2}$ as a constant depending on the overlaps in the partition $\{\Omega_j\}_{j=1}^{J}$,

$$C_o = \max_{1 \le j \le J} |\{l : \Omega_l \cap \Omega_j \neq \emptyset\}|. \tag{12.27}$$

The local operators A_j on V_j are defined as

$$(A_j u, v) = \sum_{\substack{e \in \mathcal{E} \\ e \subset \Omega_j}} \omega_e \delta_{j,e} u \delta_{j,e} v. \tag{12.28}$$

Here, $e \subset \Omega_j$ means the two vertices connected by e are in Ω_j, and $\delta_{j,e} u = u_{m_k - m_l}$ for $e = (k, l)$ and ω_e are the weights determined by the off-diagonal elements in the M-matrix relative of A. Note that A_j is a symmetric positive semi-definite matrix because all the weights ω_e are non-negative. Then (6.10) easily verified:

$$\begin{aligned} \sum_{j=1}^{m_c} \|\chi_j v\|_{A_j}^2 &= \sum_{j=1}^{m_c} \sum_{\substack{e \in \mathcal{E} \\ e \subset \Omega_j}} \omega_e (\delta_e v)^2 \le C_o \sum_{e \in \mathcal{E}} \omega_e (\delta_e v)^2 \\ &= C_o \|v\|_{A_M}^2 \le C_o C_M \|v\|_A^2. \end{aligned} \tag{12.29}$$

If D is the diagonal of A, then we set D_j, $j = 1 : J$ to be the restriction of D on Ω_j, namely, in $\mathbb{R}^{n_j \times n_j}$ and

$$(D_j)_{ii} = D_{m_i,m_i}, \quad \text{or equivalently } D_j = \chi_j D_j \chi_j' \tag{12.30}$$

We have the following lemma, which shows (6.9).

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Lemma 12.4. For D_j defined in (12.30), the following inequality holds:

$$\left\| \sum_{j=1}^{m_c} \Pi_j w_j \right\|_D^2 \le C_o \sum_{j=1}^{m_c} \|w_j\|_{D_j}^2, \quad \text{for all } w_j \in V_j. \tag{12.31}$$

Proof. Recall from the definition of Π_j that we have

$$\|\Pi_j v\|_D \le \|v\|_{D_j}, \quad \text{for all } v \in V_j. \tag{12.32}$$

Therefore,

$$\begin{aligned}\left\| \sum_{j=1}^{J} \Pi_j v_j \right\|_D^2 &= \left(D \sum_{i=1}^{J} \Pi_i v_i, \sum_{j=1}^{J} \Pi_j v_j \right) = \sum_{i=1}^{J}\sum_{j=1}^{J} (D\Pi_i v_i, \Pi_j v_j) \\ &= \sum_{\substack{1\le i,j\le J \\ \Omega_i\cap\Omega_j\ne\emptyset}} (D\Pi_i v_i, \Pi_j v_j) \le \sum_{\substack{1\le i,j\le J \\ \Omega_i\cap\Omega_j\ne\emptyset}} \frac{\|\Pi_j v_i\|_D^2 + \|\Pi_j v_j\|_D^2}{2} \\ &\le C_o \sum_{j=1}^{J} \|v_j\|_{D_j}^2. \end{aligned}$$

□

We choose the local coarse spaces V_j^c as

$$V_j^c := \operatorname{span}\{\mathbf{1}_{n_j}\}. \tag{12.33}$$

Then, by definition, we have

$$\mu_j(V_j^c) = \frac{1}{\lambda_j^{(2)}}, \tag{12.34}$$

where $\lambda_j^{(2)}$ is the second smallest eigenvalue of the matrix $D_j^{-1}A_j$. The global coarse space V_c is then obtained by (6.12), and is

$$V_c = \operatorname{span}\{P_1, P_2, \dots, P_J\}. \tag{12.35}$$

Finally, by Theorem 5.3, the convergence rate of this two-level geometric multigrid method depends on the $\min_j(\lambda_j^{(2)})$. If the discrete Poincaré inequality is true for each V_j, namely,

$$\inf_{v_c\in V_j^c} \|v - v_c\|_{D_j}^2 \le c_j \|v\|_{A_j}^2, \quad \text{for all } v \in V_j, \tag{12.36}$$

with c_j a constant, then the two-level classical AMG method converges uniformly.

12.5. Bibliographical notes

The classical coarse space definition in AMG was introduced by Brandt *et al.* (1982), and then somewhat improved by Stüben (1983), Brandt *et al.* (1985) and Ruge and Stüben (1987).

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Despite its great success in practical applications, classical AMG algorithms still lack solid theoretical justifications beyond the theory for two-level methods. It is important to note that a multigrid method that converges uniformly in the two-level case with an exact coarse grid solver may not converge uniformly in the multilevel case: see Brandt (1986) and Ruge and Stüben (1987). For classical AMG, the early theoretical studies of convergence date back to the 1980s: see Mandel (1988), McCormick (1982), Brandt (1986) and Ruge and Stüben (1987). A survey of Stüben's results can be found in the monograph by Trottenberg *et al.* (2001), who also gives the three classical prolongation constructions (direct, standard and multipass). In all cases, it is crucial to define coarsening and interpolation operators so that the interpolation error is uniformly bounded. The role of the ideal interpolation as the minimizer of an upper bound for the convergence rate was emphasized by Falgout and Vassilevski (2004).

13. Aggregation-based AMG

Aggregation (or agglomeration) refers to an algorithm that splits the set of vertices of the graph of the filtered matrix as a union of non-overlapping subsets (aggregates) (each of which forms a connected sub-graph):

$$\{1,\ldots,n\} = \bigcup_{j=1}^{J} \mathcal{A}_j, \quad \mathcal{A}_i \bigcap \mathcal{A}_j = \emptyset,\ i \neq j. \tag{13.1}$$

Such a partition can be obtained using the algorithms described in Section 9.

If we are solving a finite element system, the partition (13.1) would correspond to a non-overlapping decomposition of Ω,

$$\Omega = \bigcup_{j=1}^{J} \Omega_j, \quad \Omega_i \bigcap \Omega_j = \emptyset,\ i \neq j, \tag{13.2}$$

such that $\mathcal{A}_j$ contains the indices associated with the enumerations of the vertices in the subdomain Ω_j.

We denote the elements in $\mathcal{A}_j$ by

$$\mathcal{A}_j = \{m_1, m_2, \ldots, m_{n_j}\}, \tag{13.3}$$

and let $n_j := |\mathcal{A}_j|$, that is, n_j is the number of elements in $\mathcal{A}_j$.

13.1. Unsmoothed aggregation: preserving one kernel vector

Using the framework introduced in Section 6, we define

$$V_j := \mathbb{R}^{n_j},$$

and the associated operator $\Pi_j : \mathbb{R}^{n_j} \mapsto \mathbb{R}^n$ is the trivial extension of $v \in \mathbb{R}^{n_j}$:

$$(\Pi_j v) = \begin{cases} v_k & i = m_k, \\ 0 & i \notin \mathcal{A}_j, \end{cases} \tag{13.4}$$

$$P = (p_1, p_2, \dots, p_J), \quad p_j = \Pi_j \mathbf{1}_{n_j}. \tag{13.5}$$

This prolongation matrix obviously satisfies

$$P\mathbf{1}_J = \sum_{j-1}^{J} P_j = \mathbf{1}_n. \tag{13.6}$$

The local coarse space V_j^c is

$$V_j^c := \operatorname{span}\{\mathbf{1}_{n_j}\}.$$

Then the global coarse space V_c is obtained by (6.12). In fact

$$V_c = \operatorname{span}\{p_1, p_2, \dots, p_J\}.$$

We note that

$$(\phi_{1,c}, \dots, \phi_{J,c}) = (\phi_1, \dots, \phi_n)P, \tag{13.7}$$

where

$$\phi_{j,c} = \sum_{k \in \mathcal{A}_j} \phi_k, \quad j = 1, 2, \dots, J. \tag{13.8}$$

Further, Π_j corresponds to matrix representation of the operator $I_h(\phi_j^H \cdot)$ with the coarse grid basis $\phi_{j,c}$ defined above. In view of Section 10.3 (see especially (10.14)), aggregation-based AMG can also be viewed as a GMG method.

The above procedure gives a full description of the unsmoothed aggregation AMG method. We can use the framework in Section 6 to carry out a two-level convergence analysis. The local matrices A_j are defined in exactly the same way as in the case of classical AMG in Section 12.4. Thus we write A as in (12.4) and then define A_j by (12.28). The matrices D_j are defined as the restriction of the diagonal of A to Ω_j, as in (12.30). Using the same argument as in Section 12.4, Assumption 6.4 is satisfied by (12.29) and Lemma 12.4. Theorem 5.3 can then be applied to prove that the two-level unsmoothed aggregation method has a convergence rate depending only on the local Poincaré constants in (12.36).

13.2. Unsmoothed aggregation: preserving multiple vectors

One great advantage of the aggregation AMG method is that it can be easily extended to the case when the stiffness matrix A has multiple kernel or near-kernel vectors.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

To give an illustration of how to construct prolongation preserving more than one vector, we consider two-dimensional elasticity problems. In this case we have three functions, namely the rigid body motion, to preserve

$$u_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad u_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad u_3 = \begin{pmatrix} -y \\ x \end{pmatrix}.$$

The corresponding vectors are

$$\zeta_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 1 \\ 0 \end{pmatrix}, \quad \zeta_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad \text{and} \quad \zeta_3 = \begin{pmatrix} -y_1 \\ x_1 \\ -y_2 \\ x_2 \\ \vdots \\ -y_n \\ x_n \end{pmatrix} \in \mathbb{R}^{2n}.$$

On each aggregate $\mathcal{A}_j$, we consider

$$\zeta_1^{(j)} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 1 \\ 0 \end{pmatrix}, \quad \zeta_2^{(j)} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad \text{and} \quad \zeta_3^{(j)} = \begin{pmatrix} -y_{m_1} \\ x_{m_1} \\ -y_{m_2} \\ x_{m_2} \\ \vdots \\ -y_{m_{n_j}} \\ x_{m_{n_j}} \end{pmatrix} \in \mathbb{R}^{2n_j}.$$

The prolongation matrix is then given by

$$P = (p_1, p_2, \dots, p_J) \in \mathbb{R}^{(2n)\times(3J)}, \quad P_j = \Pi_j(\zeta_1^{(j)}, \zeta_2^{(j)}, \zeta_3^{(j)}) \in \mathbb{R}^{(2n)\times 3}.$$

This prolongation matrix satisfies

$$P(\mathbf{1}_J \otimes e_j) = \zeta_j, 1 \leq j \leq 3. \tag{13.9}$$

The rest of the AMG algorithm based on this prolongation matrix is similar to the 1-vector case.

Extension from the 3-vector case, as mentioned above, to arbitrary m-vectors is straightforward. We let $m = \dim(N(A))$, and consider the case when $m \geq 1$. We now assume that we are given m vectors $\{\zeta_j\}_{j=1}^m$ which are linearly independent and $m \ll n$, and we can then proceed to construct a prolongation P such that $P(\mathbf{1}_{n_c} \otimes e_j) = \zeta_j$, $j = 1 : m$.

We would like to point out that for the multiple kernel or near-kernel vector case we often need some geometric information to describe the corresponding vectors. In the finite element case, these vectors should be obtained by taking the canonical interpolation of the corresponding kernel functions of the underlying partial differential operators, and we then split

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

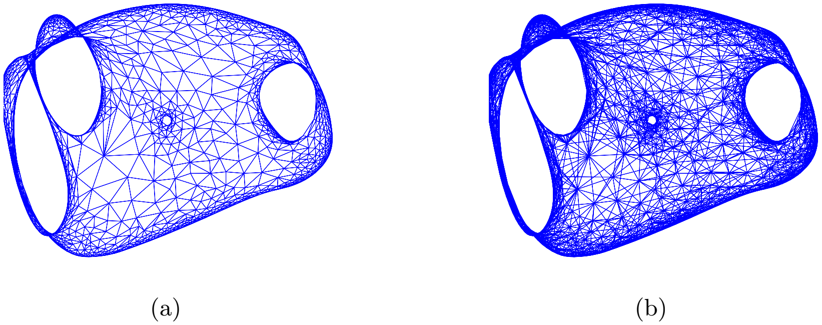


Figure 13.1. Graph of the coarse grid matrix corresponding to the unsmoothed aggregation (a) and the 'denser' graph for the coarse grid matrix obtained by smoothed aggregation (b).

these vectors into different aggregates by using the local degrees of freedom associated with different aggregates.

13.3. Smoothed aggregation

The aggregation procedure described above provides a simple yet efficient AMG method. The resulting method is known as the unsmoothed aggregation AMG (UA-AMG) method. This terminology may be justified by examining the shape of the basis function as defined by (13.8). A typical basis of this form is not smooth. There is a procedure to smooth out these basis function by using a smoother, which is equivalent to a few applications of smoothing on the prolongation matrix (13.5) as follows:

$$P_S = (I - R_s A)^\nu P, \quad \text{for some } \nu \geq 1. \tag{13.10}$$

A typical choice is the scaled Jacobi smoother with $R_S = \omega D^{-1}$. We note that, in view of (13.9), if ζ_j is in the kernel of A, we still have

$$P_S(\mathbf{1}_J \otimes e_j) = \zeta_j, 1 \leq j \leq m.$$

Conceivably, a bigger ν in (13.10) would lead to an AMG algorithm that has a better convergence rate. However, a bigger ν in (13.10) also means a denser P_S and hence a denser A_c, and a more expensive setup for the resulting AMG algorithm. As an example, Figure 13.1 shows the graphs of the coarse grid matrices corresponding to unsmoothed and smoothed aggregation. Clearly the smoothed aggregation graph is denser than the unsmoothed one.

In view of Lemma 9.3, the basis function (13.8) and the resulting UA-AMG corresponds to the use of a tentative coarse space W_c. The smoothed basis functions by means of (13.10) and resulting SA multigrid corresponds to the use of the V_s in Lemma 9.3 with $S = (I - R_s A)^\nu$.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

13.4. Bibliographical notes

One of the first aggregation algorithms comes from applications in economics, and is due to Vakhutinsky *et al.* (1979). Subsequently, aggregation methods were developed and used by Blaheta (1986) to solve discrete elliptic problems, and by Marek (1991) for the calculation of stationary points of Markov chains.

A special class of aggregation coarsening methods is based on matching in graphs (also known as pairwise aggregation), and such methods were employed in Kim *et al.* (2003) and for non-symmetric problems in Kim, Xu and Zikatanov (2004). Aggregations using matching in graphs were further used by Notay (2010), Brannick, Chen and Zikatanov (2012), Livne and Brandt (2012) and D'Ambra and Vassilevski (2013, 2014). To improve the scalability of the unsmoothed aggregation, a number of algorithms have been developed: Kim *et al.* (2003) used a variable V-cycle, Notay and Vassilevski (2008) proposed non-linear Krylov subspace acceleration, and Olson *et al.* (2011) designed a procedure for correcting the energy of the coarse-level Galerkin operator.

Aggregation, especially the unsmoothed aggregation, can be used in conjunction with non-linear (variable-step/flexible) preconditioning methods to result in an optimal algorithm. Such non-linear methods are called *algebraic multilevel iteration methods*, and were introduced by Axelsson and Vassilevski (1991). Non-linear multilevel preconditioners were proposed and an additive version of them was first analysed by Axelsson and Vassilevski (1994); see also Golub and Ye (1999/2000), Notay (2000), Saad (2003) and Kraus (2002). In these non-linear multilevel preconditioners, n steps of a preconditioned CG iterative method provide a polynomial approximation of the inverse of the coarse grid matrix. The same idea can be used to define the MG cycles, as shown by Vassilevski (2008) and Notay and Vassilevski (2008). A comprehensive convergence analysis of the non-linear AMLI-cycle multigrid method for symmetric positive definite problems was conducted by Hu, Vassilevski and Xu (2013). Based on classical assumptions for approximation and smoothing properties, the non-linear AMLI-cycle MG method is shown to be uniformly convergent.

The smoothed aggregation AMG method, first developed by Míka and Vaněk (1992a, 1992b) and later extended by Vaněk, Mandel and Brezina (1996), was motivated by some early work on aggregation-based MG studied by Blaheta (1986, 1988).

A major work on the theory and practice of the SA algorithm is found in Vaněk, Mandel and Brezina (1998). The convergence result proved there assumes the sparsity of the coarse grid matrix and a certain ratio between the number of coarse and fine degrees of freedom. For general sparse matrices, and even for general adapted finite element matrices corresponding to

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

elliptic equations, it is difficult to verify such assumptions, and this is yet to be done.

14. Problems with discontinuous and anisotropic coefficients

A good AMG method should be robust with respect to possible heterogeneity features such discontinuous jumps and anisotropy. These heterogeneities should be detected and properly resolved automatically in an AMG process. Extensive numerical experiments have shown that AMG such as classical AMG and SA-AMG are very robust with respect to these heterogeneities. One main technique used to detect and resolve these heterogeneities is by means of the connection strength (see Section 8.2). In this section we shall use the model problem (2.1) with two special sets of coefficients, (2.9) and (2.10), to discuss how classical AMG works for problems with strong heterogeneities.

14.1. Jump coefficients

In this section we consider the classical AMG method when applied to a problem with heterogeneous (jump) coefficients, namely (2.1) with (2.9). We begin with a discussion of how the connection strength is used to define the sparsity pattern of the prolongation.

The connection strength function was introduced to handle cases such as jump coefficients and anisotropies in the matrices corresponding to discretizations of scalar PDEs. An important observation regarding classical AMG is that the prolongation matrix P, which defines the basis in the coarse space, uses only strong connections. Here we need the *strength operator* $S : V \mapsto V$ defined in (8.4). We now focus on the jump coefficient problem defined in (2.9),

$$\alpha(x) = \begin{cases} \epsilon & x \in \Omega_\epsilon, \\ 1 & x \in \Omega_1, \end{cases}$$

where ϵ is sufficiently small that the graph corresponding to the strength operator has at least two connected components. Directly from the definition of S_i in (8.8), we have the following.

- The graph corresponding to the connection strength matrix S is obtained from the graph corresponding to A by removing all entries a_{ij} corresponding to an edge connecting a vertex from the interior of Ω_ϵ to a vertex in $\overline{\Omega}_1$.
- In other words, in this setting we have a *block lower triangular* S with at least two blocks, in which the first block corresponds to the vertices interior to Ω_ϵ and the second block corresponds to the rest of the vertices.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

- The considerations above apply to any configuration of the subsets Ω_ϵ and Ω_1. In fact, they can even be disconnected, forming cross-points and so on.

In Figures 14.1–14.3 we illustrate the connection strength graphs and their connected components for specific coefficients $\alpha(x)$, which is defined as in (2.9) with $\epsilon = 10^{-3}$. The domain Ω_ϵ is a union of elements, and $T \in \Omega_\epsilon$ if and only if $(x_T - 1/2)(y_T - 1/2) < 0$, where (x_T, y_T) is the barycentre of T. The domain Ω_1 is $\Omega_1 = \Omega \setminus \overline{\Omega}_1$. In Figure 14.2, Ω_ϵ and Ω_1 are disconnected in the graph of the strength matrix S. In the graph of S, Ω_ϵ is split into two connected components, denoted by $\Omega_\epsilon^{(1)}$ and $\Omega_\epsilon^{(2)}$; Ω_1 is split into two parts, denoted by $\Omega_1^{(1)}$ and $\Omega_1^{(2)}$, which are connected via only one point, $x_0 = (1/2, 1/2)$. With a reordering of indices, the strength matrix for this case can be written as

$$S = \begin{pmatrix} S_{\epsilon,1} & & & & \\ & S_{\epsilon,2} & & & \\ & & S_{11} & & \\ & & & S_{12} & \\ & & S_{x1} & S_{x2} & S_x \end{pmatrix},$$

where $S_{\epsilon,1}$ and $S_{\epsilon,2}$ are two diagonal blocks corresponding to the two connected components $\Omega_\epsilon^{(1)}$ and $\Omega_\epsilon^{(2)}$, S_{11} and S_{12} are two diagonal blocks corresponding to the two connected components $\Omega_1^{(1)}$ and $\Omega_1^{(2)}$, S_0 is the diagonal entry corresponding to the grid point x_0, and S_{x1} and S_{x2} are low-rank matrices with only a few non-zero entries (two non-zero entries in this example), which contain the connections from x_0 to points in $\Omega_1^{(1)}$ and $\Omega_1^{(2)}$ respectively.

We define an operator $A_S : V \mapsto V'$ using A and S:

$$(A_S u, v) := \sum_{S_{ij} \neq 0} \omega_{ij} (u_i - u_j)(v_i - v_j),$$

where $\omega_{ij} = |a_{ij}|$. We have the following lemma.

Lemma 14.1. For all $v \in V$, we have

$$(A_S v, v) \leq (A_+ v, v) \lesssim (Av, v).$$

For classical AMG, the definition of the prolongation matrix P only uses strong connections, and the coarse space is the union of the coarse spaces in each strongly connected component of the graph of A_+. Note that the classical AMG construction gives the same coarse space for both A and its M-matrix relative A_+. For this reason, in the considerations that follow we write A instead of A_+ and $\mathcal{E}$ instead of $\mathcal{E}_+$, as the coarse space is defined using only entries from A_+.

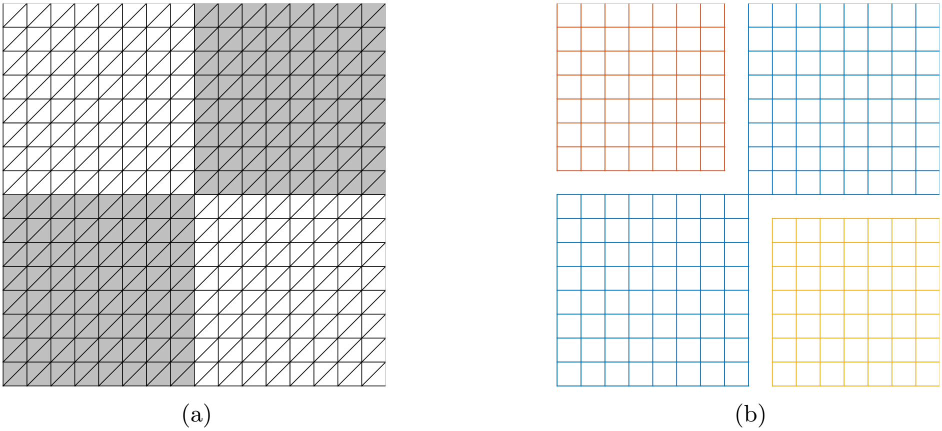


Figure 14.1. Jump coefficient problem on a uniform mesh. (a) The coefficient is 1 in the shaded region, and 10^{-3} in the blank region. (b) The strongly connected components in the graph correspond to the strength matrix S.

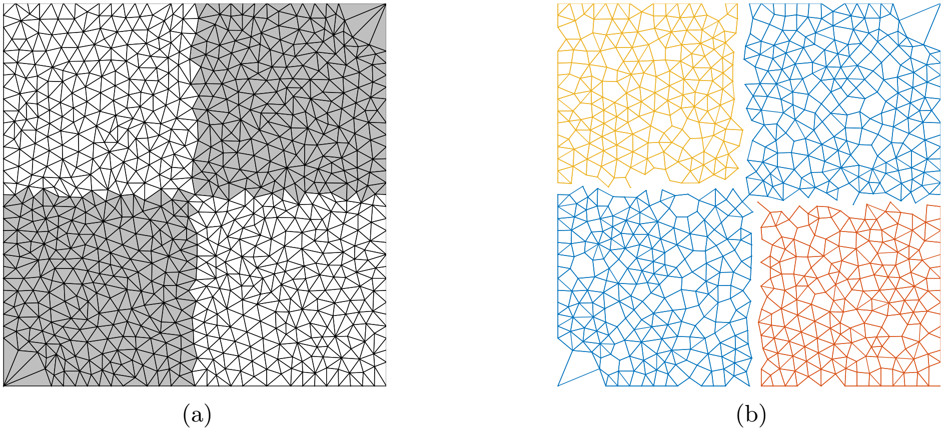
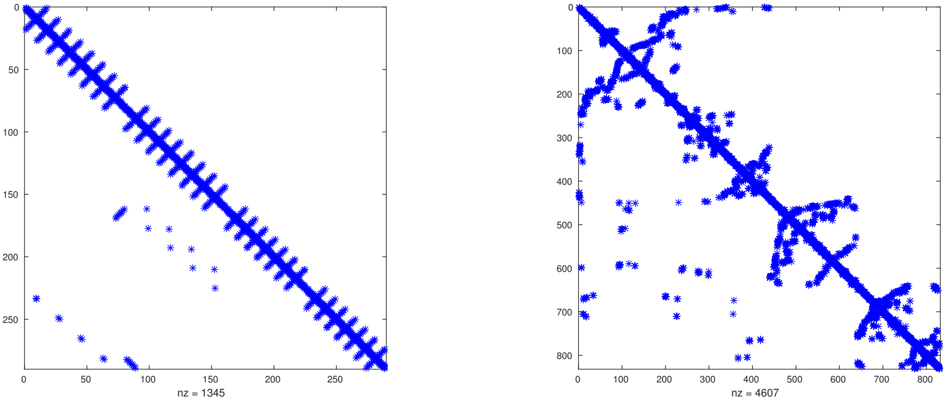


Figure 14.2. Jump coefficient problem on an unstructured mesh. (a) The coefficient is 1 in the shaded region, and 10^{-3} in the blank region. (b) The strongly connected components in the graph correspond to the strength matrix S.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press



(a)

(b)

Figure 14.3. Jump coefficient problem: non-zero pattern of strength matrix S when reordered in block lower triangular form. (a) Structured uniform grid, and (b) unstructured mesh.

We now consider the convergence of classical two-level AMG with the standard interpolation for the jump coefficient problem, and we prove a uniform convergence result for the two-level method. The same result for direct interpolation can be obtained via slight modification of the proof for the standard interpolation case. Before we go through the AMG two-level convergence proof, we first introduce the following result on a connected graph, which can be viewed as a discrete version of the Poincaré inequality.

Lemma 14.2. We consider the following graph Laplacian on a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:

$$\langle Au, v\rangle = \frac{1}{2} \sum_{(i,j)\in\mathcal{E}} (u_i - u_j)(v_i - v_j). \tag{14.1}$$

For any $v \in V$, the following estimate is true:

$$\|v - v_c\|^2_{\ell^2} \leq \mu n^2 d \langle Av, v\rangle, \tag{14.2}$$

where $n = |\mathcal{V}|$ is the size of the graph, $v_c = \sum_{j=1}^{n} w_j v_j$ is a weighted average of v, $\mu = \sum_{j=1}^{n} w_j^2$, and d is the diameter of the graph.

We now consider classical AMG coarsening as defined in Section 12.4, and with an abuse of notation we use Ω_j to denote the set of vertices defined via the C/F splitting in (12.21). The next lemma is a spectral equivalence result, showing that the local operators A_j, defined in (12.28), for a shape-regular mesh, are spectrally equivalent to a scaling of the graph Laplacian operators $A_{L,j}$ defined as

$$(A_{L,j}u, v) = \frac{1}{2} \sum_{(i,k)\in\Omega_j} (u_i - u_k)(v_i - v_k). \tag{14.3}$$

Lemma 14.3. With the assumption we made on the shape-regularity of the finite element mesh, the following inequalities hold for A_j defined as in (12.28) using the standard interpolation:

$$c_L h^{d-2}\langle A_{L,j}v_j, v_j\rangle \leq (A_j v_j, v_j) \leq c^L h^{d-2}\langle A_{L,j}v_j, v_j\rangle, \tag{14.4}$$

where $A_{L,j}$ is a graph Laplacian defined in (14.1) on the graph $\mathcal{G}_j$, h is the mesh size, and c_L, c^L are constants depending on the shape-regularity constant and the threshold θ for the connection strength.

Proof. By definition of connection strength, we have

$$a_{ii} = \sum_{k\in N_k} -a_{ik} \leq -\frac{|N_i|}{\theta} a_{ij}.$$

Since A is symmetric, we also have

$$a_{ii} \leq -\frac{|N_i|}{\theta} a_{ji}.$$

By the definition of Ω_j in the standard interpolation, for any $i \in \Omega_j \setminus \{j\}$, either $i \in F_j^s$ or there exists a $i \in F_j^s$ such that $i \in F_k^s$. For the latter, (j, k, i) forms a path between j and i going along strong connections. We then have

$$-a_{ik} \geq -\frac{\theta}{|N_k|} a_{kk} \geq -\frac{\theta}{|N_k|} a_{kj} \geq -\frac{\theta^2}{|N_k||N_j|} a_{jj}$$

and

$$a_{jj} \geq -a_{jk} \geq \frac{|N_k|}{\theta} a_{kk} \geq -\left(\frac{|N_k|}{\theta}\right)^2 a_{ik}.$$

Combining the above two inequalities and using the assumption that the mesh is shape-regular, for any $l \in \Omega_j$ that is connected to i we have

$$\sigma_1 a_{ij} \leq -a_{il} \leq \sigma_2 a_{jj},$$

with constants σ_1 and σ_2 depending on the shape-regularity constant and θ.

Since in the definition of A_j in (12.28), $\omega_e = -a_{ij}/2$ for $e = (i, j)$, we obtain

$$c_1 a_{jj}\langle A_{L,j}v_j, v_j\rangle \leq (A_j v_j, v_j) \leq c_2 a_{jj}\langle A_{L,j}v_j, v_j\rangle. \tag{14.5}$$

Then, by a scaling argument, $a_{jj} \eqsim h^{d-2}$ and the proof is complete. □

Theorem 14.4. The two-level method using a coarse space V_c defined via classical AMG is uniformly convergent.

Proof. By Theorem 5.3, we only need to show that μ_c is bounded, which can be easily obtained by combining Lemmas 14.2–14.3 with Lemma 7.3. □

14.2. Anisotropic problem

We consider the following problem:

$$\begin{cases} -u_{xx} - \varepsilon u_{yy} = f & \text{in } \Omega, \\ \dfrac{\partial u}{\partial n} = 0 & \text{on } \partial\Omega. \end{cases} \tag{14.6}$$

We discretize the problem using the finite element method on an $n \times n$ uniform triangular grid in $\Omega = (0,1) \times (0,1)$. We order the vertices of the triangulation lexicographically and denote them by $\{(ih, jh)\}_{i,j=0}^{n}$. Then the stiffness matrix is

$$A = \operatorname{diag}(-\varepsilon I, B, -\varepsilon I) \quad \text{with } B = \operatorname{diag}(-1, 2(1+\varepsilon), -1). \tag{14.7}$$

It is immediate to see that, for sufficiently small $0 < \varepsilon \ll 1$, the strength operator S takes the form

$$S = \operatorname{diag}(0, S_B, 0) \quad \text{with } S_B = \operatorname{diag}(1, 1, 1), \tag{14.8}$$

and the M-matrix relative of A is

$$A_+ = \operatorname{diag}(0, B_+, 0) \quad \text{with } B_+ = \operatorname{diag}(-1, 2, -1). \tag{14.9}$$

If we use uniform coarsening to solve the above linear problem, it has been proved by Yu, Xu and Zikatanov (2013) (see also Zikatanov 2008) that uniform convergence is not achieved when using point relaxation as a smoother and standard coarsening. A fix for this is to use connection strength and coarsen the adjacency graph of the strength operator S. The C/F splitting constructed in this way using the MIS algorithm (Algorithm 5) from Section 9.3.1 results in semi-coarsening (coarsening in only one direction).

We now move on to show uniform convergence of two-level classical AMG for the anisotropic problem (14.6). Recall that we consider a uniform grid in $\mathbb{R}^2$ with lexicographical ordering of the vertices. Then the stiffness matrix is (14.7). We further assume, without loss of generality, that $n = 2m + 1$ for some m. The coarse grid points using the strength operator defined in (14.8) are then with coordinates $((2i)h, jh)$, where $i = 1 : m$ and $j = 1 : n$.

Further, as each coarse grid function is uniquely determined by its values at the coarse points, the function that corresponds to the point $((2i)h, jh)$ for some i and j is defined as the unique piecewise linear function $\phi_{i,j,H}$ satisfying

$$\phi((2i)h, jh) = 1, \quad \phi((2i-1)h, jh) = 1/2, \quad \phi((2i+1)h, jh) = 1/2,$$

and $\phi_{i,j,H}(x) = 0$ at all other grid nodes.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Alternatively we can use the basis functions for bilinear elements, which can be written as tensor products. Let us first define the piecewise linear basis in one dimension:

$$\phi_{j,h}(t) = \begin{cases} \dfrac{(t-(j-2)h)}{h} & t \in ((j-2)h, (j-1)h), \\ \dfrac{(jh-t)}{h} & t \in ((j-1)h, jh), \\ 0 & \text{otherwise.} \end{cases}$$

The basis in V_h is then

$$\phi_{ij,h}(x,y) = \phi_{i,h}(x)\phi_{j,h}(y), \tag{14.10}$$

and for the coarse grid basis we have

$$\phi_{ij,H}(x,y) = \phi_{i,2h}(x)\phi_{j,h}(y). \tag{14.11}$$

The basis functions for the linear elements are formed by piecewise linear interpolation of the bilinear basis.

The subset Ω_{ij} is the support of this basis function, that is, the grid points where $\phi_{ij,H}$ is non-zero. More precisely,

$$\Omega_{ij} = \{((2i-1)h, jh), (2ih, jh), ((2i+1)h, jh)\}. \tag{14.12}$$

Thus Ω_{ij} consists of the coarse grid point $(2ih, jh)$ and its neighbours in the x-direction. Then we define $V_j := \mathbb{R}^3$.

The operator $\Pi_j : V_j \mapsto V$ is defined by the matrix representation of $I_h(\phi_{ij,H}\cdot)$. A_{ij} is defined as in (12.28). In this case

$$A_{ij} = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}. \tag{14.13}$$

Hence A_{ij} is symmetric positive semi-definite, and since for any $v_{ij} \in V_{ij}$ we have

$$\begin{aligned} \sum_{i,j} (v_{ij}, v_{ij})_{A_{ij}} &= \sum_{ij} \sum_{\substack{(k,l)\in\mathcal{E} \\ k,l\in\Omega_{ij}}} -a_{kl}(v_k - v_l)^2 \\ &\leq 2 \sum_{(k,l)\in\mathcal{E}} -a_{kl}(v_k - v_l)^2 \\ &= 2(v,v)_A, \end{aligned}$$

A_{ij} satisfies (6.10).

We define D_{ij} as in (12.30). As $A_{ij}\mathbf{1} = \mathbf{0}$, the minimum eigenvalue of $D_{ij}^{-1}A_{ij}$ is 0 and the corresponding eigenvector is the constant vector, we choose the local coarse space V_{ij}^c to be the space of all constant vectors in V_{ij}. The corresponding global coarse space is then defined as in (6.12).

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Theorem 14.5. The two-level method with coarse space defined above converges uniformly for the anisotropic problem in (14.6), with convergence rate independent of ε and the mesh size h.

Proof. By Theorem 5.3, the convergence rate depends on the second smallest eigenvalue of $D_{ij}^{-1}A_{ij}$, which is 1 for all i and j.

Next, Theorem 5.3 can be applied to this case, and we obtain

$$\|E\|_A \leq 1 - \frac{1}{C}, \tag{14.14}$$

with C independent of ε and the mesh size, which proves the uniform convergence of the AMG method. □

14.3. Bibliographical notes

Fast solvers for problems with heterogeneous and or anisotropic coefficients have been the focus of research for the last three or four decades. The AMG methods are among the preferred solvers due to their robust behaviour with respect to the coefficient variation and independence of the geometry. Standard multilevel methods for these problems do have limitations, as their convergence may deteriorate, as shown by Alcouffe, Brandt, Dendy and Painter (1981). The cause for this in geometric multigrid with standard coarse spaces was discussed by Bramble and Xu (1991) and Xu (1991), and later by Oswald (1999). Attempts to remove the dependence on the size of the coefficient jumps was made by introducing the matrix-dependent prolongations. We refer to Dendy (1982, 1983), Reusken (1993) and de Zeeuw (1990). Several theoretical and numerical results on geometric and algebraic methods for discontinuous coefficients can be found in the survey paper by Chan and Wan (2000) and the references therein. Anisotropic equations and AMG coarsening are considered by Brannick *et al.* (2006).

Finally, in addition to the theory presented here, some partial theoretical results on the convergence of AMG are found in the classical papers on AMG (Ruge and Stüben 1987) and smoothed aggregation (SA) (Vaněk *et al.* 1996). Related works are the frequency filtering and decompositions found in Hackbusch (1989), Wittum (1992), Wagner and Wittum (1997), Weiler and Wittum (1997) and Nägel, Falgout and Wittum (2008).

15. Bootstrap and adaptive AMG

In all the algorithms studied earlier, we assume that the near-null spaces are known in advance. Eliminating such an assumption and extending the range of applicability of optimal multigrid techniques is attempted in the framework of the bootstrap/adaptive AMG methods, which we describe in this section. In summary, the adaptive AMG (αAMG, αSA) and bootstrap AMG (BAMG) algorithms are intended for harder problems for which the

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Algorithm 7 Prolongation via least-squares minimization

```
**Input:** Matrices $\Psi \in \mathbb{R}^{n\times m}$, $\Psi_c \in \mathbb{R}^{n_c\times n_c}$, a norm $\|\cdot\|$, initial sparsity pattern $\mathrm{S}(P)$ for $P$; bound on maximum non-zeros per row $s_{\max}$.
**Output:** Prolongation matrix $P$ such that $P\Psi_c \approx \Psi$.
Set $i \leftarrow 1$.
**while** $(i \le n)$ **do**
    Find $\widetilde{p}_i \leftarrow \arg\min_{\widetilde{p}_i \in \mathbb{R}^{n_c}_{S_i}} \|\Psi_c^T \widetilde{p}_i - \widetilde{\Psi}_i\|$.
    **if** $(\|\Psi_c^T \widetilde{p}_i - \widetilde{\Psi}_i\| > \varepsilon)$ **and** $(|S(i)| \le s_{\max})$ **then**
        $S(i) \leftarrow S(i) \cup_{j_c} \{j_c\}$, where $j_c$ are close to $i$ in graph distance.
    **else**
        $i \leftarrow i+1$
    **end if**
**end while**
**for** $i = 1:n$, $j_c \in S(i)$ **do**
    **if** $(p_{i,j_c} \le \varepsilon_p)$ **then** Set $p_{i,j_c} \leftarrow 0$ **end if**
**end for**
**return** $P$
```

standard variants of the classical AMG method or the smoothed aggregation method may converge too slowly. The bootstrap/adaptive algorithms make special choices of coarse spaces. Based on a given smoother, they self-improve until they achieve the desired convergence rate.

15.1. Sparsity of prolongation matrices

Here we give a very short summary of the sparsity patterns of the prolongation matrices defined for energy-minimization AMG in Section 11, classical AMG in Section 12, and aggregation AMG in Section 13. First, for energy-minimizing AMG we have that the sparsity pattern of the prolongation could be prescribed in advance, and in some sense this approach is more general as it can also use the sparsity patterns given below for classical AMG and aggregation AMG approaches. For specific restrictions related to the number of vectors interpolated exactly by the energy-minimizing prolongation, we refer to Xu and Zikatanov (2004) and Vassilevski and Zikatanov (2006). There are also ways to define the prolongation with changing patterns. Such algorithms are useful because they provide a mechanism to control the sparsity pattern of the prolongation. Algorithm 7 was first described by Brandt (2002) and later included in the bootstrap adaptive AMG algorithm designed by Brandt, Brannick, Kahl and Livshits (2011). In the algorithm description, we refer to the graph of the M-matrix relative as defined in Section 9.

Algorithm 8 Prolongation via aggregation

```
1: **Input:** Matrix $\Psi \in \mathbb{R}^{n\times m}$, an aggregation $\cup_{i=1}^{n_c}\mathcal{A}_i = \{1,\dots,n\}$, and a prolongator smoother $S : \mathbb{R}^n \mapsto \mathbb{R}^n$ (for smoothed aggregation (SA)).
2: **Output:** Prolongation matrix $P$ such that $\text{Range}(P\Psi_c) \approx \text{Range}(\Psi)$.
3: Set $(\Psi_{\mathcal{A}_i})_{kj} = \begin{cases} \Psi_{kj} & k \in \mathcal{A}_i, j = 1 : m, \\ 0 & k \notin \mathcal{A}_i, j = 1 : m. \end{cases}$
4: Set $P = (\Psi_{\mathcal{A}_1}, \dots, \Psi_{\mathcal{A}_{nc}})$.
5: **if** (SA) **then** $P \leftarrow (I - R_s A)P$ **end if**
6: **return** $P$
```

To conclude this section, we point out that the algorithmic details for the construction of *direct*, *standard* and *multipass* interpolations are already described in detail in Section 12.

15.2. Notation

Given a matrix $A \in \mathbb{R}^{n\times n}$ and a relaxation (smoother) R for this matrix, we use an adaptive procedure to construct a sequence of coarse spaces which are characterized by the sequence of prolongation matrices

$$P_j^m : \mathbb{R}^{n_j^m} \mapsto \mathbb{R}^{n_{j+1}^m}.$$

The corresponding V-cycle matrix is denoted by B^m. We introduce the following notation, used throughout this section.

- For a matrix $Y \in \mathbb{R}^{n\times m}$, we set
$$Y = (y_1, \dots, y_m) = \begin{pmatrix} \widetilde{y}_1^T \\ \vdots \\ \widetilde{y}_n^T \end{pmatrix}.$$
Clearly, $\{y_i\}_{i=1}^m$ are the columns of Y and $\{\widetilde{y}_j^T\}_{j=1}^n$ are the rows of Y.
- $\mathcal{V} = (V_1, \dots, V_J)$ is a multilevel hierarchy of spaces $V_j \subset V_{j+1}$, $j = 1 : (J-1)$ and the fine grid spaces are denoted by V_J.
- We let $\{P_{j-1}^j\}_{j=2}^J$ denote the prolongation matrix from a coarser level $(j-1)$ to finer level j, and $P_j = P_{J-1}^J P_{J-2}^{J-1} \cdots P_j^{j+1}$, $j = 1, \dots, (J-1)$ is the prolongation from level j to level J
- We let $\mathcal{P}_j = \{P_{j-1}^j\}_{j=2}^j$ and $\mathcal{P} = \mathcal{P}_J$ denote the set of all prolongations up to level j. As we have mentioned on several occasions, the set of prolongation matrices $\mathcal{P}$ completely determines the multilevel hierarchy of spaces $\mathcal{V}$.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

- A set of test vectors on every level is available:

$$\mathcal{C} = \{\Psi_1, \ldots, \Psi_J\}, \quad \Psi_j \in \mathbb{R}^{n_j \times m}.$$

  In an adaptive method, the hierarchy of coarse spaces is constructed so that $\Psi_j \approx P_{j-1}^j \Psi_{j-1}$, or P_{j-1}^j is constructed so that Ψ_j can be well approximated by elements from Range P_{j-1}^j.

- We need the standard V-cycle preconditioner $B(\mathcal{P}_j)$ with a hierarchy of spaces given by $\mathcal{P}_j$, starting at level j.

- A_j is the restriction of the fine grid matrix A on level j, and M_j is the restriction of the 'mass' matrix, defined as $A_j = P_j^T A P_j$, $M_j = P_j^T P_j$.

A generic adaptive AMG algorithm changes the set $\mathcal{P}$, adjusting the hierarchy of spaces $\mathcal{V}$. In general, in the adaptive procedure the number of levels is not known, and in some of the algorithms below we use V_1 as the finest space by mapping the indexes in the notations above as

$$j \leftarrow J - j + 1, \quad j = 1, \ldots, J.$$

15.3. A basic adaptive algorithm

We now describe a generic adaptive algorithm for constructing coarse spaces. We use an approach slightly different to that found in the literature, and fit both BAMG and αSA into one framework.

Step 0. Given $A \in \mathbb{R}^{n\times n}$ and the associated graph $\mathcal{G}(A) = (\mathcal{V}, \mathcal{E})$.

Step 1. Initialization.

1: Choose $m_0 \geq 1$, $q \geq 1$, $1 \leq n_0 < n$ and $\delta_0 \in (0, 1)$.

2: $\mathcal{P} \leftarrow \emptyset$.

3: $V_c \leftarrow V = \mathbb{R}^n$; $\mathcal{V} \leftarrow \{V_c\}$.

4: $n_c \leftarrow n$; $m \leftarrow m_0$.

5: $B \leftarrow R$.

6: Randomly pick m test vectors $\boldsymbol{\Psi}_0 = (\psi_1, \ldots \psi_m)$, $\boldsymbol{\Psi}_0 \in \mathbb{R}^{n\times m}$.

7: $\boldsymbol{\Psi} \leftarrow \boldsymbol{\Psi}_0$; $\mathcal{C} \leftarrow \{\boldsymbol{\Psi}_0\}$.

Step 2. If $n_c \leq n_0$, go to **Step 3**, else do:

1: Make a copy of $\boldsymbol{\Psi}$: $\hat{\boldsymbol{\Psi}} \leftarrow \boldsymbol{\Psi}$. Then compute

$$\boldsymbol{\Psi} \leftarrow (I - BA)^q \boldsymbol{\Psi}, \quad \delta = \max_{1\leq i\leq m} \frac{\|\Psi_i\|_A}{\|\hat{\Psi}_i\|_A}. \tag{15.1}$$

Algorithm 9 Approximation of l_e eigenpairs of A

```
for $j = J, \dots, 1$ do
    if $j = J$ (coarsest grid) then
        Find $\{(\varphi_J^{(k)}, \lambda_J^{(k)})\}_{k=1}^{l_e}$– the solutions to the eigenvalue problems:
        $A_J \varphi_J^{(k)} = \lambda_J^{(k)} M_J \varphi_J^{(k)}$, $k = 1, \dots, l_e$;
    else
        for $k = 1, \dots, l_e$ do
            Set $v_j^{(k)} = (P_{j-1}^j)^T \varphi_{j-1}^{(k)}$, $\mu_j^{(k)} = \lambda_{j-1}^{(k)}$, and $C_j^{(k)} = A_j - \mu_j^{(k)} M_j$
            Relax on $C_j^{(k)} w = 0$, i.e., $\varphi_j^{(k)} = (I - S_j^{(k)} C_j^{(k)})\, v_j^{(k)}$.
            Set $\lambda_j^{(k)} = \dfrac{\langle A_j \varphi_j^{(k)}, \varphi_j^{(k)} \rangle}{\langle M_j \varphi_j^{(k)}, \varphi_j^{(k)} \rangle}$
        end for
    end if
end for
```

2: If $\delta \le \delta_0$ then **stop**.

3: Use a coarsening strategy (see Section 9) to update n_c and find a set of coarse grid DOFs $\{N_i(\cdot)\}_{i=1}^{n_c}$. Then set

$$V_c \leftarrow \mathbb{R}^{n_c}, \quad \mathcal{V} \leftarrow \mathcal{V} \cup \{V_c\}.$$

4: Form the 'restriction' of $\mathbf{\Psi}$ to the coarse space:

$$\mathbf{\Psi}_c \leftarrow \text{Restrict}(\mathbf{\Psi}, \{N_i(\cdot)\}_{i=1}^{n_c}).$$

Then set $\mathcal{C} \leftarrow \mathcal{C} \cup \{\mathbf{\Psi}_c\}$.

5: Identify a sparsity pattern $\mathrm{S}(P)$ (see Section 15.1; also Sections 12.3 and 13.2).

6: Find a prolongation matrix P using $\mathbf{\Psi}_c$ and $\mathbf{\Psi}$ by applying Algorithms 7 or 8. Then set

$$\mathcal{P} \leftarrow \mathcal{P} \cup \{P\}, \quad A \leftarrow P^T A P, \quad \mathbf{\Psi} \leftarrow \mathbf{\Psi}_c, B \leftarrow R_c,$$

where R_c is the relaxation on the coarse grid V_c.

Step 3. Order the spaces in $\mathcal{V}$ increasing with respect to their dimension as

$$\mathcal{V} = \{V_1, V_2, \dots, V_J\} \quad \text{with } V_J = \mathbb{R}^n,$$

and the corresponding prolongation matrices

$$\mathcal{P} = \{P_j^{j+1}\}_{j=1}^{J-1}, \quad P_j^{j+1} : V_j \mapsto V_{j+1}, \quad j = 1, 2, \dots, J-1.$$

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Algorithm 10 αSA: adding a test vector ψ to the current set of test vectors Ψ_J

```
Step 1. Update $\boldsymbol{\Psi}_J$ by adding $\psi$ as a new column: $\boldsymbol{\Psi}_J \leftarrow [\boldsymbol{\Psi}_J, \psi]$.
Step 2. For $l = J, \dots, 3$:
  1: Update $\Psi_{l-1}$ by evaluating the coarse grid degrees of freedom $N_j(\psi)$, $j = 1 : n_{l-1}$. This process adds $n_{l-1}$ rows to $\Psi_{l-1}$.
  2: Use Algorithm 8 (SA) to define a prolongation $P^l_{l-1}$ matrix and coarse grid operator $A_{l-1}$.
  3: Update $\mathcal{P}_l \leftarrow \{P^l_{l-1}\} \cup \mathcal{P}_{l-1}$.
  4: As the number of rows in $\Psi_{l-1}$ was increased in Step 2.1, we need to change $P^{l-1}_{l-2}$ in $\mathcal{P}_{l-1}$ in order to keep the set $\mathcal{P}$ consistent. We define $\widetilde{P}^{l-1}_{l-2}$ as the ‘bridge’ prolongation via Algorithm 11, and we set
       $\widetilde{\mathcal{P}}_{l-1} =\leftarrow \{\widetilde{P}^{l-1}_{l-2}\} \cup \mathcal{P}_{l-2}$.
  5: Test the convergence of $B(\widetilde{\mathcal{P}}_{l-1})$ on the newly added test vector $\psi_{l-1}$:
       $\hat{\psi}_{l-1} \leftarrow \psi_{l-1}, \quad \psi_{l-1} \leftarrow (I - B_{l-1}(\widetilde{\mathcal{P}}_{l-1})A)^q(\psi_{l-1})$.
     If $\left(\frac{\|P_{l-1}\psi_{l-1}\|^2_A}{\|P_{l-1}\widehat{\psi}_{l-1}\|^2_A}\right)^{1/q} \le \delta$ then stop.
  6: Update the coarse representation of $\boldsymbol{\Psi}_{l-1}$:
       $\boldsymbol{\Psi}_{l-1} \leftarrow [\hat{\boldsymbol{\Psi}}^{l-1}, \psi_{l-1}]$.
```

Set B to be the V-cycle AMG method based on the above coarse spaces and prolongation matrices.

Step 4. Compute δ in (15.1) using current B. If $\delta \le \delta_0$, **stop**. Else, update $\mathcal{C}$, $\mathcal{P}$ and $\mathcal{V}$ by modifying, removing from or adding to the set of test vectors; bootstrap AMG uses Algorithm 9 and adaptive SA uses Algorithm 10. Then go to **Step 3**.

The restriction operator in Step 2.4 is as follows:

$$\begin{aligned} \boldsymbol{\Psi}_c &= (N_i(\psi_j))_{1\le i\le n_c, 1\le j\le m} && \text{(bootstrap AMG)},\\ \boldsymbol{\Psi}_c &= (N_i(\psi_j))_{1\le i\le n_c} \otimes e_j, \quad j = 1 : m && \text{(adaptive aggregation)}. \end{aligned}$$

In the test vectors all the sets $\mathcal{P}$, $\mathcal{C}$, $\mathcal{V}$ are chosen so that J corresponds to the finest grid.

Algorithm 11 αSA setup: construction of a 'bridging' prolongator

```
1: Denote the last column of $\Psi_{l-1}$ by $\psi_{l-1}$, and let $\hat{\Psi}_{l-1}$ consist of all other columns of $\Psi_{l-1}$.
2: Create a prolongation $P_{l-2}^{l-1}$ using Algorithm 8 with the $\Psi_{l-2}$ by fitting all the vectors in $\hat{\Psi}_{l-1}$.
```

15.4. Bibliographical notes

The AMG approaches we have considered are aimed at the adaptive choice of coarse spaces and multilevel hierarchies in an AMG algorithm. The majority of adaptive AMG methods known to date use the operator A and aim to capture the worst-case errors. All specific details regarding *bootstrap AMG* (BAMG) are found in Brandt (2002), Brandt *et al.* (2011) and Brandt, Brannick, Kahl and Livshits (2015). Adaptive classical AMG is described in Brezina *et al.* (2006), and adaptive smoothed aggregation AMG is discussed in detail by Brezina *et al.* (2004, 2005). While the adaptive and bootstrap MG processes have been successful in several settings, they are still only a heuristic attempt to overcome serious obstacles to achieving good performance from the classical AMG perspective. Indeed, the costs of achieving added robustness using a bootstrap or adaptive MG algorithm are significant.

Central to adaptive MG is an important distinction between the role of the underlying multigrid algorithm (aggregation or classical AMG) and what additional adaptive elements it should provide. If the idea of self-improving the coarse spaces is poorly implemented, the adaptive and bootstrap multigrid algorithms can easily degenerate into an algorithm with no better convergence properties than a classical Krylov method preconditioned by the MG smoother (Falgout 2004).

The basic ideas on adaptive AMG are outlined in early works on classical AMG (Brandt *et al.* 1982). In fact, the adaptive process of constructing interpolation based on fitting a set of test vectors for badly scaled matrices was introduced by Ruge (1983). Some of the basic ideas of adaptive AMG can also be found in Ruge (1985, 1986), McCormick and Ruge (1989) and Brandt *et al.* (1985). Further advances in adaptive AMG methods, along with new ideas for using eigenvector approximations to guide the adaptive process, were introduced in the bootstrap AMG (BAMG) framework of Brandt (2002). The BAMG method is a self-learning multigrid algorithm that automatically determines the algebraically smooth errors in a given problem, and was further developed by Brandt *et al.* (2011, 2015). Adaptive AMG algorithms were further developed and new algorithms were introduced in some more recent works: αAMG in the framework of classical AMG (Brezina *et al.* 2006), and αSA in the framework of smoothed

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

aggregation AMG (Brezina *et al.* 2004, Brezina *et al.* 2005). Other adaptive multilevel algorithms are the adaptive filtering and the filtering decompositions (Wittum 1992, Wagner and Wittum 1997) and the multilevel multigraph algorithms (Bank and Smith 2002).

These references have more specific details on implementation of BAMG, αAMG and αSA. One improvement on the original BAMG method is that of Manteuffel, McCormick, Park and Ruge (2010), who introduce an *indirect* BAMG (iBAMG) method. Compared to BAMG, the iBAMG method is closer to the spirit of classical AMG, which attempts to collapse unwanted connections on the assumption that the smooth error is locally constant.

Brandt *et al.* (2011) pair BAMG with an adaptive relaxation (Brandt 2000, Kahl 2009) and a multigrid eigensolver (Brezina *et al.* 2008). A combination of the bootstrap cycling scheme (Livshits 2008, Kahl 2009) and then the multigrid eigensolver is used to compute sufficiently accurate sets of test vectors, and adaptive relaxation is used to improve the AMG setup cycle.

In recent years, other adaptive approaches to constructing hierarchy of spaces have been introduced. The classical AMG approach to defining the hierarchy of spaces is considered by Brannick, Frommer, Kahl, MacLachlan and Zikatanov (2010) and MacLachlan *et al.* (2006) in the framework of adaptive reduction algorithms. Adaptive BoxMG was considered by MacLachlan *et al.* (2012). A specialized adaptive approach for Markov chains has been presented by De Sterck, Miller, Treister and Yavneh (2011).

16. Concluding remarks

In this paper we try to give a coherent presentation of a number of algebraic multigrid (AMG) methods. However, this presentation, limited by our current theoretical understanding of AMG algorithms in general, is by no means complete. We choose to include those AMG algorithms that can fit into the theoretical frameworks presented in this paper. One notable exception is the bootstrap and adaptive AMG presented in Section 15. These types of algorithm still lack good theoretical understanding, but they provide an algorithmic framework to use and extend the AMG algorithms presented in Sections 10–13 to a wider range of applications.

There are still many AMG algorithms in the literature that we are not able to include in this paper, for two main reasons. One is that these algorithms cannot be easily cast within our theoretical framework; another is that there are algorithms that the authors are yet to comprehend on a reasonable theoretical level. Examples of algorithms and results that need further investigation and analysis include adaptive filtering, multilevel ILU methods, the multilevel convergence properties of SA-AMG, BAMG, adaptive AMG, and many others.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

The AMG methods studied in this paper are obtained by optimizing the choice of coarse spaces based on a given smoother. Indeed, almost all the existing AMG methods follow this strategy. It is possible, however, that an AMG method could also be designed by optimizing the choice of smoother based on a given coarsening strategy. This, in our view, is a subject worthy of further investigation. Theoretically, it would also be possible to optimize both coarsening and smoother simultaneously.

Finally, we would like to note that several AMG software packages have been developed and are currently in use, most notably:[1]

Hypre	http://acts.nersc.gov/hypre/
Trilinos	https://trilinos.org/
Multigraph	http://ccom.ucsd.edu/~reb/software.html
AGMG	http://homepages.ulb.ac.be/~ynotay/AGMG/
PyAMG	http://pyamg.org/
FASP	http://fasp.sourceforge.net/

Acknowledgements

The authors wish to thank Stephen F. McCormick, Xiaozhe Hu, James Brannick and Scott MacLachlan for many helpful discussions and suggestions, and especially Hongxuan Zhang for his help throughout the preparation of this paper.

The work of the first author was supported by US DOE grant DE-SC-0009249, as part of the 'Collaboratory on Mathematics for Mesoscopic Modeling of Materials', and NSF grants DMS-1412005 and DMS-1522615. The work of the second author was supported by NSF grants DMS-1418843 and DMS-1522615.

REFERENCES

R. Alcouffe, A. Brandt, J. Dendy Jr and J. Painter (1981), 'The multigrid method for the diffusion equation with strongly discontinuous coefficients', *SIAM J. Sci. Statist. Comput.* **2**, 430–454.

O. Axelsson and P. Vassilevski (1991), 'A black box generalized conjugate gradient solver with inner iterations and variable-step preconditioning', *SIAM J. Matrix Anal. Appl.* **12**, 625–644.

O. Axelsson and P. Vassilevski (1994), 'Variable-step multilevel preconditioning methods I: Self-adjoint and positive definite elliptic problems', *Numer. Linear Algebra Appl.* **1**, 75–101.

[1] The URLs cited in this work were correct at the time of going to press, but the publisher and the authors make no undertaking that the citations remain live or are accurate or appropriate.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

B. Ayuso de Dios, F. Brezzi, L. Marini, J. Xu and L. Zikatanov (2014), 'A simple preconditioner for a discontinuous Galerkin method for the Stokes problem', *J. Sci. Comput.* **58**, 517–547.

I. Babuška and J. Osborn (1983), 'Generalized finite element methods: Their performance and their relation to mixed methods', *SIAM J. Numer. Anal.* **20**, 510–536.

N. Bahvalov (1966), 'Convergence of a relaxation method under natural constraints on an elliptic operator', *Ž. Vyčisl. Mat. i Mat. Fiz.* **6**, 861–883.

R. Bank and C. Douglas (1985), 'Sharp estimates for multigrid rates of convergence with general smoothing and acceleration', *SIAM J. Numer. Anal.* **22**, 617–633.

R. Bank and T. Dupont (1980), Analysis of a two-level scheme for solving finite element equations. Technical report CNA–159, Center for Numerical Analysis, University of Texas, Austin.

R. Bank and R. Smith (2002), 'An algebraic multilevel multigraph algorithm', *SIAM J. Sci. Comput.* **23**, 1572–1592.

R. Blaheta (1986), 'A multilevel method with correction by aggregation for solving discrete elliptic problems', *Aplikace Matematiky* **31**, 365–378.

R. Blaheta (1988), 'Iterative methods for the numerical solution of the boundary value problem of elasticity' (abstract of thesis), *Commentationes Mathematicae Universitatis Carolinae* **29**, 200.

D. Braess and W. Hackbusch (1983), 'A new convergence proof for the multigrid method including the V-cycle', *SIAM J. Numer. Anal.* **20**, 967–975.

J. Bramble (1993), *Multigrid Methods*, Vol. 294 of Pitman Research Notes in Mathematics, Longman Scientific & Technical.

J. Bramble and J. Pasciak (1987), 'New convergence estimates for multigrid algorithms', *Math. Comp.* **49**(180), 311–329.

J. Bramble and J. Xu (1991), 'Some estimates for a weighted L^2 projection', *Math. Comp.* **56**(194), 463–476.

J. Bramble, J. Pasciak and J. Xu (1990), 'Parallel multilevel preconditioners', *Math. Comp.* **55**(191), 1–22.

J. Bramble, J. Pasciak and J. Xu (1991*a*), 'The analysis of multigrid algorithms with nonnested spaces or noninherited quadratic forms', *Math. Comp.* **56**(193), 1–34.

J. Bramble, J. Pasciak, J. Wang and J. Xu (1991*b*), 'Convergence estimates for multigrid algorithms without regularity assumptions', *Math. Comp.* **57**(195), 23–45.

J. Bramble, J. Pasciak, J. Wang and J. Xu (1991*c*), 'Convergence estimates for product iterative methods with applications to domain decomposition', *Math. Comp.* **57**(195), 1–21.

A. Brandt (1973), Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems. In *Proc. Third International Conference on Numerical Methods in Fluid Mechanics*, Vol. 18 of Lecture Notes in Physics, Springer, pp. 82–89.

A. Brandt (1977), 'Multi-level adaptive solutions to boundary-value problems', *Math. Comp.* **31**(138), 333–390.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

A. Brandt (1986), 'Algebraic multigrid theory: The symmetric case', *Appl. Math. Comput.* **19**, 23–56.

A. Brandt (2000), 'General highly accurate algebraic coarsening', *Electron. Trans. Numer. Anal.* **10**, 1–20.

A. Brandt (2002), Multiscale scientific computation: Review 2001. In *Multiscale and Multiresolution Methods: Theory and Applications* (T. Barth, T. Chan and R. Haimes, eds), Springer, pp. 3–95.

A. Brandt, J. Brannick, K. Kahl and I. Livshits (2011), 'Bootstrap AMG', *SIAM J. Sci. Comput.* **33**, 612–632.

A. Brandt, J. Brannick, K. Kahl and I. Livshits (2015), 'Bootstrap algebraic multigrid: Status report, open problems, and outlook', *Numer. Math. Theory Methods Appl.* **8**, 112–135.

A. Brandt, S. McCormick and J. Ruge (1982), Algebraic multigrid (AMG) for automatic multigrid solutions with application to geodetic computations. Technical report, Institute for Computational Studies, Fort Collins, CO.

A. Brandt, S. McCormick and J. Ruge (1985), Algebraic multigrid (AMG) for sparse matrix equations. In *Sparsity and its Applications: Loughborough 1983*, Cambridge University Press, pp. 257–284.

J. Brannick and R. Falgout (2010), 'Compatible relaxation and coarsening in algebraic multigrid', *SIAM J. Sci. Comput.* **32**, 1393–1416.

J. Brannick and L. Zikatanov (2006), Algebraic multigrid methods based on compatible relaxation and energy minimization. In *Domain Decomposition Methods in Science and Engineering XVI* (O. Widlund and D. Keyes, eds), Springer, pp. 15–26.

J. Brannick, M. Brezina, S. MacLachlan, T. Manteuffel, S. McCormick and J. Ruge (2006), 'An energy-based AMG coarsening strategy', *Numer. Linear Algebra Appl.* **13**, 133–148.

J. Brannick, Y. Chen and L. Zikatanov (2012), 'An algebraic multigrid method for anisotropic elliptic equations based on subgraph matching equations based on subgraph matching', *Numer. Linear Algebra Appl.* **19**, 279–295.

J. Brannick, A. Frommer, K. Kahl, S. MacLachlan and L. Zikatanov (2010), 'Adaptive reduction-based multigrid for nearly singular and highly disordered physical systems', *Electron. Trans. Numer. Anal.* **37**, 276–295.

M. Brezina and P. Vassilevski (2011), Smoothed aggregation spectral element agglomeration AMG: SA-ρAMGe. In *Large-Scale Scientific Computing 2011* (I. Lirkov, S. Margenov and J. Wasniewski, eds), Vol. 7116 of Lecture Notes in Computer Science, Springer, pp. 3–15.

M. Brezina, A. Cleary, R. Falgout, V. Henson, J. Jones, T. Manteuffel, S. McCormick and J. Ruge (2001), 'Algebraic multigrid based on element interpolation (AMGe)', *SIAM J. Sci. Comput.* **22**, 1570–1592.

M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick and J. Ruge (2004), 'Adaptive smoothed aggregation (αSA)', *SIAM J. Sci. Comput.* **25**, 1896–1920.

M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick and J. Ruge (2005), 'Adaptive smoothed aggregation (αSA) multigrid', *SIAM Rev.* **47**, 317–346.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick and J. Ruge (2006), 'Adaptive algebraic multigrid', *SIAM J. Sci. Comput.* **27**, 1261–1286.

M. Brezina, T. Manteuffel, S. McCormick, J. Ruge, G. Sanders and P. Vassilevski (2008), 'A generalized eigensolver based on smoothed aggregation (GES-SA) for initializing smoothed aggregation (SA) multigrid', *Numer. Linear Algebra Appl.* **15**, 249–269.

W. Briggs, V. Henson and S. McCormick (2000), *A Multigrid Tutorial*, second edition, SIAM.

T. Chan and W. Wan (2000), 'Robust multigrid methods for nonsmooth coefficient elliptic linear systems', *J. Comput. Appl. Math.* **123**, 323–352.

T. Chan, J. Xu and L. Zikatanov (1998), An agglomeration multigrid method for unstructured grids. In *Domain Decomposition Methods 10*, Vol. 218 of Contemporary Mathematics, AMS, pp. 67–81.

T. Chartier, R. Falgout, V. Henson, J. Jones, T. Manteuffel, S. McCormick, J. Ruge and P. Vassilevski (2003), 'Spectral AMGe (ρAMGe)', *SIAM J. Sci. Comput.* **25**, 1–26.

L. Chen (2011), Deriving the X-Z identity from auxiliary space method. In *Domain Decomposition Methods in Science and Engineering XIX*, Vol. 78 of Lecture Notes in Computational Science and Engineering, Springer, pp. 309–316.

L. Chen, R. Nochetto and J. Xu (2012), 'Optimal multilevel methods for graded bisection grids', *Numer. Math.* **120**, 1–34.

P. Ciarlet (2002), *The Finite Element Method for Elliptic Problems*, Vol. 40 of Classics in Applied Mathematics, SIAM. Reprint of the 1978 original.

A. Cleary, R. Falgout, V. Henson and J. Jones (1998), Coarse-grid selection for parallel algebraic multigrid. In *Solving Irregularly Structured Problems in Parallel: Berkeley, CA, 1998*, Vol. 1457 of Lecture Notes in Computer Science, Springer, pp. 104–115.

R. Courant and D. Hilbert (1924), *Methoden der Mathematischen Physik*, Vol. 1, Julius Springer.

P. D'Ambra and P. Vassilevski (2013), 'Adaptive AMG with coarsening based on compatible weighted matching', *Comput. Vis. Sci.* **16**, 59–76.

P. D'Ambra and P. Vassilevski (2014), αAMG based on weighted matching for systems of elliptic PDEs arising from displacement and mixed methods. Technical report LNLL-CONF-656131, Lawrence Livermore National Laboratory.

H. De Sterck, R. Falgout, J. Nolting and U. Yang (2008), 'Distance-two interpolation for parallel algebraic multigrid', *Numer. Linear Algebra Appl.* **15**, 115–139.

H. De Sterck, K. Miller, E. Treister and I. Yavneh (2011), 'Fast multilevel methods for Markov chains', *Numer. Linear Algebra Appl.* **18**, 961–980.

P. de Zeeuw (1990), 'Matrix-dependent prolongations and restrictions in a blackbox multigrid solver', *J. Comput. Appl. Math.* **33**, 1–27.

J. Dendy Jr (1982), 'Black box multigrid', *J. Comput. Phys.* **48**, 366–386.

J. Dendy Jr (1983), 'Black box multigrid for nonsymmetric problems', *Appl. Math. Comput.* **13**, 261–283.

R. Diestel (2010), *Graph Theory*, fourth edition, Vol. 173 of Graduate Texts in Mathematics, Springer.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

C. Dohrmann, A. Klawonn and O. Widlund (2008), A family of energy minimizing coarse spaces for overlapping Schwarz preconditioners. In *Domain Decomposition Methods in Science and Engineering XVII*, Vol. 60 of Lecture Notes in Computational Science and Engineering, Springer, pp. 247–254.

Y. Efendiev, J. Galvis and P. Vassilevski (2011), Spectral element agglomerate algebraic multigrid methods for elliptic problems with high contrast coefficients. In *Domain Decomposition Methods in Science and Engineering XIX* (Y. Huang, R. Kornhuber, O. Widlund and J. Xu, eds), Vol. 78 of Lecture Notes in Computational Science and Engineering, Springer, pp. 407–414.

Y. Efendiev, T. Hou and X.-H. Wu (2000), 'Convergence of a nonconforming multiscale finite element method', *SIAM J. Numer. Anal.* **37**, 888–910.

R. Falgout (2004), A note on the relationship between adaptive AMG and PCG. Technical report UCRL-TR-205838, Lawrence Livermore National Laboratory.

R. Falgout and P. Vassilevski (2004), 'On generalizing the AMG framework', *SIAM J. Numer. Anal.* **42**, 1669–1693. Also available as technical report UCRL-JC-150807, Lawrence Livermore National Laboratory.

R. Falgout, P. Vassilevski and L. Zikatanov (2005), 'On two-grid convergence estimates', *Numer. Linear Algebra Appl.* **12**, 471–494.

K. Fan (1949), 'On a theorem of Weyl concerning eigenvalues of linear transformations I', *Proc. Nat. Acad. Sci. USA* **35**, 652.

R. Fedorenko (1961), 'A relaxation method of solution of elliptic difference equations', *Ž. Vyčisl. Mat. i Mat. Fiz.* **1**, 922–927.

R. Fedorenko (1964), 'On the speed of convergence of an iteration process', *Ž. Vyčisl. Mat. i Mat. Fiz.* **4**, 559–564.

A. Gibbons (1985), *Algorithmic Graph Theory*, Cambridge University Press.

G. Golub and Q. Ye (1999/2000), 'Inexact preconditioned conjugate gradient method with inner–outer iteration', *SIAM J. Sci. Comput.* **21**, 1305–1320.

L. Grasedyck, L. Wang and J. Xu (2015), 'A nearly optimal multigrid method for general unstructured grids', *Numer. Math.* pp. 1–30.

T. Grauschopf, M. Griebel and H. Regler (1997), 'Additive multilevel preconditioners based on bilinear interpolation, matrix-dependent geometric coarsening and algebraic multigrid coarsening for second-order elliptic PDEs', *Appl. Numer. Math.* **23**, 63–95.

M. Griebel and P. Oswald (1995), 'On the abstract theory of additive and multiplicative Schwarz algorithms', *Numer. Math.* **70**, 163–180.

W. Hackbusch (1978), A fast iterative method for solving Poisson's equation in a general region. In *Numerical Treatment of Differential Equations: Proc. Conf., Math. Forschungsinst., Oberwolfach 1976*, Vol. 631 of Lecture Notes in Mathematics, Springer, pp. 51–62. Longer version: Ein Iteratives Verfahren zur Schnellen Auflösung Elliptischer Randwertprobleme, Report 76-12 (1976), Mathematisches Institut der Universität zu Köln.

W. Hackbusch (1979), 'On the computation of approximate eigenvalues and eigenfunctions of elliptic operators by means of a multi-grid method', *SIAM J. Numer. Anal.* **16**, 201–215. Earlier version: Report 77-10, Mathematisches Institut der Universität zu Köln, 1977.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

W. Hackbusch (1985), *Multigrid Methods and Applications*, Vol. 4 of Springer Series in Computational Mathematics, Springer.

W. Hackbusch (1989), 'The frequency decomposition multi-grid method I: Application to anisotropic equations', *Numer. Math.* **56**, 229–245.

W. Hackbusch (1994), *Iterative Solution of Large Sparse Systems of Equations*, Vol. 95 of Applied Mathematical Sciences, Springer.

P. Halmos (1974), *Finite-dimensional Vector Spaces*, second edition, Undergraduate Texts in Mathematics, Springer.

V. Henson and P. Vassilevski (2001), 'Element-free AMGe: General algorithms for computing interpolation weights in AMG', *SIAM J. Sci. Comput.* **23**, 629–650.

V. Henson and U. Yang (2002), 'BoomerAMG: A parallel algebraic multigrid solver and preconditioner', *Appl. Numer. Math.* **41**, 155–177.

T. Hou, X.-H. Wu and Z. Cai (1999), 'Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients', *Math. Comp.* **68**(227), 913–943.

X. Hu, P. Vassilevski and J. Xu (2013), 'Comparative convergence analysis of nonlinear AMLI-cycle multigrid', *SIAM J. Numer. Anal.* **51**, 1349–1369.

J. Jones and P. Vassilevski (2001), 'AMGe based on element agglomeration', *SIAM J. Sci. Comput.* **23**, 109–133.

K. Kahl (2009), Adaptive algebraic multigrid for lattice QCD computations. PhD thesis, Fachbereich Mathematik und Naturwissenschaften, Bergische Universität Wuppertal.

G. Karypis and V. Kumar (1998), 'A fast and high quality multilevel scheme for partitioning irregular graphs', *SIAM J. Sci. Comput.* **20**, 359–392.

H. Kim, J. Xu and L. Zikatanov (2003), 'A multigrid method based on graph matching for convection diffusion equations', *Numer. Linear Algebra Appl.* **10**, 181–195.

H. Kim, J. Xu and L. Zikatanov (2004), 'Uniformly convergent multigrid methods for convection–diffusion problems without any constraint on coarse grids', *Adv. Comput. Math.* **20**, 385–399.

T. Kolev and P. Vassilevski (2006), 'AMG by element agglomeration and constrained energy minimization interpolation', *Numer. Linear Algebra Appl.* **13**, 771–788.

J. Kraus (2002), 'An algebraic preconditioning method for M-matrices: Linear versus non-linear multilevel iteration', *Numer. Linear Algebra Appl.* **9**, 599–618.

J. Kraus (2008), 'Algebraic multigrid based on computational molecules 2: Linear elasticity problems', *SIAM J. Sci. Comput.* **30**, 505–524.

J. Kraus and J. Schicho (2006), 'Algebraic multigrid based on computational molecules 1: Scalar elliptic problems', *Computing* **77**, 57–75.

J. Lai and L. Olson (2011), 'Algebraic multigrid for high-order hierarchical $H(\text{curl})$ finite elements', *SIAM J. Sci. Comput.* **33**, 2888–2902.

Y.-J. Lee, J. Wu, J. Xu and L. Zikatanov (2007), 'Robust subspace correction methods for nearly singular systems', *Math. Models Methods Appl. Sci.* **17**, 1937–1963.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

Y.-J. Lee, J. Wu, J. Xu and L. Zikatanov (2008), 'A sharp convergence estimate for the method of subspace corrections for singular systems of equations', *Math. Comp.* **77**(262), 831–850.

K. Lipnikov, J. Moulton and D. Svyatskiy (2011), 'Adaptive strategies in the multilevel multiscale mimetic (M^3) method for two-phase flows in porous media', *Multiscale Model. Simul.* **9**, 991–1016.

O. Livne (2004), 'Coarsening by compatible relaxation', *Numer. Linear Algebra Appl.* **11**, 205–227.

O. Livne and A. Brandt (2012), 'Lean algebraic multigrid (LAMG): Fast graph Laplacian linear solver', *SIAM J. Sci. Comput.* **34**, 499–523.

I. Livshits (2008), 'One-dimensional algorithm for finding eigenbasis of the Schrödinger operator', *SIAM J. Sci. Comput.* **30**, 416–440.

M. Luby (1986), 'A simple parallel algorithm for the maximal independent set problem', *SIAM J. Comput.* **15**, 1036–1053.

S. MacLachlan and L. Olson (2014), 'Theoretical bounds for algebraic multigrid performance: Review and analysis', *Numer. Linear Algebra Appl.* **21**, 194–220.

S. MacLachlan and Y. Saad (2007), 'A greedy strategy for coarse-grid selection', *SIAM J. Sci. Comput.* **29**, 1825–1853.

S. MacLachlan, T. Manteuffel and S. McCormick (2006), 'Adaptive reduction-based AMG', *Numer. Linear Algebra Appl.* **13**, 599–620.

S. MacLachlan, J. Moulton and T. Chartier (2012), 'Robust and adaptive multigrid methods: Comparing structured and algebraic approaches', *Numer. Linear Algebra Appl.* **19**, 389–413.

J.-F. Maitre and F. Musy (1983), 'Méthodes multigrilles: Opérateur associé et estimations du facteur de convergence; le cas du V-cycle', *CR Acad. Sci. Paris Sér. I Math.* **296**, 521–524.

J. Mandel (1988), 'Algebraic study of multigrid methods for symmetric, definite problems', *Appl. Math. Comput.* **25**, 39–56.

J. Mandel, M. Brezina and P. Vaněk (1999), 'Energy optimization of algebraic multigrid bases', *Computing* **62**, 205–228.

T. Manteuffel, S. McCormick, M. Park and J. Ruge (2010), 'Operator-based interpolation for bootstrap algebraic multigrid', *Numerical Linear Algebra Appl.* **17**, 519–537.

I. Marek (1991), Aggregation methods of computing stationary distributions of Markov processes. In *Numerical Treatment of Eigenvalue Problems | Numerische Behandlung von Eigenwertaufgaben*, Vol. 5, Springer, pp. 155–169.

A. Matsokin and S. Nepomnyashchikh (1985), 'The Schwarz alternation method in a subspace', *Izv. Vyssh. Uchebn. Zaved. Mat.* **85**, 61–66.

S. McCormick (1982), 'An algebraic interpretation of multigrid methods', *SIAM J. Numer. Anal.* **19**, 548–560.

S. McCormick (1984), 'Multigrid methods for variational problems: Further results', *SIAM J. Numer. Anal.* **21**, 255–263.

S. McCormick (1985), 'Multigrid methods for variational problems: General theory for the V-cycle', *SIAM J. Numer. Anal.* **22**, 634–643.

S. McCormick and J. Ruge (1982), 'Multigrid methods for variational problems', *SIAM J. Numer. Anal.* **19**, 924–929.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

S. McCormick and J. Ruge (1989), Algebraic multigrid methods applied to problems in computational structural mechanics. Technical report 19900060129, NASA.

S. Míka and P. Vaněk (1992*a*), 'Acceleration of convergence of a two level algebraic algorithm by aggregation in smoothing process', *Appl. Math.* **37**, 343–356.

S. Míka and P. Vaněk (1992*b*), 'A modification of the two-level algorithm with overcorrection', *Appl. Math.* **37**, 13–28.

A. Nägel, R. Falgout and G. Wittum (2008), 'Filtering algebraic multigrid and adaptive strategies', *Comput. Vis. Sci.* **11**, 159–167.

A. Napov and Y. Notay (2012), 'An algebraic multigrid method with guaranteed convergence rate', *SIAM J. Sci. Comput.* **34**, A1079–A1109.

R. Nicolaides (1975), 'On multiple grid and related techniques for solving discrete elliptic systems', *J. Comput. Phys.* **19**, 418–431.

R. Nicolaides (1977), 'On the l^2 convergence of an algorithm for solving finite element equations', *Math. Comp.* **31**(140), 892–906.

Y. Notay (2000), 'Flexible conjugate gradients', *SIAM J. Sci. Comput.* **22**, 1444–1460.

Y. Notay (2010), 'An aggregation-based algebraic multigrid method', *Electron. Trans. Numer. Anal.* **37**, 123–146.

Y. Notay (2012), 'Aggregation-based algebraic multigrid for convection–diffusion equations', *SIAM J. Sci. Comput.* **34**, A2288–A2316.

Y. Notay and P. Vassilevski (2008), 'Recursive Krylov-based multigrid cycles', *Numer. Linear Algebra Appl.* **15**, 473–487.

L. Olson, J. Schroder and R. Tuminaro (2011), 'A general interpolation strategy for algebraic multigrid using energy minimization', *SIAM J. Sci. Comput.* **33**, 966–991.

P. Oswald (1999), 'On the robustness of the BPX-preconditioner with respect to jumps in the coefficients', *Math. Comp.* **68**(226), 633–650.

M. Reed and B. Simon (1978), *Methods of Modern Mathematical Physcis*, Vol. IV: *Analysis of Operators*, Elsevier.

A. Reusken (1993), 'Multigrid with matrix-dependent transfer operators for a singular perturbation problem', *Computing* **50**, 199–211.

J. Ruge (1983), Algebraic multigrid (AMG) for geodetic survey problems. In *Preliminary Proc. Internat. Multigrid Conference, Fort Collins, CO.*

J. Ruge (1985), Final report on AMG02. Report, Gesellschaft für Mathematik und Datenverarbeitung, St. Augustin, GMD contract 5110, 022090.

J. Ruge (1986), 'AMG for problems of elasticity', *Appl. Math. Comput.* **19**, 293–309.

J. Ruge and K. Stüben (1987), Algebraic multigrid (AMG). In *Multigrid Methods* (S. McCormick, ed.), Vol. 3 of Frontiers in Applied Mathematics, SIAM, pp. 73–130.

Y. Saad (2003), *Iterative Methods for Sparse Linear Systems*, second edition, SIAM.

J. Schroder (2012), 'Smoothed aggregation solvers for anisotropic diffusion', *Numer. Linear Algebra Appl.* **19**, 296–312.

B. Smith, P. Bjørstad and W. Gropp (1996), *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

R. Southwell (1940), *Relaxation Methods in Engineering Science: A Treatise on Approximate Computation*, Oxford Engineering Science Series, Oxford University Press.

R. Southwell (1946), *Relaxation Methods in Theoretical Physics*, Clarendon Press, Oxford University Press.

N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein and R. Scheichl (2014), 'Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps', *Numer. Math.* **126**, 741–770.

H. Sterck, U. Yang and J. Heys (2006), 'Reducing complexity in parallel algebraic multigrid preconditioners', *SIAM J. Matrix Anal. Appl.* **27**, 1019–1039.

K. Stüben (1983), 'Algebraic multigrid (AMG): Experiences and comparisons', *Appl. Math. Comput.* **13**, 419–451.

A. Toselli and O. Widlund (2005), *Domain Decomposition Methods: Algorithms and Theory*, Vol. 34 of Springer Series in Computational Mathematics, Springer.

E. Treister and I. Yavneh (2015), 'Non-Galerkin multigrid based on sparsified smoothed aggregation', *SIAM J. Sci. Comput.* **37**, A30–A54.

U. Trottenberg, C. Oosterlee and A. Schüller (2001), *Multigrid*, Academic Press.

I. Vakhutinsky, L. Dudkin and A. Ryvkin (1979), 'Iterative aggregation: A new approach to the solution of large-scale problems', *Econometrica* **47**, 821–841.

P. Vaněk, J. Mandel and M. Brezina (1996), 'Algebraic multigrid based on smoothed aggregation for second and fourth order problems', *Computing* **56**, 179–196.

P. Vaněk, J. Mandel and M. Brezina (1998), 'Convergence of algebraic multigrid based on smoothed aggregation', *Computing* **56**, 179–196.

R. Varga (2000), *Matrix Iterative Analysis*, expanded edition, Vol. 27 of Springer Series in Computational Mathematics, Springer.

P. Vassilevski (2008), *Multilevel Block Factorization Preconditioners: Matrix-based Analysis and Algorithms for Solving Finite Element Equations*, Springer.

P. Vassilevski and L. Zikatanov (2006), 'Multiple vector preserving interpolation mappings in algebraic multigrid', *SIAM J. Matrix Anal. Appl.* **27**, 1040–1055.

C. Wagner and G. Wittum (1997), 'Adaptive filtering', *Numer. Math.* **78**, 305–328.

W. Wan (1998), Scalable and multilevel iterative methods. PhD thesis, University of California, Los Angeles.

W. Wan, T. Chan and B. Smith (1999/2000), 'An energy-minimizing interpolation for robust multigrid methods', *SIAM J. Sci. Comput.* **21**, 1632–1649.

W. Weiler and G. Wittum (1997), 'Parallel frequency filtering', *Computing* **58**, 303–316.

H. Weyl (1911), 'Über die asymptotische Verteilung der Eigenwerte', *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch–Physikalische Klasse* **1911**, 110–117.

H. Weyl (1912), 'Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen', *Math. Ann.* **71**, 441–479.

O. Widlund (1994), Exotic coarse spaces for Schwarz methods for lower order and spectral finite elements. In *Domain Decomposition Methods in Scientific and Engineering Computing*, Vol. 180 of Contemporary Mathematics, AMS, pp. 131–136.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press

O. Widlund (2009), The development of coarse spaces for domain decomposition algorithms. In *Domain Decomposition Methods in Science and Engineering XVIII*, Vol. 70 of Lecture Notes in Computational Science and Engineering, Springer, pp. 241–248.

G. Wittum (1992), *Filternde Zerlegungen: Schnelle Löser für große Gleichungssysteme*, Teubner Skripten zur Numerik, Teubner.

J. Xu (1989), *Theory of Multilevel Methods*, Cornell University.

J. Xu (1991), 'Counterexamples concerning a weighted L^2 projection', *Math. Comp.* **57**(196), 563–568.

J. Xu (1992), 'Iterative methods by space decomposition and subspace correction', *SIAM Rev.* **34**, 581–613.

J. Xu (1996), 'The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids', *Computing* **56**, 215–235.

J. Xu (1997), An introduction to multilevel methods. In *Wavelets, Multilevel Methods, and Elliptic PDEs* (M. Ainsworth *et al.*, eds), Numerical Mathematics and Scientific Computation, Oxford University Press, pp. 213–302.

J. Xu (2009), Optimal algorithms for discretized partial differential equations. In *ICIAM 07: 6th International Congress on Industrial and Applied Mathematics*, European Mathematical Society, pp. 409–444.

J. Xu (2016), *Multilevel Iterative Methods*. Lecture notes, 1989–2016, The Pennsylvania State University.

J. Xu and L. Zikatanov (2002), 'The method of alternating projections and the method of subspace corrections in Hilbert space', *J. Amer. Math. Soc.* **15**, 573–597.

J. Xu and L. Zikatanov (2004), 'On an energy minimizing basis for algebraic multigrid methods', *Comput. Vis. Sci.* **7**, 121–127.

J. Xu and J. Zou (1998), 'Some nonoverlapping domain decomposition methods', *SIAM Rev.* **40**, 857–914.

J. Xu, H. Zhang and L. Zikatanov (2017*a*), Obtaining optimal coarse spaces for AMG via trace minimization. Submitted to *J. Comput. Math.*

J. Xu, H. Zhang and L. Zikatanov (2017*b*), On the Weyl's law for discretized elliptic operators. Submitted to *Math. Comp.*

J. Xu, H. Zhang and L. Zikatanov (2017*c*), A unified approach to the design and analysis of AMG. Submitted to *Numer. Math.*

U. Yang (2006), *Parallel Algebraic Multigrid Methods: High Performance Preconditioners*, Springer.

D. Young (1971), *Iterative Solution of Large Linear Systems*, Academic Press.

H. Yserentant (1993), Old and new convergence proofs for multigrid methods. In *Acta Numerica*, Vol. 2, Cambridge University Press, pp. 285–326.

G. Yu, J. Xu and L. Zikatanov (2013), 'Analysis of a two-level method for anisotropic diffusion equations on aligned and nonaligned grids', *Numer. Linear Algebra Appl.* **20**, 832–851.

L. Zikatanov (2008), 'Two-sided bounds on the convergence rate of two-level methods', *Numer. Linear Algebra Appl.* **15**, 439–454.

https://doi.org/10.1017/S0962492917000083 Published online by Cambridge University Press