Optimal V cycle algebraic multilevel preconditioning

Y. Notay [1]

Service de Métrologie Nucléaire
Université Libre de Bruxelles (C.P. 165)
50, Av. F.D. Roosevelt, B-1050 Brussels, Belgium.
email : ynotay@ulb.ac.be

Report GANMN 97-05

November 1997

[1]Supported by the "Fonds National de la Recherche Scientifique", Chercheur qualifié.

Abstract

We consider algebraic multilevel preconditioning methods based on the recursive use of a 2×2 block incomplete factorization procedure in which the Schur complement is approximated by a coarse grid matrix. As is well known, for discrete second order elliptic PDEs, optimal convergence properties are proved for such basic two-level schemes under mild assumptions on the PDE coefficients, but their recursive use in a simple V cycle algorithm does generally not lead to optimal order convergence.

In the present paper, we analyze the combination of these techniques with a smoothing procedure much similar to the one used in standard multigrid algorithms, except that smoothing is not required on the finest grid. Theoretical results prove optimal convergence properties for the V cycle under an assumption similar to the "approximation property" of the classical multigrid convergence theory. On the other hand, numerical experiments show that, for both 2D and 3D problems, using rather simple smoothing strategies suffices to make the condition number close to that of the two-level method.

keywords: iterative methods for linear systems, acceleration of convergence preconditioning.

AMS CLASSIFICATION : 65F10, 65B99, 65N20.

1 Introduction

This paper deals with the iterative solution of large sparse symmetric positive (nonnegative) definite linear systems

$$A\mathbf{u} = \mathbf{b} \tag{1.1}$$

arising from the discretization of second order elliptic PDEs. In such cases, the conjugate gradient method combined with a suitable preconditioning is a choice method (e.g. [4, 5, 16, 19]). The preconditioner is here a symmetric positive definite matrix B such that solving a system with B is easy whereas the (spectral) condition number of the preconditioned system

$$\kappa(B^{-1}A) \;=\; \frac{\lambda_{\max}(B^{-1}A)}{\lambda_{\min}(B^{-1}A)} \;,$$

on which depends the convergence rate, has to be as small as possible[1]. In particular, a preconditioner has an optimal order of computational complexity when

- solving a system with B requires $\mathcal{O}(n)$ operations, where n is the number of unknowns;
- $\kappa(B^{-1}A)$ is bounded independently of the grid size and possibly other problem dependent parameters.

To meet both these requirements, many recent works focus on the design algebraic multilevel preconditioners (e.g. [3, 6, 7, 10, 11, 12, 28, 29, 31, 32, 33, 34]). Like standard multigrid [20, 35], these are based on the recursive use of a two-level method. However, the basic scheme originates from a block incomplete factorization of the system matrix partitioned in a 2×2 block form

$$A \;=\; \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \;, \tag{1.2}$$

where the first block of unknowns corresponds to the fine grid nodes and the second block of unknowns to the coarse grid nodes.

For such two-level methods, eigenvalue bounds are generally proved by means of local analyzes, that is, in the finite element context, by computing certain quantities associated to element matrices. Consequently, these bounds hold independently of the grid size and of possible jumps in the PDE coefficients, as long as these may only occur across the edges of the coarse mesh (e.g. [2, 8, 22, 25, 33]).

Hence, algebraic multilevel methods exhibit a strong potential for robustness. Unfortunately, see [33], their recursive use in a simple V cycle algorithm does generally not result in an optimal order preconditioning. Moreover, so far proposed W cycle schemes (e.g. [3, 6, 7, 10, 11, 12, 31, 32, 33]) require that some parameters are determined in function of eigenvalue estimates. In some cases, one can rely on theoretical results. However, as inaccurate estimates may lead to poor performances or even to an indefinite preconditioner, a better robustness is achieved with run time eigenvalue estimation. Relevant procedures are proposed and discussed in [10], their main drawback being the additional implementation effort required and an increase of the cost of the preprocessing step.

[1] throughout this paper, $\lambda_{\max}(C)$ and $\lambda_{\min}(C)$ denote respectively the largest and the smallest eigenvalue of C

Here, we develop another approach, in which one adds to a simple V cycle some smoothing iterations in a much similar way as it is done in standard multigrid methods, except that smoothing is not required on the finest grid. As will be seen, this allows to make the condition number independent of the number of levels and close to that of the two-level preconditioner. Furthermore, this stabilization is already obtained with rather simple smoothing strategies. For instance, one damped Jacobi pre- and post-smoothing step suffice for 2D and 3D isotropic problems, allowing us to propose a variant well suited to parallel computation.

The remaining of this paper is organized as follows: in Section 2, we present algebraic multilevel preconditioning methods and motivate our approach to stabilize the V cycle; the theoretical analysis is developed in Section 3 and some practical issues are discussed in Section 4; Section 5 is devoted to numerical results.

2 Algebraic multilevel preconditioning and coarse grid correction

An exact factorization of the system matrix (1.2) writes

$$A = \begin{pmatrix} A_{11} & \\ A_{21} & S \end{pmatrix} \begin{pmatrix} I & A_{11}^{-1} A_{12} \\ & I \end{pmatrix} , \tag{2.1}$$

where $S = A_{22} - A_{21} A_{11}^{-1} A_{12}$ is the Schur complement. This leads to a direct method, which is impractical because S is generally dense. However, it is most often easy to find a sparse approximation A_c to S that has a similar structure to that of A. One then obtains the following two-level block incomplete factorization preconditioner:

$$B_{(2)} = \begin{pmatrix} A_{11} & \\ A_{21} & A_c \end{pmatrix} \begin{pmatrix} I & A_{11}^{-1} A_{12} \\ & I \end{pmatrix} . \tag{2.2}$$

Initially, this methods was proposed for finite element matrices computed with the hierarchical basis [2, 8, 11, 12, 31, 32, 33], generalizing in this way the hierarchical basis multigrid method [13, 14, 15, 37]. In this context, one uses $A_c = A_{22}^{(hb)}$, which corresponds to the finite element discretization on the coarse mesh.

However, as observed in [31], elementary computation show that, for any $B_{(2)}$ of the form (2.2),

$$B_{(2)}^{-1} A = \begin{pmatrix} I & * \\ & A_c^{-1} S \end{pmatrix} , \tag{2.3}$$

where the Schur complement S is in addition invariant under a basis transformation of the form

$$T = \begin{pmatrix} I & J_{12} \\ & I \end{pmatrix} , \tag{2.4}$$

like the one associated to the change from the hierarchical basis to the nodal basis. Consequently, identical convergence results are obtained with a block incomplete factorization (2.2) performed in the usual nodal basis. Since $A^{(hb)} = T^t A T$, one has then

$$A_c = A_{22}^{(hb)} = A_{22} + A_{21} J_{12} + J_{12}^t A_{12} + J_{12}^t A_{11} J_{12} , \tag{2.5}$$

or, equivalently,

$$A_c = r\,A\,p\,, \tag{2.6}$$

where

$$p = \begin{pmatrix} J_{12} \\ I \end{pmatrix}\,, \qquad r = p^t\,. \tag{2.7}$$

A_c is thus the "Galerkin approximation" corresponding to the prolongation p. This allows to extend the method outside the finite element context by setting an appropriate prolongation operator. Alternative, more algebraic techniques to compute A_c are also proposed in [25, 28].

Whatever the choice of A_c, the key features of the preconditioner (2.2) are twofold. On the one hand, see (2.3), the associate condition number and convergence rate will be independent of the mesh size if and only if A_c is spectrally equivalent to S, which is proved by means of local analyzes, thus independently of possible jumps in the PDE coefficients as long as these may only occur across the edges of the coarse mesh (e.g. [2, 8, 22, 25, 33]).

On the other hand, A_{11} has condition number $\mathcal{O}(1)$ [2, 8, 14], so that the two required solves of a system with A_{11} can be done in a number of operations proportional to the number of unknowns in block 1. In practice, these solves are most often made only approximately, i.e. A_{11} has to be exchanged in the formulation (2.2) for some preconditioner P. The proper way to do this depends whether or not one is working with the hierarchical basis [4, 8, 26]. This will be further discussed in Section 4. For simplicity, here and in the following section, we keep the "theoretical" formulation (2.2).

Now,

$$B_{(2)}^{-1} = \begin{pmatrix} I & -A_{11}^{-1}A_{12} \\ & I \end{pmatrix} \begin{pmatrix} A_{11}^{-1} & \\ & A_c^{-1} \end{pmatrix} \begin{pmatrix} I & \\ -A_{21}A_{11}^{-1} & I \end{pmatrix} \tag{2.8}$$

may be rewritten

$$B_{(2)}^{-1} = \begin{pmatrix} A_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + p_A\,A_c^{-1}\,r_A\,, \tag{2.9}$$

where

$$p_A = \begin{pmatrix} -A_{11}^{-1}A_{12} \\ I \end{pmatrix}\,, \qquad r_A = \begin{pmatrix} -A_{21}A_{11}^{-1} & I \end{pmatrix}\,. \tag{2.10}$$

Thus, considering p_A as a prolongation operator, the second term of the right hand side of (2.9) is similar to a coarse grid correction term as used in multigrid methods. Further, using p_A is not common in the multigrid context (except when reasoning on the one dimensional case [21]), but we shall see in Section 4 that, when A_{11}^{-1} is properly exchanged for the inverse of some preconditioner, then p_A resembles a matrix dependent prolongation as developed in [1, 18].

These observations led us to investigate the combined use of algebraic multilevel preconditioning with a smoothing procedure. At first sight, this is not a clever approach, since smoothing is costly on the finest grid whereas the preconditioner (2.2) has already grid independence convergence. But, in the context of the recursive use of the two-level method, smoothing can help to stabilize the condition number, multigrid schemes being usually optimal with a simple V cycle [20]. On the other hand, such a stabilization does not require smoothing on the finest grid, so that the extra cost is moderate and in fact smaller than that involved by a W cycle (see Section 4).

3 Theoretical analysis

First, we need to formulate the multilevel method derived from the two-level preconditioner (2.2). If ℓ levels are considered, we let the system matrix

$$A \;=\; A^{(\ell)} \;=\; \begin{pmatrix} A_{11}^{(\ell)} & A_{12}^{(\ell)} \\ A_{21}^{(\ell)} & A_{22}^{(\ell)} \end{pmatrix} , \tag{3.1}$$

correspond to level ℓ, and, for $k = \ell, \dots, 2$, we partition $A^{(k)}$ similarly to $A^{(\ell)}$ and set

$$A^{(k-1)} \;=\; A_c^{(k)} , \tag{3.2}$$

where $A_c^{(k)}$ is the approximation of the Schur complement of $A^{(k)}$.

The preconditioner $B^{(\ell)}$ of $A^{(\ell)}$ is then defined by the recursion

$$B^{(k)} \;=\; \begin{pmatrix} A_{11}^{(k)} & \\ A_{21}^{(k)} & M^{(k-1)^{-1}} \end{pmatrix} \begin{pmatrix} I & A_{11}^{(k)^{-1}} A_{12}^{(k)} \\ & I \end{pmatrix} , \qquad k = 2, \dots, \ell , \tag{3.3}$$

where

$$M^{(1)} \;=\; A^{(1)^{-1}} , \tag{3.4}$$

and where $M^{(k)}$, $k = 2, \dots, \ell - 1$, is an approximate inverse of $A^{(k)}$ defined in function of $B^{(k)^{-1}}$.

Using $M^{(k)} = B^{(k)^{-1}}$ leads to a simple V cycle which is usually not optimal [33]. Previous works on algebraic multilevel preconditioners [3, 6, 7, 10, 11, 12, 31, 32, 33] consider then $M^{(k)} = \mathcal{P}_\nu(B^{(k)^{-1}}A^{(k)})A^{(k)^{-1}}$ where $\mathcal{P}_\nu(t)$ is a ν degree polynomial such that $\mathcal{P}_\nu(0) = 0$ and whose remaining coefficients are determined in function of eigenvalue estimates for $B^{(k)^{-1}}A^{(k)}$.

Here, we consider $M^{(k)}$ defined by

$$\left(I - M^{(k)} A^{(k)}\right) \;=\; \left(I - R^{(k)} A^{(k)}\right) \left(I - B^{(k)^{-1}} A^{(k)}\right) \left(I - R^{(k)} A^{(k)}\right) , \tag{3.5}$$

where $R^{(k)}$ is a (symmetric) relaxation operator for $A^{(k)}$. $R^{(k)}$ may correspond to several smoothing iterations, but, as will be seen, satisfactory results are obtained with rather simple smoothing schemes. Note that

$$x^{(k)} \;=\; M^{(k)} y^{(k)}$$

is easily computed with

$$\begin{array}{rcllrcl} x_{(1)}^{(k)} & = & R^{(k)} y^{(k)} & ; & y_{(1)}^{(k)} & = & y^{(k)} - A^{(k)} x_{(1)}^{(k)} \;; \\ x_{(2)}^{(k)} & = & B^{(k)^{-1}} y_{(1)}^{(k)} & ; & y_{(2)}^{(k)} & = & y_{(1)}^{(k)} - A^{(k)} x_{(2)}^{(k)} \;; \\ x_{(3)}^{(k)} & = & R^{(k)} y_{(2)}^{(k)} & ; & & & \\ x^{(k)} & = & x_{(1)}^{(k)} + x_{(2)}^{(k)} + x_{(3)}^{(k)} & . & & & \end{array}$$

The basic properties of $M^{(k)}$ are proved in the following Lemma.

Lemma 3.1 *Let $M^{(k)}$ be defined by (3.5) for given symmetric positive matrices $A^{(k)}$, $B^{(k)}$, $R^{(k)}$.*

If $\rho\left(I - R^{(k)} A^{(k)}\right) \le 1$, then $M^{(k)}$ is symmetric positive definite and satisfies

$$\lambda_{\min}\left(M^{(k)} A^{(k)}\right) \;\ge\; \min\left(1 \,,\, \lambda_{\min}\left(B^{(k)^{-1}} A^{(k)}\right)\right) , \tag{3.6}$$

$$\lambda_{\max}\left(M^{(k)} A^{(k)}\right) \;\le\; \max\left(1 \,,\, \lambda_{\max}\left(B^{(k)^{-1}} A^{(k)}\right)\right) . \tag{3.7}$$

Proof Letting $A^{(k)} = L^{(k)}\,U^{(k)}$ be a symmetric factorization of $A^{(k)}$ and defining

$$\begin{aligned}
\widetilde{M}^{(k)} &= U^{(k)}\,M^{(k)}\,L^{(k)} \ , \\
\widetilde{R}^{(k)} &= U^{(k)}\,R^{(k)}\,L^{(k)} \ , \\
\widetilde{B}^{(k)\,-1} &= U^{(k)}\,B^{(k)^{-1}}\,L^{(k)} \ ,
\end{aligned}$$

one obtains

$$\widetilde{M}^{(k)} = I + \left(I - \widetilde{R}^{(k)}\right)\left(\widetilde{B}^{(k)\,-1} - I\right)\left(I - \widetilde{R}^{(k)}\right) \ ,$$

from which the symmetry of $M^{(k)}$ is obvious.

Further, let $\mathbf{v}$ be any vector and set $\mathbf{w} = \left(I - \widetilde{R}^{(k)}\right)\mathbf{v}$. One has

$$\frac{\mathbf{v}^t\,\widetilde{M}^{(k)}\,\mathbf{v}}{\mathbf{v}^t\,\mathbf{v}} = 1 + \frac{\mathbf{w}^t(\,\widetilde{B}^{(k)\,-1} - I)\mathbf{w}}{\mathbf{w}^t\,\mathbf{w}}\;\frac{\mathbf{w}^t\,\mathbf{w}}{\mathbf{v}^t\,\mathbf{v}} \ ,$$

whereas $\rho\left(I - R^{(k)}\,A^{(k)}\right) \le 1$ implies $\frac{\mathbf{w}^t\,\mathbf{w}}{\mathbf{v}^t\,\mathbf{v}} \le 1$. Hence,

$$\begin{aligned}
\lambda_{\min}\left(M^{(k)}\,A^{(k)}\right) &= \min_{\mathbf{v}} \frac{\mathbf{v}^t\,\widetilde{M}^{(k)}\,\mathbf{v}}{\mathbf{v}^t\,\mathbf{v}} \\
&\ge 1 + \min\left(0\ ,\ \min_{\mathbf{w}} \frac{\mathbf{w}^t(\,\widetilde{B}^{(k)\,-1} - I)\mathbf{w}}{\mathbf{w}^t\,\mathbf{w}}\right) \ ,
\end{aligned}$$

i.e. (3.6) and

$$\begin{aligned}
\lambda_{\max}\left(M^{(k)}\,A^{(k)}\right) &= \max_{\mathbf{v}} \frac{\mathbf{v}^t\,\widetilde{M}^{(k)}\,\mathbf{v}}{\mathbf{v}^t\,\mathbf{v}} \\
&\le 1 + \max\left(0\ ,\ \max_{\mathbf{w}} \frac{\mathbf{w}^t(\,\widetilde{B}^{(k)\,-1} - I)\mathbf{w}}{\mathbf{w}^t\,\mathbf{w}}\right) \ ,
\end{aligned}$$

i.e. (3.7). Note that (3.6) proves also the positive definiteness of $M^{(k)}$. ■

By induction, this Lemma ensures the positive definiteness of $B^{(\ell)}$ if the splittings associated to $R^{(k)}$ are convergent for all k. Further, inequalities (3.6), (3.7) show that smoothing cannot have an adverse effect on the eigenvalue distribution.

Proving a beneficial effect is of course harder. From (2.3), one readily deduces, assuming $\lambda_{\min}(M^{(\ell-1)}S^{(\ell)}) \le 1 \le \lambda_{\max}(M^{(\ell-1)}S^{(\ell)})$,

$$\begin{aligned}
\lambda_{\min}\left(B^{(\ell)^{-1}}\,A^{(\ell)}\right) &= \lambda_{\min}\left(M^{(\ell-1)}\,S^{(\ell)}\right) \\
&\ge \lambda_{\min}\left(M^{(\ell-1)}\,A^{(\ell-1)}\right)\cdot\lambda_{\min}\left(A^{(\ell-1)^{-1}}\,S^{(\ell)}\right) \ , \qquad (3.8) \\
\lambda_{\max}\left(B^{(\ell)^{-1}}\,A^{(\ell)}\right) &= \lambda_{\max}\left(M^{(\ell-1)}\,S^{(\ell)}\right) \\
&\le \lambda_{\max}\left(M^{(\ell-1)}\,A^{(\ell-1)}\right)\cdot\lambda_{\max}\left(A^{(\ell-1)^{-1}}\,S^{(\ell)}\right) \ . \qquad (3.9)
\end{aligned}$$

Hence, the convergence rate will approach that of the two-level method if $\kappa(M^{(\ell-1)}A^{(\ell-1)})$ is close to one. To analyze this, we resort to some arguments of the multigrid convergence theory. More particularly, the following theorem rewrites and adapts to our context Hackbush result for the V cycle in the symmetric positive definite case [20, Lemma 7.2.1 and Theorem 7.2.2].

Theorem 3.1 *Let $M^{(k)}$, $k=1,\ldots,\ell-1$ be defined by (3.4),(3.5), where $A^{(k)}$ and $R^{(k)}$ are symmetric positive definite matrices and where $B^{(k)}$, $k=2,\ldots,\ell-1$ is defined by (3.3) for some 2×2 partitioning of $A^{(k)}$. Assume that*

$$\lambda_{\max}\left(R^{(k)}\,A^{(k)}\right) \;\le\; 1 \qquad k=2,\ldots,\ell-1\ . \tag{3.10}$$

If, for some positive constants c_A, C_A such that $c_A<2$, one has, for $k=2,\ldots,\ell-1$,

$$-\,c_A\ \mathbf{v}^t\,R^{(k)}\,\mathbf{v} \;\le\; \mathbf{v}^t\left(A^{(k)^{-1}}-B_{(2)}^{(k)^{-1}}\right)\mathbf{v} \;\le\; C_A\ \mathbf{v}^t\,R^{(k)}\,\mathbf{v} \qquad \forall \mathbf{v}\ , \tag{3.11}$$

where

$$B_{(2)}^{(k)} \;=\; \begin{pmatrix} A_{11}^{(k)} & \\ A_{21}^{(k)} & A^{(k-1)} \end{pmatrix}\begin{pmatrix} I & A_{11}^{(k)^{-1}}A_{12}^{(k)} \\ & I \end{pmatrix}\ , \tag{3.12}$$

then, for $k=1,\ldots,\ell-1$,

$$\lambda_{\min}\left(M^{(k)}\,A^{(k)}\right) \;\ge\; \frac{2}{2+C_A}\ , \tag{3.13}$$

$$\lambda_{\max}\left(M^{(k)}\,A^{(k)}\right) \;\le\; \frac{2}{2-c_A}\ . \tag{3.14}$$

Proof Obviously, (3.13), (3.14) hold for $k=1$ and we will show by induction that they hold for any k. In this view, let

$$U^{(k)} \;=\; \begin{pmatrix} A_{11}^{(k)^{\frac12}} & \\ & S^{(k)^{\frac12}} \end{pmatrix}\begin{pmatrix} I & A_{11}^{(k)^{-1}}A_{12}^{(k)} \\ & I \end{pmatrix}\ ,$$

where $S^{(k)}$ is the Schur complement of $A^{(k)}$. Note that $A^{(k)}=L^{(k)}\,U^{(k)}$ where $L^{(k)}=U^{(k)^t}$. Let further

$$\begin{aligned}
\widetilde{M}^{(k)} &= U^{(k)}\,M^{(k)}\,L^{(k)}\ ,\\
\widetilde{R}^{(k)} &= U^{(k)}\,R^{(k)}\,L^{(k)}\ ,\\
\widetilde{B}^{(k)\,-1} &= U^{(k)}\,B^{(k)^{-1}}\,L^{(k)}\ ,\\
\widetilde{B}_{(2)}^{(k)\,-1} &= U^{(k)}\,B_{(2)}^{(k)^{-1}}\,L^{(k)}\ .
\end{aligned}$$

One has

$$\left(I-\widetilde{M}^{(k)}\right) \;=\; \left(I-\widetilde{R}^{(k)}\right)\left(I-\widetilde{B}^{(k)\,-1}\right)\left(I-\widetilde{R}^{(k)}\right)\ .$$

On the other hand, see (2.8),

$$\widetilde{B}^{(k)\,-1} \;=\; \begin{pmatrix} I & \\ & S^{(k)^{\frac12}}\,M^{(k-1)}\,S^{(k)^{\frac12}} \end{pmatrix}\ ,$$

$$\widetilde{B}_{(2)}^{(k)\,-1} \;=\; \begin{pmatrix} I & \\ & S^{(k)^{\frac12}}\,A^{(k-1)^{-1}}\,S^{(k)^{\frac12}} \end{pmatrix}\ ,$$

whence, by the induction argument,

$$\tfrac{2}{2+C_A}\ \mathbf{v}^t\ \widetilde{B}_{(2)}^{(k)\,-1}\ \mathbf{v} \;\le\; \mathbf{v}^t\ \widetilde{B}^{(k)\,-1}\ \mathbf{v} \;\le\; \tfrac{2}{2-c_A}\ \mathbf{v}^t\ \widetilde{B}_{(2)}^{(k)\,-1}\ \mathbf{v} \qquad \forall \mathbf{v}\ . \tag{3.15}$$

The right inequality implies

$$\mathbf{v}^t\left(I-\widetilde{M}^{(k)}\right)\mathbf{v} \;\geq\; \mathbf{v}^t\;\left(I-\widetilde{R}^{(k)}\right)\;\left(I-\tfrac{2}{2-c_A}\;\widetilde{B}_{(2)}^{(k)^{-1}}\right)\;\left(I-\widetilde{R}^{(k)}\right)\;\mathbf{v} \qquad \forall\mathbf{v}\;,$$

which gives, using the left inequality (3.11),

$$\mathbf{v}^t\left(I-\widetilde{M}^{(k)}\right)\mathbf{v} \;\geq\; \mathbf{v}^t\;\left(I-\widetilde{R}^{(k)}\right)\;\left(\tfrac{-c_A}{2-c_A}I-\tfrac{2c_A}{2-c_A}\widetilde{R}^{(k)}\right)\;\left(I-\widetilde{R}^{(k)}\right)\;\mathbf{v} \qquad \forall\mathbf{v}\;.$$

Therefore,

$$\begin{aligned}\lambda_{\max}\left(M^{(k)}\,A^{(k)}\right) &= \lambda_{\max}\left(\widetilde{M}^{(k)}\right)\\ &\leq 1\;+\;\frac{c_A}{2-c_A}\;\max_{0\leq\xi\leq 1}(1-\xi)^2(1+2\xi)\;,\end{aligned}$$

whence (3.14) since the maximum corresponds to $\xi=0$.

Similarly, the left inequality (3.15) implies

$$\mathbf{v}^t\left(I-\widetilde{M}^{(k)}\right)\mathbf{v} \;\leq\; \mathbf{v}^t\;\left(I-\widetilde{R}^{(k)}\right)\;\left(I-\tfrac{2}{2+C_A}\;\widetilde{B}_{(2)}^{(k)^{-1}}\right)\;\left(I-\widetilde{R}^{(k)}\right)\;\mathbf{v} \qquad \forall\mathbf{v}\;.$$

which gives, using the right inequality (3.11),

$$\mathbf{v}^t\left(I-\widetilde{M}^{(k)}\right)\mathbf{v} \;\leq\; \mathbf{v}^t\;\left(I-\widetilde{R}^{(k)}\right)\;\left(\tfrac{C_A}{2+C_A}I+\tfrac{2C_A}{2+C_A}\widetilde{R}^{(k)}\right)\;\left(I-\widetilde{R}^{(k)}\right)\;\mathbf{v} \qquad \forall\mathbf{v}\;,$$

whence

$$\begin{aligned}\lambda_{\min}\left(M^{(k)}\,A^{(k)}\right) &= \lambda_{\min}\left(\widetilde{M}^{(k)}\right)\\ &\geq 1\;-\;\frac{C_A}{2+C_A}\;\max_{0\leq\xi\leq 1}(1-\xi)^2(1+2\xi)\;,\end{aligned}$$

i.e. (3.13). ∎

For the method defined in the hierarchical basis with $A^{(k-1)}=A_{22}^{(hb)^{(k)}}$, it is easy to show that $\lambda_{\max}(A^{(k-1)^{-1}}S^{(k)})\leq 1$ [31], whence

$$\mathbf{v}^t\,B_{(2)}^{(k)^{-1}}\,\mathbf{v} \;\geq\; \mathbf{v}^t\,A^{(k)^{-1}}\,\mathbf{v} \qquad \forall\mathbf{v}\;, \tag{3.16}$$

i.e. $c_A=0$ in (3.11). Actually, from (2.3) and the invariance of the Schur complement under a transformation of the form (2.4), this result extends to any method for which

$$A^{(k-1)} \;=\; r^{(k)}\,A^{(k)}\,p^{(k)}\;, \tag{3.17}$$

where $r^{(k)}=p^{(k)^t}$ and where $p^{(k)}$ is a prolongation of the form

$$p^{(k)} \;=\; \begin{pmatrix} J_{12}^{(k)} \\ I \end{pmatrix}\;. \tag{3.18}$$

The analysis of C_A is harder. In fact, the multigrid convergence theory developed in [20] for the symmetric positive definite case makes use of the following similar "approximation property":

$$\mathbf{v}^t\left(A^{(k)^{-1}}-\,p^{(k)}\,A^{(k-1)^{-1}}\,r^{(k)}\right)\mathbf{v} \;\leq\; \overline{C}_A\;\mathbf{v}^t\,R^{(k)}\,\mathbf{v} \qquad \forall\mathbf{v}\;. \tag{3.19}$$

General analyzes of $\overline{C}_A$ are rather involved and extending them to our context lies beyond the scope of the present paper. However, it is possible to rest on the developments in [20] for the proof of the approximation property and derive simple upper bounds for C_A assuming that (3.19) holds. This is the purpose of the the following theorem and lemma, where we address the case of damped Jacobi smoothing ($R^{(k)^{-1}} = \omega\, diag(A^{(k)})$).

Lemma 3.2 *Let $A^{(k)}$ and $A^{(k-1)}$ be symmetric positive definite matrices. Let $D^{(k)} = diag(A^{(k)})$ and $D_2^{(k)} = diag(A_{22}^{(k)})$.*

If, for some $p^{(k)}$ of the form (3.18),

$$\mathbf{v}^t \left(A^{(k)^{-1}} - p^{(k)}\, A^{(k-1)^{-1}}\, r^{(k)}\right) \mathbf{v} \;\le\; \overline{C}_A\, \omega^{-1}\; \mathbf{v}^t\, D^{(k)^{-1}}\, \mathbf{v} \qquad \forall \mathbf{v}\;, \qquad (3.20)$$

where $r^{(k)} = p^{(k)^t}$, then

$$\mathbf{v}_2^t \left(S^{(k)^{-1}} - A^{(k-1)^{-1}}\right) \mathbf{v}_2 \;\le\; \overline{C}_A\, \omega^{-1}\; \mathbf{v}_2^t\, D_2^{(k)^{-1}}\, \mathbf{v}_2 \qquad \forall \mathbf{v}_2\;, \qquad (3.21)$$

where $S^{(k)}$ is the Schur complement of $A^{(k)}$ related to the partitioning induced by $p^{(k)}$.

Proof (3.21) is (3.20) restricted to vectors of the form $\mathbf{v} = \begin{pmatrix} 0 \\ \mathbf{v}_2 \end{pmatrix}$ ■

Theorem 3.2 *Let $A^{(k)}$ be a symmetric positive definite matrix partitioned in a 2×2 block form, let $S^{(k)}$ be the corresponding Schur complement, and let $D^{(k)} = diag(A^{(k)})$, $D_1^{(k)} = diag(A_{11}^{(k)})$, $D_2^{(k)} = diag(A_{22}^{(k)})$. Let $B_{(2)}^{(k)}$ be defined by (3.12), where $A^{(k-1)}$ is symmetric positive definite.*

If

$$\mathbf{v}_2^t \left(S^{(k)^{-1}} - A^{(k-1)^{-1}}\right) \mathbf{v}_2 \;\le\; \omega^{-1}\, \widehat{C}_A\; \mathbf{v}_2^t\, D_2^{(k)^{-1}}\, \mathbf{v}_2 \qquad \forall \mathbf{v}_2\;, \qquad (3.22)$$

then

$$\mathbf{v}^t \left(A^{(k)^{-1}} - B_{(2)}^{(k)^{-1}}\right) \mathbf{v} \;\le\; \omega^{-1}\, C_A\; \mathbf{v}^t\, D^{(k)^{-1}}\, \mathbf{v} \qquad \forall \mathbf{v}\;, \qquad (3.23)$$

where

$$C_A \;\le\; \widehat{C}_A \left(1 \;+\; \lambda_{\min}^{-1}\left(D_1^{(k)^{-1}}\, A_{11}^{(k)}\right) \lambda_{\min}^{-1}\left(D_2^{(k)^{-1}}\, A_{22}^{(k)}\right)\right)\;. \qquad (3.24)$$

Proof Define $U^{(k)}$, $L^{(k)}$, $\widetilde{B}_{(2)}^{(k)}$ as in the proof of Theorem 3.1. For any vector $\mathbf{v}$ and positive number α,

$$\begin{aligned}
\mathbf{v}^t \left(I - \widetilde{B}_{(2)}^{(k)^{-1}}\right) \mathbf{v} \;&=\; \mathbf{v}^t \begin{pmatrix} 0 & 0 \\ 0 & I - S^{(k)^{\frac{1}{2}}}\, A^{(k-1)^{-1}}\, S^{(k)^{\frac{1}{2}}} \end{pmatrix} \mathbf{v} \\
&\le\; \widehat{C}_A\, \omega^{-1}\, \mathbf{v}^t \begin{pmatrix} 0 & 0 \\ 0 & S^{(k)^{\frac{1}{2}}}\, D_2^{(k)^{-1}}\, S^{(k)^{\frac{1}{2}}} \end{pmatrix} \mathbf{v} \\
&\le\; \widehat{C}_A\, \omega^{-1}\, \mathbf{v}^t \begin{pmatrix} \alpha^{-1} I & 0 \\ 0 & S^{(k)^{\frac{1}{2}}}\, D_2^{(k)^{-1}}\, S^{(k)^{\frac{1}{2}}} \end{pmatrix} \mathbf{v} \\
&=\; \widehat{C}_A\, \omega^{-1}\, \mathbf{v}^t\, U^{(k)}\, G_\alpha^{-1}\, L^{(k)}\, \mathbf{v}\;,
\end{aligned}$$

where

$$
\begin{aligned}
G_\alpha &= L^{(k)} \begin{pmatrix} \alpha I & \\ & S^{(k)^{-\frac{1}{2}}} D_2^{(k)} S^{(k)^{-\frac{1}{2}}} \end{pmatrix} U^{(k)} \\
&= \begin{pmatrix} \alpha A_{11}^{(k)} & \alpha A_{12}^{(k)} \\ \alpha A_{21}^{(k)} & D_2^{(k)} + \alpha A_{21}^{(k)} A_{11}^{(k)^{-1}} A_{12}^{(k)} \end{pmatrix} .
\end{aligned}
$$

Therefore, (3.23) holds with $C_A \le \nu^{-1} \widehat{C}_A$ if

$$
\mathbf{v}^t G_\alpha^{-1} \mathbf{v} \;\le\; \nu^{-1} \, \mathbf{v}^t D^{(k)^{-1}} \mathbf{v} \qquad \forall \mathbf{v} ,
$$

or, equivalently, if $G_\alpha - \nu D^{(k)}$ is nonnegative definite. Since $\alpha A_{11}^{(k)} - \nu D_1^{(k)}$ is positive definite for α sufficiently large, $G_\alpha - \nu D^{(k)}$ will be nonnegative definite if and only if so is its Schur complement

$$
S_{G_\alpha - \nu D^{(k)}} \;=\; (1-\nu)\, D_2^{(k)} \;+\; A_{21}^{(k)} \left(\alpha \, A_{11}^{(k)^{-1}} - \alpha^2 \left(\alpha A_{11}^{(k)} - \nu D_1^{(k)} \right)^{-1} \right) A_{12}^{(k)} .
$$

On the other hand, letting $\lambda_2 = \lambda_{\min}(D_2^{(k)^{-1}} A_{22}^{(k)})$,

$$
\mathbf{v}_2^t \, D_2^{(k)} \, \mathbf{v}_2 \;\ge\; \lambda_2 \, \mathbf{v}_2^t \, A_{22}^{(k)} \, \mathbf{v}_2 \;\ge\; \lambda_2 \, \mathbf{v}_2^t \, A_{21}^{(k)} \, A_{11}^{(k)^{-1}} A_{12}^{(k)} \, \mathbf{v}_2 \qquad \forall \mathbf{v}_2 .
$$

Hence, $S_{G_\alpha - \nu D^{(k)}}$ is nonnegative definite if

$$
\left((1-\nu)\lambda_2 + \alpha \right) \mathbf{v}_1^t \, A_{11}^{(k)^{-1}} \mathbf{v}_1 \;\ge\; \alpha^2 \, \mathbf{v}_1^t \left(\alpha A_{11}^{(k)} - \nu D_1^{(k)} \right)^{-1} \mathbf{v}_1 \qquad \forall \mathbf{v}_1 ,
$$

or, equivalently, if

$$
\left((1-\nu)\lambda_2 + \alpha \right) \left(\alpha A_{11}^{(k)} - \nu D_1^{(k)} \right) \;-\; \alpha^2 \, A_{11}^{(k)} \;=\; \alpha \left((1-\nu)\lambda_2 \, A_{11}^{(k)} - \nu \, D_1^{(k)} \right) \;-\; \nu(1-\nu)\lambda_2 \, D_1^{(k)}
$$

is nonnegative definite. Setting

$$
\nu \;=\; \frac{\lambda_1 \, \lambda_2}{1 + \lambda_1 \, \lambda_2} - \frac{\lambda_2}{\alpha} ,
$$

where $\lambda_1 = \lambda_{\min}(D_1^{(k)^{-1}} A_{11}^{(k)})$, the latter matrix writes

$$
\tfrac{\alpha \, \lambda_2}{1 + \lambda_1 \, \lambda_2} \left(A_{11}^{(k)} - \lambda_1 D_1^{(k)} \right) \;+\; \lambda_2^2 \, A_{11}^{(k)} \;+\; \lambda_2 (1 - \nu(1-\nu)) D_1^{(k)} ,
$$

which is clearly nonnegative definite since $\max_t t(1-t) = \frac{1}{2}$. Therefore,

$$
C_A \;\le\; \nu^{-1} \, \widehat{C}_A \;=\; \left(\frac{\lambda_1 \, \lambda_2}{1 + \lambda_1 \, \lambda_2} - \frac{\lambda_2}{\alpha} \right)^{-1} \widehat{C}_A ,
$$

and the proof is completed by taking the limit for α going to the infinity. ■

Note that if $A_{22}^{(k)}$ is diagonal, then $\lambda_{\min}(D_2^{(k)^{-1}} A_{22}^{(k)}) = 1$. On the other hand, $A_{11}^{(k)}$ has condition number $\mathcal{O}(1)$ [2, 8, 14]. In practice, $\lambda_{\min}(D_1^{(k)^{-1}} A_{11}^{(k)})$ is unacceptably small when the problem is strongly anisotropic, but this shortcoming has to be related to the fact that damped Jacobi smoothing performs anyway poorly in such cases.

model problem analysis

Consider the model problem corresponding to the 5 point finite difference discretization of the 2D Laplacian on the unit square with Dirichlet boundary conditions and uniform mesh size $h = 2^{-\ell}$ in both directions. Let the nodes for level k correspond to the uniform grid of mesh size $h_k = 2^{-k}$, and let $A^{(k)}$ correspond to the stencil

$$\begin{array}{ccccc} & & -1 & & \\ & & | & & \\ -1 & — & 4 & — & -1 \\ & & | & & \\ & & -1 & & \end{array}$$

on that grid.

Since $A^{(k-1)}$ satisfies (3.17) for $r^{(k)}$, $p^{(k)}$ corresponding to bilinear interpolation, $c_A = 0$ in (3.11).

On the other hand, $S^{(k)}$ and $A^{(k-1)}$ have as common set of eigenvectors the discrete sine functions $\sin\, \mu\pi x h_k\ \ \sin\, \nu\pi y h_k$, $\mu = 1, \dots, \frac{h_k^{-1}}{2} - 1$, $\nu = 1, \dots, \frac{h_k^{-1}}{2} - 1$. The corresponding eigenvalues write (see [27])

$$\begin{aligned} S^{(\mu\nu)} &= 16 \left(\frac{1}{1 - \frac{c_\mu}{2} - \frac{c_\nu}{2}} + \frac{1}{1 + \frac{c_\mu}{2} - \frac{c_\nu}{2}} + \frac{1}{1 - \frac{c_\mu}{2} + \frac{c_\nu}{2}} + \frac{1}{1 + \frac{c_\mu}{2} + \frac{c_\nu}{2}} \right)^{-1} , \\ A_c^{(\mu\nu)} &= 4 - 2c_{2\mu} - 2c_{2\nu} , \end{aligned}$$

where $c_\mu = \cos\, \mu\pi h_k$, $c_\nu = \cos\, \nu\pi h_k$, $c_{2\mu} = \cos\, 2\mu\pi h_k$, $c_{2\nu} = \cos\, 2\nu\pi h_k$.

Since $diag(A^{(k)}) = 4I$, the constant $\widehat{C}_A$ in (3.22) satisfies

$$\widehat{C}_A \;=\; 4\,\omega\, \max_{\mu,\nu} \left(S^{(\mu\nu)^{-1}} - A_c^{(\mu\nu)^{-1}} \right)$$

and can therefore be analyzed directly. Using $c_{2\mu} = 2c_\mu^2 - 1$, $c_{2\nu} = 2c_\nu^2 - 1$, and letting ${c_\mu}^2 = s_\mu$, ${c_\nu}^2 = s_\nu$, tedious manipulation (easier with computer algebra) show that

$$\begin{aligned} 4\,\left(S^{(\mu\nu)^{-1}} - A_c^{(\mu\nu)^{-1}} \right) &= \frac{16\,(1 - {c_\mu}^2 - {c_\nu}^2) + 3\,({c_\mu}^4 + {c_\nu}^4) + 10\,{c_\mu}^2\,{c_\nu}^2}{\left(4 - (c_\mu + c_\nu)^2\right)\,\left(4 - (c_\mu - c_\nu)^2\right)\,(2 - {c_\mu}^2 - {c_\nu}^2)} \\ &= \frac{3\,{s_\mu}^2 + 10\,s_\mu\,s_\nu + 3\,{s_\nu}^2}{(s_\mu + s_\nu)\,(8(s_\mu + s_\nu) + (s_\mu - s_\nu)^2)} \\ &= \frac{\alpha^{(\mu\nu)}}{\beta^{(\mu\nu)}} , \end{aligned}$$

where $\alpha^{(\mu\nu)} = 3\,{s_\mu}^2 + 10\,s_\mu\,s_\nu + 3\,{s_\nu}^2$, $\beta^{(\mu\nu)} = (s_\mu + s_\nu)\,(8(s_\mu + s_\nu) + (s_\mu - s_\nu)^2)$. Since

$$\beta^{(\mu\nu)} \;-\; 2\,\alpha^{(\mu\nu)} \;=\; (s_\mu - s_\nu)^2\,\left(2 + s_\mu + s_\nu\right)$$

is always nonnegative, we deduce

$$\widehat{C}_A \;\le\; \frac{\omega}{2} . \tag{3.25}$$

On the other hand, letting $\mathbf{x}_1^{(k)}$ be the positive vector defined on the fine grid nodes at level k by

$$\left(\mathbf{x}_1^{(k)}\right)_i \;=\; \begin{cases} \sqrt{2} & \text{if } i \text{ is centered in a coarse grid square ,} \\ 1 & \text{otherwise ,} \end{cases}$$

it is easy to see that

$$\left(-\mathit{offdiag}\left(A_{11}^{(k)}\right)\right)\mathbf{x}_1^{(k)} \;\leq\; 2\sqrt{2}\,\mathbf{x}_1^{(k)}\ .$$

Hence, $\rho(\mathit{offdiag}(A_{11}^{(k)}) \leq 2\sqrt{2}$ and therefore $\lambda_{\min}(D_1^{(k)^{-1}}A_{11}^{(k)}) = \frac{1}{4}\lambda_{\min}(A_{11}^{(k)}) \geq 1 - \frac{\sqrt{2}}{2}$. Theorem 3.2 implies then

$$C_A \;\leq\; \frac{3+\sqrt{2}}{2}\,\omega\ . \tag{3.26}$$

To satisfy $\lambda_{\max}(R^{(k)}A^{(k)}) \leq 1$, $\omega = 2$ is the proper choice. Since $\lambda_{\min}(A^{(\ell-1)^{-1}}S^{(\ell)}) \geq \frac{1}{2}$ and $\lambda_{\max}(A^{(\ell-1)^{-1}}S^{(\ell)}) \leq 1$ [22], Theorem 3.1 and (3.8), (3.9) finally yield

$$\lambda_{\min}\left(B^{(\ell)^{-1}}A^{(\ell)}\right) \;\geq\; \frac{1}{5+\sqrt{2}}\ , \tag{3.27}$$

$$\lambda_{\max}\left(B^{(\ell)^{-1}}A^{(\ell)}\right) \;\leq\; 1\ , \tag{3.28}$$

independently of the number of levels.

4 Practical issues

4.1 The two-level method with approximate inversion of A_{11}

The two-level preconditioner (2.2) defined in the hierarchical basis is known for a long time to be stable when the subsystems with A_{11} are solved only approximately (e.g. [8, 11, 31]). We refer to [4, Chapter 9] for a complete analysis, from which it clearly follows that the condition number associated to

$$B_{(2)}^{(P)} \;=\; \begin{pmatrix} P & \\ A_{21} & A_c \end{pmatrix} \begin{pmatrix} I & P^{-1}A_{12} \\ & I \end{pmatrix} \tag{4.1}$$

cannot be much worse than $\kappa(P^{-1}A_{11}) \cdot \kappa(A_c^{-1}S)$, where S is the Schur complement of A.

However, it seems that such a nice behavior is partly inherited from the favorable conditioning properties associated to the hierarchical basis. Indeed, recent analysis [26] underlines that, without special care in the choice of P, preconditioners of the form (4.1) defined in the usual nodal basis are generally not spectrally equivalent to A, although problems are prevented when P satisfies some given algebraic requirements. If A is an M-matrix, these are in particular met by letting P be a MILU factorization of A_{11}. By MILU, we refer to an incomplete LU factorization in which discarded fill entries are lumped to the diagonal, to enforce P to have the same row-sum as A_{11} [4, 16]. Results in [26] prove then for $\lambda_{\min}(B_{(2)}^{(P)^{-1}}A)$ and $\lambda_{\max}(B_{(2)}^{(P)^{-1}}A)$ bounds that are closed to respectively $\lambda_{\min}(P^{-1}A_{11}) \cdot \lambda_{\min}(A_c^{-1}S)$ and $\lambda_{\max}(P^{-1}A_{11}) \cdot \lambda_{\max}(A_c^{-1}S)$.

4.2 Smoothing and approximate inversion of A_{11}

We are unfortunately not able to extend the results in Section 3 to methods for which the basic two-level scheme is of the form (4.1). One possible approach consists then in inverting A_{11} to some sufficient precision by a proper iterative scheme. The extra cost may indeed be acceptable since this is required only to ensure a good interaction with the smoother, thus not needed on the finest grid where it suffices to choose P according the recommendations above.

In the numerical experiments below, we investigate an alternative approach, where the same P is used at every level. In some sense, this approach relies on the fact that the multilevel method is likely to be stable when the basic two-level scheme itself is not deeply perturbed by the approximate inversion of A_{11}.

To support this guess, it is interesting to consider the special case where A is a 5 point M-matrix with zero row-sum except near the boundaries, and where one uses a red-black ordering of the fine grid nodes in which nodes centered in coarse grid squares are ordered first. Letting P be a a MILU factorization of A_{11} without fill-in, one then easily checks that A_{11}, P, A_{12} have the following form

$$A_{11} = \begin{pmatrix} D_{(1)} & -F \\ -E & D_{(2)} \end{pmatrix} , \tag{4.2}$$

$$P = \begin{pmatrix} Q_{(1)} & \\ -E & Q_{(2)} \end{pmatrix} \begin{pmatrix} I & -Q_{(1)}{}^{-1}F \\ & I \end{pmatrix} , \tag{4.3}$$

$$A_{12} = \begin{pmatrix} 0 \\ -G \end{pmatrix} , \tag{4.4}$$

where $D_{(1)}$, $D_{(2)}$, $Q_{(1)}$, $Q_{(2)}$ are diagonal. Therefore,

$$-P^{-1} A_{12} = \begin{pmatrix} Q_{(1)}{}^{-1}F\,G \\ Q_{(2)}{}^{-1}G \end{pmatrix} , \tag{4.5}$$

whereas the condition $P\mathbf{e} = A_{11}\mathbf{e}$ (where $\mathbf{e} = (1\dots1)^t$) implies

$$Q_{(1)} = D_{(1)} , \qquad Q_{(2)}\,\mathbf{e} = D_{(2)}\,\mathbf{e} - E\,D_{(1)}{}^{-1}\,F\,\mathbf{e} .$$

Further, the assumption $A\mathbf{e} = 0$ except near the boundaries means that

$$F\,\mathbf{e} = D_{(1)}\,\mathbf{e} \qquad \text{and} \qquad \left(D_{(2)} - E\right)\mathbf{e} = G\,\mathbf{e} ,$$

whence

$$Q_{(2)}\,\mathbf{e} = G\,\mathbf{e}$$

except near the boundaries. With $Q_{(1)}$, $Q_{(2)}$ determined in this way, it is not difficult to check from (4.5) that, apart boundary effects, the prolongation and restriction

$$p_{A(P)} = \begin{pmatrix} -P^{-1}A_{12} \\ I \end{pmatrix} , \qquad r_{A(P)} = \begin{pmatrix} -A_{21}P^{-1} & I \end{pmatrix}$$

are identical to the matrix dependent prolongation and restriction proposed in [1, 18] for robust black-box multigrid. At least in this case, it is thus quite reasonable to expect a good interaction of

$$B_{(2)}^{(P)\,-1} = \begin{pmatrix} P^{-1} & 0 \\ 0 & 0 \end{pmatrix} + p_{A(P)}\,A_c^{-1}\,r_{A(P)} ,$$

with a smoothing procedure.

4.3 Red-black ordering of the fine grid nodes

The ordering mentioned above may have also some practical interest, although the resulting approximation of A_{11}^{-1} is expected to be less accurate than with a natural ordering. Indeed, on the one hand, using a red-black ordering makes the solve with P easy to parallelize. If the smoother is also parallelizable, the whole method is then well suited to parallel computation, requiring only some handling to efficiently cope with the coarsest levels that have to few unknowns to be treated in parallel.

On the other hand, assuming P and A_{12} of the form (4.3), (4.4), and letting

$$L_P = \begin{pmatrix} I & \\ -E\,Q_{(1)}^{-1} & I \end{pmatrix} \quad , \quad Q = \begin{pmatrix} Q_{(1)} & \\ & Q_{(2)} \end{pmatrix} \quad , \quad U_P = \begin{pmatrix} I & -Q_{(1)}{}^{-1}F \\ & I \end{pmatrix} ,$$

it follows that

$$L_P^{-1} A_{12} = A_{12} \ , \qquad A_{21}\, U_P^{-1} = A_{21} \ .$$

Hence, since $P = L_P\, Q\, U_P$,

$$\begin{aligned} B^{(P)} &= \begin{pmatrix} I & \\ A_{21}P^{-1} & M \end{pmatrix} \begin{pmatrix} P & \\ & M \end{pmatrix} \begin{pmatrix} I & P^{-1}A_{12} \\ & I \end{pmatrix} \\ &= \begin{pmatrix} I & \\ A_{21}Q^{-1}L_P^{-1} & I \end{pmatrix} \begin{pmatrix} L_P Q U_P & \\ & M \end{pmatrix} \begin{pmatrix} I & U_P^{-1}Q^{-1}A_{12} \\ & I \end{pmatrix} \\ &= \begin{pmatrix} L_P & \\ A_{21}Q^{-1} & I \end{pmatrix} \begin{pmatrix} Q & \\ & M \end{pmatrix} \begin{pmatrix} U_P & Q^{-1}A_{12} \\ & I \end{pmatrix} . \end{aligned}$$

Therefore, although in general two solves with P are required at each level, here only one forward and one backward sweep will be needed, allowing to cut the associate work by a factor about 2.

4.4 Coarse grid matrix

For finite element matrices computed in the hierarchical basis, it is natural to take $A^{(k-1)} = A_{22}^{(hb)^{(k)}}$, and for usual finite difference or finite element matrices, a similar choice is obtained with

$$A^{(k-1)} = r^{(k)}\, A^{(k)}\, p^{(k)} \ , \tag{4.6}$$

where $p^{(k)}$ is an appropriate prolongation operator and $r^{(k)} = p^{(k)^t}$.

For finite difference matrices, an alternative technique is proposed in [25], which consists in letting $\Delta^{(k)}$ be the diagonal matrix with same row-sum as $A_{11}^{(k)}$, and taking as coarse grid matrix the Schur complement $\widetilde{S}^{(k)}$ of

$$\widetilde{A}^{(k)} = \begin{pmatrix} \Delta^{(k)} & A_{12}^{(k)} \\ A_{21}^{(k)} & A_{22}^{(k)} \end{pmatrix} ,$$

in which zero rows and columns are symmetrically deleted. If $A^{(k)}$ is an M-matrix, it can be proved that

$$\lambda_{\min}\left(\widetilde{S}^{(k)^{-1}} S^{(k)}\right) \geq 1 \ , \tag{4.7}$$

where $S^{(k)}$ is the Schur complement of $A^{(k)}$. On the other hand, for 2D PDEs of the type

$$-\partial_x\, a_x\, \partial_x\, u - \partial_y\, a_y\, \partial_y\, u = f\ ,$$

with coefficients piecewise constant on the coarse mesh, it is shown that

$$\widetilde{S}^{(k)}\ =\ \frac{1}{2}\, r^{(k)}\, A^{(k)}\, p^{(k)} \tag{4.8}$$

for the prolongation $p^{(k)}$ corresponding to linear (7 point) interpolation and $r^{(k)} = {p^{(k)}}^t$ [25]. Since this prolongation is of the form (3.18), $\lambda_{\max}((r^{(k)}A^{(k)}p^{(k)})^{-1}S^{(k)}) \leq 1$ and thus

$$\lambda_{\max}\left({\widetilde{S}^{(k)}}^{-1}\, S^{(k)}\right)\ \leq\ 2\ . \tag{4.9}$$

Now, although the algebraic multilevel method alone is relatively insensitive to the scaling of the coarse grid matrix (see (2.3)), its combination with a smoothing strategy requires a proper scaling of $A^{(k-1)}$, to ensure that $\mathbf{v}^t {B_{(2)}^{(k)}}^{-1}\mathbf{v} \approx \mathbf{v}^t {A^{(k)}}^{-1}\mathbf{v}$ for vectors $\mathbf{v}$ corresponding to the smallest eigenvalues of $A^{(k)}$ (otherwise, since $\mathbf{v}^t {A^{(k)}}^{-1}\mathbf{v} = \mathcal{O}(h_k^{-2})\mathbf{v}^t\mathbf{v}$, the difference $\mathbf{v}^t {A^{(k)}}^{-1}\mathbf{v} - \mathbf{v}^t {B_{(2)}^{(k)}}^{-1}\mathbf{v}$ could not be bounded independently of h_k).

Hence, if the method works properly with (4.6), it cannot work directly with $A^{(k-1)} = \widetilde{S}^{(k)}$, but the choice

$$A^{(k-1)}\ =\ 2\,\widetilde{S}^{(k)} \tag{4.10}$$

is possible and reproduces, in the piecewise constant coefficients case, the results obtained with linear interpolation for which classical multigrid convergence theory is widely developed.

This equivalence is lost in more general situations, but it should be noted that standard multigrid algorithms tend to perform poorly when the PDE coefficients highly vary or present jumps inside the coarse grid cells [1]. In such cases, our approach to stabilize the condition number becomes thus anyway more or less empirical, so that it is tempting to still rely on (4.10), since the computation of $\widetilde{S}^{(k)}$ is cheap, easy to implement, and results always in a five point matrix when $A^{(k)}$ corresponds to a 5 point scheme.

Note that the poor behavior of standard multigrid algorithms does not necessarily implies that our approach to stabilize the condition number will fail. Indeed, algebraic multilevel preconditioners (and our main theoretical assumption (3.11)) depend only on the coarse grid matrix and do not refer to any interpolation procedure, whereas works that develop more robust multigrid algorithms (e.g. [1, 17, 18]) stress that the problems of standard multigrid are due to an inappropriate interpolation rather than to a bad coarse grid approximation. Referring to our theoretical analysis, this means that (3.21) may hold in cases for which (3.20) does not hold or holds only for a prolongation $p^{(k)}$ that is not known a priori or impractical to use.

For 3D finite difference problems, it is also interesting to base $A^{(k-1)}$ on $\widetilde{S}^{(k)}$, Galerkin approximations of the type (4.6) tending to produce matrices with a 27 point sparsity pattern even though $A^{(k)}$ corresponds to a 7 point stencil. Here, no relation similar to (4.8) seems exist. However, the multigrid theory applies as well when the matrices $A^{(k)}$ are obtained by appropriately scaled discretizations of the PDE on the successive grids [20, 35]. For finite difference approximations of 3D problems, prolongations of the form (3.18) and restriction $r^{(k)} = {p^{(k)}}^t$, the right scaling is the one for which, if $A^{(k-1)}$ corresponds to the 7 point stencil

$$\begin{array}{ccccc}
 & & -1 & & \\
 & & | & & -1 \\
 & & | & / & \\
-1 & — & 6 & — & -1 \\
 & / & | & & \\
-1 & & | & & \\
 & & -1 & &
\end{array}$$

then $A^{(k-1)}$ correspond to the stencil

$$\begin{array}{ccccc}
 & & -2 & & \\
 & & | & & -2 \\
 & & | & / & \\
-2 & — & 12 & — & -2 \\
 & / & | & & \\
-2 & & | & & \\
 & & -2 & &
\end{array}$$

acting on the coarse grid nodes. On the other hand, with the given stencil for $A^{(k)}$, $\widetilde{S}^{(k)}$ will correspond to

$$\begin{array}{ccccc}
 & & -\frac{1}{2} & & \\
 & & | & & -\frac{1}{2} \\
 & & | & / & \\
-\frac{1}{2} & — & 3 & — & -\frac{1}{2} \\
 & / & | & & \\
-\frac{1}{2} & & | & & \\
 & & -\frac{1}{2} & &
\end{array}$$

We are thus led to the rule

$$A^{(k-1)} = 4\,\widetilde{S}^{(k)}\,. \tag{4.11}$$

Remark The scaling of $\widetilde{S}^{(\ell)}$ is of course no necessary when smoothing is not performed on the finest grid. For the two-level method with exact inversion of A_{11}, using the rule (4.10) (2D) or (4.11) (3D) should be equivalent to the simpler choice $A^{(\ell-1)} = \widetilde{S}^{(\ell)}$. For the multilevel method with approximate inversion of A_{11}, results may however be slightly different. Both variants are included in the numerical experiments reported below.

4.5 Choice of the smoother

Many works on multigrid methods focus on the design of efficient smoothers. Various possibilities are tested and compared in e.g. [21]. In the present context however, we feel that the results will depend less critically on the efficiency of the smoothing scheme.

Indeed, on the one hand, since no smoothing is performed on the finest grid, the involved number of arithmetic operations per unknown will have a weaker influence on the total CPU time. On the other hand, the bounds (3.8), (3.9) correspond to some kind of worst case estimate, and, in practice, slight variations of $\kappa(M^{(\ell-1)}A^{(\ell-1)})$ will not necessarily have a noticeable effect on $\kappa({B^{(\ell)}}^{-1}A^{(\ell)})$.

Therefore, we included only two basic smoothing strategies in our numerical experiments, each of them with just one pre- and one post-smoothing iteration. These are damped Jacobi smoothing with $\omega = 2$ on the one hand and Relaxed ILU($\omega = -1$) [9] on the other hand. Both satisfy $\lambda_{\max}(R^{(k)}A^{(k)}) \leq 1$ when $A^{(k)}$ is an M-matrix [24, 30]. Damped Jacobi is fully parallelizable but performs poorly in presence of anisotropy, whereas RILU based smoothers are robust for anisotropic problems, at least in 2D [36]. For both choices, performances are generally improved by optimizing the parameter ω, but, according the arguments above, we did not include such refinements in our tests.

4.6 Computational complexity of one iteration step

We evaluate the numerical cost of the multilevel preconditioner in the case where, at each level, $A_{11}^{(k)}$ is approximated by a MILU factorization without fill-in. We let n_k be the order of the matrix $A^{(k)}$, $nz(C)$ the number of nonzero entries in (sub)matrix C, and a *flop* be a basic ($+$ or $*$) arithmetic operation. We further set

$$c_o = \begin{cases} 0 & \text{if one uses a red-black ordering of the fine grid nodes} \\ & \quad \text{and considers the trick mentioned in subsection 4.3 ,} \\ 1 & \text{otherwise ,} \end{cases} \tag{4.12}$$

$$c_s = \begin{cases} 0 & \text{in case of damped Jacobi smoothing ,} \\ 1 & \text{in case of RILU smoothing .} \end{cases} \tag{4.13}$$

One application of the preconditioner at level ℓ requires then

- $1 + c_o$ solves with P, each of them involving $2\,nz(A_{11}^{(\ell)})$ *flops*;
- one multiplication by $A_{12}^{(\ell)}$ ($2\,nz(A_{12}^{(\ell)})$ *flops*) and one multiplication by $A_{21}^{(\ell)}$ ($2\,nz(A_{21})$ *flops*);
- one multiplication by M that involves two additions on vectors of length $n_{\ell-1}$ ($2\,n_{\ell-1}$ *flops*), two computation of the type $y := y - A^{(\ell-1)}x$ ($4\,nz(A^{(\ell-1)})$ *flops*), two applications of the smoother ($(4c_s\,nz(A^{(\ell-1)}) + 2(1-c_s)n_{\ell-1})$ *flops*), and one application of the preconditioner at level $\ell - 1$.

Letting $c(k)$ be the number of *flops* involved at level k, one has thus

$$\begin{aligned} c(\ell) \;&=\; 2(c_o+1)\,nz(A_{11}^{(\ell)}) \;+\; 2\,nz(A_{12}^{(\ell)}) \;+\; 2\,nz(A_{21}^{(\ell)}) \\ &\quad + 4(c_s+1)\,nz(A^{(\ell-1)}) \;+\; 2(2-c_s)n_{\ell-1} \;+\; c(\ell-1) \;. \end{aligned}$$

Let assume for simplicity that $A_{22}^{(k)}$ is diagonal for all k (i.e. $nz(A_{22}^{(k)}) = n_{k-1}$), and that the cost involved by $M^{(1)}$ is negligible. One obtains, following the recursion,

$$\begin{aligned} c(\ell) \;&=\; 2\left(c_o\,nz(A_{11}^{(\ell)}) \;+\; nz(A^{(\ell)})\right) \\ &\quad + 2\left(2(c_s+1)\,nz(A^{(\ell-1)}) \;+\; (1-c_s)\,n_{\ell-1}\right) \;+\; c(\ell-1) \\ &=\; 2\sum_{k=1}^{\ell}\left(c_o\,nz(A_{11}^{(k)}) \;+\; nz(A^{(k)})\right) \\ &\quad + 2\sum_{k=1}^{\ell-1}\left(2(c_s+1)\,nz(A^{(k)}) \;+\; (1-c_s)\,n_k\right) \;. \end{aligned}$$

For 2D problems and five point matrices, one has, with a standard coarsening, $nz(A^{(k)}) \approx 5n_k$, $nz(A_{11}^{(k)}) \approx \frac{11}{4}n_k$ and $n_k \approx 4^{k-\ell}n_\ell$. Therefore,

$$\begin{aligned} c(\ell) \;&\approx\; \frac{2}{3}\,(11c_o + 20 + 10(c_s+1) + 1 - c_s)\;n_\ell \\ &=\; \left(21 + 7c_o + 6c_s - \frac{1-c_o}{3}\right)\,n_\ell \;. \end{aligned}$$

With the most costly options ($c_o = c_s = 1$), $c(\ell) \approx 34n_\ell$. This is slightly less than with a simple W cycle instead of smoothing, for which $c(\ell) \approx 36n_\ell$ [25]. Concerning the global cost of the algorithm, since the conjugate gradient method requires additionally two inner products ($4n_\ell$ *flops*), three saxpy operations ($6n_\ell$ *flops*) and one multiplication by $A^{(\ell)}$ (about $9n_\ell$ *flops*), the total number of operations per iteration is thus about $53n_\ell$ when $c_o = c_s = 1$, $47n_\ell$ when $c_o = 1$, $c_s = 0$ and $40n_\ell$ when $c_o = c_s = 0$. By comparison, it is about $29n_\ell$ *flops* with an ILU(0) like preconditioner and $33n_\ell$ *flops* with a block ILU like preconditioner (e.g. [9]).

For 3D problems and 7 point matrices one has, with a standard coarsening, $nz(A^{(k)}) \approx 7n_k$, $nz(A_{11}^{(k)}) \approx \frac{37}{8}n_k$ and $n_k \approx 8^{k-\ell}n_\ell$. Therefore,

$$\begin{aligned} c(\ell) &\approx \frac{2}{7}\left(37c_o + 56 + 14(c_s+1) + 1 - c_s\right) n_\ell \\ &= \left(21 + 10c_o + 4c_s - \frac{5 + 2c_s - 4c_o}{7}\right) n_\ell \ , \end{aligned}$$

which is roughly the same as in 2D although matrix vector multiplications and standard preconditioners are here slightly more costly.

5 Numerical results

We have tested the multilevel preconditioner on linear systems resulting from the finite difference discretization of

$$-\partial_x\, a_x\, \partial_x\, u \;-\; \partial_y\, a_y\, \partial_y\, u \;\left(-\; \partial_z\, a_z\, \partial_z\, u\,\right) \;=\; f \quad \text{in} \quad \Omega$$

$$\left\{ \begin{array}{lcl} u & = & 0 \quad \text{on} \quad \Gamma_0 \subset \partial\,\Omega \\ \dfrac{\partial u}{\partial n} & = & 0 \quad \text{on} \quad \Gamma_1 = \partial\,\Omega \backslash \Gamma_0 \ , \end{array} \right.$$

where Ω is the unit square (2D) or unit cube (3D). We use the point mesh box integration scheme [23] with, except on Problem 3D1, uniform mesh size h in each direction.

In all experiments, we ran the preconditioned conjugate gradient algorithm with the zero vector as initial approximation, and the reported the number of iterations (# it.) is the one needed to reduce the relative residual error $\frac{\|r_k\|}{\|b\|}$ below 10^{-8}. The reported CPU times are given in seconds and include preprocessing, that is the operations needed to compute the successive incomplete factorizations of A_{11}, approximate Schur complements and relaxation operators. No particular optimization effort was made, so that the timing results cannot pretend to be optimal and should be used only as a relative criteria. Whenever the CPU time is not reported, this is because, for technical reasons, the corresponding test was run with a program using a completely different data structure, so that a comparison would be meaningless. The considered machine is a SUN SPARC20/514 workstation with 128 Mb of RAM memory.

Tested methods

Except when *no smoothing* or *smoothing at level ℓ* is specified, the considered preconditioner is the one discussed in Sections 3 and 4 with $A_{11}^{(k)}$ approximated by a MILU factorization without fill-in and in which $A^{(k-1)}$, $k = \ell - 1, \ldots, 1$ is defined by the rule (4.10) (2D) or (4.11) (3D). According the remark at the end of subsection 4.4, $A^{(\ell-1)}$ is defined either by the general rule or by $A^{(\ell-1)} = \widetilde{S}^{(\ell)}$. When nothing is said about $A^{(\ell-1)}$, the latter choice is assumed. In the tables, "nat. ord." means that the MILU factorization of $A_{11}^{(k)}$ was carried out with respect to a natural ordering of the concerned nodes, whereas "RB ord." refers to the scheme considered in subsections 4.2 and 4.3 for 2D problems; in 3D, it means that a red-black like ordering was considered, in which one orders first the nodes centered in coarse grid boxes, next the fine grid nodes embedded in coarse grid planes and centered in coarse grid squares, finally the fine grid nodes embedded in coarse grid lines. "DJ" and "RILU" refer to respectively damped Jacobi and Relaxed ILU smoothing as discussed in subsection 4.4 (in case of RILU smoothing, a natural ordering was considered); "two-level" corresponds to the reference two-level scheme (4.1). We always considered standard coarsening and, except in the two-level case, performed full reduction, leaving only one node in the coarsest level, this node being the first one in the natural ordering of the fine grid (after removing the nodes on Γ_0).

For the sake of completeness, we also tested the preconditioner defined as above, except that smoothing was also performed on the finest grid (*smoothing at level* ℓ); then, $A^{(\ell-1)}$ was always defined by the general rule.

Further, we included the V cycle case without smoothing; then, we always considered a natural ordering to define the MILU factorization of $A_{11}^{(k)}$, and, for all k , either used the rule (4.10) or (4.11) ("scaling") or the simpler choice $A^{(k-1)} = \widetilde{S}^{(k)}$ ("no scaling"). To limit the effect of the dependency of the number of levels on the condition number, the coarsest grid considered was here 16×16 (2D, $h^{-1} = 128$), 32×32 (2D, $h^{-1} = 512$) or $5 \times 5 \times 5$ (3D), these grid sizes being limited on account of the involved factorization cost.

On the other hand, W cycle without smoothing refers to the method presented in [25] for 2D problems, which is based on a MILU factorization of $A_{11}^{(k)}$ with a natural ordering, using $A^{(k-1)} = \widetilde{S}^{(k)}$ for all k and a second degree W cycle scheme to define $M^{(k)}$, with parameters fixed according eigenvalue bounds proved in [25] assuming that the PDE coefficients are piecewise constant on the coarsest mesh. To get exactly the same method as the one tested in [25], $A^{(k)}$ was in this case factorized exactly as soon as $n_k \leq 81$ was reached.

Finally, MILU refers to a simple modified ILU preconditioner, computed with respect to a natural ordering and discarding all fill-in, whereas DRILU is the perturbed variant of MILU introduced in [24].

2D Laplacian

Result for the model Poisson problem ($a_x = a_y = f \equiv 1$, $\Gamma_0 = \partial\Omega$) are reported on Table 1.

We make the following comments.

- Clearly, using smoothing is successful in stabilizing the condition number associated to the V cycle. The latter is independent of the mesh size and approaches closely that of the corresponding two-level method. On the other hand, V cycle without smoothing is far from being competitive even though the number of level was limited by factorizing exactly a rather large coarse grid matrix.

- The variants with $A^{(\ell-1)} = \widetilde{S}^{(\ell)}$ perform better because the corresponding two-level scheme itself is improved by this choice, probably on account of a better interaction

preconditioner		$\lambda_{\min}$	$\lambda_{\max}$	κ	# it.	CPU
	$h^{-1} = 128$					
$A^{\ell-1} = \widetilde{S}^{(k)}$						
nat. ord.:	DJ smoothing	0.82	2.00	2.41	14	2.61
	RILU smoothing	1.00	2.02	2.02	12	2.46
	two-level	1.01	2.11	2.10	13	
RB ord.:	DJ smoothing	0.81	3.43	4.23	20	3.12
$A^{\ell-1} = 2\widetilde{S}^{(k)}$						
nat. ord.:	DJ smoothing	0.41	1.25	3.03	16	2.92
	RILU smoothing	0.50	1.27	2.54	14	2.87
	two-level	0.50	1.29	2.56	14	
smoothing at level ℓ						
nat. ord.:	DJ smoothing	0.59	1.08	1.83	12	3.32
	RILU smoothing	0.80	1.00	1.24	8	2.75
RB ord.:	DJ smoothing	0.59	1.59	2.69	15	3.59
no smoothing						
	V cycle (no scaling)	1.00	8.25	8.23	28	
	V cycle (scaling)	0.13	1.32	10.3	27	
	W cycle	0.74	2.00	2.70	15	
	MILU	1.0	40.9	40.9	54	5.55
	$h^{-1} = 512$					
$A^{\ell-1} = \widetilde{S}^{(k)}$						
nat. ord.:	DJ smoothing	0.82	2.04	2.47	15	63.9
	RILU smoothing	1.00	2.03	2.03	13	59.7
RB ord.:	DJ smoothing	0.81	3.47	4.30	21	73.8
no smoothing						
	V cycle (no scaling)	1.00	17.1	17.0	43	
	W cycle	0.73	2.00	2.72	16	
	MILU	1.00	174.	174.	124	370.

Table 1: results for the 2D Laplacian

with the used approximate inverse of A_{11} (exact inversion would result in the same condition number).

- It does not pay off to perform smoothing also on the finest grid. We however note that the difference is slight in the case of RILU smoothing, and one may therefore think that using a well tuned smoother could make this variant competitive.

- Concerning the reference methods, the W cycle without smoothing confirms its good behavior [25], whereas MILU is as expected not optimal, further tests showing that it becomes competitive with the best V cycle variants when h^{-1} is not larger than 32.

preconditioner		$\lambda_{\min}$	$\lambda_{\max}$	κ	# it.	CPU
$h^{-1} = 100$						
V cycle	(nat. ord., DJ)	1.02	2.60	2.56	16	1.33
V cycle	(nat. ord., RILU)	1.01	2.43	2.40	15	1.43
V cycle	(RB ord., DJ)	0.92	5.85	6.34	24	1.67
W cycle	(no smoothing)	0.86	2.00	2.31	14	
two-level		1.02	2.22	2.18	15	
$h^{-1} = 128$						
V cycle	(nat. ord., DJ)	1.03	2.50	2.43	15	2.41
V cycle	(nat. ord., RILU)	1.02	2.53	2.50	14	2.59
V cycle	(RB ord., DJ)	0.96	3.58	3.71	20	3.04
W cycle	(no smoothing)	0.86	2.00	2.31	14	
two-level		1.02	2.11	2.07	14	
$h^{-1} = 400$						
V cycle	(nat. ord., DJ)	1.05	2.51	2.41	17	33.6
V cycle	(nat. ord., RILU)	1.01	2.32	2.30	15	33.6
V cycle	(RB ord., DJ)	0.96	4.98	5.20	24	40.0
W cycle	(no smoothing)	0.86	2.00	2.33	15	
$h^{-1} = 512$						
V cycle	(nat. ord., DJ)	1.04	2.52	2.41	16	64.8
V cycle	(nat. ord., RILU)	1.02	2.54	2.50	15	79.4
V cycle	(RB ord., DJ)	0.96	3.55	3.69	21	65.3
W cycle	(no smoothing)	0.86	2.00	2.33	15	

Table 2: results for Problem 2D1

Problem 2D1

Here,

$$\begin{aligned} a_x &= a_y = \begin{cases} 10^3 & \text{in } \left(\frac{1}{4},\frac{3}{4}\right)\times\left(\frac{1}{4},\frac{3}{4}\right) \\ 1 & \text{elsewhere}\ , \end{cases} \\ f &= \begin{cases} 1 & \text{in } \left(\frac{1}{4},\frac{3}{4}\right)\times\left(\frac{1}{4},\frac{3}{4}\right) \\ 0 & \text{elsewhere}\ , \end{cases} \\ \Gamma_0 &= \{(x,y)\mid 0\le x\le 1,\ y=1\}\ . \end{aligned}$$

Results are reported in Table 2 for the most interesting methods, according our preceding conclusions. The mesh sizes not corresponding to a negative power of two are especially interesting because they imply that jumps in the PDE coefficients arise inside coarse grid cells. However, this has little influence on the results, except for the variant using a red-black ordering of the fine grid nodes, which is relatively less robust.

Comparing with Table 1, the V cycle variants appear slightly more sensitive to the discontinuities than the W cycle without smoothing. In our opinion, the results obtained for the model problem with the V cycle variants are somewhat too fortunate (the condition number is sometimes better than for the two-level method!). Hence, the results obtained here better reflect the real potentialities of these methods, which appear therefore roughly equivalent to the W cycle method of [25] rather than slightly better as one could have conclude from Table 1.

Problem 2D2

Here,

$$
\begin{aligned}
a_x &= \begin{cases} d & \text{in} \quad \left(\frac{1}{4},\frac{1}{2}\right)\times\left(\frac{1}{4},\frac{1}{2}\right) \\ 1 & \text{elsewhere}\ , \end{cases} \\
a_y &= \begin{cases} d & \text{in} \quad \left(\frac{1}{2},\frac{3}{4}\right)\times\left(\frac{1}{2},\frac{3}{4}\right) \\ 1 & \text{elsewhere}\ , \end{cases} \\
f &= \begin{cases} 1 & \text{in} \quad \left(\frac{1}{4},\frac{3}{4}\right)\times\left(\frac{1}{4},\frac{3}{4}\right) \\ 0 & \text{elsewhere}\ , \end{cases} \\
\Gamma_0 &= \{(x,y)\mid 0\leq x\leq 1,\ y=1\}\ ,
\end{aligned}
$$

i.e. the problem is strongly anisotropic for large d. We therefore only tested the version using a natural ordering of the fine grid nodes and RILU smoothing, red-black ordering and damped Jacobi smoothing leading generally to poor performances in presence of anisotropy.

The results are reported in Table 3. Here, jumps inside the coarse grid cells have a more noticeable effect on the convergence rate of the V cycle, whereas the W cycle fails to produce a positive definite preconditioner. We however stress that this is not an intrinsic limitation of W cycle based methods, but rather a shortcoming of versions using parameters predetermined in function of theoretical estimates, whose use is dangerous as soon as the assumptions needed in the proofs of these estimates are not met anymore. Since one may alternatively resort to run-time parameter estimation, the main argument in favor of the V cycle stabilized with smoothing is not that it is intrinsically more robust, but rather that a robust version requires less implementation effort and less preprocessing.

3D Laplacian

We repeated for the 3D Poisson problem ($a_x = a_y = a_z = f \equiv 1$, $\Gamma_0 = \partial\Omega$) the experiments made in the 2D case, except that a comparative run with two-level methods was here not possible, because of the involved factorization cost. Also, no 3D W cycle method was available to us.

The results are reported in Table 4. Roughly, the same comments as in 2D applies, except that versions with smoothing on the finest grid perform relatively better, especially for the smallest problem size. MILU offers also a more interesting alternative. In fact, as in 2D, MILU is competitive with the best V cycle versions for $h^{-1} \leq 32$, but the 3D meshes are necessarily less refined than the 2D ones.

preconditioner		$\lambda_{\min}$	$\lambda_{\max}$	κ	# it.	CPU
		$d = 10^2$				
		$h^{-1} = 100$				
V cycle	(nat. ord., RILU)	1.00	2.93	2.92	16	1.56
W cycle	(no smoothing)	n.p.d.	n.p.d.	n.p.d.	n.c.	
two-level		1.00	2.66	2.66	15	
		$h^{-1} = 128$				
V cycle	(nat. ord., RILU)	1.00	2.43	2.41	15	2.78
W cycle	(no smoothing)	0.80	2.17	2.71	16	
two-level		1.00	2.30	2.30	14	
		$h^{-1} = 400$				
V cycle	(nat. ord., RILU)	1.00	2.76	2.75	16	36.8
W cycle	(no smoothing)	n.p.d.	n.p.d.	n.p.d.	n.c.	
		$h^{-1} = 512$				
V cycle	(nat. ord., RILU)	0.97	2.53	2.59	17	86.5
W cycle	(no smoothing)	0.80	2.20	2.76	17	
		$d = 10^4$				
		$h^{-1} = 100$				
V cycle	(nat. ord., RILU)	1.00	4.53	4.53	18	1.73
W cycle	(no smoothing)	n.p.d.	n.p.d.	n.p.d.	n.c.	
two-level		1.00	3.41	3.40	17	
		$h^{-1} = 128$				
V cycle	(nat. ord., RILU)	1.00	2.66	2.66	16	2.99
W cycle	(no smoothing)	0.73	2.24	3.09	19	
two-level		1.00	2.39	2.38	15	
		$h^{-1} = 400$				
V cycle	(nat. ord., RILU)	1.00	4.08	4.07	17	37.6
W cycle	(no smoothing)	n.p.d.	n.p.d.	n.p.d.	n.c.	
		$h^{-1} = 512$				
V cycle	(nat. ord., RILU)	0.93	2.84	3.05	18	92.0
W cycle	(no smoothing)	0.68	2.31	3.39	20	

Table 3: results for Problem 2D2

Problem 3D1

Here,

$$a_x \;=\; a_y \;= a_z \;= \begin{cases} 10^3 & \text{in} \quad \left(0, \frac{1}{8}\right) \times \left(0, \frac{1}{8}\right) \times \left(0, \frac{1}{8}\right) \\ 1 & \text{elsewhere} \;, \end{cases}$$

$$f \;=\; \begin{cases} 1 & \text{in} \quad \left(0, \frac{1}{8}\right) \times \left(0, \frac{1}{8}\right) \times \left(0, \frac{1}{8}\right) \\ 0 & \text{elsewhere} \;, \end{cases}$$

preconditioner		$\lambda_{\min}$	$\lambda_{\max}$	κ	# it.	CPU
		$h^{-1} = 40$				
$A^{\ell-1} = \widetilde{S}^{(k)}$						
nat. ord.:	DJ smoothing	0.85	4.57	5.37	22	15.5
	RILU smoothing	11.00	4.07	4.07	19	14.1
RB ord.:	DJ smoothing	0.84	10.1	12.0	34	19.4
$A^{\ell-1} = 4\widetilde{S}^{(k)}$						
nat. ord.:	DJ smoothing	0.21	1.63	7.66	27	18.6
	RILU smoothing	0.25	1.61	6.40	23	16.9
smoothing at level ℓ						
nat. ord.:	DJ smoothing	0.49	1.36	2.79	15	16.2
	RILU smoothing	0.70	1.07	1.54	9	13.0
RB ord.:	DJ smoothing	0.48	2.74	5.65	22	21.4
no smoothing						
	V cycle (no scaling)	1.00	41.5	41.5	45	
	V cycle (scaling)	0.016	1.66	103.	76	
	MILU	1.00	16.9	16.9	36	16.8
		$h^{-1} = 80$				
$A^{\ell-1} = \widetilde{S}^{(k)}$						
nat. ord.:	DJ smoothing	0.85	4.83	5.66	24	158.
	RILU smoothing	1.00	4.29	4.19	20	141.
RB ord.:	DJ smoothing	0.84	11.2	13.3	37	212.
smoothing at level ℓ						
nat. ord.:	DJ smoothing	0.48	1.37	2.84	16	178.
	RILU smoothing	0.68	1.07	1.57	10	146.
RB ord.:	DJ smoothing	0.47	2.92	6.18	24	229.
no smoothing						
	V cycle (no scaling)	1.00	165.	165.	90	
	MILU	1.00	36.2	36.2	55	247.

Table 4: results for the 3D Laplacian

$$\Gamma_0 \quad = \quad \{(x,y) \mid 0 \leq x,y \leq 1,\ z=0z=0 \text{ or } z=1\} \ .$$

We deliberately omitted 3D tests with strongly anisotropic problems because the kind of smoother considered here cannot pretend to be efficient in such cases, and alternative approaches resorting to plane relaxation or semi-coarsening should be considered instead (see e.g. [21]). On the other hand, more often than in 2D, simulations make use of different mesh sizes in the different directions, so that the effect of moderate anisotropy induced by grid stretching deserves to be investigated. The results are reported in Table 5. For the "non uniform mesh", we used $h_x = h(x)$, $h_y = h(y)$, $h_y = h(y)$, where the function

preconditioner		$\lambda_{\min}$	$\lambda_{\max}$	κ	# it.	CPU
$h_x^{-1} = h_y^{-1} = h_z^{-1} = 40$						
V cycle	(nat. ord., DJ)	0.88	4.96	5.61	21	18.4
V cycle	(nat. ord., RILU)	1.00	4.25	4.25	18	16.2
V cycle	(RB ord., DJ)	0.85	11.7	13.8	34	21.1
MILU		1.00	111.	111.	66	35.0
DRILU		0.82	20.4	25.0	42	22.6
$h_x^{-1} = h_y^{-1} = h_z^{-1} = 80$						
V cycle	(nat. ord., DJ)	0.87	5.15	5.87	23	171.
V cycle	(nat. ord., RILU)	1.00	4.39	4.38	20	164.
V cycle	(RB ord., DJ)	0.84	12.3	14.6	37	210.
MILU		1.00	548.	548.	143	717.
DRILU		0.83	44.3	53.5	67	337.
$h_x^{-1} = h_y^{-1} = 40\ ,\quad h_z^{-1} = 80$						
V cycle	(nat. ord., DJ)	0.86	6.41	7.47	26	46.0
V cycle	(nat. ord., RILU)	1.00	4.70	4.70	20	37.8
DRILU		0.79	24.7	31.2	48	52.8
$h_x^{-1} = h_y^{-1} = 40\ ,\quad h_z^{-1} = 160$						
V cycle	(nat. ord., DJ)	0.79	12.6	16.0	39	166.
V cycle	(nat. ord., RILU)	1.00	6.58	6.57	24	116.
DRILU		.64	31.5	49.6	57	143.
$h_x^{-1} = h_y^{-1} = 40\ ,\quad h_z^{-1} = 320$						
V cycle	(nat. ord., RILU)	1.00	11.2	11.2	32	307.
DRILU		0.44	43.6	98.8	69	382.
$h_x^{-1} = h_y^{-1} = 80\ ,\quad h_z^{-1} = 40$						
V cycle	(nat. ord., DJ)	0.77	8.60	11.2	30	121.
V cycle	(nat. ord., RILU)	1.00	5.86	5.85	22	102.
DRILU		0.75	35.9	47.5	60	172.
non uniform mesh ($40 \times 40 \times 40$ grid)						
V cycle	(nat. ord., DJ)	0.81	9.70	11.9	32	26.0
V cycle	(nat. ord., RILU)	1.00	6.96	6.95	24	20.9
DRILU		0.62	13.2	21.2	40	19.6
non uniform mesh ($80 \times 80 \times 80$ grid)						
V cycle	(nat. ord., DJ)	0.86	11.7	13.6	36	254
V cycle	(nat. ord., RILU)	1.00	8.55	8.53	28	222.
DRILU		.61	28.2	46.1	60	328

Table 5: results for Problem 3D1

h is defined by

$$h(t) \;=\; \begin{cases} 0.125\, h_0 & \text{if } 0 \leq t \leq 0.25 \\ 0.25\, h_0 & \text{if } 0.25 \leq t \leq 0.5 \\ 0.5\, h_0 & \text{if } 0.5 \leq t \leq 1 \end{cases} ,$$

with $h_0 = 10^{-1}$ for the $40 \times 40 \times 40$ grid and $h_0 = 20^{-1}$ for the $80 \times 80 \times 80$ grid. Here, DRILU [24] was taken as reference method, as it is an example of perturbed MILU factorizations, that perform similarly on the model problem, but improve it significantly in the variable coefficient case [4, 5].

As in 2D, the performances observed on the model problem are reproduced with only slight variations in the purely isotropic case. On the other hand, only the version using natural ordering and RILU smoothing is sufficiently robust in the anisotropic cases, although, as expected, performances tend to deteriorate for large anisotropy ratio (when $h_x^{-1} = h_y^{-1} = 40$ and $h_z^{-1} = 320$, the matrix coefficient in the z direction is 64 times smaller than those in the x and y directions).

References

[1] R. E. Alcouffe, A. Brandt, J. E. Dendy, and J. W. Painter, *The multi–grid methods for the diffusion equation with strongly discontinuous coefficients*, SIAM J. Sci. Stat. Comput., 2 (1981), pp. 430–454.

[2] O. Axelsson, *On multigrid methods of the two-level type*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., Lectures Notes in Mathematics No. 960, Berlin Heidelberg New York, 1981, Springer-Verlag, pp. 352–367.

[3] O. Axelsson, *The method of diagonal compensation of reduced matrix entries and multilevel iteration*, Journal of Computational and Applied Mathematics, 38 (1991), pp. 31–43.

[4] O. Axelsson, *Iterative Solution Methods*, University Press, Cambridge, 1994.

[5] O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems. Theory and Computation*, Academic Press, New York, 1984.

[6] O. Axelsson and V. Eijkhout, *Analysis of recursive 5-point/9-point factorization method*, in Preconditioned Conjugate Gradient Methods, O. Axelsson and L. Kolotilina, eds., Lectures Notes in Mathematics No. 1457, Springer-Verlag, 1990, pp. 154–173.

[7] O. Axelsson and V. Eijkhout, *The nested recursive two level factorization for nine-point difference matrices*, SIAM J. Sci. Stat. Comput., 12 (1991), pp. 1373–1400.

[8] O. Axelsson and I. Gustafsson, *Preconditioning and two-level multigrid methods of arbitrary degree of approximation*, Math. Comp., 40 (1983), pp. 214–242.

[9] O. Axelsson and G. Lindskog, *On the eigenvalue distribution of a class of preconditioning methods*, Numer. Math., 48 (1986), pp. 479–498.

[10] O. Axelsson and M. Neytcheva, *Algebraic multilevel iterations for Stieltjes matrices*, Num. Lin. Alg. Appl., 1 (1994), pp. 213–236.

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

[12] O. Axelsson and P. S. Vassilevski, *Algebraic multilevel preconditioning methods, II.*, SIAM J. Numer. Anal., 27 (1990), pp. 1569–1590.

[13] R. E. Bank, *Hierarchical bases and the finite element method*, Acta Numerica, 5 (1996), pp. 1–43.

[14] R. E. Bank and T. F. Dupont, *Analysis of a two-level scheme for solving finite element equations*, Tech. Rep. CNA-159, Center for Numerical Analysis, The University of Texas at Austin, Texas, USA, 1980.

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

[16] Barrett, Berry, Chan, Demmel, Donato, Dongarra, Eijkhout, Pozo, Romine, and van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM Publications, Philadelphia, 1993.

[17] P. M. de Zeeuw, *Matrix–dependent prolongations and restrictions in a blackbox multigrid solver*, J. Comput. Appl. Math., 33 (1990), pp. 1–27.

[18] J. E. Dendy, *Black box multigrid*, J. Comput. Phys., 48 (1982), pp. 366–386.

[19] G. H. Golub and C. F. van Loan, *Matrix Computations*, The John Hopkins University Press, Baltimore, Maryland, 1996. third ed.

[20] W. Hackbusch, *Multi-grid Methods and Applications*, Springer, Berlin, 1985.

[21] R. Kettler, *Linear multigrid methods for numerical reservoir simulation*, PhD thesis, Tech. Univ. Delft, Delft, 1987.

[22] J. Maitre and F. Musy, *The contraction number of a class of two-level methods; an exact evaluation for some finite element subspaces and model problems*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., Lectures Notes in Mathematics No. 960, Berlin Heidelberg New York, 1981, Springer-Verlag, pp. 535–544.

[23] S. Nakamura, *Computational Methods in Engineering and Science*, John Wiley & Sons, New York, 1977.

[24] Y. Notay, *DRIC: a dynamic version of the RIC method*, J. Num. Lin. Alg. with Appl., 1 (1994), pp. 511–532.

[25] Y. Notay, *Optimal order preconditioning of finite difference matrices*, Tech. Rep. GANMN 97-02, Université Libre de Bruxelles, Brussels, Belgium, 1997. *http://homepages.ulb.ac.be/˜ynotay.*

[26] Y. Notay, *Using approximate inverses in algebraic multilevel methods*, Tech. Rep. GANMN 97-01, Université Libre de Bruxelles, Brussels, Belgium, 1997. *http://homepages.ulb.ac.be/˜ynotay.*

[27] A. Reusken, *Fourier analysis of a robust multigrid method for convection-diffusion equations*, Numer. Math., 71 (1995), pp. 365–398.

[28] A. Reusken, *A multigrid method based on incomplete Gaussian elimination*, J. Num. Lin. Alg. with Appl., 3 (1996), pp. 369–390.

[29] A. van der Ploeg, E. Botta, and F. Wubs, *Nested grids ILU-decomposition (NGILU)*, J. Comput. Appl. Math., 66 (1996), pp. 515–526.

[30] R. S. Varga, *Matrix iterative analysis*, Prentice Hall, Englewood Cliffs, 1962.

[31] P. Vassilevski, *Nearly optimal iterative methods for solving finite element elliptic equations based on the multilevel splitting of the matrix*, Tech. Rep. # 1989-09, Institute for Scientific Computation, University of Wyoming, Laramie, USA, 1989.

[32] P. Vassilevski, *Hybrid V-cycle algebraic multilevel preconditioners*, Math. Comp., 58 (1992), pp. 489–512.

[33] P. Vassilevski, *On two ways of stabilizing the hierarchical basis multilevel method*, SIAM Review, 39 (1997), pp. 1–34.

[34] C. Wagner, W. Kinzelbach, and G. Wittum, *Schur-complement multigrid*, Numer. Math., 75 (1997), pp. 523–545.

[35] P. Wesseling, *An Introduction to Multigrid Methods*, J. Wiley and Sons, Chichester, 1992.

[36] G. Wittum, *On the robustness of ILU-smoothing*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 699–717.

[37] H. Yserentant, *On the multi-level splitting of finite element spaces*, Numer. Math., 49 (1986), pp. 379–412.