

Recursive Krylov-based multigrid cycles

Yvan Notay^{1,*} and Panayot S. Vassilevski^{2 †}

¹ *Service de Métrologie Nucléaire, Université Libre de Bruxelles (C.P. 165/84),
50, Av. F.D. Roosevelt, B-1050 Brussels, Belgium.*

² *Center for Applied Scientific Computing,
UC Lawrence Livermore National Laboratory, PO. Box 808, L-560,
Livermore, CA 94551, USA.*

Preprint version of the paper appeared in
Numerical Linear Algebra with Applications (vol. 15, pp. 473–487, 2008)
<http://dx.doi.org/10.1002/nla.542>

Abstract

We consider multigrid cycles based on the recursive use of a two-grid method, in which the coarse-grid system is solved by $\mu \geq 1$ steps of a Krylov subspace iterative method. The approach is further extended by allowing such inner iterations only at levels of given multiplicity whereas V-cycle formulation is used at all other levels. For symmetric positive definite systems and symmetric multigrid schemes, we consider a flexible (or generalized) conjugate gradient method as Krylov subspace solver for both inner and outer iterations. Then, based on some algebraic (block-matrix) properties of the V-cycle multigrid viewed as preconditioner, we show that the method can have optimal convergence properties if μ is chosen sufficiently large. We also formulate conditions that guarantee both, optimal complexity and convergence, bounded independently of the number of levels. Our analysis shows that the method is at least as effective as the standard W-cycle, whereas numerical results illustrate that it can be much faster than the latter, and actually more robust than predicted by the theory.

Key words. recursive multilevel Krylov iterations, variable-step multilevel preconditioning, flexible conjugate gradients, multigrid, Krylov subspace method, conjugate gradients, preconditioning

AMS subject classification. 65N55, 65F10, 65F50

*Supported by the Belgian FNRS (“Maître de recherches”).

†This work was performed under the auspices of the U. S. Department of Energy by the University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

1 Introduction

We consider the iterative solution of large sparse linear systems

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

by multigrid (MG) methods [1, 2]. These methods are being used in increasingly complex situations. They were initially designed as stand alone solvers, and they are quite successful as such in many applications, for which one may reach the so-called “multigrid textbook efficiency” [2]. However, MG methods, especially their algebraic variants (AMG) originating in [3], see also [4] (and more recently ([5, 6, 7, 8], etc.), are nowadays used in applications for which such efficiency is yet to be achieved. One common way to somewhat improve their robustness, is to use them as preconditioners in a Krylov subspace iterative method [9, 10], for instance, in the conjugate gradient (CG) method if the system matrix is symmetric positive definite (SPD).

Now, this still may not be sufficient to provide fast convergence if the two-grid convergence factor is too large to allow convergence properties independent of the number of levels with standard V- or W-cycles. Moreover, in real life problems, it is often impossible to predict if such a situation will occur or not, and what type of cycle would be optimal. This motivates us to consider Krylov based MG-cycles (or K-cycle, for short). With these cycles, the MG method is still based on the recursive use of a two-grid method, but the needed coarse-grid solve is defined by a few steps of a Krylov subspace iterative method with the already defined (by recursion) MG method on the previous (coarser) level as preconditioner. If μ inner iterations are performed at each level, we have more specifically a K_μ -cycle preconditioner. Such an idea is not new; it has been used, also in a multilevel setting, for the so called AMLI methods (cf., [11]). The latter can be viewed as stabilized versions of the hierarchical basis (or HB) methods. The stabilization comes from the more than one recursive calls of the preconditioner defined (by recursion) at a given level.

Observe that the MG preconditioner defined in this way becomes a nonlinear operator and thus the analysis of such techniques is not as straightforward. For this reason, we restrict ourselves to the simpler SPD case. That is, we assume that the matrix A in (1.1) is SPD, and that both inner and outer iterations are carried out with a CG type method, properly generalized to cope with nonlinear preconditioning (see [12] or §3 below). Moreover, we focus on MG schemes that preserve symmetry (see §2 for details). Nevertheless, we stress that in practice the approach is applicable to nonsymmetric problems as well, using as Krylov subspace iterative method for instance the variants of GMRES from [13] or from [14].

2 The K-cycle MG

We first introduce some notation and give the general setting of this study.

We consider a MG method with $\ell + 1$ levels; ℓ is the index of the finest level and 0 the index of the coarsest level; n_k , $k = 0, \dots, \ell$ is the number of unknowns at level k

(with $n_\ell = n$), and P_k , $k = 1, \dots, \ell$ is the $n_k \times n_{k-1}$ matrix used to interpolate a vector from $\mathfrak{R}^{n_{k-1}}$ onto \mathfrak{R}^{n_k} ; because we confine ourselves to symmetric schemes, the restriction is assumed to be the transpose of the interpolation.

The k th level $n_k \times n_k$ matrix A_k , $k = 0, \dots, \ell-1$ is assumed to be SPD. Note that we do not need any additional assumption. For example, our analysis is not restricted to Galerkin coarse-grid matrices, i.e., we do not assume that $A_{k-1} = P_k^T A_k A_k$. For convenience, we set $A_\ell = A$.

We assume that the k th level smoother M_k , $k = 1, \dots, \ell$ is an $n_k \times n_k$ nonsingular matrix such that it provides a convergent method in A_k -norm. I.e., we assume that $\|I - M_k^{-1} A_k\|_{A_k} < 1$. This assumption, as is easily seen (cf., e.g., [15, 16]), is equivalent to the statement,

$$M_k + M_k^T - A_k \text{ is SPD.} \quad (2.1)$$

The number of pre- and post-smoothing steps is denoted by ν_k . The smoother may be non symmetric, but we assume that pre- and post-smoothing are applied in a symmetric way. That is, pre-smoothing is performed with M_k and post-smoothing with M_k^T .

With these definitions, the action of a two- or multigrid preconditioner at level k on a given vector \mathbf{r}_k is computed according to the following algorithm.

Algorithm 2.1 (MG as preconditioner at level k ($k \geq 1$))

Input: \mathbf{r}_k ; *Output:* \mathbf{z}_k .

1. Relax ν_k times using smoother M_k : $\mathbf{v}_k = (I - (I - M_k^{-1} A_k)^{\nu_k}) A_k^{-1} \mathbf{r}_k$
2. Compute new residual: $\tilde{\mathbf{r}}_k = \mathbf{r}_k - A_k \mathbf{v}_k$
3. Restrict residual: $\mathbf{r}_{k-1} = P_k^T \tilde{\mathbf{r}}_k$
4. Compute an (approximate) solution \mathbf{y}_{k-1} to $A_{k-1} \mathbf{x}_{k-1} = \mathbf{r}_{k-1}$
5. Interpolate coarse-grid correction: $\mathbf{y}_k = P_k \mathbf{y}_{k-1}$
6. Compute new residual: $\bar{\mathbf{r}}_k = \tilde{\mathbf{r}}_k - A_k \mathbf{y}_k$
7. Relax ν_k times using smoother M_k^T : $\mathbf{w}_k = (I - (I - M_k^{-T} A_k)^{\nu_k}) A_k^{-1} \bar{\mathbf{r}}_k$
8. $\mathbf{z}_k = \mathbf{v}_k + \mathbf{y}_k + \mathbf{w}_k$

If an exact solution is computed at step 4, Algorithm 2.1 computes the action $\mathbf{z}_k = B_k^{-1} \mathbf{r}_k$ of the two-grid preconditioner B_k satisfying the relation

$$I - B_k^{-1} A_k = (I - M_k^{-T} A_k)^{\nu_k} (I - P_k A_{k-1}^{-1} P_k^T A_k) (I - M_k^{-1} A_k)^{\nu_k} . \quad (2.2)$$

Note that our general assumptions imply that B_k is SPD (see §4 for a proof).

Now, in practice, an exact solution is typically computed at step 4 only for $k = 1$ (that is, only the matrix A_0 on the coarsest grid is inverted exactly). For $k > 1$, the MG

preconditioner at level k computes \mathbf{y}_{k-1} approximately using the MG preconditioner at level $k-1$. The way this is done defines the so-called cycling strategy. Standard cycles are V- and W-cycles; V-cycles use only one action of the preconditioner on the coarser level, whereas W-cycles perform two stationary iterations, see, e.g., [2] for algorithms and more details.

Here we define the K_μ -cycle MG preconditioner as the preconditioner implemented by Algorithm 2.1 when the coarse-grid system at step 4 is solved by μ iterations of a Krylov subspace iterative method using the K_μ -cycle preconditioner on the coarser level, until level $k=1$ where the two-grid preconditioner B_1 is used. At level k , we denote this preconditioner $\mathcal{K}_\mu^{(k)}$. In general this is a nonlinear operator, thus for the vector computed by Algorithm 2.1 we write $\mathbf{z}_k = \mathcal{K}_\mu^{(k)}[\mathbf{r}_k]$.

In some cases, it is better to allow inner iterations only at levels of given multiplicity $k_0 > 1$, that is, at levels $\ell-k_0, \ell-2k_0, \dots$. At other levels, inner iterations are skipped and a simple V-cycle formulation is used. The $K_\mu^{(k,k_0,\ell)}$ -cycle MG preconditioner $\mathcal{K}_\mu^{(k,k_0,\ell)}[\cdot]$ is then defined as follows. For $k=1$, this is as above the two-grid preconditioner: $\mathcal{K}_\mu^{(1,k_0,\ell)}[\cdot] = B_1^{-1}$. For $k=2, 3, \dots$, $\mathcal{K}_\mu^{(k,k_0,\ell)}$ is recursively defined as the preconditioner implemented by Algorithm 2.1 at level k with, at step 4,

- if $\text{mod}(\ell - (k-1), k_0) = 0$, then \mathbf{y}_{k-1} is computed by solving $A_{k-1}\mathbf{x}_{k-1} = \mathbf{r}_{k-1}$ with μ iterations of a Krylov subspace iterative method using preconditioner $\mathcal{K}_\mu^{(k-1,k_0,\ell)}$.
- otherwise, we let $\mathbf{y}_{k-1} = \mathcal{K}_\mu^{(k-1,k_0,\ell)}[\mathbf{r}_{k-1}]$.

Note that with $k_0 = 1$ one recovers the previous definition: $\mathcal{K}_\mu^{(k,1,\ell)} = \mathcal{K}_\mu^{(k)}$. Note also that these definitions allow any Krylov subspace iterative method for the sake of generality. Because we focus on symmetric MG schemes for SPD matrices, in the following sections we restrict ourselves to the flexible (or generalized) CG method from [12] whose algorithm and convergence properties are summarized in the next section.

Finally, observe that, for $k = \ell - k_0, \ell - 2k_0, \dots$ and $k > k_0$, the $K_\mu^{(k,k_0,\ell)}$ -cycle MG preconditioner approximates the V-cycle preconditioner at level k with exact coarse grid solve at level $k-k_0$. For future reference, we then define $B_{k \mapsto j}$, the V-cycle preconditioner at level k with exact coarse grid solve at level j ($k > j$): $B_{j+1 \mapsto j} = B_{j+1}$ (the two-grid preconditioner defined by (2.2)) and, for $k = j+2, j+3, \dots$, $B_{k \mapsto j}$ is recursively defined as the preconditioner implemented by Algorithm 2.1 at level k with, at step 4, $\mathbf{y}_{k-1} = B_{k-1 \mapsto j}^{-1}\mathbf{r}_{k-1}$. Note that if $\text{mod}(\ell - k, k_0) = 0$ and $k \leq k_0$, then $\mathcal{K}_\mu^{(k,k_0,\ell)} = B_{k \mapsto 0}$.

3 Flexible (or Generalized) CG

Here we consider the version of the CG method suitable for nonlinear preconditioning that first appeared in [12]. Its analysis has then been improved in several papers [17, 18, 19]. We give below the algorithm to solve a linear system $A\mathbf{x} = \mathbf{b}$ with preconditioner action denoted $\mathbf{w} = \mathcal{B}(\mathbf{r})$. In the context of a K-cycle MG preconditioner, this algorithm will be invoked at step 4 of Algorithm 2.1 with $A = A_{k-1}$, $\mathbf{b} = \mathbf{r}_{k-1}$, $\mathcal{B} = \mathcal{K}_\mu^{(k-1,k_0,\ell)}$, and

the computed solution \mathbf{x}_{i+1} at step 3 of the final iteration will give the needed \mathbf{y}_{k-1} . This algorithm will then also be helpful as outer solver for the main system (1.1), since the top level preconditioner $\mathcal{K}_\mu^{(\ell, k_0, \ell)}$ is also nonlinear. For the sake of simplicity we assume that the initial approximation is always the zero vector.

Algorithm 3.1 (Flexible (or Generalized) CG)

Initialization: Let $\mathbf{x}_0 = 0$, $\mathbf{r}_0 = \mathbf{b}$.

For $i = 0, 1, \dots$ perform the following steps:

$$(1) \mathbf{w}_i = \mathcal{B}(\mathbf{r}_i)$$

$$(2) \mathbf{d}_i = \mathbf{w}_i - \sum_{k=i-\max(m_i, i)}^{i-1} \frac{\mathbf{w}_i^T A \mathbf{d}_k}{\mathbf{d}_k^T A \mathbf{d}_k} \mathbf{d}_k$$

$$(3) \mathbf{x}_{i+1} = \mathbf{x}_i + \frac{\mathbf{d}_i^T \mathbf{r}_i}{\mathbf{d}_i^T A \mathbf{d}_i} \mathbf{d}_i$$

$$(4) \mathbf{r}_{i+1} = \mathbf{r}_i - \frac{\mathbf{d}_i^T \mathbf{r}_i}{\mathbf{d}_i^T A \mathbf{d}_i} A \mathbf{d}_i$$

*(5) **Exit** if maximum number of iteration is reached or stopping test satisfied.*

In this algorithm, the m_i s are given parameters. Setting $m_i = 0$ for all i gives the steepest descent method. If $m_i = 1$ for all i and if the preconditioner \mathcal{B} corresponds to a SPD matrix, then Algorithm 3.1 reduces to the CG method, in an implementation that is slightly more costly than the standard one (one more inner product to compute per iteration; there is also one more vector to store).

In case of variable or nonlinear preconditioning, the following optimality property is still satisfied:

$$\|A^{-1}\mathbf{b} - \mathbf{x}_{i+1}\|_A = \min_{\mathbf{d} \in \text{span}\{\mathbf{d}_{i-m_i}, \dots, \mathbf{d}_i\}} \|A^{-1}\mathbf{b} - \mathbf{x}_{i-m_i} - \mathbf{d}\|_A. \quad (3.1)$$

Hence using larger m_i helps to maintain global optimality, at the price of increasing cost and storage. In practice however, when \mathcal{B} is close to a SPD matrix B^{-1} such that $B^{-1}A$ has no small or large isolated eigenvalues, the convergence is often not improved setting m_i larger than 1 [19]; $m_i = 1$ is then the most cost effective. In our numerical experiments, we always set $m_i = 1$ except when we want to assess the steepest descent variant ($m_i = 0$). Note however that in the context of inner iterations for the K-cycle MG preconditioner, at most few iterations are allowed (typically 2 or 3), so that $m_i = i$ would not affect significantly the cost of the algorithm.

Taking the best from the estimates in [17, 19], the proved convergence properties of Algorithm 3.1 can be stated as follows. If, for a given SPD matrix B , one has

$$\frac{\|\mathcal{B}(\mathbf{r}_i) - B^{-1} \mathbf{r}_i\|_B}{\|B^{-1} \mathbf{r}_i\|_B} \leq \varepsilon \quad (3.2)$$

for some $\varepsilon < 1$, then

$$\frac{\|\mathbf{x} - \mathbf{x}_{i+1}\|_A}{\|\mathbf{x} - \mathbf{x}_i\|_A} \leq \min \left(\sqrt{1 - \frac{4\kappa(1-\varepsilon)^2}{(\kappa + \varepsilon^2(\kappa-1) + (1-\varepsilon)^2)^2}}, \sqrt{1 - \frac{1-\varepsilon^2}{\kappa}} \right), \quad (3.3)$$

where

$$\kappa = \frac{\lambda_{\max}(B^{-1}A)}{\lambda_{\min}(B^{-1}A)} \quad (3.4)$$

is the ratio of the extremal eigenvalues of $B^{-1}A$.

This is a “local” convergence rate valid for any m_i , including $m_i = 0$. It is therefore likely to be pessimistic for larger m_i . The analysis in [18] takes into account the global behavior of the algorithm, but the resulting bound is smaller than 1 only for very small ε , and is therefore not helpful in the context of the present study.

Note that for ε approaching zero, the first term in the right hand side of (3.3) (the bound from [19]) gives the standard bound for the convergence of the steepest descent method. The second term (the bound from [17]) overestimates the error for small ε but is on the other hand fairly insensitive to ε .

The fact that we are not able to develop a specific analysis for $m_i \geq 1$ may be seen as a shortcoming inherent to nonlinearities, since, in the linear case (corresponding to $\varepsilon = 0$), it is known that the CG method is significantly faster than the steepest descent method.

4 MG as block-factorization

In order to be able to analyze the K-cycle MG preconditioner we recall some linear algebra properties of the two-grid method seen as a preconditioner. More details can be found in the monograph [20].

In the present section, for simplicity, we omit the subscripts. That is, we write A as A_l and B as B_l . Then, A_c refers to A_{l-1} and similarly B_c refers to B_{l-1} . We also omit subscripts for vectors, smoother, interpolation, etc.

The first useful fact is that the two-grid preconditioner can be represented as certain block-factorization of the original matrix. To this end, we introduce the block-factored matrix

$$\overline{B} = \begin{bmatrix} M & 0 \\ P^T A & I \end{bmatrix} \begin{bmatrix} (M + M^T - A)^{-1} & 0 \\ 0 & B_c \end{bmatrix} \begin{bmatrix} M^T & AP \\ 0 & I \end{bmatrix},$$

where B_c may be any SPD approximation to A_c . Note that \overline{B} is a $(n + n_c) \times (n + n_c)$ matrix, whereas A is $n \times n$ and A_c is $n_c \times n_c$. It is clear that \overline{B} is SPD if $M + M^T - A$ is SPD, which in fact follows from the assumption $\|I - M^{-1}A\|_A < 1$ (stated in (2.1), and proved, e.g., in [15, 16]). Next, consider the following $n \times n$ matrix,

$$B^{-1} = [I, P] \overline{B}^{-1} [I, P]^T. \quad (4.1)$$

Then the following identity holds (which can be verified by straightforward computation, see for example, [21])

$$I - B^{-1}A = (I - M^{-T}A)(I - PB_c^{-1}P^T A)(I - M^{-1}A).$$

That is, the SPD matrix B defined from (4.1) takes part in the definition of the error propagation matrix $I - B^{-1}A$, which is the product of three processes; namely, (pre-)smoothing based on M , coarse-grid correction based on B_c , and post-smoothing based on M^T . In the case of more smoothing steps, we first define a composite smoother \widehat{M} from the equation $I - \widehat{M}^{-1}A = (I - M^{-1}A)^\nu$ and modify the above definition of \overline{B} by replacing M with the composite one \widehat{M} .

This holds with various choices of B_c . For example, with $B_c = A_c$ we obtain definitions for exact two-grid preconditioner, or if B_c is defined recursively (from coarse-to-fine levels) by approximate solution of $A_c \mathbf{x}_c = \mathbf{r}_c$ (as in step 4 of Algorithm 2.1) we end up with the multilevel versions. Of our interest is the case when in step 4 we apply a flexible CG preconditioned with a recursively defined nonlinear multilevel preconditioner \mathcal{B}_c (which approximates A_c^{-1}). Such nonlinear \mathcal{B}_c is considered in Lemma 4.1 below. Before stating it we rewrite formula (4.1) in the following more explicit form

$$B^{-1} = \overline{M}^{-1} + (I - M^{-T}A)PB_c^{-1}P^T(I - AM^{-1}). \quad (4.2)$$

Here, $\overline{M} = M(M + M^T - A)^{-1}M^T$ is the so-called symmetrized smoother. Note that \overline{M} is SPD.

Lemma 4.1 *Let B_c be a given SPD $n_c \times n_c$ matrix and let $\mathcal{B}_c[\cdot]$ be a mapping that for some tolerance $\delta \in [0, 1)$ approximates B_c^{-1} in the following sense*

$$\|B_c^{-1}\mathbf{v}_c - \mathcal{B}_c[\mathbf{v}_c]\|_{B_c} \leq \delta \|\mathbf{v}_c\|_{B_c^{-1}} \quad \forall \mathbf{v}_c \in \mathbb{R}^{n_c}.$$

Let $\mathcal{B}[\cdot]$ be the mapping defined in the same way as B^{-1} where the actions $B_c^{-1}\mathbf{v}_c$ are replaced by $\mathcal{B}_c[\mathbf{v}_c]$. Then, for all $\mathbf{v} \in \mathbb{R}^n$, the following inequalities hold, letting $\mathbf{v}_c = P^T(I - AM^{-1})\mathbf{v}$,

$$\|B^{-1}\mathbf{v} - \mathcal{B}[\mathbf{v}]\|_B \leq \|B_c^{-1}\mathbf{v}_c - \mathcal{B}_c[\mathbf{v}_c]\|_{B_c} \leq \delta \|\mathbf{v}_c\|_{B_c^{-1}} \leq \delta \|\mathbf{v}\|_{B^{-1}}.$$

Proof. We have, from the representation of $B^{-1}\mathbf{v}$ in (4.2) and the fact that \overline{M}^{-1} is SPD, that (letting $\mathbf{v}_c = P^T(I - AM^{-1})\mathbf{v}$),

$$\mathbf{v}^T B^{-1} \mathbf{v} = \mathbf{v}^T \overline{M}^{-1} \mathbf{v} + (P^T(I - AM^{-1})\mathbf{v})^T B_c^{-1} (P^T(I - AM^{-1})\mathbf{v}) \geq \mathbf{v}_c^T B_c^{-1} \mathbf{v}_c. \quad (4.3)$$

Similarly to (4.2), the following expression holds for $\mathcal{B}[\mathbf{v}]$, i.e.,

$$\mathcal{B}[\mathbf{v}] = \overline{M}^{-1} \mathbf{v} + (I - M^{-T}A)P\mathcal{B}_c[P^T(I - AM^{-1})\mathbf{v}].$$

Then, using Cauchy-Schwarz inequality, (4.3), the assumption on $\mathcal{B}_c[\cdot]$ and once more

(4.3), we arrive at the desired result,

$$\begin{aligned}
\|B^{-1}\mathbf{v} - \mathcal{B}[\mathbf{v}]\|_B &= \sup_{\mathbf{w}} \frac{\mathbf{w}^T B(I - M^{-T}A)P(B_c^{-1}\mathbf{v}_c - \mathcal{B}_c[\mathbf{v}_c])}{\|\mathbf{w}\|_B} \\
&\leq \sup_{\mathbf{w}} \frac{\|B_c^{-\frac{1}{2}}P^T(I - AM^{-1})B\mathbf{w}\|}{\|\mathbf{w}\|_B} \|B_c^{-1}\mathbf{v}_c - \mathcal{B}_c[\mathbf{v}_c]\|_{B_c} \\
&= \sup_{\mathbf{w}} \frac{\|B_c^{-\frac{1}{2}}P^T(I - AM^{-1})\mathbf{w}\|}{\|\mathbf{w}\|_{B^{-1}}} \|B_c^{-1}\mathbf{v}_c - \mathcal{B}_c[\mathbf{v}_c]\|_{B_c} \\
&\leq \|B_c^{-1}\mathbf{v}_c - \mathcal{B}_c[\mathbf{v}_c]\|_{B_c} \\
&\leq \delta \|\mathbf{v}_c\|_{B_c^{-1}} \\
&\leq \delta \|\mathbf{v}\|_{B^{-1}}.
\end{aligned}$$

■

The following corollary holds then.

Corollary 4.1 *Consider the V -cycle preconditioner $B_{k \mapsto j}$ as defined in §2, for some k, j such that $k > j$. Let $\mathcal{B}_j[\cdot]$ be an approximate inverse to A_j , and let $\mathcal{B}_{k \mapsto j}$ be the V -cycle preconditioner from level k to j in which the exact solve at level j is replaced by the action of \mathcal{B}_j . That is, for $i = j + 1, \dots, k$, $\mathcal{B}_{i \mapsto j}$ is the preconditioner implemented by Algorithm 2.1 at level i with, at step 4, $\mathbf{y}_{i-1} = \mathcal{B}_j[\mathbf{r}_{i-1}]$ if $i = j + 1$ and $\mathbf{y}_{i-1} = \mathcal{B}_{i-1 \mapsto j}[\mathbf{r}_{i-1}]$ if $i > j + 1$. Then, the following deviation estimate holds*

$$\|B_{k \mapsto j}^{-1}\mathbf{v} - \mathcal{B}_{k \mapsto j}[\mathbf{v}]\|_{B_{k \mapsto j}} \leq \delta \|\mathbf{v}\|_{B_{k \mapsto j}^{-1}} \quad \forall \mathbf{v} \in \mathbb{R}^{n_k},$$

provided that at the initial coarse level j , $\|A_j^{-1}\mathbf{v}_j - \mathcal{B}_j[\mathbf{v}_j]\|_{A_j} \leq \delta \|\mathbf{v}_j\|_{A_j^{-1}}$ for a given tolerance δ .

Proof. For $k = j + 1$, this is just Lemma 4.1 with $B_c = A_c = A_j$, properly extended when $\nu > 1$ (using \widehat{M} from $I - \widehat{M}^{-1}A = (I - M^{-1}A)^\nu$ instead of M). On the other hand, if the corollary holds for some $k > j$, Lemma 4.1, with $B = B_{k+1 \mapsto j}$, $B_c = B_{k \mapsto j}$, $\mathcal{B} = \mathcal{B}_{k+1 \mapsto j}$ and $\mathcal{B}_c = \mathcal{B}_{k \mapsto j}$, further shows

$$\|B_{k+1 \mapsto j}^{-1}\mathbf{v} - \mathcal{B}_{k+1 \mapsto j}[\mathbf{v}]\|_{B_{k+1 \mapsto j}} \leq \|B_{k \mapsto j}^{-1}\mathbf{v}_c - \mathcal{B}_{k \mapsto j}[\mathbf{v}_c]\|_{B_{k \mapsto j}} \leq \delta \|\mathbf{v}_c\|_{B_{k \mapsto j}^{-1}} \leq \delta \|\mathbf{v}\|_{B_{k+1 \mapsto j}^{-1}},$$

that it is, the result also holds for $k + 1$. ■

5 Analysis of K_μ cycle MG

The convergence properties of the K_μ MG now easily follow from Corollary 4.1 based on the convergence rate estimates (as in (3.3)) for the flexible (generalized) preconditioned CG method given by Algorithm 3.1 (with $\mu \geq 1$ iterations).

For the sake of simplicity, we formulate a convergence result for K_μ -cycle MG, that is, for Algorithm 2.1 with flexible CG iterations at every level but the coarsest one. The more general case of a $K_\mu^{(k, k_0, \ell)}$ -cycle MG (with $k_0 > 1$) is analyzed similarly, see the comments below.

Theorem 5.1 Consider a K_μ -cycle MG preconditioner as defined in §2, with coarse-grid systems at step 4 of Algorithm 2.1 solved by Algorithm 3.1 (with $\mu \geq 1$ iterations). Let κ_k , $k = 1, \dots, \ell$ be the condition number of A_k preconditioned by the two-grid method:

$$\kappa_k = \frac{\lambda_{\max}(B_k^{-1}A_k)}{\lambda_{\min}(B_k^{-1}A_k)},$$

where B_k is defined from (2.2). Let ε_k be defined by

$$\varepsilon_k = \min \left(\sqrt{1 - \frac{4\kappa_k(1-\varepsilon_{k-1}^\mu)^2}{(\kappa_k + \varepsilon_{k-1}^{2\mu}(\kappa_k - 1) + (1-\varepsilon_{k-1}^\mu)^2)^2}}, \sqrt{1 - \frac{1-\varepsilon_{k-1}^{2\mu}}{\kappa_k}} \right), \quad k = 1, \dots, \ell, \quad (5.1)$$

where $\varepsilon_0 = 0$.

Then, each iteration of Algorithm 3.1 to solve the system (1.1) with this K_μ -cycle MG preconditioner is such that

$$\frac{\|\mathbf{x} - \mathbf{x}_{i+1}\|_A}{\|\mathbf{x} - \mathbf{x}_i\|_A} \leq \varepsilon_\ell. \quad (5.2)$$

If, in addition,

$$\kappa = \max_{1 \leq k \leq \ell} \kappa_k < \mu \quad (5.3)$$

then there exists ε , $0 \leq \varepsilon < 1$, satisfying

$$\min \left(\sqrt{1 - \frac{4\kappa(1-\varepsilon^\mu)^2}{(\kappa + \varepsilon^{2\mu}(\kappa - 1) + (1-\varepsilon^\mu)^2)^2}}, \sqrt{1 - \frac{1-\varepsilon^{2\mu}}{\kappa}} \right) \leq \varepsilon, \quad (5.4)$$

and for any such ε ,

$$\varepsilon_\ell \leq \varepsilon \quad (5.5)$$

independently of ℓ . Moreover, the smallest such ε is not larger than the only positive root of

$$\sum_{j=1}^{\mu-1} \varepsilon^{2j} + 1 - \kappa = 0. \quad (5.6)$$

Proof. Inequality (5.2) with ε_ℓ from (5.1) follows straightforwardly from the recursive application of (3.3) and Corollary 4.1. Because the right hand side of (5.1) is an increasing function of both κ_k and ε_{k-1} , one further sees that (5.5) holds for any ε satisfying (5.4).

In particular, the latter inequality is satisfied when $\varepsilon = \sqrt{1 - \frac{1-\varepsilon^{2\mu}}{\kappa}}$, that is when

$$\kappa = \frac{1 - \varepsilon^{2\mu}}{1 - \varepsilon^2} = \sum_{j=0}^{\mu-1} \varepsilon^{2j},$$

which amounts to (5.6). Finally, the polynomial in the left hand side of (5.6) is monotonically increasing for $\varepsilon > 0$ and takes value $1 - \kappa \leq 0$ for $\varepsilon = 0$, showing that there is a

unique positive root; moreover, because the polynomial takes value $\mu - \kappa > 0$ for $\varepsilon = 1$, this root has to be smaller than 1. \blacksquare

A similar result holds for $K_\mu^{(k,k_0,\ell)}$ -cycle MG. Then, the statement of the convergence result is the same as in Theorem 5.1 with $\kappa = \kappa_{k_0}$ now referring to a uniform bound (by assumption) of the relative condition number of the V-cycle preconditioner $B_{k \mapsto \max(k-k_0,0)}$ with respect to A_k , for $k = \ell, \ell - k_0, \dots$.

An application to second order elliptic problems

In practice, we are interested in the complexity of the multilevel methods. More specifically we want to have one action of the multilevel preconditioner be of optimal complexity, i.e., proportional to the total number of degrees of freedom (at the finest level). In a typical geometrical MG situation, the degrees of freedom at level l grow like $n_l \simeq n_0 2^{dl}$, where $d = 2$ or $d = 3$ is the dimension of the geometrical domain and 2 stands for the refinement factor of the respective mesh size. It is straightforward to estimate that the cost of flexible CG iteration with $K_\mu^{(k,k_0,\ell)}$ -cycle MG preconditioner:

$$w_k = \mathcal{O}(n_k) + \mu w_{k-k_0}.$$

The above relations imply that

$$w_\ell = \mathcal{O}(n_\ell) \sum_j \left(\frac{\mu}{2^{dk_0}} \right)^j.$$

That is, in order to have an optimal complexity method we need that

$$\mu < 2^{dk_0}.$$

The latter is a very mild restriction on μ if we choose $k_0 \geq 1$ sufficiently large. The condition on μ to have an optimal convergence reads (see (5.3)),

$$\kappa_{k_0} < \mu.$$

Thus, the conditions to have an optimal order method (i.e., both optimal complexity and optimal level-independent bound of the convergence factor) read,

$$\kappa_{k_0} < \mu < 2^{dk_0}. \quad (5.7)$$

Recall, that κ_{k_0} stands for a uniform bound of the level V-cycle preconditioner $B_{k \mapsto k-k_0}$ with respect to A_k for $k = \ell, \ell - k_0, \dots$. For the case of matrices $\{A_l\}$ coming from model second order selfadjoint elliptic PDEs discretized on uniformly refined meshes with coefficients that may have large jumps across element boundaries on the coarsest level ($l = 0$) only, the following estimate is known

$$\kappa_{k_0} = \begin{cases} \mathcal{O}(1 + k_0^2), & d = 2, \\ \mathcal{O}(2^{k_0}), & d = 3. \end{cases}$$

The constant in the \mathcal{O} -factor is independent of the coefficient jumps. This is the result based on the hierarchical basis (or HB) method.

Based on the last asymptotic behavior of κ_{k_0} , it is clear that for k_0 sufficiently large (but fixed) we can choose $\mu \geq 1$ (in both cases $d = 2$ and $d = 3$) such that the conditions (5.7) for an optimal order multilevel method are met. This result can be viewed as an extension of the HB-based AMLI method with flexible (generalized) CG inner iterations originated in [11], see also [22], now in the case of K-cycle MG.

6 Numerical illustration

We first illustrate the behavior of the estimates in Theorem 5.1. Here we want to include a comparison with the case where, instead of a Krylov subspace method, one uses stationary iterations to solve the coarse grid systems, giving thus the standard W-cycle for $\mu = 2$ (as defined in, e.g., [2]), and generalized W-cycle for $\mu > 2$, which we call W_μ -cycle for short. For such cycles, if one assumes in addition that $\lambda_{\max}(B_k^{-1}A_k) = 1$ (as occurs when using Galerkin coarse-grid matrices, that is when $A_{k-1} = P_k^T A_k P_k$), the convergence factor σ_k of the multigrid method at level k can be recursively estimated from [23, eq. (3.2)]

$$\sigma_k \leq 1 - \kappa_k^{-1} (1 - \sigma_{k-1}^\mu) \quad (6.1)$$

for $k = 1, 2, \dots$, with $\sigma_0 = 0$. With outer steepest descent or CG iterations, one has then

$$\frac{\|\mathbf{x} - \mathbf{x}_{i+1}\|_A}{\|\mathbf{x} - \mathbf{x}_i\|_A} \leq \varepsilon_\ell = \frac{\sigma_\ell}{2 - \sigma_\ell} \quad (6.2)$$

(here we take into account that, for A preconditioned by W_μ -cycle MG, the largest eigenvalues is equal to 1, hence $\kappa = (1 - \sigma_\ell)^{-1}$, whereas the convergence rate for the steepest descent method is $(\kappa - 1)/(\kappa + 1)$).

On Figure 1 we illustrate these estimates for typical values of μ . One sees that, from the theoretical side, neither K-cycle nor W-cycle has a decisive advantage. However, we generally observed that the above analysis of W-cycle MG is relatively sharp, whereas K-cycle MG often behaves better than predicted by the theory, especially when κ_k is larger than μ , so that ε_ℓ quickly grows to 1 as the number of levels increases. This may be related to the fact that K-cycles lead to nonlinear operators, whose analysis is harder. As example of shortcoming, Theorem 5.1 is based on estimates (3.3) for flexible CG that do not take into account the further optimality property (3.1) for $m_i \geq 1$.

To illustrate this, we consider the following model experiment, in fact a one dimensional problem, where one selects either AG2 (simple aggregation with two nodes in each

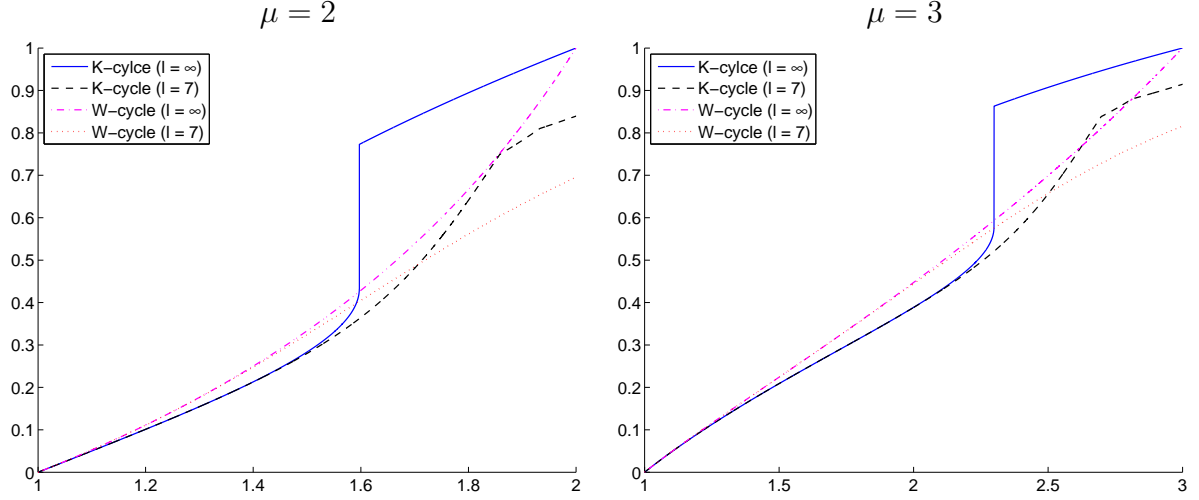


Figure 1: ε_ℓ from (5.1) (K-cycle) or (6.1), (6.2) (W-cycle) as a function of κ (with $\kappa_k = \kappa$ in (5.1),(6.1)); for the K-cycle, ε_∞ is also the smallest ε satisfying (5.4).

aggregate) or AG4 (the same with 4 nodes in each aggregate):

$$\begin{aligned}
n &= 2^{20} \\
A &= \text{tridiag}(-1, 2, -1) \\
\text{AG2} : \quad n_{\ell-k} &= 2^{-k} n \\
(P_k)_{ij} &= \begin{cases} 1 & \text{if } 2j - 1 \leq i \leq 2j \\ 0 & \text{otherwise} \end{cases} \\
\text{AG4} : \quad n_{\ell-k} &= 4^{-k} n \\
(P_k)_{ij} &= \begin{cases} 1 & \text{if } 4j - 3 \leq i \leq 4j \\ 0 & \text{otherwise} \end{cases} \\
A_{k-1} &= P_k^T A_k P_k \\
M_k &= \omega \text{diag}(A_k)^{-1} \quad (\text{damped Jacobi smoothing}) \\
\nu_k &= 1
\end{aligned}$$

This problem is artificial, but, because all matrices are tridiagonal, we are able to compute κ_k at each level and check that it remains approximately constant from $k = 1$ to $k = \ell$ with ℓ sufficiently large. Moreover, it is also possible to compute the energy norm of the error at each step.

We then proceed as follows. We select two right hand sides: the vector of all ones, and a vector with random entries uniformly distributed in $[0, 1]$. For each of these, we performed 100 steepest descent (outer) iterations (Algorithm 3.1 with $m_i = 0$) and pick

Cycle	ε_m	ε_ℓ	It1	It2	ε_m	ε_ℓ	It1	It2
AG2, $\omega = 0.5$ ($1.99 < \kappa_k < 2.00$)								
	$\ell = 7$				$\ell = 14$			
V	0.98		189	187	1.00		> 999	> 999
W ₂	0.69	0.70	37	37	0.78	0.81	50	50
W ₃	0.42	0.44	22	22	0.43	0.44	22	22
K ₂	0.30	0.84	20	20	0.30	0.93	20	20
K ₃	0.31	0.39	18	18	0.31	0.44	18	18
AG4, $\omega = 0.5$ ($4.86 < \kappa_k < 4.92$)								
	$\ell = 4$				$\ell = 7$			
V	0.99		256	256	1.00		> 999	> 999
W ₂	0.95	0.98	108	108	0.99	1.00	316	314
W ₃	0.87	0.94	70	70	0.96	0.99	120	120
K ₂	0.65	0.99	42	42	0.59	1.00	44	44
K ₃	0.61	0.96	33	33	0.61	0.99	33	33
AG4, $\omega = 0.3$ ($6.87 < \kappa_k < 6.92$)								
	$\ell = 4$				$\ell = 7$			
V	0.99		272	272	1.00		> 999	> 999
W ₂	0.97	0.99	122	122	1.00	1.00	340	340
W ₃	0.93	0.98	83	80	0.98	1.00	143	180
K ₂	0.85	1.00	72	72	0.83	1.00	84	84
K ₃	0.78	0.99	41	40	0.78	1.00	41	42

Table 1: Numerical results for $A = \text{tridiag}(-1, 2, -1)$; It1 refers to the right hand side of all ones and It2 to the random right hand side.

up ε_m as the worst ratio

$$\frac{\|\mathbf{x} - \mathbf{x}_{i+1}\|_A}{\|\mathbf{x} - \mathbf{x}_i\|_A}$$

from all these iterations. This quantity is then compared in Table 1 with the bound ε_ℓ obtained from either Theorem 5.1 (K_μ -cycle) or from (6.1), (6.2) (W_μ -cycle). In addition, we also report for both right hand sides the actual number of iterations needed to reduce the relative residual error below 10^{-12} when using the flexible CG method (Algorithm 3.1 with $m_i = 1$ for all i). Note that for inner iterations in K_μ -cycle MG, we always use Algorithm 3.1 with $m_i = 1$. For the sake of completeness, we also report the results obtained with V-cycle MG.

One sees that for W-cycles, the bound ε_ℓ gives a relatively sharp prediction of the actual convergence, whereas it is by far too pessimistic for K-cycle MG. As a consequence, the K_μ -cycle preconditioner appears to be more effective than the W_μ -cycle preconditioner

k_0		1		2	
Cycle		K ₂	W	K ₂	W
$n = 128^2$	$\ell = 3$	10	12	15	17
$n = 256^2$	$\ell = 4$	10	14	16	22
$n = 512^2$	$\ell = 5$	11	16	17	27
$n = 1024^2$	$\ell = 6$	11	18	18	34
$n = 2048^2$	$\ell = 7$	11	19	19	41

Table 2: Number of iterations needed to solve the discrete Poisson problem; “–” means no convergence within the limit of 1000 iterations.

for the same value of μ , that is for about the same computational complexity. In some cases, ε_m is slightly smaller for K₂ than for K₃, which we explain by the fact that the reported ε_m is only an approximation (from below) to $\bar{\varepsilon}_m = \sup_{\mathbf{x}-\mathbf{x}_i} \frac{\|\mathbf{x}-\mathbf{x}_{i+1}\|_A}{\|\mathbf{x}-\mathbf{x}_i\|_A}$.

On the other hand, K _{μ} -cycle MG appears fairly robust on this example. Even the relatively cheap variant K₂ is able to deliver practically grid independent convergence for κ_k up to 5, whereas the performances appear to depend only mildly on the number of levels for κ_k up to 7.

We next consider a model but more realistic problem, namely the five point finite difference approximation of

$$-\Delta u = 1$$

on the unit square with homogeneous Dirichlet boundary conditions everywhere and a uniform mesh size $h = 1/(m+1)$ in both directions, so that the order of the matrix is $n = m^2$. We consider a pure boxwise aggregation scheme with four node in each aggregate which is the natural extension to two dimensions of the AG2 scheme used above, see [24] for details. In the latter reference, it is proven that the two-grid condition number is between 2 and 4, hence too large for optimal convergence of the W-cycle.

Here we use symmetric Gauss-Seidel smoothing with $\nu_k = 1$ for all k , and the number of levels is in all cases such that the coarsest grid has exactly 256 nodes. We report in Table 2 the number of FCG iterations needed to reduce the relative residual error below 10^{-6} . Besides the standard K₂ and W cycles with inner iterations at each level, we also illustrate the possibility of doing inner iterations only at level of given multiplicity, setting $k_0 = 2$. The results confirm our previous conclusions: the K₂-cycle with $k_0 = 1$ appears optimal and for $k_0 = 2$ it tends to stabilize the condition number, whereas the W-cycle fails to do so.

7 Conclusions

We have introduced K-cycle MG and developed its analysis. According to this analysis, K-cycle and W-cycle MG should have similar convergence properties (see Figure 1). However,

numerical experiments reveal that for difficult problems, for which V-cycle is slow, K-cycle MG can be much more effective than W-cycle MG. This phenomenon can be explained as follows. The W-cycle theory is based on a linear analysis which is relatively sharp. Numerical evidence shows that it is unlikely for the W-cycle theory to be too pessimistic in practical examples. On the other hand, there are shortcomings in the analysis of the K-cycle that are inherent to the nonlinearity of the method (see the end of Section 3). This gives rise to pessimistic bounds and numerical experiments where K-cycle MG is much more effective than W-cycle MG correspond actually to cases where K-cycle MG is also much more effective than predicted by the theory.

Therefore, K-cycle MG appears more robust than W-cycle MG. It can exhibit convergence properties independent of the number of levels even when the condition number for the underlying two-grid method is relatively large. Using K-cycles may thus enhance the robustness of a MG method, in particular that of AMG schemes for real-life problems. This enhanced robustness is obtained nearly for free since the K-cycle has roughly the same computational complexity as the W-cycle.

Finally, sometimes the number of unknowns does not decrease sufficiently fast from one level to the next to allow inner iterations at each level as foreseen with standard K- or W-cycles. To cope with such cases, we introduced a variant of K-cycle MG that allows inner iterations only at levels of given multiplicity $k_0 > 1$, whereas a V-cycle formulation is used at other levels. We showed that the analysis of the standard K-cycle carries over this case. The K-cycling strategy can be implemented in virtually any MG method selecting the length k_0 of the underlying V-cycle in such a way that n_{k-k_0}/n_k is sufficiently small thus keeping the complexity of one such recursive cycle under control.

References

- [1] W. Hackbusch, *Multi-grid Methods and Applications*, Springer, Berlin, 1985.
- [2] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.
- [3] A. Brandt, S. F. McCormick, and J. W. Ruge, ‘Algebraic multigrid (amg) for sparse matrix equations’, in *Sparsity and its Application*, ed., D. J. Evans, 257–284, Cambridge University Press, Cambridge, (1984).
- [4] J. W. Ruge and K. Stüben, ‘Efficient solution of finite difference and finite element equations by algebraic multigrid (AMG)’, in *Multigrid Methods for Integral and Differential Equations*, eds., D. J. Paddon and H. Holstein, The Institute of Mathematics and its Applications Conference Series, 169–212, Clarendon Press, Oxford, (1985).
- [5] K. Stüben, *An Introduction to Algebraic Multigrid*, 413–532. In Trottenberg et al. [2], 2001. Appendix A.

- [6] M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge, ‘Algebraic multigrid based on element interpolation (AMGe)’, *SIAM Journal on Scientific Computing*, **22**, 1570–1592, (2000).
- [7] T. Chartier, R. D. Falgout, V. E. Henson, J. Jones, T. Manteuffel, S. McCormick, J. Ruge, and P. S. Vassilevski, ‘Spectral AMGe (ρ AMGe)’, *SIAM Journal on Scientific Computing*, **25**, 1–26, (2004).
- [8] Andrew J. Cleary, Robert D. Falgout, Van Emden Henson, Jim E. Jones, Thomas A. Manteuffel, Stephen F. McCormick, Gerald N. Miranda, and John W. Ruge, ‘Robustness and scalability of algebraic multigrid’, *SIAM Journal on Scientific Computing*, **21**, 1886–1908, (2000).
- [9] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, PA, 2003. Second ed.
- [10] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear systems*, Cambridge University Press, Cambridge, 2003.
- [11] O. Axelsson and P. S. Vassilevski, ‘Variable-step multilevel preconditioning methods. I. selfadjoint and positive definite elliptic problems’, *Numerical Linear Algebra with Applications*, **1**, 75–101, (1994).
- [12] O. Axelsson and P. S. Vassilevski, ‘A black-box generalized conjugate gradient solver with inner iterations and variable-step preconditioning’, *SIAM Journal on Matrix Analysis and Applications*, **12**, 625–644, (1991).
- [13] Y. Saad, ‘A flexible inner-outer preconditioned GMRES algorithm’, *SIAM Journal on Scientific Computing*, **14**, 461–469, (1993).
- [14] H. A. van der Vorst and C. Vuik, ‘GMRESR: a family of nested GMRES methods’, *Numerical Linear Algebra with Applications*, **1**(4), 369–386, (1994).
- [15] Robert D. Falgout and Panayot S. Vassilevski, ‘On generalizing the algebraic multigrid framework’, *SIAM Journal on Numerical Analysis*, **42**, 1669–1693, (2005).
- [16] Y. Notay, ‘Algebraic multigrid and algebraic multilevel methods: a theoretical comparison’, *Numerical Linear Algebra with Applications*, **12**, 419–451, (2005).
- [17] Radim Blaheta, ‘GPCG-generalized preconditioned CG method and its use with non-linear and non-symmetric displacement decomposition preconditioners’, *Numerical Linear Algebra with Applications*, **9**, 527–550, (2002).
- [18] G. H. Golub and Q. Ye, ‘Inexact preconditioned conjugate gradient method with inner-outer iterations’, *SIAM Journal on Scientific Computing*, **21**, 1305–1320, (1999).

- [19] Y. Notay, ‘Flexible conjugate gradients’, *SIAM Journal on Scientific Computing*, **22**, 1444–1460, (2000).
- [20] P. S. Vassilevski, *Multilevel Block Factorization Preconditioners*, Springer, New York, 2007. (to appear).
- [21] Robert D. Falgout, Panayot S. Vassilevski, and Ludmil T. Zikatanov, ‘On two-grid convergence estimates’, *Numerical Linear Algebra with Applications*, **12**, 471–494, (2005).
- [22] J. K. Kraus, ‘An algebraic preconditioning method for M -matrices: linear versus nonlinear multilevel iteration’, *Numerical Linear Algebra with Applications*, **9**, 599–618, (2002).
- [23] Y. Notay, ‘Convergence analysis of perturbed two-grid and multigrid methods’, *SIAM Journal on Numerical Analysis*, **45**, 1035–1044, (2007).
- [24] A. C. Muresan and Y. Notay, ‘Analysis of aggregation-based multigrid’, Technical Report GANMN 06–06, Université Libre de Bruxelles, Brussels, Belgium, (2006). <http://homepages.ulb.ac.be/~ynotay>.