SIAM J. SCI. COMPUT.
Vol. 34, No. 4, pp. A2288–A2316

AGGREGATION-BASED ALGEBRAIC MULTIGRID FOR CONVECTION-DIFFUSION EQUATIONS*

YVAN NOTAY†

Abstract. We consider the iterative solution of large sparse linear systems arising from the upwind finite difference discretization of convection-diffusion equations. The system matrix is then an M-matrix with nonnegative row sum, and, further, when the convective flow has zero divergence, the column sum is also nonnegative, possibly up to a small correction term. We investigate aggregation-based algebraic multigrid methods for this class of matrices. A theoretical analysis is developed for a simplified two-grid scheme with one damped Jacobi postsmoothing step. An uncommon feature of this analysis is that it applies directly to problems with variable coefficients; e.g., to problems with recirculating convective flow. On the basis of this theory, we develop an approach in which a guarantee is given on the convergence rate thanks to an aggregation algorithm that allows an explicit control of the location of the eigenvalues of the preconditioned matrix. Some issues that remain beyond the analysis are discussed in the light of numerical experiments, and the efficiency of the method is illustrated on a sample of large two- and three-dimensional problems with highly varying convective flow.

Key words. multigrid, convergence analysis, linear systems, aggregation, convection-diffusion

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

DOI. 10.1137/110835347

1. Introduction.

The efficient solution of large sparse $n \times n$ linear systems

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

is critical for many of today's applications in science and engineering. In this work we focus on nonsymmetric problems that arise when the convection-diffusion partial differential equation (PDE)

$$\begin{cases} -\nu\,\Delta u \;+\; \overline{v}\cdot\mathbf{grad}(u) = f & \text{in } \Omega, \\ u = g_0 & \text{on } \Gamma_0, \\ \frac{\partial u}{\partial n} = g_1 & \text{on } \Gamma_1 = \partial\Omega\backslash\Gamma_0 \end{cases} \tag{1.2}$$

is discretized by an upwind finite difference method (e.g., [36, 40]); in this equation, Ω is either a two- or a three-dimensional bounded domain, ν the viscosity, and $\overline{v}$ a convective flow. For reasons that are explained below, we restrict ourselves to the most common situation where $\overline{v}$ has zero divergence. Note that having an efficient solver for this class of applications is also a needed ingredient to several approaches for solving Navier–Stokes problems; see [1, 13] and the references therein.

For linear systems arising from discretized PDEs, efficient iterative solvers are often schemes of the multigrid or the multilevel type [16, 40]. These combine the effects of a *smoother* and of a *coarse grid correction*. The smoother is generally based on a simple iterative scheme such as the Gauss–Seidel method. The coarse grid

*Submitted to the journal's Methods and Algorithms for Scientific Computing section May 25, 2011; accepted for publication (in revised form) May 22, 2012; published electronically August 15, 2012. This work was supported by the Belgian FRNS ("Directeur de recherches").
http://www.siam.org/journals/sisc/34-4/83534.html
†Université Libre de Bruxelles, Service de Métrologie Nucléaire (C.P. 165-84), 50 Av. F.D. Roosevelt, B-1050 Brussels, Belgium (ynotay@ulb.ac.be, homepages.ulb.ac.be/∼ynotay).

correction consists in computing an approximate solution to the residual equation on a coarser grid which has fewer unknowns. The approach is applied recursively until the coarse system is small enough to make negligible the cost of an exact solution.

Now, for $\nu \to 0$, the PDE (1.2) is a singularly perturbed problem. As a general rule, such problems raise difficulties to multigrid methods [16]. Specifically, these methods usually do not possess mesh-independent convergence for the convection-diffusion equation with dominant convection unless special techniques are used to define the smoother and/or the coarse grid correction; see [10, 31, 35, 46] for a sample of results along these lines.

For constant flow problems, for example, using a downstream numbering of the unknowns makes the Gauss–Seidel method increasingly efficient as the viscosity parameter ν decreases, because all entries in the strictly upper triangular part are small (proportional to ν) compared to the diagonal. Hence using this method as a smoother in a multigrid scheme allows one to compensate for the intrinsic difficulties raised by the singular perturbation, and yields a solver that is robust with respect to both ν and the mesh size h; see, e.g., [16, 47]. It is, however, more challenging to solve problems with recirculating flow, for which a downstream numbering may not exist or become difficult to implement. In [13, 40], it is suggested that the same effect can be obtained by performing four (in two dimensions) or six (in three dimensions) line Gauss–Seidel sweeps, corresponding to the four or six possible lexicographic orderings. This makes the overall scheme very costly, whereas its theoretical validation is so far based on Fourier analysis. Hence only equations with constant flow can be analyzed directly: for more general problems, the best that can be done is, following [44], to check that the convergence rate with constant flow is bounded independently of the direction of the flow. This latter approach has also been used to validate some more specific multigrid or multilevel techniques based on Schur complement approximation [27, 34]. These methods do not require such costly smoothers, but have so far been considered for two-dimensional problems only.

The methods mentioned above are all of the *geometric* multigrid or multilevel type; that is, knowledge from the continuous problem and its discretization is needed to define the hierarchy of coarse systems. In contrast with this, *algebraic* multigrid (AMG) methods build this hierarchy exclusively from the system matrix, without additional input. For symmetric problems arising from elliptic PDEs, there are several such approaches that lead to efficient solvers [4, 5, 6, 18, 38, 42]. In particular, contrarily to geometric multigrid methods, they tend to be robust for wide classes of applications without having to adapt the smoother. However, regarding convection-diffusion equations, the only convergence analyses we are aware of are the ones in [45] and [7]. The former is related to classical AMG methods along the lines of [4, 38] and is also restricted to two-dimensional problems with constant flow. Moreover, for recirculating flows, it is again suggested to obtain robustness thanks to four Gauss–Seidel sweeps. The method developed in [7] extends the approaches in [6, 42], i.e., is based on smoothed aggregation AMG. The convergence proof is general, but requires more smoothing steps for highly nonsymmetric problems than would be necessary for symmetric problems. Moreover, despite this analysis, the method is apparently not scalable for recirculating flows.

In this work, we consider aggregation-based multigrid schemes, in which the hierarchy of coarse problems is obtained from a mere aggregation of the unknowns. This approach is sometimes referred to a "plain" or "unsmoothed" aggregation, to distinguish it from "smoothed aggregation AMG" mentioned above. It is somewhat non-

standard because it does not mimic any well-established geometric multigrid method, and, although introduced quite early [3, 8], it has been overlooked for a long time, as it was originally unclear how to achieve a mesh-independent convergence rate; see, e.g., [38, pp. 522–524]. However, a number of recent works show that the approach is both theoretically well founded and practically efficient [14, 19, 23, 24, 25, 28], providing that aggregates are formed in an appropriate way, and that one uses an enhanced multigrid cycle, that is, either the K-cycle [30] or the AMLI-cycle [43], in which the iterative solution of the residual equation at each level is accelerated with either a Krylov subspace method (K-cycle) or a semi-iterative method based on Chebyshev polynomials (AMLI-cycle).

In particular, numerical results in [19, 28] suggest that the approach can be efficient for convection-diffusion problems. However, the theoretical foundation is so far restricted to systems with symmetric positive definite (SPD) coefficient matrix A. Specifically, advanced results are obtained in [25] for the class of nonsingular symmetric M-matrices with nonnegative row sum. It is shown that, for any such matrix, the aggregates can be built in such a way that a meaningful bound on the convergence rate is guaranteed; moreover, this bound is a fixed number, i.e., independent of any problem parameter or peculiarity.

In the present work we mainly extend this latter approach to *nonsymmetric* M-matrices. Using the analysis in [29], we show that the aggregation algorithm at the heart of the approach in [25] can be applied to the symmetric part of the matrix $\frac{1}{2}(A+A^T)$ and yet guarantee some convergence result for the nonsymmetric problem. It is worth mentioning that our results apply as well to problems with variable (e.g., recirculating) flow. Thus in some sense we break the curse according to which the theory for nonsymmetric problems is restricted to the constant coefficient case (e.g., because it is based on the Fourier analysis) and hence unable to directly address the peculiarities associated with recirculating flows.

Nevertheless, whereas in [25] a full multilevel convergence proof is provided, we have to be less ambitious in the nonsymmetric case, and the following restrictions apply: the convergence analysis is for the two-grid method only (with exact solution of the coarse system); it holds for a simplified scheme with only one damped Jacobi smoothing step (whereas Gauss–Seidel smoothing works better in practice); it is based on eigenvalues, and hence reflects the asymptotic convergence properties, but does not take into account "nonnormality" effects that can deeply affect the actual convergence; see [15, 20] and [39, Chapters 25 and 26]. However, these issues are discussed through numerical experiments.

Regarding assumptions, our analysis requires that the system matrix A is a nonsingular M-matrix with row and column sum both nonnegative. Recall that an M-matrix has nonpositive offdiagonal entries, and a matrix C with nonpositive offdiagonal entries is an M-matrix if and only if $C\mathbf{x}$ is nonnegative for at least one positive vector $\mathbf{x}$ [2]; in particular (consider $\mathbf{x} = \mathbf{1}$, i.e., the vector with all components equal to 1), a matrix with nonpositive offdiagonal entries that has nonnegative row sum is an M-matrix. Then it is clear (see below for details) that the matrix A arising from the upwind finite difference discretization of (1.2) is an M-matrix with nonnegative row sum. However, one may wonder about the column sum. In the next section, we obtain an expression for the column sum at a given gridpoint which shows that it is equal to the divergence of $(-\overline{v})$ at that point, plus some $\mathcal{O}(h^2)$ or $\mathcal{O}(h^3)$ correction term. This motivates the restriction of the present study to problems with divergence-free convective flow. Note that, because of the correction term, the column sum turns

out to be negative at part of the gridpoints in some of the problems we have tested, without having a significant influence on the results.

The remainder of this paper is organized as follows. In section 2, we briefly review the matrix properties associated with the upwind finite difference discretization of (1.2). Our convergence analysis in presented in section 3, and the associated aggregation algorithm is given in section 4. Numerical results are reported in section 5 and conclusions are drawn in section 6.

2. Discretization of convection-diffusion equations. Here we want to assess the matrix properties associated with the upwind finite difference discretization of (1.2) on a regular grid with uniform mesh size h in all directions. For the sake of simplicity, we consider two-dimensional problems only. The analysis of three-dimensional problems leads to the same conclusions.

We first write the stencil of A at some typical gridpoint P, with neighbors N, E, S, W as depicted on the figure below. We assume that neither P nor its neighbors belong to the boundary.

$$\begin{array}{ccccc} \times & \times & \overset{N}{\times} & \times & \times \\ \times & \overset{W}{\times} & \overset{P}{\times} & \overset{E}{\times} & \times \\ \times & \times & \overset{S}{\times} & \times & \times \end{array}$$

Beside the stencil of A, we give the stencil of its transpose, which is obtained by considering the stencil of A (hence entries in its rows) at neighboring gridpoints. Since the discretization scheme depends on the sign of the components of the convective flow $\overline{v} = (v_x, v_y)$, we need to consider different cases. We give the result for $v_y(P) > 0$ and both $v_x(P) > 0$ and $v_x(P) < 0$; the case $v_y(P) < 0$ can be treated in a similar fashion. If $v_x(P)$ is positive (negative), we assume that it is accordingly nonnegative (nonpositive) at neighboring gridpoints. This always holds if the lines where $v_x(P)$ is zero coincide with grid lines.

If $v_x(P) > 0$:

$$\text{Stencil of } A: \quad \begin{bmatrix} & -\frac{\nu}{h^2} & \\ -\frac{\nu}{h^2} - \frac{v_x(P)}{h} & \frac{4\,\nu}{h^2} + \frac{v_x(P)+v_y(P)}{h} & -\frac{\nu}{h^2} \\ & -\frac{\nu}{h^2} - \frac{v_y(P)}{h} & \end{bmatrix}.$$

$$\text{Stencil of } A^T: \quad \begin{bmatrix} & -\frac{\nu}{h^2} - \frac{v_y(N)}{h} & \\ -\frac{\nu}{h^2} & \frac{4\,\nu}{h^2} + \frac{v_x(P)+v_y(P)}{h} & -\frac{\nu}{h^2} - \frac{v_x(E)}{h} \\ & -\frac{\nu}{h^2} & \end{bmatrix}.$$

If $v_x(P) < 0$:

$$\text{Stencil of } A: \quad \begin{bmatrix} & -\frac{\nu}{h^2} & \\ -\frac{\nu}{h^2} & \frac{4\,\nu}{h^2} + \frac{-v_x(P)+v_y(P)}{h} & -\frac{\nu}{h^2} - \frac{-v_x(P)}{h} \\ & -\frac{\nu}{h^2} - \frac{v_y(P)}{h} & \end{bmatrix}.$$

$$\text{Stencil of } A^T: \quad \begin{bmatrix} & -\frac{\nu}{h^2} - \frac{v_y(N)}{h} & \\ -\frac{\nu}{h^2} - \frac{-v_x(W)}{h} & \frac{4\,\nu}{h^2} + \frac{-v_x(P)+v_y(P)}{h} & -\frac{\nu}{h^2} \\ & -\frac{\nu}{h^2} & \end{bmatrix}.$$

Hence one sees that A is indeed a matrix with nonpositive offdiagonal entries and nonnegative row sum, and therefore an M-matrix.

The column sum at P is trivially obtained by summing the entries in the stencil of A^T. To obtain from there more insightful expressions, we further assume that v_x and v_y are twice differentiable functions of x and y, respectively, and that the second derivative is continuous. Then

$$v_y(N) = v_y(P) + h\frac{\partial v_y}{\partial y}\,|_P + h^2\frac{\partial^2 v_y}{\partial y^2}\,|_n\,,$$

where n is some point situated between P and N. Similarly,

$$\begin{aligned} v_x(E) &= v_x(P) + h\frac{\partial v_x}{\partial x}\,|_P + h^2\frac{\partial^2 v_x}{\partial x^2}\,|_e \quad , \\ v_x(W) &= v_x(P) - h\frac{\partial v_x}{\partial x}\,|_P + h^2\frac{\partial^2 v_x}{\partial x^2}\,|_w \quad , \end{aligned}$$

where e is some point situated between P and E, whereas w is some point situated between W and P. Using these relations one finds, for the case $v_x(P) > 0$,

$$\begin{aligned} (A^T\mathbf{1})_P &= h^{-1}\big(v_x(P) - v_x(E) + v_y(P) - v_y(N)\big) \\ &= -\mathbf{div}\,(\overline{v})\,|_P - h\left(\frac{\partial^2 v_x}{\partial x^2}\,|_e + \frac{\partial^2 v_y}{\partial y^2}\,|_n\right)\,, \end{aligned}$$

whereas, for the case $v_x(P) < 0$,

$$\begin{aligned} (A^T\mathbf{1})_P &= h^{-1}\big(-v_x(P) + v_x(W) + v_y(P) - v_y(N)\big) \\ &= -\mathbf{div}\,(\overline{v})\,|_P + h\left(\frac{\partial^2 v_x}{\partial x^2}\,|_e - \frac{\partial^2 v_y}{\partial y^2}\,|_n\right)\,. \end{aligned}$$

When $\overline{v}$ has zero divergence, the column sum at interior gridpoints is therefore zero up to a correction term which, relative to the diagonal element, is of order $\mathcal{O}(h^2)$ or $\mathcal{O}(h^3)$.

3. Two-grid analysis.

3.1. General setting.

As already stated, two-grid methods combine the effects of a smoother and of a coarse grid correction. If a stationary iterative method is used, it means that at each step the error vector (i.e., the difference between the exact solution and the current approximation) is multiplied by the *iteration matrix*

$$T = (I - M_2^{-1}A)^{\nu_2}(I - P\,A_c^{-1}R\,A)(I - M_1^{-1}A)^{\nu_1}\,. \tag{3.1}$$

In this expression, M_1, M_2 are, respectively, the pre- and the postsmoother, and ν_1, ν_2 are the number of pre- and postsmoothing steps; letting n_c be the number of coarse unknowns, P is the prolongation, an $n \times n_c$ matrix that allows to prolongate on the fine grid a vector defined on the coarse grid; R is the restriction, an $n_c \times n$ matrix that does the converse operation; i.e., it allows one to map on the coarse variables a given fine grid vector; A_c is the coarse grid matrix, an $n_c \times n_c$ matrix that in some sense represents the fine grid problem on the coarse variables.

Note that in practice one rarely stays with a two-grid scheme. Indeed, it remains costly to solve exactly a system with the coarse grid matrix A_c as required by the term A_c^{-1} in (3.1). Therefore, an approximate solution is used instead, which is itself based on the same two-grid scheme applied now to this coarse system, and hence using a further coarser grid with even fewer unknowns. The approach so defined is then followed recursively until the coarse grid matrix is small enough to be factorized at negligible cost. As stated in the introduction, our theoretical analysis is, however, restricted to the model two-grid method defined by (3.1).

Clearly, the efficiency of the approach critically depends on the good interplay between the smoother and the coarse grid correction. That is, modes that are not significantly damped by the smoothing iterations (i.e., by the factors $(I-M_2^{-1}A)^{\nu_2}$ and $(I-M_1^{-1}A)^{\nu_1}$) should be efficiently damped by the coarse grid correction. In AMG methods, the components P, R, A_c defining this latter are not given but set up by appropriate algorithms which use only information from the system matrix A. Most often—and this work will make no exception—one sets $A_c = R\,A\,P$, which ensures that the coarse grid correction term $I - P\,A_c^{-1}R\,A$ is a projector which perfectly filters out modes in the range of P. The setup amounts then to finding appropriate P and R, in a way that favors the good interplay mentioned above, most often fixing the smoother to some simple method such as the Gauss–Seidel or damped Jacobi methods.

According to the remark above about the recursive use of a hierarchy of two-grid schemes, this *setup phase* is not only applied at the top (fine grid) level, but actually recursively at the successively defined coarse levels. If the two-grid analysis is based on some matrix properties, it is then important that the coarse grid matrices generated by the process inherit these properties from the fine grid system matrix.

In this work, we focus on very simple (piecewise constant) prolongations associated with aggregation. The main setup task consists then in the agglomeration of the unknowns into n_c nonempty disjoint sets $G_k\,, k = 1, \ldots, n_c\,$, called *aggregates*. To each aggregate G_k is associated one unknown at the next coarse level in the hierarchy. In addition, some unknowns can be also kept outside the coarsening process, and the corresponding (possibly empty) set is denoted G_0; that is, G_0 gathers the unknowns that are not associated to any coarse unknown. Thus $G_k\,, k = 0, 1, \ldots, n_c$, is a partitioning of $[1, n]$. This partitioning uniquely determines the prolongation P: for $i = 1, \ldots, n$ and $j = 1, \ldots, n_c\,$, we set

$$(P)_{ij} = \begin{cases} 1 & \text{if } i \in G_j\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{3.2}$$

Hence a row of P is zero if and only if the corresponding unknown is in $G_0\,$, whereas the other rows have exactly one nonzero entry. Note also that P has full rank by construction.

When G_0 is empty, a constant vector on the coarse grid is then prolongated by a constant vector on the fine grid. This is a sensible choice for M-matrices with nonnegative row-sum, and, more generally, for matrices from the discretization of scalar PDEs like (1.2). As made clearer below, the role of G_0 is to gather nodes that need not to be represented on the coarse grid because the corresponding error components are sufficiently damped thanks to the sole action of the smoother.

Regarding the restriction, $R = P^T$ is the standard choice in the symmetric case, but it is still an open question how to, in general, best choose R for nonsymmetric problems; see [7, 11, 21, 29, 37, 44] for related discussions. However, given the simple form of P and the fact that the column sum is also nonnegative, we stay with $R = P^T$, which seems the most sensible option in this context; the results obtained below may also be seen as an a posteriori justification of this choice. Note that if the row and/or the column sum would be not nonnegative, it might be better to use an adaptive scheme in the spirit of the method in [7]. The discussion of such approaches lies, however, outside the scope of the present paper.

As mentioned above, we use $A_c = R\,A\,P$, or, since $R = P^T$, $A_c = P^T\,A\,P$. Hence all components P, R, A_c are fixed from (3.2) once the partitioning $G_k\,, k =$

$0, 1, \ldots, n_c$, has been defined. In particular, it is worth noting that the entries in $A_c = P^T A P$ can be obtained from a simple summation process:

$$(A_c)_{kl} = \sum_{i \in G_k} \sum_{j \in G_l} a_{ij} \qquad k, l = 1, \ldots, n_c \ . \tag{3.3}$$

The following lemma proves some properties of the matrices A and A_c, showing in particular that A_c defined in this way is nonsingular and, more generally, inherits useful algebraic properties from A. Statement (2) is essentially retrieved from Theorem 3.6 in [19].

LEMMA 3.1. *Let A be a nonsingular $n \times n$ M-matrix with row and column sum both nonnegative. Let G_k, $k = 0, \ldots, n_c$, be some partitioning of $[1, n]$, define P by (3.2), and set $A_c = P^T A P$.*

(1) *If A is irreducible, then $A_S = \frac{1}{2}(A + A^T)$ is positive definite.*

(2) *A_c is an M-matrix with row and column sum both nonnegative.*

(3) *If $A_S = \frac{1}{2}(A + A^T)$ is positive definite, then $\frac{1}{2}(A_c + A_c^T)$ is positive definite and A_c is nonsingular.*

(4) *Letting $D = \mathrm{diag}(A)$, there holds*

$$\|I - D^{-1/2} A D^{-1/2}\|_2 \leq 1 \ .$$

Proof. (1) A_S has nonpositive offdiagonal entries and nonnegative row sum, and hence it is a symmetric M-matrix and therefore nonnegative definite [2]. Further, the irreducibility of A entails that of A_S, and it is known that for an irreducible symmetric M-matrix A_S, when $A_S \mathbf{1} \geq \mathbf{0}$ (where $\mathbf{0}$ is the zero vector),[1] then either A_S is positive definite or $A_S \mathbf{1} = \mathbf{0}$ [2]. But, with $A\mathbf{1} \geq \mathbf{0}$ and $A^T \mathbf{1} \geq \mathbf{0}$, $A_S \mathbf{1} = \mathbf{0}$ is possible only if $A\mathbf{1} = A^T \mathbf{1} = \mathbf{0}$, i.e., only if A is singular, which contradicts the assumptions.

(2) One has, for all $1 \leq k \leq n_c$,

$$\sum_{l=1}^{n_c} (A_c)_{kl} = \sum_{l=1}^{n_c} \left(\sum_{i \in G_k} \sum_{j \in G_l} a_{ij} \right) = \sum_{i \in G_k} \left(\sum_{j=1}^{n} a_{ij} - \sum_{j \in G_0} a_{ij} \right) \geq \sum_{i \in G_k} (A\,\mathbf{1})_i \ .$$

Hence the row sum of A_c is nonnegative. The proof of the nonnegativity of the column sum is similar. On the other hand, one sees from (3.3) that A_c has nonpositive offdiagonal entries. Then, having nonnegative row sum, it is necessarily a (possibly singular) M-matrix.

(3) Since $\frac{1}{2}(A_c + A_c^T) = P^T A_S P$ and P has full rank, the positive definiteness of A_S entails that of $\frac{1}{2}(A_c + A_c^T)$. This further ensures the nonsingularity of A_c since $A_c \mathbf{x} = \mathbf{0}$ for any $\mathbf{x}$ would entail $\mathbf{x}^T (A_c + A_c^T) \mathbf{x} = \mathbf{0}$, which is possible only for $\mathbf{x} = \mathbf{0}$.

(4) Let $C = I - D^{-1/2} A D^{-1/2}$, and observe that

$$\|I - D^{-1/2} A D^{-1/2}\|_2^2 = \lambda_{\max}(C^T C) = \lambda_{\max}(D^{-1/2} C^T C D^{1/2}) \leq \|D^{-1/2} C^T C D^{1/2}\|_\infty \ .$$

Further, since C and therefore $D^{-1/2} C^T C D^{1/2}$ has nonnegative entries (the diagonal of C is zero),

$$\|D^{-1/2} C^T C D^{1/2}\|_\infty = \max_i \left(D^{-1/2} C^T C D^{1/2} \mathbf{1} \right)_i .$$

[1] Inequalities between vectors are to be understood componentwise.

The required result then follows because A has nonnegative row and column sum:

$$D^{-1/2}C^TCD^{1/2}\mathbf{1} = D^{-1/2}C^T\big(D^{1/2}\mathbf{1} - D^{-1/2}A\mathbf{1}\big) \le D^{-1/2}C^TD^{1/2}\mathbf{1} \\ = D^{-1/2}\big(D^{1/2}\mathbf{1} - D^{-1/2}A^T\mathbf{1}\big) \le \mathbf{1}\ . \qquad \square$$

The main assumptions of our analysis below are that A is a nonsingular M-matrix with row and column sum both nonnegative and such that its symmetric part $A_S = \frac{1}{2}(A + A^T)$ is positive definite. By (1) of the above lemma, this latter property follows from the irreducibility of A; (2) and (3) then show that A_c inherits these properties. The usefulness of statement (4) will appear later.

Now, as stated in the introduction, our theoretical analysis is restricted to the case of just one damped Jacobi postsmoothing step; that is, $\nu_1 = 0$, $\nu_2 = 1$, and $M_2 = \omega^{-1}D$, where $D = \mathrm{diag}(A)$. With $R = P^T$, the iteration matrix (3.1) is then simplified to

$$T = (I - \omega\, D^{-1}A)(I - P\, A_c^{-1}P^TA)\ . \tag{3.4}$$

On the other hand, here we mainly consider the two-grid scheme as a preconditioner for a Krylov subspace method. This is true at the fine grid level, but also at coarser levels because we intend to use the K-cycle. It means that we have to implement the action of the B_{TG}^{-1} on a vector, where B_{TG}^{-1} is defined from the iteration matrix via

$$T = I - B_{\mathrm{TG}}^{-1}A\ .$$

With (3.4) this amounts to

$$B_{\mathrm{TG}}^{-1} = \omega\, D^{-1} + (I - \omega\, D^{-1}A)P\, A_c^{-1}P^T\ . \tag{3.5}$$

In the SPD case, the eigenvalues of $B_{\mathrm{TG}}^{-1}A$ are real positive, and the convergence properties are in general well reflected by the *condition number*, which is equal to the ratio of the extremal eigenvalues. There is no such "magic" number in the nonsymmetric case, but

$$\rho_{\alpha_{\mathrm{opt}}} = \min_{\alpha} \max_{\lambda\in\sigma(B_{\mathrm{TG}}^{-1}A)} |1 - \alpha\,\lambda| \tag{3.6}$$

is often meaningful. It represents the asymptotic convergence factor of a fixed point iterative method with optimal scaling of the preconditioner. When this number is reasonably small (away from 1), then the eigenvalues of $B_{\mathrm{TG}}^{-1}A$ are necessarily both bounded and away from the origin, which are the two main requirements for efficient preconditioning. The convergence also depends on phenomena that cannot be captured by the analysis of the sole eigenvalues, but we can only discuss the latter in section 5 via some numerical experiments.

3.2. Summary of previous results. Before developing our analysis, for the sake of readability, we gather needed results from [29] and [24] in, respectively, Lemma 3.2 and Lemma 3.3 below. We give a short proof because the statements in these lemma are particularized to the context defined in the previous subsection, which requires some additional steps.

LEMMA 3.2. *Let A be a nonsingular $n \times n$ matrix with positive diagonal entries. Let P be a $n \times n_c$ matrix of rank n_c such that $A_c = P^T\,A\,P$ is nonsingular. Let B_{TG} be defined by* (3.5)*, where $D = \mathrm{diag}(A)$.*

(1) $P^T D\, P$ *is nonsingular.*

(2) *The eigenvalues of* $B_{TG}^{-1}A$ *that are not equal to one are the inverse of the nonzero eigenvalues of*

$$\omega^{-1}A^{-1}D\big(I - P(P^TDP)^{-1}P^TD\big).$$

(3) *If* $\|I - D^{-1/2}A\,D^{-1/2}\|_2 \leq 1$, *then any eigenvalue* λ *of* $B_{TG}^{-1}A$ *is either equal to one or satisfies*

$$|\lambda - \omega| \leq \omega\ . \tag{3.7}$$

(4) *If* $A_S = \frac{1}{2}(A + A^T)$ *is positive definite, then any eigenvalue* λ *of* $B_{TG}^{-1}A$ *satisfies*

$$\Re e(\lambda) \geq \min\left(1\,,\,\frac{1}{\kappa_S}\right)\ , \tag{3.8}$$

where

$$\kappa_S\ = \omega^{-1}\ \sup_{\mathbf{v}\neq 0}\frac{\mathbf{v}^T D\left(I - P(P^T D\, P)^{-1}P^T D\right)\mathbf{v}}{\mathbf{v}^T A_S \mathbf{v}}\ . \tag{3.9}$$

Moreover, κ_S *in this relation is nothing but the inverse of the smallest eigenvalue of* $\big(B_{TG}^{(S)}\big)^{-1}A_S$, *where*

$$\big(B_{TG}^{(S)}\big)^{-1} = \omega\, D^{-1} + (I - \omega\, D^{-1}A_S)P\,(P^T A_S P)^{-1}P^T\ .$$

Proof. (1) Since the diagonal entries of A are assumed positive, D is SPD, and therefore $P^T D\, P$ is also SPD because P has full rank.

(2) Observing that the eigenvalues of $B_{\rm TG}^{-1}A$ are those of $I - T$, this statement is essentially statement 1 of Theorem 2.1 in [29], applied to the case $\nu_1 = 0$, $\nu_2 = 1$, $M_2 = \omega^{-1}D$, and therefore $X = \omega^{-1}D$ (with thus $X_c = P^T D\, P$ nonsingular, as required by this theorem).

(3) Consider again $\nu_1 = 0$, $\nu_2 = 1$, $M_2 = \omega^{-1}D$, and $X = \omega^{-1}D$; X being SPD, observe that

$$\|\omega I - X^{-1}A\|_X = \omega\|I - D^{-1}A\|_D = \omega\|I - D^{-1/2}A\,D^{-1/2}\|_2\ ;$$

that is, $\|I - D^{-1/2}A\,D^{-1/2}\|_2 \leq 1$ if and only if $\|\omega I - X^{-1}A\|_X \leq \omega$. Then the required result straightforwardly follows from relation (34) in Corollary 2.1 of [29].

(4) Considering again $\nu_1 = 0$, $\nu_2 = 1$, $M_2 = \omega^{-1}D$, and $X = \omega^{-1}D$, this statement is inequality (44) in Corollary 2.2 of [29], in which one sets $Q = P(P^T D\, P)^{-1}P^T D$, and therefore $(I - Q^T)X(I - Q) = \omega^{-1}D\,\big(I - P(P^T D\, P)^{-1}P^T D\big)$. ∎

LEMMA 3.3. *Let* A_S *be an* $n \times n$ *SPD matrix, and set* $D = \mathrm{diag}(A_S)$. *Let* G_k, $k = 0, \ldots, n_c$, *be some partitioning of* $[1, n]$, *and define* P *by* (3.2). *Let* A_b, A_r *be nonnegative definite symmetric matrices such that* $A_S = A_b + A_r$ *and* A_b *is block diagonal with respect to the partitioning* G_k *(i.e.,* $(A_b)_{ij} = 0$ *if* i, j *do not belong to the same subset* G_k*). For* $k = 0, \ldots, n_c$, *let* $A_b|_{G_k}$ *be the submatrix of* A_b *corresponding to indices in* G_k. *There holds*

$$\sup_{\mathbf{v}\neq 0}\frac{\mathbf{v}^T D\left(I - P(P^T D\, P)^{-1}P^T D\right)\mathbf{v}}{\mathbf{v}^T A_S \mathbf{v}}\ \leq \max_{k=0,\ldots,n_c}\mu^{(k)}\,, \tag{3.10}$$

where

$$\mu^{(0)} = \begin{cases} 0 & \text{if } G_0 \text{ is empty}, \\ \max_{\mathbf{v}} \frac{\mathbf{v}^T D_{G_0} \mathbf{v}}{\mathbf{v}^T A_b|_{G_0} \mathbf{v}} & \text{otherwise} \end{cases}$$

and where, for $k = 1, \ldots, n_c$,

$$\mu^{(k)} = \begin{cases} 0 & \text{if } |G_k| = 1\,, \\ \sup_{\mathbf{v} \notin \mathcal{N}(A_b|_{G_k})} \frac{\mathbf{v}^T D|_{G_k} (I - \mathbf{1}_{G_k} (\mathbf{1}_{G_k}^T D|_{G_k} \mathbf{1}_{G_k})^{-1} \mathbf{1}_{G_k}^T D|_{G_k}) \mathbf{v}}{\mathbf{v}^T A_b|_{G_k} \mathbf{v}} & \text{otherwise}\,, \end{cases}$$

with $D|_{G_k}$ *being the submatrix of* D *corresponding to indices in* G_k *and* $\mathbf{1}_{G_k} = (1, 1, \cdots, 1)^T$ *(of size* $|G_k|$*).*

Proof. This lemma is Theorem 3.2 in [24], applied to A_S and particularized to the case $D = \mathrm{diag}(A)$ with $\mathbf{p}_k = \mathbf{1}_{G_k}$ for all $k > 0$. ∎

3.3. Analysis. First of all, with (2) of Lemma 3.2, one sees that changing ω only rescales all the eigenvalues of $B_{\mathrm{TG}}^{-1} A$ except the eigenvalue 1, which, nevertheless, remains inside the spectrum if ω is chosen in a sensible way. Then this rescaling has nearly no influence on the convergence when using Krylov subspace acceleration. We therefore do not need to discuss how to select ω ; we leave it as unspecified parameter in this section, whereas the numerical illustration in section 5 is given for $\omega = 0.5$.

Now we come back to the two main requirements for efficient preconditioning discussed at the end of section 3.1, that is, to have the eigenvalues of $B_{\mathrm{TG}}^{-1} A$ both bounded and away from the origin.

The first objective of having bounded eigenvalues is the easiest to check. In the SPD case, it is a quite general result that the eigenvalues of $B_{\mathrm{TG}}^{-1} A$ do not exceed 1, under mild assumptions on the smoother and whatever the prolongation P ; see, e.g., [29, 43]. There is no equivalent for nonsymmetric matrices, except perhaps the result in Corollary 2.1 of [29] that is restated in (3) of Lemma 3.2 above. The assumption $\|I - D^{-1/2} A D^{-1/2}\|_2 \leq 1$ is indeed also independent of P , but, however, in general far from trivial to check. Fortunately, we have already shown in (4) of Lemma 3.1 that it is always satisfied in the context of this paper.

On the other hand, to show that the eigenvalues are away from the origin, we can use (4) of Lemma 3.2. It proves that the smallest eigenvalue for the two-grid scheme applied to the symmetric part of the matrix is also a lower bound for the real part in the nonsymmetric case. Moreover, this result holds for any P . From there, analyzing the nonsymmetric case looks easy. This is probably true for convection-diffusion problems with constant flow. However, consider the stencils given in the previous section, and average those of A and A^T to get that of the symmetric part. It is close to that associated with a diffusion problem

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

with coefficients

$$a_x = \nu + h\,|v_x| \qquad a_y = \nu + h\,|v_y|\ .$$

For variable flows and dominating convection ($\nu \ll h|\overline{v}|$), this diffusion tensor varies continuously on the grid, and, according to the region of the domain, the anisotropy ratio a_x/a_y may be very large, very small, or moderate. Hence, even reduced to the

symmetric part, the problem presents some unusual features that may prevent fast convergence of multigrid methods.

However, the results in [24, 25] suggest that, with aggregation-based multigrid schemes, it is possible to efficiently bound below the eigenvalues for such symmetric problems with varying coefficients. The general idea is as follows. Theorem 3.2 of [24] has been restated in Lemma 3.3 above. To apply it, one first needs a proper splitting of the symmetric part A_S into two nonnegative definite matrix. When A_S is an M-matrix with nonnegative row sum, this splitting is easy to obtain: regarding the offdiagonal part, A_b gathers the connections that are internal to the aggregates (i.e., between two nodes of a same aggregate), and A_r the other ones; on the other hand the diagonals are such that A_r has zero row sum and therefore A_b has the same row sum as A_S. In other words, A_b is obtained from A_S by discarding and lumping to the diagonal the entries that are in the block offdiagonal part with respect to the partitioning in aggregates. This lumping ensures that both A_b and A_r are weakly diagonally dominant and hence nonnegative definite.

Then (see (3.10)) the main quantity involved in the lowest eigenvalue analysis can be controlled if the parameter $\mu^{(k)}$ associated with each aggregate is efficiently bounded. And this parameter depends only on the corresponding diagonal block in A_b, i.e., on quantities "local" to the aggregate: the "local" matrix entries and the row sum at "local" nodes. Observe that the lumping process mentioned above tends to make these diagonal blocks $A_b|_{G_k}$ ill-conditioned, or even singular with zero row sum when the row sum of A_S is zero at every node of the aggregate. But, for regular aggregates ($k > 0$), it does not mean that $\mu^{(k)}$ is unbounded since the matrix $D|_{G_k}(I - \mathbf{1}_{G_k}(\mathbf{1}_{G_k}^T D|_{G_k}\mathbf{1}_{G_k})^{-1}\mathbf{1}_{G_k}^T D|_{G_k})$ is also, by construction, singular with the constant vector in its kernel. This is not true for $\mu^{(0)}$, in fact, because nodes in G_0 are kept outside the aggregation and actually not represented anymore on the coarse grid. Therefore, smoothing iterations alone should be sufficient to efficiently damp the error at these nodes, which, in this analysis, is reflected by the requirement that the corresponding block $A_b|_{G_0}$ is well-conditioned. In practice, as will be seen below, this requirement can be checked via a strong diagonal dominance criterion.

Theorem 3.4 below formally states the result that can be obtained from these considerations together with the above lemma. It extends to the nonsymmetric case the approach followed in Theorem 2.1 of [25] in the symmetric case. This is, however, not a mere extension because here we restrict ourselves to damped Jacobi smoothing, whereas in [25] the focus is on a class of more sophisticated SPD smoothers.

THEOREM 3.4. *Let $A = (a_{ij})$ be an $n \times n$ nonsingular M-matrix with row and column sums both nonnegative and such that $A_S = \frac{1}{2}(A + A^T)$ is positive definite. Let G_k, $k = 0, \ldots, n_c$, be some partitioning of $[1, n]$, and define P and B_{TG} by (3.2) and (3.5), respectively, where $D = \mathrm{diag}(A)$, $A_c = P^T A P$, and where ω is some positive number. Let λ_i be any eigenvalue of $B_{TG}^{-1}A$.*

(1) *There holds $\lambda_i = 1$ or*

$$|\lambda_i - \omega| \leq \omega \ . \tag{3.11}$$

(2) *For any subset G of $[1, n]$, let $A_S|_G$ be the submatrix of A_S corresponding to indices in G, and let Σ_G be the $|G| \times |G|$ diagonal matrix with $(\Sigma_G)_{ii} = \sum_{j \notin G} |\frac{a_{G(i)j} + a_{jG(i)}}{2}|$, where $G(i)$ is the ith element in G. Set $A_G^{(S)} = A_S|_G - \Sigma_G$, $D_G = \mathrm{diag}(A_S|_G)$, and define, if $|G| > 1$,*

$$\mu(G) = \omega^{-1} \sup_{\mathbf{v} \notin \mathcal{N}(A_G^{(S)})} \frac{\mathbf{v}^T D_G (I - \mathbf{1}_G(\mathbf{1}_G^T D_G \mathbf{1}_G)^{-1}\mathbf{1}_G^T D_G)\mathbf{v}}{\mathbf{v}^T A_G^{(S)} \mathbf{v}} \ , \tag{3.12}$$

where $\mathbf{1}_G = (1, 1, \ldots, 1)^T$ *is a vector of size* $|G|$.

If, for a given $\bar{\kappa}_{TG}$, *there holds*

$$a_{ii} \geq \frac{\bar{\kappa}_{TG}}{\bar{\kappa}_{TG} - \omega^{-1}} \left(\sum_{j=1, j\neq i}^{n} \frac{|a_{ij} + a_{ji}|}{2} \right) \quad \forall i \in G_0 \tag{3.13}$$

and, for $k = 1, \ldots, n_c$, *either* $|G_k| = 1$ *or*

$$\mu(G_k) \leq \bar{\kappa}_{TG} \ , \tag{3.14}$$

then

$$\Re e(\lambda_i) \geq \min \left(1 \, , \, \frac{1}{\bar{\kappa}_{TG}} \right) . \tag{3.15}$$

Eventually, if, in addition, $\omega \geq 0.5$ *and* $\bar{\kappa}_{TG} \geq 1$, *then, defining* $\rho_{\alpha_{opt}}$ *by* (3.6), *there holds*

$$\rho_{\alpha_{opt}} \leq \sqrt{1 - \frac{1}{2\,\omega\,\bar{\kappa}_{TG}}} \ . \tag{3.16}$$

Proof. Statements Lemma 3.1(4) and Lemma 3.2(3) prove (3.11). On the other hand, statement Lemma 3.2(4) proves (3.15) if κ_S as in (3.9) satisfies $\kappa_S \leq \bar{\kappa}_{\rm TG}$. To show this, we apply Lemma 3.3 using the splitting $A_S = A_b + A_r$, where, for $k = 0, \ldots, n_c$, the diagonal block of A_b corresponding to the subset G_k is $A^{(S)}_{G_k}$. Since A_S has nonnegative row sum and nonpositive offdiagonal entries, the definition of $A^{(S)}_{G_k}$ indeed entails that both A_b and A_r are then weakly diagonally dominant; hence they are symmetric nonnegative definite and the splitting satisfies the assumptions of Lemma 3.3. This latter, together with (3.14), then yields the required result, providing that

$$\omega^{-1} \max_{\mathbf{v} \in \Re^{|G_0|}} \frac{\mathbf{v}^T D_{G_0} \mathbf{v}}{\mathbf{v}^T A^{(S)}_{G_0} \mathbf{v}} \leq \bar{\kappa}_{\rm TG} \ ,$$

and this latter inequality follows from the fact that $\omega\,\bar{\kappa}_{\rm TG}\, A^{(S)}_{G_0} - D_{G_0}$ is diagonally dominant and hence nonnegative definite: using (3.13), one obtains

$$\begin{aligned}
(\omega\,\bar{\kappa}_{\rm TG}\, A^{(S)}_{G_0} - D_{G_0})_{ii} &= (\omega\,\bar{\kappa}_{\rm TG} - 1)\, a_{G_0(i)G_0(i)} - \omega\,\bar{\kappa}_{\rm TG} \left(\sum_{j \notin G_0} |(A_S)_{G_0(i)j}| \right) \\
&\geq \omega\,\bar{\kappa}_{\rm TG} \left(\sum_{j \neq G_0(i)} |(A_S)_{G_0(i)j}| - \sum_{j \notin G_0} |(A_S)_{G_0(i)j}| \right) \\
&= \omega\,\bar{\kappa}_{\rm TG} \left(\sum_{j \in G_0\,,\, j \neq G_0(i)} |(A_S)_{G_0(i)j}| \right) \\
&= \sum_{j \neq i} |(\omega\,\bar{\kappa}_{\rm TG} A^{(S)}_{G_0} - D_{G_0})_{ij}| \ .
\end{aligned}$$

Regarding (3.16), we assess the spectral radius for the specific scaling factor $\alpha = (2\omega)^{-1}$; that is, we use

$$\rho_{\alpha_{\rm opt}} \leq \max_{\lambda \in \sigma(B^{-1}_{\rm TG} A)} |1 - (2\omega)^{-1}\,\lambda|$$

and show that the right-hand side is itself bounded by $(1 - \frac{1}{2\,\omega\,\bar{\kappa}_{\rm TG}})^{1/2}$. From (3.11), it follows that, for any eigenvalue λ_i different from 1, $|(2\omega)^{-1}\,\lambda_i - \frac{1}{2}| \leq \frac{1}{2}$, and this is also true for the eigenvalue $\lambda_i = 1$ by virtue of the condition $\omega \geq 0.5$. On the other hand, with $\bar{\kappa}_{\rm TG} \geq 1$, (3.15) entails $\Re{\rm e}({}^{-1}\lambda_i) \geq \bar{\kappa}_{\rm TG}^{-1}$. Hence,

$$|1 - (2\omega)^{-1}\,\lambda_i|^2 = |\tfrac{1}{2} - (2\omega)^{-1}\,\lambda_i|^2 + \tfrac{3}{4} - (2\omega)^{-1}\Re{\rm e}(\lambda_i) \leq \tfrac{1}{4} - (2\omega\bar{\kappa}_{\rm TG})^{-1} + \tfrac{3}{4}\ ,$$

yielding the required result. ∎

This theorem can be used in two ways. First, one may develop an a posteriori analysis of any given aggregation scheme by assessing or computing the smallest $\bar{\kappa}_{\rm TG}$ meeting assumptions (3.13) and (3.14). In particular, for constant coefficient problems and a regular ("geometric") aggregation pattern, it should not be too difficult to derive analytical bounds along the lines of the results in [24]. Such an a posteriori analysis may also, to some extent, explain the good results obtained in [28] with a heuristic aggregation method. Indeed, with the latter methods the aggregation is driven by the strong couplings, and the examples analyzed in [24] suggest that this is indeed a sensible way to obtain meaningful bounds from the assessment of the $\mu(G_k)$.

In this work we focus, however, on the second approach, where the theorem is used hand in hand with a dedicated algorithm which builds the aggregates G_k in such a way that $\mu(G_k)$ is guaranteed below a given threshold. This approach has already been developed in [25] for the symmetric case, and since $\mu(G_k)$ is based on the symmetric part anyway, we can essentially reuse these results; that is, reuse the aggregation algorithm developed there. However, a few adaptations are needed because the damped Jacobi smoother considered here is different from the smoother in [25]. These adaptations are given in the next section together with a general reminder of the approach.

4. Aggregation algorithm. Here we give the aggregation procedure with only a few comments, referring the reader to [25] for more details and motivations. Note that, regarding conditions (3.13), (3.14) that are to be met to allow the application of Theorem 3.4, only the product $\omega\,\bar{\kappa}_{\rm TG}$ matters. Hence, without loss of generality, we fix ω to 0.5 and keep only $\bar{\kappa}_{\rm TG}$ as an adjustable parameter.

In fact, the procedure is a compound of three algorithms. Inspired from [28], the approach is based on a few successive applications of a *pairwise* algorithm, which attempts to form aggregates of size 2. Moreover, for technical reasons, the pairwise algorithm for the initial step and the pairwise algorithm for the further steps are slightly different. Hence we have an "initial pairwise aggregation algorithm" (Algorithm 4.2), a "further pairwise aggregation algorithm"(Algorithm 4.3), and a "multiple pairwise aggregation algorithm"(Algorithm 4.1), the last of which makes explicit the connection between the successive passes.

As seen in Algorithm 4.1, one starts thus with Algorithm 4.2. In the latter, G_0 is first set according to condition (3.13), and this subset is fixed once for all. Then, whenever possible, pairs—that is, aggregates G_k of size 2—are formed in such a way that $\mu(G_k)$ is as small as possible and anyway below the chosen threshold $\bar{\kappa}_{\rm TG}$. The nodes with which one cannot form any valid pair give rise to aggregates of size 1; that is, they are transferred as is to the coarse grid. It is shown in Appendix A that the given expression for $\mu(\{i,j\})$ indeed matches the definition (3.12) (with $\omega = 0.5$) when G contains only two nodes. On the other hand, the conditions that $a_{ii} - s_i + a_{jj} - s_j \geq 0$ and that $\mu(\{i,j\})$ (in fact its denominator) be positive have been added in view of the possibility of having in practice some columns with negative

Algorithm 4.1 (multiple pairwise aggregation).

input: $n \times n$ matrix A
threshold $\bar{\kappa}_{\mathrm{TG}}$
maximal number of passes n_{pass}
target coarsening factor τ

output: n_c and sets $G_0\ ,\ \ldots\ ,\ G_{n_c}$
corresponding aggregation matrix A_c

(1) First pass: Apply *initial pairwise aggregation* Algorithm 4.2
to matrix A with threshold $\bar{\kappa}_{\mathrm{TG}}$;
Output: $n_c^{(1)}$, G_0 and $G_1^{(1)}\ ,\ \ldots\ ,\ G_{n_c}^{(1)}$;

(2) Form $A^{(1)} \leftarrow P^T A P$ with P defined via (3.2)
with respect to $G_1^{(1)}\ ,\ \ldots\ ,\ G_{n_c}^{(1)}$

(3) Iterate: **for** $s = 2, \ldots, n_{\mathrm{pass}}$

(3a) Next passes: Apply *further pairwise aggregation* Algorithm 4.3
to matrix A with threshold $\bar{\kappa}_{\mathrm{TG}}$, $\widetilde{n}_c = n_c^{(s-1)}$,
$\widetilde{G}_k = G_k^{(s-1)}$, $k = 1, \ldots, n_c^{(s-1)}$, and $\widetilde{A} = A^{(s-1)}$;
Output: $n_c^{(s)}$, $G_1^{(s)}\ ,\ \ldots\ ,\ G_{n_c}^{(s)}$;

(3b) Form $A^{(s)} \leftarrow P^T A P$ with P defined via (3.2)
with respect to $G_1^{(s)}\ ,\ \ldots\ ,\ G_{n_c}^{(s)}$

(3c) **if** $\left(\ nnz(A^{(s)}) \le \frac{nnz(A)}{\tau} \ \right)$ **goto** step 4

(4) $n_c \leftarrow n_c^{(s)}$, $G_k \leftarrow G_k^{(s)}$, $k = 1, \ldots, n_c$ and $A_c \leftarrow A^{(s)}$

Algorithm 4.2 (initial pairwise aggregation).

input: $n \times n$ matrix $A = (a_{ij})$
threshold $\bar{\kappa}_{\mathrm{TG}}$

output: n_c and sets $G_0\ ,\ \ldots\ ,\ G_{n_c}$

(0) Initialize: $G_0 = \left\{ i \,\middle|\, a_{ii} \ge \frac{\bar{\kappa}_{\mathrm{TG}}}{\bar{\kappa}_{\mathrm{TG}} - 2} \left(\sum_{k=1, k \ne i}^{n} \frac{|a_{ij} + a_{ji}|}{2} \right) \right\}$

(1) $U \leftarrow [1, n] \backslash G_0$
$n_c \leftarrow 0$
$s_i \leftarrow -\sum_{j \ne i} \frac{a_{ij} + a_{ji}}{2}$ for all $i \in U$

(2) Iterate: **while** $U \ne \emptyset$ **do**

(2a) Select $i \in U$

(2b) Find $j \in U \backslash \{i\}$ such that $a_{ij} \ne 0$,
$a_{ii} - s_i + a_{jj} - s_j \ge 0$ and

$$\mu(\{i,j\}) = \frac{2 \left(\frac{1}{a_{ii}} + \frac{1}{a_{jj}} \right)^{-1}}{-\frac{a_{ij} + a_{ji}}{2} + \left(\frac{1}{a_{ii} - s_i} + \frac{1}{a_{jj} - s_j} \right)^{-1}}$$

is positive and minimal

(2c) $n_c \leftarrow n_c + 1$

(2d) **if** $\left(\ \mu(\{i,j\}) \le \bar{\kappa}_{\mathrm{TG}} \ \right)$ $G_{n_c} = \{i, j\}$; $U \leftarrow U \backslash \{i, j\}$
else $G_{n_c} = \{i\}$; $U \leftarrow U \backslash \{i\}$

Algorithm 4.3 (further pairwise aggregation).

input: $n \times n$ matrix $A = (a_{ij})$
threshold $\bar{\kappa}_{\rm TG}$
tentative $\widetilde{n}_c$ and sets $\widetilde{G}_k$, $k = 1, \ldots, n_c$
corresponding $\widetilde{n}_c \times \widetilde{n}_c$ matrix $\widetilde{A} = (\widetilde{a}_{ij})$

output: n_c and sets G_1 , ... , G_{n_c}

(1) Initialize: $U \leftarrow [1, \widetilde{n}_c]$
$n_c \leftarrow 0$
$\widetilde{s}_i \leftarrow -\sum_{k \in G_i} \sum_{j \notin G_i} \frac{a_{ij}+a_{ji}}{2}$ for all $i \in U$

(2) Iterate: **while** $U \neq \emptyset$ **do**

(2a) Select $i \in U$

(2b) Set $T = \{\, j \mid \widetilde{a}_{ij} \neq 0$ and $\widetilde{a}_{ii} - \widetilde{s}_i + \widetilde{a}_{jj} - \widetilde{s}_j \geq 0$ and $0 < \widetilde{\mu}(\{i,j\}) \leq \bar{\kappa}_{\rm TG} \,\}$, where

$$\widetilde{\mu}(\{i,j\}) = \frac{2 \left(\frac{1}{\widetilde{a}_{ii}} + \frac{1}{\widetilde{a}_{jj}} \right)^{-1}}{-\frac{\widetilde{a}_{ij}+\widetilde{a}_{ji}}{2} + \left(\frac{1}{\widetilde{a}_{ii}-\widetilde{s}_i} + \frac{1}{\widetilde{a}_{jj}-\widetilde{s}_j} \right)^{-1}}$$

(2c) $n_c \leftarrow n_c + 1$

(2d) **if** ($T \neq \emptyset$)
Select $j \in T$ with minimal $\widetilde{\mu}(\{i,j\})$
if ($\mu(\widetilde{G}_i \cup \widetilde{G}_j) \leq \bar{\kappa}_{\rm TG}$)
$G_{n_c} = \widetilde{G}_i \cup \widetilde{G}_j$; $U \leftarrow U \backslash \{i,j\}$
else $T \leftarrow T \backslash \{j\}$; **goto** step 2b
else $G_{n_c} = \widetilde{G}_i$; $U \leftarrow U \backslash \{i\}$

sum (i.e., $s_i > a_{ii}$ or $s_j > a_{jj}$). Then Theorem 3.4 does not apply anymore, but we implement the following heuristic: keep the same rules to define the submatrix $A_G^{(S)}$ associated to a tentative aggregate G, and use the same criterion based on (3.14) to accept or reject the tentative aggregate, providing that $A_G^{(S)}$ is nonnegative definite; otherwise, the aggregate is always rejected; see Appendix A for a proof that $A_G^{(S)}$ is indeed nonnegative definite if and only if the conditions expressed in Algorithm 4.2 are met.

To obtain larger aggregates, Algorithm 4.1 computes the auxiliary coarse grid matrix $\widetilde{A} = (\widetilde{a})_{ij}$ corresponding to this initial pairwise aggregation. Then Algorithm 4.3 is applied to form pairs in this matrix, that is, pairs of aggregates from the previous pass. This latter algorithm is essentially the same as Algorithm 4.2, with a few adaptations. In fact, one cannot compute cheaply $\mu(G)$ if $|G|$ is larger than 2. Therefore, Algorithm 4.3 uses $\widetilde{\mu}(\{i,j\})$ as an estimate of $\mu(G)$ and thus tries to minimize this estimate instead of the true value. However, the condition (3.14) is well checked in the initial matrix. This is cheap because $A_G^{(S)}$ is nonnegative definite and satisfies (3.14) for a given $\bar{\kappa}_{\rm TG}$ if and only if the matrix

$$\omega\, \bar{\kappa}_{\rm TG}\, A_G^{(S)} - D_G \left(I - \mathbf{1}_G (\mathbf{1}_G^T D_G \mathbf{1}_G)^{-1} \mathbf{1}_G^T D_G \right)$$

is nonnegative definite, which can be confirmed by checking that its Cholesky factorization exists (no pivot is negative). On the other hand, we show in Appendix A that $\widetilde{\mu}(\{i,j\})$ as defined in Algorithm 4.3 is a *lower* bound on $\mu(G)$. Hence, it is

sensible to use it as a cheap estimate, whereas this justifies that the algorithm takes into consideration only those pairs for which $\widetilde{\mu}(\{i,j\})$ is below the chosen threshold.

We did not state explicitly how is selected the unknown in U at step 2a of the pairwise aggregation algorithms. For a given unknown i, there may also be several neighbors j for which $\mu(\{i,j\})$ or $\widetilde{\mu}(\{i,j\})$ is minimal. If no rules are specified, the resulting aggregation may be sensitive to the ordering of the unknowns and/or the way offdiagonal entries are stored. Regarding this issue, we proceed verbatim as in [25]. In short (see [25] for details and discussion), we first compute a Cuthill–McKee (CMK) permutation [9] of the top level matrix. Then Algorithm 4.2 is applied giving always (at both steps 2a and 2b) priority to the node with the smallest number in this CMK permutation.

For the subsequent application of Algorithm 4.3, we note, however, that the ordering in the auxiliary matrix $\widetilde{A}$ is driven by the order in which the pairs have been formed during the previous pass; i.e., it is driven by the CMK permutation of the top level matrix. Hence it is no more useful to compute such a permutation. A similar effect is obtained by giving priority to the node with the smallest number in the current ordering. Moreover, the order in which the aggregates are formed then also reflects the initial CMK permutation. Therefore the same policy can be consistently applied during all successive passes of Algorithm 4.3, and further also in Algorithm 4.2 at all coarser levels. Hence, to summarize, the priority is based on the ordering in a CMK permutation during the very first application of Algorithm 4.2, and in all other cases the nodes are processed according their order; i.e., priority is always given to the node with smallest index.

5. Numerical results.

5.1. Illustration of theoretical results. We first illustrate how the aggregation procedure works. In this view we consider the PDE (1.2) on the unit square with Dirichlet boundary conditions everywhere and convective flow as in Example 3.1.4 from [13] (remapped to the unit square; see Problem 2D1 below for complete specification). This flow has zero divergence and is highly varying both in direction and magnitude, as illustrated on Figure 1.

On Figure 2, we depict the aggregation produced by applying Algorithm 4.1 to the matrix resulting from the five-point upwind finite difference approximation of this problem with viscosity $\nu = 10^{-3}$, using a uniform mesh with mesh size $h = 1/64$.

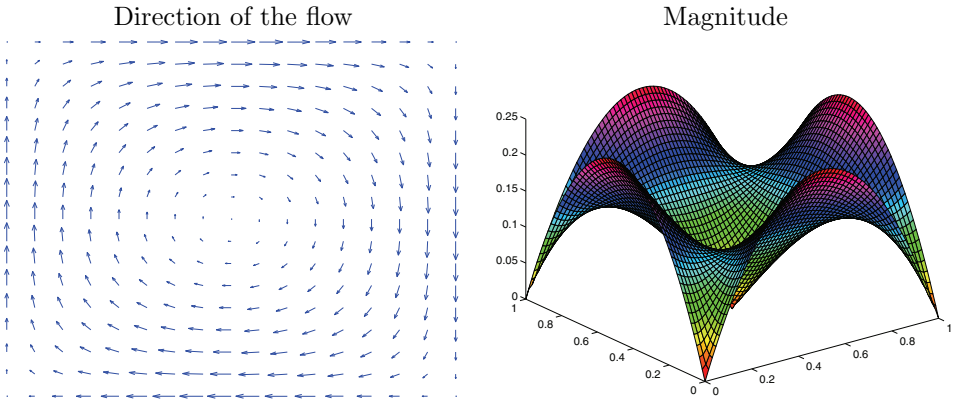

FIG. 1. *Direction and magnitude of the flow for Problem* 2*D*1.

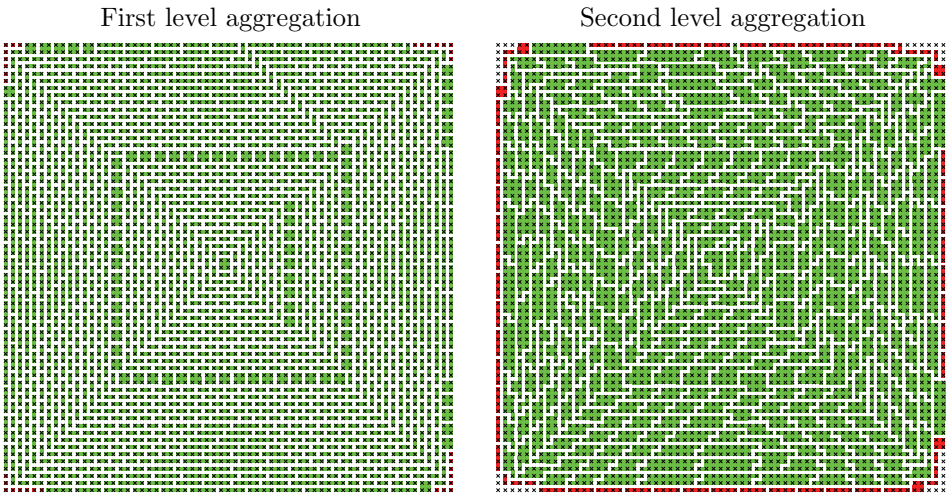

FIG. 2. *Coarsening for Problem* 2D1 *with* $h = 1/64$ *and* $\nu = 10^{-3}$ *; dark (red) boxes correspond to nodes in* G_0*, whereas light gray (green) boxes correspond to aggregates; in the right picture (aggregation generated at the second level), nodes that are not in any box are those in* G_0 *at previous level and hence no longer represented in the coarse grid matrix.*

The parameters used are as follows: threshold $\bar{\kappa}_{\rm TG} = 10$, number of passes $n_{\rm pass} = 2$ and target coarsening factor $\tau = 4$. These choices were indeed found among the most effective in the context of more realistic simulations as reported in the next subsections. We depict the first and the second level aggregation, the latter being obtained from the application of Algorithm 4.1 (with the same parameters) to the coarse grid matrix resulting from the first application. Note that the column sum turns out to be nonnegative everywhere in this example.

Next we illustrate the results of Theorem 3.4. For the same example, considering both the fine grid matrix (first level) and the first coarse grid matrix obtained as indicated above (second level), we depict in Figure 3 the eigenvalues of $B_{\rm TG}^{-1}A$ with $B_{\rm TG}$ defined by (3.5) (one damped Jacobi postsmoothing step with $\omega = 0.5$). We also depict the region which has to contain these eigenvalues, according to both (3.11) and (3.15), and we eventually depict the limit of the convex hull of the eigenvalues of $\omega D^{-1}A$. This allows us to see that the coarse grid correction is indeed necessary to move the eigenvalues away from the origin.

We turn then to a more efficient smoothing scheme, although not covered by the theory. Namely, we consider Gauss–Seidel smoothing, more precisely one forward Gauss–Seidel sweep for presmoothing and one backward Gauss–Seidel sweep for postsmoothing, that is, the preconditioner $B_{\rm TGS}$ defined from $I - B_{\rm TGS}^{-1}A = T$ with respect to the iteration matrix (3.1), where $\nu_1 = \nu_2 = 1$ and $M_1 = {\rm low}(A)$, $M_2 = {\rm upp}(A)$ (with, as usual, $R = P^T$ and $A_c = P^T A P$). The eigenvalues of $B_{\rm TGS}^{-1}A$ are depicted in Figure 4, with, again, the convex hull of the eigenvalues of the matrix preconditioned by the action of the smoother alone, that is, the eigenvalues of $M^{-1}A$, where $M^{-1} = M_1^{-1} + M_2^{-1} - M_2^{-1}AM_1^{-1}$ gathers the actions of the pre- and postsmoother. (Hence $M = {\rm low}(A)D^{-1}{\rm upp}(A)$ corresponds to symmetric Gauss–Seidel preconditioning.)

Now, considering, for example, right preconditioning, the convergence is fully characterized by the eigenvalues only if the preconditioned matrix $AB_{\rm TGS}^{-1}$ is normal,

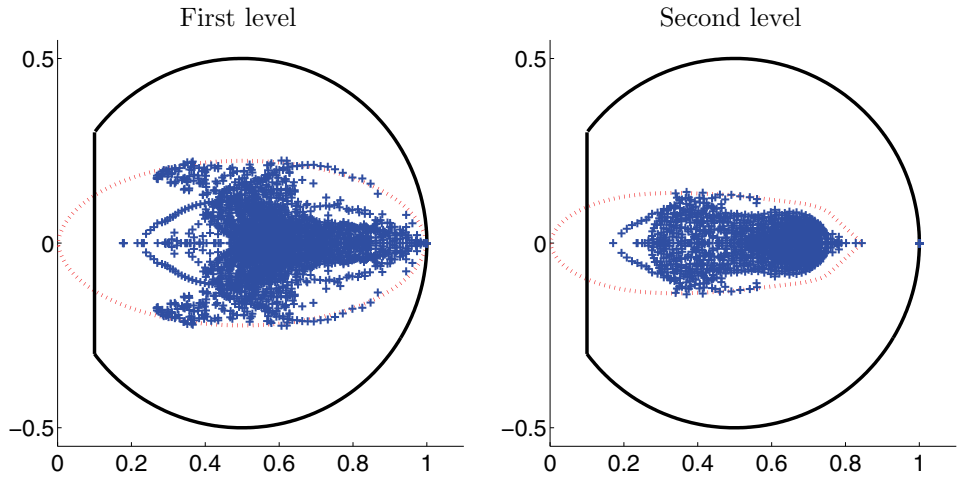

FIG. 3. *One damped Jacobi smoothing step;* +*:* $\sigma(B_{TG}^{-1}A)$ *;* —*: limit of the region of inclusion guaranteed by Theorem* 3.4*;* ıı*: limit of the convex hull of* $\sigma(\omega D^{-1}A)$ *.*

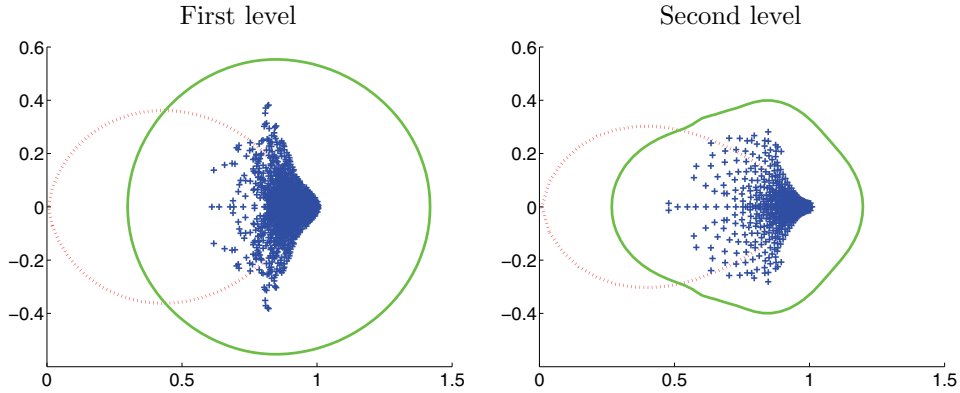

FIG. 4. *Gauss–Seidel smoothing;* +*:* $\sigma(B_{\mathrm{TGS}}^{-1}A)$ *;* ıı*: limit of the convex hull of* $\sigma(M^{-1}A)$ *;* —*: boundary of the* ε-pseudospectrum *of* AB_{TGS} *for* $\varepsilon = 1/20$ *.*

that is, has orthogonal eigenvectors. Otherwise, there may be some significant gap between the actual convergence and the one expected on the basis of, for example, the spectral radius of the iteration matrix with optimal scaling (3.6). Such "nonnormality" effects tend to be pronounced for discretized convection diffusion problems; see the analyses in [15, 20] and [39, Chapter 25]. One way to assess them is to compute, beside the spectrum, the so-called *pseudospectrum*. There are several equivalent definitions of the latter [39, Chapter 2]; here we recall only the most intuitive one: the ε-*pseudospectrum* $\sigma_\varepsilon(C)$ of a square matrix C is the set of $z \in \mathcal{C}$ such that $\|(z\,I - C)^{-1}\| \geq \varepsilon^{-1}$.

Summarizing roughly the analysis in Chapter 26 of [39], effective convergence of GMRES [36] (or other norm minimizing methods) is guaranteed if the ε*–pseudospectrum* is bounded and away from the origin. Apply this now to the right preconditioned matrix AB_{TGS}^{-1} . Observe $\|(z\,I - AB_{\mathrm{TGS}}^{-1})^{-1}\|$ is infinite at an eigenvalue. Hence the requirement to have eigenvalues bounded and away from the origin is turned into

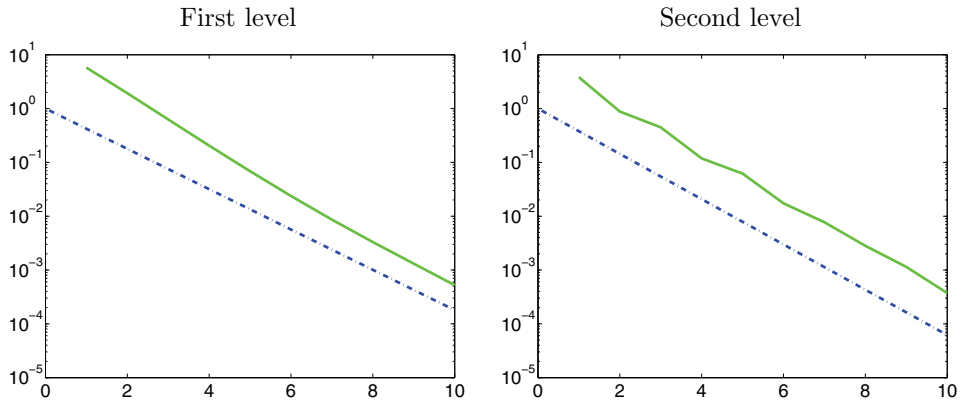

FIG. 5. *Gauss–Seidel smoothing;* —: $\|(I - \alpha_{opt} A B_{\mathrm{TGS}}^{-1})^k\|$; ·-·-: $(\rho(I - \alpha_{opt} A B_{\mathrm{TGS}}^{-1}))^k$.

a stronger requirement: to have the whole region where $\|(z\,I - AB_{\mathrm{TGS}}^{-1})^{-1}\|$ is large bounded and away from the origin. To check this, we also depicted in Figure 4 the boundary of the ε-*pseudospectrum* for $\varepsilon = 1/20$, that is, the boundary of the region in which $\|(z\,I - AB_{\mathrm{TGS}}^{-1})^{-1}\|$ is larger than 20. One sees that this region is, as desired, away from the origin and reasonably close to the convex hull of the spectrum.

Regarding the convergence of GMRES with right preconditioning, recall also that after k steps the residual satisfies

$$\|\mathbf{r}_k\| = \min_{\mathcal{P}_k:\ \text{pol. deg. } k \text{ s.t. } \mathcal{P}_k(0)=1} \|\mathcal{P}_k\left(AB_{\mathrm{TGS}}^{-1}\right)\mathbf{r}_0\| \ ,$$

where $\mathbf{r}_0$ is the initial residual. Hence

$$\frac{\|\mathbf{r}_k\|}{\|\mathbf{r}_0\|} \leq \|\left(I - \alpha AB_{\mathrm{TGS}}^{-1}\right)^k\|$$

for any α. In Figure 5, we plot this quantity as a function of k, for the α that minimizes the spectral radius of $I - \alpha AB_{\mathrm{TGS}}^{-1}$. This spectral radius is then equal to the asymptotic convergence factor $\rho_{s_{\mathrm{opt}}}$ of a fixed point iteration with optimal scaling defined in (3.6). In Figure 5, we also plot $(\rho_{s_{\mathrm{opt}}})^k$ as a function of k, as it represents in some sense an "ideal" convergence curve. One sees that the bound on the effective convergence rate given by $\|(I - \alpha AB_{\mathrm{TGS}}^{-1})^k\|$ follows this latter with only a slight delay of about two iterations.

Both this latter observation and the pseudospectrum depicted in Figure 4 suggest that our approach not only is effective with respect to the eigenvalues, but also allows us to master the nonnormality effects often associated with convection-diffusion problems.

5.2. Large scale problems. Here we want to further assess the potentialities of the method to solve large two- and three-dimensional problems. In this view we consider the following set of test examples, defined by the PDE (1.2) with $f = 0$, Dirichlet boundary conditions ($\Gamma_0 = \partial\Omega$) with $g_0 = 0$ everywhere except $g_0 = 1$ on the right boundary in two-dimensional cases and on the top boundary in three-dimensional ones. In all cases, we consider the five point (two-dimensional) or seven point (three-dimensional) upwind finite difference approximation on a uniform mesh with mesh size h in all directions. The domains and convective flows are as follows:

PROBLEM 2D1. $\Omega =$ unit square, $\overline{v}(x\,,\,y) = \begin{pmatrix} x(1-x)(2\,y-1) \\ -(2\,x-1)y(1-y) \end{pmatrix}$.

PROBLEM 2D2. $\Omega =$ unit square, $\overline{v}(x\,,\,y) = \begin{pmatrix} \cos(\pi x)\,\sin(\pi y) \\ -\sin(\pi x)\,\cos(\pi y) \end{pmatrix}$.

PROBLEM 2D3. $\Omega =$ unit square, $\overline{v}(x\,,\,y) = \begin{pmatrix} \sin(2\pi x)\,\cos(2\pi y) \\ -\cos(2\pi x)\,\sin(2\pi y) \end{pmatrix}$
if $x\,, y \le \frac{1}{2}$ and $\overline{v}(x\,,\,y) = 0$ otherwise.

PROBLEM 3D1. $\Omega =$ unit cube, $\overline{v}(x\,,\,y\,,\,z) = \begin{pmatrix} 2x(1-x)(2\,y-1)z \\ -(2\,x-1)y(1-y) \\ -(2x-1)(2y-1)z(1-z) \end{pmatrix}$.

PROBLEM 3D2. $\Omega =$ unit cube, $\overline{v}(x\,,\,y,,\,z) = \begin{pmatrix} \sin(2\pi x)\,\cos(\pi y)\,\cos(\pi z) \\ -\cos(2\pi x)\,\sin(\pi y)\,\cos(\pi z) \\ -\cos(2\pi x)\,\cos(\pi y)\,\sin(\pi z) \end{pmatrix}$
if $x \le \frac{1}{2}$ and $\overline{v}(x\,,\,y,\,z) = 0$ otherwise.

PROBLEM 3D3. $\Omega =$ unit cube, $\overline{v}(x\,,\,y,,\,z) = \begin{pmatrix} (y-0.5)(z-0.5) \\ (x-0.5)(z-0.5) \\ -2(x-0.5)(y-0.5) \end{pmatrix}$
if $|(x\,,\,y,\,z) - (0.5\,,\,0.5\,,\,0.5)| \le \frac{2}{5}$ and $\overline{v}(x\,,\,y,\,z) = 0$ otherwise.

The flow for Problem 2D1 is illustrated in Figure 1. The flow is similarly varying and rotating in other examples, while being sometimes zero in part of the domain.

In each case, we recursively apply the aggregation algorithm to generate the needed hierarchy of coarse systems and associated prolongations. We uniformly use the parameters indicated above: threshold $\bar{\kappa}_{\rm TG} = 10$, number of passes $n_{\rm pass} = 2$, and target coarsening factor $\tau = 4$.

The coarsening is stopped when the number of unknowns is below $40\,n^{1/3}$. This is indeed sufficiently small to make negligible (relatively to fine grid operations) the time needed to perform an exact factorization of the matrix with a sparse direct solver. In the present case we use MUMPS [22]. Occasionally (mainly for Problem 3D2), the coarsening becomes slow from a certain stage. This seems due to columns with negative sum: the corresponding nodes are often not aggregated by our algorithm, and these nodes are then transferred as is to the coarser grid. To avoid an excessive increase of the complexity in such cases, when it is detected that the number of nonzero entries in the matrix is reduced by a factor less than 2 from one grid to the next, then the coarse grid matrix is factorized when the number of unknowns is below $400\,n^{1/3}$. Note that these choices were performed in a fully automatic way, the solver being called in a black box fashion. Note also that, as illustrated later in this section, it is actually harmless to pursue the coarsening until the number of unknowns is fairly small, with the exception of those few problems being when the coarsening is slow from a certain stage. In these cases a fairly small size cannot be attained without changing the aggregation algorithm, for instance, introducing some heuristics to better deal with situations where the column sum is negative at a significant fraction of the nodes.

The coarsening scheme is combined with Gauss–Seidel smoothing and the K-cycle to define a multigrid preconditioner. The smoothing scheme is the one described above. Using the K-cycle means that, at each level of the hierarchy, a two-grid preconditioner is defined in which the coarse grid system is solved approximately with a few steps of a Krylov subspace iterative method; this definition is then recursive because for this inner iteration we use the two-grid preconditioner on that coarser level;

TABLE 1
Computational complexities and number of iterations.

		$\mathcal{C}_A$			$\mathcal{C}_W$			#Iterations		
Problem	ν	S1	S2	S3	S1	S2	S3	S1	S2	S3
2D1	10^0	1.3	1.3	1.3	1.9	2.0	2.0	10	9	9
	10^{-2}	1.4	1.3	1.3	2.1	2.0	2.0	12	10	9
	10^{-4}	1.8	1.9	1.9	3.5	3.8	3.7	16	18	22
	10^{-6}	1.6	1.6	1.6	2.8	2.9	3.0	14	14	14
2D2	10^0	1.3	1.3	1.3	2.0	2.2	2.1	10	9	9
	10^{-2}	1.5	1.4	1.3	2.6	2.5	2.1	19	22	11
	10^{-4}	1.5	1.5	1.5	2.6	2.6	2.6	12	16	21
	10^{-6}	1.4	1.4	1.4	2.5	2.5	2.5	7	8	10
2D3	10^0	1.3	1.3	1.3	2.0	2.0	2.0	9	10	10
	10^{-2}	1.5	1.4	1.3	2.4	2.3	2.0	14	13	10
	10^{-4}	1.4	1.4	1.5	2.3	2.4	2.5	13	14	14
	10^{-6}	1.4	1.4	1.4	2.2	2.2	2.3	17	18	15
3D1	10^0	1.3	1.3	1.3	1.9	2.0	2.0	8	8	8
	10^{-2}	1.6	1.6	1.6	3.2	3.5	3.7	11	11	11
	10^{-4}	1.7	1.7	1.7	3.6	3.9	4.0	11	11	11
	10^{-6}	1.6	1.6	1.6	3.4	3.6	3.8	14	15	17
3D2	10^0	1.4	1.4	1.3	2.4	2.2	2.3	8	8	8
	10^{-2}	1.5	1.5	1.5	3.0	3.1	2.9	11	11	12
	10^{-4}	1.5	1.5	1.5	3.2	3.3	3.0	12	12	12
	10^{-6}	1.5	1.5	1.5	3.2	3.1	3.2	13	15	16
3D3	10^0	1.3	1.3	1.3	1.9	2.0	2.0	8	8	8
	10^{-2}	1.6	1.6	1.6	3.0	3.3	3.7	11	12	13
	10^{-4}	1.6	1.5	1.5	3.2	3.3	3.4	12	13	14
	10^{-6}	1.5	1.5	1.5	3.1	3.2	3.4	13	14	15

see, e.g., [28] for details and algorithms. Here we use a simple Galerkin projection that makes the residual orthogonal to the search vectors, as does the conjugate gradient method in the symmetric case.[2]

We use the generalized conjugate residual (GCR) method [12] as the main iterative method on the finest level. This Krylov subspace method provides the minimal residual norm solution and allows for variable preconditioning [41]. To avoid excessive memory requirements, the method is restarted each 10 iterations. In all cases we use the zero vector as initial approximation, and iterations were stopped when the relative residual error was below 10^{-6}. Note that such an extensive use of Krylov subspace methods has also been advised to enhance the robustness of geometric multigrid methods when applied to convection-diffusion problems; see [32, 33].

Each problem was tested for viscosity ν ranging from 1 (diffusion dominating) to 10^{-6} (convection dominating), and different mesh sizes according to the following table:

	2D problems		3D problems	
	h^{-1}	n	h^{-1}	n
S1	600	$3.6\ 10^5$	80	$5.1\ 10^5$
S2	1600	$2.5\ 10^6$	160	$4.1\ 10^6$
S3	5000	$2.5\ 10^7$	320	$3.3\ 10^7$

Basic statistics are reported in Table 1 and timing results in Table 2. Test reported here and in the next subsection were run on a computer with two Intel XEON L5420

[2]This turns out to be slightly faster on average than the GCR method used in [28] and has become the default in the AGMG software [26].

TABLE 2

Setup time, solution time and total time reported here per million of unknowns (with $\mathcal{T}_{\text{total}} = \mathcal{T}_{\text{sol}} + \mathcal{T}_{\text{setup}}$*).*

		$\mathcal{T}_{\text{setup}}$			$\mathcal{T}_{\text{sol}}$			$\mathcal{T}_{\text{tot}}$		
Problem	ν	S1	S2	S3	S1	S2	S3	S1	S2	S3
2D1	10^0	0.9	0.8	1.1	2.1	2.1	2.2	3.1	2.9	3.3
	10^{-2}	1.0	0.8	1.1	2.6	2.4	2.2	3.6	3.2	3.4
	10^{-4}	1.6	1.5	1.9	5.3	6.7	8.7	6.8	8.2	10.5
	10^{-6}	1.4	1.1	1.5	3.9	4.3	4.6	5.3	5.5	6.1
2D2	10^0	1.0	0.8	1.2	2.3	2.4	2.3	3.3	3.2	3.5
	10^{-2}	1.1	0.9	1.1	5.2	6.3	2.9	6.3	7.2	4.1
	10^{-4}	1.1	1.0	1.3	2.9	4.5	6.3	4.0	5.6	7.6
	10^{-6}	1.1	1.0	1.3	1.8	2.0	2.6	2.9	3.0	3.9
2D3	10^0	1.0	0.8	1.2	2.0	2.4	2.5	3.0	3.2	3.7
	10^{-2}	1.1	0.9	1.1	3.5	3.4	2.5	4.5	4.4	3.7
	10^{-4}	1.0	0.9	1.3	3.1	3.7	4.0	4.1	4.7	5.3
	10^{-6}	1.1	0.9	1.3	4.1	4.7	4.1	5.2	5.6	5.3
3D1	10^0	1.4	1.9	2.3	1.9	2.1	2.3	3.3	4.1	4.6
	10^{-2}	1.7	2.5	2.9	4.2	5.1	5.6	5.9	7.5	8.5
	10^{-4}	1.9	2.8	3.1	4.7	5.8	6.0	6.6	8.6	9.2
	10^{-6}	1.9	2.6	3.1	5.6	7.4	9.3	7.5	10.0	12.4
3D2	10^0	1.3	2.0	2.3	2.4	2.6	2.7	3.7	4.6	5.0
	10^{-2}	1.5	2.2	2.6	4.1	4.8	5.7	5.6	7.0	8.3
	10^{-4}	1.6	2.3	2.8	4.8	5.6	6.1	6.4	7.9	8.9
	10^{-6}	1.6	2.4	2.8	5.1	7.3	8.7	6.7	9.7	11.5
3D3	10^0	1.3	1.9	2.3	2.0	2.2	2.4	3.2	4.1	4.6
	10^{-2}	1.6	2.4	2.9	4.0	5.3	6.5	5.6	7.7	9.4
	10^{-4}	1.7	2.5	2.9	4.6	5.8	6.8	6.3	8.3	9.6
	10^{-6}	1.6	2.5	3.0	4.9	6.3	7.6	6.5	8.8	10.5

processors at 2.50 GHz and 16 Gb RAM memory.

In Table 1, we give both the standard *operator complexity* and the *weighted complexity*, respectively defined as

$$\mathcal{C}_A = \sum_{\ell=1}^{L} \frac{nnz(A_\ell)}{nnz(A_1)}, \qquad \mathcal{C}_W = \sum_{\ell=1}^{L} 2^{\ell-1} \frac{nnz(A_\ell)}{nnz(A_1)},$$

where $nnz(A_\ell)$ is the number of nonzero entries in A_ℓ, A_1 standing for the fine grid matrix, A_2 for the first coarse grid one, etc (L is the number of levels). Compared with the operator complexity, the weighted complexity incorporates the factor $2^{\ell-1}$ to take into account that two inner iterations are performed at each level; hence $\mathcal{C}_W$ correctly reflects the cost of one fine grid iteration step [25]. It therefore characterizes the success or failure of the approach, together with the number of iterations which informs about the convergence speed.

One sees that small viscosities tend to induce somewhat larger weighted complexities; $\mathcal{C}_W$ is nevertheless equal to at most 3.7, i.e., less than twice the value 2, which is the standard bound on the weighted complexity that is obtained when the coarsening is as desired, that is, reduces the number of nonzero entries by a factor of 4 from one level to the next. Diffusion-dominating problems also lead to faster convergence, and, for small viscosities, one sees that in some cases the number of iterations slightly increases with the mesh size. Recall, however, that the size S3 is fairly large and with about 100 times more unknowns than the size S1. On the other hand one may observe that $\mathcal{C}_A$ remains below 2 in all cases, which indicates that the

TABLE 3

Weighted computational complexity, number of iterations, and total time per million of unknowns for Problem 3D1 with size S2.

$\bar{\kappa}_{\rm TG}$	8	9	10	11	12	14	16	20	30
				$n_{\rm pass}=2$					
$\mathcal{C}_W$	10.0	5.2	4.0	3.5	3.2	2.9	2.8	2.7	2.6
#Iterations	15	15	15	16	16	18	18	19	19
$\mathcal{T}_{\rm tot}$	15.4	11.6	10.1	9.7	9.2	9.4	9.0	9.2	9.1
				$n_{\rm pass}=3$					
$\mathcal{C}_W$	7.2	4.2	3.2	2.7	2.4	2.1	2.0	1.8	1.6
#Iterations	15	16	16	18	19	20	21	24	26
$\mathcal{T}_{\rm tot}$	15.5	12.5	10.8	10.5	10.0	9.4	9.2	9.2	8.8

additional memory needed to store the preconditioner does not exceed the memory needed to store the system matrix.

Regarding timings, the setup time varies between 0.8 and 3.1 seconds per million of unknowns. It is always below the solution time, which varies between 2.1 and 9.3 seconds per million of unknowns, for a total time ranging between 3.1 and 12.4 seconds per million of unknowns.

Eventually, it is interesting to check the influence of the main parameters of our coarsening algorithm. This is done in Table 3 for Problem 3D1 with size S2. Note that in these tests the coarsening is pursued until the number of coarse unknowns does not exceed 10, making the performance of the algorithm even more critical. As expected, relaxing the constraints by increasing the threshold and/or the number of passes is beneficial with respect to the weighted complexity but may result in somewhat slower convergence, whereas the solution time is remarkably stable, except when going to smaller threshold $\bar{\kappa}_{\rm TG}$.

5.3. Comparison with other methods. Here we first compare our new method as described in the previous subsection (referred to as AggGar) with the following competitors.

1. AGMG 2.3, the 2.3 version of the AGMG software [26], which implements an aggregation-based method similar the one in this paper except that the aggregation procedure is based on the heuristic method described in [28].

2. The classical AMG method from [38] as implemented in the HSL Library [17]. Note that the comparison of solution time is meaningful because this library meets the highest quality standards, and also because we integrated it into the same software infrastructure as those used for testing aggregation-based methods, using the same top-level iterative method and only substituting calls for setup and preconditioner application for those provided by the library.

3. For two-dimensional problems, the sparse direct solver available in MATLAB via the "\" command.

Results are reported in Table 4 (results for $\nu = 1$ are omitted because they are nearly identical to those obtained for $\nu = 10^{-2}$). One sees that the theoretically well founded coarsening approach presented in Algorithms 4.1–4.3 allows us to obtain faster code on average with more stable performances than with the heuristic aggregation algorithm used in AGMG 2.3. The difference, however, is not that dramatic, probably because the coarsening driven by the minimization of $\mu(\{i,j\})$ or $\widetilde{\mu}(\{i,j\})$ is, in practice, not far from the coarsening driven by the strongest negative coupling used in [28] (observe that μ or $\widetilde{\mu}$ decreases when the offdiagonal coupling becomes more negative).

TABLE 4
Total time for the different solvers, reported per million of unknowns.

	$\nu = 10^{-2}$			$\nu = 10^{-4}$			$\nu = 10^{-6}$		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
Problem 2D1									
AggGar	3.6	3.2	3.4	6.8	8.2	10.5	5.3	5.5	6.1
AGMG 2.3	5.3	6.0	6.4	29.9	8.3	10.4	6.7	7.6	8.0
AMG	3.5	3.5	4.1	36.8	16.8	6.6	208.1	–	–
MATLAB "\"	14.9	21.5		15.5	30.2		30.4	67.3	
Problem 3D1									
AggGar	5.9	7.5	8.5	6.6	8.6	9.2	7.5	10.0	12.4
AGMG 2.3	7.2	8.9	10.2	8.5	12.9	15.9	8.4	12.9	17.6
AMG	12.8	13.8	2.6e3	5.3e4	–	–	1.2e5	–	–

On the other hand, AMG performs well for moderate ν. However, for $\nu = 10^{-6}$ in two-dimensional cases and $\nu = 10^{-4}$, 10^{-6} in three-dimensional ones, it turns out that the coarsening breaks down at some early stage; the coarsest grid matrix is then very large, hence its factorization is no longer feasible or slows down the whole process, even using the highly efficient sparse direct solver that comes with the HSL Library. Note for completeness that the slowness of AMG for the largest three-dimensional example with $\nu = 10^{-2}$ is not due to a coarsening failure, but probably to the fact that the code starts swapping because the memory requirements exceed the available RAM memory (16 Gb as indicated above).

Eventually, the comparison with MATLAB "\" is interesting because this method is known to be practically scalable and very hard to beat for two-dimensional problems, as long as there is enough memory. Indeed, for the two-dimensional Poisson problem approximated by five point finite difference (SPD matrix), results reported in [25] show that this method requires 6.2 and 6.6 seconds per million of unknowns for sizes S1 and S2, respectively, and breaks down because of insufficient memory only for size S3. Somewhat surprisingly, using to the nonsymmetric version to solve the two-dimensional convection-diffusion problem requires more than twice the time and seems to entail a significant loss of scalability. Even more surprisingly, a stronger degradation is observed for very small viscosity. Hence we have the unexpected result that, despite some variations in performances, our aggregation-based AMG method is actually more robust than a reference direct solver.

In addition, it is also interesting to compare our approach with "smoothed aggregation AMG" [6, 7, 37, 42]. Unfortunately, we did not find an implementation that could be easily integrated into our software infrastructure and therefore allow a fair comparison of solution times. In particular, the most robust version for challenging convection-diffusion problems seems the one developed in [7], but the high-level MATLAB implementation used to produce the numerical results in this paper is, as written in [7], "not an environment that is reasonable for timing." However, Problem 2D1 is among the test problems in this paper (under the name "recirc"), as well as in [37], where other versions of smoothed aggregation AMG are tested. Then we decided to repeat the experiments in these papers, and give the results for our method next to those for other approaches as published in [7, 37]. Note that we adapted our tests to the specifications in these papers, regarding the stopping criterion, the right-hand side, and even the discretization stencil, which in [7] is slightly different from that presented in section 2. Note also that because the results reported in [7, 37] do not correspond to very fine meshes, we also changed the default in our code, and coarsen-

TABLE 5
Number of iterations for Problem 2D1, using the specifications from [37] *(problem recirc).*

$h^{-1} = 512$ (V-cycle for NSA, SA, and EMIN)

ν	$2.5\,10^{-2}$	$2.5\,10^{-3}$	$2.5\,10^{-4}$	$2.5\,10^{-5}$	$2.5\,10^{-6}$
AggGar	14	26	22	18	21
NSA	92	189	–	–	–
SA	9	12	33	–	–
EMIN	9	10	21	65	119

$\nu = 2.5\,10^{-6}$ (W-cycle for EMIN)

h^{-1}	64	128	256	512	1024
AggGar	15	17	19	21	21
EMIN	22	38	55	65	61

ing is here pursued until the number of coarse unknowns does not exceed 10. Hence good results of AggGar cannot be explained by the fact that the method truncates to only a few levels for the given problem sizes.

In Table 5, we compare our results with those in [37]; NSA stands for "nonsmoothed aggregation" and may be seen as representative of what can be obtained with a naive implementation of plain aggregation, using an aggregation procedure tailored for classical smoothed aggregation and resorting to the V-cycle, i.e., ignoring the K-cycle. SA stands for classical smoothed aggregation along the lines of the seminal work for symmetric problems [42], and EMIN for the particular nonsymmetric variant developed in [37]. The first part of the table is not completely meaningful because results in [37] are for V-cycle variants that should be cheaper per iteration step than AggGar. However, as acknowledged in this paper, all of these variants clearly lack scalability. This motivated the authors of [37] to perform further tests with the W-cycle, the clear winner being then EMIN. The second line of the table compares AggGar with these latter results. Possibly, even with the W-cycle, the weighted complexity is somehow smaller for EMIN as it is for AggGar in the most difficult examples, but, on the other hand, the smoothing procedure used for EMIN is twice more costly than the smoothing procedure in AggGar.

In [7], the quantity reported is the number of work units needed to reduce the error by a factor of 10 (i.e., to gain one digit of accuracy), using as convergence factor the mean convergence factor for the last 5 iterations in a sequence of 25; one work unit represents the cost of one fine grid residual iteration. We computed the same quantity for AggGar, estimating the number of work units from the flop count reported by the program (whereas in [7] the number of work units per iteration is estimated from complexities). Results are reported in Table 6, where SA and αSA refer to the particular nonsymmetric variants developed in [7], which thus differ from the SA and αSA methods in [6, 42]; note that αSA is an "adaptive" variants that requires a significant amount of preprocessing.

6. Conclusions. We have developed a theoretical analysis that applies to any M-matrix with row and column sums both nonnegative. This allows us to cover linear systems arising from the upwind finite difference discretization of convection-diffusion equations with divergence-free, but possibly variable, convective flow. An aggregation procedure has been proposed that allows us to automatically match the conditions that guarantee the convergence of a model two-grid scheme. More realistic two-grid schemes and the recursive use of the approach to solve fairly large problems have been

TABLE 6
Number of work units per digit of accuracy for Problem 2D1, using the specifications from [7] (problem recirc).

	$h^{-1}=64$			$h^{-1}=128$		
ν	$3.9\,10^{-2}$	$3.9\,10^{-4}$	$3.9\,10^{-6}$	$1.9\,10^{-2}$	$1.9\,10^{-4}$	$1.9\,10^{-6}$
AggGar	13.3	25.4	17.9	14.2	29.8	20.2
SA [7]	16.2	40.8	40.7	20.1	76.6	68.6
αSA [7]	9.9	27.0	42.0	10.7	48.7	69.4

investigated through numerical experiments. The results suggest that the proposed technique can indeed lead to a fast and robust solver. Some sensitivity with respect to the viscosity parameter is observed, as well as, for small viscosities, some dependency with respect to the mesh size. These effects are, however, limited and in fact less pronounced than for a reference direct solver and other algebraic multigrid methods.

Software. The results of this research have been integrated into the AGMG software [26], whose version 3.1.2 is nearly identical to the code used to run the numerical experiments in sections 5.2 and 5.3.

Appendix A. We first show that the conditions $a_{ii}-s_i+a_{jj}-s_j\geq 0$ and $\mu(\{i,j\})>0$ used in step (2b) of Algorithm 4.2 are met if and only if $A^{(S)}_{\{i,j\}}$ is nonnegative definite. Consider an aggregate $G=\{i,j\}$, and let $s_i=-\sum_{k\neq i}\frac{a_{ik}+a_{ki}}{2}$, $s_j=-\sum_{k\neq j}\frac{a_{jk}+a_{kj}}{2}$, $\alpha=-\frac{a_{ij}+a_{ji}}{2}$. In this context,

$$A^{(S)}_{\{i,j\}}=\begin{pmatrix} a_{ii}-s_i+\alpha & -\alpha \\ -\alpha & a_{jj}-s_j+\alpha \end{pmatrix}=\alpha\begin{pmatrix}1&-1\\-1&1\end{pmatrix}+\begin{pmatrix}\delta_1&\\&\delta_2\end{pmatrix},$$

where $\delta_1=a_{ii}-s_i$ and $\delta_2=a_{jj}-s_j$.

Noting $\mathbf{1}_2=\mathbf{1}_{\{i,j\}}=(1\ 1)^T$, a necessary and sufficient condition for this matrix being nonnegative definite is $\mathbf{1}_2^T A^{(S)}_{\{i,j\}}\mathbf{1}_2\geq 0$ together with $\det\left(A^{(S)}_{\{i,j\}}\right)\geq 0$, yielding the conditions $\delta_1+\delta_2\geq 0$ and $\alpha(\delta_1+\delta_2)+\delta_1\delta_2\geq 0$, latter of which is itself equivalent to the requirement $\mu(\{i,j\})>0$, since $(1/\delta_1^{-1}+1/\delta_2^{-1})^{-1}=(\delta_1\delta_2)/(\delta_1+\delta_2)$.

We now show that the expression used for $\mu(\{i,j\})$ in step (2b) of Algorithm 4.2 matches the definition (3.12) when $A^{(S)}_{\{i,j\}}$ is nonnegative definite. One has

$$\begin{aligned} &D_G(I-\mathbf{1}_G(\mathbf{1}_G^TD_G\mathbf{1}_G)^{-1}\mathbf{1}_G^TD_G)\\ &\quad=\begin{pmatrix}a_{ii}&\\&a_{jj}\end{pmatrix}\left(I-\begin{pmatrix}1\\1\end{pmatrix}(a_{ii}+a_{jj})^{-1}\begin{pmatrix}1&1\end{pmatrix}\begin{pmatrix}a_{ii}&\\&a_{jj}\end{pmatrix}\right)\\ &\quad=\frac{a_{ii}\,a_{jj}}{a_{ii}+a_{jj}}\begin{pmatrix}1&-1\\-1&1\end{pmatrix}. \end{aligned}$$

Hence, for nonnegative definite $A^{(S)}_{\{i,j\}}$, (3.12) amounts to

$$\mu(G)=\omega^{-1}\frac{\dfrac{a_{ii}\,a_{jj}}{a_{ii}+a_{jj}}}{\alpha\ +\ \inf_{(v_1,v_2)\neq(1,1)}\dfrac{\begin{pmatrix}v_1&v_2\end{pmatrix}\begin{pmatrix}\delta_1&\\&\delta_2\end{pmatrix}\begin{pmatrix}v_1\\v_2\end{pmatrix}}{\begin{pmatrix}v_1&v_2\end{pmatrix}\begin{pmatrix}1&-1\\-1&1\end{pmatrix}\begin{pmatrix}v_1\\v_2\end{pmatrix}}}.$$

From there, one recovers the expression in step (2b) of Algorithm 4.2, noting that the infimum on the right-hand side is equal to zero if any of the δ_i is zero, and, otherwise, is obtained for $(v_1\,,\,v_2)=(-\delta_2\,,\,\delta_1)$ (using again $(1/\delta_1^{-1}+1/\delta_2^{-1})^{-1}=(\delta_1\delta_2)/(\delta_1+\delta_2)$).

Eventually, we show that the expression used for $\widetilde{\mu}(\{i,j\})$ in Algorithm 4.3 is a lower bound on $\mu(G_i\cup G_j)$ (as defined in (3.12)), where G_i and G_j are the aggregates corresponding to nodes i and j (respectively) in the auxiliary coarse grid matrix $\widetilde{A}$.

Let $\widetilde{s}_i=\sum_{k\neq i}\frac{|\widetilde{a}_{ik}+\widetilde{a}_{ki}|}{2}$, $\widetilde{s}_j=\sum_{k\neq j}\frac{|\widetilde{a}_{jk}+\widetilde{a}_{kj}|}{2}$, $\widetilde{\alpha}=\frac{|\widetilde{a}_{ij}+\widetilde{a}_{ji}|}{2}$. Also let

$$\widetilde{A}^{(S)}_{\{i,j\}}=\begin{pmatrix}\widetilde{a}_{ii}-\widetilde{s}_i+\widetilde{\alpha} & -\widetilde{\alpha}\\ -\widetilde{\alpha} & \widetilde{a}_{jj}-\widetilde{s}_j+\widetilde{\alpha}\end{pmatrix}\,,\ \widetilde{D}_{\{i,j\}}=\begin{pmatrix}\widetilde{a}_{ii} & \\ & \widetilde{a}_{jj}\end{pmatrix}\,.$$

Repeating the above reasoning then proves

$$\widetilde{\mu}(\{i,j\})=\omega^{-1}\sup_{\mathbf{v}\notin\mathcal{N}\left(\widetilde{A}^{(S)}_{\{i,j\}}\right)}\frac{\mathbf{v}\widetilde{D}_{\{i,j\}}\left(I-\mathbf{1}_2\left(\mathbf{1}_2^T\widetilde{D}_{\{i,j\}}\mathbf{1}_2\right)^{-1}\mathbf{1}_2^T\widetilde{D}_{\{i,j\}}\right)\mathbf{v}}{\mathbf{v}^T\widetilde{A}^{(S)}_{\{i,j\}}\mathbf{v}}\,. \tag{A.1}$$

On the other hand, let

$$P_f=\begin{pmatrix}\mathbf{1}_{G_i} & \\ & \mathbf{1}_{G_j}\end{pmatrix}\,.$$

In Appendix B of [25], it is shown that

$$\widetilde{A}^{(S)}_{\{i,j\}}=P_f{}^T\,A^{(S)}_{G_i\cup G_j}P_f\,,$$

whereas it is clear from (3.3) that when the offdiagonal entries are nonpositive, then a diagonal entry in the aggregated matrix cannot be larger than the sum of the diagonal entries in the corresponding aggregate; hence

$$\widetilde{D}_{\{i,j\}}\leq D_{\{i,j\}}=P_f{}^T\,D_{G_i\cup G_j}P_f\,.$$

From this latter relation, using the same arguments as in the proof of inequality (39) in [29, Corollary 2.2], one can show the first of the following inequalities, the other ones being straightforward, noting that $\mathbf{1}_{G_i\cup G_j}=P_f\mathbf{1}_2$:

$$\begin{aligned}
&\sup_{\mathbf{v}\notin\mathcal{N}\left(\widetilde{A}^{(S)}_{\{i,j\}}\right)}\frac{\mathbf{v}\widetilde{D}_{\{i,j\}}\left(I-\mathbf{1}_2\left(\mathbf{1}_2^T\widetilde{D}_{\{i,j\}}\mathbf{1}_2\right)^{-1}\mathbf{1}_2^T\widetilde{D}_{\{i,j\}}\right)\mathbf{v}}{\mathbf{v}^T\widetilde{A}^{(S)}_{\{i,j\}}\mathbf{v}}\\
&\leq\sup_{\mathbf{v}\notin\mathcal{N}\left(\widetilde{A}^{(S)}_{\{i,j\}}\right)}\frac{\mathbf{v}D_{\{i,j\}}\left(I-\mathbf{1}_2\left(\mathbf{1}_2^TD_{\{i,j\}}\mathbf{1}_2\right)^{-1}\mathbf{1}_2^TD_{\{i,j\}}\right)\mathbf{v}}{\mathbf{v}^T\widetilde{A}^{(S)}_{\{i,j\}}\mathbf{v}}\,.\\
&=\sup_{\mathbf{v}\notin\mathcal{N}\left(P_f{}^TA^{(S)}_{G_i\cup G_j}P_f\right)}\frac{\mathbf{v}\,P_f{}^TD_{G_i\cup G_j}\left(I-\mathbf{1}_{G_i\cup G_j}\left(\mathbf{1}_{G_i\cup G_j}^TD_{G_i\cup G_j}\mathbf{1}_{G_i\cup G_j}\right)^{-1}\mathbf{1}_{G_i\cup G_j}^TD_{G_i\cup G_j}\right)P_f\mathbf{v}}{\mathbf{v}^TP_f{}^TA^{(S)}_{G_i\cup G_j}P_f\mathbf{v}}\\
&\leq\sup_{\mathbf{w}\notin\mathcal{N}\left(A^{(S)}_{G_i\cup G_j}\right)}\frac{\mathbf{w}^TD_{G_i\cup G_j}\left(I-\mathbf{1}_{G_i\cup G_j}\left(\mathbf{1}_{G_i\cup G_j}^TD_{G_i\cup G_j}\mathbf{1}_{G_i\cup G_j}\right)^{-1}\mathbf{1}_{G_i\cup G_j}^TD_{G_i\cup G_j}\right)\mathbf{w}}{\mathbf{w}^TA^{(S)}_{G_i\cup G_j}\mathbf{w}}\\
&=\omega\;\mu(G_i\cup G_j)\,.
\end{aligned}$$

The required result then follows from (A.1).

Acknowledgment. The referees are acknowledged for their useful suggestions.

REFERENCES

[1] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.
[2] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.
[3] D. Braess, *Towards algebraic multigrid for elliptic problems of second order*, Computing, 55 (1995), pp. 379–393.
[4] A. Brandt, S. F. McCormick, and J. W. Ruge, *Algebraic multigrid (AMG) for sparse matrix equations*, in Sparsity and Its Application, D. J. Evans, ed., Cambridge University Press, Cambridge, UK, 1984, pp. 257–284.
[5] 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 J. Sci. Comput., 22 (2000), pp. 1570–1592.
[6] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive smoothed aggregation (αSA) multigrid*, SIAM Rev., 47 (2005), pp. 317–346.
[7] M. Brezina, T. Manteuffel, S. McCormick, J. Ruge, and G. Sanders, *Towards adaptive smoothed aggregation (αSA) for nonsymmetric problems*, SIAM J. Sci. Comput., 32 (2010), pp. 14–39.
[8] V. E. Bulgakov, *Multi-level iterative technique and aggregation concept with semi-analytical preconditioning for solving boundary-value problems*, Comm. Numer. Methods Engrg., 9 (1993), pp. 649–657.
[9] E. Cuthill and J. McKee, *Reducing the bandwidth of sparse symmetric matrices*, in Proceedings of the 24th National Conference of the Association for Computing Machinery, Brandon Press, NJ, 1969, pp. 157–172.
[10] P. M. de Zeeuw, *Matrix-dependent prolongations and restrictions in a blackbox multigrid solver*, J. Comput. Appl. Math., 33 (1990), pp. 1–27.
[11] J. E. Dendy, *Black box multigrid for nonsymmetric problems*, Appl. Math. Comput., 13 (1983), pp. 261–283.
[12] S. C. Eisenstat, H. C. Elman, and M. H. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 20 (1983), pp. 345–357.
[13] H. Elman, D. Silvester, and A. Wathen, *Finite Elements and Fast Iterative Solvers*, Oxford University Press, Oxford, 2005.
[14] M. Emans, *Performance of parallel AMG-preconditioners in CFD-codes for weakly compressible flows*, Parallel Comput., 36 (2010), pp. 326–338.
[15] O. G. Ernst, *Residual-minimizing Krylov subspace methods for stabilized discretizations of convection-diffusion equations*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1079–1101.
[16] W. Hackbusch, *Multi-grid Methods and Applications*, Springer, Berlin, 1985.
[17] *The HSL Mathematical Software Library*, a collection of Fortran codes for large-scale scientific computation, 2011, http://www.hsl.rl.ac.uk.
[18] J. E. Jones and P. S. Vassilevski, *AMGe based on element agglomeration*, SIAM J. Sci. Comput., 23 (2001), pp. 109–133.
[19] H. Kim, J. Xu, and L. Zikatanov, *A multigrid method based on graph matching for convection-diffusion equations*, Numer. Linear Algebra Appl., 10 (2003), pp. 181–195.
[20] J. Liesen and Z. Strakoš, *GMRES convergence analysis for a convection-diffusion model problem*, SIAM J. Sci. Comput., 26 (2005), pp. 1989–2009.
[21] S. P. MacLachlan and C. W. Oosterlee, *Algebraic multigrid solvers for complex-valued matrices*, SIAM J. Sci. Comput., 30 (2008), pp. 1548–1571.
[22] *MUMPS software and documentation*, available online from http://graal.ens-lyon.fr/MUMPS/.
[23] A. C. Muresan and Y. Notay, *Analysis of aggregation-based multigrid*, SIAM J. Sci. Comput., 30 (2008), pp. 1082–1103.
[24] A. Napov and Y. Notay, *Algebraic analysis of aggregation-based multigrid*, Numer. Linear Algebra Appl., 18 (2011), pp. 539–564.
[25] A. Napov and Y. Notay, *An algebraic multigrid method with guaranteed convergence rate*, SIAM J. Sci. Comput., 34 (2012), pp. A1079–A1109.
[26] Y. Notay, *AGMG software and documentation*, available online from http://homepages.ulb.ac.be/~ynotay/AGMG.
[27] Y. Notay, *A robust algebraic multilevel preconditioner for non symmetric M-matrices*, Numer. Linear Algebra Appl., 7 (2000), pp. 243–267.

[28] Y. Notay, *An aggregation-based algebraic multigrid method*, Electron. Trans. Numer. Anal., 37 (2010), pp. 123–146.
[29] Y. Notay, *Algebraic analysis of two-grid methods: The nonsymmetric case*, Numer. Linear Algebra Appl., 17 (2010), pp. 73–97.
[30] Y. Notay and P. S. Vassilevski, *Recursive Krylov-based multigrid cycles*, Numer. Linear Algebra Appl., 15 (2008), pp. 473–487.
[31] M. A. Olshanskii and A. Reusken, *Convergence analysis of a multigrid method for a convection-dominated model problem*, SIAM J. Numer. Anal., 42 (2004), pp. 1261–1291.
[32] C. W. Oosterlee and T. Washio, *An evaluation of parallel multigrid as a solver and a preconditioner for singularly perturbed problems*, SIAM J. Sci. Comput., 19 (1998), pp. 87–110.
[33] C. W. Oosterlee and T. Washio, *Krylov subspace acceleration of nonlinear multigrid with application to recirculating flows*, SIAM J. Sci. Comput., 21 (2000), pp. 1670–1690.
[34] A. Reusken, *A multigrid method based on incomplete Gaussian elimination*, Numer. Linear Algebra Appl., 3 (1996), pp. 369–390.
[35] A. Reusken, *Convergence analysis of a multigrid method for convection-diffusion equations*, Numer. Math., 91 (2002), pp. 323–349.
[36] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, PA, 2003.
[37] M. Sala and R. S. Tuminaro, *A new Petrov–Galerkin smoothed aggregation preconditioner for nonsymmetric linear systems*, SIAM J. Sci. Comput., 31 (2008), pp. 143–166.
[38] K. Stüben, *An introduction to algebraic multigrid*, in Multigrid, U. Trottenberg, C. W. Oosterlee, and A. Schüller, eds., Academic Press, London, 2001, pp. 413–532, Appendix A.
[39] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra*, Princeton University Press, Princeton, NJ, Oxford, 2005.
[40] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.
[41] H. A. van der Vorst and C. Vuik, *GMRESR: A family of nested GMRES methods*, Numer. Linear Algebra Appl., 1 (1994), pp. 369–386.
[42] P. Vaněk, J. Mandel, and M. Brezina, *Algebraic multigrid based on smoothed aggregation for second and fourth order elliptic problems*, Computing, 56 (1996), pp. 179–196.
[43] P. S. Vassilevski, *Multilevel Block Factorization Preconditioners*, Springer, New York, 2008.
[44] R. Wienands and C. W. Oosterlee, *On three-grid Fourier analysis for multigrid*, SIAM J. Sci. Comput., 23 (2001), pp. 651–671.
[45] C.-T. Wu and H. C. Elman, *Analysis and comparison of geometric and algebraic multigrid for convection-diffusion equations*, SIAM J. Sci. Comput., 28 (2006), pp. 2208–2228.
[46] I. Yavneh, *Coarse-grid correction for nonelliptic and singular perturbation problems*, SIAM J. Sci. Comput., 19 (1998), pp. 1682–1699.
[47] I. Yavneh, C. H. Venner, and A. Brandt, *Fast multigrid solution of the advection problem with closed characteristics*, SIAM J. Sci. Comput., 19 (1998), pp. 111–125.