SIAM J. SCI. COMPUT.
Vol. 32, No. 2, pp. 544–562

ALGEBRAIC MULTIGRID FOR MARKOV CHAINS*

H. DE STERCK†, T. A. MANTEUFFEL‡, S. F. MCCORMICK‡, K. MILLER†, J. RUGE‡, AND G. SANDERS‡

Abstract. An algebraic multigrid (AMG) method is presented for the calculation of the stationary probability vector of an irreducible Markov chain. The method is based on standard AMG for nonsingular linear systems, but in a multiplicative, adaptive setting. A modified AMG interpolation formula is proposed that produces a nonnegative interpolation operator with unit row sums. We show how the adoption of a previously described lumping technique maintains the irreducible singular M-matrix character of the coarse-level operators on all levels. Together, these properties are sufficient to guarantee the well-posedness of the algorithm. Numerical results show how it leads to nearly optimal multigrid efficiency for a representative set of test problems.

Key words. multilevel method, Markov chain, stationary probability vector, algebraic multigrid

AMS subject classifications. 65C40, 60J22, 65F10, 65F15

DOI. 10.1137/090753589

1. Introduction.

This paper describes an algebraic multigrid (AMG) method for computing the stationary probability vector of large, sparse, irreducible Markov transition matrices.

While multigrid methods of aggregation type have been considered before for Markov chains [13, 10, 9], our present approach is based on standard AMG for nonsingular linear systems, but in a multiplicative, adaptive setting. The current method is, in fact, an extension to nonvariational coarsening of the variational adaptive AMG scheme originally developed in the early stages of the AMG project by Brandt, McCormick, and Ruge [3] (described earlier in [18]). One of the features of the earlier approach is that it constructed interpolation to exactly match the minimal eigenvector of the matrix. A closely related technique called the exact interpolation scheme (EIS) was proposed by Brandt and Ron [4]. The EIS has been applied to eigenvalue problems, for example, as a multigrid solver for one-dimensional Helmholtz eigenvalue problems [14]. Moreover, the current method also incorporates some aspects of early work on aggregation multigrid for Markov chains. In particular, it uses a multiplicative correction form of the coarse-grid correction process that is similar to the two-level aggregated equations proposed in [21], and its framework is similar to the two-level iterative aggregation/disaggregation method for Markov chains pioneered in [24] and since used and analyzed extensively (see [22] and [9] for references). The so-called square-and-stretch multigrid algorithm by Treister and Yavneh [25] is a recent aggregation-based multigrid method for Markov chains that is also related to our approach.

Starting from the classical definition of AMG interpolation described in [6], we propose a modified interpolation formula that produces a nonnegative interpolation

*Received by the editors March 22, 2009; accepted for publication (in revised form) November 17, 2009; published electronically February 17, 2010.
http://www.siam.org/journals/sisc/32-2/75358.html
†Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada (hdesterck@uwaterloo.ca, k7miller@uwaterloo.ca).
‡Department of Applied Mathematics, University of Colorado at Boulder, Boulder, Colorado 80309-0526 (tmanteuf@colorado.edu, stevem@colorado.edu, jruge@colorado.edu, sandersg@colorado.edu).

operator with unit row sums. Furthermore, it is shown how the adoption of a lumping technique, which was recently employed in an aggregation-based method for Markov chains [9], maintains the irreducible singular M-matrix character of the coarse-level operators on all levels. Together, these properties are sufficient to prove the well-posedness of our algorithm. We show numerically that the resulting lumped AMG method for Markov chains (MCAMG) leads to nearly optimal multigrid efficiency for a representative set of test problems for which traditional iterative methods are slow to converge. The aim here is to use a sequence of successively coarser versions of the original problem to remedy the slow convergence that plagues traditional one-level iterative methods, like the power method, when the subdominant eigenvalue satisfies $|\lambda_2| \approx 1$ [17].

The set of test problems is composed of two classes, those for which the probability transition matrix is similar to a symmetric matrix and for which the eigenvalue spectrum is thus real (their solution can be determined by an inexpensive, local calculation) and more challenging problems with nonsymmetric sparsity structure for which the spectrum is complex. Note that the use of AMG has already been explored for Markov chain problems [27] in the context of additive AMG used as a preconditioner for GMRES. Our formulation, however, is multiplicative, and near-optimal results are obtained without GMRES acceleration. Our formulation is also different in that it is related to adaptive AMG [5].

Large sparse Markov chains are of interest in a wide range of applications, including information retrieval and web ranking, performance modeling of computer and communication systems, dependability and security analysis, and analysis of biological systems [22]. Multilevel solvers for Markov problems with improved efficiency thus promise to have significant impact in many disciplines.

This paper is organized as follows. We begin with the problem formulation and definitions in section 2, and in section 3, we provide some essential theoretical background concerning the class of (irreducible) singular M-matrices. As such, it is our aim that this paper be self-contained. In section 4, we recall the general framework of AMG methods for nonsingular systems and discuss where our method deviates from the classical approach. We present the MCAMG V-cycle algorithm in section 4.6, and in section 4.7, we rigorously prove the well-posedness of MCAMG. Numerical convergence tests are presented for a set of representative test problems in section 5. Conclusions and future work are discussed in section 6.

2. Mathematical formulation. The problem of finding the stationary probability vector of a Markov chain can be stated as follows. Given a column-stochastic transition-probability matrix, $B \in \mathbb{R}^{n\times n}$, i.e., $0 \le b_{ij} \le 1$ and

$$\mathbf{1}^T B = \mathbf{1}^T, \tag{2.1}$$

we seek the vector $\mathbf{x} \in \mathbb{R}^n$ such that

$$B\,\mathbf{x} = \mathbf{x}, \qquad x_i \ge 0\ \forall\, i, \qquad \|\mathbf{x}\|_1 = 1. \tag{2.2}$$

Here, $\mathbf{1}$ is the column vector of all ones. It can be shown that if B is *irreducible*, then there exists a unique solution to (2.2) with strictly positive components. This is a consequence of the Perron–Frobenius theorem for nonnegative matrices [12]. In what follows, we consider the case where B is irreducible, a concept that we now formally define.

Definition 2.1 (directed walk and directed path). *For nodes u and v in a directed graph, $D = (\mathcal{N}, \mathcal{A})$, with node set $\mathcal{N}$ and arc set $\mathcal{A}$, a u-v walk in D is a*

finite sequence of nodes $u = v_0, v_1, \ldots, v_{k-1}, v_k = v$, beginning at u and ending at v, such that $(v_{i-1}, v_i) \in \mathcal{A}$ for $i = 1, \ldots, k$. A directed u-v path is a directed u-v walk in which no node is repeated.

DEFINITION 2.2 (directed graph of a matrix). *The directed graph of $A \in \mathbb{R}^{n\times n}$, denoted by $\Gamma(A)$, is the directed graph on n nodes $v_1, \ldots, v_n$ such that an arc exists from v_i to v_j if and only if $a_{ji} \neq 0$.*

DEFINITION 2.3 (irreducible matrix). *Matrix $A \in \mathbb{R}^{n\times n}$ is called irreducible if and only if there exists a directed path from v_i to v_j for any two distinct nodes $v_i, v_j \in \mathcal{N}(\Gamma(A))$.*

3. Singular M-matrices. Following the approach outlined in [27], we can, equivalently, restate the problem of finding the stationary probability vector as solving for a strictly positive vector of unit length in the null space of an irreducible singular M-matrix. Mathematically, we seek the vector $\mathbf{x} \in \mathbb{R}^n$ such that

$$A\,\mathbf{x} = \mathbf{0}, \qquad x_i > 0 \;\forall\, i, \qquad \|\mathbf{x}\|_1 = 1, \tag{3.1}$$

where $A := I - B$. Here, A is an irreducible singular M-matrix, and $\mathbf{1}^T A = \mathbf{0}$.

We now define singular M-matrices, show that A belongs to this class, and state a number of properties shared by all singular M-matrices. These properties, together with irreducibility, provide a solid theoretical foundation with which we can prove the well-posedness of our algorithm in the the sense that, given an iterate that is strictly positive, the algorithm gives a proper definition for the next iterate. Let $\rho(B)$ be the spectral radius of B. Then, we have the following definition.

DEFINITION 3.1 (singular M-matrix). *$A \in \mathbb{R}^{n\times n}$ is a singular M-matrix if and only if there exists $B \in \mathbb{R}^{n\times n}$, with $b_{ij} \geq 0 \;\forall\; i, j$, such that $A = \rho(B)I - B$.*

The justification that $A = I - B$ is a singular M-matrix now follows readily from Definition 3.1 and from the fact that $\rho(B) = 1$ for any column-stochastic matrix B. Furthermore, it is easy to see that if B is irreducible, then A must also be irreducible, since subtracting B from I cannot zero out any off-diagonal elements of B (refer to Definition 2.3). The following properties of singular M-matrices are used throughout this paper and can be found in [2, 9, 27].

THEOREM 3.2 (some properties of singular M-matrices).

(1) *Irreducible singular M-matrices yield a unique solution to $A\,\mathbf{x} = \mathbf{0}$, up to scaling, which can be chosen such that all components of $\mathbf{x}$ are strictly positive.*

(2) *Irreducible singular M-matrices have nonpositive off-diagonal elements and strictly positive diagonal elements ($n > 1$).*

(3) *If A has a strictly positive vector in its left or right null space and its off-diagonal elements are nonpositive, then A is a singular M-matrix.*

(4) *If A is an irreducible singular M-matrix, then each of its principal submatrices, other than A itself, is a nonsingular M-matrix.*

As we shall see, property (2) allows us to construct an interpolation operator with nonnegative entries and unit row sums. These properties are also used below to prove that the coarse operators of our AMG method are irreducible singular M-matrices on all levels.

4. AMG for Markov chains. In this section, we recall the principal features of the classical AMG V-cycle [3, 6, 26] on which our method is based. We discuss how our approach for Markov chains deviates from the classical approach for nonsingular linear systems and how it incorporates aspects of recent work on aggregation multigrid for Markov chains [9]. We conclude this section by describing our V-cycle algorithm and proving well-posedness.

4.1. Multiplicative AMG method and coarsening. One major difference between our approach for Markov chains and that of classical AMG for nonsingular linear systems is the use of multiplicative error, $\mathbf{e}_i$, defined by $\mathbf{x} = \operatorname{diag}(\mathbf{x}_i)\,\mathbf{e}_i$, where $\mathbf{x}$ is the exact solution of (3.1) and $\mathbf{x}_i$ is the ith iterate. As we see below, the iterates produced by our algorithm have strictly positive components. Equation (3.1) can then be rewritten as

$$A \operatorname{diag}(\mathbf{x}_i)\,\mathbf{e}_i = \mathbf{0}. \tag{4.1}$$

We observe that, at convergence, $\mathbf{x}_i = \mathbf{x}$ and hence, $\mathbf{e}_i = \mathbf{1}$. This motivates the following definition of the multiplicative coarse-level error correction:

$$\mathbf{x}_{i+1} = \operatorname{diag}(\mathbf{x}_i)\,P\,\mathbf{e}_c, \tag{4.2}$$

where P is the interpolation operator (see section 4.2) and $\mathbf{e}_c$ is the error approximation on the coarse level. It is easy to see that (4.2) is the natural extension of the additive coarse-level error correction to the multiplicative case.

Now consider the scaled fine-level operator given by $\bar{A} := A \operatorname{diag}(\mathbf{x}_i)$. By Theorem 3.2(3), $\bar{A}$ is also an irreducible singular M-matrix. We can rewrite (4.1) in terms of $\bar{A}$, which results in the following fine-level error equation:

$$\bar{A}\,\mathbf{e}_i = \mathbf{0}. \tag{4.3}$$

To seek a coarse representation of (4.3), we first perform the two-pass AMG coarsening routine described in [6], which determines the set of points on the coarse level. The coarsening routine partitions the points on the current level into a set of coarse points, C, and a set of fine points, F. Its goal is to choose C large enough so that interpolation is accurate, but not so large that the work done in a V-cycle is prohibitive. The first pass proceeds by constructing a preliminary partition of coarse points (C-points) and fine points (F-points) such that for any F-point i there exists at least one C-point j that *strongly influences* i (see (4.4)). As well, the first pass attempts to satisfy the condition that C be a maximal subset with the property that no C-point strongly influences another. The second pass refines the initial partition by changing some of the F-points into C-points. New C-points are chosen such that any point j that strongly influences F-point i either belongs to C_i or is strongly influenced by at least one point in C_i. Here, C_i is the set of C-points that strongly influence point i. Performing the second pass improves the accuracy that can be obtained by interpolation and is necessary to ensure the well-posedness of the interpolation formula presented in section 4.2.

We base *strength of connection* on the scaled operator, $\bar{A}$, and not on A itself (see also [10, 9]). On the finest level, we can offer a probabilistic justification for this: for $i \neq j$, $-a_{ij}$ is the probability of moving to state i, given that the chain is currently in state j, i.e., $-a_{ij}$ is a conditional probability. In order for conditional probabilities to be meaningful, they must be interpreted in terms of the underlying probability distribution on which they are conditioned. For example, in terms of a Markov chain, X_n,

$$\mathbb{P}(X_{k+1} = i) = \sum_{j=1}^{n} \mathbb{P}(X_{k+1} = i \,|\, X_k = j)\mathbb{P}(X_k = j).$$

Thus, the transition probabilities, $\mathbb{P}(X_{k+1} = i \,|\, X_k = j)$, tell us little about $\mathbb{P}(X_{k+1} = i)$, unless we know the $\mathbb{P}(X_k = j)$. This is an indication that basing strength of

connection on A may not be meaningful and may result in poor convergence of our method. Instead, by basing strength of connection on $\bar{A}$, we make use of the most current information available on the underlying distribution and consider the joint probability of being in state i.

Strength of connection for our multigrid method is defined as follows: given a threshold value, $\theta \in [0,1]$, point j strongly influences point i if

$$-\bar{a}_{ij} \geq \theta \max_{k \neq i}\{-\bar{a}_{ik}\}. \tag{4.4}$$

In this paper, unless stated otherwise, we use strength threshold $\theta = 0.25$. At convergence, $\mathbf{1}$ lies in the null space of $\bar{A}$, so standard AMG coarsening and interpolation approaches work well. The coarse-level version of (4.3) is given by

$$R\,\bar{A}\,P\,\mathbf{e}_c = \mathbf{0} \quad \text{or} \quad \bar{A}_c\,\mathbf{e}_c = \mathbf{0}, \tag{4.5}$$

with $\bar{A}_c := R\,\bar{A}\,P$ and the restriction operator defined by the variational property, $R = P^T$.

The attentive reader may question the well-posedness of the coarse-level equation: coarse-level operator $\bar{A}_c$ may not be an irreducible singular M-matrix, which implies that the coarse-level equation may not have a unique, strictly positive solution (up to scaling). As we show in section 4.3, this problem is remedied by applying a *lumping method* [9] to the coarse-level operator whereby we obtain the *lumped* coarse-level operator, $\hat{A}_c$. We prove below that $\hat{A}_c$ is an irreducible singular M-matrix and that the exact solution, $\mathbf{x}$, is a fixed point of the V-cycle with the lumped coarse-level error equation, $\hat{A}_c\,\mathbf{e}_c = \mathbf{0}$.

We conclude this section by stating two identities for the unlumped $\bar{A}_c$ which are used in section 4.3. Let the coarse-level column vector of all ones be denoted by $\mathbf{1}_c$. We choose P such that $P\,\mathbf{1}_c = \mathbf{1}$ (see below), which implies that

$$\mathbf{1}_c^T\,\bar{A}_c = \mathbf{0} \quad \forall\ \mathbf{x}_i, \tag{4.6}$$

$$\bar{A}_c\,\mathbf{1}_c = \mathbf{0} \quad \text{for } \mathbf{x}_i = \mathbf{x}. \tag{4.7}$$

4.2. Interpolation. The interpolation operator, P, transfers information from coarse to fine levels. It is constructed in such a way that it accurately represents fine-level *algebraically smooth components*, by which we mean components whose error is not effectively reduced by relaxation. In the AMG method, interpolation is accomplished by approximating the error at each fine-level point (F-point) as a weighted sum of the error at coarse-level points (C-points). In what follows, we recall the definition of the AMG interpolation operator from [6] and explain how the formula for the interpolation weights is modified to obtain the properties for P that are desirable for Markov chain problems.

Suppose we have already performed coarsening on the current set of fine-level points $H = \{1, \ldots, n\}$ and have thus partitioned H into a set of coarse (C) and fine (F) points. Without restricting generality, we assume that H is ordered so that

$$H = \{\underbrace{1, \ldots, n_c}_{C}, \underbrace{n_c + 1, \ldots, n_c + n_f}_{F}\},$$

where $|C| = n_c$, $|F| = n_f$, and $n = n_c + n_f$. Then, for any point $i \in H = C \cup F$, we require that

$$(P\,\mathbf{e}_c)_i = \begin{cases} (\mathbf{e}_c)_i & \text{if } i \in C, \\ \sum_{j \in C_i} w_{ij} (\mathbf{e}_c)_j & \text{if } i \in F, \end{cases} \tag{4.8}$$

where $\mathbf{e}_c$ is the coarse-level error approximation, the w_{ij}s are the interpolation weights, and C_i is the set of C-points that strongly influence point i according to (4.4). Observe that, for any $i \in C$, row i of P is all zeros except for the entry corresponding to C-point i, which equals 1. A classical formula for AMG interpolation weights is derived in [6]:

$$w_{ij} = -\frac{\bar{a}_{ij} + \sum_{m \in D_i^s} \left(\frac{\bar{a}_{im}\bar{a}_{mj}}{\sum_{k \in C_i} \bar{a}_{mk}} \right)}{\bar{a}_{ii} + \sum_{r \in D_i^w} \bar{a}_{ir}}, \tag{4.9}$$

where $C_i \cup D_i^s \cup D_i^w = N_i$, the directed neighborhood of point i, which is the set of all points $k \neq i$ such that $\bar{a}_{ik} \neq 0$. Here, D_i^s is the set of F-points that strongly influence i, and D_i^w is the set of points in the neighborhood N_i that weakly influence i. Note that D_i^w may contain both F-points and C-points.

However, we desire an interpolation operator whose rows sum to unity, that is, we desire a P such that $P\,\mathbf{1}_c = \mathbf{1}$. This is a necessary condition to establish identities (4.6) and (4.7), which are essential for the well-posedness of our method (see below). To ensure that P enjoys this property, we simply rescale the w_{ij}s of (4.9) such that they sum to one, with the rescaled weights $\bar{w}_{ij}$ given by

$$\bar{w}_{ij} = \frac{\bar{a}_{ij} + \sum_{m \in D_i^s} \left(\frac{\bar{a}_{im}\bar{a}_{mj}}{\sum_{k \in C_i} \bar{a}_{mk}} \right)}{\sum_{p \in C_i} \bar{a}_{ip} + \sum_{r \in D_i^s} \bar{a}_{ir}}. \tag{4.10}$$

Note that, in the case of singular M-matrices, the classical interpolation formula, (4.9), can lead to negative weights and division by zero. The rescaled formula, (4.10), does not suffer from these deficiencies. Indeed, under the premise that $\bar{A}$ is an irreducible singular M-matrix, Theorem 3.2(2) ensures that all matrix elements used in (4.10) are nonpositive. Since the two-pass AMG coarsening routine ensures that $C_i \neq \emptyset$, it follows that the denominator in (4.10) is nonzero. Furthermore, together with the fact that C_i, D_i^s, and D_i^w do not have points in common (which precludes diagonal elements $\bar{a}_{mm}$ from occurring in (4.10)), we find that $\bar{w}_{ij} > 0 \;\forall\; i \in F$ and $j \in C_i$. Thus, the redefined interpolation operator has nonnegative entries and unit row sums. Note that it is important to perform both passes of the coarsening routine, since this ensures that $\sum_{k \in C_i} \bar{a}_{mk} \neq 0$ for any $i \in F$ and $m \in D_i^s$, which is required for the w_{ij}s to be well-defined. It is the second pass of the coarsening routine that ensures that every point in D_i^s strongly depends on at least one point in C_i.

4.3. Lumping. As we mentioned at the close of section 4.1, the coarse-level operator, $\bar{A}_c$, may not be an irreducible singular M-matrix. To illustrate this point, let matrices D, L, and U be such that $\bar{A} = D - (L + U)$, where D is diagonal, L is strictly lower triangular, and U is strictly upper triangular. Then

$$\bar{A}_c = P^T \bar{A}\, P = P^T D\, P - P^T (L + U)\, P = S - G, \tag{4.11}$$

where both $S = P^T D\, P$ and $G = P^T (L + U)\, P$ are nonnegative matrices because $\bar{A}$ is a singular M-matrix and P has nonnegative entries. $P^T D\, P$ is generally not diagonal, so $\bar{A}_c$ may have positive off-diagonal entries, meaning it may not be a singular M-matrix. Furthermore, $\bar{A}_c$ may lose irreducibility due to new zero entries being introduced. To rectify this problem, we adopt the lumping method described in

[9] for smoothed aggregation multigrid methods for Markov chains. In what follows, we illustrate why lumping is necessary for our algorithm and then provide an overview of the lumping procedure.

If we do not perform lumping, then the irreducible singular M-matrix sign structure of the coarse-level operators is not guaranteed. This leads in many cases to erratic convergence of our method and, in some cases, may even result in stalling or divergence. Indeed, many things can go wrong in the algorithm when coarse-level operators are not singular M-matrices. For example, incorrect signs in coarse-level operators may produce negative interpolation weights. Coarse-grid correction may then lead to the generation of multiplicative error vectors with some vanishing or negative components. Components with incorrect signs may also be generated after relaxation (see section 4.4). These pathological error vectors propagate incorrect signs upward in the cycle via coarse-grid correction and downward via column-scaled operators that may have entire columns that vanish or have incorrect signs. A unique, strictly positive solution is also no longer guaranteed for the coarse-level direct solve (see section 4.4): it may become singular itself or may produce a solution with incorrect signs that propagate upwards in the cycle. We have found that, in some cases, lumping is not necessary, but unfortunately we know of no easy way to determine a priori whether or not a particular problem requires lumping. In our experience, problem matrices that are similar to symmetric matrices (and thus have real eigenvalue spectra) often do not require lumping, except sometimes in the first few cycles. In fact, for certain subclasses of symmetric M-matrices (in particular, weakly diagonally dominant matrices) and under some fairly restrictive assumptions on the coarsening and interpolation routines, it can be shown that the AMG coarse-level Galerkin operator will also be an M-matrix [19], in which case we expect lumping to be unnecessary, at least close to convergence. Even for cases like this, however, if lumping is not performed in the early cycles, we have found that convergence may become erratic (especially for large problems). Problems with less symmetry typically require lumping in all cycles. In summary, we have found that lumping is required for the algorithm to be robust.

We thus consider a modified version, $\hat{S}$, of S, obtained by *lumping* parts of S to the diagonal (explained below), resulting in the modified coarse-level operator

$$\hat{A}_c = \hat{S} - G. \tag{4.12}$$

Our goal is to modify S in such a way that $\hat{A}_c$ has nonpositive off-diagonal elements and retains nonzero off-diagonal elements where G has them (to guarantee irreducibility).

Define an *offending index pair* as a tuple (i, j) such that $i \neq j$, and $s_{ij} \neq 0$, and $(\bar{A}_c)_{ij} \geq 0$. It is for these indices that lumping is performed. Let (i, j) be an offending index pair. To correct the sign in $\bar{A}_c$ at location (i, j), let

$$S_{\{i,j\}} = \begin{array}{c} \\ i \\ \\ j \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccc} & & i & & j & \end{array} \\ \begin{bmatrix} \ddots & \vdots & & \vdots & \\ \cdots & \beta_{\{i,j\}} & \cdots & -\beta_{\{i,j\}} & \cdots \\ & \vdots & & \vdots & \\ \cdots & -\beta_{\{i,j\}} & \cdots & \beta_{\{i,j\}} & \cdots \\ & \vdots & & \vdots & \end{bmatrix} \end{array}, \tag{4.13}$$

where $\beta_{\{i,j\}} > 0$ and the other elements are zero. We add $S_{\{i,j\}}$ to S, which corresponds to lumping parts of S to the diagonal in the sense that $\beta_{\{i,j\}}$ is removed

from off-diagonal elements s_{ij} and s_{ji} and added to diagonal elements s_{ii} and s_{jj}. We choose $\beta_{\{i,j\}}$ so that

$$\begin{aligned} s_{ij} - g_{ij} - \beta_{\{i,j\}} &< 0, \\ s_{ji} - g_{ji} - \beta_{\{i,j\}} &< 0, \end{aligned} \tag{4.14}$$

resulting in strictly negative off-diagonal elements in $\hat{A}_c$ at locations (i,j) and (j,i). Note that $\beta_{\{i,j\}}$ is chosen such that adding $S_{\{i,j\}}$ for correcting the sign at location (i,j) also corrects the sign at location (j,i) if necessary. This means that if both (i,j) and (j,i) are offending index pairs, then only one matrix $S_{\{i,j\}}$ has to be added to S. In our implementation, $\beta_{\{i,j\}} = \max\left(\beta^{(1)}_{\{i,j\}}, \beta^{(2)}_{\{i,j\}}\right)$, with

$$\begin{aligned} s_{ij} - g_{ij} - \beta^{(1)}_{\{i,j\}} &= -\eta\, g_{ij}, \\ s_{ji} - g_{ji} - \beta^{(2)}_{\{i,j\}} &= -\eta\, g_{ji}, \end{aligned} \tag{4.15}$$

and η a fixed parameter $\in (0,1]$. It is important to note that while lumping may introduce new nonzero entries into $\hat{A}_c$, it cannot create a zero entry in $\hat{A}_c$ where G is nonzero. Finally, we experimentally observed that we should lump as little as possible, so a small η should be chosen [9]. In practice, $\eta = 0.01$ seems to be a good choice.

Finally, symmetric matrices of the form in (4.13) are used to modify S so that column sums and row sums of $\bar{A}_c$ are conserved. This ensures that properties (4.6) and (4.7) are retained after lumping: $\hat{A}_c$ has $\mathbf{1}$ as a left-kernel vector on all levels and, at convergence, has $\mathbf{1}$ as a right-kernel vector. Indeed, since $\hat{S} - S = \sum S_{\{i,j\}}$, where the sum is over all matrices $S_{\{i,j\}}$ added to S, it follows that

$$\mathbf{1}_c^T \hat{A}_c = \mathbf{1}_c^T \bar{A}_c + \mathbf{1}_c^T(\hat{S} - S) = \mathbf{1}_c^T \bar{A}_c = \mathbf{0} \quad \forall\ \mathbf{x}_i, \tag{4.16}$$

$$\hat{A}_c\, \mathbf{1}_c = \bar{A}_c\, \mathbf{1}_c + (\hat{S} - S)\, \mathbf{1}_c = \bar{A}_c\, \mathbf{1}_c = \mathbf{0} \quad \text{for } \mathbf{x}_i = \mathbf{x}. \tag{4.17}$$

4.4. Relaxation and coarsest level direct solve. This paper uses weighted Jacobi for all relaxation operations. Decomposing matrix A into its diagonal and negative strictly upper- and lower-triangular parts, $A = D - (L+U)$, weighted Jacobi for solving $A\,\mathbf{x} = \mathbf{0}$ is given by

$$\mathbf{x}^{(k+1)} = (1-\omega)\mathbf{x}^{(k)} + \omega D^{-1}(L+U)\,\mathbf{x}^{(k)}, \tag{4.18}$$

where $\omega \in (0,1)$ is a fixed weight parameter. We observe that if A is an irreducible singular M-matrix, then Theorem 3.2 confirms that D^{-1} exists and that $D^{-1}(L+U)$ has nonnegative entries. Thus, $\mathbf{x}^{(k+1)}$ has strictly positive entries if $\mathbf{x}^{(k)}$ has strictly positive entries and $\omega \in (0,1)$. Since we can normalize the result after relaxation, the constraint that $\mathbf{x}$ be a probability vector is easily obtained.

At the coarsest level, our goal is to perform a direct solve to obtain a nontrivial solution of $\hat{A}_c\, \mathbf{e}_c = \mathbf{0}$, where $\hat{A}_c$ is an irreducible singular M-matrix obtained by lumping. However, since $\hat{A}_c$ is singular, we cannot use Gaussian elimination. Instead, we show that the solution can be obtained by solving an equivalent nonsingular problem, to which Gaussian elimination can then be applied.

By Theorem 3.2(1), there exists a unique right-kernel solution, $\mathbf{e}_c$ (up to scaling), whose components are positive. Without loss of generality, we can scale $\mathbf{e}_c$ to $\bar{\mathbf{e}}_c$, to have its first component equal to 1. Thus, it is clear that

$$\begin{bmatrix} \mathbf{z}^T \\ \hat{A}_c \end{bmatrix} \bar{\mathbf{e}}_c = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix}, \tag{4.19}$$

where $\mathbf{z} = (1, 0, \ldots, 0)^T$. Dropping the second equation in the overdetermined system above, we see that $\bar{\mathbf{e}}_c$ also solves the resulting square system,

$$K\,\bar{\mathbf{e}}_c = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{a} & \tilde{A} \end{bmatrix} \bar{\mathbf{e}}_c = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} := \mathbf{z}, \tag{4.20}$$

where $\tilde{A}$ is a principal submatrix of $\hat{A}_c$ and $\mathbf{a}$ is the vector of the off-diagonal entries in the first column of $\hat{A}_c$. Since $\tilde{A}$ is a nonsingular M-matrix (see Theorem 3.2(4)), $\det(\tilde{A}) \neq 0$, which implies that $\det(K) = 1 \cdot \det(\tilde{A}) \neq 0$. Thus, $\bar{\mathbf{e}}_c$ is also the unique solution of (4.20).

We conclude that, in order to obtain a nontrivial solution to $\hat{A}_c\,\mathbf{e}_c = \mathbf{0}$, it is sufficient to solve the invertible system $K\,\bar{\mathbf{e}}_c = \mathbf{z}$. In exact arithmetic, we are guaranteed that the solution vector has positive components. For small problems on the coarsest level, we perform a direct solve via Gaussian elimination, and, for each of our test cases, it was verified numerically that $\bar{\mathbf{e}}_c$ has strictly positive components. However, for larger problems, $\bar{\mathbf{e}}_c$ may have nonpositive components due to rounding error incurred during Gaussian elimination. For example, this may be an issue if K is very ill-conditioned. In our implementation, we do not consider large problems on the coarsest level. However, if nonpositivity was encountered, one possibility would be to replace Gaussian elimination by a sufficient number of weighted Jacobi relaxations.

4.5. MCAMG V-cycle algorithm. Now that our method has been described, we state our V-cycle algorithm for Markov chains:

ALGORITHM 1 MCAMG(A, $\mathbf{x}$, ν_1, ν_2), AMG FOR MARKOV CHAINS (V-CYCLE).

if *not at the coarsest level,* **then**
  $\mathbf{x} \leftarrow \text{Relax}(A, \mathbf{x})$ ν_1 times
  $\bar{A} \leftarrow A\,\text{diag}(\mathbf{x})$
  Compute the set of coarse-level points C
  Construct the interpolation operator P
  Construct the coarse-level operator $\bar{A}_c \leftarrow P^T \bar{A}\,P$
  Obtain the lumped coarse-level operator $\hat{A}_c \leftarrow \text{Lump}(\bar{A}_c, \eta)$
  $\mathbf{e}_c \leftarrow \text{MCAMG}(\hat{A}_c, \mathbf{1}_c, \nu_1, \nu_2)$ `/* coarse-level solve */`
  $\mathbf{x} \leftarrow \text{diag}(\mathbf{x})\,P\,\mathbf{e}_c$ `/* coarse-level correction */`
  $\mathbf{x} \leftarrow \text{Relax}(A, \mathbf{x})$ ν_2 times
else
  $\mathbf{x} \leftarrow$ direct solve of $K\,\mathbf{x} = \mathbf{z}$ `/* see section 4.4 */`.
end

Note that the set of coarse-level points, C, and the interpolation operator, P, are recalculated for each V-cycle on each level. In principle, however, the sets of coarse-level points and the interpolation operators can be "frozen" after a few cycles to reduce the amount of work, but this is not done for the results presented in this paper.

4.6. Well-posedness of MCAMG. We recall that we require well-posedness of this algorithm in the sense that, given an iterate that is strictly positive, the algorithm gives a proper definition for the next iterate. We begin by proving the following proposition, which is the key result necessary to prove irreducibility of $\hat{A}_c$.

PROPOSITION 4.1 (irreducibility of G). *If $\bar{A} = D - (L + U)$ is an irreducible singular M-matrix, then $G = P^T(L + U)\,P$ is irreducible.*

Proof. We need to show that, for any C-points with coarse-level labels I and J, there exists a directed path from node I to node J in the directed graph of G. First, observe that if $\bar{A}$ is irreducible, then $(L+U)$ is irreducible, since diagonal entries do not matter for irreducibility. Assume that $(L+U)_{kl} \neq 0$ for some fine-level labels k and l, and let I be any C-point that interpolates to l, that is, $p_{lI} \neq 0$. Similarly, let J be any C-point that interpolates to k, that is, $p_{kJ} \neq 0$. In section 4.2, we showed that every row of P contains at least one nonzero element, hence, indices I and J exist. Now,

$$g_{IJ} = \mathbf{p}_I^T (L+U)\, \mathbf{p}_J,$$

where $\mathbf{p}_I$ denotes column I of P and $\mathbf{p}_J$ denotes column J of P. Since both $(\mathbf{p}_I)_l$ and $(\mathbf{p}_J)_k$ are nonzero and $(L+U)_{kl} \neq 0$, it follows by the nonnegativity of P and $(L+U)$ that $g_{IJ} \neq 0$. Thus, for any fine-level points l and k such that there exists an arc from node l to node k in $\Gamma(\bar{A})$, there must also exist coarse-level points I and J such that there is an arc from node I to node J in $\Gamma(G)$.

Now, let I and J be any distinct C-points. Furthermore, let i and j be the fine-level labels of I and J, respectively. By the irreducibility of $(L+U)$, there exists a directed path of distinct fine-level points from node i to node j. Denote this path by

$$i = v_0, v_1, \ldots, v_{k-1}, v_k = j,$$

where the nodes $v_0, \ldots, v_k$ are fine-level points. By the result above, there must exist coarse-level points $V_0, \ldots, V_k$ that form the directed walk (see Definition 2.1)

$$V_0, V_1, \ldots, V_{k-1}, V_k$$

in $\Gamma(G)$. However, any directed U-V walk contains a directed U-V path [7]. Thus, we can find a directed path in $\Gamma(G)$ that begins at V_0 and ends at V_k. Recall that C-points V_0 and V_k were chosen such that they interpolate to i and j, respectively. Now, since the only point that interpolates to a given C-point is the point itself (by the definition of P), it follows that $V_0 = I$ and $V_k = J$. Therefore, there exists a directed path from node I to node J in the directed graph of G. Since I and J were arbitrary, G is irreducible. ∎

Well-posedness of the algorithm now follows from the first of the following two theorems; the second theorem is a requirement for convergence of the method.

THEOREM 4.2 (singular M-matrix property of lumped coarse-level operator). *$\hat{A}_c$ is an irreducible singular M-matrix on all coarse levels and, thus, has a unique right-kernel vector with positive components (up to scaling) on all levels.*

Proof. Assume that $\bar{A}$ is an irreducible singular M-matrix, and let $\bar{A} = D-(L+U)$. By Proposition 4.1, matrix $G = P^T(L+U)\,P$ is irreducible. Lumping ensures that $\hat{A}_c$ has nonzero entries where G has nonzero entries. Hence, $\hat{A}_c$ is irreducible. To establish the singular M-matrix property, observe that lumping ensures that $\hat{A}_c$ has nonpositive off-diagonal entries. It follows by (4.16) and Theorem 3.2(3) that $\hat{A}_c$ is an irreducible singular M-matrix. By Theorem 3.2(1), $\hat{A}_c$ has a unique right-kernel vector with strictly positive components (up to scaling). The proof now follows formally by induction over the levels. ∎

THEOREM 4.3 (fixed-point property). *The exact solution, $\mathbf{x}$, is a fixed point of the MCAMG V-cycle.*

Proof. Property (4.17) implies that $\mathbf{e}_c = \mathbf{1}_c$ is a solution of the coarse-level equation $\hat{A}_c\, \mathbf{e}_c = \mathbf{0}$ for $\mathbf{x}_i = \mathbf{x}$. We note that this solution is unique (up to scaling),

since $\hat{A}_c$ is an irreducible singular M-matrix. The coarse-level correction formula then gives $\mathbf{x}_{i+1} = \mathrm{diag}(\mathbf{x}_i)\, P\, \mathbf{e}_c = \mathrm{diag}(\mathbf{x})\, P\, \mathbf{1}_c = \mathbf{x}$. The result now follows by the fact that the exact solution, $\mathbf{x}$, is a fixed point of the weighted Jacobi relaxation scheme (see section 4.4). ∎

5. Numerical results. In this section we present numerical convergence results for MCAMG. Testing is performed for a variety of problems that fall into two distinct categories: those for which B has a real spectrum and those for which the spectrum of B is complex. In the latter case, we plot the spectrum of B, and, in both cases, we analyze how the magnitude of the subdominant eigenvalue approaches 1 as the problem size increases. Recall that for irreducible stochastic matrices, a subdominant eigenvalue is an eigenvalue with magnitude $|\lambda_2| = \max_{\lambda \in \Sigma(B), |\lambda|<1}\{|\lambda|\}$, where $\Sigma(B)$ is the spectrum of B. We are interested in the behavior of the subdominant eigenvalue as the problem size increases, since traditional one-level iterative methods, like the power method, are increasingly slow to converge when $|\lambda_2| \to 1$ as n increases.

In the tables that follow, n is the number of degrees of freedom on the finest level, and γ is the geometric mean of the convergence factors of the last five V-cycles, which are defined as the ratios of the one-norm of the residual, $\|A\,\mathbf{x}_i\|_1$, after and before each cycle. Note that the $\mathbf{x}_i$ are scaled such that $\|\mathbf{x}_i\|_1 = 1$. For all the numerical results presented in this paper, we start from a random, strictly positive initial guess and iterate until the residual has been reduced by a factor of 10^{-8} measured in the one-norm. We perform a direct solve on the coarse level when $n < 12$. All V-cycles used are $(1,1)$ cycles, with one prerelaxation and one postrelaxation on each level. A scalable (or optimal) method requires γ to be uniformly bounded away from one as n is increased, resulting in the number of required iterations to be bounded as well. In the tables, *it* is the number of iterations performed, and *lev* is the number of levels in the last cycle. Initially, the number of levels may occasionally change slightly from cycle to cycle. However, as the algorithm converges, the number of levels per cycle becomes constant. This is due to the adaptive nature of the algorithm: as the approximate solution converges, the aggregation hierarchy essentially becomes fixed. The weight in the weighted Jacobi relaxation is chosen as $\omega = 0.7$. The operator complexity of the last cycle, C_{op}, is defined as the sum of the number of nonzero elements in all operators, A, on all levels, divided by the number of nonzero elements in the fine-level operator. This number gives a good indication of the amount of work required for a cycle and, for a scalable (or optimal) method, it should be bounded by a constant not too much larger than one as n increases. We also provide an effective convergence factor defined as $\gamma_{eff} = \gamma^{1/C_{op}}$. This effective convergence factor takes work into account and makes it easier to evaluate the overall efficiency of the method as n increases. For a scalable method, γ_{eff} should be uniformly bounded below one as the problem size increases. Finally, R_l is the lumping ratio of the last cycle, defined as the sum of the number of "offending" elements in operators A on all levels divided by the sum of the number of nonzero elements in A on all levels. This ratio gives the fraction of matrix elements for which lumping is required and is, thus, an indication of the extra work required for lumping. Note that no lumping is required in the fine-level matrix, so lumping contributes only extra work starting from the second level.

For each test problem, we compare our results with numerical tests performed using algebraic smoothed aggregation for Markov chains (A-SAM) in [9]. Depending on the case, so-called distance-one or distance-two aggregations are employed (see [9]), whichever is the most efficient. Strength parameter $\theta = 0.25$ is used for the A-SAM simulations, except where noted.

TABLE 5.1
Uniform chain.

	MCAMG						A-SAM [9] distance-two					
n	γ	it	C_{op}	γ_{eff}	lev	R_l	γ	it	C_{op}	γ_{eff}	lev	R_l
2187	0.18	11	1.99	0.42	9	0	0.31	12	1.49	0.46	6	0
6561	0.18	11	2.00	0.43	10	0	0.31	12	1.49	0.46	7	0
19683	0.18	11	2.00	0.42	12	0	0.32	12	1.49	0.47	8	0
59049	0.18	11	2.00	0.43	14	0	0.32	12	1.50	0.47	9	0

5.1. Real spectrum problems. In this section, we consider test problems for which B has a real spectrum. These include a uniform chain, a uniform two-dimensional (2D) lattice, an anisotropic 2D lattice, and a random walk on an unstructured planar graph. Each test problem has also been considered in [9], so our description is brief. The test problems are generated by undirected graphs with weighted edges. The weights determine the transition probabilities: the transition probability from nodes i to j is given by the weight of the edge from nodes i to j divided by the sum of the weights of all outgoing edges from node i. It is easy to show that the spectrum of the resulting transition matrices is real (they are similar to their symmetric weight matrices). These problems are rather academic test problems, since the exact solution in each node can easily be calculated using only local information. Nevertheless, these problems constitute interesting initial test cases for our algorithm because they also have a strong connection with linear problems from PDEs, where much is understood about AMG.

The first test problem we consider is a one-dimensional Markov chain generated by a linear graph with weighted edges. We choose the weights equal to 1 so the transition probabilities from interior nodes are $\frac{1}{2}$ and the transition probabilities from end nodes are 1. Table 5.1 shows the numerical convergence results for MCAMG. Observe that MCAMG V-cycles lead to computational complexity that is optimal: C_{op} is bounded, γ is constant and much smaller than one, and the number of required iterations is small and constant for increasing n. Compared to the A-SAM results from [9], it is apparent that MCAMG performs similarly to A-SAM. Note that MCAMG does not require any lumping on the last cycle.

The remaining tests we consider are for 2D problems. In the uniform 2D lattice, all weights are chosen equal to 1, and, in the anisotropic lattice, horizontal weights are 1 while vertical weights are 10^{-6} (see [9]). The numerical results for the uniform 2D lattice and the anisotropic lattice are given in Tables 5.2 and 5.3, respectively. The results obtained for the uniform lattice and the anisotropic lattice are very similar, and, in both cases, the MCAMG V-cycles lead to computational complexity that is optimal. Furthermore, we observe that, in each case, the same number of iterations are required to achieve convergence, and the effective convergence factors are almost identical. Note also that no lumping is required on the last cycle. In both cases, it is clear that MCAMG significantly outperforms A-SAM [9].

In our final test problem of this section, we consider an unstructured planar (undirected) graph and calculate the stationary probability distribution of the random walk on the graph. The graph is generated by choosing n random points in the unit square and triangulating them using Delaunay triangulation. The random walk on the graph is modeled by a Markov chain with the transition probability from node i to node j given by the reciprocal of the number of edges incident on node i (equal weights). Table 5.4 shows good convergence results for the unstructured planar graph problem with very little lumping on the last cycle. It appears that C_{op} is bounded,

TABLE 5.2
Uniform 2D lattice.

	MCAMG						A-SAM [9] distance-two					
n	γ	it	C_{op}	γ_{eff}	lev	R_l	γ	it	C_{op}	γ_{eff}	lev	R_l
1024	0.23	11	2.25	0.53	6	0	0.49	20	1.42	0.60	4	4.5e−3
4096	0.23	11	2.26	0.52	7	0	0.49	20	1.47	0.62	4	1.7e−3
16384	0.23	11	2.27	0.53	8	0	0.59	20	1.56	0.72	5	1.4e−3
65536	0.23	11	2.26	0.52	9	0	0.66	21	1.59	0.77	6	1.3e−3

TABLE 5.3
Anisotropic 2D lattice ($\varepsilon = 1e-6$).

	MCAMG						A-SAM [9] distance-two					
n	γ	it	C_{op}	γ_{eff}	lev	R_l	γ	it	C_{op}	γ_{eff}	lev	R_l
1024	0.18	11	2.41	0.49	8	0	0.49	20	1.42	0.60	4	4.5e−3
4096	0.18	11	2.50	0.50	10	0	0.49	20	1.47	0.62	4	1.7e−3
16384	0.18	11	2.56	0.51	12	0	0.59	20	1.56	0.72	5	1.4e−3
65536	0.18	11	2.61	0.52	14	0	0.66	21	1.59	0.77	6	1.3e−3

TABLE 5.4
Unstructured planar graph.

	MCAMG						A-SAM [9] distance-one					
n	γ	it	C_{op}	γ_{eff}	lev	R_l	γ	it	C_{op}	γ_{eff}	lev	R_l
1024	0.44	16	2.15	0.68	6	0	0.53	20	1.69	0.68	5	2.6e−2
2048	0.36	14	2.20	0.62	6	6.4e−5	0.52	19	1.68	0.68	5	2.1e−2
4096	0.40	15	2.23	0.66	7	1.3e−4	0.61	21	1.80	0.76	5	2.4e−2
8192	0.40	15	2.27	0.67	8	1.1e−4	0.64	22	1.92	0.79	7	2.5e−2
16384	0.37	14	2.30	0.65	8	8.3e−5	0.76	30	2.03	0.87	7	2.4e−2
32768	0.37	14	2.29	0.65	9	1.0e−4	0.74	28	2.08	0.86	7	2.4e−2

and consideration of γ and the number of iterations suggests that the computational complexity is optimal. Compared to the results from [9], it is evident that again, MCAMG significantly outperforms A-SAM.

To investigate the nature of these test problems, we seek the following asymptotic relationship between the subdominant eigenvalue of B and the problem size n:

$$1 - |\lambda_2| \approx C \left(\frac{1}{n}\right)^p, \tag{5.1}$$

where $C > 0$ and $p > 0$ are constants. We are interested in the exponent p, which determines how rapidly $|\lambda_2| \to 1$ as $n \to \infty$. An estimate of p provides insight into the rate at which traditional one-level iterative methods converge for this type of problem. For problems for which $|\lambda_2|$ approaches 1 as in (5.1), we expect multilevel methods to outperform traditional one-level iterative methods. Figure 5.1 shows log-log plots of $1 - |\lambda_2|$ as a function of n (hollow circles) and the linear best fit to these data points. For the uniform chain test problem, observe that $p \approx 2$ and, for all other test problems, that $p \approx 1$. As expected, $p \approx 2/d$, with d the dimensionality of the problem.

5.2. Complex spectrum problems. In this section, we consider the test problems for which B has a complex spectrum and for which the exact solution cannot easily be computed. These include a tandem queueing network, a stochastic Petri net (SPN) problem, an asynchronous transfer mode (ATM) queueing network, and an octagonal mesh problem. The first three test problems have also been considered

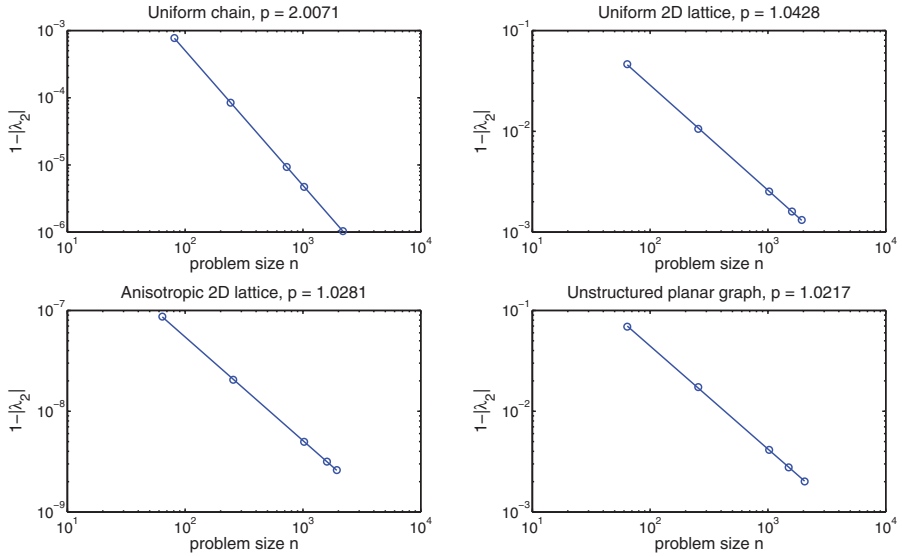

FIG. 5.1. *Magnitude of subdominant eigenvalue as a function of problem size.*

TABLE 5.5
Tandem queueing network.

	MCAMG						A-SAM [9] distance-two					
n	γ	it	C_{op}	γ_{eff}	lev	R_l	γ	it	C_{op}	γ_{eff}	lev	R_l
1024	0.31	15	4.68	0.78	7	1.2e−1	0.41	20	2.04	0.64	4	7.6e−2
4096	0.32	16	4.53	0.78	8	1.3e−1	0.45	24	2.12	0.69	5	5.5e−2
16384	0.32	15	4.57	0.78	10	1.1e−1	0.56	30	2.18	0.77	6	5.3e−2
65536	0.32	15	4.61	0.78	11	6.5e−2	0.71	37	2.37	0.86	6	1.3e−1

in [9, 13, 17, 22]. We conclude this section with an analysis of the subdominant eigenvalues and with plots of the spectra for each test problem. Note that, for brevity, we do not go into great detail describing our test cases, but instead we refer the reader to the appropriate sources.

The first test problem is an open tandem queueing network from [22]; see also [9]. Two finite queues with single servers are placed in tandem. Customers arrive according to a Poisson distribution with rate μ, and the service time distribution at the two single-server stations is Poisson with rates μ_1 and μ_2. In our numerical experiments, we limit the number of customers in the queues to $N = 31, 63, 127$, and 255. We choose $(\mu, \mu_1, \mu_2) = (10, 11, 10)$ for the weights. The states of the system can be represented by tuples (n_1, n_2), with n_1 the number of customers waiting in the first queue and n_2 in the second queue. The total number of states is given by $(N+1)^2$. The states can be represented on a 2D regular lattice. In the directed graph, the transition probability from node i to node j is given by the weight of the edge from node i to j divided by the sum of the weights of all outgoing edges from node i. Table 5.5 shows the numerical results for the tandem queueing network test problem. Iteration numbers are constant, and the operator complexity grows somewhat as a function of problem size for this nonsymmetric problem but appears bounded. The amount of lumping required for this nonsymmetric 2D problem is larger than for the previous problems but is still relatively small and does not add much extra work. These results are competitive with those obtained using A-SAM in [9].

TABLE 5.6
Stochastic Petri net, $\Lambda = (1, 3, 7, 9, 5)$.

	MCAMG ($\theta = 0.7$)						A-SAM [9] distance-two ($\theta = 0.5$)					
n	γ	it	C_{op}	γ_{eff}	lev	R_l	γ	it	C_{op}	γ_{eff}	lev	R_l
1496	0.39	17	2.56	0.69	8	1.1e−3	0.40	17	3.76	0.79	5	1.5e−1
2470	0.38	17	2.62	0.69	8	1.4e−3	0.38	16	4.26	0.80	5	1.5e−1
3795	0.40	17	2.71	0.71	9	1.1e−3	0.37	16	4.53	0.80	5	1.5e−1
10416	0.41	18	2.92	0.74	10	1.1e−3	0.45	18	5.31	0.86	5	1.4e−1
16206	0.43	18	2.98	0.75	11	9.7e−4	0.45	18	5.55	0.87	5	1.3e−1
23821	0.46	18	3.08	0.78	11	9.1e−4	0.41	18	6.01	0.86	6	1.3e−1
33511	0.43	18	3.16	0.77	11	8.4e−4	0.45	18	6.57	0.88	6	1.2e−1
45526	0.42	18	3.22	0.76	12	7.6e−4	0.42	18	6.91	0.88	6	1.2e−1

The next problem we consider is derived from an SPN. Petri nets are a formalism for the description of concurrency and synchronization in distributed systems. They consist of the following: *places*, which model conditions or objects, *tokens*, which represent the specific value of the condition or object; *transitions*, which model activities that change the value of conditions or objects; and *arcs*, which specify interconnection between places and transitions. A stochastic Petri net is a standard Petri net together with a tuple $\Lambda = (r_1, \ldots, r_n)$ of exponentially distributed transition firing rates. Furthermore, we know from [15] that a finite place, finite transition, marked SPN is isomorphic to a one-dimensional discrete-space Markov process. For an in-depth discussion of Petri nets, the reader is referred to [1, 15].

We test MCAMG on the SPN described in [13]. Table 5.6 shows that the number of iterations is bounded and that the effective convergence factor, γ_{eff}, is bounded well below one, from which we conclude that optimal computational complexity is achieved. Comparing the effective convergence factors of MCAMG and A-SAM, it is evident that MCAMG outperforms A-SAM. Note that in order to obtain optimal results for MCAMG, it was necessary to use $\theta = 0.7$ instead of $\theta = 0.25$.

The next test problem we consider is a multiclass, finite buffer priority system. This model can be applied to telecommunications modeling and has been used to model ATM queueing networks as discussed in [17]. For a complete description, including all the model parameters, see [22, 23]. The code and data files used to build the transition rate matrix corresponding to this Markov chain model are provided freely on the Web [23]. To obtain a column-stochastic transition probability matrix B from the row-oriented transition rate matrix Q, we set

$$B = (\mathrm{diag}(Q) - Q^T)(\mathrm{diag}(Q))^{-1}.$$

Transition rate matrices were constructed for buffer sizes 16, 20, 26, 32, 36, 42, and 50.

Table 5.7 shows the numerical results for the MCAMG V-cycles. The number of iterations and operator complexity are bounded, and it appears that MCAMG performs optimally for this problem. Note that in our initial implementation of the first pass of the coarsening routine, ties for C-point choice were broken in lexicographical order. Due to the particular directionality of the ATM queueing problem, this ordering led to a "worst-case," unusually high number of levels and consequently to poor operator complexity of MCAMG. Instead, we now break ties in a different way using the natural ordering of the heap data structure we use, which has resolved this difficulty and maintains good convergence properties for the other test problems.

The last test problem we consider is an octagonal mesh problem. This problem is constructed to have a full spectrum that fills a large part of the unit circle and features

TABLE 5.7
ATM queueing network.

	MCAMG ($\theta = 0.5$)						A-SAM [9] distance-two					
n	γ	it	C_{op}	γ_{eff}	lev	R_l	γ	it	C_{op}	γ_{eff}	lev	R_l
1940	0.53	25	2.95	0.81	9	1.2e−2	0.26	14	3.23	0.66	5	1.6e−1
3060	0.48	21	2.85	0.77	8	9.4e−3	0.26	13	3.29	0.66	5	1.6e−1
5220	0.51	23	2.88	0.79	9	7.6e−3	0.25	14	3.50	0.67	5	1.6e−1
7956	0.45	21	2.87	0.76	10	6.6e−3	0.25	14	3.74	0.69	5	1.6e−1
10100	0.47	22	2.86	0.77	10	6.0e−3	0.25	13	3.98	0.71	5	1.6e−1
19620	0.42	20	2.83	0.73	11	4.3e−3	0.25	13	4.24	0.72	6	1.7e−1
32276	0.43	22	2.87	0.75	11	5.4e−3	0.25	14	4.54	0.74	6	1.7e−1

elements of Web traffic modeling, restricted to a specific planar 2D graph. In what follows, we provide a complete description of how transition matrix B is constructed. In general, consider graph $\mathcal{H}$, represented by an $n \times n$ binary matrix,

$$H_{ij} = \begin{cases} 1, & \text{if an arc exists from node } j \text{ to node } i \\ 0, & \text{otherwise} \end{cases}.$$

We assume that graph $\mathcal{H}$ has the following properties:

(P1) Every node has an *outgoing arc*.
(At least one nonzero entry in every column of H.)

(P2) Every node has an *incoming arc*.
(At least one nonzero entry of every row of H.)

(P3) There is a directed path between any two nodes.
(H is *irreducible*.)

Then, for any node j in graph $\mathcal{H}$, we assign the following probabilities:

μ_+ Probability of moving forward, distributed equally among outgoing arcs.
(Transition from state j to i for $H_{ij} = 1$.)

μ_0 Probability of staying at current node.
(No transition.)

μ_- Probability of moving backward, distributed equally among incoming arcs.
(Transition from state j to i for $H_{ji} = 1$.)

These values satisfy $\mu_+ + \mu_0 + \mu_- = 1$. We use matrix $H\,\mathrm{diag}(\mathbf{1}^T H)^{-1}$ for transition along outgoing arcs and matrix $H^T \mathrm{diag}(\mathbf{1}^T H^T)^{-1}$ for transition along incoming arcs. Our final column-stochastic matrix is then given by

$$B = \mu_0 I + \mu_+ H\ \mathrm{diag}(\mathbf{1}^T H)^{-1} + \mu_- H^T \mathrm{diag}(\mathbf{1}^T H^T)^{-1}.$$

Note that the graph associated with matrix B is the symmetrized version of graph $\mathcal{H}$. The graph we use to construct B for the numerical test is pictured in Figure 5.2. The transition probabilities used are $\mu_+ = 0.80$, $\mu_0 = 0.15$, and $\mu_- = 0.05$.

Table 5.8 shows good convergence results for both MCAMG and A-SAM. For each algorithm, it appears that C_{op} is bounded, and consideration of γ and the number of iterations suggests that the computational complexity is optimal. Note that the MCAMG operator complexity can be decreased by only performing the first pass of the coarsening routine together with a suitably redefined interpolation formula [11].

One further observation is that the number of levels used by A-SAM is always less than the number of levels used by MCAMG. This is a consequence of the fact that A-SAM coarsens faster than MCAMG, especially when distance-two aggregation is used, since this results in larger aggregates. The number of levels grows logarithmically with

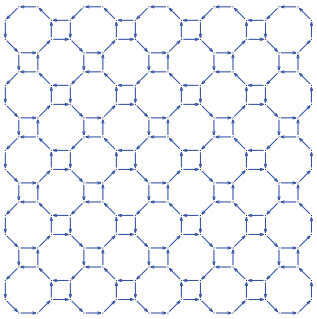

FIG. 5.2. *Graph $\mathcal{H}$ representing the octagonal mesh.*

TABLE 5.8
Octagonal mesh.

	MCAMG						A-SAM [9] distance-two					
n	γ	it	C_{op}	γ_{eff}	lev	R_l	γ	it	C_{op}	γ_{eff}	lev	R_l
1024	0.55	29	4.84	0.88	9	2.4e−2	0.37	18	2.88	0.71	5	1.8e−1
4096	0.55	29	5.41	0.90	11	2.2e−2	0.52	20	3.11	0.81	6	1.7e−1
16384	0.56	29	5.43	0.90	12	2.2e−2	0.60	22	3.17	0.85	6	1.6e−1
32768	0.56	29	5.60	0.90	12	2.2e−2	0.67	24	3.33	0.89	7	1.6e−1
65536	0.56	29	5.46	0.90	13	2.2e−2	0.65	26	3.28	0.88	7	1.6e−1

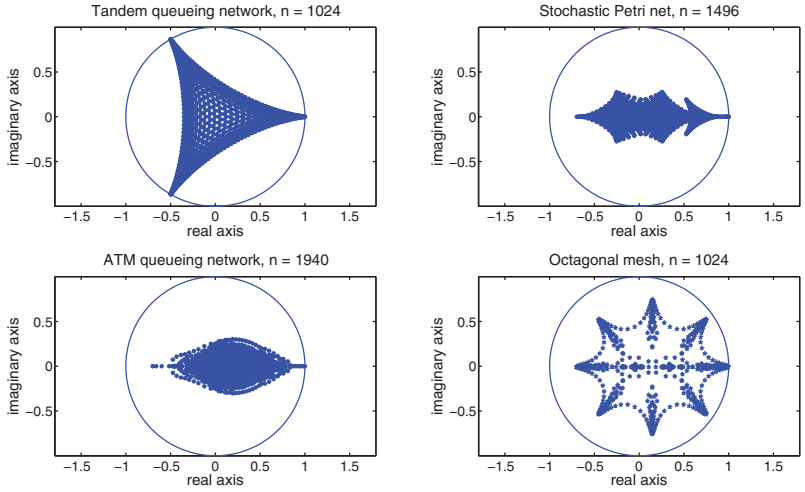

FIG. 5.3. *Complex spectra $\Sigma(B)$.*

problem size, since we coarsen until the coarsest problems are below a fixed size, rather than up to a fixed number of levels, which would not ultimately yield a near-optimal method. For different problems with the same problem size, the number of levels depends of the problem (different problems have different strong connections).

Figure 5.3 shows the unit circle in the complex plane and the spectrum of B for each test problem. Figure 5.4 shows log-log plots of $1 - |\lambda_2|$ as a function of problem size n. We observe that $p \approx 1$ for the tandem queueing network and octagonal mesh problems, $p \approx \frac{2}{3}$ for the stochastic Petri net problem, and $p \approx \frac{1}{2}$ for the ATM queueing network problem.

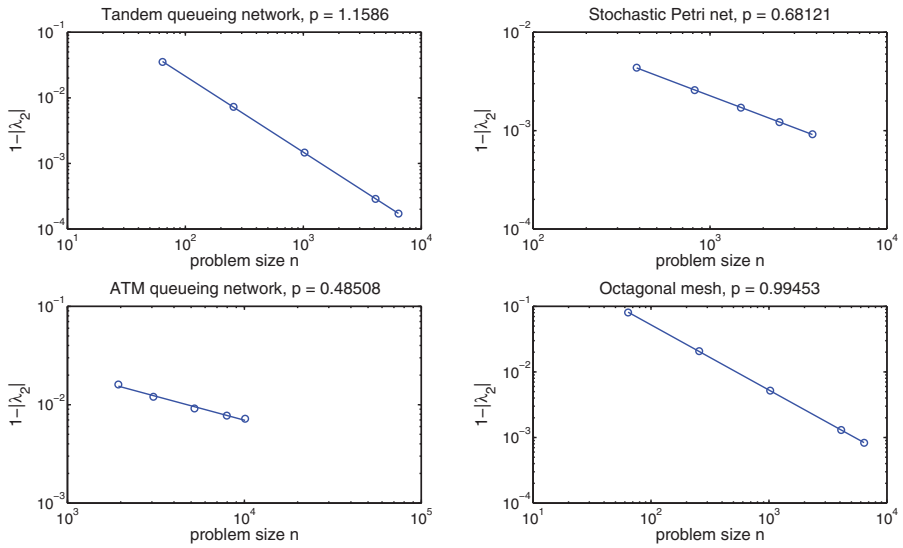

FIG. 5.4. *Magnitude of subdominant eigenvalue as a function of problem size.*

6. Conclusions and future work. We demonstrated how multiplicative algebraic multigrid with lumping and a modified interpolation formula can be used to find the stationary probability vector of a Markov chain. It was shown that the coarse-level operators are irreducible singular M-matrices on all levels, resulting in strictly positive coarse-level corrections. It was also shown that the exact solution is a fixed point of our algorithm. We performed numerical testing for a wide variety of test problems and considered problems with both real and complex spectra. For each test case, we observed that MCAMG V-cycles lead to optimal or nearly optimal multigrid efficiency, that is, the effective convergence factor, γ_{eff}, was bounded uniformly by a constant less than one as the problem size increased. We observed that MCAMG was competitive with A-SAM from [9] and, in most cases, outperformed A-SAM. Furthermore, it was shown that the multilevel method performed well on problems for which traditional one-level iterative methods are slow to converge, i.e., problems for which the magnitude of the subdominant eigenvalue approaches 1 rapidly as the problem size increases.

Further possible avenues of research include parallel implementations of MCAMG, acceleration of AMG V-cycle convergence using combinations of previous iterates as in Krylov methods [20], and the use of acceleration on all recursive levels as in K-cycle methods [16].

REFERENCES

[1] F. BAUSE AND P. KRITZINGER, *Stochastic Petri Nets*, Verlag Vieweg, Germany, 1996.
[2] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, 1987.
[3] A. BRANDT, S. F. MCCORMICK, AND J. W. RUGE, *Algebraic multigrid (AMG) for sparse matrix equations*, in Sparsity and Its Applications, D. J. Evans, ed., Cambridge University Press, Cambridge, 1984.
[4] A. BRANDT AND D. RON, *Multigrid solvers and multilevel optimization strategies*, in Multilevel Optimization and VLSICAD, J. Cong and J. R. Shinnerl, eds., Kluwer, Boston, 2003, pp. 1–69.

[5] M. Brezina, R. D. Falgout, S. MacLachlan, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge, *Adaptive algebraic multigrid*, SIAM J. Sci. Comput., 27 (2006), pp. 1261–1286.
[6] W. L. Briggs, V. Emden Henson, and S. F. McCormick, *A Multigrid Tutorial*, SIAM, Philadelphia, 2000.
[7] G. Chartrand and L. Lesniak, *Graphs & Digraphs*, Chapman and Hall/CRC, Boca Raton, 2005.
[8] H. De Sterck, R. D. Falgout, J. Nolting, and U. M. Yang, *Distance-two interpolation for parallel algebraic multigrid*, Numer. Linear Algebra Appl., 15 (2008), pp. 115–139.
[9] H. De Sterck, T. A. Manteuffel, S. F. McCormick, K. Miller, J. Pearson, J. Ruge, and G. Sanders, *Smoothed aggregation multigrid for Markov chains*, SIAM J. Sci. Comp., to appear.
[10] H. De Sterck, T. A. Manteuffel, S. F. McCormick, Q. Nguyen, and J. Ruge, *Multilevel adaptive aggregation for Markov chains, with application to Web ranking*, SIAM J. Sci. Comput., 30 (2008), pp. 2235–2262.
[11] H. De Sterck, U. M. Yang, and J. J. Heys, *Reducing complexity in parallel algebraic multigrid preconditioners*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 1019–1039.
[12] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.
[13] G. Horton and S.T. Leutenegger, *A multi-level solution algorithm for steady-state Markov chains*, in ACM SIGMETRICS Performance Evaluation Review 22, http://portal.acm.org/citation.cfm?id=183019.183040.
[14] I. Livshits, *An algebraic multigrid wave-ray algorithm to solve eigenvalue problems for the Helmholtz operator*, Numer. Linear Algebra Appl., 11 (2004), pp. 229–239.
[15] M. K. Molloy, *Performance analysis using stochastic Petri nets*, IEEE Trans. Comput., C-31 (1982), pp. 913–917.
[16] Y. Notay and P. S. Vassilevski, *Recursive Krylov-based multigrid cycles*, Numer. Linear Algebra Appl., 15 (2008), pp. 473–487.
[17] B. Philippe, Y. Saad, and W. J. Stewart, *Numerical methods for Markov chain modeling*, Oper. Res., 40 (1992), pp. 1156–1179.
[18] J. W. Ruge, *Algebraic multigrid (AMG) for geodetic survey problems*, in Proceedings of the International Multigrid Conference, Copper Mountain, CO, 1983.
[19] J. W. Ruge and K. Stueben, *Algebraic Multigrid (AMG)*, in Multigrid Methods Frontiers Appl. Math., S. F. McCormick, ed., SIAM, Philadelphia, 1987, pp. 73–130.
[20] Yo. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.
[21] H. A. Simon and A. Ando, *Aggregation of variables in dynamic systems*, Econometrica, 29 (1961), pp. 111–138.
[22] W. J. Stewart, *An Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.
[23] W. J. Stewart, *MARCA_Models*, from http://www4.ncsu.edu/~billy/MARCA_Models/MARCA_Models.html.
[24] Y. Takahashi, *A Lumping Method for Numerical Calculations of Stationary Distributions of Markov Chains*, Research report B-18, Department of Information Sciences, Tokyo Institute of Technology, 1975.
[25] E. Treister and I. Yavneh, *Square and stretch multigrid for stochastic matrix eigenproblems*, Numer. Linear Algebra Appl., submitted.
[26] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Elsevier, Academic Press, 2001.
[27] E. Virnik, *An algebraic multigrid preconditioner for a class of singular M-matrices*, SIAM J. Sci. Comput., 29 (2007), pp. 1982–1991.