ALGEBRAIC MULTIGRID FOR MARKOV CHAINS

KILLIAN MILLER*‡‡

Abstract. An algebraic multigrid (AMG) method is presented for the calculation of the stationary probability vector of an irreducible Markov chain. We propose a modified AMG interpolation formula, which produces a nonnegative interpolation operator with unit row sums. It is shown how the adoption of a lumping technique maintains the irreducible singular M-matrix character of the coarse-level operators on all levels. Together, these properties are sufficient to establish the well-posedness of our algorithm. Numerical results show how our method 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 Computational Markov chains, 60J22 Computational methods in Markov chains, 65F10 Iterative methods for linear systems, 65F15 Eigenvalues, eigenvectors

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

Multigrid methods have already been considered for Markov chains, including multilevel aggregation methods [9, 7] and smoothed aggregation [6]. Our present approach for Markov chains is based on AMG for nonsingular linear systems, and incorporates aspects of recent work on aggregation multigrid for Markov chains. Starting from the classical definition of AMG interpolation [4], we propose a modified interpolation formula, which produces a nonnegative interpolation 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 [6], maintains the irreducible singular M-matrix character of the coarse-level operators on all levels. 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. Note that the use of AMG has already been explored for Markov chain problems [16], in the context of additive AMG used as a preconditioner for GMRES. Our formulation, however, is multiplicative, and near-optimal results are obtained without needing GMRES acceleration. Our formulation is also different in that it is related to adaptive AMG [3].

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

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 [8]. In what follows, we consider the case where B is irreducible. Irreducibility is an essential ingredient in proving the well-posedness of our algorithm, so for the sake of completeness, we formally define it here.

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.

*Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario, Canada
‡‡k7miller@uwaterloo.ca

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 there is an arc from v_i to v_j if and only if $a_{ij} \neq 0$.

DEFINITION 2.3 (Irreducible matrix).
A 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 [16], we can equivalently restate the problem of solving for the stationary probability vector as solving for a strictly positive vector of unit length in the nullspace of a 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 $A \equiv I - B$ and B is irreducible. The matrix A is a singular M-matrix. Note that $\mathbf{1}^T A = \mathbf{0}$.

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. 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 $b_{ij} \geq 0$ for all i, j, such that $A = \rho(B)I - B$.

The justification that $A \equiv I - B$ 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, 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 taken from [2, 6, 16]. We choose to restate them here so that the paper is self-contained.

THEOREM 3.2 (Some properties of singular M-matrices).
(1) Irreducible singular M-matrices have a unique solution to the problem $A\,\mathbf{x} = \mathbf{0}$, up to scaling. Scaling 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 nullspace and the off-diagonal elements of A are nonpositive, then A is a singular M-matrix.
(4) If A is an irreducible singular M-matrix, then each principal submatrix of A 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 used below to prove that the coarse operators of our AMG method are irreducible singular M-matrices on all levels.

4. Algebraic multigrid for Markov Chains. In this section we recall the principal features of the classical AMG V-cycle [4, 15] 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 [6]. We conclude this section by stating 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. The multiplicative error, $\mathbf{e}_i$, is defined by $\mathbf{x} = \mathrm{diag}(\mathbf{x}_i)\,\mathbf{e}_i$, where $\mathbf{x}$ is the exact solution of (3.1), and $\mathbf{x}_i$ is the ith iterate. As we will see below, the iterates $\mathbf{x}_i$ obtained by our algorithm have strictly positive components. Equation 3.1 can then be rewritten as

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

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:

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

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 $\bar{A} \equiv A\,\mathrm{diag}(\mathbf{x}_i)$. It follows by Theorem 3.2(3) that $\bar{A}$ is also an irreducible singular M-matrix. Thus, we can rewrite equation (4.1) in terms of $\bar{A}$, which results in the following fine-level error equation

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

In order to seek a coarse representation of equation (4.3), we first perform the two-pass AMG coarsening routine described in [4], which determines the set of points on the coarse level. Basing *strength of connection* (and coarsening) on the scaled operator $\bar{A}$ is motivated heuristically in [7] (at convergence, $\mathbf{1}$ lies in the nullspace of $\bar{A}$, so standard AMG coarsening and interpolation approaches work well). Recall that, given a threshold value $\theta \in [0,1]$, the point i depends strongly on the point j if

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

In this paper we use $\theta = 0.25$. 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 $\bar{A}_c \equiv R\,\bar{A}\,P$ 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. In fact, the coarse-level operator $\bar{A}_c$ may not be an irreducible singular M-matrix. As we show in the next section, this problem can be remedied by applying a *lumping method* [6] to the coarse-level operator, whereby we obtain the *lumped* coarse-level operator, $\hat{A}_c$. We prove below that this lumped $\bar{A}_c$ 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 *smooth components* are accurately transferred between these levels. By smooth components we are referring to the components of the solution 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 [4], 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 fine level H, and thus have partitioned H into a set of coarse points C and fine points 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} 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 [4]:

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

where $C_i \cup D_i^s \cup D_i^w = N_i$, which is the set of all points $k \neq i$ on the fine level that are connected to point i in $\bar{A}$ ($\bar{a}_{ik} \neq 0$). Here, D_i^s is the set of F-points that strongly influence i, and D_i^w is the set of points that do not strongly influence i. Note that D_i^w may contain both F-points and C-points.

We cannot directly use this formula for our problem for several reasons: the denominator of (4.9) may be zero, some of the weights may be negative, and the weights do not sum to one. We do indeed 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). Note that this also implies that constants are interpolated exactly. To ensure that P enjoys this property, we rescale the w_{ij}s of (4.8). The rescaled weights $\bar{w}_{ij}$ are given by

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

Under the premise that $\bar{A}$ is an irreducible singular M-matrix, Theorem 3.2(2) assures us that all matrix elements in (4.10) are negative. Since the two-pass AMG coarsening routine ensures that $C_i \neq \emptyset$, it follows that the denominator term in equation (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, we find that $\bar{w}_{ij} > 0$ for all $i \in F$ 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 which makes certain 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, we define the matrices D, L and U 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 since $\bar{A}$ is a singular M-matrix and P has nonnegative entries. In general the matrix $P^T D\, P$ is not diagonal, and as a result $\bar{A}_c$ may have positive off-diagonal entries, in which case it cannot 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 [6] for smoothed aggregation multigrid methods for Markov chains. An overview of the lumping procedure is provided below.

We consider a modified version, $\hat{S}$, of S, which is 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).

We define an *offending index pair* as the tuple (i, j) such that $s_{ij} \neq 0$ 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 for the sign in $\bar{A}_c$ at location (i, j) we add a matrix $S_{\{i,j\}}$ to S, with the elements of $S_{\{i,j\}}$ equaling $\beta_{\{i,j\}}$ at positions (i, i) and (j, j), $-\beta_{\{i,j\}}$ at positions (i, j) and (j, i), and otherwise being zero. This 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 $s_{ij} - g_{ij} - \beta_{\{i,j\}} < 0$ and $s_{ji} - g_{ji} - \beta_{\{i,j\}} < 0$, resulting in strictly negative off-diagonal elements in $\hat{A}_c$ at locations (i, j) and (j, i). Note 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, only one matrix $S_{\{i,j\}}$ has to be added to S. In our implementation,

$$\beta_{\{i,j\}} = \max\{s_{ij} + (\eta - 1)g_{ij},\ s_{ji} + (\eta - 1)g_{ji}\}, \tag{4.13}$$

and η is a fixed parameter $\in (0,1]$. Note that $\beta_{\{i,j\}} > 0$ always holds. Symmetric matrices such as $S_{\{i,j\}}$ are used to modify S so that column sums and row sums of $\bar{A}_c$ are conserved - we shall see shortly why this is of significance. It is also 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 comment on η, which determines how much of S we lump to the diagonal. Experimentally, it has been observed that we should lump as little as possible and so η should be chosen small [6]. In practice, $\eta = 0.01$ seems to be a good choice.

We conclude this section by showing that $\hat{A}_c$ has a positive left kernel vector on all levels, and at convergence, a positive right kernel vector. Since $(\hat{S} - S) = \sum S_{\{i,j\}}$, where the sum is over all matrices $S_{\{i,j\}}$ added to S, it follows that

$$\mathbf{1}_c^T \hat{A}_c = \mathbf{1}_c^T \bar{A}_c + \mathbf{1}_c^T (\hat{S} - S) = \mathbf{1}_c^T \bar{A}_c = \mathbf{0} \quad \forall\, \mathbf{x}_i, \tag{4.14}$$

$$\hat{A}_c\, \mathbf{1}_c = \bar{A}_c\, \mathbf{1}_c + (\hat{S} - S)\, \mathbf{1}_c = \bar{A}_c\, \mathbf{1}_c = \mathbf{0} \quad \text{for } \mathbf{x}_i = \mathbf{x}. \tag{4.15}$$

4.4. Relaxation. In this paper we use the weighted Jacobi method for all relaxation operations. Decomposing matrix A into its diagonal part and strictly upper and lower triangular parts $A = D - (L + U)$, the weighted Jacobi iterative scheme for solving $A\,\mathbf{x} = \mathbf{0}$ is given by

$$\mathbf{x}_{i+1} = (1 - \omega)\mathbf{x}_i + \omega D^{-1}(L + U)\, \mathbf{x}_i \tag{4.16}$$

where $\omega \in (0,1)$ is a fixed weight parameter. By the singular M-matrix character of A, it follows that $\mathbf{x}_i$ has strictly positive entries for all $i \geq 0$ if the initial guess $\mathbf{x}_0$ has strictly positive entries and $\omega \in (0,1)$. By normalizing the result after relaxation, the constraint that $\mathbf{x}$ be a probability vector can be satisfied.

4.5. MCAMG V-cycle algorithm. Now that a solid theoretical footing for our approach has been established, 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**
  $\mathbf{x} \leftarrow$ Relax(A, $\mathbf{x}$) ν_1 times
  $\bar{A} \leftarrow A$ diag($\mathbf{x}$)
  Compute the set of coarse-level points C
  Construct the interpolation operator P
  Obtain lumped coarse-level operator $\hat{A}_c \leftarrow$ Lump($\bar{A}$, P, η)
  $\mathbf{e}_c \leftarrow$ MCAMG($\hat{A}_c$, $\mathbf{1}_c$, ν_1, ν_2) `/* coarse-level solve */`
  $\mathbf{x} \leftarrow$ diag($\mathbf{x}$) $P\,\mathbf{e}_c$ `/* coarse-level correction */`
  $\mathbf{x} \leftarrow$ Relax(A, $\mathbf{x}$) ν_2 times
else
  $\mathbf{x} \leftarrow$ direct solve of $A\,\mathbf{x} = \mathbf{0}$
end

4.6. Well-posedness of MCAMG. This section establishes the well-posedness of the MCAMG algorithm. 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 coarse-level points 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 in $\bar{A}$ correspond to self-links in the directed graph of $\bar{A}$. Let $(L+U)_{kl} \neq 0$ for some k and l. 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 $p_{kJ} \neq 0$. Note that since P has unit row sums, every row of P contains at least one nonzero element and 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 is 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 indices 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 [5]. Thus, we can find a directed path in $\Gamma(G)$ that begins at V_0 and ends at V_k. Finally, we observe that C-points V_0 and V_k were chosen such that they interpolate to i and j, respectively. But, by the definition of P, the only point that interpolates to a given C-point is the point itself. Thus, 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. □

We are now in a position to state and prove the two main results of this section. The first theorem guarantees the existence and uniqueness of a right kernel vector of $\hat{A}_c$ with positive components on all levels. The second theorem guarantees that the exact solution $\mathbf{x}$ is a fixed point of our algorithm.

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 the matrix $G = P^T(L+U)\,P$ is irreducible. However, lumping ensures that $\hat{A}_c$ has nonzero entries where G has nonzero entries. Hence, $\hat{A}_c$ is irreducible. To show the singular M-matrix property we observe that lumping ensures $\hat{A}_c$ has nonpositive off-diagonal entries. It now follows by (4.14) and Theorem 3.2(3) that $\hat{A}_c$ is an irreducible singular M-matrix. The proof now follows formally by induction over the levels.

Since $\hat{A}_c$ is an irreducible singular M-matrix on all coarse levels, by Theorem 3.2(1) there exists a unique right kernel vector with strictly positive components (up to scaling) on all levels. □

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

Proof. Property (4.15) 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 $\mathbf{x}_{i+1} = \mathrm{diag}(\mathbf{x}_i)\,P\,\mathbf{e}_c = \mathrm{diag}(\mathbf{x})\,P\,\mathbf{1}_c = \mathbf{x}$. □

Together, Theorems 4.2 and 4.3 establish the well-posedness of our algorithm. In the following section we present numerical results for a wide range of test problems.

5. Numerical results. In this section we present numerical convergence results for MCAMG. Testing is performed for a variety of problems which 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 for a given problem size.

In the tables that follow, 'n' is the number of degrees of freedom and 'γ' is the geometric mean of the convergence factors of the last five V-cycles (the convergence factor of a V-cycle is defined as the ratio of the one-norm of the residual, $\|A\,\mathbf{x}_i\|_1$, after and before the cycle with $\|\mathbf{x}_i\|_1 = 1$ for all i). 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, or until 100 cycles have been performed, whichever comes first. We perform a direct solve on the coarse level when $n < 12$. All V-cycles used are $(1,1)$ cycles, with one pre-relaxation and one post-relaxation 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. The weight in the weighted Jacobi relaxation is chosen as $\omega = 0.7$. 'C_{op}' is the operator complexity of the last cycle, 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. Finally, 'R_{lump}' 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 only contributes extra work starting from the second level.

For the first four test problems, we compare our results with numeric tests performed using algebraic smoothed aggregation (A-SAM) in [6]. Depending on the case, so-called distance-one or distance-two aggregation are employed (see [6]), whichever is the most efficient.

5.1. Real spectrum problems. In this section we consider test problems for which B has a real spectrum. These include a uniform 2D lattice, an anisotropic 2D lattice and a random walk on an unstructured planar graph. Each of these test problems has also been considered in [6], and for this reason our description of each is brief. The test problems are generated by undirected graphs with weighted edges. The weights determine the transition probabilities: the transition probability from node i to 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. In each case, it is easy to show that the spectrum of the resulting transition matrix B is real.

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 [6]). The numerical results for the uniform 2D lattice and the anisotropic lattice are given in Table 5.1 and Table 5.2, respectively. The results obtained

	MCAMG						A-SAM [6] distance-two					
n	γ	it	C_{op}	γ_{eff}	lev	R_{lump}	γ	it	C_{op}	γ_{eff}	lev	R_{lump}
64	0.22	11	2.02	0.47	3	0	0.42	18	1.25	0.50	2	0.0e-0
256	0.25	12	2.20	0.54	5	0	0.47	19	1.35	0.57	3	1.2e-3
1024	0.23	11	2.20	0.51	6	0	0.49	20	1.42	0.60	4	4.5e-3
4096	0.23	11	2.20	0.52	7	0	0.49	20	1.47	0.62	4	1.7e-3
16384	0.24	11	2.20	0.52	8	0	0.59	20	1.56	0.72	5	1.4e-3
65536	0.24	11	2.20	0.52	9	0	0.66	21	1.59	0.77	6	1.3e-3

TABLE 5.1
Uniform 2D lattice.

	MCAMG						A-SAM [6] distance-two					
n	γ	it	C_{op}	γ_{eff}	lev	R_{lump}	γ	it	C_{op}	γ_{eff}	lev	R_{lump}
64	0.19	11	2.15	0.46	4	0	0.40	17	1.76	0.59	3	0.0e-0
256	0.19	11	2.42	0.51	6	0	0.33	15	2.23	0.61	4	7.4e-4
1024	0.18	11	2.58	0.52	8	0	0.33	14	2.81	0.68	5	1.6e-3
4096	0.18	11	2.67	0.53	10	0	0.33	14	3.43	0.73	7	4.9e-4
16384	0.18	11	2.73	0.54	12	0	0.33	13	4.17	0.77	7	2.5e-4
65536	0.18	11	2.76	0.54	14	0	0.32	13	4.80	0.79	9	7.6e-5

TABLE 5.2
Anisotropic 2D lattice ($\varepsilon = 1e-6$).

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 [6].

In this test problem, 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 modelled 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 5.3 shows the numerical results. We observe good convergence results for the unstructured

	MCAMG						A-SAM [6] distance-one					
n	γ	it	C_{op}	γ_{eff}	lev	R_{lump}	γ	it	C_{op}	γ_{eff}	lev	R_{lump}
1024	0.40	15	2.13	0.65	6	0	0.53	20	1.69	0.68	5	2.6e-2
2048	0.33	14	2.22	0.61	7	3.2e-5	0.52	19	1.68	0.68	5	2.1e-2
4096	0.40	15	2.19	0.66	7	3.2e-5	0.61	21	1.80	0.76	5	2.4e-2
8192	0.40	15	2.25	0.66	8	4.7e-5	0.64	22	1.92	0.79	7	2.5e-2
16384	0.37	14	2.26	0.65	9	3.5e-5	0.76	30	2.03	0.87	7	2.4e-2
32768	0.37	14	2.28	0.65	9	7.6e-5	0.74	28	2.08	0.86	7	2.4e-2

TABLE 5.3
Unstructured planar graph.

planar graph problem with very little lumping on the last cycle. It appears that C_{op} is bounded, and consideration of γ and the number of iterations suggest that the computational complexity is optimal. Compared to the results in [6], it is evident that MCAMG significantly outperforms A-SAM for this test problem.

5.2. Complex spectrum problems. In this section we consider the test problems for which B has a complex spectrum. These include a tandem queueing network and a stochastic Petri net problem. These test problems have also been considered in [6, 9]. We conclude this section with a plot of the spectrum of B for each test problem.

The first test problem is an open tandem queueing network from [14], see also [6]. Table 5.4 shows the numerical results for the tandem queueing network test problem. Iteration numbers are constant

	MCAMG						A-SAM [6] distance-two					
n	γ	it	C_{op}	γ_{eff}	lev	R_{lump}	γ	it	C_{op}	γ_{eff}	lev	R_{lump}
256	0.32	15	4.08	0.75	5	7.6e-2	0.39	18	1.94	0.61	4	1.1e-1
1024	0.32	15	4.41	0.77	7	7.4e-2	0.41	20	2.04	0.64	4	7.6e-2
4096	0.32	16	4.49	0.78	8	7.0e-2	0.45	24	2.12	0.69	5	5.5e-2
16384	0.32	16	4.58	0.78	10	9.2e-2	0.56	30	2.18	0.77	6	5.3e-2
65536	0.33	15	4.56	0.78	11	4.9e-2	0.71	37	2.37	0.86	6	1.3e-1

TABLE 5.4
Tandem queueing network.

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 small and does not add much extra work. These results are competitive with those obtained using A-SAM in [6].

The final test problem is derived from a stochastic Petri net (SPN). Petri nets are a formalism for the description of concurrency and synchronization in distributed systems. They consist of *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 [10] that a finite place, finite transition, marked stochastic Petri net is

isomorphic to a 1D discrete space Markov process. For an in-depth discussion of Petri Nets, the reader is referred to [1, 10].

We test MCAMG on the SPN described in [9]. Table 5.5 shows the numerical results. We observe

n	γ	it	C_{op}	γ_{eff}	lev	R_{lump}
1496	0.38	16	4.72	0.82	9	2.28e-2
2470	0.38	16	4.86	0.82	10	2.28e-2
3795	0.38	15	5.17	0.83	10	2.16e-2
10416	0.39	16	5.71	0.85	11	2.09e-2
16206	0.39	16	5.89	0.85	11	2.07e-2
23821	0.39	16	6.11	0.86	12	2.08e-2
33511	0.39	16	6.29	0.86	13	2.08e-2
45526	0.39	16	6.47	0.87	13	2.09e-2

TABLE 5.5
Stochastic Petri net, $\Lambda = (1, 3, 7, 9, 5)$.

that the operator complexity is fairly large for this example and grows somewhat as the problem size increases. However, C_{op} appears to grow very slowly for larger problem sizes and may indeed be bounded above. Furthermore, the iterations are constant and so optimal complexity may yet be achievable.

Figure 5.1 shows the unit circle in the complex plane and the spectrum of B for each test problem.

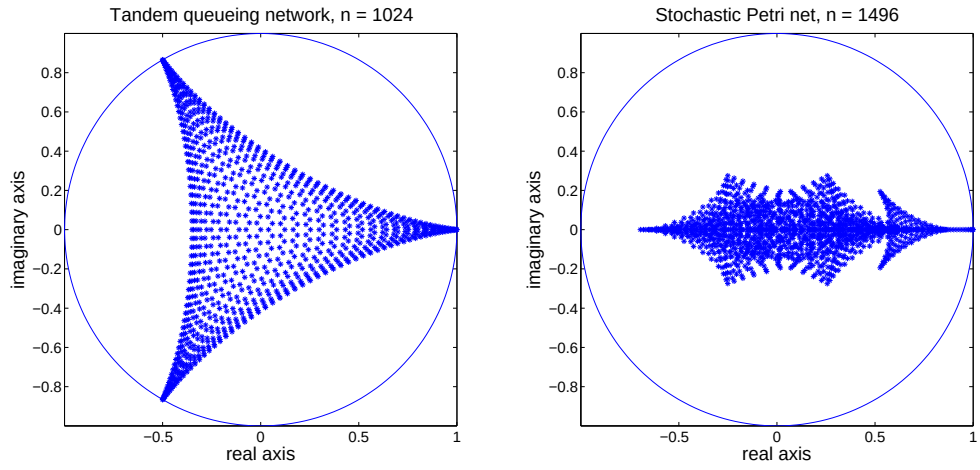

FIG. 5.1. *Complex spectra of* B.

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 with lumping and a modified interpolation formula, 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. Together, these results established the well-posedness 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 nearly optimal multigrid efficiency. We observed that MCAMG was competitive with the algebraic smoothed aggregation multigrid method (A-SAM) from [6], and in most cases it outperformed A-SAM.

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 [13], and use of acceleration on all recursive levels as in K-cycle methods [11].

Acknowledgments. This work was performed as a research project for a Master's degree in Applied Mathematics at the University of Waterloo. I would like to thank Hans De Sterck and Geoff Sanders for their assistance with this paper. I would also like to thank Steve F. McCormick, Thomas A. Manteuffel and John Ruge for interesting and insightful discussions.

REFERENCES

[1] Falko Bause and Pieter S. Kritzinger, *Stochastic Petri Nets*, Verlag Vieweg, Germany, 1996.
[2] Abraham Berman and Robert J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, 1987.
[3] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive algebraic multigrid*, SIAM J. Sci. Comp. 27:1261-1286, 2006.
[4] William L. Briggs, Van Emden Henson, and Stephen F. McCormick, *A Multigrid Tutorial*, SIAM, Philadelphia, 2000.
[5] G. Chartrand and L. Lesniak, *Graphs & Digraphs*, Chapman and Hall/CRC, Boca Raton, 2005.
[6] Hans De Sterck, Thomas A. Manteuffel, Stephen F. McCormick, Killian Miller, James Pearson, John Ruge, and Geoffrey Sanders, *Smoothed aggregation multigrid for Markov chains*, submitted to SIAM J. Sci. Comp., March 2008.
[7] Hans De Sterck, Thomas A. Manteuffel, Stephen F. McCormick, Quoc Nguyen, and John Ruge, *Multilevel adaptive aggregation for Markov chains with application to web ranking*, SIAM J. Sci. Comp. 30:2235-2262, 2008.
[8] Roger A. Horn and Charles R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, 1985.
[9] Graham Horton and S.T. Leutenegger, *A multi-level solution algorithm for steady-state Markov chains*, ACM SIGMETRICS 191-200, 1994.
[10] Michael K. Molloy, *Performance analysis using stochastic Petri nets*, IEEE Transactions on Computers C-31:913-917, 1982.
[11] Yvan Notay and Panayot S. Vassilevski, *Recursive Krylov-based multigrid cycles*, Numer. Linear Algebra Appl. 15:473-487, 2008.
[12] Bernard Philippe, Yousef Saad, and William J. Stewart, *Numerical methods for Markov chain modeling*, Operations Research 40:1156-1179, 1992.
[13] Yousef Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.
[14] William J. Stewart, *An Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, 1994.
[15] Klaus Stueben, *Algebraic multigrid (AMG): an introduction with applications*, GMD Report 70, Institut für Algorithmen und Wissenschaftliches Rechnen, November 1999.
[16] Elena Virnik, *An algebraic multigrid preconditioner for a class of singular M-matrices*, SIAM J. Sci. Comp. 29:1982-1991, 2007.