SIAM J. Sci. Comput.
Vol. 32, No. 1, pp. 40–61

SMOOTHED AGGREGATION MULTIGRID FOR MARKOV CHAINS[*]

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

Abstract. A smoothed aggregation multigrid method is presented for the numerical calculation of the stationary probability vector of an irreducible sparse Markov chain. It is shown how smoothing the interpolation and restriction operators can dramatically increase the efficiency of aggregation multigrid methods for Markov chains that have been proposed in the literature. The proposed smoothing approach is inspired by smoothed aggregation multigrid for linear systems, supplemented with a new lumping technique that assures well-posedness of the coarse-level problems: the coarse-level operators are singular M-matrices on all levels, resulting in strictly positive coarse-level corrections on all levels. Numerical results show how these methods lead to nearly optimal multigrid efficiency for an extensive set of test problems, both when geometric and algebraic aggregation strategies are used.

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

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

DOI. 10.1137/080719157

1. Introduction.

This paper is concerned with multilevel methods for the efficient numerical calculation of the stationary probability vector of large, sparse irreducible Markov matrices [1]. Let $B \in \mathbb{R}^{n\times n}$ be a column-stochastic matrix, i.e., $0 \le b_{ij} \le 1 \ \forall i, j$ and

$$\mathbf{1}^T B = \mathbf{1}^T, \tag{1.1}$$

with $\mathbf{1}$ the column vector of all ones. We want to find a vector $\mathbf{x} \in \mathbb{R}^n$ that satisfies

$$B\,\mathbf{x} = \mathbf{x}, \qquad x_i \ge 0 \ \forall i, \qquad \|\mathbf{x}\|_1 = 1. \tag{1.2}$$

If B is irreducible, then there exists a unique solution $\mathbf{x}$ to (1.2). (Matrix B is irreducible *iff* there exists a path from each vertex i to each vertex j in the directed graph of matrix B.) Moreover, this stationary probability vector $\mathbf{x}$ then satisfies the strict inequality $x_i > 0 \ \forall i$. The eigenvalue of B with largest modulus is $\lambda_1 = 1$, and traditional iterative methods for calculating $\mathbf{x}$ converge very slowly when the subdominant eigenvalue satisfies $|\lambda_2| \sim 1$ [2, 3]. The Markov chain is said to be *slowly mixing* in this case. Multigrid methods aim to accelerate convergence for this type of problem by reducing error components with different scales on progressively coarser levels. Note that problem (1.2) with a reducible matrix B requires special

[*] Received by the editors March 24, 2008; accepted for publication (in revised form) January 28, 2009; published electronically February 5, 2010. This work was sponsored by grants and fellowships from the Natural Sciences and Engineering Research Council of Canada and by an Early Researcher Award from the Province of Ontario. This work was sponsored by the Department of Energy under grants DE-FG02-03ER25574 and DE-FC02-06ER25784, Lawrence Livermore National Laboratory under contract B568677, and the National Science Foundation under grants DMS-0621199, DMS-0749317, and DMS-0811275.

http://www.siam.org/journals/sisc/32-1/71915.html

[†] Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario, Canada (hdesterck@uwaterloo.ca, k7miller@uwaterloo.ca, jrpearso@uwaterloo.ca).

[‡] Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO (tmanteuf@colorado.edu, stevem@colorado.edu, jruge@colorado.edu, sandersg@colorado.edu).

treatment that is beyond the scope of the present work, but we expect to develop methods for this case in future work.

Multigrid methods of aggregation type have been studied for (1.2) in the literature [4, 5, 6]. These methods are multilevel generalizations of the large class of two-level methods of iterative aggregation/disaggregation type for Markov chains (see, e.g., [7, 8, 9, 10, 11, 12, 13, 14, 3]). They are of a multiplicative nature and include both geometric versions, in which aggregates are chosen based on the a priori known topology (or connectivity) of the chain, and algebraic versions, in which aggregates are chosen based on strength of connection in the problem matrix. The latter class is similar to algebraic multigrid (AMG) methods for linear systems [15, 16] and, in particular, to recent adaptive versions of AMG [17, 18, 6, 19]. Two-level methods of iterative aggregation/disaggregation type are effective for so-called nearly completely decomposable Markov chains, which can be decomposed in weakly connected blocks with globally strong connections (and thus fast mixing) inside the blocks. While, for some applications, significant speedup has been reported for Markov chain methods of aggregation type on more than two levels [4, 5, 6], compared to traditional one-level or two-level iterative methods, it appears that true multilevel methods for Markov chains have not been widely used so far, due likely to the fact that their convergence properties are often far from optimal. Indeed, while multigrid methods have been developed for large classes of (nonsingular) linear systems with optimal computational complexity (i.e., number of floating point operations and memory use are linear in the number of unknowns), the general performance of the aggregation-type multigrid methods for Markov chains that are described in the literature is far from optimal. As illustrated below, suboptimal performance of these methods can be observed even for very simple model problems.

In this paper, we show how smoothing the interpolation and restriction operators can dramatically increase the efficiency of aggregation multigrid for Markov chains. The proposed approach is inspired by the so-called smoothed aggregation (SA) multigrid methods for linear systems [20, 21, 17], supplemented with a new lumping technique that assures well-posedness of the coarse-level problems. It is shown numerically that the resulting lumped smoothed aggregation for Markov chains (SAM) methods lead to nearly optimal multigrid efficiency for an extensive set of test problems, both when geometric and algebraic aggregation strategies are used. Our experience with partial differential equation (PDE) problems suggests that our proposed methods will be most effective for the class of Markov chains that have a subdominant eigenvalue that approaches one fast with increasing problem size (slowly mixing Markov chains), have fairly small numbers of inlinks and outlinks per node that are fairly constant over all nodes, are close to symmetric, and effectively have low-dimensional connectivity. For instance, chains on two-dimensional (2D) and three-dimensional (3D) lattices will work well, but higher-dimensional problems are more challenging. We will investigate in future work how much our method in the form presented here extends beyond this class, and which modifications may be required.

Almost all existing multigrid methods for Markov chains use nonoverlapping aggregates. Yet, smoothing the interpolation and restriction operators introduces overlap between the aggregates, which, similar to the case of linear systems, is a crucial enhancement that can significantly improve convergence. Some results have recently been reported for multilevel Markov chain methods that employ the equivalent of overlapping agglomerates, but obtained in a way different from our smoothing approach: an AMG-like approach for building the interpolation matrix of a multilevel preconditioner has been proposed in [22] for a Krylov subspace method applied to Markov chains, and some results have been reported in [23] for a method that uses overlapping aggregates that are determined based on strength of connection. However, to our knowledge, the approach proposed in our paper is the first to use interpolation and restriction operator smoothing for bringing the performance of aggregation multigrid for Markov chains to nearly optimal levels in a general, systematic way.

Large sparse Markov chains are of interest in a wide range of applications, including information retrieval and web ranking [24, 25], performance modeling of computer and communication systems, dependability and security analysis, and analysis of biological systems [3]. Multilevel solvers for Markov problems with improved efficiency thus promise to have significant impact in many disciplines.

This paper is organized as follows. We start in the next section with some background information on an important theoretical concept in the formulation of aggregation multigrid methods for Markov chains, namely, singular M-matrices. In section 3, we recall the general framework of the aggregation-type multigrid methods that have been proposed in the literature. This introduces relevant notation and sets the stage for the introduction of SAM in section 4. Well-posedness of the smoothed method is proven in section 4 as well. Numerical convergence tests are presented for an extensive set of test problems, for the case of geometric aggregation in section 5 and for the case of algebraic aggregation in section 6. Conclusions and future work are discussed in section 7.

2. Singular M-matrices.

An important theoretical concept in the formulation of aggregation multigrid methods for Markov chains is that of singular M-matrices, which can be defined as follows [1].

DEFINITION 2.1 (singular M-matrix). *$A \in \mathbb{R}^{n\times n}$ is a singular M-matrix $\Leftrightarrow$ $\exists\, B \in \mathbb{R}^{n\times n}$, $b_{ij} \geq 0\ \forall i,j : A = \rho(B)\, I - B$.*

Here, $\rho(B)$ is the spectral radius of B. Note that $A = I - B$, with B the column-stochastic matrix from problem (1.2), is a singular M-matrix.

In this paper, we call a real number x *positive* (or *strictly positive*) when $x > 0$, and *nonpositive* when $x \leq 0$, with a similar convention for negative and nonnegative numbers. A nonpositive (nonnegative) vector or matrix is a vector or matrix that has all nonpositive (nonnegative) elements.

The Perron–Frobenius theorem for nonnegative matrices provides important insight about the spectral radius of a nonnegative matrix and its associated eigenvector. The following version is useful for our purposes.

THEOREM 2.2 (Perron–Frobenius [1, pp. 26, 27, 28]). *Let $B \in \mathbb{R}^{n\times n}$, $b_{ij} \geq 0$ $\forall i,j$. Then the following hold:*

(1) *$\rho(B)$ is an eigenvalue of B.*

(2) *$\exists \mathbf{x} \in \mathbb{R}^n$, $x_i \geq 0\ \forall i : B\,\mathbf{x} = \rho(B)\,\mathbf{x}$ and $\exists \mathbf{y} \in \mathbb{R}^n$, $y_i \geq 0\ \forall i : \mathbf{y}^T B = \rho(B)\,\mathbf{y}^T$.*

(3) *If B is irreducible, then the eigenvectors $\mathbf{x}$ and $\mathbf{y}$ in (2) are unique up to scaling, and the inequalities in (2) are strict.*

(4) *If B has a left or right eigenvector with strictly positive components, then this eigenvector has $\rho(B)$ as its eigenvalue.*

Note that the choice of B in Definition 2.1 is not unique. Indeed, the Perron–Frobenius theorem implies that $\rho(B + s\,I) = \rho(B) + s$ for any real $s > 0$. Then $A = \rho(B)\,I - B = (\rho(B) + s)\,I - (B + s\,I) = \rho(B + s\,I)\,I - (B + s\,I)$, which means that $B + s\,I$ can be used instead of B in the above definition.

The following properties of singular M-matrices are important in this paper.

THEOREM 2.3 (some properties of singular M-matrices (see, e.g., [1])).

(1) *Irreducible singular M-matrices have a unique solution to the problem $A\,\mathbf{x} = 0$,*

up to scaling. All components of $\mathbf{x}$ have strictly the same sign (i.e., scaling can be chosen s.t. $x_i > 0\ \forall i$). (This follows directly from the Perron–Frobenius theorem.)

(2) *An equivalent definition for singular M-matrices is the following: $A \in \mathbb{R}^{n\times n}$ is a singular M-matrix $\Leftrightarrow$ A is singular and all elements of $(A+\alpha\, I)^{-1}$ are nonnegative, $\forall\,\alpha > 0$.*

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

(4) *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 (see also* [22]*).*

The theoretical concepts introduced above serve to prove the well-posedness of the proposed smoothed aggregation multigrid methods for Markov chains in section 4 below.

3. Aggregation multigrid for Markov chains (AM). In this section, we recall the general framework of the aggregation-type multigrid methods that have been proposed in the literature for Markov chains [4, 5, 6]. The main difference with traditional multigrid methods for linear systems $A\,\mathbf{x} = \mathbf{f}$ is the multiplicative nature of the aggregation-type methods for Markov chains.

Consider the equation for the stationary probability vector of irreducible Markov matrix B,

$$B\,\mathbf{x} = \mathbf{x}, \tag{3.1}$$

which we rewrite as

$$A\,\mathbf{x} = 0 \tag{3.2}$$

with

$$A = I - B. \tag{3.3}$$

Remember that A is an irreducible singular M-matrix. We can then rewrite the exact solution, $\mathbf{x}$, in terms of the current approximate, $\mathbf{x}_i$, and its multiplicative error, $\mathbf{e}_i$, as $\operatorname{diag}(\mathbf{x}_i)\,\mathbf{e}_i$, obtaining

$$A\,\operatorname{diag}(\mathbf{x}_i)\,\mathbf{e}_i = 0. \tag{3.4}$$

Note that we have to assume here that all components of the current approximate, $\mathbf{x}_i$, are nonzero (the exact solution, $\mathbf{x}$, also has this property). At convergence, the multiplicative error is $\mathbf{e}_i = \mathbf{1}$.

The n fine-level degrees of freedom are aggregated into m groups according to the columns of aggregation matrix $Q \in \mathbb{R}^{n\times m}$, where $q_{ij} = 1$ if fine-level node i belongs to aggregate j and $q_{ij} = 0$ otherwise. For example, if the fine-level degrees of freedom are ordered according to the agglomerates they belong to, then Q has the form

$$Q = \left[\begin{array}{c|c|c|c} 1 & 0 & 0 & \cdots \\ 1 & 0 & 0 & \cdots \\ \hline 0 & 1 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ \hline 0 & 0 & 1 & \cdots \\ 0 & 0 & 1 & \cdots \\ \hline 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{array}\right]. \tag{3.5}$$

In a multilevel setting, the aggregation matrix Q can be determined at successive coarse levels using topological information when the Markov chain is structured [4] (we call this geometric aggregation, in analogy with geometric multigrid). Alternatively, aggregates can be chosen algebraically based on strength of connection in the problem matrix [5, 6] (we call this algebraic aggregation). Experience with AMG and SA for PDE problems suggests that optimal convergence properties can be achieved when aggregates are selected based on strength of connection. We determine strong connections between points by considering relative sizes of entries in each row of scaled problem matrix $A\,\mathrm{diag}(\mathbf{x}_i)$. AMG heuristics show that large entries indicate connections in the direction of slowly varying error, and these connections are employed to generate aggregates that lead to efficient coarse-level error correction. (See section 6 for full details about our strength-based aggregation procedure.)

Once Q has been determined, a coarse-level version of (3.4) is constructed:

$$Q^T A\,\mathrm{diag}(\mathbf{x}_i)\,Q\,\mathbf{e}_c = 0, \tag{3.6}$$

where $\mathbf{e}_c$ represents the coarse-level approximation of unknown fine-level multiplicative error $\mathbf{e}_i$.

Define the restriction and prolongation operators, R and P, by

$$R = Q^T \tag{3.7}$$

and

$$P = \mathrm{diag}(\mathbf{x}_i)\,Q, \tag{3.8}$$

and write

$$R\,A\,P\,\mathbf{e}_c = 0. \tag{3.9}$$

Define the coarse-level operator, A_c, by

$$A_c = R\,A\,P. \tag{3.10}$$

Let the coarse-level vector of all ones be denoted by $\mathbf{1}_c$. It then follows from $\mathbf{1}_c^T\,R = \mathbf{1}^T$ that $\mathbf{1}_c^T\,A_c = 0$.

Note now that $P^T\,\mathbf{1} = R\,\mathbf{x}_i$ is the restriction of current fine-level approximate $\mathbf{x}_i$ to the coarse level. Instead of solving coarse-level equation (3.9) for the multiplicative error, $\mathbf{e}_c$, one can also equivalently seek an improved coarse-level approximation, $\mathbf{x}_c$, of probability vector $\mathbf{x}$. This improved coarse-level approximation $\mathbf{x}_c$ is related to coarse-level error $\mathbf{e}_c$ by

$$\mathbf{x}_c = \mathrm{diag}(P^T\,\mathbf{1})\,\mathbf{e}_c, \tag{3.11}$$

leading to the coarse-level probability equation

$$A_c\,(\mathrm{diag}(P^T\,\mathbf{1}))^{-1}\,\mathbf{x}_c = 0. \tag{3.12}$$

We define coarse-level stochastic matrix B_c as

$$B_c = Q^T B\,\mathrm{diag}(\mathbf{x}_i)\,Q\,(\mathrm{diag}(Q^T\,\mathbf{x}_i))^{-1}. \tag{3.13}$$

This matrix satisfies $\mathbf{1}_c^T B_c = \mathbf{1}_c^T$. We then obtain

$$\begin{aligned} A_c\,(\mathrm{diag}(P^T\,\mathbf{1}))^{-1} &= R\,(I-B)\,P\,(\mathrm{diag}(P^T\,\mathbf{1}))^{-1} \\ &= Q^T \mathrm{diag}(\mathbf{x}_i)\,Q\,(\mathrm{diag}(Q^T\,\mathbf{x}_i))^{-1} - Q^T B\,\mathrm{diag}(\mathbf{x}_i)\,Q\,(\mathrm{diag}(Q^T\,\mathbf{x}_i))^{-1} \\ &= I_c - B_c. \end{aligned} \tag{3.14}$$

Coarse-level equation (3.12) was first introduced in [26] and has a straightforward probabilistic interpretation (see, e.g., [5, 6]). It is well known that (3.12) can be used to accelerate simple one-level iterative methods for (3.2), like the weighted Jacobi or Gauss–Seidel relaxation methods. For example, a two-level numerical method may proceed by relaxation on (3.2) on the fine level, followed by a coarse-level solve of (3.12), a coarse-level correction according to

$$\mathbf{x}_{i+1} = P\,(\mathrm{diag}(P^T\,\mathbf{1}))^{-1}\,\mathbf{x}_c = P\,\mathbf{e}_c, \tag{3.15}$$

and another relaxation on the fine level. Some theoretical convergence results for this two-level method have been derived in [11, 12, 14].

In this paper, we use the weighted Jacobi method for all relaxation operations. We split problem matrix A into its diagonal and lower and upper triangular parts as

$$A = D - (L+U), \tag{3.16}$$

using standard notation. Note that, since A is an irreducible singular M-matrix, L and U are nonnegative, and D has strictly positive elements on the diagonal. Weighted Jacobi relaxation with weight $w \in (0,1)$ is then given by

$$\mathbf{x} \leftarrow (1-w)\,\mathbf{x} + w\,D^{-1}\,(L+U)\,\mathbf{x}. \tag{3.17}$$

Note that D^{-1} exists because $d_{ii} > 0\ \forall i$. Also, Jacobi relaxation maintains strict positivity of the components of $\mathbf{x}$, because $(1-w)\,\mathbf{x}$ is strictly positive and $w\,D^{-1}\,(L+U)\,\mathbf{x}$ is nonnegative ($w \in (0,1)$).

A multilevel method can then be obtained by recursively applying the two-level method to coarse-level equation (3.12).

Algorithm AM($A, \mathbf{x}, \nu_1, \nu_2$). *Aggregation multigrid for Markov chains (V-cycle).*

```
if not on coarsest level
    $\mathbf{x} \leftarrow$ Relax($A, \mathbf{x}$)   $\nu_1$ times
    build $Q$
    $R = Q^T$ and $P = \mathrm{diag}(\mathbf{x})\,Q$
    $A_c = R\,A\,P$
    $\mathbf{x}_c \leftarrow$ AM($A_c\,\mathrm{diag}(P^T\,\mathbf{1})^{-1}, P^T\,\mathbf{1}, \nu_1, \nu_2$)   (coarse-level solve)
    $\mathbf{x} \leftarrow P\,(\mathrm{diag}(P^T\,\mathbf{1}))^{-1}\,\mathbf{x}_c$   (coarse-level correction)
    $\mathbf{x} \leftarrow$ Relax($A, \mathbf{x}$)   $\nu_2$ times
else
    $\mathbf{x} \leftarrow$ direct solve of $A\,\mathbf{x} = 0$
end
```

This algorithm uses the simplest type of recursion, resulting in a so-called V-cycle.

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. Well-posedness of the algorithm follows from the first of the following two theorems (see [4, 3, 11, 12]), and the second theorem is required for convergence of the method.

Theorem 3.1 (singular M-matrix property of AM coarse-level operators). *A_c is an irreducible singular M-matrix on all coarse levels, and thus has a unique right kernel vector $\mathbf{e}_c$ with strictly positive components (up to scaling) on all levels.*

Proof. Equation (3.14) shows that A_c has nonpositive off-diagonal elements, and $\mathbf{1}_c^T A_c = 0$ because $\mathbf{1}_c^T R = \mathbf{1}^T$. This implies that A_c is a singular M-matrix, due to Theorem 2.3(4). Irreducibility of A_c can be proved as follows. Let fine-level node i belong to aggregate I, and fine-level node j to aggregate J. Then a link exists from J to I ($(Q^T B \operatorname{diag}(\mathbf{x}_k)\, Q)_{IJ} \neq 0$) if a link exists from j to i ($b_{ij} \neq 0$), by virtue of the shape of Q (equation (3.5)) and the strict positivity of the components of $\mathbf{x}_k$. This implies that every aggregate J is connected to every aggregate I via a directed path, because every $i \in I$ is connected to every $j \in J$ via a directed path due to A's irreducibility. The second part of the theorem then follows directly from Theorem 2.3(1). ▯

Theorem 3.2 (fixed-point property of AM V-cycle). *Exact solution $\mathbf{x}$ is a fixed point of the AM V-cycle.*

Proof. It is easy to see that $\mathbf{e}_c = \mathbf{1}_c$ is a solution of coarse-level equation (3.9) for $\mathbf{x}_i = \mathbf{x}$: $R\,A\,P\,\mathbf{e}_c = R\,A\,P\,\mathbf{1}_c = Q^T A \operatorname{diag}(\mathbf{x})\, Q\, \mathbf{1}_c = Q^T A \operatorname{diag}(\mathbf{x})\, \mathbf{1} = Q^T A\, \mathbf{x} = 0$. This solution is unique (up to scaling) because A_c is an irreducible M-matrix. Coarse-level correction equation (3.15) then gives $\mathbf{x}_{i+1} = P\,\mathbf{e}_c = \operatorname{diag}(\mathbf{x})\, Q\, \mathbf{1}_c = \operatorname{diag}(\mathbf{x})\, \mathbf{1} = \mathbf{x}$. ▯

In the numerical results presented below, we sometimes also use so-called W-cycles, which are identical to the V-cycles described above, except that each occurrence of the recursive coarse-level solve and coarse-level correction is duplicated (resulting in two coarse-level solves at the first level, four coarse-level solves at the second level, eight at the third level, and so on). While W-cycles are more expensive than V-cycles, they can retain computational complexity that is linear in the number of unknowns, as long as coarsening is sufficiently fast.

4. Smoothed aggregation for Markov chains (SAM).

As shown in the numerical results of sections 5 and 6, performance of the aggregation multigrid method for Markov chains described in the previous section is often far from optimal. Inspired by smoothed aggregation (SA) multigrid methods for linear systems [20, 21, 17], we propose smoothing the interpolation and restriction operators, P and R, to improve the convergence properties of the multilevel aggregation method. The main difficulty is that smoothing may destroy the singular M-matrix nature of the coarse-level operators, thus potentially compromising the well-posedness of the method, so we then use a lumping process to overcome this difficulty.

We smooth the rows of the interpolation operator, P, with A, using weighted Jacobi with weight w:

$$\begin{aligned} P_s &= (1-w)\,P + w\,D^{-1}\,(L+U)\,P \\ &= (1-w)\operatorname{diag}(\mathbf{x}_i)\,Q + w\,D^{-1}\,(L+U)\operatorname{diag}(\mathbf{x}_i)\,Q \\ &= (I - w\,D^{-1}A)\operatorname{diag}(\mathbf{x}_i)\,Q. \end{aligned} \tag{4.1}$$

For the restriction operator, we smooth the rows of R with A:

$$\begin{aligned} R_s &= R\,(1-w) + R\,w\,(L+U)\,D^{-1} \\ &= Q^T\,(I - w\,A\,D^{-1}). \end{aligned} \tag{4.2}$$

Note that the resulting smoothed operators, R_s and P_s, are nonnegative and of full rank. Note also that $\mathbf{1}_c^T R_s = \mathbf{1}^T$ for any $\mathbf{x}_i$, and $P_s\,\mathbf{1}_c = \mathbf{x}$ for $\mathbf{x}_i = \mathbf{x}$.

This particular choice for the smoothed P and R can be motivated heuristically as follows. The left kernel vector of A is known: it is the vector with all ones, $\mathbf{1}^T$. The right kernel vector of A is the unknown stationary probability vector, $\mathbf{x}$. For the multilevel method to be efficient, the range of interpolation should provide a good approximation for the solution, $\mathbf{x}$. At convergence, the range of $P = \mathrm{diag}(\mathbf{x})\, Q$ exactly contains $\mathbf{x}$, but the problem is that, during the iteration process, corrections may produce sharp jumps between aggregates, because the interpolation in Q is piecewise constant. These sharp jumps between aggregates may reintroduce errors with high frequencies in the new fine-level approximation, and this may lead to slow convergence. Smoothing the piecewise defined columns of P using operator A may provide a remedy. Indeed, smoothing the columns of P with A leaves the interior parts of the aggregates largely unaffected ($\mathbf{x}_i$ is close to the kernel of A), while the jumps at aggregate boundaries are smoothed out. This results in $\mathbf{x}_i$ still approximately being in the range of P, while the high-frequency errors related to the jumps are smoothed out. In fact, the range of the smoothed P contains the image of $\mathbf{x}_i$ under the smoothing operator, which can only improve the approximation to the right kernel. See [20, 21, 17] for formal theoretical justification of smoothed aggregation in the context of symmetric linear systems $A\,\mathbf{x} = \mathbf{f}$. Note that, in our case, smoothing must be applied to the columns of $P = \mathrm{diag}(\mathbf{x})\, Q$, as opposed to the columns of Q, because that would adversely affect the interiors of the aggregates ($\mathbf{1}$ is generally not in the kernel of A) and move $\mathbf{x}$ away from the range of the smoothed P. In a similar way, $\mathbf{1}^T$ is the left kernel vector of A, and A is applied to the rows of Q^T, from the right. Note that this way of smoothing restriction and interpolation separately leads to a method in which the smoothed restriction operator, R_s, and the smoothed interpolation operator for the multiplicative error, $\mathrm{diag}(\mathbf{x}_i)^{-1}\, P_s$, are not related by a transpose operation (they are in the unsmoothed case). While there are good reasons (of a variational nature) to have them related by a transpose for the case of symmetric matrices, there does not appear to be a compelling reason for this in the case of nonsymmetric matrices. In fact, it appears more important to preserve the local nature of the left and right kernel vectors when smoothing. We have confirmed this in extensive numerical tests, in which the above-described smoothing approach performs better than the possible alternatives, like smoothing Q instead of P, or smoothing one of R and P and forming the other smoothed operator such that $\mathrm{diag}(\mathbf{x}_i)^{-1}\, P_s$ is the transpose of R_s.

As above, we can proceed to the coarse-level equation by writing

$$R_s\, A\, P_s\, \mathbf{e}_c = 0 \tag{4.3}$$

and define the smoothed coarse-level operator by

$$A_{cs} = R_s\, A\, P_s. \tag{4.4}$$

Coarse-level operator A_{cs} enjoys the following two properties on all levels:

$$\mathbf{1}_c^T\, A_{cs} = 0 \quad \forall\, \mathbf{x}_i, \tag{4.5}$$

$$A_{cs}\, \mathbf{1}_c = 0 \quad \text{for } \mathbf{x}_\mathrm{i} = \mathbf{x}. \tag{4.6}$$

The former property follows from $\mathbf{1}_c^T\, R_s = \mathbf{1}^T$ (for any $\mathbf{x}_i$), and the latter from $P_s\, \mathbf{1}_c = \mathbf{x}$ (for $\mathbf{x}_i = \mathbf{x}$). The coarse-level probability equation is given by

$$A_{cs}\, (\mathrm{diag}(P_s^T\, \mathbf{1}))^{-1}\, \mathbf{x}_c = 0, \tag{4.7}$$

and the V-cycle follows as above.

There is, however, a significant problem with the well-posedness of the resulting V-cycle. Coarse-level operator A_{cs} is singular on all levels, but strict positivity of all components of $\mathbf{e}_c$ can no longer be guaranteed, because A_{cs} is not an irreducible singular M-matrix in general. To see this, note that

$$\begin{aligned} A_{cs} &= R_s\,(D-(L+U))\,P_s \\ &= R_s\,D\,P_s - R_s\,(L+U)\,P_s. \end{aligned} \tag{4.8}$$

Both $R_s\,D\,P_s$ and $R_s\,(L+U)\,P_s$ are nonnegative matrices because A is a singular M-matrix. Matrix $R_s\,D\,P_s$ is, in general, not diagonal because of the smoothing of R_s and P_s, and A_{cs} may have strictly positive off-diagonal elements as a result, which means that A_{cs} may not be a singular M-matrix. Note that A_{cs} may also lose irreducibility due to new zero elements being created. Thus, strict positivity of the components of $\mathbf{e}_c$ cannot be guaranteed. Numerical experiments confirm that the resulting multilevel method leads to erratic convergence behavior or divergence for most problems.

It is possible, however, to modify the term $R_s\,D\,P_s$ in A_{cs}, such that the modified coarse-level operator is an irreducible singular M-matrix. For convenience, define $S = R_s\,D\,P_s$ and $G = R_s\,(L+U)\,P_s$, and write

$$A_{cs} = S - G. \tag{4.9}$$

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

$$\hat{A}_{cs} = \hat{S} - G. \tag{4.10}$$

Our goal is to modify S in such a way that $\hat{A}_{cs}$ has nonpositive off-diagonal elements, but retains nonzero off-diagonal elements where G has them (to guarantee irreducibility). While modifying S, we also want to conserve properties (4.5) and (4.6) of A_{cs}:

$$\mathbf{1}_c^T\,\hat{A}_{cs} = 0 \quad \forall\,\mathbf{x}_i, \tag{4.11}$$

$$\hat{A}_{cs}\,\mathbf{1}_c = 0 \quad \text{for } \mathbf{x}_\mathrm{i} = \mathbf{x}. \tag{4.12}$$

The first of these properties implies well-posedness of the resulting multilevel method: it guarantees that $\hat{A}_{cs}$ is a singular M-matrix when its off-diagonal elements are nonpositive (see Theorem 4.1 below). The second assures that the exact solution, $\mathbf{x}$, is a fixed point of the multigrid cycle, as required for convergence (see Theorem 4.2).

We proceed as follows. Considering all off-diagonal index pairs (i,j) for which S is nonzero, we select the *offending* pairs of indices that have nonnegative elements in A_{cs}. Lumping is done for these pairs. Suppose (i,j) is an offending index pair. To correct the sign of A_{cs} 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. We choose $\beta_{\{i,j\}}$ so that

$$\begin{aligned} s_{ij} - g_{ij} - \beta_{\{i,j\}} &< 0, \\ s_{ji} - g_{ji} - \beta_{\{i,j\}} &< 0, \end{aligned} \tag{4.13}$$

resulting in strictly negative off-diagonal elements in locations (i,j) and (j,i). We explain below how we choose $\beta_{\{i,j\}}$ in practice. Adding a matrix of this form to S

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 perform lumping using symmetric matrices $S_{\{i,j\}}$ because they conserve the row sums and column sums of A_{cs}, and thus conserve the crucial properties (4.5) and (4.6). Note that adding this matrix 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. (Note that S and G are generally not symmetric.) Note also that this procedure may generate new nonzero off-diagonal elements in $\hat{A}_{cs}$, but that $\hat{A}_{cs}$ cannot have zero off-diagonal elements at locations that have nonzero elements in G.

As expected, numerical tests show that it is advantageous to lump as little of S as possible: the factors $\beta_{\{i,j\}}$ in $S_{\{i,j\}}$ should be chosen small. This is so because lumping may adversely affect convergence properties of the multilevel method. We choose $\beta_{\{i,j\}} = \max(\beta^{(1)}_{\{i,j\}}, \beta^{(2)}_{\{i,j\}})$, with

$$\begin{aligned} s_{ij} - g_{ij} - \beta^{(1)}_{\{i,j\}} &= -\eta\, g_{ij}, \\ s_{ji} - g_{ji} - \beta^{(2)}_{\{i,j\}} &= -\eta\, g_{ji}, \end{aligned} \tag{4.14}$$

and η a fixed parameter $\in (0,1]$. Note that $\beta_{\{i,j\}} > 0$ always holds. Choosing $\eta = 1$ corresponds to lumping the full value of the largest of s_{ij} and s_{ji} to the diagonal. For $\eta < 1$, a smaller value is lumped. Parameter η has to be taken strictly larger than zero, because $\eta = 0$ could introduce new zero elements in $\hat{A}_{cs}$ that may lead to reducibility. In our experience, choosing η close to zero results in the best multilevel convergence properties for the smoothed method. We chose $\eta = 0.01$ for the numerical results reported below. (In tests for our most simple test problems, we found that choosing η smaller than 0.01 did not improve performance further.) Our numerical experiments also indicate that the number of offending elements at any level is normally only a small fraction of the total number of nonzero elements in A_{cs}.

We now prove the following two theorems on the well-posedness and fixed-point property of the smoothed aggregation for Markov chains (SAM) algorithm with lumped coarse-level operators $\hat{A}_{cs}$.

THEOREM 4.1 (singular M-matrix property of lumped SAM coarse-level operators). *$\hat{A}_{cs}$ is an irreducible singular M-matrix on all coarse levels, and thus has a unique right kernel vector $\mathbf{e}_c$ with strictly positive components (up to scaling) on all levels.*

Proof. Lumped coarse-level operator $\hat{A}_{cs}$ has nonpositive off-diagonal elements, and $\mathbf{1}_c^T\,\hat{A}_{cs} = 0$, i.e., (4.11), holds. This implies that $\hat{A}_{cs}$ is a singular M-matrix, due to Theorem 2.3(4). Irreducibility of $\hat{A}_{cs}$ follows as in the proof of Theorem 3.1 (because $\hat{A}_{cs}$ has nonzeros where G has nonzeros), and the second part of the theorem follows directly from Theorem 2.3(1). □

THEOREM 4.2 (fixed-point property of lumped SAM V-cycle). *Exact solution $\mathbf{x}$ is a fixed point of the SAM V-cycle (with lumping).*

Proof. Property (4.12) implies that $\mathbf{e}_c = \mathbf{1}_c$ is a solution of coarse-level equation $\hat{A}_{cs}\,\mathbf{e}_c = 0$ for $\mathbf{x}_i = \mathbf{x}$. Note that this solution is unique (up to scaling) because $\hat{A}_{cs}$ is an irreducible M-matrix. The coarse-level correction equation then gives $\mathbf{x}_{i+1} = P_s\,\mathbf{e}_c = P_s\,\mathbf{1}_c = \mathbf{x}$. □

5. Numerical results: Geometric aggregation.

In this section, we report on numerical convergence results for SAM using a geometric aggregation process, in which aggregates are chosen based on the a priori known topology of the chain. Algebraic aggregation is considered in the next section. We compare the results of the unsmoothed aggregation algorithm and the new smoothed aggregation algorithm with lumping for a representative suite of test problems. Some of these test problems have also been considered in [23]. We use the shorthand notation G-AM (geometric aggregation for Markov chains) for the unsmoothed algorithm and G-SAM (geometric smoothed aggregation for Markov chains) for the smoothed algorithm with lumping. Results for six test problems are discussed, all of which have the subdominant eigenvalue approaching one as the number of states is increased, so traditional one-level or two-level iterative methods are ineffective.

5.1. Uniform chain. The first three test problems are 1D Markov chains generated by linear graphs with weighted edges. Figure 5.1 shows the graph for the simplest case, in which all weights are equal; this constitutes our first test problem. The weights determine the transition probabilities: 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. (Note that the edge weights can also be interpreted as transition rates in continuous time.) For this example, the transition probabilities from interior nodes are $\frac{1}{2}$, and they equal 1 from the end nodes.

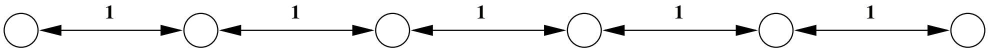

FIG. 5.1. *Graph for uniform chain test problem. The Markov chain is generated by a linear graph with weighted edges. The weights determine the transition probabilities. In this uniform case, all weights are equal. The transition probabilities from interior nodes are thus* $\frac{1}{2}$*, and they equal* 1 *from the end nodes.*

Tables 5.1 and 5.2 show numerical convergence results for G-AM and G-SAM. For the 1D problems in this section, aggregates of size three are chosen. In the tables, “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). 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 do a direct solve on the coarse level when $n < 12$. All multilevel cycles used are $(1, 1)$ cycles, with one prerelaxation and one postrelaxation on each level. A scalable (or optimal) method requires γ to be 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. For simplicity, the weight in our weighted Jacobi relaxation scheme is always chosen as 0.7. This value works well for all tests considered, but, if desired, convergence factors can be further reduced by choosing problem-dependent optimal weights. “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. Note that, for the case of W-cycles, all operators on each level are counted in C_{op} (i.e., two operators on level two, four on level three, eight on level four, etc.). We also provide an effective convergence factor, defined as $\gamma_{\text{eff}} = \gamma^{1/C_{\text{op}}}$. This effective convergence factor takes work into account and makes it easier to compare the overall efficiency of different methods applied to a problem. 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 contributes extra work starting only from the second level.

Table 5.1

Uniform chain. G-AM with V-cycles (left) and W-cycles (right). (Size-three aggregates, no smoothing.)

n	lev	it	C_{op}	γ	γ_{eff}	it	C_{op}	γ	γ_{eff}
27	2	32	1.32	0.66	0.73	32	1.63	0.66	0.78
81	3	85	1.43	0.87	0.91	52	2.07	0.80	0.90
243	4	>100	1.47	0.95	0.97	73	2.37	0.87	0.94
729	5	>100	1.49	0.98	0.98	>100	2.58	0.92	0.97
2187	6	>100	1.50	0.98	0.98	>100	2.72	0.95	0.98

Table 5.2

Uniform chain. G-SAM with V-cycles and size-three aggregates. (Smoothing with lumping.)

n	it	C_{op}	lev	γ	γ_{eff}	R_{lump}
27	13	1.32	2	0.27	0.37	0
243	13	1.47	4	0.27	0.41	0
2187	13	1.50	6	0.27	0.42	0
19683	13	1.50	8	0.27	0.42	0
59049	13	1.50	9	0.27	0.42	0

Table 5.1 shows that G-AM, the unsmoothed aggregation approach which is similar to methods described by various authors in the literature, is not effective for this problem: The operator complexity, C_{op}, appears bounded by a small constant, but the convergence factor, γ, appears to approach one, and is close to one even for relatively small n, resulting in an excessive number of iterations that grows as a function of n. W-cycles do not significantly improve the convergence of the V-cycles.

Table 5.2 shows that G-SAM 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. Note that C_{op} is not larger than for the unsmoothed method. This is because the smoothed coarse-level operators remain tridiagonal matrices when aggregates of size three are chosen for this simple, highly structured 1D test problem. This is, in fact, the reason for choosing aggregates of size three. Size-two aggregates lead to large operator complexity, and size-four aggregates lead to poor convergence factors. Size-three aggregates thus give the best overall results. Note that no lumping is required for this highly regular 1D problem.

Table 5.3 illustrates the superiority of the multilevel approach over a one-level and a two-level method. In these tests, one prerelaxation and one postrelaxation are performed on all levels, and two relaxations are also performed on the coarsest level, instead of a direct solve. It is clear that one-level relaxation and two-level aggregation (similar to aggregation/disaggregation methods) fail to improve convergence for this model problem.

Table 5.3

Uniform chain with 729 nodes. Comparison of true multilevel G-SAM V-cycles (bottom line) with one-level, two-level, etc., smoothed V-cycles. One prerelaxation and one postrelaxation are performed on all levels, and two relaxations are also performed on the coarsest level, instead of a direct solve. (Size-three aggregates, smoothing with lumping.)

Number of levels used	it	C_{op}	γ	γ_{eff}	R_{lump}
1	>100	1.00	0.99	0.99	0
2	>100	1.33	0.99	0.99	0
3	>100	1.44	0.98	0.99	0
4	>100	1.48	0.99	0.99	0
5	22	1.49	0.88	0.92	0
6	13	1.49	0.27	0.41	0

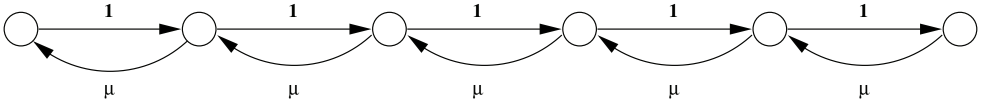

Fig. 5.2. *Graph for birth-death chain test problem. The weights of the edges in the graph determine the transition probabilities. The transition probabilities from interior nodes are $\frac{1}{1+\mu}$ for transitions to the right (birth of a new population member), and $\frac{\mu}{1+\mu}$ for transitions to the left (death of a population member).*

Table 5.4

Birth-death chain ($\mu = 0.96$). (Left) G-AM (no smoothing) and (right) G-SAM (smoothing with lumping). (V-cycles and size-three aggregates.)

n	C_{op}	lev	it	γ	γ_{eff}	it	γ	γ_{eff}	R_{lump}
27	1.32	2	33	0.66	0.73	13	0.27	0.37	0
81	1.43	3	95	0.88	0.92	12	0.27	0.40	0
243	1.47	4	>100	0.95	0.97	13	0.26	0.40	0
729	1.49	5	>100	0.97	0.98	12	0.24	0.38	0

5.2. Birth-death chain. The next test problem is a birth-death chain with constant birth and death rates (Figure 5.2). Birth-death chains are used in queueing theory models. Table 5.4, for a birth-death chain with $\mu = 0.96$, shows that the smoothing of G-SAM dramatically improves efficiency of the multilevel algorithm, leading to optimal computational complexity. Note that choosing μ not close to one gives solutions with extremely small probabilities toward one side of the chain, of the order of μ^n. These small values are hard to compute due to finite precision number representation. For this reason, $\mu = 0.96$ was chosen for our numerical tests. Note again that no lumping is required for this problem.

5.3. Uniform chain with two weak links. The next test problem is a 1D chain with uniform weights, except for two weak links with weight ϵ in the middle of the chain (Figure 5.3). G-AM converges very slowly for this problem, as in Table 5.1. Table 5.5 shows convergence results of G-SAM when the aggregates are chosen such that the weak links occur between two aggregates at each level. The smoothed method shows optimal convergence, and no lumping is required for this problem.

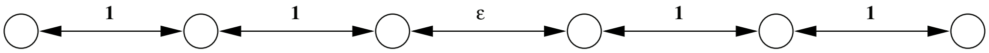

Fig. 5.3. *Graph for uniform chain with two weak links in the middle.*

Table 5.5
Uniform chain with two weak links ($\epsilon = 0.001$). G-SAM with V-cycles and size-three aggregates. The two weak links occur between aggregates at all levels. (Smoothing with lumping.)

n	it	C_{op}	lev	γ	γ_{eff}	R_{lump}
54	12	1.43	3	0.26	0.39	0
486	13	1.49	5	0.27	0.41	0
4374	13	1.50	7	0.27	0.42	0
39366	13	1.50	9	0.27	0.42	0

Table 5.6
Uniform chain with two weak links ($\epsilon = 0.001$). G-SAM with V-cycles and size-three aggregates. The two weak links occur inside an aggregate on the finest level. (Smoothing with lumping.)

n	it	C_{op}	lev	γ	γ_{eff}	R_{lump}
27	>100	1.32	2	1.00	1.00	0
81	>100	1.43	3	1.00	1.00	0
243	>100	1.47	4	1.00	1.00	0
729	>100	1.49	5	1.00	1.00	0

However, Table 5.6 shows that when the aggregates are chosen such that the weak links occur within aggregates, G-SAM fails. This can be explained as follows. When the weak links are contained within an aggregate, the differences in the error in the states joined by the weak links cannot be balanced out efficiently by relaxation because the link (and matrix element) between those states is weak compared to the other links. The error differences also cannot be smoothed out efficiently by the multiplicative coarse-level correction, which tends to correct states in the interior of an aggregate with the same multiplicative factor. This example is important because it shows that smoothing cannot remedy a bad choice of aggregates: aggregates have to be chosen carefully such that no states are aggregated together that are locally weakly connected. While it is not difficult to geometrically choose aggregates in this way for mostly uniform, structured problems, it is much harder to do this for less uniform and unstructured problems, especially because the operators change on coarse levels, and strength of connection has to be taken into account in forming the aggregates on all recursive levels. This is the main motivation for an algebraic choice of aggregates based on strength in the problem matrix, which is discussed in section 6.

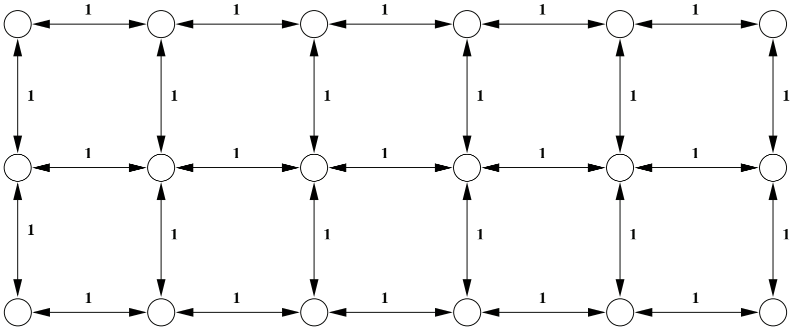

Fig. 5.4. *Graph for 2D lattice with uniform weights.*

5.4. Uniform 2D lattice. The next problem is a 2D lattice with uniform weights (Figure 5.4). Aggregates of size three-by-three are used. Tables 5.7 and 5.8

Table 5.7

Uniform 2D lattice. G-AM with V-cycles (left) and W-cycles (right). (Three-by-three aggregates, no smoothing.)

n	lev	it	C_{op}	γ	γ_{eff}	it	C_{op}	γ	γ_{eff}
64	2	43	1.11	0.71	0.73	43	1.23	0.71	0.75
256	3	83	1.14	0.87	0.88	60	1.30	0.81	0.85
1024	4	>100	1.13	0.95	0.95	81	1.30	0.88	0.91
4096	4	>100	1.13	0.95	0.96	92	1.30	0.89	0.92
16384	5	>100	1.13	0.96	0.97	>100	1.29	0.92	0.94

Table 5.8

Uniform 2D lattice. G-SAM with V-cycles (left) and W-cycles (right), using three-by-three aggregates. (Smoothing with lumping.)

n	lev	it	C_{op}	γ	γ_{eff}	R_{lump}	it	C_{op}	γ	γ_{eff}	R_{lump}
64	2	22	1.17	0.49	0.55	0	17	1.34	0.39	0.49	0
1024	4	25	1.22	0.57	0.63	6.6e-4	18	1.49	0.44	0.58	4.3e-3
16384	5	27	1.22	0.61	0.67	4.0e-5	19	1.51	0.45	0.59	1.3e-4
65536	6	27	1.23	0.62	0.67	1.5e-5	19	1.52	0.45	0.59	2.6e-4

show again that G-SAM improves the convergence properties of G-AM in a way similar to that for the 1D problems. G-AM W-cycles do not sufficiently improve efficiency. While the convergence rates are admittedly not as low as for additive multigrid applied to isotropic elliptic PDE problems, the improvement over the unsmoothed method is very significant, especially because the smoothed method appears scalable. Note also that the operator complexity remains quite low. G-SAM W-cycles appear somewhat more effective than V-cycles. Very small amounts of lumping are applied for some problem sizes.

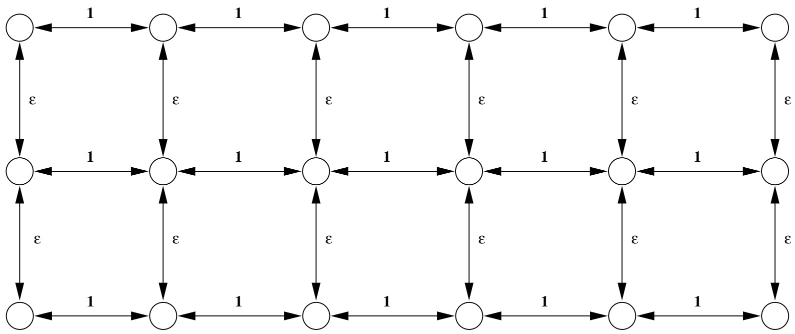

Fig. 5.5. *Graph for 2D lattice with anisotropic weights.*

5.5. Anisotropic 2D lattice. The next test problem is an anisotropic 2D lattice (Figure 5.5). The results in Table 5.9 show that straightforward smoothing does not improve convergence in this case. This is completely analogous to the case of elliptic PDEs: the anisotropy is a real difficulty, because error components with high frequencies across weak connections cannot be reduced effectively by relaxation, nor by coarse-level correction on three-by-three aggregates. It is well known that modifications of geometrical multigrid like semicoarsening or line relaxation can remedy this shortcoming [27]. Rather than pursuing this remedy here, we refer the reader to section 6.1, where it is shown that strength-based algebraic aggregation with smoothing

Table 5.9
Anisotropic 2D lattice ($\epsilon = 1e{-}6$). (Left) G-AM (no smoothing) and (right) G-SAM (smoothing with lumping) with V-cycles and three-by-three aggregates.

n	lev	it	C_{op}	γ	γ_{eff}	it	C_{op}	γ	γ_{eff}	R_{lump}
64	2	>100	1.11	1.00	1.00	>100	1.17	1.00	1.00	0
256	3	>100	1.14	0.97	0.97	>100	1.22	0.95	0.96	0
1024	4	>100	1.13	0.99	0.99	>100	1.22	0.99	0.99	0
4096	4	>100	1.13	0.99	0.99	>100	1.23	0.99	0.99	0

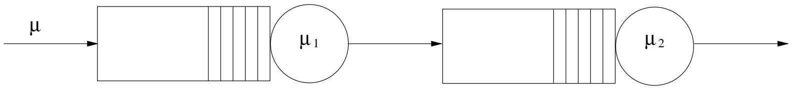

Fig. 5.6. *Tandem queueing network.*

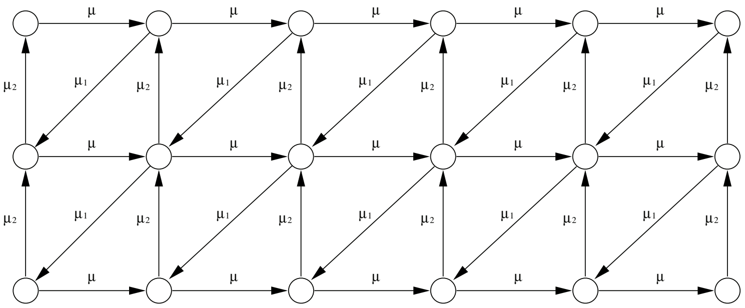

Fig. 5.7. *Graph for tandem queueing network.*

Table 5.10
Tandem queueing network. G-AM with V-cycles and three-by-three aggregates. (No smoothing.)

n	it	C_{op}	lev	γ	γ_{eff}
256	>100	1.19	3	0.91	0.93
1024	>100	1.19	4	0.96	0.97
4096	>100	1.19	4	0.96	0.97

leads to good results for this problem, because aggregates are chosen automatically in a way that allows for efficient relaxation and coarse-level correction (see also [16, 27] for further explanation in the PDE context).

5.6. Tandem queueing network. The final test problem is an open tandem queueing network from [3, p. 55] (Figure 5.6). 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 $N = 15, 31, 63, 127, 255$. We choose $\mu = 10$, $\mu_1 = 11$, and $\mu_2 = 10$. 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, and transition rates are as indicated on Figure 5.7. In this 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. The results in Tables 5.10 and 5.11 show again that smoothing improves efficiency dramatically, and

TABLE 5.11

Tandem queueing network. G-SAM with V-cycles (left) and W-cycles (right), using three-by-three aggregates. (Smoothing with lumping.)

n	lev	it	C_{op}	γ	γ_{eff}	R_{lump}	it	C_{op}	γ	γ_{eff}	R_{lump}
256	3	19	1.28	0.44	0.52	2.4e-2	14	1.60	0.33	0.50	3.9e-2
4096	4	20	1.29	0.49	0.57	7.4e-3	14	1.65	0.33	0.51	1.4e-2
16384	5	21	1.28	0.56	0.63	8.1e-4	14	1.63	0.33	0.51	2.0e-3
65536	6	21	1.28	0.51	0.59	1.6e-3	14	1.64	0.33	0.51	3.8e-3

that W-cycles are somewhat more efficient than V-cycles. Small amounts of lumping are applied for this nonsymmetric 2D problem.

6. Numerical results: Algebraic aggregation.

In this section, we show performance results for the unsmoothed and smoothed aggregation methods with algebraic aggregation. Simulation parameters for the numerical results presented in this section are the same as for the geometric aggregation results presented in section 5; see section 5.1. We use the shorthand notation A-AM (algebraic aggregation for Markov chains) for the unsmoothed algorithm and A-SAM (algebraic smoothed aggregation for Markov chains) for the smoothed algorithm with lumping. (Note that the unsmoothed algebraic method, A-AM, is identical to the method described in [6], where we used the name MAA.) In these algebraic methods, aggregation is based on the sizes of entries in the problem matrix. As shown in [6], it is crucial to base strength on the sizes of the entries in the scaled problem matrix, $A\,\mathrm{diag}(\mathbf{x}_i)$, at all levels, because the matrix elements of the unscaled A may give a false indication of strength. We use the aggregation strategy from [6]: denoting the scaled matrix with matrix elements $\bar{a}_{jk}$ by

$$\bar{A} = A\,\mathrm{diag}(\mathbf{x}_i), \tag{6.1}$$

we base the aggregation procedure on a strength matrix, S, defined as follows:

$$S_{jk} = \begin{cases} 1 & \text{if } \; j \neq k \text{ and } \; -\bar{a}_{jk} \geq \theta \max\limits_{l \neq j} \left(-\bar{a}_{jl}\right), \\ 0 & \qquad\qquad \text{otherwise,} \end{cases} \tag{6.2}$$

where $\theta \in [0,1]$ is a strength threshold parameter. In this paper, we choose $\theta = 0.25$.

After strength matrix S has been determined, the aggregates are formed by the following procedure.

ALGORITHM. *Aggregation based on strength matrix S.*

```
repeat
    • among the unassigned states, choose state $j$ which has the largest value
    in current iterate $\mathbf{x}_i$ as the seed point of a new aggregate
    • add all unassigned states $k$ that are strongly influenced by seed point $j$
    (i.e., $S_{kj} = 1$) to the new aggregate
until all states are assigned to aggregates
```

See [6] for further details and motivation for this aggregation strategy. Note that we recompute aggregates at every level of every V-cycle. This is especially important early on in the iteration process, because aggregates are calculated using the scaled problem matrix which depends on the current iterate. In principle, the aggregates can be "frozen" after a few cycles to save some work, but this is not done for the scaling results presented in this paper.

Distance-one aggregation in the results below denotes aggregation according to the algorithm above, which takes a seed point and its strongly influenced unassigned neighbors to form an aggregate. We also employ distance-two aggregation, which takes, in addition, all strongly influenced unassigned neighbors of the strong neighbors initially selected for the aggregate and makes them part of the aggregate as well. We can similarly extend the method to distance-three aggregation, and so on. The advantages of algebraic aggregation over geometric aggregation are that algebraic aggregation groups states with strongly connected neighbor states automatically on all recursive levels, which is crucial for good performance, and that unstructured chains can be considered for which there is no direct geometric approach to aggregation. The numerical results below show that our aggregation process performs well for most problems. Other types of aggregation [5, 20, 23, 17, 28, 29] could be considered for use with our algorithm. The best choice of aggregation may be problem-dependent.

6.1. Structured problems. Tables 6.1 to 6.8 show algebraic aggregation performance results for the structured test problems from section 5. It can be observed that, for all problems, the unsmoothed method, A-AM, performs poorly. Smoothing (with lumping) improves efficiency dramatically and, for most problems, highly scalable results are obtained: convergence rates and iteration counts are low and nearly constant, and operator complexities are small and appear bounded. We can make the following observations for the specific test problems.

TABLE 6.1
Uniform chain. A-AM with V-cycles and distance-one aggregation. (No smoothing.)

n	it	C_{op}	lev	γ	γ_{eff}
27	39	1.71	3	0.74	0.84
81	83	1.85	4	0.87	0.93
243	>100	1.96	6	0.96	0.98
729	>100	1.98	7	1.00	1.00

TABLE 6.2
Uniform chain. A-SAM with V-cycles using distance-one aggregation (left) and distance-two aggregation (right). (Smoothing with lumping.)

n	it	C_{op}	lev	γ	γ_{eff}	R_{lump}	it	C_{op}	lev	γ	γ_{eff}	R_{lump}
27	10	2.09	3	0.20	0.46	2.4e-2	13	1.33	2	0.32	0.43	0
243	10	2.20	5	0.21	0.49	1.3e-3	12	1.46	4	0.32	0.46	0
6561	11	2.24	8	0.24	0.53	1.4e-3	12	1.49	7	0.31	0.46	0
19683	11	2.25	8	0.26	0.55	1.4e-3	12	1.49	8	0.32	0.47	0
59049	12	2.25	9	0.30	0.59	1.0e-3	12	1.50	9	0.32	0.47	0

For the 1D uniform chain test problem, Tables 6.1 and 6.2 show that operator complexity is somewhat high and slowly growing for distance-one aggregation because smoothing of the restriction and interpolation operators causes fill-in of the coarse-level operators, which are formed by matrix multiplication of the fine-level operator and the restriction and interpolation operators. Fill-in can be avoided for this 1D problem when aggregates are chosen of fixed size three, but it generally does occur for algebraic aggregation procedures, except when they would be especially tailored to this 1D problem. However, C_{op} behaves much better for distance-two aggregation, without compromising convergence factors significantly (scalability is retained). We observed the same effect for the other test problems and use distance-two aggregation

Table 6.3

Birth-death chain ($\mu = 0.96$). (Left) A-AM (no smoothing, distance-one aggregation) and (right) A-SAM (smoothing with lumping, distance-two aggregation) with V-cycles.

n	it	C_{op}	lev	γ	γ_{eff}	it	C_{op}	lev	γ	γ_{eff}	R_{lump}
27	34	1.32	2	0.64	0.71	15	1.32	2	0.35	0.45	0
81	>100	1.43	3	0.88	0.92	15	1.43	3	0.35	0.48	0
243	>100	1.47	4	0.95	0.97	15	1.47	4	0.35	0.49	0
729	>100	1.49	5	0.97	0.98	15	1.49	5	0.35	0.49	0

Table 6.4

Uniform chain with two weak links ($\epsilon = 0.001$). (Left) A-AM (no smoothing, distance-one aggregation) and (right) A-SAM (smoothing with lumping, distance-two aggregation) with V-cycles.

n	it	C_{op}	lev	γ	γ_{eff}	it	C_{op}	lev	γ	γ_{eff}	R_{lump}
54	79	1.43	3	0.86	0.90	14	1.38	3	0.32	0.44	0
486	>100	1.49	5	0.98	0.99	13	1.48	5	0.32	0.46	0
4374	>100	1.49	7	0.98	0.98	12	1.49	6	0.32	0.46	0

Table 6.5

Uniform 2D lattice. (Left) A-AM (no smoothing, distance-one aggregation) and (right) A-SAM (smoothing with lumping, distance-two aggregation) with V-cycles.

n	it	C_{op}	lev	γ	γ_{eff}	it	C_{op}	lev	γ	γ_{eff}	R_{lump}
64	47	1.70	3	0.75	0.85	18	1.25	2	0.42	0.50	0.0e-0
1024	>100	1.87	5	0.92	0.95	20	1.42	4	0.49	0.60	4.5e-3
4096	>100	1.91	6	0.98	0.99	20	1.47	4	0.49	0.62	1.7e-3
16384	>100	1.92	7	0.98	0.99	20	1.56	5	0.59	0.72	1.4e-3
65536	>100	1.92	9	0.97	0.98	21	1.59	6	0.66	0.77	1.3e-3

Table 6.6

Anisotropic 2D lattice ($\epsilon = 1e-6$). (Left) A-AM (no smoothing, distance-one aggregation) and (right) A-SAM (smoothing with lumping, distance-two aggregation) with V-cycles.

n	it	C_{op}	lev	γ	γ_{eff}	it	C_{op}	lev	γ	γ_{eff}	R_{lump}
64	27	1.78	4	0.55	0.72	17	1.76	3	0.40	0.59	0.0e-0
1024	80	2.01	8	0.86	0.93	14	2.81	5	0.33	0.68	1.6e-3
4096	>100	2.04	10	0.93	0.96	14	3.43	7	0.33	0.73	4.9e-4
16384	>100	2.06	11	0.96	0.98	13	4.17	7	0.33	0.77	2.5e-4
65536	>100	2.16	13	0.97	0.99	13	4.80	9	0.32	0.79	7.6e-5

for them. Note that, for this highly regular 1D problem, almost no lumping is required for distance-two aggregation, due to the very small fill-in.

Table 6.4 for the uniform chain with two weak links shows that the strength-based aggregation procedure chooses aggregates automatically in such a way that no states are aggregated together that are locally weakly connected.

Comparing Table 6.6 with Table 5.9 (geometric aggregation) for the 2D anisotropic lattice test problem, it appears that algebraic aggregation performs much better for unsmoothed V-cycles than geometric aggregation, because it chooses aggregates in the direction of strong connections. Smoothed V-cycles with distance-two aggregation are a significant improvement over unsmoothed V-cycles. Smoothing leads to higher operator complexity for this test problem. Highly anisotropic problems are difficult for smoothed aggregation methods [29], because the strength-based aggregation tends to create elongated, 1D aggregates. Smoothing these aggregates leads to large increases in the aggregate sizes, which causes significant fill-in and increasing operator complexity. Convergence factors for the smoothed method in Table 6.6 are scalable. Very small amounts of lumping are required.

Table 6.7
Tandem queueing network. A-AM with V-cycles and distance-one aggregation. (No smoothing.)

n	it	C_{op}	lev	γ	γ_{eff}
256	>100	1.86	5	0.92	0.96
4096	>100	2.03	7	0.98	0.99
16384	>100	2.09	9	0.98	0.99

Table 6.8
Tandem queueing network. A-SAM with V-cycles and distance-two aggregation. (Smoothing with lumping.)

n	it	C_{op}	lev	γ	γ_{eff}	R_{lump}
256	18	1.94	4	0.39	0.61	1.1e-1
4096	24	2.12	5	0.45	0.69	5.5e-2
16384	30	2.18	6	0.56	0.77	5.3e-2
65536	37	2.37	6	0.71	0.86	1.3e-1

The amount of lumping required for the nonsymmetric tandem queueing problem (Table 6.8) is somewhat larger than for the previous problems, but it is still small and does not add much extra work.

6.2. Random walk on unstructured planar graph. Random walks on graphs have important applications in several fields, including information retrieval and statistical physics [30]. For example, importance rankings can be determined as stationary distributions of random walks on graphs that describe relations between the items to be ranked. Google's PageRank algorithm is a well-known example [24, 25, 6]. In our final 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 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.

Tables 6.9 and 6.10 show numerical results for the random walk test problem with algebraic aggregation. For this application, smoothed V-cycles with distance-one ag-

Table 6.9
Unstructured planar graph. A-SAM with V-cycles and distance-one aggregation. (Smoothing with lumping.)

n	it	C_{op}	lev	γ	γ_{eff}	R_{lump}
1024	20	1.69	5	0.53	0.68	2.6e-2
4096	21	1.80	5	0.61	0.76	2.4e-2
8192	22	1.92	7	0.64	0.79	2.5e-2
16384	30	2.03	7	0.76	0.87	2.4e-2
32768	28	2.08	7	0.74	0.86	2.4e-2

Table 6.10
Unstructured planar graph with $n = 32768$ nodes. Comparison of multilevel A-AM, two-level A-AM (aggregation-disaggregation), and one-level relaxation. (V-cycles and distance-one aggregation, no smoothing. The $V(1,1)$-cycles have two relaxations per level per cycle.)

$n = 32768$	it	C_{op}	lev	γ	γ_{eff}
Multilevel A-AM	>100	1.26	8	0.98	0.99
Two-level A-AM	>100	1.20	2	0.98	0.99
One-level method	>100	1.00	1	0.98	0.98

gregation perform better than cycles with distance-two aggregation. Table 6.10 compares one-level relaxation (with two relaxations on the fine level per iteration), two-level A-AM aggregation (with two relaxations on the coarse level; this is essentially equivalent to the aggregation/disaggregation methods of [7, 8, 9, 10, 11, 12, 13, 14]), and unsmoothed, multilevel A-AM (which is the same as MAA in [6] and similar to the multilevel aggregation methods of [4, 5]). It can be observed that traditional two-level and multilevel aggregation methods hardly manage to accelerate convergence for this problem. Table 6.9, however, shows that our smoothed multilevel method does improve convergence significantly.

7. Conclusions and future work. We showed how smoothing of the interpolation and restriction operators can dramatically increase the efficiency of aggregation multigrid for Markov chains. A new lumping technique assures well-posedness of the coarse-level problems. These results may lead to the adoption of true multilevel aggregation methods for calculating stationary probability vectors of irreducible Markov chains in several application areas.

There are several interesting avenues for future research. More rigorous theoretical underpinning of the proposed smoothed aggregation method for Markov chains, and the algebraic aggregation strategy, need to be considered. It is plausible that the theoretical results of [11, 12, 14], which demonstrate convergence of two-level, unsmoothed aggregation methods for Markov chains under some conditions, can be extended to the case of our smoothed method. However, proving optimal computational complexity, linear in the number of unknowns, is expected to be much more difficult, because the Markov chain matrices are, in general, not symmetric. This is similar to the case of AMG and smoothed aggregation for linear systems, for which there is some theory for the symmetric case [31, 20, 21, 17], but for which there is also still no theory yet about optimality of convergence for the nonsymmetric case. Nevertheless, for AMG, decades of numerical experience in the field show that nearly optimal performance can often be obtained [32, 16].

Also, it would be interesting to extend the applicability of the methods proposed in this paper to larger classes of Markov chains. Developing optimal multilevel solvers for various classes of well-studied Markov chains for which one-level or two-level iterative methods converge slowly is an interesting topic of further research.

REFERENCES

[1] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, Philadelphia, 1994.

[2] B. Philippe, Y. Saad, and W. J. Stewart, *Numerical methods in Markov chain modeling*, Oper. Res., 40 (1992), pp. 1156–1179.

[3] W. J. Stewart, *An Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.

[4] G. Horton and S. T. Leutenegger, *A multi-level solution algorithm for steady-state Markov chains*, in Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems (Nashville, TN), 1994, pp. 191–200.

[5] S. T. Leutenegger and G. Horton, *On the utility of the multi-level algorithm for the solution of nearly completely decomposable Markov chains*, in Numerical Solution of Markov Chains, W. Stewart, ed., Kluwer Academic, Dordrecht, The Netherlands, 1995, pp. 425–443.

[6] 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.

[7] 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.

[8] J. R. Koury, D. F. McAllister, and W. J. Stewart, *Iterative methods for computing stationary distributions of nearly completely decomposable Markov chains*, SIAM J. Algebraic Discrete Methods, 5 (1984), pp. 164–186.

[9] P. J. Schweitzer and K. W. Kindle, *An iterative aggregation-disaggregation algorithm for solving linear equations*, Appl. Math. Comput., 18 (1986), pp. 313–354.

[10] U. R. Krieger, B. Mller-Clostermann, and M. Sczittnick, *Modeling and analysis of communication systems based on computational methods for Markov chains*, IEEE J. Selected Areas Commun., 8 (1990), pp. 1630–1648.

[11] U. R. Krieger, *On a two-level multigrid solution method for finite Markov chains*, Linear Algebra Appl., 223/224 (1995), pp. 415–438.

[12] I. Marek and P. Mayer, *Convergence analysis of an iterative aggregation/disaggregation method for computing stationary probability vectors of stochastic matrices*, Numer. Linear Algebra Appl., 5 (1998), pp. 253–274.

[13] T. Dayar and W. J. Stewart, *Comparison of partitioning techniques for two-level iterative solvers on large, sparse Markov chains*, SIAM J. Sci. Comput., 21 (2000), pp. 1691–1705.

[14] I. Marek and P. Mayer, *Convergence theory of some classes of iterative aggregation/disaggregation methods for computing stationary probability vectors of stochastic matrices*, Linear Algebra Appl., 363 (2003), pp. 177–200.

[15] 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, UK, 1984.

[16] J. W. Ruge and K. Stüben, *Algebraic multigrid*, in Multigrid Methods, Frontiers Appl. Math. 3, S. F. McCormick, ed., SIAM, Philadelphia, 1987, pp. 73–130.

[17] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive smoothed aggregation (αSA) multigrid*, SIAM Rev., 47 (2005), pp. 317–346.

[18] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive algebraic multigrid*, SIAM J. Sci. Comput., 27 (2006), pp. 1261–1286.

[19] M. Brezina, T. Manteuffel, S. McCormick, J. Ruge, G. Sanders, and P. Vassilevski, *A generalized eigensolver based on smoothed aggregation (GES-SA) for initializing smoothed aggregation (SA) multigrid*, Numer. Linear Algebra Appl., 15 (2008), pp. 249–269.

[20] P. Vanek, J. Mandel, and M. Brezina, *Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems*, Computing, 56 (1996), pp. 179–196.

[21] P. Vanek, M. Brezina, and J. Mandel, *Convergence of algebraic multigrid based on smoothed aggregation*, Numer. Math., 88 (2001), pp. 559–579.

[22] E. Virnik, *An algebraic multigrid preconditioner for a class of singular M-matrices*, SIAM J. Sci. Comput., 29 (2007), pp. 1982–1991.

[23] C. Isensee and G. Horton, *A multi-level method for the steady state solution of Markov chains*, Simulation und Visualisierung 2004, T. Schulze, S. Schlechtweg, and V. Hinz, eds., SCS Publishing House e.V., Erlangen, Germany, 2004, pp. 191–202.

[24] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Technical Report 1999-0120, Computer Science Department, Stanford University, Stanford, CA, 1999.

[25] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, NJ, 2006.

[26] H. A. Simon and A. Ando, *Aggregation of variables in dynamic systems*, Econometrica, 29 (1961), pp. 111–138.

[27] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2nd ed., SIAM, Philadelphia, 2000.

[28] A. C. Muresan and Y. Notay, *Analysis of aggregation-based multigrid*, SIAM J. Sci. Comput., 30 (2008), pp. 1082–1103.

[29] M. W. Gee, J. J. Hu, and R. S. Tuminaro, *A new smoothed aggregation multigrid method for anisotropic problems*, Numer. Linear Algebra Appl., 16 (2009), pp. 19–37.

[30] L. Lovasz, *Random walks on graphs: A survey*, in Combinatorics, Paul Erdős Is Eighty, Vol. 2, Bolyai Soc. Math. Stud. 2, János Bolyai Mathematical Society, Budapest, 1996, pp. 353–397.

[31] A. Brandt, *Algebraic multigrid theory: The symmetric case*, Appl. Math. Comput., 19 (1986), pp. 23–56.

[32] A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, G. N. Miranda, and J. W. Ruge, *Robustness and algorithmic scalability of algebraic multigrid*, SIAM J. Sci. Comput., 21 (2000), pp. 1886–1908.