A BOOTSTRAP ALGEBRAIC MULTILEVEL METHOD FOR MARKOV CHAINS

MATTHIAS BOLTEN[†], ACHI BRANDT[‡], JAMES BRANNICK[§], ANDREAS FROMMER[†], KARSTEN KAHL[†], AND IRA LIVSHITS[¶]

Abstract. This work concerns the development of an algebraic multilevel method for computing state vectors of Markov chains. We present an efficient bootstrap algebraic multigrid method for this task. In our proposed approach, we employ a multilevel eigensolver, with interpolation built using ideas based on compatible relaxation, algebraic distances, and least squares fitting of test vectors. Our adaptive variational strategy for computation of the state vector of a given Markov chain is then a combination of this multilevel eigensolver and an associated additive multilevel preconditioned correction process. We show that the bootstrap algebraic multigrid eigensolver by itself can efficiently compute accurate approximations to the steady state vector. An additional benefit of the bootstrap approach is that it yields an accurate interpolation operator for many other eigenmodes. This in turn allows for the use of the resulting multigrid hierarchy as a preconditioner to accelerate the GMRES iteration for computing an additive correction equation for the approximation to the steady state vector. Unlike other existing multilevel methods for Markov chains, our method does not employ any special processing of the coarse-level systems to ensure that stochastic properties of the fine-level system are maintained there. The proposed approach is applied to a range of test problems, involving non-symmetric M-matrices arising from stochastic matrices, showing promising results.

Key words. Markov chains, algebraic multigrid, bootstrap, adaptive methods, least squares prolongation, preconditioned GMRES iteration

AMS subject classifications. 65F15, 65F08, 60J22

1. Introduction.

We consider the task of computing a non-zero vector x such that

$$Ax = 1\,x, \tag{1.1}$$

where A denotes the transition matrix of a given irreducible Markov process and x is the associated steady state vector (s.s.v.), an eigenvector of A with eigenvalue equal to one. Since A is irreducible, x is unique up to scalar factors. The approach considered in this paper is to approximate x iteratively using an adaptively constructed algebraic multilevel eigensolver (MLE) for the matrix

$$B = I - A.$$

In addition, the multilevel hierarchy thus obtained is used in an algebraic multigrid method (AMG) acting as a preconditioner for the GMRES iteration, which is used to approximately solve the error equation

$$Be = -B\tilde{x},$$

where $\tilde{x}$ is the approximation from the MLE and $x = \tilde{x} + e$.

[†] Fachbereich C, Mathematik und Naturwissenschaften, Bergische Universität Wuppertal, 42097 Wuppertal, Germany, {bolten,frommer,kahl}@math.uni-wuppertal.de

[‡] Department of Applied Mathematics & Computer Science, Weizmann Institute of Science, 76100 Rehovot, Israel, achi.brandt@weizmann.ac.il

[§] Department of Mathematics, Pennsylvania State University, University Park, PA 16802, USA, brannick@psu.edu

[¶] Department of Mathematical Sciences, Ball State University, Muncie, IN 47306, USA, ilivshits@bsu.edu

Our multilevel eigensolver relies on the bootstrap framework, a fully adaptive algebraic multigrid scheme proposed in [3]. Here, the term 'bootstrap' is meant to indicate that this adaptive scheme is a self-sustaining, iteratively improving process.

For solving complex-valued linear systems, this BAMG (bootstrap AMG) framework was efficiently put into action in [5]. In the present paper, we develop a variant of BAMG specifically tailored to compute x in (1.1). Although we are not (yet) in a position to give a full rigorous mathematical analysis of the method, we demonstrate the efficiency of the BAMG approach for a series of Markov chain test problems.

This paper is organized as follows: Section 2 contains a review of basic material on Markov chain systems as well as a discussion of various other multilevel approaches for computing steady state vectors of Markov chains. Section 3 contains a general description of the bootstrap AMG ideas along with full algorithmic descriptions of the realizations that we use in this paper. This includes the "compatible relaxation" techniques used to determine coarse-grid variables as a subset of the fine grid variables and an adaptive procedure to determine interpolation operators based on least squares fits of certain test vectors. In section 4, this BAMG setup process is specified in all its details for the case of Markov chain problems considered here. We develop a multilevel eigensolver setup which aims at obtaining a multigrid hierarchy in which the eigenvectors corresponding to small eigenvalues are well represented. This is done adaptively and recursively by using relaxation to expose such eigenvectors, at each level of the hierarchy and by building least squares based interpolation operators that well interpolate these approximate eigenvectors. We view this multilevel eigensolver setup as the main contribution of this paper together with the idea of combining it with AMG preconditioned GMRES for the error equation to cheaply compute an additive correction to the approximation to the steady state vector once an efficient multigrid hierarchy has been established. This additive correction process is also discussed in section 4. Numerical experiments for larger Markov chain problems are presented in section 5 and concluding remarks are given in section 6.

2. Review of Basic Material and Previous Multilevel Approaches.

The transition matrix $A \in \mathbb{R}^{n\times n}$ of a Markov process contains as its entries a_{ij} the transition probabilities from state i to state j, $a_{ij} \geq 0$ for all i, j. Matrix A is column stochastic, i.e., $A^*\mathbf{1} = \mathbf{1}$, with $\mathbf{1}$ being the vector of all ones and A^* denoting the adjoint of A in the euclidean inner product on $\mathbb{R}^n$. It is always possible to eliminate self-transitions (see, e.g. [30]), so we assume $a_{ii} = 0$ for all i from now on. The steady state vector x satisfies

$$Ax = x,$$

with $x \neq 0$, $0 \leq x_i, i = 1, \ldots, n$. By the Perron-Frobenius theorem (cf. [1, p. 27], e.g.) such a vector x always exists.

A general square matrix $A \in \mathbb{R}^{n\times n}$ induces a directed graph $D(A) = (\Omega, E)$ with vertices $\Omega = \{1, \ldots, n\}$ and directed edges $E = \{(i, j) \in V^2 : i \neq j \text{ and } a_{ij} \neq 0\}$. Two vertices i and j in $D(A)$ are said to be strongly connected if there exist directed paths in $D(A)$ from i to j and from j to i. Since this is an equivalence relation on the vertices, it induces a partitioning of V into the strong components of $D(A)$. If $D(A)$ has exactly one strong component, the matrix A is called irreducible; otherwise it is called reducible which is precisely the case when there exists a permutation matrix $\mathcal{P}$ such that

$$\mathcal{P}^* A \mathcal{P} = \begin{pmatrix} A_{11} & 0 \\ A_{12} & A_{22} \end{pmatrix},$$

where A_{11} and A_{22} are square submatrices.

If A is irreducible – which we assume throughout – the Perron-Frobenius Theorem guarantees that x has all positive values and is unique up to a scalar factor. From this theorem it also follows that the spectral radius of A satisfies $\rho(A) = 1$, implying that $B = I - A$ is a singular M-matrix; recall that $0 \leq a_{ij} \leq 1$. In addition, B is irreducible, because of the irreducibility of A.

The idea of using multilevel approaches to compute the steady state vector of an irreducible transition matrix dates back at least to work in [31] from 1961 and [33] from 1975. Since then, a variety of methods have evolved which are usually termed as aggregation/disaggregation (a/d) methods, since they build a hierarchy of increasingly smaller Markov chains by grouping variables. A full review of the complete literature on a/d methods would be far beyond the scope of this paper. We therefore just mention the paper [13] which gives a thorough introduction into the concept of aggregation in Markov chains with a discussion of multiplicative vs. additive methods and the paper [20] which gives a survey on different a/d methods together with a unifying approach which comprises many a/d methods published so far. In terms of convergence analysis we point to [27] where a local convergence result for a/d methods was given, and then further developed to a global convergence result in [28].

Practical applications for communication networks are discussed in [23], where several a/d methods as well as direct and iterative solvers were compared. Many large Markov chains coming from real applications expose an inherent tensor structure. This was pointed out in [10] where the a/d framework was adjusted so as to take advantage of this tensor structure. Numerical results for larger problems, refined cycling strategies and adaptivity in the choice of aggregating certain domains in the tensor structure are presented in [11, 12].

Apart from the a/d methods, which certainly represent the prevailing multilevel scheme for Markov chains, multilevel methods which are closer to the idea of algebraic multigrid methods for solving linear systems have gained some attention in recent years. One such approach is developed in [35] as a Schur complement based algebraic multigrid method for singular M-matrices, combined with an accelerating Krylov subspace method. Another recent development is the extension of the adaptive smoothed aggregation multigrid method from [9] to problems arising in Markov chain modeling [14, 15, 16, 17].

The method we propose in the present paper tries to further develop algebraic multigrid approaches for Markov chains. It can be loosely categorized as a variation of classical algebraic multigrid in the following way. We choose the coarse-level variables as a subset of the fine level variables. In this aspect, our approach is most closely related to the recent work from [35]; we too employ the AMG-type method as a preconditioner in GMRES. Our proposed new approach, however, differs from all previous AMG-type solvers for Markov chains in several ways. Most importantly, we build interpolation adaptively using a least squares approach, and we use the resulting multilevel hierarchy in an AMG preconditioner for GMRES for the error equation. This has the advantage that we do not have to recompute interpolation and thus the entire coarse-level hierarchy at each step. In addition, in order to achieve high accuracy for the state vector our approach does not use lumping—a manipulation used in [14, 35] and other such approaches which aim to maintain the stochastic properties of the transition matrix and state vectors on all coarse levels of the hierarchy.

3. Bootstrap AMG.

In this section, we provide a general review of the bootstrap AMG framework [3, 6] together with some heuristic motivation of our choices for the individual components of the BAMG multigrid algorithm.

The first key ingredient to any (algebraic) multigrid method is its *relaxation* or *smoothing* iteration. For a given system matrix $B \in \mathbb{R}^{n\times n}$ it is in general a stationary iterative process, e.g., a splitting based iteration

$$\begin{aligned} Mu^{\nu+1} &= Nu^{\nu} + b, \quad \text{where } B = M - N,\ M \text{ non-singular}, \\ \iff \quad u^{\nu+1} &= u^{\nu} + M^{-1}(b - Bu^{\nu}),\ \ \nu = 0,1,\dots \end{aligned} \tag{3.1}$$

with its iteration matrix E, usually termed *error propagation matrix* in the multigrid context, given by

$$E = M^{-1}N = I - M^{-1}B.$$

Here, M is chosen such that after a few iterations the error e^{ν} w.r.t. a solution of $Bx = 0$ is *algebraically smooth*, i.e., $\|Be^{\nu}\| \ll \|e^{\nu}\|$. In this paper, unless stated otherwise, $\|\cdot\|$ denotes the ℓ_2-norm. In many situations, an algebraically smooth error can be achieved using point-wise Gauss-Seidel or (under-relaxed) Jacobi iterations[1]. In our Markov chain setting we used ω-Jacobi relaxation with $\omega = 0.7$ on all levels, i.e., we took $M = \frac{1}{\omega}\operatorname{diag}(B)$, resulting in $E = I - \omega\cdot(\operatorname{diag}(B))^{-1}B$ and the smoothing iteration

$$u^{\nu+1} = u^{\nu} + \omega \operatorname{diag}(B)^{-1}(b - Bu^{\nu}),\ \ \nu = 0,1,\dots.$$

The other key ingredient of a multigrid method is its coarse space, together with a corresponding operator on that space and prolongation and restriction operators to transfer between fine and coarse spaces.

In the bootstrap AMG process, coarse variables are selected as a subset of the fine variables $\Omega = \{1,\dots,n\}$. In multigrid terminology, Ω is termed the *fine grid*, motivated by the fact that often the graph $D(A)$ represents a structure with a geometrical interpretation. Let $|D(A)|$ denote the undirected graph obtained from $D(A)$. Then, if $|D(A)|$ is a regular one- or two-dimensional grid, the standard way of obtaining the coarse variables is *full coarsening*. This means that in the one-dimensional case we take every other variable as a coarse variable, where in the two-dimensional case we take every other variable in every other row.

A more general approach is the compatible relaxation (CR) coarsening scheme [2]. The CR scheme can be either started from scratch, or, if geometric information is given and a suitable candidate set of coarse variables is known, such a set can be tested and improved by CR. Once the coarse variables are selected, the crucial point is to construct an appropriate interpolation operator for which we will use the least squares based approach from [5]. Finally, the operator on the coarse grid will be defined via a Petrov-Galerkin condition.

3.1. Choosing the coarse variables: compatible relaxation.

CR is a relaxation based coarsening process which can be viewed as a special case of a general approach for constructing coarse-level representations of given fine-level systems, including non-stationary, highly non-linear, and also non-deterministic systems [4]. The

[1] In case a diagonal entry b_{ii} of B happens to be zero for some row(s) i, Kaczmarz [22] or some other distributive relaxation scheme can be applied to the i^{th} equation.

basic idea of CR is to use the given relaxation scheme (3.1), restricted to appropriately defined subspaces, to measure the quality of the given coarse space and also to iteratively improve it if needed. We proceed with a brief overview of CR and its use in AMG coarsening. A detailed discussion, theory and comparisons between various measures of the quality of coarse spaces and their relations to compatible relaxation schemes are presented in [2, 7, 8, 18, 19, 25].

3.1.1. Classical AMG CR-based coarsening. We want to define the set of coarse-level variables, $\mathcal{C}$, as a subset of the set of fine-level variables, Ω. In this case, one possible form of CR is given by $\mathcal{F}$-relaxation for the homogeneous system — relaxation applied only to the set of $\mathcal{F}$ variables, with $\mathcal{F} := \Omega \setminus \mathcal{C}$. Given the partitioning of Ω into $\mathcal{F}$ and $\mathcal{C}$, we have

$$u = \begin{pmatrix} u_f \\ u_c \end{pmatrix}, \quad B = \begin{pmatrix} B_{ff} & B_{fc} \\ B_{cf} & B_{cc} \end{pmatrix}, \quad \text{and} \quad M = \begin{pmatrix} M_{ff} & M_{fc} \\ M_{cf} & M_{cc} \end{pmatrix},$$

assuming the equations are permuted such that the unknowns in $\mathcal{F}$ come before those in $\mathcal{C}$. The $\mathcal{F}$-relaxation of CR is then defined by

$$u_f^{\nu+1} = u_f^{\nu} - M_{ff}^{-1} B_{ff} u_f^{\nu}, \;\; \nu = 0, 1, \ldots. \tag{3.2}$$

Note that this is a splitting-based iteration for the homogeneous system $B_{ff} u_f = 0$ with error propagation matrix

$$E_f = I - M_{ff}^{-1} B_{ff}.$$

If M is symmetric, the asymptotic convergence rate of CR

$$\rho_f = \rho(E_f),$$

where ρ denotes the spectral radius, provides a measure of the quality of the coarse space, that is, a measure of the ability of the set of coarse variables to represent error not eliminated by the given fine-level relaxation. This measure can be approximated using $\mathcal{F}$-relaxation for the homogeneous system with a random initial guess u^0. Since $\lim_{\nu \to \infty} \|E_f^{\nu}\|^{1/\nu} = \rho(E_f)$ for any norm $\|\cdot\|$, the measure

$$\rho_f(u^0, \nu) = \left(\|u_f^{\nu}\| / \|u_f^0\| \right)^{1/\nu} \tag{3.3}$$

estimates ρ_f. We note that in the actual compatible relaxation process it is only necessary to work with u_f instead of the "full" vector u, but for the sake of simplicity in notation we refrain from using u_f explicitly here and in Algorithm 1.

In choosing $\mathcal{C}$, we use the CR-based coarsening algorithm developed in [8]. This approach is described in Algorithm 1. In our numerical experiments for Markov chains, we use weighted Jacobi $\mathcal{F}$-relaxation, i.e., we take $M_{ff} = \frac{1}{\omega}\operatorname{diag}(B_{ff})$ with $\omega = 0.7$, set the CR tolerance $\theta = .8$, the number of CR sweeps $\nu = 8$, choose the components of u^0 to be uniformly distributed in the interval $[1, 2]$, take $\|\cdot\|$ in (3.3) as the maximum norm $\|\cdot\|_\infty$ and select $\mathcal{C}_0$ using the standard maximal independent set algorithm (see [34]) based on the full undirected graph $|D(A)|$ of the system matrix B. For further information on CR we refer the reader to [8].

Algorithm 1 compatible_relaxation (Computes $\mathcal{C}$ using compatible relaxation)

```
Input: $\mathcal{C}_0$ {$\mathcal{C}_0 = \emptyset$ is allowed}
Output: $\mathcal{C}$
Initialize $\mathcal{C} = \mathcal{C}_0$
Initialize $\mathcal{F} = \Omega \setminus \mathcal{C}$
while $\rho_f(u^0, \nu) > \theta$ do
  Perform $\nu$ CR iterations (3.2) with components of $u^0$ randomly generated
  $\mathcal{N} = \{i \in \mathcal{F} : \frac{|u_i^{\nu}|}{|u_i^{\nu-1}|} > \theta\}$
  $\mathcal{C} = \mathcal{C} \cup$ maximal independent set of $\mathcal{N}$
  $\mathcal{F} = \Omega \setminus \mathcal{C}$
end while
```

3.2. Building bootstrap AMG interpolation. We now outline the least squares approach for defining interpolation, see also [5]. Interpolation is a linear operator which maps vectors from the coarser level $\ell + 1$ to vectors on the finer level ℓ. For notational simplicity, instead of using an index ℓ, we describe the generic situation where the fine level is the space $\mathbb{R}^n$ with fine variables from $\Omega = \{1, \ldots, n\}$ and the set of coarse variables is $\mathcal{C} \subset \Omega$ with $|\mathcal{C}| = n_c$. So $\mathcal{C}$ might have been previously determined, for instance, by full coarsening and/or using compatible relaxation. In the least squares approach, the interpolation operator P, a linear mapping from $\mathbb{R}^{n_c}$ to $\mathbb{R}^n$, is built such that it approximates well a given (specifically chosen) set of (potentially complex) test vectors $\mathcal{T} \subset \mathbb{R}^n$. Denoting by $R \in \mathbb{R}^{n_c \times n}$ the canonical injection operator which maps every vector $x \in \mathbb{R}^n$ on its components from $\mathcal{C}$ we thus aim at $P(Rx) \approx x$ for $x \in \mathcal{T}$.

We define c, the *caliber of interpolation*, as the maximum number of coarse-level variables used to interpolate to a single fine-level variable, or equivalently, the maximum number of non-zero entries in any row of P. The key ingredient of the BAMG setup is to choose $\mathcal{T}$such that it collectively represents those error components not reduced by relaxation. (In Markov context, the test set also contains a representation of the solution - the steady state vector.) We assume for now that such a set of test vectors, $\mathcal{T} = \{x^{(k)}\}_{k=0}^{r} \subset \mathbb{R}^n$ is known on the fine level. The rows of the prolongation operator P are then obtained individually. For each variable $i \in \mathcal{F}$, we first determine a set of its neighboring coarse-level variables, $\mathcal{N}_i^z$, using the directed graph $D(A)$

$$\mathcal{N}_i^z = \{j \in \mathcal{C} : \text{ there is a path of length} \leq z \text{ in } D(A) \text{ from } i \text{ to } j \ \}. \tag{3.4}$$

Note that for small values of z, the set $\mathcal{N}_i^z$ can be regarded as a local graph neighborhood of i. Typical values for z are 1 or 2. We then determine an appropriate set of (coarse level) interpolatory variables $\mathcal{J}_i \subseteq \mathcal{N}_i^z$ with $|\mathcal{J}_i| \leq c$ that fits well our interpolation needs for point $i \in \mathcal{F}$. For this purpose, we define the (weighted) local least squares functional for the nonzero entries $P_i = \{p_{ij} : j \in \mathcal{J}_i\}$ of row i of P as

$$L(P_i; \mathcal{J}_i) = \sum_{\kappa=0}^{k} \omega_k \left(x_i^{(k)} - \sum_{j \in \mathcal{J}_i} p_{ij} X_j^{(k)} \right)^2 / \, \|x^{(k)}\|^2, \tag{3.5}$$

where each $X^{(k)} = Rx^{(k)}$ is a coarse-level representative of $x^{(k)}$, and R is the injection operator. Dividing by $\|x^{(k)}\|^2$ makes the least squares functional independent of the scaling of the vectors $x^{(k)}$. The weights ω_k should be chosen according to the algebraic smoothness of $x^{(k)}$ to bias the least squares functional towards the smoothest vectors. We do so by taking

$$\omega_k = \left(\frac{\|x^{(k)}\|}{\|Bx^{k)}\|} \right)^2 .$$

The task is then to find a set $\mathcal{J}_i$ of interpolating points for which the minimum of L (as a function of P_i) is small, and to find the corresponding values p_{ij} of the minimizer that yield the coefficients for the interpolation operator. To obtain such a set $\mathcal{J}_i$ for which the minimum of L is small, we use a greedy strategy. A pseudo-code of this strategy is given as Algorithm 2.

Algorithm 2 ls_interpolation (Computes least squares based interpolation)

```
Input: $\mathcal{U}$, $c$
Output: interpolation $P$
Set $X^{(k)} = R\, x^{(k)}, x^{(k)} \in \mathcal{U}$
for $i \in \mathcal{F}$ do
  Take $\mathcal{N}_i = \mathcal{N}_i^z$ from (3.4) for some small value of $z$
  Set $\mathcal{J}_i = \emptyset$
  repeat
    Determine $g^* \in \mathcal{N}_i$ s.t. $\min_{P_i} L(P_i; \mathcal{J}_i \cup \{g^*\}) = \min_{g \in \mathcal{N}_i} \min_{P_i} L(P_i; \mathcal{J}_i \cup \{g\})$
    Set $\mathcal{N}_i = \mathcal{N}_i \setminus \{g^*\}$ and $\mathcal{J}_i = \mathcal{J}_i \cup \{g^*\}$
  until $|\mathcal{J}_i| \geq c$ or $\mathcal{N}_i = \emptyset$
end for
```

4. MLE setup and additive correction. Given the general description of BAMG from the previous section, we now specify in detail its ingredients and its implementation for Markov chains.

We are looking for a non-trivial solution x to $Bx = 0$, where $B = I - A$, A is an irreducible column stochastic transition matrix so that B is a possibly non-symmetric and singular M-matrix. Taking an appropriate start vector, we approximate x by computing a multilevel eigensolver (MLE) using BAMG for the matrix B. The purpose of the BAMG setup is actually twofold: In addition to improving the approximation $\tilde{x}$ to the solution it also produces a multigrid hierarchy defining an AMG method for B.

We use this AMG method as a preconditioner for the GMRES method to approximately solve the error equation

$$Be = -B\tilde{x},$$

a singular, consistent linear system. With $\tilde{e}$ the approximate solution thus obtained, the new approximation to the steady state vector is $\tilde{x} + \tilde{e}$.

The resulting overall method will then alternate between relatively expensive MLE setup steps and relatively cheap additive correction steps using AMG preconditioned GMRES. Actually, it will turn out that in many cases one obtains the state vector to sufficient accuracy if one computes the MLE setup just once and then runs a fairly small, ten say, number of AMG preconditioned GMRES iterations in the additive correction.

4.1. BAMG Setup Using a Multilevel Eigensolver. Algorithm 3 describes the MLE (multilevel eigensolver) setup for BAMG for Markov chains. It combines the multilevel eigensolver approach with the the least squares approach for obtaining prolongation operators in BAMG: The multilevel eigensolver yields approximations to eigenvectors corresponding to small eigenvalues of B which are used as test vectors in the least squares interpolation. The parameter μ in Algorithm 3 determines the type of the setup cycle: For $\mu = 1$ we have a V-cycle, $\mu = 2$ yields a W-cycle.

Algorithm 3 takes as input the parameters $B_\ell, T_\ell, \mathcal{U}_\ell$ and $\mathcal{V}_\ell$. Then, to perform the MLE setup the approach will – possibly repeatedly – be called from the finest level ($\ell = 0$), where $B_0 = B, T_0 = I$. Here, the set of test vectors $\mathcal{U}_0 = \{u_0^\kappa\}, \kappa = 1, \ldots, k_u$ is chosen randomly for the first call; it is just re-used in any further call to the algorithm at level 0. The set $\mathcal{V}_0$ represents approximations to the smallest eigenvectors obtained so far. It is empty for the first call at level 0, but for any further call at this level it is taken as the output from the previous call. The number k_v of approximate eigenpairs $v_\ell^{(\kappa)}$ in $\mathcal{V}_\ell$ is constant for all levels. It is determined by the number of exact eigenpairs that we compute at the coarsest level ($\ell = L$).

Algorithm 3 BAMG_mle (One cycle of the MLE setup for BAMG)

```
Input: $B_\ell$ ($B_0 = B$), $T_\ell$ ($T_0 = I$), $\mathcal{U}_\ell = \{u_\ell^{(\kappa)}\}$, $\mathcal{V}_\ell = \{v_\ell^{(\kappa)}\}$, $\Lambda_\ell = \{\lambda_\ell^{(\kappa)}\}$;
Output: $\mathcal{V}_\ell = \{v_\ell^{(\kappa)}\}$, $\Lambda_\ell = \{\lambda_\ell^{(\kappa)}\}$, approximations to the lowest eigenvectors and eigenvalues;
if $\ell = L$ then
  Compute $\mathcal{V}_L = \{v_L^{(\kappa)} \mid B_L v_L^{(\kappa)} = \lambda_L^{(\kappa)} T_L v_L^{(\kappa)}\}$
        {the $k_v$ eigenpairs of pencil $(B_L, T_L)$ with smallest modulus of $\lambda_L^{(\kappa)}$}
else
  Relax $B_\ell u_\ell^{(\kappa)} = 0$ for all $u_\ell^{(\kappa)} \in \mathcal{U}_\ell$          {$s_1$ iterative steps of (4.1)}
  Relax $(B_\ell - \lambda_\ell^{(\kappa)} T_\ell) v_\ell^{(\kappa)} = 0$ for all $v_\ell^{(\kappa)} \in \mathcal{V}_\ell$          {$s_1$ iterative steps of (4.1)}
  for $m = 1, \ldots, \mu$ do
    $P_\ell = $ ls_interpolation$(\mathcal{U}_\ell \cup \mathcal{V}_\ell, c)$
    Compute averaging $Q_\ell$ with sparsity$(Q_\ell^*) = $ sparsity$(P_\ell)$
    $B_{\ell+1} = Q_\ell B_\ell P_\ell,\ T_{\ell+1} = Q_\ell T_\ell P_\ell$
    $\mathcal{U}_{\ell+1} = \{R_\ell u_\ell^{(\kappa)} \mid u_\ell^{(\kappa)} \in \mathcal{U}_\ell\}$
    $\mathcal{V}_{\ell+1} = \{R_\ell v_\ell^{(\kappa)} \mid v_\ell^{(\kappa)} \in \mathcal{V}_\ell\}$
    $\mathcal{V}_{\ell+1} = $ BAMG_mle$(B_{\ell+1}, T_{\ell+1}, \mathcal{U}_{\ell+1}, \mathcal{V}_{\ell+1})$
    $\mathcal{V}_\ell = \{P_\ell v_{\ell+1}^{(\kappa)} \mid v_{\ell+1}^{(\kappa)} \in \mathcal{V}_{\ell+1})\}$
    Relax $(B_\ell - \lambda_\ell^{(\kappa)} T_\ell) v_\ell^{(\kappa)} = 0$ for all $v_\ell^{(\kappa)} \in \mathcal{V}_\ell$          {$s_2$ iterative steps of (4.1)}
    Update $\lambda_\ell^{(\kappa)}$: (4.2)
  end for
end if
```

To discuss the various features of Algorithm 3, let us focus on the case $\mu = 1$, i.e., we have just one recursive call. The statements before the recursive call then represent the fine-coarse process and those after the recursive call represent the coarse-to-fine process. In the fine-to-coarse process, at each level we update the test set $\mathcal{U}_\ell$ by performing s steps of ω-Jacobi relaxation on the homogeneous system, i.e., for each $u^{(\kappa)} = u^{(\kappa),0}$ from $\mathcal{U}_\ell$ we iterate

$$u^{(\kappa),\nu+1} = u^{(\kappa),\nu} + \omega \operatorname{diag}(B_\ell)^{-1} B_\ell u^{(\kappa),\nu}, \ \ \nu = 0, \ldots, s_1 - 1. \tag{4.1}$$

with $\omega = 0.7$ and we update $u^{(\kappa)} := u^{(\kappa),s_1}$. The rationale behind this relaxation is that it should damp those eigenvector components in $u^{(\kappa)}$ which belong to large eigenvalues, i.e., we expect the set $\mathcal{U}_\ell$ of the updated vectors to well approximate the space belonging to the small eigenvalues of B_ℓ. Note that individually none of these vectors will usually represent a single eigenvector. Relaxation for $v_\ell^{(\kappa)}$, although technically the same follows a slightly different rationale. As explained in detail later on, these vectors already represent the small eigenvectors of B, and are expected to further improve with additional relaxation. After relaxation, $\mathcal{U}_\ell$ and $\mathcal{V}_\ell$ together should thus well represent *all* eigenvectors corresponding to small eigenvalues.

With this set of updated test vectors $\mathcal{U}_\ell \cup \mathcal{V}_\ell$, we then compute the set of coarse variables and the interpolation operator P_ℓ using the least squares approach.

Note that we may encounter (complex conjugate) pairs of complex eigenvalues and eigenvectors on the coarsest level L, so that the sets $\mathcal{V}_\ell$ will contain complex vectors. These are used as such in all steps of the algorithm except when we perform the least squares interpolation where instead of a complex conjugate pair of vectors in $\mathcal{V}_\ell$ we actually use their real and imaginary parts. In doing so we guarantee that all occurring operators P_ℓ and B_ℓ are real matrices.

The system matrix on level $\ell + 1$ is obtained as a Petrov-Galerkin projection, $B_{\ell+1} = Q_\ell B_\ell P_\ell$, with the "restriction" Q_ℓ given by the averaging operator associated with P_ℓ. The averaging operator leaves the values at coarse variables unchanged, whereas for all other variables i it yields the value given by the arithmetic mean of the values at all coarse variables from $\mathcal{J}_i$. So Q_ℓ has exactly the same non-zero pattern as P_ℓ^T, and within each column the non-zero values are constant and add up to one, i.e., $\mathbf{1}^T Q_\ell = \mathbf{1}^T$. It follows that $\mathbf{1}^T B_\ell = \mathbf{0}$ for all levels ℓ. We also determine the set $\mathcal{U}_{\ell+1}$ of test vectors for level $\ell + 1$ by standard restriction (injection), i.e., we map every vector from $\mathcal{U}_\ell$ onto its coarse components.

On the coarsest level, $\ell = L$, we determine the k_ν smallest (in modulus) eigenvalues and corresponding eigenvectors of the matrix pencil (B_L, T_L) explicitly. Note that the smallest eigenvalue, λ_1 is 0, since B_L is singular. In the coarse-to-fine process, we now aim at obtaining accurate approximations to the smallest eigenpairs of the pencils (B_ℓ, T_ℓ). A tentative set of eigenvectors is computed by prolongating the set $\mathcal{V}_{\ell+1}$ to level ℓ. This candidate set is then improved using relaxation (we again take ω-Jacobi relaxation with $\omega = 0.7$) motivated by the same rationale as for the fine-coarse process, i.e., we expect to reduce contributions of eigenvectors belonging to the larger eigenvalues. Note that an exact eigenvalue equation $B_\ell x - \lambda_\ell T_\ell x = 0$ can be projected onto level $\ell + 1$ in the Petrov-Galerkin sense by requiring x to be in the range of P_ℓ, i.e., $x = P_\ell x_c$ for some x_c from the coarser level, and by requiring the residual $B_\ell P_\ell x_c - \lambda_{\ell+1} T_\ell P_\ell x_c$ to be orthogonal to the range of the adjoint of Q_ℓ. This results in the projected equation $B_{\ell+1} x_c - \lambda_{\ell+1} T_{\ell+1} x_c = 0$ which is precisely the eigenvalue problem considered on level $\ell + 1$. So the prolongations of the vectors from $\mathcal{V}_{\ell+1}$ can be regarded as approximations to eigenvectors at level ℓ.

Once these approximations have been further improved via relaxation, the corresponding eigenvalues are also updated during the coarse-to-fine process: Considering $v_\ell^{(\kappa)} \in \mathcal{V}_\ell$ a reasonable approximation to an eigenvector on level ℓ, i.e,

$$B_\ell v_\ell^{(\kappa)} \approx \lambda_\ell^{(\kappa)} T_\ell v_\ell^{(\kappa)}$$

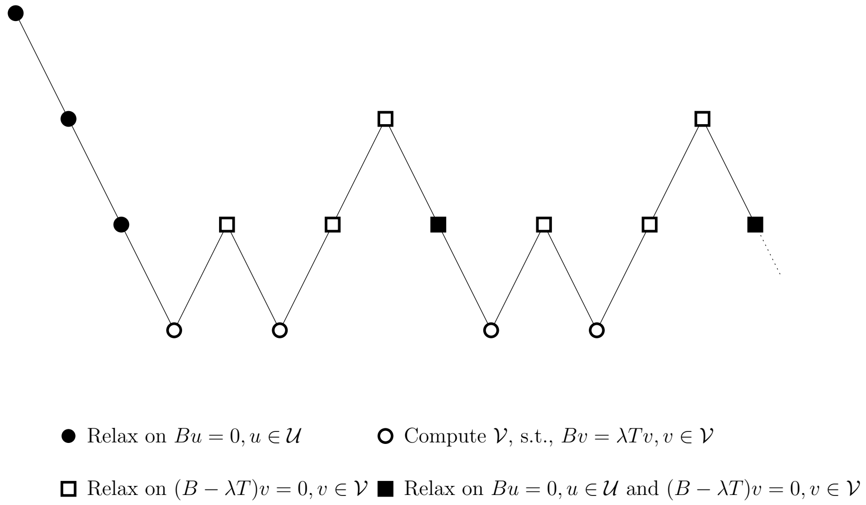

● Relax on $Bu = 0, u \in \mathcal{U}$
○ Compute $\mathcal{V}$, s.t., $Bv = \lambda T v, v \in \mathcal{V}$
□ Relax on $(B - \lambda T)v = 0, v \in \mathcal{V}$
■ Relax on $Bu = 0, u \in \mathcal{U}$ and $(B - \lambda T)v = 0, v \in \mathcal{V}$

Fig. 4.1: Bootstrap AMG setup W-cycle.

for an ℓ^{th} level eigenvalue $\lambda_\ell^{(\kappa)}$, a good eigenvalue approximation can be obtained by

$$\hat{\lambda}^{(\kappa)} = \frac{\langle B_\ell v_\ell^{(\kappa)}, v\rangle}{\langle T_\ell v_\ell^{(\kappa)}, v\rangle} \tag{4.2}$$

for any vector v. We choose $v = v_\ell^{(\kappa)}$ and thus calculate an approximation to an eigenvalue using what would be a Rayleigh quotient for Hermitian matrices. The approximate eigenpair $(v^{(1)}, \hat{\lambda}^{(1)})$ with the smallest eigenvalue $\hat{\lambda}^{(1)} = 0$ corresponds to the zero eigenpair of the pencil (B_ℓ, T_ℓ) so that $v_\ell^{(1)}$ represents the state vector at level ℓ. In particular, $v_0^{(1)}$ approximates the steady state vector of B.

We remark that the fine-coarse process of Algorithm 3 is similar to the “exact interpolation scheme” (see, e.g., [26]).

Switching from $\mu = 1$ to $\mu = 2$ (or even more) allows the MLE setup to improve the approximate eigenvectors from $\mathcal{V}_\ell$ at all levels $\ell \geq 1$ and thus also to readjust the prolongations P_ℓ at all intermediate levels ℓ before one setup cycle is completed. Calling the algorithm several times from the finest level ($\ell = 0$) allows the setup to use and also improve the eigenvector approximations $\mathcal{V}_0$ on the finest level and to update the multigrid hierarchy accordingly. To reflect the cycling strategy in our notation, let a $V^m(s_1, s_2)$ and $W^m(s_1, s_2)$ setup denote one where we call the algorithm k times from the finest level with $\mu = 1$ and $\mu = 2$, respectively, and where we use s_1 relaxation sweeps in the fine-coarse and s_2 in the coarse-to-fine process. Figure 4.1 visualizes a W-cycle. Other cycling strategies are possible, depending on whether at places which are depicted by a square in the figure one decides to recompute the prolongation or to advance to the next finer grid in the multigrid hierarchy.

4.2. AMG preconditioning of the error equation. Each MLE setup cycle is quite costly due to the construction of the interpolation operators. Although we are only interested in computing the steady state vector, the multigrid hierarchy we built up is able to resolve a larger subspace. This leads to the idea of exploiting this richness of the given hierarchy as a preconditioner in methods to solve an additive correction equation for the steady state vectors. This correction phase will be explained now.

With $\tilde{x}$ being the current approximation to the steady state vector, we are looking for the error $e = x - \tilde{x}$ to the exact s.s.v. x. Since $Bx = 0$, the error is a solution of the consistent singular system

$$Be = r \text{ with } r = -B\tilde{x}. \tag{4.3}$$

In principle, we can use the GMRES method [29] to solve this equation iteratively. Each iterative step of GMRES requires one multiplication with the matrix B. In step k the iterate e^k is obtained as the one for which $\|r - B\hat{e}^k\|$ is minimal for all $\hat{e}^k$ from the Krylov subspace $\operatorname{span}\{r, Br, \ldots, B^{k-1}r\}$. Results from [21, 36] show that for a consistent singular system the GMRES method converges to a solution if the nullspace and range of B have only the trivial intersection, $\mathcal{N}(B) \cap \mathcal{R}(B) = 0$. This is true for $B = I - A$ with A the transition matrix of an irreducible Markov chain. Then $\mathcal{N}(B) = \langle x \rangle$ and $\mathcal{R}(B) = \langle (1, \ldots, 1)^T \rangle^\perp$, and since x is positive it cannot belong to $\langle (1, \ldots, 1)^T \rangle^\perp$.

Convergence of the GMRES iteration is slow if B has several or many eigenvalues clustered around 0. A remedy to this situation is to use a preconditioner $P \in \mathbb{C}^{n \times n}$ to solve the preconditioned system

$$(PB)e = Pr$$

instead of (4.3). The preconditioner should be chosen such that the spectrum of PB is well separated from 0 and, ideally, clustered around 1. The geometric interpretation given in [21] shows that this is also desirable in the (consistent) singular case, where, of course, one eigenvalue will stay at 0.

We obtain P implicitly by using the AMG method based on the multigrid hierarchy built up during the MLE setup and perform one $V(s_1, s_2)$-cycle for solving the system $Be = r$. To be specific, let us define the error propagator E_0 of the AMG $V(p_1, p_2)$-cycle for the matrix B recursively by

$$E_\ell = S_\ell^{s_2} \cdot (I - P_\ell E_{\ell+1} Q_\ell B_\ell) \cdot S_\ell^{s_1}, \;\; \ell = 0, \ldots, L-1, \tag{4.4}$$

where $E_{\ell+1}$ denotes the error propagation operator on the next coarser level and $S_\ell = (I - M_\ell^{-1} B_\ell)$ represents the error propagation operator for one smoothing step at level ℓ. This recursive definition terminates at the coarsest level with $E_L = B_L^\dagger$, the Moore-Penrose inverse of the matrix on the coarsest level, since we compute the minimal norm solution of the respective system at that level. Practically this is done using the singular value decomposition of B_L.

Recasting E_0 in the form

$$E_0 = I - PB,$$

one step of the $V(s_1, s_2)$ cycle for $Be = r$ is given as

$$e^1 = (I - PB)e^0 + Pr,$$

so starting with $e^0 = 0$ corresponds to applying one multiplication with the (implicitly defined) preconditioner P.

In the singular case the error propagation matrix E_0 has one eigenvalue equal to 1. An efficient AMG method is characterized by the other eigenvalues being close

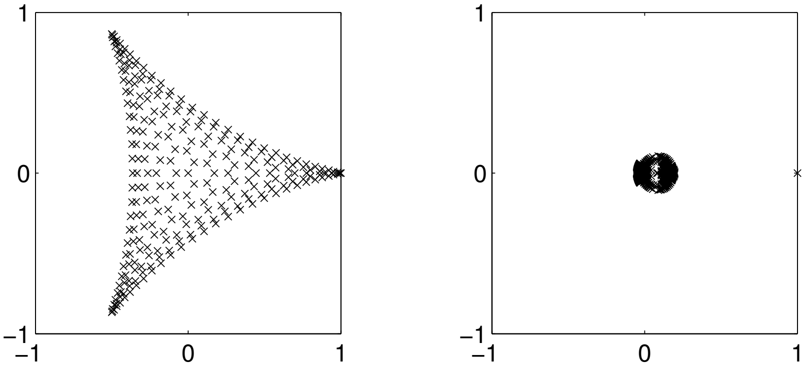


Fig. 4.2: Spectra of original (left) and AMG preconditioned (right) matrix for the tandem queueing network problem with $N = 33^2$.

to 0. Then, the non-zero eigenvalues of PB are clustered around 1, i.e. P is a good preconditioner for B.

Figure 4.2 illustrates the effect of AMG preconditioning with one $V(2,2)$-cycle. It actually shows spectra for the matrices $A = I - B$ and $I - PB$, so the 'critical' eigenvalues are those around 1. The left part gives the spectrum of the transition matrix A arising from the tandem queueing network problem with $N = 33^2$, see section 5. The right part shows the spectrum of the AMG preconditioned matrix $I - PB$. We see that the AMG preconditioning drastically improves the separation of the eigenvalue 1 from the rest of the spectrum which, upon preconditioning, is nicely clustered around 0.

In order for GMRES to converge for the consistent singular preconditioned system $(PB)e = -PB\tilde{x}$ we now need $\mathcal{N}(PB) \cap \mathcal{R}(PB) = 0$. Unfortunately, this seems very difficult to prove in our general context. We can expect P to be non-singular, so that as in the non-preconditioned case $\mathcal{N}(PB)$ is spanned by the steady state vector. But the vector of all ones cannot be guaranteed to be a left eigenvector of P, so that the characterization of $\mathcal{R}(PB)$ does not carry over from the non-preconditioned case.

In our numerical experiments the AMG preconditioned GMRES iteration often converged very rapidly. In these cases there was thus no need to alternate between MLE setups and GMRES iterations; one MLE setup was enough. In some rare cases where the GMRES convergence was slower, we applied our general approach alternating between MLE setups and AMG preconditioned GMRES correction steps.

5. Numerical Results.

In this section, we provide results obtained using the bootstrap MLE/AMG method when applied to a series of Markov chains. Our numerical experiments include results for a variety of Markov chains related to planar graphs. We consider various realizations of our proposed approach and show that with minor modifications it provides an effective algorithm for a wide range of Markov chain models.

We first discuss some common parameters for our numerical tests. In the MLE setup we always use ω-Jacobi as smoother with $\omega = 0.7$. Unless stated otherwise, we use $k_u = k_v = 6$ test vectors and perform $s_1 = s_2 = 3$ relaxation steps to compute them in the MLE algorithm, Algorithm 3. The entries of the test vectors are always chosen randomly from the interval $[1,2]$. The rationale behind this choice is that positive entries are sensible because we want to approximate a steady state vector and that we want to avoid that the ratios of the entries vary too much as they would if we took them randomly from $[0,1]$. The choice of the caliber c of interpolation is problem dependent; it varies between 1 and 4. The coarsening is done by full coarsening if the underlying graph has a 1d or 2d grid structure, otherwise it is obtained with the CR coarsening from Algorithm 1 and its parameters chosen as explained there. The first rows of all our tables always show results obtained when the MLE is applied iteratively without the additive correction step. In these cases, each single MLE iteration is performed as a V-cycle. Here, as everywhere else, the iteration is stopped as soon as the approximation $\tilde{x}$ to the steady state vector satisfies

$$\|B\tilde{x}\|/\|\tilde{x}\| \leq 10^{-8}. \tag{5.1}$$

For most of the examples we also show results where we perform just one MLE setup – which can be a V, a V^2 or a W-cycle – followed by an additive correction step using AMG preconditioned GMRES. In these cases we report the number of GMRES iterations required to satisfy the stopping criterion. The AMG preconditioner used was always a $V(2,2)$-cycle. So, for example, if in a row labeled "W + AMG-GMRES" we report k iterations, this means that we performed one W-cycle MLE setup yielding an approximate s.s.v. $\tilde{x}$. We then started ($V(2,2)$-cycle AMG) preconditioned GMRES on the error equation $Be = -B\tilde{x}$, and after the k-th GMRES iteration the computed approximation to the error, e^k, for the first time satisfied (5.1) for the improved approximation $\tilde{x} \leftarrow \tilde{x} + e^k$.

Only for the last two examples will it be necessary to alternate between MLE and AMG-preconditioned GMRES phases.

For all MLE setups we also report the grid and operator complexities. The grid complexity is the sum of the cardinality of the variable sets at all levels divided by the cardinality of the level 0 variable set. A small grid complexity indicates that the number of variables shrinks fast from one level to the next. For full coarsening of a 1d grid, the grid complexity is about 2; for a 2d grid it is about 1.33. The operator complexity is the sum of all nonzero entries of all operators B_ℓ, divided by the number of non-zero entries of $B = B_0$. The operator complexity is a measure of the relative cost of one V-cycle. Usually, operator complexities below 2 are considered acceptable, whereas the cost for a V-cycle becomes increasingly prohibitive if the operator complexity grows beyond 2.

We begin our experiments with a very simple model taken from [14], namely the one-dimensional network where the transition probability of the inner nodes is given by 1/2 and the probability of boundary nodes is 1, see Figure 5.1. So the entries of $A = (a_{ij})$ for a one-dimensional chain of length N are given as

$$a_{ij} = \begin{cases} 1 & \text{for } (i,j) = (1,2) \text{ and } (i,j) = (N, N-1), \\ \frac{1}{2} & \text{for } 1 < i < N \text{ and } j \in \{i-1, i+1\}. \end{cases}$$

We mention here that although our initial tests are for 1d problems for which direct methods can also be used, they provide a basis for gauging the performance of our approach in that most of the existing literature on AMG methods applied to Markov chain models considers these same test problems. Further, it is common when developing iterative solvers to first test a given approach on simplified 1d problems as this allows for a comprehensive analysis in terms of the asymptotic performance of the solver.

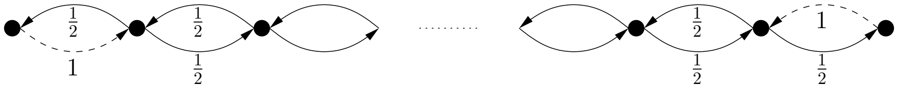


Fig. 5.1: Uniform network model on a one-dimensional grid.

N	65	129	257	513
MLE	5	5	5	5
V + AMG-GMRES	3	3	3	4
V^2 + AMG-GMRES	2	2	2	3
W + AMG-GMRES	2	2	2	3

Table 5.1: Results for the one-dimensional uniform chain of length N.

For the numerical tests we use 1d full coarsening at all levels, and we stop coarsening when the coarsest system size is 33. The caliber c for interpolation was set to $c = 2$, and we took $z = 1$ in Algorithm 2, meaning that we interpolate from direct neighbors only. The resulting operator and grid complexities are both bounded by 2 for all problem sizes. The results presented in Table 5.1 show that the number of iterations does (almost) not depend on the problem size for both methods considered, i.e., when just iterating the MLE and when using the hierarchy obtained from one single MLE to continue with AMG preconditioned GMRES on the error equation. Thus both methods exhibit the optimal scaling that one usually aims for in multigrid methods. Note that the cost of the methods using GMRES is substantially smaller, since we build up the multigrid hierarchy only once at the beginning, whereas we construct it five times for the "pure" MLE method.

In the next test, which is motivated by another example from [14], we change the central link within this example to $\epsilon = 10^{-4}$ in both directions, see Figure 5.2. The coarse grids are chosen such that this weak link is located between two coarse grid points on all levels; this is also reflected in the system sizes. The other parameters are the same as in the previous example, and the grid and operator complexities are again bounded by 2 for all sizes. The results are provided in Table 5.2. Again, we observe that our approach yields an optimal method.

We now consider a uniform two-dimensional network (see [14]), which can be viewed as the Markov chain analogue of the Laplace operator. It is defined on an equidistant grid Ω of size $N \times N$. We denote this grid in graph notation as $G_\Omega = (V_\Omega, E_\Omega)$. The entries of A are then given as

$$a_{ij} = \begin{cases} \frac{1}{d_{out}(j)}, & \text{if } (i,j) \in E_\Omega \\ 0, & \text{otherwise,} \end{cases}$$

where $d_{out}(j)$ is the number of outgoing edges of $j \in \Omega$. Figure 5.3 illustrates the two-dimensional uniform network.

In these tests, we use full-coarsening in 2d, i.e., we choose $\mathcal{C}$ to be every other grid point in both spatial dimensions, coarsen until the size of the coarsest-level system is 17×17, and define interpolation from immediate neighbors, i.e., we set $z = 1$ and $c = 4$ when computing interpolation in Algorithm 2. The grid and operator complexities are always bounded by 1.4 and 1.8, respectively. The results of our

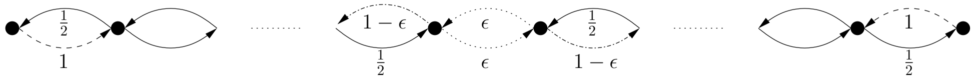


Fig. 5.2: Weak link model on a one-dimensional grid.

N	66	130	258	514
MLE	5	5	5	5
V + AMG-GMRES	3	3	3	4
V^2 + AMG-GMRES	2	2	3	3
W + AMG-GMRES	2	2	3	3

Table 5.2: Multilevel results for the one-dimensional weak-link chain of length N.

experiments are given in Table 5.3. They show that the MLE approach and the AMG-preconditioned GMRES approach with the more expensive setup (V^2 and W cycle) are again effective and scalable for this problem. For the standard V-cycle the computed AMG preconditioner is less efficient, and particularly so for the 257^2 grid. We interpret this loss in efficiency as coming from a poor choice of the initial random test vectors used by the MLE setup. The effect of this randomness is attenuated if we iterate in Algorithm 3, as we do with V^2 or W cycles.

The next Markov chain model to be considered is taken from [14, 15, 17]. It describes a state space representation of a finite capacity tandem queueing network with capacity N for each of the two queues with states (i, j), $1 \leq i, j \leq N$. There is a transition with probability λ from state (i, j) to state $(i, j+1)$, a transition from $(i, j+1)$ to $(i+1, j)$ with probability μ_1 and from $(i+1, j)$ to (i, j) with probability μ_2. The problem formulation is illustrated in Figure 5.4. We use the transition probabilities $\lambda = 11/31$ and $\mu_1 = \mu_2 = 10/31$ so that we end up with the same system matrix as the one considered in [14, 15, 17]. Note that this is a truncation of the infinite model different from the one considered in [32], resulting in a matrix that cannot be triangularized. The spectrum of the resulting transition matrix is complex; Figure 4.2 contains a plot for $N = 33$.

For the MLE setup we took the same parameters as for the uniform 2d grid, and we again used full coarsening down to a system of size 17×17. The grid complexity is bounded by 1.4, and the operator complexity by 1.6 for all sizes. The numerical results are given in Table 5.4, Again, we see that the MLE method converges rapidly to the steady state vector and also yields a very efficient preconditioner for our AMG-GMRES method. The number of iterations does not depend on the size of the problem for “pure” MLE, whereas for the AMG-preconditioned GMRES method we see some increase in the number of iterations for the largest grid size. For this example we also illustrate how the MLE method approximates the eigenvectors of B corresponding to small (in modulus) non-zero eigenvalues. Figure 5.5 shows the eigenvectors corresponding to these eigenvalues. We plot the modulus of each entry with the vectors arranged as a 2d mesh corresponding to the underlying 2d structure of the Markov chain. Table 5.5 reports the accuracy of these approximate eigenvectors upon convergence of the state vector.

To demonstrate the robustness of our approach we now turn to three more difficult test problems – a polling system model and two unstructured planar graph

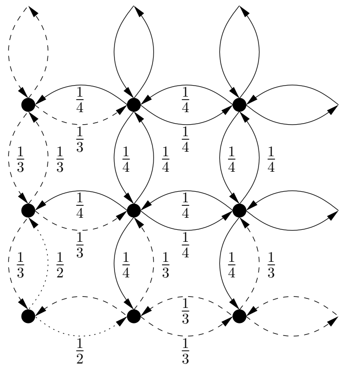


Fig. 5.3: Uniform network model on a two-dimensional grid.

N	33^2	65^2	129^2	257^2	513^2
MLE	6	6	6	6	6
V + AMG-GMRES	6	6	6	11	7
V^2 + AMG-GMRES	4	4	4	4	4
W + AMG-GMRES	4	4	4	4	4

Table 5.3: Results for the two-dimensional uniform network model on an $N \times N$ grid.

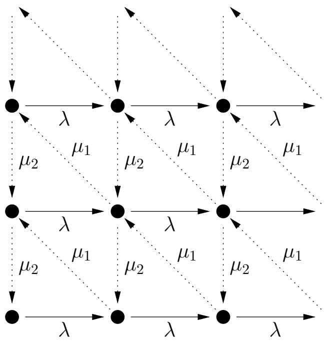


Fig. 5.4: Tandem queueing network problem.

N	33^2	65^2	129^2	257^2	513^2
MLE	6	6	6	6	6
V + AMG-GMRES	6	6	9	10	11
V^2 + AMG-GMRES	5	5	5	5	9
W + AMG-GMRES	5	5	5	5	10

Table 5.4: Results for the tandem queueing network model on an $N \times N$ grid.

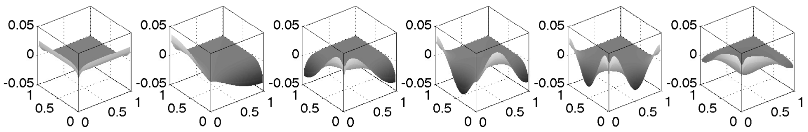


Fig. 5.5: Approximations to the eigenvectors corresponding to the smallest 6 eigenvalues of the tandem queueing network problem on a 129 × 129 grid.

i	1	2	3	4	5	6
$\|v_i^L - v_i\|_2$	$1.02E{-}8$	$9.04E{-}3$	$2.68E{-}2$	$2.84E{-}2$	$1.42E{-}1$	$3.77E{-}1$

Table 5.5: Accuracy of the eigenvectors corresponding to the smallest 6 eigenvalues of the tandem queueing network problem on a 129 × 129 grid.

problems. For these problems, the coarsening of the variables is difficult to determine in a geometric sense and thus we apply the CR-based coarsening scheme discussed earlier (see Algorithm 1) with $\theta = .7$ to select the coarse variables. We stop the CR coarsening when the coarse level system size is of the order of 100.

To begin, we consider the polling system model studied in [10, 11, 12]. In this model, two servers serve customers from K finite capacity queues, which are visited by the servers in a cyclic order. Each queue is assigned a capacity of 3, and customers arrive according to a Poisson process with rate 1.5 and are distributed with queue specific probabilities among the queues. If a server visits a nonempty queue, it serves one customer and then moves to the next queue. A server arriving at an empty queue immediately travels to the next queue. Service and traveling times are exponentially distributed with rates 1 and 10, respectively; cf. [12].

The polling system can be modeled by a hierarchical Markovian model (HMM), in terms of low-level models (LLMs) which are themselves components of a high-level model (HLM). The states of the HLM are referred to as macrostates, and those of the associated generator matrix, containing information on the interactions of the LLMs, are referred to as microstates; the macrostates define a partition of the microstates. We note that the HLM is characterized by a single matrix, in terms of Kronecker products of multiple local matrices characterizing the interaction of a given LLM with others (see [12] for additional details). This hierarchical structure can in turn be exploited to develop efficient multilevel methods for these systems, as in [12].

Here, we consider applying our bootstrap approach to this polling system model (Example 2 considered in [12]) with $K = 5$ queues, which yields a graph with 42,880 vertices. The Kronecker-based description of the polling system example gives rise to a sparse system matrix containing dense subblocks, i.e., nearly complete subgraphs. As such, a modification of our CR algorithm 1 is needed because it would otherwise produce sets of coarse variables which are too large, resulting in unacceptable grid complexities. The caliber of interpolation is chosen to be $c = 1$. This turned out to be the best choice for these problems. It produces a sparser interpolation operator and, hence, helps to control the growth in operator complexities that would result from caliber-two or more interpolation. All other parameters of the algorithm are

$(k_u = k_v \, , \, s_1 = s_2)$	(1,1)	(1,3)	(1,4)	(3,1)	(2,2)
MLE	> 50	35	41	> 50	35
V + AMG-GMRES	45^*	21	17	38^*	22
V^2 + AMG-GMRES	38^*	16	13	36^*	22
W + AMG-GMRES	36^*	16	13	30^*	19
(V + 5 AMG-GMRES) – alt	6	4	4	7	4

Table 5.6: Results for the polling system model with $K = 5$ finite capacity queues and $N = 42,880$ vertices. A * marks runs, where another setup cycle was issued after 30 iterations of preconditioned GMRES.

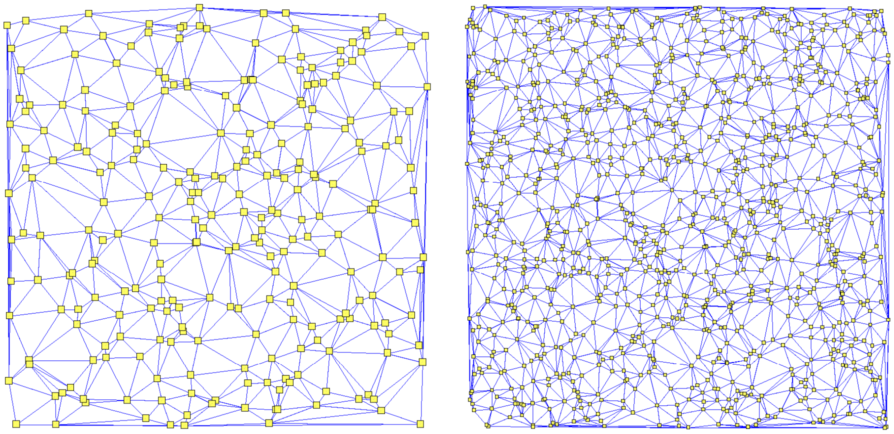
set as before. The grid and operator complexities are then bounded by 1.7 and 2.5, respectively.

For this example, we varied the number of test vectors and the number of relaxations in the MLE. We also tested a new alternating strategy where we repeat cycles consisting of one MLE setup followed by five steps of AMG-preconditioned GMRES. The results are reported in Table 5.6. In the last row, which gives the results for the alternating strategy, the number reported refers to the number of cycles. So 4 iterations means 4 MLE setups and a total of $4 \times 5 = 20$ preconditioned GMRES iterations.

We see that for the "pure" MLE approach (first row), applying more relaxation to fewer test vectors dramatically improves the performance of the overall method, whereas applying fewer relaxations to more test vectors is not as effective. This observed behavior is perhaps expected since the MLE hierarchy needs only to approximate the single state vector well, not the entire lower end of the spectrum as typically required of AMG correction schemes. Performing the MLE only once and then using the AMG preconditioned GMRES on the error equation improves the performance of the method significantly. Computing the steady state vector up to the prescribed tolerance requires only a single MLE $V(4,4)$-cycle followed by 17 iterations of AMG preconditioned GMRES. Both the V^2- and W- setup cycles further reduce the number of GMRES steps needed to reach the stopping tolerance. Here, the additional work invested on coarse levels in the MLE V^2- and W-cycles produces a more accurate initial guess and a better multigrid hierarchy for the AMG preconditioned GMRES iteration. All these approaches show a similar behavior as "pure" MLE when the number of test vectors and the relaxation steps are varied. The alternating strategy works well for this problem and appears less sensitive to the choices of k and s.

We note that the multilevel approach developed for this system in [12], which makes explicit use of the HMM structure in its coarsening process, requires 92 iterations, each of which is similar to what we would call a $V(1,1)$ MLE setup in the context of this paper. So, as far as the number of iterations is concerned, and with the right choice of parameters, our approach gives methods which require quite less MLE setups than in [12]. However, as opposed to [12], we do not exploit the Kronecker structure in the matrix vector multiplications, so we expect the timings of our method to be worse.

Our remaining numerical experiments are for random walks on two types of unstructured planar graphs arising from Delaunay triangulations induced by N randomly chosen vertices in $[0,1] \times [0,1]$, see [17]. In the first case, the transition probabilities



(a) Random planar graph with $N = 256$ (b) Random planar graph with $N = 1024$

Fig. 5.6: Network resulting from Delaunay-triangulations of N random vertices in the unit square.

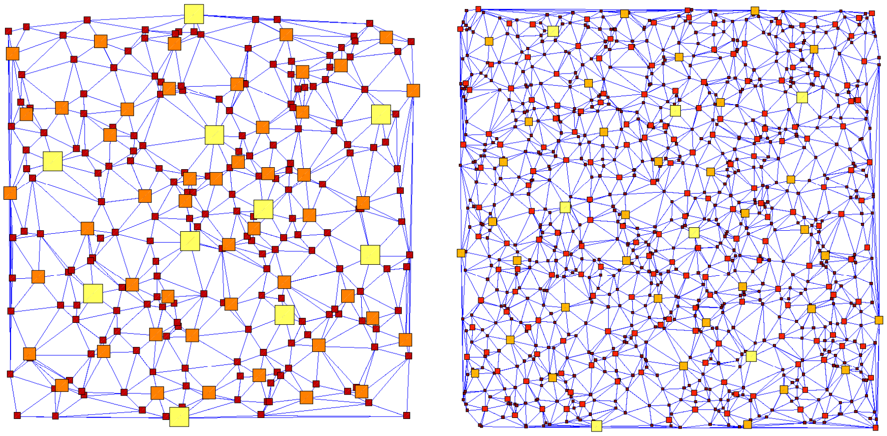
in the network describing the random walk are defined as the inverse values of the number of outgoing edges of each vertex, similar to the uniform network example. In the second case, unstructured planar graphs with similarly weighted but (partly) uni-directional edges are considered. Our construction follows the procedure suggested in [17], i.e., we chose an unmarked random triangle, mark it as “deletable” and its neighbors as “undeletable”. This process is repeated until all triangles are marked and in each triangle marked as “deletable” a random edge is made uni-directional while the two others remain bi-directional. This yields a non-symmetric and irreducible transition matrix with complex spectrum [17]. In Figure 5.6, we show two examples of the graphs of such networks, one with $N = 256$ and one with $N = 1024$ vertices.

In the MLE setup we take $k_v = k_u = 6$, the number of relaxations in the MLE setup is $s_1 = s_2 = 5$ and the caliber of the interpolation is $c = 1$. The coarsening is done via CR as given in Algorithm 1. Figure 5.7 depicts the multilevel coarsening obtained using the CR-based approach. Here, the larger points are those that appear on successively coarser grids. For all tests the grid and operator complexities were both bounded by 1.4.

In Table 5.7, we report the number of iterations of the alternating scheme needed to achieve our chosen stopping criteria for the two types of unstructured planar graph models for different problem sizes. As before, in the alternating methods we perform five iterations of AMG preconditioned GMRES for the additive correction process in between two MLE setup steps. We report the number of a complete cycle, i.e., setup plus up to 5 GMRES-iterations, here.

We note that for these problems the maximum number of outgoing edges from a given grid point increases with increasing problem sizes, making it more difficult to coarsen the problem algebraically via compatible relaxation. From this aspect it is not clear whether we can safely interpret the systems for different sizes as instances of the same problem. So aiming at iteration numbers to be independent of the system size might be overambitious.

Nonetheless, our results show that our algebraic coarsening scheme is effective



(a) Random planar graph, $N = 256$ (b) Random planar graph, $N = 1024$

Fig. 5.7: Coarsening of networks from Figure 5.6 using compatible relaxation.

N # levels	1024 3 (3)	4096 5 (5)	16384 7 (6)	65536 8 (8)
MLE	26 (24)	> 50 (> 50)	> 50 (> 50)	> 50 (> 50)
V + AMG-GMRES	10 (11)	13 (16)	19 (21)	21 (43*)
V^2 + AMG-GMRES	9 (9)	12 (14)	19 (20)	27 (29)
W + AMG-GMRES	9 (9)	13 (15)	22 (21)	29 (30*)
(V + 5 AMG-GMRES) – alt	2 (2)	3 (3)	3 (5)	4 (6)
(V^2 + 5 AMG-GMRES) – alt	2 (2)	3 (3)	3 (3)	5 (5)
(W + 5 AMG-GMRES) – alt	2 (2)	3 (3)	3 (4)	5 (6)

Table 5.7: Results for the non-symmetric (symmetric) unstructured planar graph models. A * marks runs, where another setup cycle was issued after 30 iterations of preconditioned GMRES.

for these systems and produces efficient solvers. For all the variants we tested, the number of iterations increases by a factor of 2 to 3 only when we increase the problem size by a factor of 64. V cycle MLE setups seem to be sufficient for the bidirectional case, whereas in the unidirectional case the V^2 setups performed best. Our results demonstrate the efficiency of the methods based on MLE setups with additive correction via preconditioned GMRES in that plain (unsmoothed) aggregation-based linear solvers, such as our approach with caliber-one interpolation, with W-cycles are known not to scale for Poisson's equation.

We finish this section with a qualitative comparison of our results with the algebraic multilevel method from [17, 15, 14]. We first note that the idea of re-using the expensive setup to construct a preconditioner for a cheap Krylov subspace extraction process is new to our approach. In [17], for the structured 2d problems, the number of iterations increases noticeably with the problem size, even for what is called "optimized RAMA+ cycles". The situation is similar for the unstructured planar graph problems, although for these problems the results of our bootstrap approach show a mild dependence on the problem size. The approaches described in [15, 14] have a similar scaling behavior than our method, but they tend to result in either higher iteration counts or higher operator complexities. Wall clock time comparisons between the various approaches are beyond the scope of this paper, since those will highly depend on issues of implementation and the hardware used.

We note that there are notoriously difficult Markov chain problems where the sizes of the entries of the steady state vector vary from order 1 to well below machine accuracy. One example is the birth-death chain as described in [14]. Without giving details, we just mention that the V + AMG-GMRES variant of our method worked well up to large problem sizes in the sense that we reached the stopping criterion (5.1) after a few iterations, independently of the problem size. For larger problem sizes, however, the approximation to the steady state vector contained (tiny) negative entries which is, of course, undesirable. In this sense our approach fails for problems of this kind as do many other general purpose Markov chain methods.

6. Conclusions.

The proposed bootstrap algebraic multilevel method produces an effective multilevel eigensolver for the Markov chain test problems considered. We have proposed and implemented two new ideas: First, the construction and use of a bootstrap multilevel eigensolver which improves a given approximation to the steady state vector and establishes a multigrid hierarchy. Second, the use of this multigrid hierarchy in an AMG preconditioner for the GMRES iteration on the error equation, thus producing an additive correction to such vector. Both ideas, separately or combined can be incorporated into any given multilevel method used for solving Markov chain problems or other problems targeting smooth eigenvectors. The accurate representation of the near-kernel of the fine-level operator B on coarser levels that results from our bootstrap AMG setup yields a very effective preconditioner to the GMRES method. A benefit of the proposed method over other existing multilevel methods for Markov chains is that we do not require any special processing of the coarse-level systems to ensure that stochastic properties of the fine-level system are maintained there. Note that in some situations maintaining stochasticity is perhaps preferable, for example to avoid negative entries in the computed steady state vector.

Our numerical examples show that our approach is efficient for a variety of test problems, often yielding optimal methods. At present, however, sometimes some parameters of the approach have to be specifically tuned.

We mention that the developed approach is not restricted to Markov chain problems. The MLE can be applied to other eigenvalue problems [24] and the alternating approach which uses both the MLE and the AMG preconditioned GMRES iterations can be used as a general non-linear AMG solver for linear systems of equations. In such a scheme, the basic idea is to solve the system for the given right hand side and the homogeneous system simultaneously and then use the error resulting from solving the latter to redefine the solver for the former whenever its performance degrades. Such a scheme is actually ideal for Markov chains in that the aim is to solve the homogenous system and hence computing the evolving error is actually equivalent to computing the desired solution.

Acknowledgement. We would like to thank two anonymous referees whose comments contributed to improving a first version of this paper. We also thank Peter Buchholz for giving us the data for the polling system example.

REFERENCES

[1] A. Berman and R. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, vol. 9 of Classics in Appl. Math., SIAM, Philadelphia, 1994.

[2] A. Brandt, *General highly accurate algebraic coarsening*, Electron. Trans. Numer. Anal., 10 (2000), pp. 1–20.

[3] ———, *Multiscale scientific computation: Review 2001*, in Multiscale and Multiresolution Methods: Theory and Applications, T.J. Barth, T.F. Chan, and R. Haimes, eds., Springer Verlag, Heidelberg, 2001, pp. 1–96. Also available at www.wisdom.weizmann.ac.il/∼achi/review00.ps.

[4] ———, *Principles of systematic upscaling*, in Bridging the Scales in Science and Engineering, J. Fish, ed., Oxford University Press, Oxford, 2008, pp. 193–215.

[5] A. Brandt, J. Brannick, K. Kahl, and I. Livshits, *Bootstrap AMG*, SIAM J. Sci. Comput., 33 (2011), pp. 612–632.

[6] A. Brandt and I. Livshits, *A multiscale method for finding many eigenfunctions of pde operators*. In preparation.

[7] J. Brannick, *Adaptive Algebraic Multigrid Coarsening Strategies*, PhD thesis, Department of Applied Mathematics, University of Colorado at Boulder, 2005.

[8] J. Brannick and R. Falgout, *Compatible relaxation and coarsening in algebraic multigrid*, SIAM J. Sci. Comput., 32 (2010), pp. 1393–1416.

[9] 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.

[10] P. Buchholz, *Multi level solutions for structured Markov chains*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 342–357.

[11] P. Buchholz and T. Dayar, *Comparison of multilevel methods for Kronecker-based Markovian representations*, Computing, 73 (2004), pp. 349–371.

[12] ———, *On the convergence of a class of multilevel methods for large, sparse Markov chains*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1025–1049.

[13] F. Chatelin and W. Miranker, *Acceleration by aggregation of successive approximation methods*, Linear Algebra Appl., 43 (1982), pp. 17–47.

[14] H. De Sterck, T. Manteuffel, S. McCormick, K. Miller, J. Pearson, J. Ruge, and G. Sanders, *Smoothed aggregation multigrid for Markov chains*, SIAM J. Sci. Comput., 32 (2010), pp. 40–61.

[15] H. De Sterck, T.A. Manteuffel, S.F. McCormick, K. Miller, J. Ruge, and G. Sanders, *Algebraic multigrid for Markov chains.*, SIAM J. Sci. Comput., 32 (2010), pp. 544–562.

[16] H. De Sterck, T. Manteuffel, S. 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.

[17] H. De Sterck, K. Miller, G. Sanders, and M. Winlaw, *Recursively accelerated multilevel aggregation for Markov chains*, SIAM J. Sci. Comput., 32 (2010), pp. 1652–1671.

[18] R. Falgout and P. Vassilevski, *On generalizing the AMG framework*, SIAM J. Numer. Anal., 42 (2004), pp. 1669–1693.

[19] R. Falgout, P. Vassilevski, and L. Zikatanov, *On two–grid convergence estimates*, Numer. Linear Algebra Appl., 12 (2005), pp. 471–494.

[20] M. Haviv, *Aggregation/disaggregation methods for computing the stationary distribution of a Markov chain*, SIAM J. Numer. Anal., 24 (1987), pp. 952–966.

[21] K. Hayami and M. Sugihara, *A geometric view of Krylov subspace methods on singular systems*, Numer. Linear Algebra Appl., (2010 (published online)). doi: 10.1002/nla.737.

[22] S. Kaczmarz, *Angenäherte Auflösung von Systemen linearer Gleichungen*, Bull. Internat. Acad. Polon. Sci. Lettres A, (1937), pp. 335–357.

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

[24] D. Kushnir, M. Galun, and A. Brandt, *Efficient multilevel eigensolvers with applications to data analysis tasks*, IEEE Transactions on Pattern Analysis and Machine Intelligence - PAMI, 32 (2010), pp. 1377–1391.

[25] O. Livne, *Coarsening by compatible relaxation*, Numer. Linear Algebra Appl., 11 (2004), pp. 205–227.

[26] I. Livshits, *One-dimensional algorithm for finding eigenbasis of the Schrödinger operator*, SIAM J. Sci. Comput., 30 (2008), pp. 416–440.

[27] J. Mandel and B. Sekerka, *A local convergence proof for the iterative aggregation method*, Linear Algebra Appl., 51 (1983), pp. 163–172.

[28] I. Marek and I. Pultarova, *A note on local and global convergence analysis of iterative aggregation/disaggregation methods*, Linear Algebra Appl., 413 (2006), pp. 327–341.

[29] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[30] E. Seneta, *Non-negative Matrices and Markov Chains*, Springer Series in Statistics, Springer, Heidelberg, New York, 2nd rev. ed., 1981.

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

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

[33] Y. Takahashi, *A lumping method for numerical calculations of sationary distributions of Markov chains*, Tech. Report B-18, Department of Information Sciences, Tokyo Institute of Technology, 1975.

[34] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, San Diego (CA), 2001.

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

[36] H. Walker and P. Brown, *GMRES on (nearly) singular systems*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 37–51.