

 Open access • Journal Article • DOI:10.1137/060651161

## On the Convergence of a Class of Multilevel Methods for Large Sparse Markov Chains

— [Source link](#) 

Peter Buchholz, Tug caron, ru Dayar

**Published on:** 01 Oct 2007 - SIAM Journal on Matrix Analysis and Applications (Society for Industrial and Applied Mathematics)

**Topics:** Markov chain, Markov process, Multigrid method and Iterative method

Related papers:

- [Introduction to the Numerical Solution of Markov Chains](#)
- [Comparison of Multilevel Methods for Kronecker-based Markovian Representations](#)
- [Efficient descriptor-vector multiplications in stochastic automata networks](#)
- [Block SOR Preconditioned Projection Methods for Kronecker Structured Markovian Representations](#)
- [Hierarchical structuring of superposed GSPNs](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/on-the-convergence-of-a-class-of-multilevel-methods-for-1dhmwn5q09>

## ON THE CONVERGENCE OF A CLASS OF MULTILEVEL METHODS FOR LARGE SPARSE MARKOV CHAINS\*

PETER BUCHHOLZ<sup>†</sup> AND TUĞRUL DAYAR<sup>‡</sup>

**Abstract.** This paper investigates the theory behind the steady state analysis of large sparse Markov chains with a recently proposed class of multilevel methods using concepts from algebraic multigrid and iterative aggregation-disaggregation. The motivation is to better understand the convergence characteristics of the class of multilevel methods and to have a clearer formulation that will aid their implementation. In doing this, restriction (or aggregation) and prolongation (or disaggregation) operators of multigrid are used, and the Kronecker-based approach for hierarchical Markovian models is employed, since it suggests a natural and compact definition of grids (or levels). However, the formalism used to describe the class of multilevel methods for large sparse Markov chains has no influence on the theoretical results derived.

**Key words.** Markov chains, multigrid, aggregation-disaggregation, Kronecker-based numerical techniques, multilevel methods

**AMS subject classifications.** 60J27, 65F50, 65F10, 65B99, 65F15, 65F05, 15A72

**DOI.** 10.1137/060651161

**1. Introduction.** Markov chains (MCs) are a popular mathematical tool to model systems from various application areas like engineering, computer science, biology, or economics. For system analysis often one needs the steady state distribution of the MC to compute result measures for the modeled system. The problem in the continuous-time case is then to solve

$$(1.1) \quad \pi Q = 0 \quad \text{subject to} \quad \pi e = 1 \quad \text{and} \quad \pi \geq 0,$$

where  $Q$  is the infinitesimal generator or generator matrix of the continuous-time Markov chain (CTMC) underlying the modeled system,  $\pi$  is its (row) stationary probability vector, and  $e$  is the column vector of ones of appropriate length. We assume that the state space is finite and contains  $n$  states numbered starting from 0;  $Q$  is irreducible, implying  $\pi > 0$ ; and  $\pi$  is also the steady state vector. The nonnegative off-diagonal elements of  $Q$  represent exponential transition rates between different states, and its diagonal elements are negated row sums of its off-diagonal elements. Hence,  $Q$  has row sums of zero (i.e.,  $Qe = 0$ ) and is a singular matrix of rank  $(n - 1)$ , and (1.1) represents a homogeneous linear system subject to a normalization condition, so that its solution vector  $\pi$  can be uniquely determined [29, Chap. 1]. At this level, states of the CTMC are numbered by consecutive integers. However, in almost all applications CTMCs result from some high level model like a stochastic automata network, a queueing network, or a stochastic Petri net. In all these cases, the state space is multidimensional and is mapped for solution onto a set of consecutive

---

\*Received by the editors January 30, 2006; accepted for publication (in revised form) by D. A. Bini March 13, 2007; published electronically October 31, 2007. Part of this work has been carried out through a grant from the Alexander von Humboldt Foundation at Dortmund University and grant TÜBA-GEBİP from the Turkish Academy of Sciences.

<http://www.siam.org/journals/simax/29-3/65116.html>

<sup>†</sup>Informatik IV, Universität Dortmund, D-44221 Dortmund, Germany (peter.bucholz@cs.uni-dortmund.de).

<sup>‡</sup>Department of Computer Engineering, Bilkent University, TR-06800 Bilkent, Ankara, Turkey (tugrul@cs.bilkent.edu.tr.).

integers. The multidimensional structure can be exploited in a compact representation of  $Q$  and can also be exploited to develop fast solvers for the computation of  $\pi$ .

Practical problems arise due to the state space size of MCs resulting from applications, which often grows exponentially with the number of components in the specification. A popular way of dealing with this so-called state space explosion problem is to employ Kronecker- (or tensor)-based representations of  $Q$ , which remain compact even for considerably large state spaces. In the Kronecker-based approach, the system of interest is modeled so that it is formed of smaller interacting components, and its larger underlying generator matrix is neither generated nor stored but rather represented using Kronecker products of the smaller component matrices. This introduces significant storage savings at the expense of some overhead in the solution phase. In order to analyze large structured Markovian models efficiently, various algorithms for vector-Kronecker product multiplication are devised [14, 16, 17] and used as kernels in iterative solution methods. The most effective solvers known for Kronecker representations of dimension four or larger are multilevel (ML) methods [11] and block successive over relaxation (BSOR) preconditioned projection methods [12] as recently shown empirically by comparing different solvers on a large number of hierarchical Markovian models (HMMs). Unfortunately, solvers using BSOR [10, 31] are sensitive to the ordering of components, the block partitionings chosen, and the amount of fill-in in the factorized diagonal blocks, so that a robust implementation for arbitrary models is difficult to achieve.

In this paper, we investigate the theory behind the steady state analysis of large sparse MCs with the class of ML methods proposed in [11] using concepts from algebraic multigrid (AMG) [6, Chap. 8], [24] and iterative aggregation-disaggregation (IAD) [29, Chap. 6]. Our motivation is to better understand the convergence characteristics of the class of ML methods and to have a clearer formulation that will aid their implementation. Convergence analysis of a two-level IAD method for MCs and its equivalence to AMG is provided in [20]. Another paper that investigates the convergence of a two-level IAD method for MCs using concepts from multigrid is [21]. Here we consider more than two levels, different types of smoothers, different types of cycles, and different orders of aggregation. In doing this, we use restriction (or aggregation) and prolongation (or disaggregation) operators of multigrid, and employ the Kronecker-based approach for HMMs in [11]. This is for three reasons. First, the hierarchy present in the HMM description suggests a natural definition of grids (or levels). This simplifies the description of the class of ML methods. Second, with the HMM description, one can store the aggregated MC at each level during implementation compactly in Kronecker form. It is not clear how the same effect can be achieved with an MC in sparse format (see [19]). Third, Kronecker operations to define large MCs underlying structured representations are natural for many application areas since complex systems are usually composed of interacting components. Almost all MCs resulting from applications can be represented as HMMs [15], and this representation can be derived from the specification using an appropriate modeling tool [1]. Otherwise, the HMM formalism used in this paper to describe the class of ML methods for large sparse MCs has no influence on the theoretical results derived. In general, our approach can be applied for any irreducible MC with a set of nested partitions defined on its state space.

The next section introduces the Kronecker-based description of CTMCs underlying HMMs on a small example. The third section presents the proposed class of ML methods for HMMs with multiple macrostates and discusses how they work. The

fourth section provides results on the convergence of ML methods. The fifth section illustrates the convergence behavior of the class of ML methods on two larger problems. The sixth section concludes the paper.

In what follows, calligraphic uppercase letters denote sets and lists, uppercase letters denote matrices, sets are defined using curly brackets, lists are defined using square brackets, matrices (and vectors) are defined using brackets,  $|\cdot|$  denotes the cardinality of a set (list) when its argument is a set (list),  $\emptyset$  denotes the empty set,  $\|\cdot\|$  denotes the norm of a vector,  $\cdot^T$  denotes the transpose operator, and  $\text{diag}(\cdot)$  represents a diagonal matrix having its vector argument along its diagonal.

**2. Hierarchical Markovian models.** HMMs are defined using the operations of Kronecker product and Kronecker sum [32]. First we introduce these operations.

**DEFINITION 2.1.** *The Kronecker product of two matrices  $X \in \mathbb{R}^{r_X \times c_X}$  and  $Y \in \mathbb{R}^{r_Y \times c_Y}$  is written as  $X \otimes Y$  and yields a block matrix  $Z$  with  $r_X \times c_X$  blocks each of size  $r_Y \times c_Y$ , where the  $(i, j)$ th block equals  $x(i, j)Y$  for  $i = 0, \dots, r_X - 1, j = 0, \dots, c_X - 1$ .*

*The Kronecker sum of two square matrices  $U \in \mathbb{R}^{r_U \times r_U}$  and  $V \in \mathbb{R}^{r_V \times r_V}$  is written as  $U \oplus V$  and yields the matrix  $W \in \mathbb{R}^{r_U r_V \times r_U r_V}$ , which is defined in terms of two Kronecker products as  $W = U \otimes I_{r_V} + I_{r_U} \otimes V$ . Here  $I_{r_U}$  and  $I_{r_V}$  denote identity matrices of orders  $r_U$  and  $r_V$ , respectively.*

Both Kronecker product and Kronecker sum are associative and defined for more than two matrices. For further properties of Kronecker operations, see [29].

HMMs consist of multiple low level models (LLMs) which can be perceived as components, and a high level model (HLM) that defines how LLMs interact. The HLM is characterized by a single matrix, whereas each LLM is characterized by multiple matrices that define its interaction with other LLMs. The order of each LLM matrix is equal to the number of states of the particular component to which the matrix belongs. A formal definition of HMMs can be found in [8, pp. 387–390]. Here we extend the definition from [12] and introduce HMMs on an example. An HMM describes a CTMC and its generator matrix  $Q$ . Since we consider the steady state analysis of irreducible finite CTMCs,  $Q$  is sufficient to characterize the CTMC. We name the states of the HLM as *macrostates*, those of  $Q$  as *microstates*, and remark that macrostates define a partition of the microstates.

**DEFINITION 2.2.** *In a given HMM, let  $K$  be the number of LLMs,  $\mathcal{S}^{(k)} = \{0, 1, \dots, |\mathcal{S}^{(k)}| - 1\}$  be the state space of LLM  $k$  for  $k = 1, 2, \dots, K$ ,  $\mathcal{S}^{(K+1)} = \{0, 1, \dots, |\mathcal{S}^{(K+1)}| - 1\}$  be the state space of the HLM,  $\mathcal{S}_j^{(k)}$  be the partition of states of LLM  $k$  mapped to macrostate  $j \in \mathcal{S}^{(K+1)}$  so that  $\cup_j \mathcal{S}_j^{(k)} = \mathcal{S}^{(k)}$  and  $\mathcal{S}_i^{(k)} \cap \mathcal{S}_j^{(k)} = \emptyset$  when  $i \neq j$ ,  $t_0$  be a local transition (one per LLM),  $\mathcal{T}_{i,j}$  be the set of LLM nonlocal transitions in element  $(i, j)$  of the HLM matrix, and  $D_j$  be the diagonal correction matrix that sums the rows of  $Q$  corresponding to macrostate  $j$  to zero. Then the diagonal block  $(j, j)$  of  $Q$  corresponding to element  $(j, j)$  of the HLM matrix is given by*

$$(2.1) \quad Q(j, j) = \bigoplus_{k=1}^K Q_{t_0}^{(k)}(\mathcal{S}_j^{(k)}, \mathcal{S}_j^{(k)}) + \sum_{t_e \in \mathcal{T}_{j,j}} \bigotimes_{k=1}^K Q_{t_e}^{(k)}(\mathcal{S}_j^{(k)}, \mathcal{S}_j^{(k)}) + D_j,$$

and, when there are multiple macrostates, the off-diagonal block  $(i, j)$  of  $Q$  corresponding to element  $(i, j)$  of the HLM matrix is given by

$$(2.2) \quad Q(i, j) = \sum_{t_e \in \mathcal{T}_{i,j}} \bigotimes_{k=1}^K Q_{t_e}^{(k)}(\mathcal{S}_i^{(k)}, \mathcal{S}_j^{(k)}),$$

where  $Q_{t_e}^{(k)}(\mathcal{S}_i^{(k)}, \mathcal{S}_j^{(k)})$  is a submatrix of order  $(|\mathcal{S}_i^{(k)}| \times |\mathcal{S}_j^{(k)}|)$  including all transitions<sup>1</sup> between states from  $\mathcal{S}_i^{(k)}$  and  $\mathcal{S}_j^{(k)}$  for LLM  $k$  under  $t_e$ .

We remark that  $D_j$  can be expressed as a sum of Kronecker products, as follows.

PROPOSITION 2.3. *If  $D_j$  is the diagonal correction matrix that sums the rows of  $Q$  corresponding to macrostate  $j$  to zero, then*

$$D_j = - \bigoplus_{k=1}^K \text{diag}(Q_{t_0}^{(k)}(\mathcal{S}_j^{(k)}, \mathcal{S}_j^{(k)})e) \\ - \sum_{i \in \mathcal{S}^{(K+1)}} \sum_{t_e \in \mathcal{T}_{j,i}} \bigotimes_{k=1}^K \text{diag}(Q_{t_e}^{(k)}(\mathcal{S}_j^{(k)}, \mathcal{S}_i^{(k)})e) \quad \text{for } j \in \mathcal{S}^{(K+1)}.$$

In order to enable the efficient implementation of numerical solvers, most of the time  $D_j$  is precomputed and stored explicitly as a vector. However, the off-diagonal part of  $Q$  is never stored explicitly, but is represented in memory through Definition 2.2 as sums of Kronecker products of small matrices, which are generally very sparse and therefore held in row sparse format [29, pp. 80–81].

For a definition of mapping used in the next proposition, see, for instance, [30, pp. 192–197].

PROPOSITION 2.4. *When the multidimensional states of  $Q$  are identified by the tuple  $(s^{(1)}, s^{(2)}, \dots, s^{(K)}, j)$ , where  $s^{(k)} \in \mathcal{S}^{(k)}$  is the state of LLM  $k$  for  $k = 1, 2, \dots, K$  and  $j \in \mathcal{S}^{(K+1)}$  is the corresponding macrostate, the Kronecker product operation orders the state space of  $Q$  lexicographically, where each state is linearized through the one-to-one onto mapping*

$$(s^{(1)}, s^{(2)}, \dots, s^{(K)}, j) \longleftrightarrow \sum_{k=1}^K s^{(k)} \prod_{l=k+1}^K |\mathcal{S}_j^{(l)}| + \sum_{i=0}^{j-1} \prod_{k=1}^K |\mathcal{S}_i^{(k)}| \in \{0, 1, \dots, n-1\},$$

where  $n = \sum_{j=0}^{|\mathcal{S}^{(K+1)}|-1} \prod_{k=1}^K |\mathcal{S}_j^{(k)}|$ .

The microstates corresponding to each macrostate result from the Cartesian (or cross) product [30, pp. 123–124] of the state space partitions of LLMs that are mapped to that particular macrostate. In contrast to other representations of CTMCs using Kronecker operators (e.g., [29, Chap. 9]), HMMs are generated in such a way that only reachable states are considered [7, 8]. Note that each macrostate in an HLM may have a different number of microstates if LLMs have partitioned state spaces. When there are multiple macrostates,  $Q$  is effectively a block matrix having as many blocks in each dimension as  $|\mathcal{S}^{(K+1)}|$ . The diagonal and off-diagonal blocks of this partitioning are respectively the  $Q_{j,j}$  and  $Q_{i,j}$  matrices defined by (2.1) and (2.2). Due to the Kronecker structure suggested by Definitions 2.1 and 2.2, each of the blocks defined by the HLM matrix is also formed of blocks, and hence HMMs have nested block partitionings [10, 31].

Now, let us consider a small example HMM which gives rise to a  $(5 \times 5)$  CTMC. In [13, sec. 5], we step through the ML method on this example, which is chosen deliberately to be very small. After this small example, we briefly present two larger examples which will be used in section 5 to show the convergence behavior of the class of ML methods.

<sup>1</sup>In this section, the concept of transition is used to refer to those that take place at the HMM level, except for this case, where it is used to refer to nonzeros in a matrix at the state level.

TABLE 2.1  
 Mapping between LLM states and HLM states in Example 1.

| LLM 1 | LLM 2 | HLM | # of microstates |
|-------|-------|-----|------------------|
| {0,1} | {0,1} | {0} | 2 . 2 = 4        |
| {2}   | {2}   | {1} | 1 . 1 = 1        |

Example 1. The HLM of two states describes the interaction among two LLMs (i.e.,  $K = 2$ ), each of which has three states. All states are numbered starting from 0. The mapping between LLM states and HLM states and the number of microstates are given in Table 2.1. In this example,  $Q$  has the following states in its rows and columns:  $\{0, 1\} \times \{0, 1\} \times \{0\} \cup \{2\} \times \{2\} \times \{1\} = \{(0, 0, 0), (0, 1, 0), (1, 0, 0), (1, 1, 0), (2, 2, 1)\}$ . One can think of these five states written in the given order as corresponding to the integers 0 through 4.

The values of the nonzeros in  $Q$  are determined by the rates of the transitions and their associated matrices. In Example 1, two transitions denoted by  $t_0$  and  $t_1$  take place and affect the LLMs. Transition  $t_0$  covers all local transitions inside the LLMs, whereas transition  $t_1$  is captured by the following  $(2 \times 2)$  HLM matrix:

$$(2.3) \quad \begin{array}{c} 0 \quad 1 \\ 0 \quad \left( \begin{array}{c|c} & t_1 \\ \hline t_1 & \end{array} \right) \\ 1 \end{array}.$$

To each transition in the HLM matrix corresponds a Kronecker product of two (i.e., number of LLMs,  $K$ ) LLM matrices. The matrices associated with those LLMs that do not participate in a transition are identity. LLM 1 participates in  $t_1$  with the matrix  $Q_{t_1}^{(1)}$ , and LLM 2 participates in  $t_1$  with the matrix  $Q_{t_1}^{(2)}$ . In this example, the transition  $t_1$  affects exactly two LLMs.

Other than Kronecker products due to the transitions in (2.3), there is a Kronecker sum implicitly associated with each diagonal element of the HLM matrix. Each Kronecker sum is formed of two (i.e.,  $K$ ) LLM matrices corresponding to local transition  $t_0$ . In the HLM matrix of (2.3), there does not exist any nonlocal transition along the diagonal. In general, this need not be so, as can be seen from Definition 2.2.

In our example, the second term in (2.1) is missing, and the matrices associated with  $t_0$  and  $t_1$  are given by

$$Q_{t_0}^{(1)} = \left( \begin{array}{cc|c} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right), \quad Q_{t_1}^{(1)} = \left( \begin{array}{cc|c} 0 & 0 & 2 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{array} \right), \quad Q_{t_0}^{(2)} = \left( \begin{array}{cc|c} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right),$$

$$Q_{t_1}^{(2)} = \left( \begin{array}{cc|c} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{array} \right).$$

Then the CTMC underlying the HMM can be obtained from

$$(2.4) \quad Q = \left( \begin{array}{cc|cc} Q_{t_0}^{(1)}(\{0, 1\}, \{0, 1\}) \oplus Q_{t_0}^{(2)}(\{0, 1\}, \{0, 1\}) & & Q_{t_1}^{(1)}(\{0, 1\}, \{2\}) \otimes Q_{t_1}^{(2)}(\{0, 1\}, \{2\}) & \\ & & & \\ Q_{t_1}^{(1)}(\{2\}, \{0, 1\}) \otimes Q_{t_1}^{(2)}(\{2\}, \{0, 1\}) & & Q_{t_0}^{(1)}(\{2\}, \{2\}) \oplus Q_{t_0}^{(2)}(\{2\}, \{2\}) & \end{array} \right) + D,$$

where  $D$  is the diagonal correction matrix that sums the rows of  $Q$  to zero; hence,

written explicitly, we have

$$(2.5) \quad Q = \left( \begin{array}{cccc|c} -4 & 1 & 1 & 0 & 2 \\ 1 & -2 & 0 & 1 & 0 \\ 1 & 0 & -3 & 1 & 1 \\ 0 & 1 & 1 & -2 & 0 \\ \hline 1 & 0 & 0 & 0 & -1 \end{array} \right).$$

If we neglect the diagonal of  $Q$ , which is handled separately, from Definition 2.2 it follows that each nonzero element of the HLM matrix is essentially a sum of Kronecker products, since Kronecker sums can be expressed as sums of Kronecker products. This has a very nice implication for the choice of grids in the proposed ML method when LLM aggregation is used in forming the coarse grids. LLMs 1 through  $K$  and the HLM define the least coarse (in other words, the finest) grid. This grid is  $Q$  and in our example has five states. Regarding the intermediate grids, let us assume that LLMs are aggregated starting from 1 up to  $K$ . Thus LLMs 2 through  $K$  and the HLM define the first coarser grid when LLM 1 is aggregated. In our example, this grid has the states in  $\{(0,0), (1,0), (2,1)\}$ , where the first state in each tuple is an LLM 2 state and the second state in each tuple is the corresponding HLM state. The HLM and LLMs 3 through  $K$  define the second coarser grid when LLMs 1 and 2 are aggregated. In our example, this grid is the coarsest grid corresponding to the HLM and has the states  $\{(0), (1)\}$ . There are no other LLMs left to be aggregated in our example; otherwise aggregation continues with the next LLM.

Now, let us concentrate on the sizes of the grids defined by the LLMs and the HLM for the assumed order in which LLMs are aggregated. In Example 1, the grids defined in this way by LLMs 1–2 and the HLM, by LLM 2 and the HLM, and by the HLM alone have respectively the sizes  $(5 \times 5)$ ,  $(3 \times 3)$ ,  $(2 \times 2)$  (see Table 2.1 and (2.1)–(2.2)). Clearly, we are not limited to aggregating LLMs in the order 1 through  $K$ , and can consider other orderings. The number of possible orderings of LLMs equals  $K!$ .

*Example 2.* The second example we consider is a polling system. Two servers serve customers from  $K$  finite capacity queues, which are visited by the servers in cyclic order. We assume that each queue has 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. Each LLM describes one queue, and macrostates for this model are defined according to the number of servers serving customers at a queue or traveling to the next queue. For each LLM we obtain 20 states partitioned into three subsets. The complete model has  $\binom{K+1}{K-1}$  macrostates. Table 2.2 contains the number of microstates for different values of  $K$ .

*Example 3.* The third example describes an availability model with  $K$  LLMs. Each LLM consists of two active components and a cold spare which becomes active when a component fails. Time to failure is exponentially distributed with mean  $10^k$  for the components of the  $k$ th LLM. With 90% probability a failure is local, requiring a local repair with an exponential duration and mean  $10^{-k+1}$  for the  $k$ th component. With a probability of 10%, a failure has to be repaired by a global repair unit; repair times are identical to the local case. The system has one global repair unit which repairs failed components with preemptive priority such that components from the

TABLE 2.2

Number of macrostates and state space sizes versus number of LLMs in Examples 2 and 3.

| $K$ | Polling example         |           | Availability example    |            |
|-----|-------------------------|-----------|-------------------------|------------|
|     | $ \mathcal{S}^{(K+1)} $ | $n$       | $ \mathcal{S}^{(K+1)} $ | $n$        |
| 2   | –                       | –         | 1                       | 100        |
| 3   | 6                       | 1,020     | 1                       | 1,000      |
| 4   | 10                      | 7,008     | 1                       | 10,000     |
| 5   | 15                      | 42,880    | 1                       | 100,000    |
| 6   | 21                      | 243,456   | 1                       | 1,000,000  |
| 7   | 28                      | 1,311,744 | 1                       | 10,000,000 |

first LLM get the highest priority and components from the  $K$ th LLM obtain the least priority. As can be seen in Table 2.2, the system contains one macrostate and  $10^K$  microstates. Note that this is an example in which different time scales occur and is therefore expected to be harder to solve by classical iterative methods.

In the next section, we introduce the class of ML methods with the grid choices suggested by the Kronecker structure of HMMs and remark that, just like  $Q$ , none of the grids except the coarsest is explicitly generated.

**3. A class of ML methods.** The class of ML methods presented in this section are related to IAD for the analysis of MCs [29, sec. 6.3] and AMG for general systems of equations [24]. IAD is applied in the context of MCs to coefficient matrices with a two-level block structure, where blocks are loosely coupled. Different variants of the method exist; all combine the solution of an aggregated system, whose elements correspond to blocks in the two-level block partitioning, with iteration steps or solutions of systems of equations at the block level. The solution of the aggregated system distributes the steady state probability mass over the loosely coupled subsets of states, whereas at the block level the probability mass is distributed inside the subsets. AMG solves a system of equations by performing iterations on systems of equations of decreasing size. Our approach can be interpreted as a specific form of AMG for singular M-matrices, a class of matrices which will be defined in the next section. However, like in geometric multigrid, our grids have a physical meaning, since they are defined according to subsets of LLMs. Furthermore, the grids may change from one ML iteration to the next by varying the order in which LLMs are aggregated. Like in geometric multigrid, the goal is to achieve convergence that is independent of the size of the original problem. This means that the number of ML iterations to reach a predefined tolerance should be more or less independent of the number of LLMs for a given model structure. The proposed class of ML methods are related to IAD, since aggregation-disaggregation steps are used to realize the mapping between different levels. However, in contrast to IAD, varying and possibly more than two levels are defined, and the Kronecker structure is exploited to represent the aggregated matrix at each level. This implies that the class of ML methods are also expected to be efficient for large models where LLMs are tightly coupled.

**3.1. Algorithms.** One iteration of AMG over a system of linear equations is referred to as a *cycle*. Throughout the text, we use *ML iteration* and *cycle* interchangeably. The order in which the smaller aggregated systems are visited during each AMG iteration gives rise to different cycle types. Within an AMG cycle, the iterative method used to improve the solution of each aggregated system is called a *smoother*, since it is perceived to smooth the error in the solution at that level. The class of ML methods for HMMs with multiple macrostates have the capability of using (V, W, F) cycles



[33], (power, Jacobi over relaxation (JOR), successive over relaxation (SOR)) methods as smoothers, and (fixed, circular, dynamic) orders in which LLMs can be aggregated in an ML iteration. These parameters are respectively denoted by  $C$  for cycle type,  $S$  for smoother type, and  $O$  for order of aggregating LLMs. Hence,  $C \in \{V, W, F\}$ ,  $S \in \{POWER, JOR, SOR\}$ , and  $O \in \{FIXED, CIRCULAR, DYNAMIC\}$ . In a particular ML solver,  $C$ ,  $S$ , and  $O$  are fixed at the beginning and do not change.

Algorithm 1 is the driver of the ML solver. It starts executing at the finest grid involving the LLMs and the HLM, and then invokes the recursive ML function in Algorithm 2 with the order in which LLMs are to be aggregated in the list  $\mathcal{C}$ . Each pass through the body of the repeat-until loop in Algorithm 1 corresponds to one ML iteration (i.e., cycle). Observe that steps 3 through 8 in Algorithm 2 are almost identical to the statements between steps 3 and 4 in Algorithm 1.

---

ALGORITHM 1. *ML driver.*

```

main()
 $\mathcal{D} = [1, 2, \dots, K + 1]$ ;  $\tilde{Q}_{\mathcal{D}} = Q$ ;  $x_{\mathcal{D}}$  = initial approximation;  $it = 0$ ;  $stop = FALSE$ ;
(step 1)
if ( $C == W$  or  $C == F$ ) then (step 2)
     $\gamma = 2$ ;
else
     $\gamma = 1$ ;
repeat (step 3)
     $x'_{\mathcal{D}} = S(\tilde{Q}_{\mathcal{D}}, x_{\mathcal{D}}, w, \nu_1)$ ;
    remove  $\mathcal{D}_1$  from  $\mathcal{D}$  by aggregation to give  $\mathcal{C}$ ;
     $\tilde{Q}_{\mathcal{C}} = P_{x'_{\mathcal{D}}} \tilde{Q}_{\mathcal{D}} R_{\mathcal{D}}$ ;  $x_{\mathcal{C}} = x_{\mathcal{D}} R_{\mathcal{D}}$ ;
    if ( $\gamma == 1$ ) then
         $y_{\mathcal{C}} = \text{ML}(\tilde{Q}_{\mathcal{C}}, x_{\mathcal{C}}, \mathcal{C}, \gamma)$ ;
    else
         $y_{\mathcal{C}} = \text{ML}(\tilde{Q}_{\mathcal{C}}, x_{\mathcal{C}}, \mathcal{C}, \gamma)$ ;
         $y_{\mathcal{C}} = \text{ML}(\tilde{Q}_{\mathcal{C}}, y_{\mathcal{C}}, \mathcal{C}, \gamma)$ ;
     $y_{\mathcal{D}} = y_{\mathcal{C}} P_{x'_{\mathcal{D}}}$ ;
     $y'_{\mathcal{D}} = S(\tilde{Q}_{\mathcal{D}}, y_{\mathcal{D}}, w, \nu_2)$ ;
    if ( $C == F$ ) then (step 4)
         $\gamma = 2$ ;
     $x_{\mathcal{D}} = y_{\mathcal{D}}$ ;  $it = it + 1$ ; (step 5)
     $x_{\mathcal{D}} = x_{\mathcal{D}} / (x_{\mathcal{D}} e)$ ;  $r = -x_{\mathcal{D}} \tilde{Q}_{\mathcal{D}}$ ; (step 6)
    if ( $it \geq MAX\_IT$  or  $time \geq MAX\_TIME$  or  $\|r\| \leq STOP\_TOL$ ) then (step 7)
         $stop = TRUE$ ;
    else if ( $O == DYNAMIC$ ) then (step 8)
        sort LLM indices  $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_K$  into increasing order of  $\|r_k\|$ ,
        where  $r_k$  is the residual associated with LLM  $k$  and is computed from  $r$ ;
    else if ( $O == CIRCULAR$ ) then
         $\mathcal{D}_k = \mathcal{D}_{(k \bmod K) + 1}$  for  $k = 1, 2, \dots, K$ ;
until( $stop$ );
take  $x_{\mathcal{D}}$  as the steady state vector  $\pi$  of the HMM;

```

---

ALGORITHM 2. *Recursive ML function on LLMs in  $\mathcal{D}$ .*

```

function ML( $\tilde{Q}_{\mathcal{D}}, x_{\mathcal{D}}, \mathcal{D}, \gamma$ )
if ( $|\mathcal{D}| == 1$ ) then
     $y'_{\mathcal{D}} = \text{solve}(\tilde{Q}_{\mathcal{D}}, x_{\mathcal{D}})$  subject to  $y'_{\mathcal{D}} e = 1$ ; (step 1)

```

```

if ( $C == F$ ) then (step 2)
     $\gamma = 1$ ;
else
     $x'_D = S(\tilde{Q}_D, x_D, w, \nu_1)$ ; (step 3)
    remove  $\mathcal{D}_1$  from  $\mathcal{D}$  by aggregation to give  $\mathcal{C}$ ; (step 4)
     $\tilde{Q}_C = P_{x'_D} \tilde{Q}_D R_D$ ;  $x_C = x_D R_D$ ; (step 5)
    if ( $\gamma == 1$ ) then (step 6)
         $y_C = \text{ML}(\tilde{Q}_C, x_C, \mathcal{C}, \gamma)$ ;
    else
         $y_C = \text{ML}(\tilde{Q}_C, x_C, \mathcal{C}, \gamma)$ ;
         $y_C = \text{ML}(Q_C, y_C, \mathcal{C}, \gamma)$ ;
     $y_D = y_C P_{x'_D}$ ; (step 7)
     $y'_D = S(\tilde{Q}_D, y_D, w, \nu_2)$ ; (step 8)
return( $y'_D$ );

```

The variable  $\gamma$  in the two algorithms determines the number of recursive calls to the ML function. It is initialized to 2 for a W- or an F-cycle and to 1 for a V-cycle before ML starts executing for the first time. After this point, there are two places where the value of  $\gamma$  changes, and these happen only for an F-cycle. Hence, for a V-cycle  $\gamma$  remains 1, and for a W-cycle it remains 2, meaning for V- and W-cycles 1 and 2 recursive calls, respectively, are made to the ML function on the next coarser grid. On the other hand, for an F-cycle  $\gamma$  is set to 1 at the boundary case of the recursion (see step 2 in Algorithm 2). Hence, an F-cycle can be seen as a recursive call to a W-cycle followed by a recursive call to a V-cycle. After the F-cycle is over,  $\gamma$  is reset to 2 in step 4 of Algorithm 1 so as to be ready for a new ML iteration [33, pp. 174–175].

Each ML iteration starts and ends with some number of iterations using the smoother  $S$ . See respectively the two statements after step 3 and before step 4 in Algorithm 1. The same is true for each execution of the recursive ML function at intermediate grids, as can be seen in steps 3 and 8 of Algorithm 2. The first two arguments of the call to  $S$  in both algorithms represent the grid to be used in the smoothing process and the vector to be smoothed. The parameter  $\omega$  in the call to  $S$  is the relaxation parameter for JOR and SOR. Although the user can be given the flexibility to change the numbers of pre- and postsmoothings in the two algorithms, depending on the residual norms (see Algorithms 1 and 2 in [13]), we consider  $\nu_1$  pre- and  $\nu_2$  postsmoothings at each level in order to simplify the description of the algorithms in this presentation.

The order of aggregating LLMs in each ML iteration is determined by the list  $\mathcal{D}$  defined in Algorithm 1. The elements of  $\mathcal{D}$  from its head to its tail are denoted respectively by  $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_{K+1}$ . The subscripts of these elements indicate their positions in  $\mathcal{D}$ . In each ML iteration, the HLM is always the last model to be handled due to its special position in the hierarchy. Hence,  $\mathcal{D}_{K+1}$  is given the value  $(K + 1)$  and is associated with the HLM; the tail of  $\mathcal{D}$  always has this value and does not change. Initially, LLM  $k$  is associated with element  $\mathcal{D}_k$ , which has the value  $k$  for  $k = 1, 2, \dots, K$  (see step 1 of Algorithm 1). In each ML iteration, LLMs are aggregated according to these values starting from the element at the head of the list (see the second statement in the repeat-until loop of Algorithm 1). Hence, LLM  $\mathcal{D}_1$  is the first LLM to be aggregated.

In the *FIXED* order of aggregating LLMs, the initial assignment of values to the elements of  $\mathcal{D}$  does not change after the ML method starts executing; this is the default

order. In the *CIRCULAR* order, at the end of each ML iteration a circular shift of elements  $\mathcal{D}_1$  through  $\mathcal{D}_K$  in the list is performed; this ensures some kind of fairness in aggregating LLMs in the next ML iteration. On the other hand, the *DYNAMIC* order sorts the elements  $\mathcal{D}_1$  through  $\mathcal{D}_K$  according to the residual norms mapped (or restricted) to the corresponding LLM at the end of the ML iteration, and aggregates the LLMs in this sorted order in the next ML iteration (see step 8 of Algorithm 1). This ensures that LLMs which have smaller residual norms are aggregated earlier at finer grids. We expect small residual norms to be indicative of good approximations in those LLMs. Note that at each intermediate grid the recursive ML function is invoked for the next coarser grid with the list of LLMs in  $\mathcal{C}$ , which is formed by removing the LLM at the head of the incoming list  $\mathcal{D}$  (i.e.,  $\mathcal{D}_1$ ) by aggregation (see step 4 in Algorithm 2). Once the list of LLMs is exhausted, that is  $(K + 1)$  is the only value remaining in list  $\mathcal{D}$ , backtracking from recursion starts by solving a linear system as large as the HLM matrix (see step 1 in Algorithm 2). This is indicated by the call to the function *solve*, which takes the coarsest grid  $\tilde{Q}_{\mathcal{D}}$  as input and produces the solution  $y'_{\mathcal{D}}$  up to machine precision directly (i.e., by Gaussian elimination) if  $|\mathcal{S}^{(K+1)}|$  is relatively small, else iteratively using the smoother  $S$  and the current approximation  $x_{\mathcal{D}}$  as the starting vector.

The ML solver starts with  $x_{\mathcal{D}}$ , which is usually set to the uniform distribution, and  $r$  as the corresponding residual vector. The repeat-until loop increments the number of ML iterations denoted by *it* and continues until *it* reaches the maximum number of iterations in *MAX\_IT*, solution time reaches *MAX\_TIME*, or the residual  $r$  reaches the user-defined *STOP\_TOL*. We remark that the smoothers of choice require two vectors of length  $n$  and two vectors (three in SOR) as long as the maximum number of microstates per macrostate in the HMM. One of the vectors of length  $n$  in SOR is required for the computation of residuals in the implementation of *DYNAMIC* ordering of LLMs for aggregation. Furthermore, if one turns off the call(s) in Algorithm 1 to Algorithm 2, Algorithm 1 reduces to an iterative solver in which  $(\nu_1 + \nu_2)$  iterations are performed on  $Q$  with the iterative method  $S$  at each ML cycle. This is a useful feature for debugging.

**3.2. Operators and implementation.** Before we discuss the operation that computes the next coarser grid  $\tilde{Q}_{\mathcal{C}}$  from the grid  $\tilde{Q}_{\mathcal{D}}$  using the smoothed vector  $x_{\mathcal{D}}$  (see step 5 in Algorithm 2), let us define the state spaces of the grids used in the ML method for large sparse MCs in terms of a mapping [30, pp. 192–197].

**DEFINITION 3.1.** *Let  $\mathcal{S}_{\mathcal{D}}$  and  $\mathcal{S}_{\mathcal{C}}$  respectively denote the state spaces of  $\tilde{Q}_{\mathcal{D}}$  and  $\tilde{Q}_{\mathcal{C}}$ . Then the mapping  $f_{\mathcal{D}} : \mathcal{S}_{\mathcal{D}} \rightarrow \mathcal{S}_{\mathcal{C}}$  represents the transformation of states in  $\mathcal{S}_{\mathcal{D}}$  to states in  $\mathcal{S}_{\mathcal{C}}$ .*

The mapping  $f_{\mathcal{D}}$  is surjective (i.e., onto); it satisfies

$$\exists s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}, f_{\mathcal{D}}(s_{\mathcal{D}}) = s_{\mathcal{C}} \quad \text{for each } s_{\mathcal{C}} \in \mathcal{S}_{\mathcal{C}}$$

and  $|\mathcal{S}_{\mathcal{C}}| \leq |\mathcal{S}_{\mathcal{D}}|$ . When  $|\mathcal{S}_{\mathcal{C}}| = |\mathcal{S}_{\mathcal{D}}|$ , the mapping becomes bijective (i.e., one-to-one onto). From Definition 3.1 and [30, p. 179], we have the next proposition.

**PROPOSITION 3.2.** *If  $\tilde{f}_{\mathcal{D}}$  denotes the converse of  $f_{\mathcal{D}}$ , then  $\tilde{f}_{\mathcal{D}}$  is a relation from  $\mathcal{S}_{\mathcal{C}}$  to  $\mathcal{S}_{\mathcal{D}}$  and will not be a mapping unless  $|\mathcal{S}_{\mathcal{C}}| = |\mathcal{S}_{\mathcal{D}}|$  (i.e.,  $f_{\mathcal{D}}$  is bijective).*

Proposition 3.2 says that, if there is at least one state in  $\mathcal{S}_{\mathcal{C}}$  to which multiple states from  $\mathcal{S}_{\mathcal{D}}$  are mapped under  $f_{\mathcal{D}}$  (i.e.,  $|\mathcal{S}_{\mathcal{C}}| < |\mathcal{S}_{\mathcal{D}}|$ ), then the converse of  $f_{\mathcal{D}}$  cannot be a function; it is just a relation.

For HMMs, the Kronecker structure (see Definition 2.2 and Proposition 2.4) and the order of component aggregation determine  $\mathcal{S}_{\mathcal{D}}$  and  $\mathcal{S}_{\mathcal{C}}$  as in the next proposition.

PROPOSITION 3.3. *In Algorithms 1 and 2, the components in  $\mathcal{D}$  and  $\mathcal{C}$ , respectively, define  $\mathcal{S}_{\mathcal{D}}$  and  $\mathcal{S}_{\mathcal{C}}$  for HMMs, and*

$$\mathcal{S}_{\mathcal{D}} = \bigcup_{j \in \mathcal{S}^{(K+1)}} \times_{k=1}^{|\mathcal{D}|} \mathcal{S}_j^{(\mathcal{D}_k)} \quad \text{and} \quad \mathcal{S}_{\mathcal{C}} = \bigcup_{j \in \mathcal{S}^{(K+1)}} \times_{k=1}^{|\mathcal{C}|} \mathcal{S}_j^{(\mathcal{C}_k)},$$

where  $\times$  is the Cartesian product operator. Furthermore,

$$|\mathcal{S}_{\mathcal{D}}| = \sum_{j=0}^{|\mathcal{S}^{(K+1)}|-1} \prod_{k=1}^{|\mathcal{D}|} |\mathcal{S}_j^{(\mathcal{D}_k)}| \quad \text{and} \quad |\mathcal{S}_{\mathcal{C}}| = \sum_{j=0}^{|\mathcal{S}^{(K+1)}|-1} \prod_{k=1}^{|\mathcal{C}|} |\mathcal{S}_j^{(\mathcal{C}_k)}|.$$

At the finest level in Algorithm 1,  $|\mathcal{S}_{\mathcal{D}}| = n$ .

Observe from Definition 2.2 that  $\mathcal{S}_{\mathcal{D}}$  and  $\mathcal{S}_{\mathcal{C}}$  for HMMs given in Proposition 3.3 satisfy the mapping  $f_{\mathcal{D}} : \mathcal{S}_{\mathcal{D}} \rightarrow \mathcal{S}_{\mathcal{C}}$  in Definition 3.1.

Now we return to the computation of the coarser grid and the coarser approximation. For each state  $s_{\mathcal{C}} \in \mathcal{S}_{\mathcal{C}}$ , the columns of the grid  $\tilde{Q}_{\mathcal{D}}$  corresponding to the states in  $\mathcal{S}_{\mathcal{D}}$  that get mapped to the same state  $s_{\mathcal{C}}$  are summed. The aggregation on the columns of  $\tilde{Q}_{\mathcal{D}}$  is also performed on the columns of the smoothed row vector  $x'_{\mathcal{D}}$  yielding the vector  $x_{\mathcal{C}}$  in step 5 of Algorithm 2. These are achieved by using the restriction [25] (or aggregation) operator defined next.

DEFINITION 3.4. *The  $(|\mathcal{S}_{\mathcal{D}}| \times |\mathcal{S}_{\mathcal{C}}|)$  restriction operator  $R_{\mathcal{D}}$  for the mapping  $f_{\mathcal{D}} : \mathcal{S}_{\mathcal{D}} \rightarrow \mathcal{S}_{\mathcal{C}}$  has its  $(s_{\mathcal{D}}, s_{\mathcal{C}})$ th element given by*

$$r_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{C}}) = \begin{cases} 1 & \text{if } f_{\mathcal{D}}(s_{\mathcal{D}}) = s_{\mathcal{C}}, \\ 0 & \text{otherwise,} \end{cases} \quad \text{for } s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}} \text{ and } s_{\mathcal{C}} \in \mathcal{S}_{\mathcal{C}}.$$

PROPOSITION 3.5. *The restriction operator  $R_{\mathcal{D}}$  is nonnegative, has only a single nonzero with the value 1 in each row, and therefore row sums of 1. Furthermore, since there is at least one nonzero in each column of  $R_{\mathcal{D}}$ , it is also the case that  $\text{rank}(R_{\mathcal{D}}) = |\mathcal{S}_{\mathcal{C}}|$ . Thus the product  $\tilde{Q}_{\mathcal{D}}R_{\mathcal{D}}$  yields a column aggregated grid whose row sums are zero if  $\tilde{Q}_{\mathcal{D}}$  has row sums of zero.*

For each state  $s_{\mathcal{C}} \in \mathcal{S}_{\mathcal{C}}$ , the rows of  $\tilde{Q}_{\mathcal{D}}R_{\mathcal{D}}$  corresponding to the states in  $\mathcal{S}_{\mathcal{D}}$  that are mapped to the same state  $s_{\mathcal{C}}$  are multiplied with the corresponding normalized elements of the smoothed row vector  $x'_{\mathcal{D}}$  and summed. This is achieved by using the prolongation [25] (or disaggregation) operator defined next.

DEFINITION 3.6. *The  $(|\mathcal{S}_{\mathcal{C}}| \times |\mathcal{S}_{\mathcal{D}}|)$  prolongation operator  $P_{x'_{\mathcal{D}}}$  for the mapping  $f_{\mathcal{D}} : \mathcal{S}_{\mathcal{D}} \rightarrow \mathcal{S}_{\mathcal{C}}$  has its  $(s_{\mathcal{C}}, s_{\mathcal{D}})$ th element given by*

$$p_{x'_{\mathcal{D}}}(s_{\mathcal{C}}, s_{\mathcal{D}}) = \begin{cases} x'_{\mathcal{D}}(s_{\mathcal{D}}) / \sum_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}, f_{\mathcal{D}}(s_{\mathcal{D}})=s_{\mathcal{C}}} x'_{\mathcal{D}}(s_{\mathcal{D}}) & \text{if } f_{\mathcal{D}}(s_{\mathcal{D}}) = s_{\mathcal{C}}, \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{for } s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}} \text{ and } s_{\mathcal{C}} \in \mathcal{S}_{\mathcal{C}}.$$

PROPOSITION 3.7. *If  $x'_{\mathcal{D}} > 0$ , the prolongation operator  $P_{x'_{\mathcal{D}}}$  is nonnegative, has the same nonzero structure as the transpose of  $R_{\mathcal{D}}$ , a single nonzero in each column, and at least one nonzero in each row, implying  $\text{rank}(P_{x'_{\mathcal{D}}}) = |\mathcal{S}_{\mathcal{C}}|$ . Furthermore, when  $x'_{\mathcal{D}} > 0$ , each row of  $P_{x'_{\mathcal{D}}}$  is a probability vector, implying that  $P_{x'_{\mathcal{D}}}$  has row sums of 1 just like  $R_{\mathcal{D}}$ . Thus premultiplying  $\tilde{Q}_{\mathcal{D}}R_{\mathcal{D}}$  by  $P_{x'_{\mathcal{D}}}$  yields the  $(|\mathcal{S}_{\mathcal{C}}| \times |\mathcal{S}_{\mathcal{C}}|)$  square grid  $\tilde{Q}_{\mathcal{C}}$ , which has row sums of zero regardless of the norm of  $x'_{\mathcal{D}}$ .*

The prolongation operator depends not only on  $\mathcal{S}_{\mathcal{D}}$  and  $\mathcal{S}_{\mathcal{C}}$ , but also on the smoothed vector  $x'_{\mathcal{D}}$ , which is indicated by using the subscript  $x'_{\mathcal{D}}$  rather than  $\mathcal{D}$ . This implies that the elements of  $\tilde{Q}_{\mathcal{C}}$  depend on  $x'_{\mathcal{D}}$  and will be different in each iteration of the ML solver.

LEMMA 3.8. *If  $x'_{\mathcal{D}} > 0$ , then  $P_{x'_{\mathcal{D}}} R_{\mathcal{D}} = I_{\mathcal{C}}$ , where  $I_{\mathcal{C}}$  is the identity matrix of order  $|\mathcal{S}_{\mathcal{C}}|$ .*

*Proof.* The identity follows from Propositions 3.5 and 3.7 by the facts that  $P_{x'_{\mathcal{D}}} \geq 0$ ,  $R_{\mathcal{D}} \geq 0$ ,  $P_{x'_{\mathcal{D}}}$  has the same nonzero structure as  $R_{\mathcal{D}}^T$ ,  $P_{x'_{\mathcal{D}}} e = e$ , and  $e^T R_{\mathcal{D}}^T = e^T$ .  $\square$

When  $x'_{\mathcal{D}} > 0$ , we can state the next corollary [23, p. 387] using  $R_{\mathcal{D}}(P_{x'_{\mathcal{D}}} R_{\mathcal{D}})P_{x'_{\mathcal{D}}} = R_{\mathcal{D}}(I_{\mathcal{C}})P_{x'_{\mathcal{D}}} = R_{\mathcal{D}}P_{x'_{\mathcal{D}}}$  from Lemma 3.8,  $R_{\mathcal{D}} \geq 0$ ,  $R_{\mathcal{D}}e = e$  and  $P_{x'_{\mathcal{D}}} \geq 0$ ,  $P_{x'_{\mathcal{D}}}e = e$  from Propositions 3.5 and 3.7, respectively.

COROLLARY 3.9. *When  $x'_{\mathcal{D}} > 0$ , the  $(|\mathcal{S}_{\mathcal{D}}| \times |\mathcal{S}_{\mathcal{D}}|)$  matrix*

$$H_{x'_{\mathcal{D}}} = R_{\mathcal{D}}P_{x'_{\mathcal{D}}}$$

*defines a nonnegative projector (i.e.,  $H_{x'_{\mathcal{D}}} \geq 0$  and  $H_{x'_{\mathcal{D}}}^2 = H_{x'_{\mathcal{D}}}$ ) which satisfies  $H_{x'_{\mathcal{D}}}e = e$ .*

LEMMA 3.10. *If  $x'_{\mathcal{D}} > 0$ , then  $x'_{\mathcal{D}}H_{x'_{\mathcal{D}}} = x'_{\mathcal{D}}$ .*

*Proof.* The identity follows from the definitions of restriction and prolongation operations (see Definitions 3.4 and 3.6) and the fact that the restricted and then prolonged row vector is  $x'_{\mathcal{D}}$ .  $\square$

The analysis in section 4 is based on showing that the coarser grid  $\tilde{Q}_{\mathcal{C}}$  is an irreducible CTMC and  $x_{\mathcal{C}} > 0$  if the finer grid  $\tilde{Q}_{\mathcal{D}}$  is an irreducible CTMC and  $x'_{\mathcal{D}} > 0$ . This has been done for HMMs with one macrostate in [9, p. 348]. In section 4, we show the results for the mapping  $f : \mathcal{S}_{\mathcal{D}} \rightarrow \mathcal{S}_{\mathcal{C}}$  in Definition 3.1.

Step 7 in Algorithm 2 corresponds to the opposite of what is done on  $x'_{\mathcal{D}}$  in step 5; that is, it performs disaggregation using the newly computed vector  $y_{\mathcal{C}}$  and the prolongation operator  $P_{x'_{\mathcal{D}}}$  (which is based on the smoothed vector  $x'_{\mathcal{D}}$ ) to obtain the vector  $y_{\mathcal{D}}$ . The next result follows from Proposition 3.7

PROPOSITION 3.11. *If  $y_{\mathcal{C}} > 0$  and  $x'_{\mathcal{D}} > 0$ , then  $y_{\mathcal{D}} = y_{\mathcal{C}}P_{x'_{\mathcal{D}}} > 0$ , since  $e^T P_{x'_{\mathcal{D}}} > 0$ .*

Similar aggregation and disaggregation operations are performed in Algorithm 1 at the finest grid  $Q$ .

The Kronecker representation of  $\tilde{Q}_{\mathcal{C}}$  for an HMM with one macrostate is given in [9, p. 347]. Here we extend it to multiple macrostates and show that  $\tilde{Q}_{\mathcal{C}}$  can be expressed as a sum of Kronecker products as in Definition 2.2 using  $\sum_{i,j \in \mathcal{S}^{(K+1)}} |\mathcal{T}_{i,j}|$  vectors each of length at most  $\max_{j \in \mathcal{S}^{(K+1)}} (\prod_{k=2}^{|\mathcal{C}|} |\mathcal{S}_j^{(C_k)}|)$  and the matrices corresponding to the components in  $\mathcal{C}$  excluding  $(K+1)$ , which denotes the HLM (see Proposition 3.3). More specifically, we have the next definition.

DEFINITION 3.12. *If  $h = \mathcal{D}_1$  is the index of the aggregated component, then the  $s_{\mathcal{C}}$ th element of the vector corresponding to the  $t_e$ th term in block  $(i,j)$  of the aggregated CTMC  $\tilde{Q}_{\mathcal{C}}$  is defined as*

$$a_{(\mathcal{C}, t_e), (i,j)}(s_{\mathcal{C}}) = \frac{\left( \sum_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}, f_{\mathcal{D}}(s_{\mathcal{D}}) = s_{\mathcal{C}}} x'_{\mathcal{D}}(s_{\mathcal{D}}) a_{(\mathcal{D}, t_e), (i,j)}(s_{\mathcal{D}}) (e_{s_{\mathcal{D}}(h)}^T Q_{t_e}^{(h)}(\mathcal{S}_i^{(h)}, \mathcal{S}_j^{(h)})e) \right)}{x_{\mathcal{C}}(s_{\mathcal{C}})}$$

*for  $s_{\mathcal{C}} \in \mathcal{S}_{\mathcal{C}}$ ,  $t_e \in \mathcal{T}_{i,j}$ , and  $i, j \in \mathcal{S}^{(K+1)}$ ,*

where  $a_{(\mathcal{D},t_e),(i,j)} = e$  if  $\mathcal{D}$  corresponds to the finest level,  $s_{\mathcal{D}}(h) \in \mathcal{S}^{(h)}$ , and  $e_{s_{\mathcal{D}}(h)}$  is the  $s_{\mathcal{D}}(h)$ th column of the identity matrix of order  $|\mathcal{S}_i^{(h)}|$ .

With this definition, blocks of the matrix  $\tilde{Q}_{\mathcal{C}}$  become

$$\begin{aligned} \tilde{Q}_{\mathcal{C}}(j, j) &= \bigoplus_{k=1}^{|\mathcal{C}|-1} Q_{t_0}^{(C_k)}(\mathcal{S}_j^{(C_k)}, \mathcal{S}_j^{(C_k)}) + \sum_{t_e \in \mathcal{T}_{j,j}} \bigotimes_{k=1}^{|\mathcal{C}|-1} \text{diag}(a_{(\mathcal{C},t_e),(j,j)}) Q_{t_e}^{(C_k)}(\mathcal{S}_j^{(C_k)}, \mathcal{S}_j^{(C_k)}) \\ &\quad - \bigoplus_{k=1}^{|\mathcal{C}|-1} \text{diag}(Q_{t_0}^{(C_k)}(\mathcal{S}_j^{(C_k)}, \mathcal{S}_j^{(C_k)})e) \\ &\quad - \sum_{i \in \mathcal{S}^{(K+1)}} \sum_{t_e \in \mathcal{T}_{j,i}} \bigotimes_{k=1}^{|\mathcal{C}|-1} \text{diag}(a_{(\mathcal{C},t_e),(j,i)}) \text{diag}(Q_{t_e}^{(C_k)}(\mathcal{S}_j^{(C_k)}, \mathcal{S}_i^{(C_k)})e) \\ &\qquad\qquad\qquad \text{for } j \in \mathcal{S}^{(K+1)}, \\ \tilde{Q}_{\mathcal{C}}(i, j) &= \sum_{t_e \in \mathcal{T}_{i,j}} \bigotimes_{k=1}^{|\mathcal{C}|-1} \text{diag}(a_{(\mathcal{C},t_e),(i,j)}) Q_{t_e}^{(C_k)}(\mathcal{S}_i^{(C_k)}, \mathcal{S}_j^{(C_k)}) \quad \text{for } i, j \in \mathcal{S}^{(K+1)}, i \neq j. \end{aligned}$$

Observe from Proposition 2.3 that the last two terms of  $\tilde{Q}_{\mathcal{C}}(j, j)$  return a diagonal matrix which sums the rows of  $\tilde{Q}_{\mathcal{C}}(j, j)$  to zero. Furthermore, the vectors  $a_{(\mathcal{D},t_e),(i,j)}$  for  $t_e \in \mathcal{T}_{i,j}$  and  $i, j \in \mathcal{S}^{(K+1)}$  at the finest level consist of all 1's, and therefore need not be stored. When the recursion ends at the HLM,  $\tilde{Q}_{\mathcal{C}}$  is a  $(|\mathcal{S}^{(K+1)}| \times |\mathcal{S}^{(K+1)}|)$  CTMC, and therefore is generated and stored explicitly in sparse format so that it can be solved either directly or iteratively, as we discussed. We remark that  $a_{(\mathcal{C},t_e),(i,j)} = e$  for those  $t_e$  which have all  $Q_{t_e}^{(C_k)}(\mathcal{S}_i^{(C_k)}, \mathcal{S}_j^{(C_k)})$  as diagonal matrices of size  $(|\mathcal{S}_i^{(C_k)}| \times |\mathcal{S}_j^{(C_k)}|)$  with 1's along their diagonal for  $k = 1, 2, \dots, |\mathcal{C}| - 1$  and  $i, j \in \mathcal{S}^{(K+1)}$ . Since component matrices forming  $\tilde{Q}_{\mathcal{C}}(i, j)$  for  $i, j \in \mathcal{S}^{(K+1)}, i \neq j$ , can very well be rectangular, we refrain from using  $I$ , and remark that such vectors need not be stored either.

The next section presents results on the convergence of the proposed class of ML methods for large sparse MCs.

**4. Convergence of ML methods.** Convergence analysis of AMG with a post-smoother of the Richardson relaxation type (see [26, p. 412]) and a two-level grid for symmetric positive definite linear systems arising from finite element approximations to a particular differential operator appears in [18]. Therein, it is shown that the convergence rate of the method is independent of the problem size when the relaxation parameter of the smoother is chosen appropriately [18, p. 480]. On the other hand, [27] casts AMG as a special case of multi-iterative methods for positive definite linear systems in which two or more iterative techniques are successively used in each iteration to improve the error in different subspaces. When the method is AMG, one of these multi-iterative methods has an iteration matrix associated with the coarse grid correction. A convergence analysis for a two-level grid with a Richardson iteration as the presmoother and a prolongation operator with (block) antidiagonal structure is provided. Using information about the eigenvalues of the coefficient matrix together with the particular smoother, it is shown that the AMG method possesses a convergence rate independent of the problem size for banded (block) Toeplitz matrices. Although the *POWER* smoother used by the proposed class of ML methods is also a Richardson relaxation, as will be shown in this section, the methods are geared towards CTMCs, which have different characteristics. Recently, in [22] the results in

[21] are improved, and an asymptotic convergence result is provided for a two-level IAD method which uses postsmoothings of the POWER type. However, fast convergence cannot be guaranteed in a general setting even when there are only two levels [22, p. 340]. Hence, the results in the next subsections should be received as a step towards improving the formulation and understanding the convergence behavior of the proposed class of ML methods.

Let  $\mathcal{D}$  represent the current level and  $\mathcal{C}$  represent the next coarser level in the ML iteration, as in Algorithms 1 and 2. Let  $\mathcal{S}_{\mathcal{D}}$  and  $\mathcal{S}_{\mathcal{C}}$  denote respectively the state spaces of  $\tilde{Q}_{\mathcal{D}}$  and  $\tilde{Q}_{\mathcal{C}}$ , and assume that the mapping of states from  $\mathcal{S}_{\mathcal{D}}$  to the states in  $\mathcal{S}_{\mathcal{C}}$  is onto and satisfies  $|\mathcal{S}_{\mathcal{C}}| \leq |\mathcal{S}_{\mathcal{D}}|$  as in Definition 3.1. The results that are presented in this section for Algorithms 1 and 2 are general in that the Kronecker representation of the grids particular to HMMs is not utilized.

**4.1. Irreducibility of the coarser grids.** Recall that  $R_{\mathcal{D}} \geq 0$ ,  $R_{\mathcal{D}}e = e$ ,  $e^T R_{\mathcal{D}} > 0$  from Proposition 3.5, and if  $x'_{\mathcal{D}} > 0$ , then  $P_{x'_{\mathcal{D}}} \geq 0$ ,  $P_{x'_{\mathcal{D}}}e = e$ ,  $e^T P_{x'_{\mathcal{D}}} > 0$  from Proposition 3.7. Now, consider the definition of irreducibility given in [23, p. 209] and [29, p. 13]. Then the following lemma, which will be used to discuss the convergence of the ML method, can be proved.

LEMMA 4.1. *The coarser grid  $\tilde{Q}_{\mathcal{C}} = P_{x'_{\mathcal{D}}} \tilde{Q}_{\mathcal{D}} R_{\mathcal{D}}$  is an irreducible CTMC and  $x_{\mathcal{C}} = x'_{\mathcal{D}} R_{\mathcal{D}} > 0$  if the finer grid  $\tilde{Q}_{\mathcal{D}}$  is an irreducible CTMC and  $x'_{\mathcal{D}} > 0$ .*

*Proof.* First, we show that  $\tilde{Q}_{\mathcal{C}} = P_{x'_{\mathcal{D}}} \tilde{Q}_{\mathcal{D}} R_{\mathcal{D}}$  is an irreducible CTMC. Without losing generality, consider the pair of different states  $s_{\mathcal{D}}, s'_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}$ . Through  $f : \mathcal{S}_{\mathcal{D}} \rightarrow \mathcal{S}_{\mathcal{C}}$  in Definition 3.1, this pair of states are mapped respectively to the states  $s_{\mathcal{C}}, s'_{\mathcal{C}} \in \mathcal{S}_{\mathcal{C}}$  (i.e.,  $f(s_{\mathcal{D}}) = s_{\mathcal{C}}$  and  $f(s'_{\mathcal{D}}) = s'_{\mathcal{C}}$ ). Since  $\tilde{Q}_{\mathcal{D}}$  is irreducible, there exists a path of transitions from  $s_{\mathcal{D}}$  to  $s'_{\mathcal{D}}$  in  $\mathcal{S}_{\mathcal{D}}$  in the form  $s_{\mathcal{D}} = s_1, s_2, \dots, s_m = s'_{\mathcal{D}}$ , where  $m \leq |\mathcal{S}_{\mathcal{D}}|$ ,  $s_k \in \mathcal{S}_{\mathcal{D}}$ , and  $\tilde{q}_{\mathcal{D}}(s_k, s_{k+1}) > 0$  for  $k \in \{1, 2, \dots, m-1\}$ . Mapping this path onto  $\mathcal{S}_{\mathcal{C}}$  yields the path  $s_{\mathcal{C}} = t_1, t_2, \dots, t_m = s'_{\mathcal{C}}$ , where  $f(s_k) = t_k \in \mathcal{S}_{\mathcal{C}}$ . Now, let  $e_{t_k}$  denote the  $t_k$ th column of  $I_{\mathcal{C}}$ . Then, in the mapped path, we either have  $t_k = t_{k+1}$  or  $\tilde{q}_{\mathcal{C}}(t_k, t_{k+1}) > 0$ , where the latter follows from

$$\begin{aligned} \tilde{q}_{\mathcal{C}}(t_k, t_{k+1}) &= e_{t_k}^T \tilde{Q}_{\mathcal{C}} e_{t_{k+1}} \\ &= (e_{t_k}^T P_{x'_{\mathcal{D}}}) \tilde{Q}_{\mathcal{D}} (R_{\mathcal{D}} e_{t_{k+1}}) \geq p_{x'_{\mathcal{D}}}(t_k, s_k) \tilde{q}_{\mathcal{D}}(s_k, s_{k+1}) r_{\mathcal{D}}(s_{k+1}, t_{k+1}), \end{aligned}$$

since  $x_{\mathcal{D}}(s_k) > 0$  (implying  $p_{x'_{\mathcal{D}}}(t_k, s_k) > 0$  from Definition 3.6),  $\tilde{q}_{\mathcal{D}}(s_k, s_{k+1}) > 0$ , and  $f(s_{k+1}) = t_{k+1}$  (implying  $r_{\mathcal{D}}(s_{k+1}, t_{k+1}) = 1$  from Definition 3.4). Thus we conclude that  $s'_{\mathcal{C}}$  is reachable from  $s_{\mathcal{C}}$ .

We have effectively shown that each state in  $\tilde{Q}_{\mathcal{C}}$  is reachable from every other state. The question that arises at this point is whether a row of  $\tilde{Q}_{\mathcal{C}}$  can become zero after the restriction. The answer is no, as long as  $\mathcal{S}_{\mathcal{C}}$  has multiple states (i.e.,  $|\mathcal{S}_{\mathcal{C}}| > 1$ ), since all states in  $\mathcal{S}_{\mathcal{D}}$  that are mapped to a particular state in  $\mathcal{S}_{\mathcal{C}}$  cannot have all their transitions among themselves. This would imply that  $\tilde{Q}_{\mathcal{D}}$  is reducible, which is a contradiction. Furthermore, since the row sums of  $\tilde{Q}_{\mathcal{C}}$  are zero (i.e.,  $\tilde{Q}_{\mathcal{C}}e = (P_{x'_{\mathcal{D}}} \tilde{Q}_{\mathcal{D}} R_{\mathcal{D}})e = P_{x'_{\mathcal{D}}} \tilde{Q}_{\mathcal{D}} (R_{\mathcal{D}}e) = P_{x'_{\mathcal{D}}} \tilde{Q}_{\mathcal{D}} e = 0$  because  $\tilde{Q}_{\mathcal{D}}$  is a CTMC and  $\tilde{Q}_{\mathcal{D}}e = 0$ ), its diagonal must be equal to its negated off-diagonal row sums. Hence,  $\tilde{Q}_{\mathcal{C}}$  is an irreducible CTMC.

Now we show that  $x_{\mathcal{C}} > 0$ . Since  $x_{\mathcal{C}} = x'_{\mathcal{D}} R_{\mathcal{D}}$ ,  $x'_{\mathcal{D}} = e^T \text{diag}(x'_{\mathcal{D}})$ , where  $\text{diag}(x'_{\mathcal{D}})$  is the diagonal matrix with  $x'_{\mathcal{D}}$  along its diagonal,  $\text{diag}(x'_{\mathcal{D}}) R_{\mathcal{D}}$  has the same nonzero structure as  $R_{\mathcal{D}}$ , and  $e^T R_{\mathcal{D}} > 0$ , we have  $x_{\mathcal{C}} = x'_{\mathcal{D}} R_{\mathcal{D}} = (e^T \text{diag}(x'_{\mathcal{D}})) R_{\mathcal{D}} = e^T (\text{diag}(x'_{\mathcal{D}}) R_{\mathcal{D}}) > 0$  when  $x'_{\mathcal{D}} > 0$ .  $\square$

COROLLARY 4.2. *If  $\tilde{Q}_D$  is an irreducible CTMC,  $x'_D > 0$ , and  $x'_D \tilde{Q}_D = 0$ , then  $x_C \tilde{Q}_C = 0$ , where  $\tilde{Q}_C = P_{x'_D} \tilde{Q}_D R_D$  and  $x_C = x'_D R_D$ .*

*Proof.* We have  $x_C \tilde{Q}_C = (x'_D R_D)(P_{x'_D} \tilde{Q}_D R_D) = (x'_D R_D P_{x'_D}) \tilde{Q}_D R_D = (x'_D H_{x'_D}) \tilde{Q}_D R_D = (x'_D) \tilde{Q}_D R_D = (x'_D \tilde{Q}_D) R_D = 0$ , since  $x'_D H_{x'_D} = x'_D$  from Lemma 3.10 and  $x'_D \tilde{Q}_D = 0$  by assumption.  $\square$

PROPOSITION 4.3. *If  $\pi_D = \pi > 0$  denotes the steady state vector of the irreducible grid  $Q_D = Q$  at the finest level  $D$ , then the irreducible grid obtained by exact aggregation at the next coarser level  $C$  is  $Q_C = P_{\pi_D} Q_D R_D$  and has the steady state vector  $\pi_C = \pi_D R_D > 0$ . The result extends to all adjacent pairs of levels  $D$  and  $C$  as long as level  $D$  has the exact irreducible grid  $Q_D$  and its steady state vector  $\pi_D$  is used to compute the irreducible grid  $Q_C$  at the next coarser level  $C$ .*

The proposition follows from  $\pi_C Q_C = (\pi_D R_D)(P_{\pi_D} Q_D R_D) = (\pi_D R_D P_{\pi_D}) Q_D R_D = (\pi_D H_{\pi_D}) Q_D R_D = (\pi_D) Q_D R_D = (\pi_D Q_D) R_D = 0$  since  $\pi_D H_{\pi_D} = \pi_D$  from Lemma 3.10 and  $\pi_D Q_D = 0$  by assumption.

The next subsection specifies sufficient conditions for a converging smoother to provide improved solutions at each level.

**4.2. Convergence of the smoothers.** By definition at the finest level in Algorithm 1 and by construction at the coarser levels in Algorithm 2, the matrix  $\tilde{Q}_D$  is an irreducible CTMC when  $x'_D > 0$  (see Lemma 4.1). Now, consider the nontransposed homogeneous singular linear system in the next definition (cf. (1.1)).

DEFINITION 4.4. *The problem at level  $D$  in the ML method is to solve*

$$\tilde{\pi}_D \tilde{Q}_D = 0 \quad \text{subject to} \quad \tilde{\pi}_D e = 1,$$

where  $\tilde{\pi}_D > 0$  is the steady state vector of the irreducible CTMC  $\tilde{Q}_D$ .

PROPOSITION 4.5. *At the finest level  $D$ , the steady state vector of the irreducible CTMC  $\tilde{Q}_D$  satisfies  $\tilde{\pi}_D = \pi$  since  $\tilde{Q}_D = Q$ .*

Now, consider the splitting of  $\tilde{Q}_D$  in the next definition.

DEFINITION 4.6. *Let  $\tilde{Q}_D$  be split as*

$$\tilde{Q}_D = D_D - U_D - L_D = M_D - N_D,$$

where  $D_D$ ,  $U_D$ , and  $L_D$  are respectively the diagonal, negated strictly upper-triangular, and negated strictly lower-triangular parts of  $\tilde{Q}_D$ , and  $M_D$  is nonsingular (i.e.,  $M_D^{-1}$  exists).

PROPOSITION 4.7. *If  $\tilde{Q}_D$  is an irreducible CTMC, each of the terms  $D_D$ ,  $U_D$ , and  $L_D$  in the splitting of  $\tilde{Q}_D$  is nonpositive; furthermore,  $\tilde{q}_D(s_D, s_D) \neq 0$  for all  $s_D \in \mathcal{S}_D$ , implying that  $D_D^{-1}$  and  $(D_D - U_D)^{-1}$  exist.*

The next definition involving the iteration matrices of the POWER, JOR, and SOR smoothers follows from [29, Chap. 3].

PROPOSITION 4.8. *If  $\tilde{Q}_D$  is an irreducible CTMC, then the POWER, JOR, and SOR smoothers are based on different splittings of  $\tilde{Q}_D$ , where each yields an iteration matrix of the form*

$$T_D = N_D M_D^{-1}$$

and the sequence of approximations

$$x_D^{(m+1)} = x_D^{(m)} T_D \quad \text{for} \quad m = 0, 1, \dots$$



The particular splittings corresponding to the three smoothers are

$$\begin{aligned} M_{\mathcal{D}}^{\text{POWER}} &= -\alpha_{\mathcal{D}} I_{\mathcal{D}}, & N_{\mathcal{D}}^{\text{POWER}} &= -\alpha_{\mathcal{D}}(I_{\mathcal{D}} + \tilde{Q}_{\mathcal{D}}/\alpha_{\mathcal{D}}), \\ M_{\mathcal{D}}^{\text{JOR}} &= D_{\mathcal{D}}/\omega, & N_{\mathcal{D}}^{\text{JOR}} &= (1 - \omega)D_{\mathcal{D}}/\omega + L_{\mathcal{D}} + U_{\mathcal{D}}, \\ M_{\mathcal{D}}^{\text{SOR}} &= D_{\mathcal{D}}/\omega - U_{\mathcal{D}}, & N_{\mathcal{D}}^{\text{SOR}} &= (1 - \omega)D_{\mathcal{D}}/\omega + L_{\mathcal{D}}, \end{aligned}$$

where  $\alpha_{\mathcal{D}} \in [\max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|, \infty)$  is the uniformization parameter of POWER and  $\omega \in (0, 2)$  is the relaxation parameter of JOR and SOR. The JOR and SOR splittings reduce to Jacobi and Gauss-Seidel (GS) splittings for  $\omega = 1$ . Hence, the iteration matrices corresponding to the three splittings are

$$\begin{aligned} T_{\mathcal{D}}^{\text{POWER}} &= I_{\mathcal{D}} + \tilde{Q}_{\mathcal{D}}/\alpha_{\mathcal{D}}, \\ T_{\mathcal{D}}^{\text{JOR}} &= (1 - \omega)I_{\mathcal{D}} + \omega(L_{\mathcal{D}} + U_{\mathcal{D}})D_{\mathcal{D}}^{-1}, \\ T_{\mathcal{D}}^{\text{SOR}} &= ((1 - \omega)D_{\mathcal{D}}/\omega + L_{\mathcal{D}})(D_{\mathcal{D}}/\omega - U_{\mathcal{D}})^{-1}. \end{aligned}$$

Since  $\tilde{Q}_{\mathcal{D}}$  is the generator matrix of an irreducible CTMC, the relation  $\tilde{\pi}_{\mathcal{D}} T_{\mathcal{D}}^S = \tilde{\pi}_{\mathcal{D}}$  holds for  $S \in \{\text{POWER}, \text{SOR}, \text{JOR}\}$  [29].

Before we state another lemma, we recall the definitions of primitivity and M-matrix from [29, pp. 352, 170] and remark that detailed information concerning M-matrices may be found in [4].

**DEFINITION 4.9.** Let  $\sigma(A)$  denote the set of eigenvalues (or spectrum) of the square matrix  $A$ , and let  $\rho(A)$  be the spectral radius of  $A$  (i.e.,  $\rho(A) = \{\max |\lambda| \mid \lambda \in \sigma(A)\}$ ). A nonnegative irreducible matrix  $B$  is said to be primitive if it has a single eigenvalue with magnitude  $\rho(B)$ .

**DEFINITION 4.10.** Any square matrix  $A$  of the form  $A = \beta I - B$  with  $\beta > 0$  and  $B \geq 0$  for which  $\beta \geq \rho(B)$  is called an M-matrix.

Hence, the negated CTMC  $-\tilde{Q}_{\mathcal{D}}$  is a singular M-matrix. The next proposition follows from [23, p. 640] and [29, p. 118].

**PROPOSITION 4.11.** For the irreducible CTMC  $\tilde{Q}_{\mathcal{D}}$ , the matrix  $e\tilde{\pi}_{\mathcal{D}}$  has the steady vector of  $\tilde{Q}_{\mathcal{D}}$  in each of its rows, and therefore is a positive, stochastic matrix of rank 1.

**COROLLARY 4.12.** When  $\tilde{Q}_{\mathcal{D}}$  has a single state (i.e.,  $|\mathcal{S}_{\mathcal{D}}| = 1$ ),  $\tilde{Q}_{\mathcal{D}} = 0$  and  $\tilde{\pi}_{\mathcal{D}} = 1$ .

For HMMs, Corollary 4.12 applies at the coarsest level when the HLM has one macrostate.

Now we are in a position to state and prove a lemma, which is essential in characterizing the convergence of the three smoothers.

**LEMMA 4.13.** If the smoother  $S \in \{\text{POWER}, \text{JOR}, \text{SOR}\}$  satisfies  $\alpha_{\mathcal{D}} \in (\max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|, \infty)$  and  $\omega \in (0, 1)$ , then the iteration matrix  $T_{\mathcal{D}}$  associated with the irreducible CTMC  $\tilde{Q}_{\mathcal{D}}$  is nonnegative, irreducible, primitive, and has a spectral radius and an eigenvalue of 1; furthermore,  $T_{\mathcal{D}} = W_{\mathcal{D}} B_{\mathcal{D}} W_{\mathcal{D}}^{-1}$ , where  $B_{\mathcal{D}}$  is a stochastic matrix and  $W_{\mathcal{D}}$  is a nonnegative diagonal matrix having the right eigenvector of  $T_{\mathcal{D}}$  corresponding to one along its diagonal, implying  $\lim_{m \rightarrow \infty} T_{\mathcal{D}}^m = (W_{\mathcal{D}} e) \tilde{\pi}_{\mathcal{D}} / (\tilde{\pi}_{\mathcal{D}} W_{\mathcal{D}} e) > 0$  and is of rank 1. When POWER is the smoother,  $W_{\mathcal{D}} = I_{\mathcal{D}}$  and  $T_{\mathcal{D}}$  is a stochastic matrix, implying  $\lim_{m \rightarrow \infty} T_{\mathcal{D}}^m = e\tilde{\pi}_{\mathcal{D}} > 0$ .

*Proof.* The proof follows from Theorem 17 of [29].  $\square$

Using Lemma 4.13, the next proposition expresses the pre- and postsmoothings at level  $\mathcal{D}$  concisely.

**PROPOSITION 4.14.** Given the generator matrix  $\tilde{Q}_{\mathcal{D}}$  of an irreducible CTMC and a vector  $x_{\mathcal{D}} > 0$ , after  $\nu > 0$  iterations of pre- or postsmoothings at level  $\mathcal{D}$  with the

smoother  $S$  satisfying Lemma 4.13, the smoothed vector becomes

$$x'_D = x_D T_D^{\nu} > 0.$$

The next proposition follows from Theorem 4.4 in [28, pp. 45–46] and is introduced to aid the characterization of the nonasymptotic convergence behavior of smoothings.

PROPOSITION 4.15. *Let  $A_D \in \mathbb{R}^{|\mathcal{S}_D| \times |\mathcal{S}_D|}$  be nonsingular (i.e.,  $A_D^{-1}$  exists). Then the function defined as*

$$\|w\|_{A_D} = \|wA_D\|_1 \quad \text{for } w \in \mathbb{R}^{1 \times |\mathcal{S}_D|}$$

is a vector norm.<sup>2</sup>

The next theorem characterizes the *nonasymptotic convergence* behavior of the smoothings through a lemma for positive stochastic matrices based on the discussion in [2, pp. 270–271] and proved in [13, appendix], and two results on nonnegative irreducible matrices similar to positive matrices [5, pp. 371 and 375]. We remark that a similar theorem may be stated for the initial approximation  $y_D$ .

THEOREM 4.16. *Given the initial approximation  $x_D^{(0)} = x_D > 0$  for the irreducible CTMC  $\tilde{Q}_D$  and the smoother  $S \in \{\text{POWER}, \text{JOR}, \text{SOR}\}$  with iteration matrix  $T_D$  such that  $x_D^T \notin \text{Range}(I_D - T_D^T)$  if  $T_D^{\nu_1}$  is nonnegative, irreducible, and satisfies any of the three conditions*

- (i)  $T_D^{\nu_1}$  is positive,
- (ii)  $T_D^{\nu_1}$  has a positive row  $i_D$  or a positive column  $j_D$ ,
- (iii)  $T_D^{\nu_1}$  has a zero element in position  $(i_D, j_D)$ ,
  - (a) all other elements in row  $i_D$  are positive and  $e_{i_D}^T T_D^{\nu_1} e_{i_D} > e_{j_D}^T T_D^{\nu_1} e_{j_D}$ ,  
or
  - (b) all other elements in column  $j_D$  are positive and  $e_{i_D}^T T_D^{\nu_1} e_{i_D} < e_{j_D}^T T_D^{\nu_1} e_{j_D}$ ,

then

$$\|c_D x'_D - \tilde{\pi}_D\|_{A_D} \leq \left(1 - \min_{i_D, j_D \in \mathcal{S}_D} g_D(i_D, j_D)\right) \|c_D x_D - \tilde{\pi}_D\|_{A_D},$$

where  $x'_D = x_D T_D^{\nu_1}$ ,  $G_D$  is a positive stochastic matrix defined as  $G_D = A_D^{-1} T_D^{\nu_1} A_D$  for some  $A_D \geq 0$  such that  $0 < \min_{i_D, j_D \in \mathcal{S}_D} g_D(i_D, j_D) \leq 1/|\mathcal{S}_D|$ ,  $\tilde{\pi}_D$  is the steady state vector of  $\tilde{Q}_D$ , and  $c_D = (\tilde{\pi}_D A_D e)/(x_D A_D e)$ .

*Proof.* From Corollary 3 and Theorem 4 in [5], if  $T_D^{\nu_1}$  is nonnegative, is irreducible, and satisfies either of the conditions (ii) or (iii), then it is similar to a positive matrix; that is,  $X_D^{-1} T_D^{\nu_1} X_D = H_D > 0$  for some  $(|\mathcal{S}_D| \times |\mathcal{S}_D|)$  nonnegative matrix  $X_D$ . Condition (i) is a special case for which  $X_D = I_D$ . Since these imply  $\sigma(H_D) = \sigma(T_D^{\nu_1})$  and we have  $\rho(T_D^{\nu_1}) = 1$  from Lemma 4.13,  $H_D > 0$  must be similar to a positive stochastic matrix  $G_D$  as in  $Y_D^{-1} H_D Y_D = G_D > 0$ , where  $Y_D$  is a nonnegative diagonal matrix having the positive right eigenvector of  $H_D$  along its diagonal. Now, let  $A_D = X_D Y_D$  to obtain  $T_D^{\nu_1} = A_D G_D A_D^{-1}$ , where  $A_D \geq 0$ ,  $G_D > 0$ , and  $G_D e = e$ .

For a sequence of converging approximations, one needs to ensure for the initial approximation that  $x_D^T \notin \text{Range}(I_D - T_D^T)$  [3, pp. 26–28]; otherwise, there will be no improvement. Furthermore, since  $\tilde{\pi}_D$  is the unique positive fixed point of  $T_D^{\nu_1}$  such that  $\tilde{\pi}_D e = 1$ , the unique positive fixed point of  $G_D$  with unit 1-norm must be  $\psi_D = (\tilde{\pi}_D A_D)/(\tilde{\pi}_D A_D e)$ . Now, rewrite  $x'_D = x_D T_D^{\nu_1}$  using  $T_D^{\nu_1} = A_D G_D A_D^{-1}$

<sup>2</sup>This norm should not be confused with the elliptical norm [23, p. 288] defined as  $\|w\|_{A_D} = \|wA_D\|_2$ .

to obtain  $x'_D A_D = x_D A_D (G_D)$ . Since  $x_D > 0$ ,  $A_D \geq 0$ , and  $A_D$  has full rank, we have  $x'_D > 0$ . Furthermore, note that  $x'_D A_D e = x_D A_D (G_D e) = x_D A_D e$ . Letting  $\bar{x}'_D = (x'_D A_D)/(x_D A_D e)$  and  $\bar{x}_D = (x_D A_D)/(x_D A_D e)$ , we have from Lemma A.1 in [13, appendix]

$$\|\bar{x}'_D - \psi_D\|_1 \leq \left(1 - \min_{i_D, j_D \in S_D} g_D(i_D, j_D)\right) \|\bar{x}_D - \psi_D\|_1.$$

The result follows by taking each of  $(\bar{x}'_D - \psi_D)$  and  $(\bar{x}_D - \psi_D)$  into  $A_D$  parentheses, multiplying both sides of the inequality by  $\bar{\pi}_D A_D e$ , letting  $c_D = (\bar{\pi}_D A_D e)/(x_D A_D e)$ , and using Proposition 4.15.  $\square$

Theorem 4.16 indicates that the *normalized* solution vector,  $c_D x_D$ , improves with  $\nu_1$  presmoothings if  $T_D^{\nu_1}$  is positive or has a(n almost) positive row or column. Now, observe that the ordering of grids suggested by  $O \in \{FIXED, CIRCULAR, DYNAMIC\}$  has no effect on the assumptions of Theorem 4.16. Note also from Lemma 4.13 that as  $\nu_1$  increases,  $T_D^{\nu_1}$  converges to a positive rank 1 matrix. Hence, there is a value of  $\nu_1 > 0$  for which the assumptions of Theorem 4.16 hold. We remark that  $\tilde{Q}_D$  is almost always sparse, and the iteration matrices associated with the *POWER* and *JOR* smoothers have the same off-diagonal nonzero structure as that of  $\tilde{Q}_D$ . Hence, compared to *POWER* and *JOR*, the *SOR* smoother has a higher chance of satisfying the conditions of Theorem 4.16 for a smaller value of  $\nu_1$ , since its iteration matrix is likely to have a larger number of nonzeros, as suggested in the proof of Lemma 4.17 in [13]. Similar arguments are valid for postsmoothings. These results can be perceived as an extension of the local convergence result available in [22, sec. 2] to include the *JOR* and *SOR* smoothers and another sufficient condition (i.e., Theorem 4.16(iii)). In summary, the smoothings can always be enforced to yield improved positive approximations at each level.

**4.3. Convergence of the ML solver.** Using the results in the previous subsections, we show that under certain conditions the devised class of ML methods provide converging iterations for different choices of the cycle parameter  $C \in \{V, W, F\}$ .

First, we define the ML iteration matrix at level  $\mathcal{D}$  in Algorithms 1 and 2 using Propositions 3.5, 3.7, 4.11, and 4.14. Note that when there are only two levels, the W- and F-cycles are not defined, and the V-cycle yields a two-level IAD solver. In order not to complicate the notation further, we refrain from introducing an index for the cycle number to the matrices and vectors at this point.

**DEFINITION 4.17.** Let  $T_D^{ML}$  denote the ML iteration matrix that operates at level  $\mathcal{D}$  on  $x_D > 0$  to give  $y_D > 0$  at a particular cycle using the smoother  $S \in \{POWER, JOR, SOR\}$  with iteration matrix  $T_D$  for the irreducible CTMC  $\tilde{Q}_D$ , where  $\alpha_D \in (\max_{s_D \in S_D} |\tilde{q}_D(s_D, s_D)|, \infty)$  and  $\omega \in (0, 1)$ , the restriction operator  $R_D$ , and the prolongation operator  $P_{x'_D}$ . Similarly let  $T_C^{ML}$  and  $T_B^{ML}$  denote the ML iteration matrices that operate at the next two coarser levels  $\mathcal{C}$  and  $\mathcal{B}$ , respectively. Then

$$y'_D = x_D T_D^{ML},$$

where

$$T_D^{ML} = \begin{cases} T_D^{\nu_1} R_D T_C^{ML} P_{x'_D} T_D^{\nu_2} & \text{if } C = V, \\ T_D^{\nu_1} R_D (T_C^{ML})^2 P_{x'_D} T_D^{\nu_2} & \text{if } C = W, \\ T_D^{\nu_1} R_D T_C^{ML} T_C^{ML} P_{x'_D} T_D^{\nu_2} & \text{if } C = F, \end{cases}$$

$$T_C^{ML'} = T_C^{\nu_1} R_C T_B^{ML'} P_{x'_c} T_C^{\nu_2}, \quad x'_D = x_D T_D^{\nu_1},$$

and when  $\tilde{Q}_C$  is the coarsest grid and solved exactly,  $T_C^{ML} = T_C^{ML'} = (ey'_C)/(x_C e) > 0$ , where  $y'_C = \tilde{\pi}_C$ .

**COROLLARY 4.18.** *When POWER is the smoother and  $x_D > 0$  satisfies  $x_D e = 1$ , the ML iteration matrix  $T_D^{ML}$  for  $C \in \{V, W, F\}$  is a positive stochastic matrix and therefore has a spectral radius of 1.*

*Proof.* For the POWER smoother, at the coarsest level  $C$  we have  $T_C^{ML} = T_C^{ML'} = e\tilde{\pi}_C$  from Definition 4.17 when  $x_D e = 1$ , implying a positive stochastic matrix, which has a spectral radius and an eigenvalue value of 1. This forms the base case. Now, let us assume that the result is true for all levels from the coarsest up to an arbitrary level  $C$ ; this is the inductive hypothesis. We show that the result must be true for the next finer level  $D$ . Noting that  $R_D e = e$  from Proposition 3.5,  $(T_C^{ML})e = e$  from the inductive hypothesis,  $P_{x'_D} e = e$  from Proposition 3.7, and  $T_D e = e$  from Lemma 4.13, we have  $T_D^{ML} e = T_D^{\nu_1} R_D T_C^{ML} P_{x'_D} T_D^{\nu_2} e = T_D^{\nu_1} R_D T_C^{ML} (P_{x'_D} e) = T_D^{\nu_1} R_D (T_C^{ML} e) = T_D^{\nu_1} (R_D e) = T_D^{\nu_1} e = e$  for the V-cycle. The result follows similarly for W- and F-cycles.  $\square$

The interpretation of  $T_D^{ML}$  for V- and W-cycles is as follows. If the recursive call(s) to level  $C$  are turned off, then only  $(\nu_1 + \nu_2)$  iterations are performed on  $x_D$  with the smoother  $S$ . Otherwise, the smoothed solution vector is restricted to level  $C$  (i.e.,  $x_D T_D^{\nu_1}$  is the smoothed solution vector and  $x_D T_D^{\nu_1} R_D$  is the restricted solution vector), the restricted solution vector is improved respectively one or two times with the iteration matrix  $T_C^{ML}$ , and the improved solution vector is projected back to level  $D$  and smoothed. The interpretation of  $T_D^{ML}$  for an F-cycle is similar to that for V- and W-cycles with the difference that the restricted solution vector is improved with the iteration matrix  $T_D^{ML}$  once followed by the iteration matrix of the V-cycle. This is exactly what is meant with a W-cycle followed by a V-cycle at each level.

The next lemma follows from Lemma 4.1, Lemma 4.13, and Definition 4.17.

**LEMMA 4.19.** *If  $\tilde{Q}_D$  is an irreducible CTMC,  $x_D > 0$ , and the smoother  $S \in \{POWER, JOR, SOR\}$  satisfies  $\alpha_D \in (\max_{s_D \in \mathcal{S}_D} |\tilde{q}_D(s_D, s_D)|, \infty)$  and  $\omega \in (0, 1)$ , then the ML iteration matrix  $T_D^{ML}$  for  $C \in \{V, W, F\}$  is positive.*

*Proof.* The proof is by induction. At the coarsest level  $C$ , we have  $T_C^{ML} = T_C^{ML'} > 0$  from Definition 4.17. This is the base case and implies  $(T_C^{ML})^2 = T_C^{ML} T_C^{ML'} > 0$ . Now, let us assume that the statement is true for all levels from the coarsest up to an arbitrary level  $C$ . This is the inductive hypothesis. We show that the statement must be true for the next finer level  $D$ . Since  $P_{x'_D} \geq 0$  and each column of  $P_{x'_D}$  has one nonzero element from Proposition 3.7, the  $(|\mathcal{S}_C| \times |\mathcal{S}_D|)$  matrices  $T_C^{ML} P_{x'_D}$ ,  $(T_C^{ML})^2 P_{x'_D}$ , and  $T_C^{ML} T_C^{ML'} P_{x'_D}$  are positive. Furthermore, since  $R_D \geq 0$  and each row of  $R_D$  has one nonzero element from Proposition 3.5, the  $(|\mathcal{S}_D| \times |\mathcal{S}_D|)$  matrices  $R_D T_C^{ML} P_{x'_D}$ ,  $R_D (T_C^{ML})^2 P_{x'_D}$ , and  $R_D T_C^{ML} T_C^{ML'} P_{x'_D}$  are also positive. Then the result follows from Lemma 4.13 by the fact that the iteration matrix associated with the smoother is nonnegative and irreducible, implying at least one nonzero in each row and column of  $T_D$  which pre- and postmultiplies the positive matrices  $R_D T_C^{ML} P_{x'_D}$ ,  $R_D (T_C^{ML})^2 P_{x'_D}$ , and  $R_D T_C^{ML} T_C^{ML'} P_{x'_D}$ .  $\square$

The next result follows from Lemma 4.19 in that the positivity of  $T_D^{ML}$  implies its irreducibility and a positive diagonal, and hence its primitivity [4, p. 47].

**COROLLARY 4.20.** *If  $\tilde{Q}_D$  is an irreducible CTMC,  $x_D > 0$ , and the smoother  $S \in$*

$\{POWER, JOR, SOR\}$  satisfies  $\alpha_{\mathcal{D}} \in (\max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|, \infty)$  and  $\omega \in (0, 1)$ , then the ML iteration matrix  $T_{\mathcal{D}}^{ML}$  for  $C \in \{V, W, F\}$  is irreducible and primitive.

The next lemma shows that the steady state vector,  $\pi_{\mathcal{D}}$ , of the exactly aggregated grid,  $Q_{\mathcal{D}}$ , is the unique, positive, unit 1-norm fixed point of the ML iteration matrix,  $T_{\mathcal{D}}^{ML}$ , at level  $\mathcal{D}$  upon convergence.

LEMMA 4.21. *If  $\tilde{Q}_{\mathcal{D}}$  is an irreducible CTMC and equal to  $Q_{\mathcal{D}}$ ,  $x_{\mathcal{D}} = \pi_{\mathcal{D}}$ , and the smoother  $S \in \{POWER, JOR, SOR\}$  satisfies  $\alpha_{\mathcal{D}} \in (\max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|, \infty)$  and  $\omega \in (0, 1)$ , then the ML iteration matrix  $T_{\mathcal{D}}^{ML}$  for  $C \in \{V, W, F\}$  has the unique positive fixed point  $\pi_{\mathcal{D}}$  (i.e.,  $\pi_{\mathcal{D}} T_{\mathcal{D}}^{ML} = \pi_{\mathcal{D}}$ ) such that  $\pi_{\mathcal{D}} e = 1$ ; furthermore,  $\rho(T_{\mathcal{D}}^{ML}) = 1$  and  $y'_{\mathcal{D}} = \pi_{\mathcal{D}}$ .*

*Proof.* The proof is by induction. At the coarsest level  $\mathcal{C}$ , we have  $\tilde{Q}_{\mathcal{C}} = Q_{\mathcal{C}}$  and  $x_{\mathcal{C}} = \pi_{\mathcal{C}} > 0$ , implying  $T_{\mathcal{C}}^{ML} = T_{\mathcal{C}}^{ML'} = e\pi_{\mathcal{C}} > 0$  from Definition 4.17. This positive matrix is stochastic and has the unique positive fixed point  $\pi_{\mathcal{C}}$  such that  $\pi_{\mathcal{C}} e = 1$ . Furthermore, it has a spectral radius of 1 and  $y'_{\mathcal{C}} = x_{\mathcal{C}} T_{\mathcal{C}}^{ML} = \pi_{\mathcal{C}}(e\pi_{\mathcal{C}}) = (\pi_{\mathcal{C}} e)\pi_{\mathcal{C}} = \pi_{\mathcal{C}}$ . This is the base case and yields  $(T_{\mathcal{C}}^{ML})^2 = T_{\mathcal{C}}^{ML} T_{\mathcal{C}}^{ML'} = (e\pi_{\mathcal{C}})(e\pi_{\mathcal{C}}) = e(\pi_{\mathcal{C}} e)\pi_{\mathcal{C}} = e\pi_{\mathcal{C}} > 0$ . Now, let us assume that the statement is true for all levels from the coarsest up to an arbitrary level  $\mathcal{C}$ . This is the inductive hypothesis. We show that the statement must be true for the next finer level  $\mathcal{D}$ .

Since  $x_{\mathcal{D}} = \pi_{\mathcal{D}} > 0$  is the fixed point of  $T_{\mathcal{D}}$ ,  $\pi_{\mathcal{D}} R_{\mathcal{D}} = \pi_{\mathcal{C}}$  from Definition 3.4,  $\pi_{\mathcal{C}} T_{\mathcal{C}}^{ML} = \pi_{\mathcal{C}}$  by the inductive hypothesis, and  $\pi_{\mathcal{C}} P_{\pi_{\mathcal{D}}} = \pi_{\mathcal{D}}$  from Definition 3.6, the result follows from Definition 4.17 for the V-cycle as  $y'_{\mathcal{D}} = \pi_{\mathcal{D}} T_{\mathcal{D}}^D = (\pi_{\mathcal{D}} T_{\mathcal{D}}^{\nu_1}) R_{\mathcal{D}} T_{\mathcal{C}}^{ML} P_{\pi_{\mathcal{D}}} T_{\mathcal{D}}^{\nu_2} = (\pi_{\mathcal{D}} R_{\mathcal{D}}) T_{\mathcal{C}}^{ML} P_{\pi_{\mathcal{D}}} T_{\mathcal{D}}^{\nu_2} = (\pi_{\mathcal{C}} T_{\mathcal{C}}^{ML}) P_{\pi_{\mathcal{D}}} T_{\mathcal{D}}^{\nu_2} = (\pi_{\mathcal{C}} P_{\pi_{\mathcal{D}}}) T_{\mathcal{D}}^{\nu_2} = \pi_{\mathcal{D}} T_{\mathcal{D}}^{\nu_2} = \pi_{\mathcal{D}}$ . The result follows similarly for W- and F-cycles after interchanging  $T_{\mathcal{C}}^{ML}$  respectively with  $(T_{\mathcal{C}}^{ML})^2$  and  $T_{\mathcal{C}}^{ML} T_{\mathcal{C}}^{ML'}$ . The uniqueness and positivity of the fixed point of  $T_{\mathcal{D}}^{ML}$  follows from Lemma 4.19 by the fact that  $T_{\mathcal{D}}^{ML}$  is positive [23, p. 666]. Clearly the spectral radius of  $T_{\mathcal{D}}^{ML}$  is 1.  $\square$

The next theorem characterizes the *nonasymptotic convergence* behavior of the ML solver with the initial approximation  $x_{\mathcal{D}}$  by defining a unique, positive, unit 1-norm fixed point for the particular cycle.

THEOREM 4.22. *If  $T_{\mathcal{D}}^{ML}$  is the ML iteration matrix that operates at level  $\mathcal{D}$  on  $x_{\mathcal{D}} > 0$ , such that  $x_{\mathcal{D}}^T \notin \text{Range}(I_{\mathcal{D}} - T_{\mathcal{D}}^T)$ , to give  $y'_{\mathcal{D}} > 0$  at a particular cycle using the smoother  $S \in \{POWER, JOR, SOR\}$  associated with the irreducible CTMC  $\tilde{Q}_{\mathcal{D}}$ , where  $\alpha_{\mathcal{D}} \in (\max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|, \infty)$  and  $\omega \in (0, 1)$ , then  $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})$  has a spectral radius of 1, and a vector  $\phi_{\mathcal{D}}$  as its unique positive fixed point (i.e.,  $\phi_{\mathcal{D}}(T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})) = \phi_{\mathcal{D}}$ ) such that  $\phi_{\mathcal{D}} e = 1$ . Furthermore,  $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}) = Z_{\mathcal{D}} H_{\mathcal{D}} Z_{\mathcal{D}}^{-1}$ , where  $H_{\mathcal{D}}$  is a positive stochastic matrix and  $Z_{\mathcal{D}}$  is a nonnegative diagonal matrix having the positive right eigenvector of  $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})$  along its diagonal. The unique positive fixed point of  $H_{\mathcal{D}}$  is given by  $\psi_{\mathcal{D}} = (\phi_{\mathcal{D}} Z_{\mathcal{D}})/(\phi_{\mathcal{D}} Z_{\mathcal{D}} e)$  (i.e.,  $\psi_{\mathcal{D}} H_{\mathcal{D}} = \psi_{\mathcal{D}}$ ) such that  $\psi_{\mathcal{D}} e = 1$ . Finally,*

$$\|(b_{\mathcal{D}}/\rho(T_{\mathcal{D}}^{ML}))y'_{\mathcal{D}} - \phi_{\mathcal{D}}\|_{Z_{\mathcal{D}}} \leq \left(1 - \min_{i_{\mathcal{D}}, j_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} h_{\mathcal{D}}(i_{\mathcal{D}}, j_{\mathcal{D}})\right) \|b_{\mathcal{D}} x_{\mathcal{D}} - \phi_{\mathcal{D}}\|_{Z_{\mathcal{D}}},$$

where  $b_{\mathcal{D}} = (\phi_{\mathcal{D}} Z_{\mathcal{D}} e)/(x_{\mathcal{D}} Z_{\mathcal{D}} e)$  and  $0 < \min_{i_{\mathcal{D}}, j_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} h_{\mathcal{D}}(i_{\mathcal{D}}, j_{\mathcal{D}}) \leq 1/|\mathcal{S}_{\mathcal{D}}|$ . At the coarsest level,  $\|(b_{\mathcal{D}}/\rho(T_{\mathcal{D}}^{ML}))y'_{\mathcal{D}} - \phi_{\mathcal{D}}\|_{Z_{\mathcal{D}}} = 0$  if the system is solved directly. When POWER is the smoother,  $Z_{\mathcal{D}} = I_{\mathcal{D}}$ ,  $H_{\mathcal{D}} = T_{\mathcal{D}}^{ML}$ ,  $\rho(T_{\mathcal{D}}^{ML}) = 1$ ,  $\psi_{\mathcal{D}} = \phi_{\mathcal{D}}$ , and  $b_{\mathcal{D}} = 1$ .

*Proof.* Recall from Lemma 4.19 that  $T_{\mathcal{D}}^{ML} > 0$ . Since  $\rho(T_{\mathcal{D}}^{ML}) > 0$  for  $T_{\mathcal{D}}^{ML} \neq 0$ , the matrix  $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})$  is also positive, satisfies  $\sigma(T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})) = \{\lambda/\rho(T_{\mathcal{D}}^{ML}) \mid \lambda \in \sigma(T_{\mathcal{D}}^{ML})\}$ , and therefore has a spectral radius of 1. The uniqueness and positivity

of the fixed point  $\phi_{\mathcal{D}}$  follow from Corollary 4.20. The row vector  $\phi_{\mathcal{D}} > 0$  is assumed to be normalized so as to have unit 1-norm (i.e.,  $\phi_{\mathcal{D}}e = 1$ ).

To prove the second part, recall Corollary 4.20 and the result in [4, p. 49], which is also used in the proof of Lemma 4.13. These imply that  $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})$  must have a positive right eigenvector  $\zeta_{\mathcal{D}}$  for which

$$T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}) = Z_{\mathcal{D}}H_{\mathcal{D}}Z_{\mathcal{D}}^{-1},$$

where  $Z_{\mathcal{D}} = \text{diag}(\zeta_{\mathcal{D}})$ ,  $H_{\mathcal{D}} > 0$ , and  $H_{\mathcal{D}}e = e$ . In other words,

$$H_{\mathcal{D}} = Z_{\mathcal{D}}^{-1}(T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}))Z_{\mathcal{D}}$$

is a stochastic matrix similar to  $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})$ , and its positivity follows from  $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}) > 0$  and  $\zeta_{\mathcal{D}} > 0$ . Note that it does not matter whether  $\zeta_{\mathcal{D}}$  is normalized or not, since  $H_{\mathcal{D}}$  is defined in terms of  $Z_{\mathcal{D}}$  and  $Z_{\mathcal{D}}^{-1}$ . The uniqueness and positivity of the fixed point  $\psi_{\mathcal{D}}$  follows from  $H_{\mathcal{D}} > 0$ . The row vector  $\phi_{\mathcal{D}} > 0$  is assumed to be normalized so as to have unit 1-norm (i.e.,  $\phi_{\mathcal{D}}e = 1$ ), and it is given by  $\psi_{\mathcal{D}} = \phi_{\mathcal{D}}Z_{\mathcal{D}}$  since  $H_{\mathcal{D}}$  and  $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML})$  are related by a similarity transformation, where the transformation matrix is  $Z_{\mathcal{D}}$ .

To prove the last part, rewrite

$$y'_{\mathcal{D}} = \rho(T_{\mathcal{D}}^{ML})x_{\mathcal{D}}(T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}))$$

using  $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}) = Z_{\mathcal{D}}H_{\mathcal{D}}Z_{\mathcal{D}}^{-1} > 0$  as

$$(y'_{\mathcal{D}}Z_{\mathcal{D}})/(\rho(T_{\mathcal{D}}^{ML})x_{\mathcal{D}}Z_{\mathcal{D}}e) = (x_{\mathcal{D}}Z_{\mathcal{D}})H_{\mathcal{D}}/(x_{\mathcal{D}}Z_{\mathcal{D}}e),$$

which is equivalent to  $\bar{y}'_{\mathcal{D}} = \bar{x}_{\mathcal{D}}H_{\mathcal{D}}$ . Since  $x_{\mathcal{D}} > 0$ ,  $y'_{\mathcal{D}} > 0$ ,  $\rho(T_{\mathcal{D}}^{ML}) > 0$ , and  $\zeta_{\mathcal{D}} > 0$ , we have  $\bar{x}_{\mathcal{D}} = (x_{\mathcal{D}}Z_{\mathcal{D}})/(x_{\mathcal{D}}Z_{\mathcal{D}}e) > 0$ , implying  $\bar{x}_{\mathcal{D}}e = 1$ , and  $\bar{y}'_{\mathcal{D}} = (y'_{\mathcal{D}}Z_{\mathcal{D}})/(\rho(T_{\mathcal{D}}^{ML})x_{\mathcal{D}}Z_{\mathcal{D}}e) > 0$ . Furthermore, since  $H_{\mathcal{D}} > 0$ ,  $H_{\mathcal{D}}e = e$ , and  $\bar{x}_{\mathcal{D}}e = 1$ , we obtain  $\bar{y}'_{\mathcal{D}}e = 1$ . Then, from Lemma A.1 in [13, appendix] we have

$$\|\bar{y}'_{\mathcal{D}} - \psi_{\mathcal{D}}\|_1 \leq \left(1 - \min_{i_{\mathcal{D}}, j_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} h_{\mathcal{D}}(i_{\mathcal{D}}, j_{\mathcal{D}})\right) \|\bar{x}_{\mathcal{D}} - \psi_{\mathcal{D}}\|_1.$$

The result follows by taking each of  $(\bar{y}'_{\mathcal{D}} - \psi_{\mathcal{D}})$  and  $(\bar{x}_{\mathcal{D}} - \psi_{\mathcal{D}})$  into  $Z_{\mathcal{D}}$  parentheses, multiplying both sides of the inequality by  $\phi_{\mathcal{D}}Z_{\mathcal{D}}e$ , letting  $b_{\mathcal{D}} = (\phi_{\mathcal{D}}Z_{\mathcal{D}}e)/(x_{\mathcal{D}}Z_{\mathcal{D}}e)$ , and using Proposition 4.15. The part for the coarsest level follows from Definition 4.17 by the fact that  $T_{\mathcal{D}}^{ML} = (ey'_{\mathcal{D}})/(x_{\mathcal{D}}e)$  and  $\rho(T_{\mathcal{D}}^{ML}) = 1/(x_{\mathcal{D}}e)$ , implying  $T_{\mathcal{D}}^{ML}/\rho(T_{\mathcal{D}}^{ML}) = H_{\mathcal{D}} = ey'_{\mathcal{D}}$  and  $Z_{\mathcal{D}} = I_{\mathcal{D}}$ . For the *POWER* smoother, Corollary 4.18 implies  $Z_{\mathcal{D}} = \text{diag}(e) = I_{\mathcal{D}}$ , and therefore, the respective results.  $\square$

The ML iteration matrix,  $T_{\mathcal{D}}^{ML}$ , changes at each cycle due to the dependence of  $P'_{x_{\mathcal{D}}}$  on  $x'_{\mathcal{D}}$ , and therefore the ML iteration is nonstationary. At the end of each ML iteration, the solution vector at the finest level  $\mathcal{D}$ ,  $y'_{\mathcal{D}}$ , is normalized to be unit 1-norm and then assigned to  $x_{\mathcal{D}}$  so as to start the next ML iteration. As long as  $x'_{\mathcal{D}} \neq \pi_{\mathcal{D}}$ , the aggregated CTMC  $\tilde{Q}_{\mathcal{C}}$  at the next coarser level can be only approximative. Theorem 4.22 indicates that the *normalized* solution vector,  $b_{\mathcal{D}}x_{\mathcal{D}}$ , improves with respect to the fixed point  $\phi_{\mathcal{D}}$  with a converging smoother as long as  $x_{\mathcal{D}} > 0$  is not in the range of  $(I - T_{\mathcal{D}}^{ML})^T$ . For the solution to improve with respect to steady state vector  $\tilde{\pi}_{\mathcal{D}}$  at each level, one requires sufficient conditions on the smoother, as in Theorem 4.16.

Then  $x_{\mathcal{D}}$  at the finest level will improve from one ML iteration to the next, implying an improvement in the aggregated CTMC at each level and thus an improved solution at each level. Then, recalling from Lemma 4.21 that  $\tilde{Q}_{\mathcal{D}} = Q_{\mathcal{D}}$  and  $\rho(T_{\mathcal{D}}^{ML}) = 1$  upon convergence,  $\rho(T_{\mathcal{D}}^{ML})$  and  $\phi_{\mathcal{D}}$  must be approaching 1 and  $\pi_{\mathcal{D}}$ , respectively, while the subdominant eigenvalue of  $T_{\mathcal{D}}^{ML}$  in magnitude is approaching zero with an increasing number of ML iterations.

In [11], extensive numerical experiments have been conducted with the ML solver on HMMs. Therein, the values chosen for the parameters of the *POWER*, *JOR*, and *SOR* smoothers are  $\alpha_{\mathcal{D}} = \max_{s_{\mathcal{D}} \in \mathcal{S}_{\mathcal{D}}} |\tilde{q}_{\mathcal{D}}(s_{\mathcal{D}}, s_{\mathcal{D}})|/0.999$  and  $\omega = 1$ , and the initial approximation is the uniform distribution. Furthermore, at least one pre- and one postsmoothing is performed at each level, and the coarsest system is solved using Gaussian elimination. Hence, *POWER* is enforced to yield a converging smoother, and the *JOR* and *SOR* iteration matrices are nonnegative. Although  $\omega = 1$  does not guarantee converging *JOR* and *SOR* smoothers (see Lemma 4.8), the results indicate that convergence may still be achieved. Hence, we conclude that the conditions stated in Theorem 4.16 for the smoothers are sufficient for convergence, but not necessary.

**5. Experimental results.** In [13, sec. 5], we step through the ML method on Example 1 in section 2. Here we consider the number of iterations and the time in seconds required to reach  $\|r\|_{\infty} < 10^{-8}$  (see Algorithm 1) for Examples 2 and 3. We compare *SOR* and ML methods with  $(C, S, O) \in \{(V, \textit{SOR}, \textit{FIXED}), (W, \textit{SOR}, \textit{CIRCULAR}), (F, \textit{SOR}, \textit{DYNAMIC})\}$ . In all cases, the relaxation parameter of *SOR* is set to 1. All experiments are performed on PCs with AMD opteron 2.3 GHz CPU and 1 GBytes of main memory.

TABLE 5.1  
Number of iterations and solution times for Example 2.

| $K$ | SOR  |        | ML                                  |        |  |        |                                       |        |
|-----|------|--------|-------------------------------------|--------|--|--------|---------------------------------------|--------|
|     | $it$ | $time$ | $(V, \textit{SOR}, \textit{FIXED})$ |        | $(W, \textit{SOR}, \textit{CIRCULAR})$ |        | $(F, \textit{SOR}, \textit{DYNAMIC})$ |        |
|     | $it$ | $time$ | $it$                                | $time$ | $it$                                   | $time$ | $it$                                  | $time$ |
| 3   | 180  | 0      | 90                                  | 0      | 44                                     | 0      | 46                                    | 0      |
| 4   | 260  | 0      | 106                                 | 0      | 40                                     | 0      | 42                                    | 0      |
| 5   | 290  | 2      | 92                                  | 1      | 34                                     | 1      | 36                                    | 1      |
| 6   | 360  | 42     | 104                                 | 8      | 30                                     | 3      | 36                                    | 4      |
| 7   | 420  | 123    | 114                                 | 57     | 30                                     | 19     | 32                                    | 20     |

Table 5.1 contains the results for Example 2. For the solution we choose *SOR* and three variants of ML methods. For the latter we choose  $\nu_1 = \nu_2 = 1$  in all cases. It can be seen that the number of iterations of *SOR* increases with an increasing number of LLMs. For the ML solver with *FIXED* aggregation of LLMs, a small increase in the number of iterations can also be observed. For the other two ML solvers with *CIRCULAR* and *DYNAMIC* order of aggregating LLMs, the number of ML iterations does not increase, even becomes smaller with an increasing number of LLMs, and is much smaller than that of the corresponding *FIXED* order. This behavior can be observed for all cycle types; it depends on the order of aggregation and shows the importance of modifying the order for this example. Hence, although convergence does not depend on the order of aggregating LLMs, rate of convergence does. It should be mentioned that this example is not particularly hard to solve with *SOR* since LLMs are strongly coupled and the number of iterations is fairly small. Nevertheless, the use of ML steps increases convergence speed significantly, reducing the solution times for the larger configurations by almost an order of magnitude.

The third example is much harder to solve with *SOR* or other classical iterative

TABLE 5.2  
 Number of iterations and solution times for Example 3.

| $K$ | SOR    |         | ML  |        |   |        |  |        |
|-----|--------|---------|---|--------|---|--------|--|--------|
|     | $it$   | $time$  | $(V, SOR, FIXED)$<br>$(\nu_1 = 1, \nu_2 = 1)$ |        | $(V, SOR, FIXED)$<br>$(\nu_1 = 1, \nu_2 = 5)$ |        | $(W, SOR, CIRCULAR)$<br>$(\nu_1 = 1, \nu_2 = 1)$ |        |
|     |        |         | $it$  | $time$ | $it$  | $time$ | $it$   | $time$ |
| 2   | 60     | 0       | 12  | 0      | 12  | 0      | 12   | 0      |
| 3   | 400    | 0       | 12  | 0      | 12  | 0      | 14   | 0      |
| 4   | 3,200  | 21      | 12  | 0      | 12  | 0      | 18   | 0      |
| 5   | 26,560 | 3,310   | 12  | 2      | 12  | 3      | 20   | 4      |
| 6   |        | >10,000 | 18  | 45     | 12  | 42     | 14   | 38     |
| 7   |        |         | 16  | 492    | 12  | 554    | 14   | 529    |

methods. With the addition of a new LLM a new time scale is introduced in the model. It is known that such models are difficult to solve. Results for Example 3 are shown in Table 5.2. The results for SOR therein indicate that with an increasing number of LLMs the number of iterations grows drastically, and the system becomes practically unsolvable for  $K > 5$ . In the ML methods with fixed order of aggregation, at every aggregation step the fastest time scale is removed, and the system is mainly solved for the fastest remaining time scale. This implies that during a cycle each time scale is considered. Thus, we can expect fast convergence, which is confirmed by the results in Table 5.2. The number of iterations is almost independent of the number of LLMs, and even the largest configuration with 10,000,000 states can be solved in less than 10 minutes, whereas SOR requires almost an hour to solve the system with only 10,000 states. Since the ordering of LLMs is optimal according to the time scales, *CIRCULAR* or *DYNAMIC* ordering of aggregation do not help. The last two columns contain results for *CIRCULAR* ordering and a *W*-cycle; results are similar to the *FIXED* ordering. *DYNAMIC* ordering gives worse results since the projected residuals which we use as a heuristic for choosing LLMs to be aggregated depend on the transition rates such that LLMs with small rates are aggregated first, resulting in a poor convergence in this example.

**6. Conclusion.** In this paper, the convergence of a class of multilevel (ML) methods for large sparse Markov chains (MCs) has been investigated. The particular class of ML methods is inspired by algebraic multigrid and iterative aggregation-disaggregation, and has the capability of using (V, W, F) cycles, (power, Jacobi over relaxation (JOR), successive over relaxation (SOR)) methods as smoothers, and (fixed, circular, dynamic) orders in which coarser MCs can be formed by aggregation in a cycle. The conditions sufficient for convergence are an irreducible MC, a positive initial approximation from an appropriate subspace, an onto mapping of states from a finer MC to a coarser MC at each level, a uniformization parameter larger than the minimum magnitude of the diagonal elements for the power method, a relaxation parameter less than 1 for JOR and SOR, a sufficient number of pre- and postsmoothings at each level so as to ensure a smoothing matrix which is positive or has a(n almost) positive row/column, and the accurate solution of the coarsest system at each cycle. The asymptotic convergence rate of the class of ML methods across multiple levels is yet to be investigated.

**Acknowledgments.** We thank the anonymous referees for their constructive remarks and suggesting some of the references.



## REFERENCES

- [1] F. BAUSE, P. BUCHHOLZ, AND P. KEMPER, *A toolbox for functional and quantitative analysis of DEDS*, in Quantitative Evaluation of Computing and Communication Systems, R. Puigjaner, N. N. Savino, and B. Serra, eds., Lecture Notes in Comput. Sci. 1469, Springer-Verlag, Heidelberg, Germany, 1998, pp. 356–359.
- [2] R. BELLMAN, *Introduction to Matrix Analysis*, 2nd ed., Classics in Appl. Math. 19, SIAM, Philadelphia, PA, 1997.
- [3] M. BENZI AND T. DAYAR, *The arithmetic mean method for finding the stationary vector of Markov chains*, Parallel Algorithms and Appl., 6 (1995), pp. 25–37.
- [4] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Classics in Appl. Math. 9, SIAM, Philadelphia, PA, 1994.
- [5] A. BOROBIA AND J. MORO, *On nonnegative matrices similar to positive matrices*, Linear Algebra Appl., 266 (1997), pp. 365–379.
- [6] W. L. BRIGGS, V. E. HENSON, AND S. F. MCCORMICK, *A Multigrid Tutorial*, 2nd ed., SIAM, Philadelphia, PA, 2000.
- [7] P. BUCHHOLZ, *Hierarchical structuring of superposed GSPNs*, IEEE Trans. Softw. Engrg., 25 (1999), pp. 166–181.
- [8] P. BUCHHOLZ, *Structured analysis approaches for large Markov chains*, Appl. Numer. Math., 31 (1999), pp. 375–404.
- [9] P. BUCHHOLZ, *Multilevel solutions for structured Markov chains*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 342–357.
- [10] P. BUCHHOLZ AND T. DAYAR, *Block SOR for Kronecker structured Markovian representations*, Linear Algebra Appl., 386 (2004), pp. 83–109.
- [11] P. BUCHHOLZ AND T. DAYAR, *Comparison of multilevel methods for Kronecker structured Markovian representations*, Computing, 73 (2004), pp. 349–371.
- [12] P. BUCHHOLZ AND T. DAYAR, *Block SOR preconditioned projection methods for Kronecker structured Markovian representations*, SIAM J. Sci. Comput., 26 (2005), pp. 1289–1313.
- [13] P. BUCHHOLZ AND T. DAYAR, *On the Convergence of a Class of Multilevel Methods for Large, Sparse Markov Chains*, Technical Report BU-CE-0601, Department of Computer Engineering, Bilkent University, Ankara, Turkey, 2006; available online from <http://www.cs.bilkent.edu.tr/tech-reports/2006/BU-CE-0601.pdf>.
- [14] P. BUCHHOLZ, G. CIARDO, S. DONATELLI, AND P. KEMPER, *Complexity of memory-efficient Kronecker operations with applications to the solution of Markov models*, INFORMS J. Comput., 12 (2000), pp. 203–222.
- [15] P. BUCHHOLZ AND P. KEMPER, *On generating a hierarchy for GSPN analysis*, Performance Eval. Rev., 26 (1998), pp. 5–14.
- [16] P. FERNANDES, B. PLATEAU, AND W. J. STEWART, *Efficient descriptor-vector multiplications in stochastic automata networks*, J. ACM, 45 (1998), pp. 381–414.
- [17] P. FERNANDES, B. PLATEAU, AND W. J. STEWART, *Optimizing tensor product computations in stochastic automata networks*, RAIRO Oper. Res., 32 (1998), pp. 325–351.
- [18] A. GREENBAUM, *Analysis of a multigrid method as an iterative technique for solving linear systems*, SIAM J. Numer. Anal., 21 (1984), pp. 473–485.
- [19] G. HORTON AND S. LEUTENEGGER, *A multi-level solution algorithm for steady state Markov chains*, Performance Eval. Rev., 22 (1994), pp. 191–200.
- [20] U. KRIEGER, *Numerical solution of large finite Markov chains by algebraic multigrid techniques*, in Computations with Markov Chains, W. J. Stewart, ed., Kluwer Academic Publishers, Boston, MA, 1995, pp. 403–424.
- [21] 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.
- [22] 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.
- [23] C. D. MEYER, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, PA, 2000.
- [24] J. W. RUGE AND K. STÜBEN, *Algebraic multigrid*, in Multigrid Methods, S. F. McCormick, ed., Frontiers in Appl. Math. 3, SIAM, Philadelphia, PA, 1987, pp. 73–130.
- [25] U. RÜDE, *The Multigrid Workbench*, <http://www.mgnet.org/mgnet/tutorials/xwb.html>.
- [26] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, PA, 2003.
- [27] S. SERRA, *Multi-iterative methods*, Comput. Math. Appl., 26 (1993), pp. 65–87.
- [28] G. W. STEWART, *Matrix Algorithms, Vol. I: Basic Decompositions*, SIAM, Philadelphia, PA, 1998.
- [29] W. J. STEWART, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.

- [30] J.-P. TREMBLAY AND R. MANOHAR, *Discrete Mathematical Structures with Applications to Computer Science*, McGraw-Hill, New York, 1975.
- [31] E. UYSAL AND T. DAYAR, *Iterative methods based on splittings for stochastic automata networks*, *European J. Oper. Res.*, 110 (1998), pp. 166–186.
- [32] C. F. VAN LOAN, *The ubiquitous Kronecker product*, *J. Comput. Appl. Math.*, 123 (2000), pp. 85–100.
- [33] P. WESSELING, *An Introduction to Multigrid Methods*, John Wiley & Sons, Chichester, New York, 1992.