https://doi.org/10.20378/irbo-52617

Numerical Solution Methods for Large Finite Markov Chains

U.R. Krieger
Deutsche Bundespost Telekom
Research and Technology Centre Am Kavalleriesand 3
D-64295 Darmstadt, Germany
fz224c@vmxa.fz.telekom.de

Abstract

Modeling and analysis of modern computer and telecommunication systems by homogeneous discrete- or continuous-time Markov chains with finite state spaces is a standard performance evaluation technique. We sketch the interactive construction of the corresponding models by the advanced software techniques of the tool MACOM. We present efficient algorithms to compute the steady-state distributions of the underlying finite Markov chains and illustrate their application by some examples arising from teletraffic theory. For further study of the subject a comprehensive bibliography is provided.

Key Words
finite Markov chains, steady-state distribution, computational methods for Markov chains, direct methods, iterative methods

1 Introduction

Modeling and analysis of advanced distributed systems such as computer and telecommunication systems in a narrow- or broadband ISDN by means of homogeneous discrete- and continuous-time Markov chains (DTMCs, CTMCs) with finite or infinite state spaces is a current performance evaluation technique originating in Erlang's pioneering work [11]. To cope with the investigation of large Markovian models, numerical solution methods are often the only feasible approach since explicit analytic solutions of the steady-state distributions and the relevant performance characteristics are normally not available and simulation may be too expensive or too difficult, particularly in the case of rare events. For instance, cell loss probabilities arising from the investigation of multiplexers or switching fabrics in BISDN are in the range of 10^{-7} to 10^{-9} (cf. [115]). Therefore, considerable attention has recently been devoted to the analysis of finite Markov chains by computational methods (cf. [131], [78],

[2]). For Markov chains with infinite state spaces the latter can be derived by state space truncation techniques (cf. [48]).
From a theoretical point of view, the problem of computing the steady-state distribution of an ergodic positive recurrent homogeneous Markov chain has already been solved since it can be calculated as nonnegative normalized solution of a homogeneous linear system, called steady-state balance equations (cf. [63], [49]). The corresponding transient solution satisfies a system of linear ODEs with constant coefficients. Hence, both issues are not very attractive to theorists any more. In practice, however, the application of the theory raises a lot of difficulties due to the enormous complexity of the models regarding computing time and memory requirements.
Considering the sketched performance evaluation approach of distributed systems there are two main issues: first, the difficulties to translate an application-oriented formulation of the system such as a queueing, stochastic Petri-net or SDL model into the low logical level of the underlying Markov chain and, secondly, the management of the perhaps immense number of states of the derived Markov chain and the sophisticated relations of these mathematical objects, i.e., state descriptions, the transitions between states and their corresponding rates. To analyze large models at least by parts, suitable techniques including special decomposition methods have to be employed. Moreover, the process of constructing and evaluating Markovian models has to be supported by convenient software tools. For these reasons, a software package called MACOM (Markovian analysis of communication systems) has been developed (cf. [123], [78], [79], [80]). Its capabilities are tailored to the requirements arising from the performance evaluation of modern distributed systems by means of queueing networks. Efficient numerical procedures for finite continuous-time Markov chains are employed as solution technique (cf. [53], [43], [66], [4]). The package can cope with large models comprising up to 150000 states. It is implemented in C and currently available on a workstation with a graphical user interface.

In this paper we present the advanced numerical solution methods employed in MACOM. It is our main objective to summarize these recommended simple algorithms within a well-known mathematical framework. To illustrate their application, several teletraffic models are used. We also discuss the construction of the Markov chains since it is a very important ingredient of any solution technique. We do not claim to present new mathematical results nor to give a comprehensive survey of computational methods for finite Markov chains and their application. Readers who are interested in the mathematical background and details of the proposed algorithms are referred to the literature and the references therein (see the excellent surveys of Barker [2] and Stewart et al. [107], also [78]). Although our bibliography cannot be exhaustive, it is comprehensive enough to provide a source for further studies of the subject.
The paper is organized as follows: In section 2 we discuss the modeling and construction techniques used in MACOM. Section 3 provides both the mathematical background for calculating the steady-state distribution of a finite Markov chain and a classification of computational methods. In section 4 we present some important direct methods. Section 5 is concerned with

point and block iterative procedures derived from matrix splittings of singular M-matrices. In section 6 the proposed methods are applied to some teletraffic models arising from the analysis of modern telecommunication networks. Finally, the merits of MACOM are summarized in the conclusions.

2 Modeling and analysis by Markovian techniques

The tool MACOM combines state-of-the-art techniques for systems modeling and analysis by finite continuous-time Markov chains, software methods for the specification of models by graphical means and for the enhanced support of the users by menu techniques and the advanced technology of modern workstations (cf. [122], [14], [99]). It provides an interactive environment for the management of modeling data, the description and performance evaluation of models and the presentation of results. All steps of the evaluation process are supported by a graphical user interface and enhanced by convenient window techniques. Besides the construction of models combining predefined graphical elements, the tool offers functions to specify evaluation schemes and series of experiments. Furthermore, it performs the evaluation of the specified performance measures automatically.

MACOM supports modeling and analysis of modern distributed systems by a predefined model world in terms of queueing networks. They consist of multi-class delay-loss stations of type $\sum_i \mathrm{MAP}_i^{d_i}/\overrightarrow{\mathrm{PH}}/n/m$ and allow state-dependent routing (cf. [123]). To evaluate the performance of a system, e.g. time- or arrival-stationary distributions, throughputs or loss rates, the steady-state vector of the underlying finite CTMC is calculated. The structures of the modeling components guarantee the Markovian nature of a specified model. The basic elements consist of sinks, loss exits, links, specific routing elements, called conexes, offering probabilistic and state-dependent routing, queueing stations with multiple, homogeneous servers and queues with finite capacities, different service disciplines (infinite server, processor sharing, random, non-preemptive/preemptive priority, polling) and general Coxian service-time distributions, as well as sources. The latter generate versatile arrival streams such as phase-type renewal processes, Markov-modulated Poisson (MMPP) processes and general Markov arrival (MAP) processes. Therefore, the resulting class of queueing models includes Markovian queueing networks with state-dependent routing which do not possess a steady-state distribution of product form.

The model world of MACOM is provided by means of a graphical user interface. Models are specified by a graphical editor (see Fig. 1). The resulting network graph is interactively constructed from predefined elements. Moreover, further textual information is added to the model elements by means of pop-up menus. They are activated by a mouse. The text comprises the types of the distributions (Coxian, Erlang, Hyperexponential, Exponential) of the service and arrival processes and the parameters of a station, i.e. its name, the service discipline, the number of processors, the service speed, which may depend on the population, and the capacity of the station defined as the number of customers which can be held simultaneously. Furthermore, the attributes of the routing elements have to be described.

The performance characteristics of a system can be evaluated after calculating the steady-

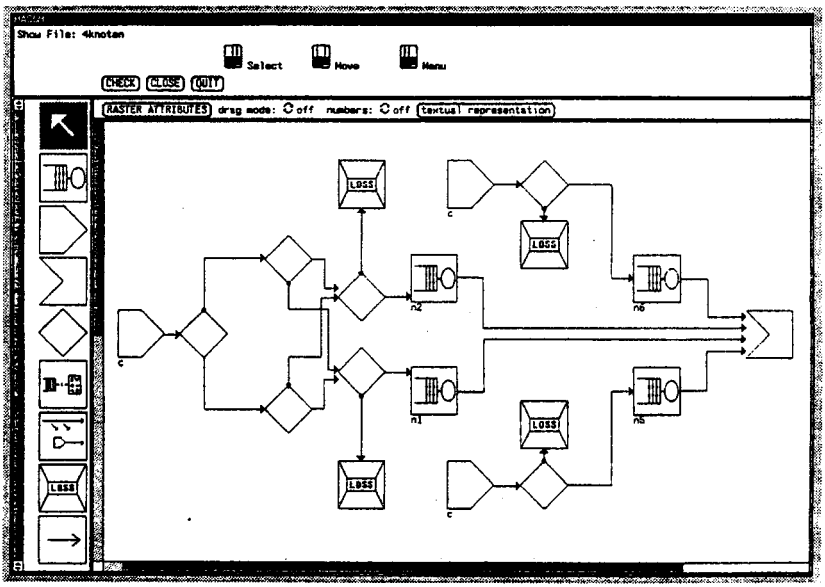

Figure 1: MACOM model of a circuit-switched network with mutual overflow routing

state vector of the finite CTMC derived from the graphical model. This is a complex process that must be performed in several steps. Hereby, it is a major benefit of MACOM that the generator matrix Q of the CTMC is generated automatically (cf. [14]). For this purpose, MACOM has a layered software structure formed by three different levels of abstraction, called High (HLL), Medium (MLL) and Low Level Language (LLL) representation (cf. Fig. 2). The HLL reflects an application-oriented view of the model. It specifies the model and its relevant performance characteristics in terms of a graphical representation of a queueing network (cf. Fig. 1). The MLL arises from a construction-oriented view. It describes the model and its parameters in terms of an event-oriented representation, i.e. by states and transitions caused by all feasible events. The LLL is a mathematical-oriented view representing the model in terms of the generator matrix of its underlying CTMC.

In the HLL the model and its required characteristics are specified by combining predefined graphical elements of the Markovian model world (cf. [123]). Then the HLL is transformed into the MLL replacing each graphical item by an associated vector of state variables, a set of inital states and transition rules describing the behavior of the component.

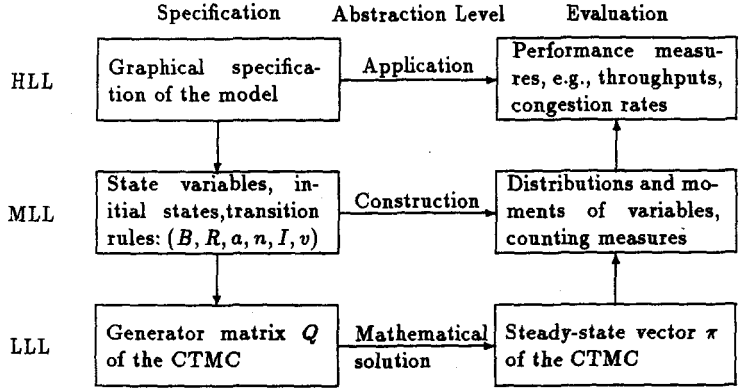

Figure 2: Structure of MACOM

In the MLL, the model is described by a state-transition scheme (B, R, a, n, I, v) (cf. [14, §5]). It comprises the products of all state descriptions B and initial states $I \subset B$ and the finite set R of all transition rules. These rules $r \in R$ specify whether a transition $b \to b' \equiv (b, b') \in D(r) = \{(b, b') \in B \times B \mid n(r, b) = b', a(r, b) = 1\}$ of a specific type r can occur in a state b by $a(r, b) = 1$ (or whether it cannot by $a(r, b) = 0$) and how the new state $b' \in B$ is determined by $n(r, b) = b'$. Additionally, nonnegative numbers $q \in \mathbb{R}$ are attached to each transition $b \to b' \in D(r)$ by $v(r, b, b') = q$. They depend on its type r and describe the transition rates or branching probabilities associated with a transition. Performance characteristics are represented by random variables on the state space, i.e. by functions on the state variables and transitions between the corresponding states (cf. [14, §7]).

The second transformation of MLL into LLL is logically divided into two steps, namely, the generation of the state space and the construction of the generator matrix $Q \in \mathbb{R}^{n \times n}$. This transformation generates the lists of all possible states of the CTMC and all feasible transitions between these states. Its result may be represented by a directed attributed graph (S, T, v_T) comprising the states as nodes $s \in S$ and the transitions as edges $t = (s, s') \in T$. Attributes of a node s are the values of the corresponding vectors of state variables, attributes of an edge t are the types r of the occurring transitions including their corresponding rates $v_T(s, s') \in \mathbb{R}$ (cf. [14, §5]). Loops and multiple links between states are admitted. Applying the transition rules $r \in R$ successively to all initial states $b \in I$ and their successors $b' = n(r, b)$ in a way similar to attributed graph grammars, where the attributes of the state variables b control the selection of all applicable transition rules $\{r \in R \mid a(r, b) = 1\}$, the state space of the underlying CTMC is incrementally generated (cf. [14, §6.1]). This generation procedure may be considered as breadth-first-search in the fictitious graph representation of the state space. Starting from the initial states $b \in I$, all reachable states

$d(b) = \{b' \mid \exists r \in R, n(r,b) = b', a(r,b) = 1\}$ and transitions $\{b \to b' \in B \times B \mid b' \in d(b)\}$ will be generated and visited exactly once. In the case of a finite state space the procedure terminates. To determine whether a reached state $b' = n(r,b)$ is new, a search in the set of already generated states is required. This operation is the complexest step of the algorithm that must be performed very efficiently. Therefore, hashing is used as basic technique in MACOM. Based on the structure of the states the hash function is automatically generated.
Finally, the construction of the generator matrix Q consists of the enumeration of all states according to a fixed ordering, the evaluation of the matrix elements and their integration into a data structure arising from a given sparse matrix storage scheme. The enumeration is represented by a position mapping $p : S \longrightarrow \{1, ..., |S|\} = S_n \subset \mathbb{N}$ on the set S with $|S| = n$ states. The simplest enumeration is determined by the order of generation. But macrostructures of the model can also be taken into account. If the states of a macrostate are enumerated in a sequential order and macrostates are ordered in a lexicographical way, a block structured matrix Q results. This process may be represented by an appropriate partition and permutation on S_n. Then the generator matrix Q is built determining first its zero structure, i.e. the adjacency matrix $Z(Q) = (z(s_1, s_2))$ with $z(s_1, s_2) = 1$ if $(s_1, s_2) \in T$ and zero otherwise. Each pair (s_1, s_2) of directly connected states yields a matrix entry in position $(p(s_1), p(s_2))$. The corresponding value is computed according to the type r of the applied transition and the attributes of the states. Values of multiple edges between states are added. The diagonal elements are set to the negative sums of the off-diagonal elements of the rows (cf. [14, §6.2]). In a way similar to other applications, queueing network models determine generator matrices Q that are large sparse nonsymmetric Metzler-Leontief matrices (ML-matrices) with zero row sums (see section 3 - cf. [7]). As it is more convenient to use the transposed generator matrix in numerical algorithms, Q^t is built up by MACOM during the construction process. It is represented by a row-oriented sparse matrix storage scheme based on a list $(q_1, \ldots, q_n)$ of the rows $q_j = (Q_{lj})_{l=1,\ldots,n}$ of Q^t, i.e. the columns of Q. One stored row contains the value of the diagonal element Q_{jj}, an array for the nonzero off-diagonal elements Q_{lj} of that row j and an array for the corresponding column indices $l \in \{1, \ldots, n\} \setminus \{j\}$. This storage scheme is similar to one employed in ITPACK (cf. [72]), but it uses dynamic memory allocation.
The stationary distribution p of the CTMC is computed as the solution of the linear system

$$Q^t \cdot p = 0, \quad \sum_{i=1}^{n} p_i = 1. \tag{2.1}$$

It is unique if Q is irreducible. In MACOM this condition can be verified automatically using Tarjan's algorithm. The numerical solver at the Markovian level (LLL) employs advanced direct and iterative methods for the solution of (2.1) (cf. [78]). For small matrices with some hundred states a direct method based on an LU-decomposition technique is employed (cf. [53], [43]). For larger non-banded matrices iterative techniques based on the Gauss-Seidel scheme and the successive overrelaxation procedure (SOR) are used (cf. [66], [4]). They may be accelerated by relaxation techniques or by inserting some aggregation-disaggregation steps (cf. [96], [61], [121], [76], [120]). In the following sections these procedures are discussed in detail.

Then the required performance characteristics, defined in terms of this steady-state vector p and specified by the user in terms of an evaluation description, are automatically computed by MACOM. Further details concerning this forward and backward transformations are discussed in [123] and [14, §5–7].

3 The mathematical background of computational methods

The behavior of distributed systems is often described by homogeneous discrete- or continuous-time Markov chains (DTMCs/CTMCs) with discrete valued finite state spaces $S = \{1, 2, \ldots, n\}$ (see section 6 - [78], [66], [89]). Then it is an important task to calculate the corresponding steady-state distributions. For this purpose, computational solution methods must be employed if the models do not belong to special classes such as product-form networks (cf. [69]) or matrix-geometric models (cf. [101]) since analytic solutions are rarely available.

In this section we provide the mathematical background of these computational methods based on a simple mathematical framework. It is derived from the theory of nonnegative matrices and M-matrices. We assume the reader to be familiar with this theory to the extent of Berman's and Plemmons' book [7] and adopt their notation regarding vector and matrix orderings (see [7, Chap. 2, p. 26]): Let $x \in \mathbb{R}^n$, then $x \gg 0 \Leftrightarrow x_i > 0$ for each $i \in \{1, \ldots, n\}$; $x > 0 \Leftrightarrow x_i \geq 0$ for each $i \in \{1, \ldots, n\}$ and $x_j > 0$ for some $j \in \{1, \ldots, n\}$; $x \geq 0 \Leftrightarrow x_i \geq 0$ for each $i \in \{1, \ldots, n\}$.

We show that, from the computational point of view, discrete- and continuous-time Markov chains can be treated in the same way. For this purpose, we assume that the reader is familiar with the theory of DTMCs/CTMCs to the extent of the books of Heyman and Sobel [63, Chap. 7, 8] and Kemeny and Snell [71].

For an irreducible DTMC $\{Y_n, n \geq 0\}$ with a *finite* state space $S = \{1, 2, \ldots, n\}$ and an irreducible transition probability matrix (t.p.m.) P, a solution of the stated problem is provided by Markov's well-known theorem (cf. [63, Corollary 7-4, p. 231], [25, Corollary 2.11, p. 153]). The vector π comprising its stationary probabilities is the unique solution of the linear system:

$$z^t = z^t P, \quad \sum_{i \in S} z_i = 1, \quad z_i \geq 0 \quad \text{for all } i \in S \tag{3.1}$$

Considering an irreducible CTMC $\{X(t), t \geq 0\}$ with finite state space S and the irreducible generator matrix $Q = (q_{ij})$, the vector p of its steady-state distribution is determined by the solution of the system

$$x^t Q = 0, \quad \sum_{i \in S} x_i = 1, \quad x_i \geq 0 \quad \text{for all } i \in S \tag{3.2}$$

(cf. [63, Theorem 8-6 , p. 304]).

Moreover, it is well known that the existence of these steady-state distributions can be proved

by algebraic results only (cf. [46, §13.7, p. 431ff], [7, Theorem 8.3.11]). In this context, a major mathematical tool is given by the distinguished Perron-Frobenius theorem for nonnegative matrices (cf. [7, Chap. 2], [46, §13.2, p. 397f], [125, Theorem 1.5, p. 20], [25, Appendix, Sec. 4], [142, Theorem 2.1, p. 30]). This set of nonnegative matrices, particularly the subset of probability matrices, is very important in computational probability theory.

It is well known that under appropriate regularity conditions (i.e. for conservative, uniformizable MCs - cf. [63, §8-7]) homogeneous continuous-time and discrete-time Markov chains with denumerable state spaces are related to each other in a specific stochastic manner. There is a natural way to associate an embedded DTMC $\{Y_n, n \geq 0\}$, called the *jump chain*, with each such CTMC possessing a standard generator matrix Q. This tranformation of a CTMC into its associated embedded DTMC is called *uniformization* or *randomization* (cf. [65, p. 228], [69, p. 3], [63, p. 290 and p. 310]). It is arising from the CTMC by observing its states only immediately after a state transition. Thus the transition probabilities $p_{ij} = P(Y_{n+1} = j \mid Y_n = i)$ of Y_n are just the conditional probabilities of a transition of $X(t)$ from state i to state j given that a transition occurs, i.e. $p_{ij} = \begin{cases} q_{ij}/q_i & , i \neq j \\ 0 & , i = j \end{cases}$ with $q_i = -q_{ii} = \sum_{j:j \neq i} q_{ij}$. Hence the t.p.m. $P = (p_{ij})$ of $\{Y_n, n \geq 0\}$ is defined by

$$P = D^{-1} \cdot B = I + D^{-1}Q \tag{3.3}$$

where $Q = (q_{ij}) = -D + B$ is a decomposition into the negative of the diagonal part of Q, $D = -\mathrm{Diag}(q_{11}, \ldots, q_{nn}) = \mathrm{Diag}(q_1, \ldots, q_n) \geq 0$, and the nonnegative off-diagonal elements $B \geq 0$. Here $\mathrm{Diag}(x) = D$ denotes a diagonal matrix generated from any list $(x_1, \ldots, x_n)$ of elements such as a vector x by setting $D_{ii} = x_i$. Subsequently, let e denote the vector with all ones.

If the vector z comprises the stationary distribution of the jump chain, we conclude from (3.3) that

$$p = \frac{D^{-1}z}{e^t D^{-1} z} = \left(\frac{z_j/q_j}{\sum_l z_l/q_l} \right)_{j \in S} \tag{3.4}$$

is a probability vector determined by the equation (3.2). If the parameters q_j of the exponentially distributed sojourn times in the states would not depend on j, i.e. $q = q_j$, then the jump epochs of the embedded chain form a Poisson process with rate q and the stationary distribution z of the jump chain coincides with the steady-state vector p of the CTMC (cf. [63, Theorem 8-5, p. 303]).

A matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ is called *M-matrix* if $a_{ij} \leq 0$ holds for all $i, j \in S$ with $i \neq j$ and if A can be represented in the form $A = sI - B$, $s > 0$, $B \geq 0$, $s \geq \rho(B)$. Here $\rho(B)$ denotes the spectral radius of B and I is the identity matrix (cf. [7, Chap. 6]). A is a regular M-matrix if $s > \rho(B)$ holds and a singular M-matrix for $s = \rho(B)$. An important subset of the class of singular M-matrices is given by the *Q-matrices*. A Q-matrix $Q = (q_{ij}) \in \mathbb{R}^{n \times n}$ satisfies the conditions $q_{ij} \leq 0$ for all $i, j \in S$ with $i \neq j$ and $\sum_{i=1}^{n} q_{ij} = 0$ for all $j \in \{1, \ldots, n\}$. A is a called *ML-matrix* or Metzler-Leontief matrix if $-A$ is an M-matrix. Thus the generator matrix Q of a CTMC is an ML-matrix and the negative transpose $A = -Q^t$ is a Q-matrix

(cf. [109]). If P is the t.p.m. of a DTMC, then $A = I - P^t$ is also a singular M-matrix with zero column sums, i.e. its associated Q-matrix.
To calculate the vector π of the steady-state distribution of such an irreducible Markov chain with finite state space $S = \{1, \ldots, n\}$, we have to compute a positive solution $x \in \mathbb{R}^n$ of the linear system

$$A \cdot x = 0 \tag{3.5}$$

$$\sum_{i=1}^{n} x_i = e^t \cdot x = 1. \tag{3.6}$$

Here $A \in \mathbb{R}^{n \times n}$ is a nonsymmetric irreducible singular M-matrix with zero column sums, i.e., an irreducible Q-matrix with positive diagonal elements. If A is reducible, the same problem arises from the transformation of A into the well-known block triangular canonical form (cf. [7, p. 39], [46, §13.4]). In the following we assume that A denotes always an irreducible nonsymmetric Q-matrix. For practical purposes, a diagonally scaled version $\tilde{A} = A \cdot [\mathrm{Diag}(A_{11}, \ldots, A_{nn})]^{-1}$ of matrix A with diagonal elements $\tilde{A}_{ii} = 1$ should always be used. The resulting transformation $\tilde{A} \cdot z = 0$, $z = \mathrm{Diag}(A_{11}, \ldots, A_{nn}) \cdot x$ of the system (3.5) has the same stochastic interpretation as relation (3.4) and yields a stochastic matrix $T = I - \tilde{A}$ with diagonal elements $T_{ii} = 0$ (see (3.3)).
From the theory of M-matrices we know that the system (3.5), (3.6) has a unique positive solution $x \gg 0$ (cf. [7, Theorem 6.4.16, p. 156]). Furthermore, $\lambda = 0$ is a simple eigenvalue of A with one-dimensional eigenspaces spanned by a positive right and left eigenvector, i.e. π, e with $e^t\pi = 1$.
Solution methods for system (3.5) have been discussed in detail in [78], [107], and [2]. The proposed computational procedures can be classified according to the formulation of the problem and the applied numerical solution technique. They can be divided into the following categories:

(I) procedures based on the formulation by a linear system (3.5)

(II) procedures based on a formulation by an eigenvalue problem

$$x = T \cdot x \tag{3.7}$$

using $T = P^t$ or $T = I + Q^tD^{-1} = B^tD^{-1}$ with $Q = -D + B$ and $D = -\mathrm{Diag}(Q_{11}, \ldots, Q_{nn}) \geq 0$ or $D = qI$, $q \geq max_{i=1,\ldots,n}(-Q_{ii}) > 0$ (uniformization procedure for a CTMC).

A survey of some direct and iterative solution methods is provided by Table 1. There is a major difference between our context and similar singular systems with the structure (3.5) arising from other fields of application, e.g., from system theory or the theory of difference methods for the solution of the Neumann problem (cf. [42], [100], [108]). It is given by the possibility to interpret some well-known direct or iterative solution methods such as Gaussian elimination, the Gauss-Seidel scheme, and the underrelaxation procedure in a stochastic manner (cf. [117],

Category	Methods	Algorithms	References
Direct methods	Gaussian elimination techniques	LU decomposition for singular M-matrices	[42], [45], [44], [7, Cor. 6.4.17, p. 157]
		GTH algorithm	[53], [52], [126]
		Crout-algorithm	[50, p. 58]
	QR method		[50, Sec. 7.5, p.228], [51]
	rank reduction techniques	block LU algorithms	[75], [47], [58], [50, Sec. 5.5], [5], [117], [49, §12.4], [146]
		blockwise bordering	[49, §12.3], [24]
	deflation techniques		[70], [60], [106], [59]
Iterative methods	inverse iteration		[66], [67], [130]
	procedures based on matrix splittings for singular M-matrices and their accelerated variants	Jacobi procedure	[3], [66], [95]
		Gauss-Seidel procedure	[3], [66], [133]
		JOR	[16]
		SOR	[133], [55]
		SSOR	[35]
		block SOR	[66], [67]
		AOR	[28], [116]
		A/D algorithms	[22], [18], [74], [119], [121], [76]
	including		
	semi-iterative methods	Chebyshev algorithms	[33], [66], [142], [56]
	multi-splitting methods	parallel SOR algorithms	[104], [145], [68], [41], [116], [12], [34]
	conjugate gradient techniques	CG procedure	[112], [19], [20], [21]
		ICCG	[19], [35, Sec. 3, p.45ff]
	projection methods	power method	[144], [130], [17], [66], [3]
		simultaneous iteration	[130], [129]
		Lanczos' procedure and Arnoldi's procedure	[111], [110], [17], [105], [107], [113], [114]
		GMRES	[111], [107]
		QMR, TFQMR	[64], [39], [37], [38], [40]
	multigrid algorithms	AMG	[9, Sec. 10, p. 257ff], [29], [135], [30], [54]
		IAD schemes	[76], [77], [138], [22], [61], [134]

Table 1: Survey of numerical solution methods for finite Markov chains

[53], [52], [126], [146], [83], [73], [47], [76], [77]).
The transformation of the singular system (3.5) into a regular one is an obvious solution technique. The substitution of the normalization (3.6) for the last equation of (3.5) is an approach used very often. Let $e_n \in \mathbb{R}^n$ denote the nth unit vector. Then this procedure yields the inhomogeneous system $(A - e_n e_n^t A + e_n e^t) \cdot x = e_n$ with a unique positive solution x. However, Barlow [5], [6] has shown that a transformation of the singular system (3.5) into a regular inhomogeneous one by specifying a component of the solution vector x and deleting the corresponding row of A will not significantly change or improve the condition of the linear system. Similar observations have been derived from extensive numerical comparisons (cf. [60], [66], see also [24]).
An alternative approach is given by a deflation technique proposed by Kemeny [70] (see also [71, VIII, p. 211 ff], [59]). It is a rank-1-modification that can be applied to any matrix A with a simple eigenvalue. But it destroys the sparsity and sign pattern of A. Therefore, these well-known transformation techniques are not used in MACOM.
To select an appropriate algorithm, the structure of the given Q-matrix A and the properties of the chosen procedure must be taken into account. Some selection criteria are listed in Table 2. The advantages and disadvantages of direct and iterative procedures are discussed by Evans [35, Sec. 3, p. 45f] (cf. also [140]). Computational aspects concerning the efficient implementation of iterative methods on a vector computer are treated by Kincaid et al. [72].

4 Direct methods

In this section we study the solution of the homogeneous system (3.5) by direct methods, i.e. variants of Gaussian elimination. After the solution phase, the vector x must be normalized such that (3.6) is satisfied. But it is not necessary to proceed to a regular system by deflation or rank-1-modification techniques which are often proposed in the literature (cf. [106], [97]).

4.1 A block LU-algorithm

Solution techniques from numerical linear algebra reducing the order n of the linear system (3.5) are very often discussed in the literature (cf. [49], [66], [75], [106], [62], [53], [126], [73], [146], [117], [83]). All these procedures are based on a block partition of the irreducible Q-matrix

$$A = \begin{pmatrix} A_{11} & \vdots & A_{12} \\ \cdots & \cdots & \cdots \\ A_{21} & \vdots & A_{22} \end{pmatrix} \tag{4.1}$$

into matrices $A_{11} \in \mathbb{R}^{k_1 \times k_1}$, $A_{12} \in \mathbb{R}^{k_1 \times k_2}$, $A_{21} \in \mathbb{R}^{k_2 \times k_1}$, $A_{22} \in \mathbb{R}^{k_2 \times k_2}$ with $k_2 \geq 2, k_1 + k_2 = n$. A_{12} and A_{21} are nonpositive. A_{11} and A_{22} are regular M-matrices that are weakly diagonally dominant with respect to columns, i.e. $e^t A_{ii} > 0, i = 1, 2$.
Based on this partition, a block elimination scheme can be applied to reduce the solution of (3.5) to that of smaller subproblems. The following block Gaussian elimination procedure is

Criteria	Recommended Methods	
	modified Gaussian elimination	Iterative procedures based on splittings
Properties of Q-matrix A		
order n	$n \leq 500$	$n > 500$
structure:		
dense	recommended	suitable
banded	for small bandwidth	for large bandwidth
block structured	for additional structure	generally recommended
sparse	-	recommended
spectral properties:		
well-conditioned	suitable	suitable
ill-conditioned including NCD-type	-	recommended
Properties of the procedure		
algorithmic complexity:	$s = \max_{1 \leq i \leq n}\{\sharp\{A_{ij} \neq 0, 1 \leq j \leq n\}\}$	
regarding time	$O(2/3n^3)$	$O(n \times s)$ per step
regarding memory	$O(n^2)$	$O(n \times s)$
efficiency of the implementation:		
algorithmic structure	matrix-vector scheme	matrix-vector scheme
data structures	matrices, vectors	matrices, vectors or sparse matrix schemes
vectorizable or parallel executable	yes	yes
accuracy aspects:		
stability w.r.t. rounding errors	yes	yes
appropriate termination test	-	available

Table 2: Criteria for the choice of a procedure

a classical approach (cf. [50, Sec. 5.5, p. 110ff], [117], [75], [47]):

Algorithm:

1. Choose a nontrivial partition $A^{(1)} = A$ of the irreducible Q-matrix A according to (4.1).
2. Compute a factorization $A_{11}^{(1)} = L_{11}^{(1)} U_{11}^{(1)}$ by Gaussian elimination without pivoting (Crout-algorithm - cf. [45, Cor. 1], [42], [7, Cor. 6.4.17]).
3. Calculate a nonnegative solution $R^{(2)}$ of $R^{(2)} L_{11}^{(1)} U_{11}^{(1)} = -A_{21}^{(1)}$ by forward elimination and back substitution (taking into account $e^t R^{(2)} = e^t$ as accuracy check).
4. Set

$$A_{22}^{(2)} = A_{22}^{(1)} + R^{(2)} A_{12}^{(1)}. \tag{4.2}$$

5. Calculate a solution $x^{(2)} > 0$ of
$$A_{22}^{(2)} x^{(2)} = 0 \ . \tag{4.3}$$

6. Compute a solution $x^{(1)} > 0$ of $A_{11}^{(1)} x^{(1)} = L_{11}^{(1)} U_{11}^{(1)} x^{(1)} = (-A_{12}^{(1)}) x^{(2)}$ by forward elimination and back substitution.

7. Normalize the resulting vector $x = \begin{pmatrix} x^{(1)} \\ x^{(2)} \end{pmatrix}$ by setting $\pi = x/(e^t x)$.

□

One step of this elimination scheme yields $A^{(1)} = L^{(1)} U^{(1)}$ with the regular M-matrix $L^{(1)} = \begin{pmatrix} I & 0 \\ A_{21}^{(1)} A_{11}^{(1)-1} & I \end{pmatrix}$ and the singular M-matrix $A^{(2)} = U^{(1)} = \begin{pmatrix} A_{11}^{(1)} & A_{12}^{(1)} \\ 0 & A_{22}^{(2)} \end{pmatrix}$. Here the irreducible Q-matrix $A_{22}^{(2)} = A_{22}^{(1)} + (-A_{21}^{(1)}[A_{11}^{(1)}]^{-1}) A_{12}^{(1)}$ is called the *Schur complement* of $A_{11}^{(1)}$ (cf. [50, P4.2-3., p. 58]). If we perform k_1 steps of the normal Gaussian elimination on matrix $A^{(1)}$, the resulting submatrix $A_{22}^{(k_1+1)}$ coincides with the Schur complement (4.2).

As $L^{(1)}$ is a regular matrix, the solution of $A^{(1)}x = 0$ is equivalent to that of $A^{(2)}x = 0$. In order to solve the linear system (4.3), the same reduction step can be applied to the resulting submatrix $A_{22}^{(2)}$ or any other direct or iterative solution technique for singular systems can be employed (cf. [43], [85], [84], [8]). The successive application of this reduction process is stopped after $m - 1 < n - 1$ steps. It yields a submatrix $A_{22}^{(m)}$ of order l with rank $l - 1$. Then we fix one component of the solution vector $x^{(m)}$ resulting from the decomposition of x according to the block partition of A, for instance, its last one $x_l^{(m)} = 1$, and solve the linear system $A_{22}^{(m)} x^{(m)} = 0$. The back substitution yields the other component vectors $x^{(m-1)}, \ldots, x^{(1)}$ of x.

Exploiting its Q-matrix structure, the computation of $A_{22}^{(2)}$ can be improved. As $A_{22}^{(1)}$ is a Q-matrix and $R^{(2)} \geq 0$, $A_{12}^{(1)} \leq 0$ hold, the off-diagonal elements of $A_{22}^{(2)}$ are computed only from nonpositive elements whereas the diagonal elements are given by the negative column sums of the off-diagonal elements. Hence, there is no loss of accuracy due to cancellation errors while performing these additions. Therefore, the computation of $A_{22}^{(2)}$ according to (4.2) is a rather stable process.

Considering the propagation of rounding errors, the matrix inversion which is required to calculate $R^{(2)}$ is the crucial step of the algorithm. It has to be performed very carefully. Le Boudec [84], for instance, proposes to use an inversion algorithm for strictly substochastic matrices based on the representation of the inverse by a power series.

A major advantage of the proposed block LU-algorithm is its inherent probabilistic interpretation (cf. [117], [53], [52], [73], [47]).

We recommend either to avoid successive reduction or to apply this block LU-algorithm if the algebraic structure of A can be exploited (cf. [66], [60], [85], [84]). In practice, for example, matrices with block tridiagonal structure $A = (A_{ij})_{1 \leq i,j \leq m}$, $A_{ij} = 0$ for $|i - j| > 1$, arise very often (cf. [66], [85], [147]). In this case, standard block LU-factorization [50, (5.5-4), p. 111] based on Gaussian elimination without pivoting can be applied. However, a necessary condition for the stability of the overall process is given by a block diagonally dominance

condition

$$\|A_{ii}^{-1}\| \left(\|A_{i-1i}\| + \|A_{i+1i}\|\right) < 1, \qquad i = 1, \ldots, m$$

with $A_{01} = A_{m+1m} = 0$ (cf. [50, Th. 5.5-1., p. 112], [147], [132]).

4.2 The GTH algorithm

Choosing a block decomposition of A with $k_1 = 1$, we can consecutively apply the proposed block elimination procedure to the generated block matrices $A_{22}^{(i)}$. If we calculate the negative values of its diagonal elements by means of relation $e^t A_{22}^{(i)} = 0$ as sum of the off-diagonal elements, we obtain a numerical procedure that does not contain any subtractions or additions of numbers with different signs. This *GTH algorithm* avoids cancellation errors, a main difficulty of Gaussian elimination, and it still coincides with the result of Gaussian elimination without pivoting applied to A. Grassmann, Taksar, and Heyman [53] have proved the feasibility of this approach for a DTMC with denumerable state space by probabilistic arguments. Furthermore, its application to CTMCs has been studied in [52] (see also [73]). The procedure can be derived from the factorization

$$A = \begin{pmatrix} B_{n-1} & y_n \\ z_n^t & A_{nn} \end{pmatrix} = \begin{pmatrix} L_{n-1} & 0 \\ z_n^t U_{n-1}^{-1} & 1 \end{pmatrix} \cdot \begin{pmatrix} U_{n-1} & L_{n-1}^{-1} y_n \\ 0 & 0 \end{pmatrix}.$$

Here $B_{n-1} = L_{n-1} U_{n-1}$ is a factorization of the leading principal submatrix of A of order $n-1$ into two regular M-matrices and U_{n-1} has only ones along the diagonal. $z_n^t \leq 0, y_n \leq 0$ denote the nth row and column vectors of A with exception of its last element. To avoid negative numbers, the corresponding algorithm works with $-A$ and looks as follows:

Assumptions:
Given an irreducible Markov chain with a finite state space $S = \{1, \ldots, n\}$, set $A = Q^t \in \mathbb{R}^{n \times n}$ in the case of a CTMC and $A = P^t - I$ in the case of a DTMC.
Algorithm:

1. Modified Gaussian elimination:

```
For k = 1 to n − 1 do
    Diag = Σ_{i=k+1}^{n} A_ik
    For j = k + 1 to n do
        A_kj = A_kj/Diag
        For i = k + 1 to n do
            A_ij = A_ij + A_ik A_kj
        Endfor
    Endfor
Endfor
```

2. Norm = 1
   $x_n = 1$

3. Back substitution:

   For $i = n - 1$ to 1 do
   $x_i = \sum_{k=i+1}^{n} A_{ik} \cdot x_k$
   Norm = Norm + x_i
   Endfor

4. Normalization:

   For $k = 1$ to n do
   $x_k = x_k/\text{Norm}$
   Endfor

□

The proposed algorithm requires $O(2/3n^3)$ flops and $O(n^2)$ memory places to calculate the steady-state vector π. Moreover, it has the advantage to preserve the band structure of a matrix. An implementation based on a sparse matrix storage scheme is also possible. The storage scheme should support access to matrix elements by rows and columns (cf. [31]).
Computing the steady-state vector in floating-point arithmetic with unit roundoff u, the relative error of the components of the steady-state vector π due to the initial roundoff of matrix A is in the worst case $O(2n)u$ and the relative error induced by the GTH algorithm itself is $9.54n^2u$. Hence, the relative error of the calculated probabilities is $O(n^3)u$ (cf. [103]). If double-precision arithmetic with $u = 5 \cdot 10^{-14}$ is used, the probabilities of a 1000-state Markov chain are computed with 4 correct digits and for 10000 states one digit is accurate (cf. [103]). Nevertheless, for ill-conditioned systems like NCD models modified methods such as the aggregation/disaggregation or stochastic complementation methods with a well-conditioned formulation of the coupling matrix are required to overcome instabilities (cf. [128], [93], [91], [94], [127]).

4.3 Further reading

From the mathematical literature it is known that only in the last elimination step of Gaussian elimination without pivoting a zero pivot occurs if we decompose an irreducible singular M-matrix (cf. [45, Cor. 1], [7, Cor. 6.4.17]). Therefore, the Crout-algorithm can be employed for the factorization of a Q-matrix that specifies the steady-state distribution of a Markov chain (cf. [42], [60], [59]).
Generally speaking, results concerning the factorization of a (singular) M-matrix A into a (singular) upper triangular M-matrix U having the same rank as A and a regular lower triangular M-matrix L are well known (cf. [36], [7, Theorem 6.2.3, p. 135], [82], [42], [45], [143, Theorem 3, p. 182], [1]). Although a singular M-matrix A does not possess an LU-decomposition in general (see [143, (1.15), p. 183]), it is known that a factorization into M-matrices always exists for a permuted version PAP^t (see Kuo [82, Theorem 2], also [7, Theorem 6.4.18], [45,

Cor. 1, p. 106]). In general, the LU-decomposition into M-matrices is not unique. In the special case of an irreducible singular matrix, however, uniqueness of the factorization can be established.

It is known that, even in the case of a singular linear system, this LU-factorization is a stable method with growth factor one (cf. [44], [60, Theorem 1], [59], [127]). The problem (3.5) is good conditioned relative to entrywise relative errors, i.e. the perturbation of the matrix elements by rounding errors due to their representation by finite floating-point numbers does not seriously harm the accuracy of the solution vector (cf. [103]). However, ill-conditioned systems such as NCD-models yield large condition numbers of order $O(\varepsilon^{-1})$ for a given coupling parameter ε (see (5.9) - cf. [128]) and cannot be solved accurately, in general. But even for well-conditioned systems, the adjustment of the diagonal elements according to the GTH approach is required to avoid cancellation errors and instability in practice (cf. [103]).

Further *block LU*-procedures for block tridiagonal matrices A have been studied by several authors (cf. [49, §12, p. 390ff], [146], [75], [62] and [47]). Meyer (cf. [91], [92], [87]) has recently extended the concept of block LU-factorization to nearly decomposable systems which are also known as *nearly completely decomposable* (NCD) models (cf. [26]). His method, called *stochastic complementation*, uncouples a Markov chain in several smaller independent chains. It is well suited for parallel computation (cf. [87]). Other variants of rank-reduction techniques that exploit particularly the sparsity structure of matrix A and properties of separable matrices are discussed by Kaufman [66].

An alternative direct method is given by a *QR-factorization* algorithm for singular M-matrices with zero row sums, i.e. $A = -Q$ or $A = I - P$ (cf. [50, Alg. 6.2-1, p. 148]). It yields a decomposition $A = \hat{Q}R \in \mathbb{R}^{n\times n}$ into an orthogonal matrix $\hat{Q}$ and $R = \begin{pmatrix} U & -Ue \\ 0 & 0 \end{pmatrix}$ with a regular upper triangular matrix $U \in \mathbb{R}^{n-1\times n-1}$ with positive diagonal elements. Then the steady-state distribution vector $\pi = \hat{Q} \cdot e_n/(e^t \cdot \hat{Q} \cdot e_n)$ is given by the normalized last column vector of $\hat{Q}$. From a computational point of view, however, this technique is not superior to the LU-factorization since it requires in most cases more work than the latter. More information about this procedure and its probabilistic relevance is provided by Golub and Meyer [51].

Further applications of LU-factorization algorithms are studied in the context of combined direct-iterative methods such as *incomplete LU-factorization*. The latter yields a decomposition $A = L \cdot U - R$ into a regular lower triangular M-matrix L with unit diagonal, an upper triangular M-matrix U and a nonnegative matrix R. These matrices satisfy $L_{ij} = U_{ij} = 0$ for $(i,j) \in \mathcal{P}$ where $\mathcal{P} \subset \mathcal{P}_n = \{(i,j) \in \mathbb{N}^2 \mid i \neq j, 1 \leq i,j \leq n\}$ is an arbitrary position set, e.g. reflecting the zero structure of A, $\mathcal{P} = \{(i,j) \in \mathbb{N}^2 \mid i \neq j, A_{ij} = 0\}$ (cf. [2], [90]). Then $A = M - N = L \cdot U - R$ is an M-splitting with $M^{-1} = U^{-1}L^{-1} \geq 0$ if $N = R > 0$ holds. It can be used as preconditioner of system (3.5): $0 = M^{-1}A \cdot x = (I - M^{-1}R)x$. Here $T = M^{-1}R = I - M^{-1}A$ is the nonnegative iteration matrix with steady-state vector $x > 0$. The resulting iteration scheme has the form

$$\begin{aligned} LUd^{(k)} &= -Ax^{(k)} \\ x^{(k+1)} &= x^{(k)} + d^{(k)} \qquad k = 0, 1, \ldots \end{aligned}$$

for a given initial vector $x^{(0)} \geq 0$. Normalising $x^{(k+1)}$ after convergence, the solution vector is obtained. The interested reader is referred to [43], [2], [15] and [90].
Another preconditioning technique derived from a decomposition $A = M - N \in \mathbb{R}^{n \times n}$ into a singular M-matrix M of $\mathrm{rank}(M) = n - 1$ with a separable structure $M = B \otimes I + I \otimes C$ has been studied by Chan [19], [20], [21]. Here $\otimes$ denotes the Kronecker product defined by $(B \otimes C) = (B_{ij}C)_{i,j}$ (cf. [66]).

4.4 Applicability of direct methods

We recommend to apply either the GTH algorithm or the proposed block LU-factorization if the diagonal dominance condition is fulfilled. The first has the additional advantage to avoid cancellation errors. It is suitable for computing the stationary distribution of small Markov chains with up to 500 states in double-precision arithmetic with approximately seven accurate digits if the latter do not belong to the class of nearly completely decomposable systems.
Apart from the latter case, the application of these direct methods is only limited by memory constraints and rounding errors. An efficient row-oriented implementation is sketched in [107]. The required availability of the entire matrix, the memory to store it during the elimination process by sparse matrix schemes as well as the limited accuracy due to the floating-point representation are the major limitations of direct methods.

5 Iterative methods

Iterative methods can also be employed to solve the linear system (3.5) for a given irreducible nonsymmetric Q-matrix $A \in \mathbb{R}^{n \times n}$. These methods are preferable for banded, sparse or large matrices with more than 1000 states. To apply an iterative procedure, its convergence must be guaranteed. As A is singular now, a straightforward application of standard results developed for regular matrices is impossible. Based on the framework of M-matrices, the required generalization of the classical convergence theory to singular M-matrices has fortunately been developed over the last decade (cf. [118], [109], [100], [16], [66]).
The classical iteration procedures are derived from an additive decomposition of matrix A, called *matrix splitting*. It has the form $A = M - N$, $M \in \mathbb{R}^{n \times n}$, $N \in \mathbb{R}^{n \times n}$ with a regular matrix M and yields the *iteration matrix* $T = M^{-1} \cdot N$. If $N \geq 0$ and $M^{-1} \geq 0$ hold, then $T = I - M^{-1}A \geq 0$ follows and the splitting is called *regular splitting*. Note that an M-matrix M satisfies $M^{-1} \geq 0$. The corresponding regular splittings are called *M-splittings* (cf. [118, Def. 2.3, p. 410]). Then an iteration procedure can be defined by $M \cdot x^{(k+1)} = N \cdot x^{(k)}$ or the fixed point iteration

$$x^{(k+1)} = T \cdot x^{(k)} \qquad k = 0, 1, \ldots, \tag{5.1}$$

respectively. The scheme (5.1) is called preconditioned power method (cf. [134]). Furthermore, $A \cdot x = 0$ is equivalent to $x = T \cdot x$ (cf. [66, Lemma 4.1]), i.e. each nontrivial solution of the homogeneous system (3.5) is also a right eigenvector corresponding to the eigenvalue $\rho(T) = 1$ of T and vice versa. Recall that due to the distinguished Perron-Frobenius Theorem

the spectral radius $\rho(T)$ of a nonnegative matrix T is always a real eigenvalue of T with maximal modulus and nonnegative right and left eigenvectors (cf. [7, Chap. 2]).

5.1 Point iteration methods

In the following, let $A = D - L - U$ be a *point splitting* into the diagonal part D, the strictly lower triangular part L and the strictly upper triangular part U of A.

5.1.1 The point Jacobi procedure

The *Jacobi procedure*

$$Dx^{(m+1)} = (L + U)x^{(m)} \qquad m = 0, 1, \ldots \tag{5.2}$$

results from the M-splitting $M = D$, $N = L + U$. As the diagonal D of an irreducible M-matrix is positive (cf. [7, Th. 6.4.16, p. 156]), this splitting with the Jacobi iteration matrix $J = D^{-1}(L + U)$ is well defined. The Jacobi procedure is equivalent to the power method applied to $D^{-1}A = I - D^{-1}(L + U)$.
It is known that any eigenvalues on the unit circle apart from $\rho(T) = 1$ prevent the convergence of the scheme (5.1) derived from a regular splitting $A = M - N$ of an irreducible singular M-matrix. On the other hand, there is the simple idea to shift these eigenvalues into the unit circle by an appropriate transformation $T_\alpha = \alpha T + (1 - \alpha)I = I - \alpha M^{-1}A$, $0 < \alpha < 1$, of T called stationary first-order Richardson extrapolation (cf. [71, Chap. V, Theorem 5.1.1, p. 99], [66, p. 540] and [100, p. 273 ff]). Then the resulting *extrapolated Jacobi method* (JOR) $x^{(k+1)} = J_\alpha \cdot x^{(k)} = [\alpha J + (1 - \alpha)I]^{k+1} \cdot x^{(0)}$ converges to a nontrivial solution of (3.5), which depends on $x^{(0)}$, for each $\alpha \in (0, 1)$ and each initial vector $x^{(0)} \gg 0$ (cf. [16, Theorem 3.4, p. 191], [100, p. 273 ff]).
Furthermore, the optimal extrapolation parameter α can be determined for some special cases (cf. [100, Theorem 7, p. 275], [7, Theorem 8.4.32, p. 234], [55], [56]).

5.1.2 The point Gauss-Seidel procedure

There are two possible Gauss-Seidel iteration procedures for the solution of the homogeneous system (3.5): the *forward Gauss-Seidel iteration*

$$(D - L)x^{(m+1)} = Ux^{(m)} \qquad m = 0, 1, \ldots \tag{5.3}$$

with the iteration matrix $T_L = (D - L)^{-1}U$ and the *backward Gauss-Seidel iteration*

$$(D - U)x^{(m+1)} = Lx^{(m)} \qquad m = 0, 1, \ldots \tag{5.4}$$

with the iteration matrix $T_U = (D - U)^{-1}L$. Note that the Gauss-Seidel splitting of an M-matrix is an M-splitting.
Simple examples reveal that the convergence of these schemes is not guaranteed for arbitrary M-matrices A. But it can be shown that a Gauss-Seidel scheme is convergent, too, if the iteration matrix T (-the index of T is omitted-) has no further eigenvalues on the unit circle

apart from the maximal eigenvalue $\rho(T) = 1$, i.e. if $\delta(T) = \max\{|\lambda|: \lambda \in \sigma(T), \lambda \neq 1\} < 1$ holds (cf. [3], [118], [109]). Here, $\sigma(T)$ denotes the spectrum of T. This criterion can be verified a priori (cf. [3, Theorem 1, p. 394]). There are some simple sufficient conditions for the convergence of the point Gauss-Seidel procedures which can easily be verified by the inspection of the zero structure of A (cf. [3, Cor. 1, p. 394], [118, Cor. 3.6], [109, Cor. 2]). Recall that the *directed matrix graph* $\Gamma(A) = (V, E)$ associated with $A \in \mathbb{R}^{n\times n}$ has nodes $V = \{1, \ldots, n\}$ and directed arcs $(i, j) \in E$ for all $A_{ij} \neq 0$. Then a sequence $\alpha = (i_0, i_1, \ldots, i_l, i_0)$ of distinct nodes (except the end point) in $\Gamma(A)$ is called a *monotone cycle* if $\alpha_1 = (i_0, i_1, \ldots, i_l)$ is monotone with either $i_l < i_0$ for decreasing α_1 or $i_l > i_0$ for increasing α_1 (cf. [109], [3]).

Result 1
If the directed graph $\Gamma(A)$ of the irreducible singular M-matrix A has a monotone decreasing *cycle, then the* forward *Gauss-Seidel iteration converges for each $x^{(0)} > 0$.*
If $\Gamma(A)$ has a monotone increasing *cycle, then the* backward *Gauss-Seidel iteration converges for each $x^{(0)} > 0$.*

We state another useful criterion for matrices with *symmetric zero structures* (cf. [43], [66]). These matrices A have the property that $A_{ij} \neq 0$ for $i \neq j$ implies $A_{ji} \neq 0$, i.e. $A_{ij}A_{ji} \neq 0$.

Result 2
If the irreducible M-matrix A has a symmetric zero structure, both the forward and backward Gauss-Seidel iterations are convergent.

To avoid expensive divisions during the execution of the Jacobi- or Gauss-Seidel procedure, the schemes should be applied to the irreducible M-matrix $A \cdot D^{-1} = I - \check{L} \cdot D^{-1} - \hat{U} \cdot D^{-1} = I - L - U$. It is derived from a scaling of the columns of A by the diagonal matrix $D^{-1} > 0$ (cf. [133], [95]). This transformation is equivalent to the transition from a CTMC to its embedded jump chain with the t.p.m. $P = I + D^{-1} \cdot Q = (L + U)^t$ and requires a scaling of the steady-state distribution π: $A \cdot D^{-1} \cdot (D \cdot \pi) = 0$, $x = D \cdot \pi$. Thus, the steady-state vector can be calculated from the vector x obtained after convergence of the iterative scheme by the modified normalization $\pi = D^{-1} \cdot x/(e^t \cdot D^{-1} \cdot x)$ provided that $x > 0$ holds (see (5.8)).

5.2 The block Gauss-Seidel procedure and R-regular splittings

An important class of iterative methods is arising from a block partition $A = (A_{ij})_{1\leq i,j\leq p}$, $p > 1$, of a given Q-matrix $A \in \mathbb{R}^{n\times n}$. Regarding the corresponding block iterative schemes, Rose [109] has established some convergence results. They are derived from a generalization of the block Gauss-Seidel splitting technique, called *R-regular splitting* (cf. [109, p. 138]). Appropriate LU-factorization methods for sparse matrices, e.g. an incomplete LU-factorization, can be used for the diagonal blocks of A to construct a splitting according to this approach. We state a variant of the corresponding iterative algorithm that is accelerated by relaxation.

Assumptions:
Let $A \in \mathbb{R}^{n \times n}$ be the Q-matrix associated with a Markov chain. Define a block partition $A = (A_{ij})_{1 \le i,j \le p}$, given $p > 1$, and extend it to the steady-state vector x specified by (3.5),(3.6). Choose an *R-regular block splitting* $A = M - N = (D - D(N)) - (L + L(N)) - U(N)$ which has the following properties:

(1) $D = \mathrm{Diag}(D_{ii})_{1 \le i \le p}$ and $D(N)$ are block diagonal matrices with $D(N) \ge 0$. L and $L(N)$ are strictly lower block triangular matrices with $L \ge 0, L(N) \ge 0$. $U(N)$ is a strictly upper block triangular matrix with $U(N) \ge 0$.

(2) $D_{ii}^{-1} \gg 0$ for $1 \le i \le p$. (If $0 < \omega < 1$ is used in the algorithm, only $D_{ii}^{-1} > 0$ is required.)

(3) $M = D - L$ is a lower block triangular matrix.

(4) $N = L(N) + U(N) + D(N) \ge 0$

(5) $A_0 = D - L - U(N)$ is irreducible.

(6) The block matrix graph $\Gamma(A_0) = (V, E)$ has a monotone decreasing cycle. This is a sequence $c = (i_1, i_2, \ldots, i_l, i_1)$ of adjacent nodes with the property $i_l \ne i_1$ and $i_j \ge i_{j+1}$ for $1 \le j \le l-1$.
The block matrix graph $\Gamma(A_0) = (V, E)$ is a directed matrix graph with nodes $V = \{V_i \mid 1 \le i \le p\}$ and directed edges $(V_i, V_j) \in E$. V_i results from the partition of the index set $\{1, \ldots, n\}$ according to the block partition. $(V_i, V_j) \in E$ iff $(A_0)_{ij} \ne 0$, that means, there are indices $l \in V_i, m \in V_j$ such that $(l, m) \in E_{\Gamma(A_0)}$ is an edge in the matrix graph $\Gamma(A_0) = (\{1, \ldots, n\}, E_{\Gamma(A_0)})$ of A_0.

Algorithm:

1. Initialization:
   Select an initial vector $x^{(0)} \gg 0$, e.g. $x^{(0)} = e/n$, $e \in (0,1)$, an appropriate relaxation parameter $\omega \in (0,2)$ and the number $m_T \in [1, 10]$ of consecutive iteration steps without termination test.
   Set $k = 0$, $d^{(0)} = 1, r^{(0)} = 1, e^{(0)} = 1$.

2. Iteration step:

   For $m = 1$ to m_T do
   For $i = 1$ to p do
   Solve $D_{ii} \cdot \tilde{x}_i^{(k+m)} = \sum_{j=1}^{i-1} L_{ij} \cdot x_j^{(k+m)} + \sum_{j=1}^{p} N_{ij} \cdot x_j^{(k+m-1)}$

   $$x_i^{(k+m)} = x_i^{(k+m-1)} + \omega \cdot (\tilde{x}_i^{(k+m)} - x_i^{(k+m-1)})$$

   endfor
   endfor

(Let $\sum_{j=1}^{0} \equiv 0$.)

3. Convergence test:

$$d^{(k+m_T)} = \omega \| \tilde{x}^{(k+m_T)} - x^{(k+m_T-1)} \|_\infty$$
$$r^{(k+m_T)} = \left(d^{(k+m_T)} / d^{(k)} \right)^{1/m_T}$$

If $r^{(k+m_T)} \geq 1$ then
$k = k + m_T$
goto step 2
endif

$$e^{(k+m_T)} = d^{(k+m_T)} \max(1, r^{(k+m_T)}/(1 - r^{(k+m_T)}))$$

If $e^{(k+m_T)} \leq \varepsilon$ then
goto step 4
else
$k = k + m_T$
goto step 2
endif

4. Normalization:

Normalize the resulting vector $x^{(k+m_T)} = \begin{pmatrix} x_1^{(k+m_T)} \\ \vdots \\ x_p^{(k+m_T)} \end{pmatrix} > 0$ by setting

$\pi = x^{(k+m_T)} / \| x^{(k+m_T)} \|_1$

□

Here, the term $r^{(k)}$, called reduction factor, converges to the modulus $\delta(T)$ of the subdominant eigenvalues of the iteration matrix $T = M^{-1}N$. Hence, it is an estimate of the rate of convergence. The term $e^{(k)}$ approximates the error of the solution π. If $r^{(k)}$ is close to one, then aggregation/disaggregation steps should be performed during the iteration to improve the rate of convergence. An approximation of the required number L of iteration steps to pass the termination test is given by $L \geq \ln(\varepsilon)/\ln(r^{(k)})$ (cf. [14, p. 35ff], [57, p. 20]).
Every R-regular splitting is also a regular splitting. If all diagonal blocks D_{ii} are M-matrices, it is an M-splitting. Without loss of generality we assume that D_{ii} are irreducible regular M-matrices which implies condition (2) and the irreducibility of A (cf. [109]). Regarding condition (6) the monotony of a cycle, which exists due to the irreducibility of A, can be enforced by an appropriate block permutation (cf. [109, Prop. 3, p. 138]).
The importance of R-regular splittings stems from Rose's result [109, Theorem 1, p. 138] that each R-regular splitting $A = M - N$ of an irreducible singular M-matrix $A \in \mathbb{R}^{n \times n}$ generates a convergent iteration scheme. Employing this special splitting technique, Rose has also shown that there always exists a convergent block Gauss-Seidel splitting of the irreducible Q-matrix corresponding to an irreducible Markov chain with appropriately ordered states (cf. [109, Theorem 1, p. 138]). In this context, the following sufficient convergence criterion can often be employed in practice (cf. [109, Cor. 2, p. 139]).

Result 3
Let $A \in \mathbb{R}^{n \times n}$ be an irreducible singular M-matrix, e.g. a Q-matrix, and $A = M - N = (D - L) - U(N)$ a block Gauss-Seidel splitting with irreducible block matrices D_{ii} along the diagonal of M. Suppose there exist block matrices $A_{ij} \neq 0$ and $A_{ji} \neq 0$ for $i, j \in \{1, \ldots, p\}$, $i \neq j$. Then the block Gauss-Seidel method converges.

Other sufficient convergence criteria concerning the block Jacobi and block Gauss-Seidel method as well as their implementation by a two-step iteration procedure derived from an A/D-scheme are provided by Courtois et al. [27]. These methods are particularly suitable for NCD models (see section 5.3).

5.3 The IAD method

Now we present a universal iterative solution procedure called *iterative aggregation/disaggregation* (IAD) method. It is derived from a partition $\Gamma = \{J_1, \ldots, J_m\}$ of the state space $S = \{1, \ldots, n\}$ into $m \geq 2$ disjoint sets J_i with $n_i \geq 1$ elements each. We assume that the elements of these sets are enumerated in a consecutive order such that $i < j$ holds for $i \in J_l, j \in J_k, l < k$. In the following, let $\mathcal{P} = \{x \in \mathbb{R}^n \mid x \geq 0, e^t x = 1\}$ denote the subset of all probability vectors in $\mathbb{R}^n$.
To illustrate the use of the IAD-method in the general setting, we assume that $A = M - N$ is a regular splitting of the irreducible Q-matrix associated with a finite Markov chain. We denote the nonnegative iteration matrix by $J = M^{-1}N$ and the corresponding extrapolated iteration matrix by $J_\omega = (1 - \omega)I + \omega J$, $0 < \omega < 1$. Then the insertion of some aggregation/disaggregation (A/D) steps during the iteration is a standard convergence acceleration technique that can be used in addition to the relaxation method mentioned previously (cf. [119], [121], [61], [22], [23], [98]). In order to guarantee the convergence of the resulting nonstationary iterative scheme (cf. [121, Theorem 4, p. 328]), we have to define a fallback procedure $x^{(k+1)} = T \cdot x^{(k)}$. It is based on a stochastic matrix T that converges to the normalized right eigenvector $x^* \in \mathcal{P}$, called Perron-Frobenius eigenvector, corresponding to its spectral radius $\rho(T) = 1$. The latter is a simple real eigenvalue of T (cf. [7, Chap. 2]). x^* is related to the stationary distribution π of the Markov chain by some transformation (see (5.8) - cf. [95, p. 126]).
Normally, neither J nor J_ω is stochastic. In order to construct a stochastic iteration matrix, we proceed to a nonnegative matrix T, called dual iteration matrix, by a similarity transformation: $T = M \cdot J \cdot M^{-1} = N \cdot M^{-1}$ or $T_\omega = M \cdot J_\omega \cdot M^{-1} = (1-\omega)I + \omega T$, $0 < \omega < 1$. Then T_ω is column stochastic and $\|T_\omega\|_1 = \rho(T_\omega) = 1$ holds for $0 < \omega \leq 1$. For $\omega \in (0, 1)$ T_ω determines a convergent scheme and $\rho(T_\omega) = 1$ is a simple eigenvalue. Subsequently, the subscript ω will be omitted. Furthermore, let the stochastic iteration matrix T and its Perron-Frobenius vector $x^{*t} = (x_1^{*t}, \ldots, x_m^{*t}) \in \mathcal{P}$ be arranged according to the state space partition Γ and the ordering.
Following the approach of Chatelin and Miranker (cf. [23], [61], [22]), we define an aggregation

matrix $R \in \mathbb{R}^{m \times n}$ by

$$R_{ij} = \begin{cases} 1 & \text{if } j \in J_i \\ 0 & \text{otherwise} \end{cases} \qquad 1 \le i \le m, 1 \le j \le n. \tag{5.5}$$

For a fixed probability vector $x = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \in \mathcal{P}$ the prolongation matrix $P_{(x)} \in \mathbb{R}^{n \times m}$ is given by

$$P_{(x)_{ij}} = \begin{cases} (y_j)_i & \text{if } i \in J_j \\ 0 & \text{otherwise} \end{cases} \qquad 1 \le i \le n, 1 \le j \le m, \tag{5.6}$$

where the vector $y = y_{(x)} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$ is defined for $j \in \{1, \ldots, m\}$ as follows:

$$\mathbb{R}^{n_j} \ni y_{(x)_j} = \begin{cases} x_j / \alpha_{(x)_j} & \text{if } x_j > 0 \\ 1/n_j \cdot e & \text{if } x_j = 0 \end{cases} \qquad \text{with} \quad \alpha_{(x)_j} = e^t \cdot x_j$$

$y_{(x)_j}$ comprises the conditional probabilities of the states in the aggregate J_j and y is called the vector of the intra-aggregate probabilities. Based on these matrices

$$R = \begin{pmatrix} e^t & 0 & \ldots & 0 \\ 0 & e^t & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & e^t \end{pmatrix} > 0, \quad P_{(x)} = \begin{pmatrix} y_1 & 0 & \ldots & 0 \\ 0 & y_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & y_m \end{pmatrix} > 0$$

we define a stochastic aggregated iteration matrix by

$$B = B_{(x)} = R \cdot T \cdot P_{(x)} \in \mathbb{R}^{m \times m}. \tag{5.7}$$

In a way, B describes the behaviour of a new system approximating the original one. It results from the aggregation of the original state space S into a smaller set $\{1, \ldots, m\}$ of macrostates determined by the partition $\Gamma = \{J_1, \ldots, J_m\}$. Hence, there exists a probability vector $\alpha^* \equiv \alpha^*_{(x)} = (\alpha_{(x)^*_1}, \ldots, \alpha_{(x)^*_m})^t > 0$ in $\mathbb{R}^m$ satisfying $B_{(x)} \cdot \alpha^*_{(x)} = \alpha^*_{(x)}$ and $e^t \cdot \alpha^*_{(x)} = 1$. α^* is called the vector of inter-aggregate probabilities. This steady-state information about the aggregated system is extended to the whole state space by the generation of the disaggregated probability vector $\tilde{x}_{(x)} \in \mathbb{R}^n$ defined by $\tilde{x}_{(x)} = P_{(x)} \cdot \alpha^*_{(x)} \ge 0$. Hence, for each component of the steady-state vector $\pi = (\pi_j)_{1 \le j \le m}$ we obtain the approximation $(\tilde{x}_{(x)})_j = (\alpha^*_{(x)})_j \cdot (y_{(x)})_j$, $1 \le j \le m$.

Now we can construct a globally convergent iterative A/D-algorithm for the related finite Markov chain with the Q-matrix $A = I - T \in \mathbb{R}^{n \times n}$. It is based on a scheme developed by Schweitzer and Kindle (cf. [121, p. 326f]) and reads as follows:

Assumptions: Let $A = M - N$ be a regular splitting of the irreducible Q-matrix $A \in \mathbb{R}^{n\times n}$ with an extrapolated dual iteration matrix $T = I - \omega A M^{-1}$ for some $\omega \in (0,1)$. Choose a partition $\Gamma = \{J_1, \ldots, J_m\}$ of $\{1, \ldots, n\}$ into $m \geq 2$ disjoint sets. Define $r(x) = \|(I-T)\cdot x\|_1$, $x \in \mathbb{R}^n$.

Algorithm:

(1) Initialization:
Select an initial vector $x^{(0)} \gg 0$ with $e^t x^{(0)} = 1$, an integer $\xi \in \mathbb{N}$, and three real numbers $\varepsilon, c_1, c_2 \in (0,1)$.
Set $k = 0$.

(2) A/D step:

For $j = 1$ to m do

$$[y_{(x^{(k)})}]_j = \begin{cases} x_j^{(k)}/e^t \cdot x_j^{(k)} & \text{if } x_j^{(k)} > 0 \\ 1/n_j \cdot e & \text{if } x_j^{(k)} = 0 \end{cases}$$

For $i = 1$ to m do

$$\left(B_{(x^{(k)})}\right)_{ij} = e^t \cdot T_{J_i J_j} \cdot [y_{(x^{(k)})}]_j$$

endfor

endfor

Solve $B_{(x^{(k)})} \cdot \alpha^*_{(x^{(k)})} = \alpha^*_{(x^{(k)})}$

subject to $e^t \cdot \alpha^*_{(x^{(k)})} = 1,\ \alpha^*_{(x^{(k)})} > 0$

and compute $\tilde{x}^{(k)} = P_{(x^{(k)})} \cdot \alpha^*_{(x^{(k)})} = \left([\alpha^*_{(x^{(k)})}]_j \cdot [y_{(x^{(k)})}]_j\right)_{j=1,\ldots,m}$

(3) Iteration step:

Compute $x^{(k+1)} = T^l \cdot \tilde{x}^{(k)}$

(4) Test of the A/D-gain (optional step - enforcing convergence):

If $r(x^{(k+1)}) \leq c_1 \cdot r(x^{(k)})$ then
goto step (5)
else compute $x^{(k+1)} = T^l \cdot x^{(k)}$
with $l = l(x^{(k)}) \in \mathbb{N}$ such that
$r(x^{(k+1)}) \leq c_2 \cdot r(x^{(k)})$
endif

(5) Termination test:

If $\|x^{(k+1)} - x^{(k)}\|_\infty / \|x^{(k)}\|_\infty < \varepsilon$ then
goto step (6)
else $k = k + 1$
goto step (2)
endif

(6) Normalization:

$$\pi = \frac{M^{-1} \cdot x^{(k+1)}}{e^t \cdot M^{-1} \cdot x^{(k+1)}} \tag{5.8}$$

□

As regular splitting $A = M - N$ we can choose an M-splitting such as the popular block Gauss-Seidel splitting $A = (D - L) - U$ (cf. [78]). It can be shown that the IAD-algorithm converges for any initial vector $x^{(0)} \gg 0$ with $e^t x^{(0)} = 1$ to the steady-state probability vector $\pi \in \mathcal{P}$ of the Markov chain (cf. [121, Theorem 4]). Furthermore, $r(x^{(k+1)}) \leq \max(c_1, c_2) \cdot r(x^{(k)})$, $k \geq 0$, holds.
Moreover, the IAD-scheme without the test of the A/D-gain in step (4), that enforces the global convergence of the scheme, converges locally.

Result 4
Let $T \in \mathbb{R}^{n \times n}$ be a stochastic matrix with the simple eigenvalue $\rho(T) = 1$ and the associated Perron-Frobenius vector $x^* \in \mathcal{P}$. Let T have no further eigenvalues on the unit circle.
Let $\Gamma = \{J_1, \ldots, J_m\}$ be a partition of the state space $S = \{1, \ldots, n\}$ into $m \geq 2$ disjoint sets J_i and let $x^{*t} = (x_1^{*t}, \ldots, x_m^{*t})$ be arranged and numbered according to Γ. We assume that the conditions $(x^*)_{J_i} = x_i^* > 0, 1 \leq i \leq m$, are fulfilled.
Under some mild additional rank-perturbation conditions imposed on $(I - P_{(x)} R \cdot T)$ it follows that the iterative A/D-scheme with ξ post-smoothing iterations is locally convergent to eigenvector $x^* \in \mathcal{P}$ of T. This means that the corresponding equivalent nonstationary scheme

$$\begin{aligned} x^{(k+1)} &= J_{(x^{(k)})} \cdot x^{(k)} \\ &= T^\xi \left[I - P_{(x^{(k)})} (I - B_{(x^{(k)})} + \alpha^*_{(x^{(k)})} \cdot e^t)^{-1} R(I - T) \right] \cdot x^{(k)}, \qquad k = 0, 1, \ldots \end{aligned}$$

converges for each initial vector $x^{(0)} \in \mathcal{P}$ near x^*.

The initial error $\epsilon = x^* - x \equiv x^* - x^{(k)}$ is reduced after these ξ smoothing iterations following an A/D-step to

$$\begin{aligned} \hat{\epsilon}^{(\xi)} &= x^* - \hat{x}^{(\xi)}_{(x)} = x^* - J_{(x)} \cdot x \\ &= T^\xi \left[I - P_{(x)} (I - B_{(x)} + \alpha^*_{(x)} \cdot e^t)^{-1} R(I - T) \right] (x^* - x) \\ &= T^\xi \cdot \left[(I - P_{(x)} RT + P_{(x)} \alpha^*_{(x)} e^t)^{-1} \cdot (I - P_{(x)} R) \right] \cdot \epsilon. \end{aligned}$$

Then the term

$$\hat{\beta}^{(\xi)} = \frac{\|\hat{\epsilon}^{(\xi)}\|}{\|\tilde{\epsilon}\|} \leq \|T(I - P_{(s)} \cdot \alpha^*_{(s)} e^t)\|^\xi$$

provides an upper bound on the error reduction gained by ξ smoothing iterations. Hence, $\chi_{(s)} = \|T(I - P_{(s)} \cdot \alpha^*_{(s)} e^t)\|$ determines the corresponding rate of convergence. It can be monitored to control the progress of the error reduction during the IAD process.

To calculate the steady-state vector $\alpha^*_{(s^{(k)})}$ of the stochastic matrix $B_{(s^{(k)})}$, any of the proposed direct or iterative solution methods including the sketched IAD-method can be applied. If an iterative solution technique is used, the resulting multilevel algorithm is locally convergent, too (cf. [138]). A comparison of different solution methods in a two-level scheme has been provided by Stewart and Wu [134].

Originally, the aggregation/disaggregation method has been developed only for finite irreducible aperiodic DTMCs of *nearly completely decomposable* (NCD) structure with respect to the partition Γ. Let $T^t = \tilde{P} = (\tilde{P}_{ij})_{1 \leq i,j \leq m}$ be the irreducible transition probability matrix (t.p.m.) of the given Markov chain whose block structure is derived from the partition Γ, i.e. each block $\tilde{P}_{ij} = \tilde{P}_{J_i J_j}, 1 \leq i,j \leq m$, comprises all elements with row and column indices $l \in J_i, k \in J_j$, respectively. Then the Markov chain has an NCD structure if the corresponding t.p.m. $\tilde{P}$ has the form $\tilde{P} = \tilde{P}_0 + \varepsilon \tilde{P}_1$ with following properties: $\varepsilon \in (0,1)$ is a small real parameter called maximal degree of coupling that is determined by the maximum row-sum matrix norm $\|\cdot\|_\infty$ of the matrix $\tilde{P}$ without its diagonal blocks (cf. [26, p. 13], [50, p. 15]):

$$\varepsilon = \max_{1 \leq i \leq m} \max_{l \in J_i} \Big(\sum_{j=1, j \neq i}^{m} \sum_{k \in J_j} (\tilde{P}_{ij})_{lk} \Big) = \|\tilde{P} - \mathrm{Diag}(\tilde{P}_{11}, \ldots, \tilde{P}_{mm})\|_\infty \tag{5.9}$$

$\tilde{P}_0 = \mathrm{Diag}((\tilde{P}_0)_{11}, \ldots, (\tilde{P}_0)_{mm})$ is a t.p.m. with irreducible aperiodic stochastic matrices $(\tilde{P}_0)_{ii}, 1 \leq i \leq m$, as diagonal blocks and $\tilde{P}_1 = ((\tilde{P}_1)_{ij})_{1 \leq i,j \leq m}$ is a block structured matrix with zero row sums, i.e. $\tilde{P}_1 e = 0$, satisfying $(\tilde{P}_1)_{ij} \geq 0, i \neq j$, $(\tilde{P}_1)_{ii} \leq 0$, $\|\mathrm{Diag}((\tilde{P}_1)_{11}, \ldots, (\tilde{P}_1)_{mm})\|_\infty = 1$, and $|(\tilde{P}_1)_{ij}| \leq 1$ for all $i,j \in \{1, \ldots, m\}$ (cf. [26], [134], [121], [119]).

To determine an NCD partition Γ, several exact and approximate methods have been proposed. For example, a threshold $\delta > 0$ can be set up and all elements of $\tilde{P}$ less than δ are deleted. Then the partition is determined by the strongly connected components of the directed matrix graph $\Gamma(\tilde{P}_\delta)$ of the resulting modified matrix $\tilde{P}_\delta$. It has an NCD structure, if it satisfies (5.9) and the other criteria can be fulfilled by appropriate construction (cf. [134], [14]). Buchholz [13, p. 106] has introduced related criteria to classify state space partitions and developed corresponding construction algorithms. They are derived from well-known hierarchical clustering techniques. Details can be found in [13, Chap. 4].

The investigations of Courtois [26], Vantilborgh [141], Koury et al. [74], Cao and Stewart [18] and Stewart and Wu [134] show that IAD-procedures with $\xi = 1$ smoothing iteration per step provide an efficient approach for computing the steady-state vector of finite Markov chains with NCD structure. Under some additional regularity conditions on $\tilde{P}$, the error of the approximate solution is reduced per IAD-step by a factor of the order $O(\varepsilon)$ of the coupling

degree (cf. [18]).
Recently, it was observed that the IAD-method is equivalent to the V-cycle of a two-level algebraic multigrid method (AMG) that is applied to the irreducible Q-matrix $A = I - \tilde{P}^t \in \mathbb{R}^{n \times n}$ (cf. [77], [54], [30]). In the case of a CTMC with an irreducible generator matrix Q, the same method can be applied after proceeding to its embedded jump chain with the t.p.m. $\tilde{P} = I + (-\mathrm{Diag}(Q_{11}, \ldots, Q_{nn}))^{-1} Q$ (cf. [78]). If the corresponding stochastic matrix $\tilde{P}$ has further eigenvalues on the unit circle apart from $\rho(\tilde{P}) = 1$, the extrapolated variant $\tilde{P}_\omega = (1 - \omega) I + \omega \tilde{P}$, $0 < \omega < 1$, must be used.
In conclusion, we see that the IAD-method can be employed as a building block of an analysis program not only for nearly completely decomposable Markov chains, but for all irreducible Markov chains (cf. [26], [61], [121], [119], [120], [76]). For this reason, it is used as basic iterative solver in MACOM.

5.4 Further reading

As in the case of regular matrices, all iterative methods for *singular systems* are variants of the *power method* (cf. [50, Sec. 7.3., p. 209]). Considering the splitting $A = M - N$ of a regular matrix A, its convergence is guaranteed if the spectral radius of $T = M^{-1}N$ satisfies $\rho(T) < 1$ (cf. [142, 3.2 , p. 61ff]). In the case of singular systems, however, other conditions are necessary to guarantee the convergence of the scheme (5.1) because the spectral radius $\rho(T)$ is equal to one now (cf. [7, Ex. 6.4.9, p. 152, Lemma 7.6.9, p. 197], [109, Prop. 1, p. 136]).
Necessary and sufficient conditions guaranteeing the convergence of standard iteration schemes applied to singular M-matrices, like the Jacobi or Gauss-Seidel procedure, have been derived by Rose [109], Schneider [118], Barker and Plemmons [3] and Barker and Yang [4] among others.
Considering the block or point Gauss-Seidel splitting $A = (D - L) - U$ of an irreducible M-matrix, it is convenient to accelerate the Gauss-Seidel procedure by the standard *over- or underrelaxation technique* (cf. [66], [142], [3]) or by a *semi-iterative technique* such as the stationary or nonstationary Chebyshev method or Eiermann's stationary fourth-order scheme (cf. [142, §5], [33], [102], [32, Lemma 8.4, p. 28], [4], [55], [56]). Regarding these procedures the determination of 'optimal' relaxation parameters is the main difficulty. As there is no a priori information about the location of the eigenvalues of the iteration matrices, heuristic procedures estimating approximately optimal parameters seem to be the only practicable approach (cf. [57, §9.5, p. 223ff], [133], [124], [55], [56]).
The convergence of the Jacobi underrelaxation scheme has been established by Varga et al. [16, Theorem 3.4, p. 191]. They generalized the classical theorem of Stein-Rosenberg [142, Th. 3.15, Th. 3.16, p. 90ff] to singular M-matrices A with positive diagonal elements. Furthermore, they have shown that the Gauss-Seidel underrelaxation scheme with the iteration matrix $T_\omega = (D - \omega L)^{-1}((1 - \omega)D + \omega U)$, $0 < \omega < 1$, is also convergent if it is applied to an irreducible singular M-matrix. A corresponding result was also proved by Barker and Plemmons [3, Cor. 3, p. 395]. Therefore, both methods can be employed for the solution of (3.5).

Applying a continuity argument, we conclude that the accelerated iteration procedure derived from an R-regular splitting converges for $\omega > 1$ if the corresponding scheme converges for $\omega = 1$ (cf. [142, p. 109]). But suitable formulas for the determination of the optimal relaxation parameter $\omega \in (1,2)$ only exist for special cases (cf. [142, §4.3], [57, §9], [55], [56]).
Semi-iterative variants of the Gauss-Seidel procedure for singular M-matrices have been studied by Kaufman [66], Eiermann, Varga, and Niethammer [32] and Eiermann, Marek, and Niethammer [33] among others. For details the reader is referred to these articles and the references therein. We recommend to consult the survey of Eiermann et al. [33].
Aggregation/disaggregation (A/D) methods for NCD models have been studied by Courtois [26], Vantilborgh [141], Koury et al. [74], and Cao et al. [18] among others. Regarding the application of *iterative A/D-procedures* to arbitrary finite Markov chains in general, important results have been derived by Haviv [61], Chatelin [22], Sumita et al. [136], Schweitzer [119], Schweitzer and Kindle [121]. Particularly Chatelin [22], Schweitzer [120] and Haviv [61] provide excellent surveys of this topic. An analysis of the error reduction of an A/D-step and the gain of an iteration step following the A/D-step was given by Krieger [76]. Moreover, the equivalence between IAD-schemes and the V-cycle of an algebraic multigrid method can be shown (cf. [138], [54], [30], [77]). The interested reader is referred to these contributions.
Last but not least, it is worthwhile mentioning that the power method, the point Jacobi procedure, the block Gauss-Seidel procedure and its underrelaxation variant have a stochastic interpretation (cf. [95, p. 122], [76, §4.3], [77]).

5.5 Applicability of iterative methods

We recommend to apply variants of the IAD-algorithm as basic iteration procedures. They should be derived from an appropriate regular splitting $A = M - N$ taking into account the structure such that the linear system $Mx^{(k+1)} = Nx^{(k)}$ can be solved efficiently. Block or point Gauss-Seidel splittings or an R-regular splitting are recommended. If a relaxation technique is used, a careful choice of the parameter $\omega \in (0,2)$ is required. It has to be supported by a heuristic procedure (cf. [57, §9.5, p. 223ff]). During the iteration process, the insertion of a few expensive, but effective A/D steps is useful, particularly in the initial phase. Both the splitting and the partition of the state space must be adapted to the natural structure of the Q-matrix A.
In general, an iteration procedure of this kind can be applied to all banded, sparse or large matrices and to all models with NCD-structure. Regarding the analysis of models with huge state spaces, the main difficulties result from the memory requirements and slow convergence. In this respect, it is a major advantage of an iterative approach that it is not necessary to generate and store the entire Q-matrix of a model a priori. The latter can either be generated dynamically and by parts during the iteration or routines calculating only matrix-vector products Ax can be provided. If the tensor structures of submatrices of matrix A can be used, efficient implementations are possible. By this means, Markovian models with more than 500000 states can be analyzed (cf. [14, p. 200f], [13]).

6 Examples

In this section we first describe two computer models illustrating the benefits and difficulties of the recommended GTH method. In the following subsections we apply the presented iterative procedures to some Markovian models arising from the study of advanced routing strategies in telecommunication networks.

6.1 Models of computer systems in telecommunication networks

The first computer model, provided by [107], is a *retrial queue*. It describes a telephone exchange (station $S2$) with impatient customers. Each customer waiting for a ringing tone may become impatient if his post-dialing delay is too long. Then he reattempts his call request after a random period (modeled by station $S1$) or he gives up with a fixed probability $1-h$ (see Fig. 3). The number of customers at each station Si is denoted by $X_i, i \in \{1,2\}$. The arrival stream of new customers is modeled by a Poisson process with intensity A.
The exchange is described by a special $G/M_n/1/K2$ *delay-loss system* $S2$ where requests are lost if the system is occupied. If $X_2 = j$ requests are pending in $S2$, a successful customer gets the ringing tone and departs from the queueing network at $S2$ in an interval of infinitesimally small length dt with probability μdt. He does not get the tone and decides to reattempt with probability $j\tau dt$. This means, that one of the $X_2 = j$ pending requests is processed with service rate μ and each corresponding customer may become impatient with rate τ. The fictitious orbit for impatient customers where they are waiting before initiating a new call request is a $G/M/K1/K1$ loss system $S1$. The time until the retrial of a customer is exponentially distributed with mean $1/\lambda$.
The parameters are set to $A = 0.6$, $\mu = 1.0$, $\tau = 0.05$, $h = 0.85$ and $\lambda = 5.0$. Capacities $K1$ and $K2$ can be varied, thereby yielding different sizes of the state space. They should be large enough such that congestion effects are negligible.
The second model is a *modified central server system* that arises from modeling the exchange of information between an I/O subsystem and a computer (cf. [139, p. 382]). It is a closed queueing network with a fixed population of size n consisting of 4 stations $S1, \ldots, S4$ of type $\cdot/GI/1/\infty$ (see Fig. 4). The service times at $S1$ and $S4$ are exponentially distributed with rates μ_1 and μ_4, respectively. The service times at $S3$ are governed by an Erlang distribution with k phases and those of $S2$ by a Coxian distribution with two phases with the same rates. The coefficient of variation of this Coxian service-time distribution is an adjustable parameter of the model and set to 1.5. The mean service times at $S2, S3$ and $S4$ are set to 0.5. The size of the state space is determined by the number n of customers in the network and the number k of Erlangian phases at $S3$. The NCD property of the model is controlled by the routing probability p. To reduce the influence of p on the distribution of the population at $S1$, the service rate μ_1 is set to $1-p$.
The fill-in resulting from Gaussian elimination strongly depends on the enumeration of states, hence, the bandwidth structure of the resulting Q-matrix. Using four distinct enumeration schemes in the central server model with 2109 states, for instance, the actual fill-in varies between 349360 and 446871. In most cases, the natural enumeration induced by the order of

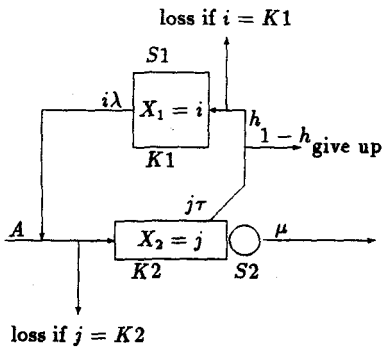

Figure 3: Queueing model with impatient customers

generation produces rather low fill-in since most models possess some kind of locality. It is implicitly exploited during the automatic construction process.
Regarding the exchange model with impatient customers GTH factorization yields a solution vector with negative values due to the extremely small probabilities in the range of 10^{-100} for a large buffer $K2 = 220$ and $K1 = 10$ retrial places. Considering the central server model with a low degree of coupling between $S1$ and the rest of the network for $p = 0.99999999$ and a state space with 2109 states, the result of LU-factorization without the GTH modification of the diagonal elements becomes inaccurate due to the NCD property of the network. The errors depend on the used enumeration. If we apply, however, the GTH algorithm, no difficulties arise. It is more stable, even for some NCD systems, if it is applied carefully (cf. [91]).

6.2 A model of alternative routing

Considerable attention has been devoted to the study of advanced routing schemes in circuit-switched digital networks based on efficient modern signalling systems. In this context, the analysis of a circuit-switched network with alternative routing is a classical issue of teletraffic theory (cf. [137]). A single, fully available trunk group that carries both direct traffic and multiple overflow traffic streams is a well-known model (cf. [10]). It describes a part of the network where several origin-destination pairs $O - D_1, \ldots O - D_l$ share the same overflow trunk group $O - T$. It carries additionally direct traffic (cf. Fig. 5). The direct traffic streams corresponding to $O - D_i$ overflow to the common trunk group $O - T$ with n trunks if the corresponding direct routes are blocked.
Considering this model, we want to demonstrate the use of an iterative method derived from an R-regular splitting. Let $X(t)$ denote the number of busy trunks at time $t \geq 0$ in this fully

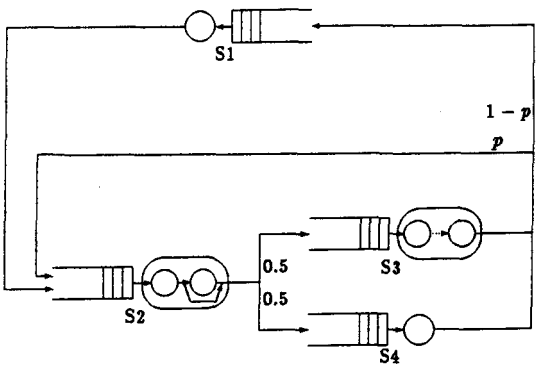

Figure 4: Modified central server model

available common overflow trunk group. The offered load consists of a Poisson stream (stream 0) with intensity $\lambda_0 > 0$ and two overflow streams (1 and 2). These streams are assumed to be independent of each other. The call holding times are mutually independent, exponentially distributed random variables with a common finite mean $1/\mu$. They are assumed to be independent of the arrival processes.

Following a standard approach in teletraffic theory (cf. [88], [89]), the overflow streams are modeled by two Interrupted Poisson processes (IPPs) resulting from a two-moment approximation. They will be represented by two mutually independent Markov-modulated Poisson processes (MMPPs 1 and 2) with the generator matrices $Q_1 = \begin{pmatrix} -\gamma_1 & \gamma_1 \\ \omega_1 & -\omega_1 \end{pmatrix}$, $Q_2 = \begin{pmatrix} -\gamma_2 & \gamma_2 \\ \omega_2 & -\omega_2 \end{pmatrix}$ and the rate vectors $\hat{\lambda}_1 = \begin{pmatrix} \lambda_1 \\ 0 \end{pmatrix}$, $\hat{\lambda}_2 = \begin{pmatrix} \lambda_2 \\ 0 \end{pmatrix}$ (cf. [88], [89], [81]). Here, $\lambda_i > 0$ is the intensity of the Poisson process associated with the IPP stream i, $1/\gamma_i$ its mean on-time and $1/\omega_i$ its mean off-time, $i \in \{1, 2\}$ (cf. [81, p. 438]). Each Poisson stream can be regarded as an MMPP, too.

We denote the phase of the controlling CTMC in the MMPP representation of the IPP $i \in \{1, 2\}$ at time $t \geq 0$ by $Y_i(t)$. Its associated irreducible generator matrix is Q_i. As the superposition of independent MMPPs is still an MMPP, the Markovian environment resulting from the composition of the arrival streams is given by $Y(t) = (Y_1(t), Y_2(t))$. It possesses the

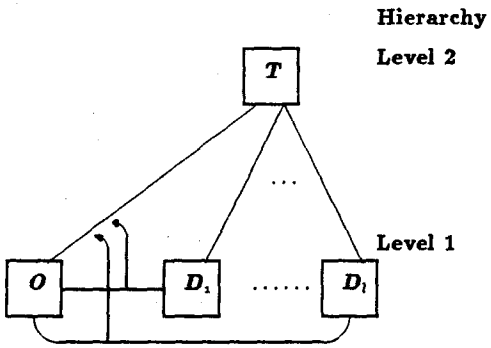

Figure 5: Part of a 2-level circuit-switched network with alternative routing

irreducible generator matrix

$$Q = Q_1 \oplus Q_2 = \begin{pmatrix} -\gamma_1-\gamma_2 & \gamma_2 & \gamma_1 & 0 \\ \omega_2 & -\gamma_1-\omega_2 & 0 & \gamma_1 \\ \omega_1 & 0 & -\omega_1-\gamma_2 & \gamma_2 \\ 0 & \omega_1 & \omega_2 & -\omega_1-\omega_2 \end{pmatrix} \in \mathbb{R}^{m\times m},$$

$m = 4$. Its states $Y(t) = (k_1, k_2) \equiv k$ will be ordered lexicographically and enumerated by integers $k \in \{1,2,3,4\}$. The rate vector of the resulting MMPP is given by $\tilde{\lambda} = (\lambda_0+\lambda_1+\lambda_2, \lambda_0+\lambda_1, \lambda_0+\lambda_2, \lambda_0)^t \gg 0$.
Then the model of alternative routing may be regarded as $M/M/n/n$ loss system in a Markovian environment. It can be described by an irreducible CTMC $\{Z(t) = (X(t), Y(t)), t \geq 0\}$ with a finite state space $S = \{(i,k) \in \mathbb{N}_0^2 \mid 1 \leq k \leq 4; 0 \leq i \leq n\}$.
We assume a lexicographical ordering of states. Then the Q-matrix $A = -\tilde{Q}^t \in \mathbb{R}^{L\times L}$, $L = m\cdot(n+1)$, associated with the irreducible generator matrix $\tilde{Q}$ of $Z(t)$ has the following block tridiagonal structure:

$$A = \begin{pmatrix} -Q^t+\Lambda & -1\mu I_4 & 0 & \dots & \dots & 0 \\ -\Lambda & -Q^t+1\mu I_4+\Lambda & -2\mu I_4 & \ddots & \dots & \vdots \\ 0 & -\Lambda & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & \ddots & \ddots & \ddots & -n\mu I_4 \\ 0 & \dots & \dots & 0 & -\Lambda & -Q^t+n\mu I_4 \end{pmatrix} \tag{6.1}$$

Here, $\Lambda_0 = \lambda_0 I_4$, $\Lambda_1 = \mathrm{Diag}(\lambda_1, \lambda_1, 0, 0)$, $\Lambda_2 = \mathrm{Diag}(\lambda_2, 0, \lambda_2, 0)$ are the arrival rate matrices of the Poisson stream and the two IPPs. $\Lambda = \Lambda_0 + \Lambda_1 + \Lambda_2$ is the arrival rate matrix of the MMPP resulting from their superposition and I_m denotes the identity matrix of order m.
Taking advantage of the block structure of A, the steady-state vector π of $Z(t)$ can be computed either by a block iteration scheme derived from an R-regular splitting of A or by an accelerated point iteration scheme such as JOR or SOR (cf. [109], [89]). Such an R-regular splitting technique has been suggested by Meier-Hellstern [88, §3], [89]. The proposed procedure is derived from the following block splitting $A = M - N$:

$$M = \begin{pmatrix} -Q^t + \Lambda + n\mu I_4 & 0 & \dots & \dots & 0 \\ -\Lambda & -Q^t + \Lambda + n\mu I_4 & \ddots & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -\Lambda & -Q^t + \Lambda + n\mu I_4 \end{pmatrix} \tag{6.2}$$
$$= D - L$$

$$N = \begin{pmatrix} n\mu I_4 & 1\mu I_4 & 0 & \dots & 0 \\ 0 & (n-1)\mu I_4 & 2\mu I_4 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & 0 & 1\mu I_4 & n\mu I_4 \\ 0 & \dots & \dots & 0 & \Lambda \end{pmatrix} \tag{6.3}$$
$$= L(N) + U(N) + D(N) \quad \geq 0$$

Obviously, $L(N) = 0$, $D = \mathrm{Diag}(D_{00}, \dots, D_{nn}) = I_{n+1} \otimes (-Q^t + \Lambda + n\mu I_4)$ and $D(N) = \mathrm{Diag}(n\mu I_4, (n-1)\mu I_4, \dots, 1\mu I_4, \Lambda) \geq 0$,

$$L = \begin{pmatrix} 0 & 0 & \dots & \dots & 0 \\ 1 & 0 & \ddots & \dots & \vdots \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix} \otimes \Lambda \geq 0, \quad U(N) = \begin{pmatrix} 0 & 1\mu & 0 & \dots & 0 \\ \vdots & \ddots & 2\mu & \ddots & \vdots \\ \vdots & \dots & \ddots & \ddots & 0 \\ \vdots & \dots & \dots & \ddots & n\mu \\ 0 & \dots & \dots & \dots & 0 \end{pmatrix} \otimes I_4 \geq 0$$

hold. $\otimes$ is the Kronecker product of two matrices defined by $A \otimes B = (A_{ij} \cdot B)$ (cf. [101, p. 53]). Thus, $D_{ii} = -Q^t + \Lambda + n\mu I_4$ is an irreducible column diagonally dominant regular M-matrix. Hence, $D_{ii}^{-1} \gg 0$ holds (cf. [7, Theorem 6.2.7, p. 141]). The matrix $A_0 = D - L - U(N)$ has the same zero structure as A. Thus, it is an irreducible block tridiagonal M-matrix. The corresponding block matrix graph $\Gamma(A_0)$ possesses a monotone descreasing cycle, for instance, $[2, 1, 2]$ since $(A_0)_{21} = -\Lambda \neq 0$ and $(A_0)_{12} = -\mu I_4 \neq 0$ hold.
Therefore, the proposed block splitting $A = M - N$ defined by (6.2), (6.3) is an R-regular splitting. Moreover, it is an M-splitting since the diagonal blocks of M are regular M-matrices.

Hence, we conclude from Rose's convergence Theorem [109, Theorem 1, p. 138] that the resulting block iteration procedure $x^{(k+1)} = M^{-1}Nx^{(k)}$, $k = 0, 1, \ldots$ converges for each initial vector $x^{(0)} \gg 0$ to the steady-state vector π of $Z(t)$ if the iteration vector $x^{(k+1)}$ is normalized after convergence.
Of course, this algorithm can be accelerated by inserting several aggregation / disaggregation steps during the iteration according to the IAD-scheme (see sections 5.3). In comparison with the block Gauss-Seidel scheme, the proposed procedure has the advantage that all diagonal blocks of the matrix M are identical. Therefore it is necessary to decompose only one small matrix D_{ii} and to store its inverse during the iteration process. The resulting algorithm is well suited for an implementation on a vector processor. Experimental results of the scheme have been provided by Meier-Hellstern [88].

6.3 A teletraffic model of mutual overflow routing

The next example concerns the study of an adaptive routing strategy in circuit-switched networks. We consider a network consisting of a local exchange (EX_0) and two exchanges of the long-distance network (EX_1, EX_2). They are connected to each other by two both-way trunk groups. The traffic outgoing from the local exchange is split into two portions and each portion is offered to an outgoing group. These partial traffic streams are routed according to an adaptive routing scheme called mutual overflow routing (cf. [86]). Additionally each trunk group carries external traffic (cf. Fig. 6).
The network can be described by a loss system composed of two fully available trunk groups called systems 1 and 2 with N_1 and N_2 lines. Two originating traffic streams (streams 2 and 3) representing the portions of the outgoing traffic and two incoming external traffic streams (streams 1 and 4) are offered to the loss system. These arrival processes are modeled by mutually independent Poisson processes with positive rates λ_2, λ_3 and λ_1, λ_4.
The external traffic streams 1 and 4 offered to systems 1 and 2 follow a random hunting scheme for free lines. Their calls are lost without further impact on the system if the corresponding trunk group is busy upon arrival. The outgoing streams 2 and 3 follow a mutual overflow routing scheme. This means that, upon arrival at system 1, a call of flow 2, for instance, is searching for a free line. If possible, a free trunk is selected in a random manner and occupied. If system 1 is busy and there are free lines in system 2, the incoming call of flow 2 will immediately overflow to system 2 upon arrival and occupy a line selected at random. If both systems are busy, the call will be blocked and lost without further impact on the system (lost calls cleared).
Call holding times are mutually independent, exponentially distributed random variables with a common finite mean $1/\mu$. They are also assumed to be independent of the arrival processes.

The occupation process of both trunk groups in this loss system can be modeled by an irreducible CTMC $\{X(t) = (X_1(t), X_2(t)), t \geq 0\}$ with finite state space $S = \{(i,j) \in \mathbb{N}_0^2 \mid 0 \leq i \leq N_1; 0 \leq j \leq N_2\}$. Here the state variables $X_i(t)$ denote the number of busy trunks in group $i \in \{1,2\}$ at time t. Regarding the size $N = (N_1 + 1) \cdot (N_2 + 1)$ of the two-dimensional state space, some examples are shown in Table 3. It includes the number of nonzero elements

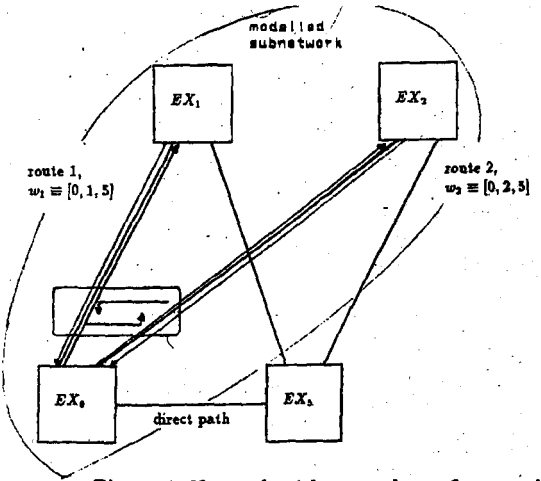

Figure 6: Network with mutual overflow routing

N_1	N_2	N	$5N$
30	60	1891	9 271
60	60	3721	18 605
60	90	5551	27 755
120	120	14641	73 205
210	210	44521	222 605

Table 3: Size of the state space of the mutual overflow model

of the generator matrix, which is approximately $5N$ due to 5 nonzero diagonals.
We assume a lexicographical ordering of states and denote the corresponding Q-matrix by $A = -Q^t \in \mathbb{R}^{N\times N}$, $N = (N_1+1)\cdot(N_2+1)$. It is irreducible and has a block tridiagonal structure

$$A = \begin{pmatrix} B_0 & C_0 & 0 & \dots & \dots & \dots & 0 \\ D_1 & B_1 & C_1 & \ddots & \dots & \dots & \vdots \\ 0 & D_2 & \ddots & \ddots & \ddots & \dots & \vdots \\ \vdots & \ddots & \ddots & B_k & \ddots & \ddots & \vdots \\ \vdots & \dots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & \dots & \ddots & \ddots & \ddots & C_{N_1-1} \\ 0 & \dots & \dots & \dots & 0 & D_{N_1} & B_{N_1} \end{pmatrix} \in \mathbb{R}^{N\times N} \tag{6.4}$$

with irreducible tridiagonal M-matrices

$$B_0 = \begin{pmatrix} a & -1\mu & 0 & \dots & \dots & \dots & 0 \\ -a_2 & a+1\mu & -2\mu & \ddots & \dots & \dots & \vdots \\ 0 & -a_2 & \ddots & \ddots & \ddots & \dots & \vdots \\ \vdots & \ddots & \ddots & a+k\mu & \ddots & \ddots & \vdots \\ \vdots & \dots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & \dots & \ddots & \ddots & \ddots & -N_2\mu \\ 0 & \dots & \dots & \dots & 0 & -a_2 & a_3+N_2\mu \end{pmatrix},$$

$$B_{N_1} = \begin{pmatrix} a_4 & -1\mu & 0 & \dots & \dots & \dots & 0 \\ -a_4 & a_4+1\mu & -2\mu. & \ddots & \dots & \dots & \vdots \\ 0 & -a_4 & \ddots & \ddots & \ddots & \dots & \vdots \\ \vdots & \ddots & \ddots & a_4+k\mu & \ddots & \ddots & \vdots \\ \vdots & \dots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & \dots & \ddots & \ddots & a_4+(N_2-1)\mu & -N_2\mu \\ 0 & \dots & \dots & \dots & 0 & -a_4 & N_2\mu \end{pmatrix} + N_1\mu \cdot I,$$

$B_i = B_0 + i\mu \cdot I$, $i = 0, \dots N_1 - 1$, of order N_2+1 along the diagonal. The off-diagonal matrices $D_i = -\mathrm{Diag}(a_1, \dots, a_1, a_3)$, $i = 1, \dots N_1$, $C_i = -(i+1)\mu \cdot I$, $i = 0, \dots N_1 - 1$, have diagonal structures. Here, we have set $a_1 = \lambda_1 + \lambda_2 > 0$, $a_2 = \lambda_3 + \lambda_4 > 0$, $a_3 = \lambda_1 + \lambda_2 + \lambda_3 \geq a_1$, $a_4 = \lambda_2 + \lambda_3 + \lambda_4 \geq a_2$, $a = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = a_1 + a_2$ and λ_i denotes the intensity of the offered Poisson stream $i \in \{1, \dots, 4\}$. $1/\mu$ is the mean call holding time and I the identity matrix.

Hence, A is a consistently ordered 2-cyclic Q-matrix regarding this block partition (cf. [142, p. 102]). Considering the point partition A is also a consistently ordered 2-cyclic matrix and it has property A (cf. [142, p. 187], [109], [7]). Thus, the point Jacobi matrix $J = D^{-1}(L+U)$ is cyclic of index 2, i.e. the greatest common divisor of the lengths of all proper cycles in the matrix graph of A is 2. According to [3, Prop. 1, p. 392] the Jacobi procedure is not convergent (see also [7, Theorem 2.2.30, p. 35]), whereas the corresponding JOR and SOR procedures with the iteration matrices $J_\omega = (1-\omega)I + \omega J$ and $L_\omega = (D - \omega L)^{-1}((1-\omega)D + \omega U)$ are convergent for each relaxation parameter $\omega \in (0, 1)$ (see [4, Theorem 3.9], [3, Cor. 3, p. 395], [16, Theorem 3.4, p. 191]).

As A has a symmetric zero structure, we can also conclude from Result 2 that the point Gauss-Seidel procedure is convergent (see also [109, Cor. 2, p. 139], [118, Cor. 3.8, p. 417]). Taking into account $M_{21} = (D-L)_{21} = -a_2 < 0$ and $N_{12} = U_{12} = \mu > 0$, this result also follows from Result 1. This point iteration can be accelerated by the standard overrelaxation technique (cf. [66], [133]) or a semi-iterative technique (cf. [142, §5], [33], [102], [32, Lemma 8.4, p. 28], [4], [55], [56]). Regarding these procedures the main difficulty concerns the determination of 'optimal' relaxation parameters ω. As there is no a priori information about the location of the eigenvalues of the iteration matrices, heuristic procedures estimating approximately optimal

Method	Relaxation parameter	Number of A/D steps	Number of iterations	Time for the solution in sec
SOR	1.2	-	180	16.8
	1.3	-	124	11
	1.4	-	144	13
	dynamic	-	100	9.1
SOR-A/D	1.0	10	102	16.2
	1.2	10	116	16.1
	1.3	2	104	13.9
	dynamic	3	140	19.6
JOR-A/D	0.9	51	216	38.3
	dynamic	45	201	33
JOR	0.9	-	> 500	-
GTH algorithm		Fill-in 1.37 %		67
LU-factorization		Fill-in 1.37 %		64.8

Table 4: Comparison of the solution methods based on the loss system with mutual overflow and external traffic with $\lambda_1 = 40$, $\lambda_2 = 30$, $\lambda_3 = 60$, $\lambda_4 = 10$, $\mu = 1$, $N_1 = 30$, $N_2 = 60$. The model has $N = 1891$ states and the generator matrix has 9271 nonzero elements. The required accuracy of the point iteration procedures is $\varepsilon = 10^{-6}$. The data are computed by the package MACOM on a SUN 3/470 with floating point accelerator.

parameters are the only practicable approach (cf. [57, §9.5, p. 223ff], [124], [55], [56]). But in the case of a consistently ordered 2-cyclic matrix the well-known relation $(\lambda+\omega-1)^2 = \lambda\omega^2\mu^2$ between the eigenvalues μ and λ of the (block) Jacobi and SOR iteration matrix can be used (see [142, Theorem 4.3, p. 106] and [142, §4.3, p. 109]).

An alternative is provided by block iteration schemes such as the block Gauss-Seidel procedure or its modified versions arising from Rose's R-regular splitting (see [89]). According to Result 3, the block Gauss-Seidel scheme derived from the given block tridiagonal structure (6.4) of A is convergent since the diagonal blocks are irreducible regular M-matrices.

All methods may be combined with A/D-steps if the IAD-procedure is used (see section 5.3 - cf. [98], [74], [136]). The transformation of each block of A into a single scalar is a very natural way of aggregation. This procedure corresponds to the aggregation of each macrostate $\{(i,j) \in \mathbb{N}_0^2 \mid 0 \le j \le N_2\}$, $0 \le i \le N_1$, into a single state $[i]$. In this way, a simple birth-death process is generated. Its aggregated steady-state distribution α can be calculated in an efficient way by an explicit analytic representation. A comparison of different solution techniques yields the results shown in Table 4.

Let us consider both alternative paths in the network depicted in Figure 6 and an adaptively routed traffic stream with source EX_0 and destination EX_3 again. If this part of the network with four additional traffic streams carried on the different links of the alternative paths w_1, w_2 is modeled by a Markovian queueing network, the size of the state space grows very rapidly. If, for instance, $N_1 = 5$, $N_2 = 4$, $N_3 = 6$, $N_4 = 5$ are the sizes of the four trunk groups,

N	$2(N+1)^5$
4	15 552
9	200 000
29	48 600 000
89	11 809 800 000

Table 5: Size of the state space of the 4-node network with five different traffic classes

the state space already consists of 7840 up to 15680 states depending on the employed routing strategy. Due to the five different classes of streams (- four traffic streams on direct routes and one adaptively routed stream -), the size of the state space is approximately $2(N+1)^5$ if all trunk groups comprise N trunks. Some examples of its complexity are shown in Table 5. If only three streams are carried on three links of both alternative paths w_1, w_2, but more complicated adaptive routing algorithms are used, then MACOM generates even for $N_1 = 12, N_2 = 10, N_3 = 13, N_4 = 12$ Markovian models with 18486 up to 79376 states. But in all casses only 0.12 % of the elements of the generator matrix are nonzero. This extremely small percentage facilitates the use of iterative solution methods.
Considering real networks, the analysis is, of course, limited to small parts containing trunk groups with small sizes. Nevertheless, the effects of different adaptive routing strategies can be studied even in such small networks. Moreover, the quality of approximation methods can be evaluated by a comparison with exact results arising from such small networks (cf. [79]).

6.4 Application of the IAD-scheme

Iterative methods require much more monitoring and control regarding the progress of the solution process compared with direct methods. For instance, the stopping criterion for convergence must be evaluated, decisions on control parameters such as the relaxation parameter in SOR have to be taken and, after termination, the accuracy of the approximate solution vector should be estimated. Furthermore, it is very important that the selected iteration procedures are carefully adapted to the algebraic structure of an investigated Markovian model to guarantee fast convergence. Compared with iterative methods with few dynamic control parameters such as preconditioned CG procedures (cf. [107]), iteration schemes derived from regular splittings of the singular M-matrix $-Q^t$, e.g., point and block SOR with optional A/D steps, require more monitoring and control activities. For unexperienced users or simple black box solvers, it seems to be a disadvantage of these procedures that the control parameters have to be chosen very carefully. On the other hand, we are convinced that these features and the flexibility of splitting methods offer the chance to improve the convergence of the corresponding iterative algorithms considerably if the control parameters are carefully adapted to the underlying algebraic and stochastic structure of a given Markov chain.
As mentioned previously, slow convergence is the main difficulty of iteration procedures. In most cases, it has the reason that the selected method and its control parameters, e.g., the relaxation factor of SOR, are not very well adapted to the structure of the model. This beha-

#iteration steps k	$d^{(k)}$	$r^{(k)}$	$e^{(k)}$
20	7.2300e-4	0.96818	2.2002e-2
100	1.2736e-6	0.92055	1.4758e-5
500	1.6296e-6	0.99990	1.6176e-2
1000	1.5505e-6	0.99990	1.5758e-2
5000	1.0828e-6	0.99992	1.3238e-2
10000	7.4970e-7	0.99993	1.1328e-2

Table 6: Behavior of the Gauss-Seidel procedure in the case of the central server model

Algorithm	partition	#iter. steps k	#A/D steps	$d^{(k)}$	$r^{(k)}$	$e^{(k)}$	CPU time in sec.
Gauss-Seidel	none	4000	0	1.28e-06	0.99991	1.50e-02	382.9
	EQ_1	181	8	5.14e-13	0.91929	5.85e-12	19.6
	EQ_2	4023	63	6.13e-06	0.99988	4.93e-02	385.5
	EQ_3	4086	84	1.07e-06	0.99992	1.28e-02	389.3
block Gauss-Seidel	EQ_1	80	1	3.76e-13	0.85984	2.31e-12	224.3
	EQ_2	60	1	7.96e-13	0.64910	1.47e-12	1069.8
	EQ_3	4040	80	2.92e-06	0.99998	1.35e-01	579.5

Table 7: Solution of the central server model with different partitions on a SUN 4/110 by MACOM

vior may be caused by an undetected NCD property of the investigated CTMC. This problem can be illustrated by the modified central server model with a weak coupling of station $S1$ if $p = 0.9999$ is used (see Table 6). After 10 000 iteration steps, the throughput at station $S1$ still has a large error of about 10% which is indicated by the large magnitude of $e^{(k)}$.
MACOM tries to overcome this difficulty by a simple strategy. The idea is to select a point or block SOR procedure with a suitable relaxation parameter and to insert several A/D steps during the iteration. The required partition of the state space, however, must be empirically adapted to the NCD structure of the model which is unknown in most cases.
The effect of three different partitions EQ_1, EQ_2, EQ_3 on the rate of convergence is shown by the central server model with $n = 17$ customers, $k = 3$ service phases at $S3$ and $p = 0.9999$. In this model the state space must be partitioned by a relation EQ_1 defined by the equality of the population at $S1$. EQ_2, that combines three macrostates of EQ_1 to a new macrostate, is coarser than EQ_1. The macrostates of EQ_3 are defined by the equality of the population at $S4$. Table 7 shows the influence of the chosen partition on the Gauss-Seidel and block Gauss-Seidel schemes. The block structure is derived from the partition used for A/D steps. The results in Table 7 show that partition EQ_1 improves the rate of convergence whereas the others do not, with the exception of EQ_2 in the block Gauss-Seidel scheme. The latter scheme

Algorithm	partition	relaxation parameter	#iter. steps k	A/D steps	$d^{(k)}$	$r^{(k)}$	$e^{(k)}$	time in sec.
GTH alg.	$i+j$							275.3
SOR		1.0	1000		3.03e-12	0.945	5.26e-11	211.8
		1.3	1000		3.72e-13	0.897	3.25e-12	223.4
		automatic	920		4.64e-14	0.889	3.72e-13	200.9
SOR with A/D steps	$i+j$	1.0	100	4	1.13e-13	0.686	2.47e-13	54.6
		1.3	132	4	3.44e-15	0.663	6.79e-15	82.3
		automatic	100	4	4.17e-14	0.688	9.19e-14	55.6
block SOR	$i+j$	1.0	850		8.49e-14	0.901	7.71e-13	380.5
		1.3	810		5.88e-14	0.811	2.53e-13	381.8
	$(i+j)/2$	1.0	430		7.19e-14	0.814	3.14e-13	236.8
		1.3	400		7.85e-14	0.633	1.36e-13	221.2
IAD with block SOR	$i+j$	1.0	68	2	3.21e-13	0.511	3.36e-13	63.4
		1.3	116	1	1.38e-13	0.877	9.82e-13	85.3

Table 8: Comparison of the solution algorithms of MACOM on a SUN 4/110 based on the model with impatient customers

behaves well since it captures the NCD property in the diagonal blocks of the partitioned generator matrix and in the aggregated system. Obviously, the block Gauss-Seidel procedure can overcome slow convergence. Regarding, however, the CPU time used for the block scheme with large macrostates, there is always a trade-off between the rate of convergence and the computational effort. The latter arises from the solution of the large linear systems defined by the diagonal matrices of the block scheme. As there is no general rule for the selection of a method, experiments performed on variants of a model with small state spaces are required to get insight into the behavior of the CTMC before starting a series of experiments.
The insertion of A/D steps during the iteration is a general strategy of MACOM to improve the rate of convergence, even if a model has no NCD structure. In the start-up phase of the iteration this technique can improve the search of a subspace containing the solution vector. This effect is shown by the first model with parameters $K1 = 30$ and $K2 = 550$. A comparison of different solution algorithms is provided by Table 8. The size of the state space is 17081 and all CPU times are obtained on a Sun 4/110. Two partitions are used for the block scheme and the A/D variants. In the first partition $P1$, states belong to the same macrostate if the corresponding total numbers of customers in the system, i.e. $X_1 + X_2 = i + j$, are equal. The second partition $P2$ is coarser. Here states belong to the same macrostate if $X_1 + X_2 = (i + j)/2$ holds where '/' denotes an integer division. We see that, although the model has no NCD structure, the IAD-scheme converges very well. In view of the equivalence between the IAD-scheme and the V-cycle of an algebraic multigrid method, this result can be expected (cf. [77], [54], [30]). The aggregated system with partition $P1$ has 581 states. Furthermore, the block SOR scheme converges faster than point SOR, but the CPU time

increases, of course, with the block size. As partition $P1$ yields a matrix with the small bandwidth of 64 supporting the application of a direct solution method, some efficiency may be gained.
The selected examples have illustrated both the difficulties of the procedures implemented in MACOM and the strategies to overcome them. Last but not least, we want to stress that variants of the proposed algorithms based on regular multisplittings are most suitable for an implementation on vector and multiprocessor computers. This potential of the proposed iteration procedures and their flexibility of adaptation to the structure of a given model are their major advantages.

7 Conclusions

We have discussed a computational approach for modeling and analysis of distributed systems by finite homogeneous continuous-time Markov chains. The concept includes numerical solution methods for finite Markov chains and has been implemented in the tool MACOM.
MACOM provides the user with a predefined model world describing distributed systems in terms of Markovian queueing networks. The package is endowed with a graphical user interface that facilitates the interactive specification of models. The underlying finite Markov chain is automatically generated from this graphical specification. Based on the computation of its steady-state vector, the performance characteristics of the system are evaluated. From a mathematical point of view, this task requires the solution of a linear system with a nonsymmetric singular irreducible M-matrix. MACOM offers the evaluation of different performance characteristics that are specified by the user. It also supports the graphical representation of the calculated results.
We have discussed the algebraic background of some recommended direct and iterative solution methods for finite Markov chains that are employed in MACOM. The GTH algorithm and the point and block SOR procedure with optional aggregation-disaggregation according to the IAD scheme constitute the numerical solver of the package. Using some examples arising from modern telecommunication networks, the benefits and difficulties of the proposed solution methods were pointed out.
Regarding the solution of the singular M-matrix systems arising from the numerical solution of finite Markov chains, a lot of interesting problems are only solved partially or not at all, e.g. the determination of optimal state space partitions and suitable splittings for fast iterative methods or a study of the effect of preconditioning techniques and multilevel algorithms on the rate of convergence. Nevertheless, we are convinced that the proposed iterative algorithms are most suitable solution methods for finite Markov chains since they possess a great potential of adaptation to the stochastic structure of a model and the features for an efficient implementation on a vector or multiprocessor computer. Moreover, the equivalence between the IAD-method and the algebraic multigrid approach can offer new insights into the behavior of iteration schemes and stimulate improvements of the concept in the near future.

Acknowledgment
The author wishes to express his appreciation to M. Sczittnick, S. Zäske, Dr. P. Buchholz, Prof. B. Müller-Clostermann, Prof. H. Beilner, H. Stahl and M. Verwohlt who contributed to the project MACOM.

References

[1] A. A. Ahac and D. D. Olesky. A stable method for the LU factorization of M-matrices. *SIAM J. Alg. Disc. Meth.*, 7, 368–378, 1986.

[2] V. A. Barker. Numerical solution of sparse singular systems of equations arising from ergodic Markov chains. *Commun. Statist.-Stochastic Models*, 5(3), 335–381, 1989.

[3] G. P. Barker and R. J. Plemmons. Convergent iterations for computing stationary distributions of Markov chains. *SIAM J. Alg. Disc. Meth.*, 7(3), 390–398, 1986.

[4] G. P. Barker and S. J. Yang. Semi-iterative and iterative methods for singular M-matrices. *SIAM J. Matrix Anal. Appl.*, 9(2), 168–180, 1988.

[5] J. L. Barlow. On the smallest positive singular value of a singular M-matrix with applications to ergodic Markov chains. *SIAM J. Alg. Disc. Meth.*, 7(3), 414–424, 1986.

[6] J. L. Barlow. Error bounds for the computation of null vectors with applications to Markov chains. In C. D. Meyer and R. J. Plemmons, eds., *Linear Algebra, Markov Chains, and Queueing Models*, pp. 1–12, Springer, New York, 1993.

[7] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences.* Academic Press, New York, 1979.

[8] C. Blondia and O. Casals. Statistical multiplexing of VBR sources: A matrix-analytic approach. *Performance Evaluation*, 16, 5–20, 1992.

[9] A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In D. Evans, ed., *Sparsity and Its Applications*, pp. 257–284, Cambridge University Press, London, 1985.

[10] E. Brockmeyer. Det simple overflowproblem i telefontrafikteorien. *Teleteknik*, 361–374, 1954.

[11] E. Brockmeyer, H. L. Halstrøm, and A. Jensen. *The Life and Works of A.K. Erlang.* Acta Polytechnica Scandinavica, Kopenhagen, 1960.

[12] R. Bru, L. Elsner, and M. Neumann. Models of parallel chaotic iteration methods. *Linear Algebra and its Applications*, 103, 175–193, 1988.

[13] P. Buchholz. *Die strukturierte Analyse Markovscher Ketten.* Informatik-Fachberichte, Vol. 282, Springer, Heidelberg, 1991.

[14] P. Buchholz et al.. *Quantitative Systemanalyse mit Markovschen Ketten.* Teubner-Texte zur Informatik, Vol. 8, Teubner, Stuttgart, 1994.

[15] J. J. Buoni. Incomplete factorization of singular M-matrices. *SIAM J. Alg. Disc. Meth.*, 7, 193–198, 1986.

[16] J. Buoni, M. Neumann, and R. Varga. Theorems of Stein-Rosenberg type: III, the singular case. *Linear Algebra and its Applications*, 42, 183–198, 1982.

[17] F. Cachard. *Création et comparaison de logiciels numériques associés à l' analyse de réseaux de files d' attente dans le cadre de QNAP.* Rapport de Contract, IMAG, Grenoble, 1981.

[18] W. Cao and W. J. Stewart. Iterative aggregation/disaggregation techniques for nearly uncoupled Markov chains. *Journal of the ACM*, 32, 702–719, 1985.

[19] R. H. Chan. Iterative methods for overflow queueing models I. *Numerische Mathematik*, 51, 143–180, 1987.

[20] R. H. Chan. Iterative methods for overflow queueing models II. *Numerische Mathematik*, 54, 57–78, 1988.

[21] R. H. Chan. Iterative methods for queueing networks with irregular state-spaces. In C. D. Meyer and R. J. Plemmons, eds., *Linear Algebra, Markov Chains, and Queueing Models*, pp. 89–109, Springer, New York, 1993.

[22] F. Chatelin. Iterative aggregation/disaggregation methods. In G. Iazeolla, P. J. Courtois, and A. Hordijk, eds., *Mathematical Computer Performance and Reliability*, pp. 199–207, North-Holland, Amsterdam, 1984.

[23] F. Chatelin and W. Miranker. Acceleration by aggregation of successive approximation methods. *Linear Algebra and its Applications*, 43, 17–47, 1982.

[24] K. E. Chu. Bordered matrices, singular systems, and ergodic Markov chains. *SIAM J. Sci. Stat. Comput.*, 11(4), 688–701, 1990.

[25] E. Cinlar. *Introduction to Stochastic Processes.* Prentice Hall, Englewood Cliffs, 1975.

[26] P. Courtois. *Decomposability.* Academic Press, New York, 1977.

[27] P. Courtois and P. Semal. Block decomposition and iteration in stochastic matrices. *Philips J. Res.*, 39, 178–194, 1984.

[28] L. Cvetković and D. Herceg. Relaxation methods for singular M-matrices. *Zeitschrift angew. Math. Mech.*,70, 552–553, 1990.

[29] W. Dahmen and L. Elsner. Algebraic multigrid methods and the Schur complement. In W. Hackbusch, ed., *Robust Multi-Grid Methods*, pp. 58–68, Notes on Numerical Fluid Mechanics, Vol. 23, Vieweg, Braunschweig, 1988.

[30] C. C. Douglas and W. L. Miranker. Constructive interference in parallel algorithms. *SIAM J. Numer. Analysis*, 25(2), 376–398, 1988.

[31] I. S. Duff. Data structures, algorithms and software for sparse matrices. In D. Evans, ed., *Sparsity and Its Applications*, pp. 1–29, Cambridge University Press, London, 1985.

[32] M. Eiermann, R. Varga, and W. Niethammer. Iterationsverfahren für nichtsymmetrische Gleichungssysteme und Approximationsmethoden im Komplexen. *Jahresbericht d. dt. Math.-Verein., Teubner*, 89, 1–32, 1987.

[33] M. Eiermann, I. Marek, and W. Niethammer. On the solution of singular linear systems of algebraic equations by semiiterative methods. *Numerische Mathematik*, 53, 265–283, 1988.

[34] L. Elsner. Comparison of weak regular splittings and multisplitting methods. *Numerische Mathematik*, 56, 283–289, 1989.

[35] D. J. Evans. Iterative methods for sparse matrices. In D. Evans, ed., *Sparsity and Its Applications*, pp. 45–111, Cambridge University Press, London, 1985.

[36] M. Fiedler and V. Pták. On matrices with positive off-diagonal elements and positive principal minors. *Czechoslovakian Math. Journal*, 12, 382–400, 1962.

[37] R. W. Freund. A transpose-free quasi-minimal residual algorithm for non-Hermitian linear systems. *SIAM J. Sci. Stat. Comput.*, 14, 1993.

[38] R. W. Freund, G. H. Golub, and N. M. Nachtigal. Iterative solution of linear systems. *Acta Numerica*, 1, 57–100, 1991.

[39] R. W. Freund and M. Hochbruck. On the use of two QMR algorithms for solving singular systems and applications in Markov chain modeling. Technical Report 91.25, RIACS, NASA Ames Research Center, December 1991.

[40] R. W. Freund and N. M. Nachtigal. QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Numerische Mathematik*, 60, 315–339, 1991.

[41] A. Frommer and G. Meyer. Convergence of relaxed parallel multisplitting methods. *Linear Algebra and its Applications*, 119, 141–152, 1989.

[42] R. Funderlic and J. Mankin. Solution of homogeneous systems of linear equations arising from compartmental models. *SIAM J. Sci. Stat. Comput.*, 2, 375–383, 1981.

[43] R. Funderlic and R. J. Plemmons. A combined direct-iterative method for certain M-matrix linear systems. *SIAM J. Alg. Disc. Meth.*, 5, 33–42, 1984.

[44] R. E. Funderlic, M. Neumann, and R. J. Plemmons. LU decomposition of generalized diagonally dominant matrices. *Numerische Mathematik*, 40, 57–69, 1982.

[45] R. E. Funderlic and R. J. Plemmons. LU decomposition of M-matrices by elimination without pivoting. *Linear Algebra and its Applications*, 41, 99–110, 1981.

[46] F. R. Gantmacher. *Matrizentheorie*. Springer, Heidelberg, 1986.

[47] D. P. Gaver, P. A. Jacobs, and G. Latouche. Finite birth-and-death models in randomly changing environments. *Advances in Applied Probability*, 16, 715–731, 1984.

[48] D. Gibson and E. Seneta. Augmented truncation of infinite stochastic matrices. *Journal of Applied Probability*, 24, 600–608, 1987.

[49] B. W. Gnedenko and D. König. *Handbuch der Bedienungstheorie I.* Akademie-Verlag, Berlin, 1983.

[50] G. H. Golub and C. F. van Loan. *Matrix Computations*. North Oxford, London, 1986.

[51] G. H. Golub and C. D. Meyer. Using the LU factorisation and group inversion to compute, differentiate and estimate the sensitivity of stationary probabilities for Markov chains. *SIAM J. Alg. Disc. Meth.*, 7, 273–280, 1986.

[52] W. Grassmann, S. Kumar, and R. Billinton. A stable algorithm to calculate steady-state probability & frequency of a Markov system. *IEEE Transactions on Reliability*, 36(1), 58–61, 1987.

[53] W. Grassmann, M. Taksar, and D. P. Heyman. Regenerative analysis and steady-state distributions for Markov chains. *Operations Research*, 33(5), 1107–1116, 1985.

[54] W. Hackbusch. *Iterative Lösung großer schwachbesetzter Gleichungssysteme*. Teubner, Stuttgart, 1991.

[55] A. Hadjidimos. On the optimization of the classical iterative schemes for the solution of complex singular linear systems. *SIAM J. Alg. Disc. Meth.*, 6(4), 555–566, 1985.

[56] A. Hadjidimos. Optimum stationary and nonstationary iterative methods for the solution of singular linear systems. *Numerische Mathematik*, 51, 517–530, 1987.

[57] L. A. Hageman and D. M. Young. *Applied Iterative Methods*. Academic Press, New York, 1981.

[58] B. Hajek. Birth-and-death processes on the integers with phases and general boundaries. *Journal of Applied Probability*, 19, 488–499, 1982.

[59] W. J. Harrod. *Rank modification method for certain singular systems of linear equations*. Ph.D. thesis, University of Tennessee, Knoxville, 1982.

[60] W. J. Harrod and R. J. Plemmons. Comparison of some direct methods for computing stationary distributions of Markov chains. *SIAM J. Stat. Comput.*, 5(2), 453–469, 1984.

[61] M. Haviv. Aggregation/disaggregation methods for computing the stationary distribution of a Markov chain. *SIAM J. Numer. Analysis*, 24(4), 952–966, 1987.

[62] U. Herzog, L. Woo, and K. M. Chandy. Solution of queuing problems by a recursive technique. *IBM J. Res. Dev.*, 19, 295–300, 1975.

[63] D. P. Heyman and M. J. Sobel. *Stochastic Models in Operations Research.* Volume I: Stochastic Processes and Operating Characteristics, McGraw-Hill, New York, 1982.

[64] M. Hochbruck. *Lanczos- und Krylov-Verfahren für nicht-Hermitsche lineare Systeme.* Ph.D. thesis, University of Karlsruhe, 1992.

[65] S. Karlin. *A First Course in Stochastic Processes.* Academic Press, New York, 1966.

[66] L. Kaufman. Matrix methods for queueing problems. *SIAM J. Sci. Stat. Comput.*, 4, 525–552, 1983.

[67] L. Kaufman. Solving large sparse linear systems arising in queuing problems. In J. Hinze, ed., *Numerical Integration of Differential Equations and Large Linear Systems, Bielefeld 1980*, pp. 352–360, Springer, Berlin, 1980.

[68] J. P. Kavanagh and M. Neumann. Consistency and convergence of the parallel multisplitting method for singular M-matrices. *SIAM J. Matriz Anal. Appl.*, 10(2), 210–218, 1989.

[69] F. P. Kelly. *Reversibility and Stochastic Networks.* John Wiley, New York, 1979.

[70] J. G. Kemeny. Generalization of a Fundamental Matrix. *Linear Algebra and its Applications*, 38, 193–206, 1981.

[71] J. G. Kemeny and L. J. Snell. *Finite Markov Chains.* Springer, New York, 1976.

[72] D. R. Kincaid, T. C. Oppe, and D. M. Young. Vector computations for sparse linear systems. *SIAM J. Alg. Disc. Meth.*, 7, 99–112, 1986.

[73] J. Kohlas. Numerical computation of mean passage times and absorption probabilities in Markov and Semi-Markov models. *Zeitschrift für Operations Research, Serie A*, 30, 197–207, 1986.

[74] J. R. Koury, D. F. McAllister, and W. J. Stewart. Iterative methods for computing stationary distributions of nearly completely decomposable Markov chains. *SIAM J. Alg. Disc. Meth.*, 5(2), 164–186, 1984.

[75] M. Kramer. Computational methods for Markov chains occurring in queueing theory. In U. Herzog and M. Paterok, eds., *Messung, Modellierung und Bewertung von Rechensystemen*, pp. 164–175, Springer, Berlin, 1987.

[76] U. Krieger. Analysis of a loss system with mutual overflow in a Markovian environment. In W. J. Stewart, ed., *Numerical Solution of Markov Chains*, pp. 303–328, Marcel Dekker, New York, 1991.

[77] U. R. Krieger. *On iterative aggregation/disaggregation methods for finite Markov chains.* submitted, 1994.

[78] U. R. Krieger, B. Müller-Clostermann, and M. Sczittnick. Modeling and analysis of communication systems based on computational methods for Markov chains. *IEEE Journal on Selected Areas in Communications*, 8(9), 1630–1648, 1990.

[79] U. Krieger and M. Sczittnick. A Markovian approach for modelling and analysis of advanced telecommunication networks. In A. Jensen und V. B. Iversen, eds., *Teletraffic and Datatraffic in a Period of Change, Proceedings ITC 13*, pp. 717–722, North-Holland, Amsterdam, 1991.

[80] U. Krieger and M. Sczittnick. Application of numerical solution methods for singular systems in the field of computational probability theory. In R. Beauwens and P. de Groen, *Iterative Methods in Linear Algebra*, North-Holland, Amsterdam, 1992.

[81] A. Kuczura. The interrupted Poisson process as an overflow process. *Bell System Technical Journal*, 52(3), 437–448, 1973.

[82] I. Kuo. A note on factorizations of singular M-matrices. *Linear Algebra and its Applications*, 16, 217–220, 1977.

[83] R. Lal and U. N. Bhat. Reduced systems in Markov chains and their applications in queueing theory. *Queueing Systems*, 2, 147–172, 1987.

[84] J.-Y. Le Boudec. *A generalization of matrix geometric solutions for Markov models.* Research Report RZ 1903, IBM, Zurich Research Laboratory, Rüschlikon, Switzerland, ,1989.

[85] J.-Y. Le Boudec. An efficient solution method for Markov models of ATM links with loss priorities. *IEEE Journal on Selected Areas in Communications*, 9, 408–417, 1991.

[86] F. LeGall and J. Bernussou. An analytical formulation for grade of service determination in telephone networks. *IEEE Trans. on Communications*, 31(3), 420–424, 1983.

[87] R. B. Mattingly. *Vector and parallel algorithms for computing the stationary distribution vector of an irreducible Markov chain.* Ph.D. thesis, North Carolina State University, Raleigh, 1988.

[88] K. Meier-Hellstern. The analysis of a queue arising in overflow models. *IEEE Trans. on Communications*, 37, 367–372, 1989.

[89] K. Meier-Hellstern. Parcel overflows in queues with multiple inputs. In *Proceedings ITC 12, Turino, Italy*, pp. 5.1B.3.1 – 5.1B.3.8, 1988.

[90] J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of Computation*, 31, 148–162, 1977.

[91] C. D. Meyer. Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Review*, 31, 240–272, 1989.

[92] C. D. Meyer. Uncoupling the Perron eigenvector problem. *Linear Algebra and its Applications*, 114/115, 69–94, 1989.

[93] C. D. Meyer. The character of a finite Markov chain. In C. D. Meyer and R. J. Plemmons, eds., *Linear Algebra, Markov Chains, and Queueing Models*, pp. 47–58, Springer, New York, 1993.

[94] C. D. Meyer and W. J. Stewart. Well-conditioned aggregation-disaggregation for nearly uncoupled Markov chains. NCSU Technical Report NA-019301, April 1993.

[95] D. Mitra and P. Tsoucas. Convergence of relaxations for numerical solutions of stochastic problems. In G. Iazeolla, P. J. Courtois, and O. J. Boxma, eds., *Applied Mathematics and Performance/Reliability Models of Computer/Communication Systems*, pp. 119–133, North-Holland, Amsterdam, 1987.

[96] B. Müller. Decomposition methods in the construction and numerical solution of queueing network models. In F. J. Kylstra, ed., *Proceedings of Performance '81*, pp. 99 – 112, North Holland, Amsterdam, 1981.

[97] B. Müller. Numas - a tool for the numerical analysis of computer systems. In D. Potier, ed., *Modelling Techniques and Tools for Performance Analysis*, pp. 141 – 154, North Holland, Amsterdam, 1985.

[98] B. Müller. *Zerlegungsorientierte, numerische Verfahren für Markovsche Rechensystemmodelle.* Ph.D. thesis, Universität Dortmund, Abteilung Informatik, 1980.

[99] B. Müller-Clostermann, M. Sczittnick, and U. R. Krieger. *Modelling and analysis of modern telecommunication networks by Markovian techniques: Foundations, algorithms and examples.* Technical Report, Nr. 341, Informatik IV, Universität Dortmund, 1990.

[100] M. Neumann and R. J. Plemmons. Convergent nonnegative matrices and iterative methods for consistent linear systems. *Numerische Mathematik*, 31, 265–279, 1978.

[101] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models.* Johns Hopkins University Press, Baltimore, 1981.

[102] W. Niethammer and R. Varga. The analysis of k-step iterative methods for linear systems from summability theory. *Numerische Mathematik*, 41, 177–206, 1983.

[103] C. A. O'Cinneide. Entrywise perturbation theory and error analysis for Markov chains. *Numerische Mathematik*, 65, 109–120, 1993.

[104] D. P. O'Leary and R. E. White. Multi-splittings of matrices and parallel solution of linear systems. *SIAM J. Alg. Disc. Meth.*, 6(4), 630–640, 1985.

[105] D. P. O'Leary. Iterative methods for finding the stationary vector for Markov chains. In C. D. Meyer and R. J. Plemmons, eds., *Linear Algebra, Markov Chains, and Queueing Models*, pp. 125–136, Springer, New York, 1993.

[106] C. C. Paige, G. P. H. Styan, and P. G. Wachter. Computation of the stationary distribution of a Markov chain. *J. Statist. Comput. Simul.*, 4, 173–186, 1975.

[107] B. Philippe, Y. Saad, and W. J. Stewart. *Numerical methods in Markov chain modeling.* Operations Research, 40(6), 1992.

[108] R. J. Plemmons. Regular splittings and the discrete Neumann problem. *Numerische Mathematik*, 25, 153–161, 1976.

[109] D. Rose. Convergent regular splittings for singular M-matrices. *SIAM J. Alg. Disc. Meth.*, 5(1), 133–144, 1984.

[110] Y. Saad. Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices. *Linear Algebra and its Applications*, 34, 269–295, 1980.

[111] Y. Saad. Projection Methods for the Numerical Solution of Markov Chain Models. In W. J. Stewart, ed., *Numerical Solution of Markov Chains*, pp. 455–471, Marcel Dekker, New York, 1991.

[112] Y. Saad and M. H. Schultz. Conjugate gradient-like algorithms for solving nonsymmetric linear systems. *Mathematics of Computation*, 44, 417–424, 1985.

[113] M. Sadkane. A block Arnoldi-Chebyshev method for computing the leading eigenpairs of large sparse unsymmetric matrices. *Numerische Mathematik*, 64, 181–193, 1993.

[114] M. Sadkane. Block-Arnoldi and Davidson methods for unsymmetric large eigenvalue problems. *Numerische Mathematik*, 64, 195–211, 1993.

[115] H. Saito, M. Kawarasaki, and H. Yamada. An analysis of statistical multiplexing in an ATM transport network. *IEEE Journal on Selected Areas in Communications*, 9(3), 359–367, 1991.

[116] Y. G. Saridakis. *Parallelism, applicability and optimality of modern iterative methods.* Ph.D. thesis, Clarkson University, 1986.

[117] R. Schassberger. An aggregation principle for computing invariant probability vectors in Semi-Markovian models. In G. Iazeolla, P. J. Courtois, and A. Hordijk, eds., *Mathematical Computer Performance and Reliability*, pp. 259–272, North-Holland, Amsterdam, 1984.

[118] H. Schneider. Theorems on M-splittings of a singular M-matrix which depend on graph structure. *Linear Algebra and its Applications*, 58, 407–424, 1984.

[119] P. J. Schweitzer. Aggregation methods for large Markov chains. In G. Iazeolla, P. J. Courtois, and A. Hordijk, eds., *Mathematical Computer Performance and Reliability*, pp. 275–285, North-Holland, Amsterdam, 1984.

[120] P. J. Schweitzer. A survey of aggregation-disaggregation in large Markov chains. In W. J. Stewart, ed., *Numerical Solution of Markov Chains*, pp. 63–88, Marcel Dekker, New York, 1991.

[121] P. J. Schweitzer and K. W. Kindle. An iterative aggregation-disaggregation algorithm for solving linear equations. *Applied Mathematics and Computation*, 18, 313–353, 1986.

[122] M. Sczittnick. *Techniques for the functional and quantitative analysis of Markovian computing system models (in German).* Master's thesis, Universität Dortmund, 1987.

[123] M. Sczittnick and B. Müller-Clostermann. *MACOM - A tool for the Markovian analysis of communication systems.* Proceedings of the Fourth International Conference on Data Communication Systems and their Performance, Barcelona, Spain, June 20 – 22, 1990.

[124] L. P. Seelen. An algorithm for PH/PH/c queues. *European Journal of Operational Research*, 23, 118–127, 1986.

[125] E. Seneta. *Nonnegative Matrices.* George Allen & Unwin Ltd., London, 1972.

[126] T. J. Sheskin. A Markov chain partitioning algorithm for computing steady-state probabilities. *Operations Research*, 33(1), 228–235, 1985.

[127] G. W. Stewart. Gaussian elimination, perturbation theory, and Markov chains. In C. D. Meyer and R. J. Plemmons, eds., *Linear Algebra, Markov Chains, and Queueing Models*, pp. 59–69, Springer, New York, 1993.

[128] G. W. Stewart and G. Zhang. On a direct method for the solution of nearly uncoupled Markov chains. *Numerische Mathematik*, 59, 1–11, 1991.

[129] W. J. Stewart. Algorithm 570 LOPSI: a simultaneous iteration algorithm for real matrices. *ACM Transactions on Mathematical Software*, 7(2), 230–232, 1981.

[130] W. J. Stewart. A comparison of numerical techniques in Markov modeling. *Communications of the ACM*, 21(2), 144–152, 1978.

[131] W. J. Stewart. *Numerical Solution of Markov Chains.* Marcel Dekker, New York, 1991.

[132] W. J. Stewart. *On the use of numerical methods for ATM models.* IFIP Workshop, Modeling and Performance Evaluation of ATM Technology, La Martinique, January 25–27, 1993.

[133] W. J. Stewart and A. Goyal. *Matrix methods in large dependability models.* IBM Research Report, RC 11485, 11/4/85, IBM, Yorktown Heights, 1985.

[134] W. J. Stewart and W. Wu. Numerical experiments with iteration and aggregation for Markov chains. *ORSA Journal of Computing*, 4(3), 336–350, 1992.

[135] K. Stüben, U. Trottenberg, and K. Witsch. *Software development based on multigrid techniques.* Arbeitspapiere der GMD Nr. 84, GMD, Bonn, 1984.

[136] U. Sumita and M. Rieders. A new algorithm for computing the ergodic probability vector for large Markov chains: Replacement process approach. *Probability in the Engineering and Informational Sciences*, 4, 89–116, 1990.

[137] R. Syski. *Introduction to congestion theory in telephone systems.* North-Holland, New York, 2nd edition, 1986.

[138] D. B. Szyld. Local convergence of (exact and inexact) iterative aggregation. In C. D. Meyer and R. J. Plemmons, eds., *Linear Algebra, Markov Chains, and Queueing Models*, pp. 137–143, Springer, New York, 1993.

[139] H. Tijms. *Stochastic Modelling And Analysis: A Computational Approach.* John Wiley, New York, 1986.

[140] W. Törnig, M. Gipser, and B. Kaspar. *Numerische Behandlung von partiellen Differentialgleichungen der Technik. Mathematische Methoden in der Technik 1*, Teubner, Stuttgart, 1985.

[141] H. Vantilborgh. Aggregation with an error of $O(\epsilon^2)$. *Journal of the ACM*, 32(1), 162–190, 1985.

[142] R. Varga. *Matrix Iterative Analysis.* Prentice-Hall, Englewood Cliffs, 1962.

[143] R. S. Varga and D. Cai. On the LU factorization of M-matrices. *Numerische Mathematik*, 38, 179–192, 1981.

[144] V. L. Wallace and R. S. Rosenberg. Markovian models and numerical analysis of computer system behavior. In *Proc. AFIPS Spring Joint Computer Conference 28*, pp. 141–148, Spartan Books, Washington, 1966.

[145] R. E. White. Multi-splittings with different weighting schemes. *SIAM J. Matrix Anal. Appl.*, 10(4), 481–493, 1989.

[146] D. Wikarski. An algorithm for the solution of linear equation systems with block structure. *Elektronische Informationsverarbeitung und Kybernetik*, 16(10–12), 615–620, 1980.

[147] J. Ye and S. Q. Li. Analysis of multi-media traffic queues with finite buffer and overload control - Part I: Algorithm. In *Proceedings INFOCOM '91*, pp. 1464–1474, 1991.