

SYSTEM-THEORETIC METHODS FOR MODEL REDUCTION OF LARGE-SCALE SYSTEMS: SIMULATION, CONTROL, AND INVERSE PROBLEMS

Peter Benner

Corresponding author: Peter Benner, Professur Mathematik in Industrie und Technik, Fakultät für Mathematik,
Technische Universität Chemnitz, D-09107 Chemnitz, benner@mathematik.tu-chemnitz.de

Abstract. Model (order) reduction, MOR for short, is an ubiquitous tool in the analysis and simulation of dynamical systems, control design, circuit simulation, structural dynamics, CFD, etc. In systems and control, MOR methods based on balanced truncation (BT) and its relatives have been widely used. In other areas, they have been less successful as it is common belief that their computational complexity is too high to apply them to large-scale problems involving sparse matrices. We will review the recent development of efficient algorithms for solving matrix equations that make balancing-related model reduction methods competitive to other MOR approaches - these new implementations fall into the same complexity class as the omnipresent Krylov subspace methods. As balancing-related methods offer the advantage of computable error bounds that allow for an adaptive choice of the order of the reduced model and moreover, they can be shown to preserve certain system properties like stability, passivity, dissipativity, etc., these new BT implementations become attractive in various application areas. These include

- nanoelectronics/VLSI design, where MOR is inevitable for circuit *simulation*,
- (optimal) *control* of physical processes described by partial differential equations (PDEs),
- *inverse problems* related to the identification of input signals, e.g., for tracking control.

We will discuss some particular aspects arising in these areas when applying BT-related MOR techniques. The performance of several BT-related approaches will be demonstrated using examples from a variety of application areas.

1 Introduction

We consider linear, time-invariant (LTI) systems of the form

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), & t > 0, & & x(0) = x^0, \\ y(t) &= Cx(t) + Du(t), & t \geq 0, & & \end{aligned} \quad (1)$$

where $A, E \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$, and $x^0 \in \mathbb{R}^n$ is the initial state of the system. Here, n is the order (or state-space dimension) of the system and $x(t) \in \mathbb{R}^n$, $y(t) \in \mathbb{R}^p$, $u(t) \in \mathbb{R}^m$ are the state, output and input of the system, respectively. In some application areas like structural dynamics, only the differential equation in (1) is used to describe the model dynamics while in other areas like control or circuit simulation, the system description provided in (1) almost always contains the (algebraic) output equation. If the output equation is not present in the mathematical model used to describe the investigated physical process, one might simply set $y(t) = x(t)$, i.e., $C = I_n$, $D = 0$, if a method is to be used that needs the C and D matrices. But often it is also natural in these applications to define specific variables that can serve as outputs as the complete state is seldom measurable in practice.

Applying the Laplace transform to (1) under the assumption that $x(0) = 0$, we obtain a set of algebraic equations from which an input-to-output mapping can be defined as follows:

$$Y(s) = (C(sE - A)^{-1}B + D)U(s), \quad (2)$$

where s is the Laplace variable and Y, U are the Laplace transforms of y and u , respectively. Usually, inputs and outputs are assumed to be in $L_2([0, \infty), \mathbb{R}^q)$, $q = m, p$, respectively. The associated *transfer function matrix (TFM)*

$$G(s) = C(sE - A)^{-1}B + D \quad (3)$$

is a real-rational matrix-valued function. Note that any *restricted system equivalence* with $T, Z \in \mathbb{R}^{n \times n}$ nonsingular, yielding a new system description via

$$(A, B, C, D, E) \mapsto (TAZ, TB, CZ, D, TEZ)$$

leaves the dynamics of the system and its transfer function invariant as can be seen from

$$(CZ)(sTEZ - TAZ)^{-1}(TB) + D = C(sE - A)^{-1}B + D = G(s).$$

Therefore, there exist infinitely many matrix tuples (A, B, C, D, E) representing the same LTI system. Each element of the associated equivalence class is called a *realization* of the LTI system. It is easy to see that there exist realizations of (1) of arbitrarily high order, but there is a lower limit on the order n of the system. This number is called the *McMillan degree* of the system and will be denoted here by \hat{n} . A realization of (1) of order $n = \hat{n}$ is called a *minimal realization*. In the model reduction methods discussed here, we will use several specific realizations of LTI systems.

The model reduction problem considered here consists of finding a reduced-order LTI system,

$$\begin{aligned}\hat{E}\hat{x}(t) &= \hat{A}\hat{x}(t) + \hat{B}u(t), & t > 0, & \hat{x}(0) = \hat{x}^0, \\ \hat{y}(t) &= \hat{C}\hat{x}(t) + \hat{D}u(t), & t \geq 0,\end{aligned}\quad (4)$$

of order r , $r \ll n$, with the same number of inputs m , the same number of outputs p , and associated TFM

$$\hat{G}(s) = \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B} + \hat{D},$$

so that for the same input function $u \in L_2([0, \infty), \mathbb{R}^m)$, we have $y(t) \approx \hat{y}(t)$, or, in frequency domain, $Y(s) \approx \hat{Y}(s)$. Employing the Paley-Wiener theorem (Parseval's identity) and the operator norm induced by the 2-norm in the frequency domain \mathcal{L}_2 , defined for real-rational TFMs by

$$\|G\|_\infty := \sup_{\omega \in \mathbb{R}} \sigma_{\max}(G(i\omega)) \quad (i := \sqrt{-1}, \sigma_{\max} = \text{maximum singular value}), \quad (5)$$

the approximation error can be quantified (at least for stable systems, i.e., the TFM $G(s)$ of the system has all its poles in the open left half plane \mathbb{C}^-) as

$$\|y - \hat{y}\|_2 = \|Y - \hat{Y}\|_2 = \|(G - \hat{G})U\|_2 \leq \|G - \hat{G}\|_\infty \|U\|_2 = \|G - \hat{G}\|_\infty \|u\|_2. \quad (6)$$

Here, $\|\cdot\|$ denotes the 2-norm either in the input and output spaces $L_2([0, \infty), \mathbb{R}^q)$ or the frequency domain \mathcal{L}_2 .

Note that model reduction of discrete-time LTI systems (i.e., linear systems where the dynamics is driven by difference equations) can be formulated in an analogous manner using the Z- instead of the Laplace transformation; see, e.g., [33, 79].

Model (order) reduction is a common task within the simulation, control, and optimization of complex physical processes. Often, large systems arise due to accuracy requirements on the spatial discretization of control problems for fluids or structures described by PDEs, in the context of lumped-circuit approximations of distributed circuit elements, such as the interconnect or package of VLSI chips. or in simulations of micro-electro-mechanical systems (MEMS), which have both electrical and mechanical components, and many other areas. Dimension reduction is generally required for purposes of computational feasibility and/or storage reduction.

Various reduction techniques have been devised, but many of these are described in terms that are discipline-oriented or application-specific even though they share many common features and origins. See the recent monographs and surveys [6, 8, 13, 22, 28, 44, 79, 93] for the discussion of various methods. In case of linear systems, it seems that three approaches play the most prominent role, these are

- modal truncation and the related techniques of substructuring and static condensation,
- Padé and Padé-type approximations, and
- balancing-related truncation techniques.

All three approaches rely on efficient numerical linear algebra techniques to be applicable to very large-scale problems with state-space dimensions of order in the thousands or even in the millions. It is well-known that the first two approaches listed above can be applied to very large-scale problems, see, e.g., [13, 38, 43, 44]. In contrast to common belief, it is also possible to apply balanced truncation techniques for large-scale problems. It is often stated that balanced truncation is not suitable for large-scale problems as it requires the solution of two Lyapunov equations, followed by an SVD and that both steps require $\mathcal{O}(n^2)$ storage and $\mathcal{O}(n^3)$ flops. This is no longer true due to several recent developments in numerical linear algebra, allowing to implement balanced truncation at a cost essentially proportional to the number of nonzeros in A if it is a sparse matrix (see [20, 53, 69, 81]) or in $\mathcal{O}(n \log^2(n))$ (see [17]) if A is approximated by a hierarchical matrix [49]. Here, we will focus on these new techniques.

The outline of the paper is as follows: in the next section, we will give a brief survey of the application of model reduction techniques in the main fields listed in the title of this paper: simulation, control, and (certain) inverse problems. This will motivate the study of variations of the basic balanced truncation technique in the sequel. In Section 3, we provide the necessary background material on balanced truncation and its relatives. In order to apply balanced truncation and relatives to large-scale systems, it is necessary to solve large scale matrix equations (Lyapunov and algebraic Riccati equations). Recent numerical algorithms developed for this purpose will be discussed in Section 4. The efficiency of the balancing-related model reduction techniques will be reported in Section 5 for large-scale problems from benchmark collections in several application fields. We give some concluding remarks in Section 6.

2 Applications of Model Reduction

2.1 Numerical Simulation of Dynamical Systems

Time domain simulation. In practically all application areas of dynamical systems, time domain simulation is a frequently employed technique in order to study the behavior of the physical process described by the mathematical model. In particular, studying the influence of varying initial conditions or forcing functions requires the repeated solution of the underlying system of differential equations. A particular example is transient analysis of electronic circuits, one of the major tasks in very large-scale integrated circuit (VLSI) design. VLSI design is one of the major challenges in micro- and nanoelectronics and is inevitable when developing new electronic devices like CPUs, GPUs, DRAMs, ASICs, and many more, see, e.g., [55] and references therein.

Despite the availability of the variation-of-constants formula for linear systems which yields

$$y(t) = C \exp(At) x^0 + \int_0^t C \exp(A(t-\tau)) B u(\tau) d\tau = C \exp(At) \left(x^0 + \int_0^t \exp(-A\tau) B u(\tau) d\tau \right)$$

as the global, unique solution of (1) (under suitable conditions on $u(t)$ — as we assume $u \in L_2([0, \infty), \mathbb{R}^m)$, these are satisfied), time domain simulation for large-scale LTI systems (1) is usually done via numerical simulation, i.e. via numerical integration techniques. For one, evaluating the exponential map itself is a numerically difficult task [74], and moreover, solving the integral in general requires numerical integration itself.

As many applications leading to large-scale systems of differential equations exhibit a certain degree of stiffness, implicit methods are usually required in order to compute a numerical solution of (1). As the most simple example consider the backwards Euler method. When applied to (1) for a time discretization $t_0 = 0$, $t_{k+1} = t_k + h_k$, $k = 0, 1, \dots$ with mesh sizes $h_k > 0$, this leads to

$$y_h(t_{k+1}) = C(E - h_k A)^{-1} (E x_h(t_k) + h_k B u(t_{k+1})) + D u(t_{k+1}), \quad (7)$$

where $x_h(t), y_h(t)$ denote the numerical approximations to $x(t), y(t)$, respectively. Thus, the time consuming part in solving (1) numerically is the solution of a linear system of equations with the coefficient matrix $\mathcal{A}_k := E - h_k A$. Depending on the structure and the size of the problem, this may be a formidable task when the simulation has to be repeated many times for varying x^0 , different $u(t)$, and possibly a long integration horizon with small step sizes h_k . For higher-order implicit methods, the cost will be higher, but still will be mainly proportional to the cost of solving a linear system of equations with \mathcal{A}_k as coefficient matrix. Although there exists a variety of methods to solve large-scale linear systems efficiently, the computation time may be significantly reduced if a reduced-order model is used to replace (7) by

$$\hat{y}_h(t_{k+1}) = \hat{C}(\hat{E} - h_k \hat{A})^{-1} (\hat{E} x_h(t_k) + h_k \hat{B} u(t_{k+1})) + \hat{D} u(t_{k+1}). \quad (8)$$

This is efficient whenever the solution of a linear system with $\hat{E} - h_k \hat{A}$ as coefficient matrix is faster than when working with \mathcal{A}_k in \mathbb{R}^n . If the approximation error for the reduced-order model can be guaranteed to be negligible compared to the discretization error induced by the numerical integration scheme, then the output function \hat{y}_h from (8) can safely replace y_h as an approximation to the desired function y .

It should be noted, though, that reduced-order models are often computed such that they are basically independent of the choice of $u \in L_2([0, \infty), \mathbb{R}^m)$. The same cannot be claimed for varying x^0 although numerical simulation with reduced-order models often yields satisfactory results in this situation, too. As most methods are based on the assumption that (2) and (6) are valid, the implicit assumption $x^0 = 0$ is made. To the best of our knowledge, there is no detailed analysis about the quality of reduced-order models for time domain simulation with varying $x^0 \neq 0$. Some ideas are discussed in [18], but in general this appears to be an open issue.

Frequency domain analysis In systems and control theory, an often used tool for analyzing LTI systems as in (1) is the frequency response. This is required, e.g., when diagnosing an LTI system using Bode, Nyquist or Nichols plots, see, e.g., [67, 77]. For a system with transfer function $G(s)$ as in (3), frequency response analysis requires the evaluation of

$$G(i\omega_k) = C(i\omega_k E - A)^{-1} B + D, \quad k = 1, \dots, N_f, \quad (9)$$

where the $\omega_k \geq 0$ define a mesh of frequencies on $[0, \infty)$. For large-scale systems, model reduction can reduce the computation time needed for (9) if solving a system of linear equations with $i\omega_k \hat{E} - \hat{A}$ is cheaper than working in \mathbb{C}^n (note that (9) requires complex arithmetic even if A, E are real!) with $i\omega_k E - A$ as coefficient matrix. It should be noted that MOR methods mostly lead to dense matrices \hat{A}, \hat{E} even if A, E are sparse. Thus, it is not clear *per se* that evaluating the transfer function for a reduced-order model is cheaper than using the original model.

A very simple trick can be used to make the evaluation of $\hat{G}(i\omega_k)$, $k = 1, \dots, N_f$, significantly faster so that usually, frequency response analysis using a reduced-order model becomes much faster: compute orthogonal $U, V \in \mathbb{R}^{r \times r}$

such that $U\hat{A}V =: \hat{A}_H$ is upper Hessenberg and $U\hat{E}V =: \hat{E}_H$ is upper triangular. This *Hessenberg-triangular reduction* always exists and can be computed with $15r^3$ operations, see [47, Algorithm 7.7.1]. Next, we set $\hat{B}_H := U\hat{B}$, $\hat{C}_H := \hat{C}V$, and $\hat{D}_H := \hat{D}$. Then the coefficient matrices of all linear systems of equations required in the evaluation of

$$\hat{G}(i\omega_k) = \hat{C}(i\omega_k\hat{E} - \hat{A})^{-1}\hat{B} + \hat{D} = \hat{C}_H(i\omega_k\hat{E}_H - \hat{A}_H)^{-1}\hat{B}_H + \hat{D}_H, \quad k = 1, \dots, N_f,$$

are upper Hessenberg and can thus be solved in $\mathcal{O}(r^2)$ operations only, employing $r - 1$ Gaussian eliminations or Givens rotations. Using the Hessenberg-triangular reduction of course only becomes efficient when the number of frequencies N_f is large enough so that the savings in evaluating $\hat{G}(i\omega_k)$ compensate for the computational overhead needed for transforming the reduced-order model to this form. This procedure was first suggested for standard state-space systems in [64] and is implemented, e.g., in the MATLAB function `frequresp` [70] and the SLICOT¹ function `TB05AD` [27] for systems with $E = I_n$.

Also note that the same trick can be used to accelerate time-domain analysis based on implicit integration schemes applied to a reduced-order model (this pays off at least in case varying step sizes h_k are used in (8) or the corresponding formulae for other integration techniques).

2.2 Feedback Control Design

The main objective in automatic control is to find a feedback control law such that the resulting input function $u(t)$ steers the system to a desired state. A central task is *stabilization*, i.e., the solution trajectory $x(t) := x(t; x^0, u)$ of (1) obtained for the specified control law satisfies $\lim_{t \rightarrow \infty} x(t) = 0$. More generally, control problems are often formulated so that the aim is that the output $y(t)$ of (1) tracks a desired reference output trajectory $y_{ref}(t)$. These and many other problems are solved by feedback control: derive a *controller* (*dynamic compensator*) of the form

$$\begin{aligned} \tilde{E}\dot{\tilde{x}}(t) &= \tilde{A}\tilde{x}(t) + \tilde{B}y(t), \\ u(t) &= \tilde{C}\tilde{x}(t) + \tilde{D}y(t), \end{aligned} \quad (10)$$

where $\tilde{A}, \tilde{E} \in \mathbb{R}^{N \times N}$, $\tilde{B} \in \mathbb{R}^{N \times p}$, $\tilde{C} \in \mathbb{R}^{m \times N}$, and $\tilde{D} \in \mathbb{R}^{m \times p}$, such that for the input $y(t)$ equal to the output of (1), an output $u(t)$ is produced which serves as input to (1) and defines a control law with the desired properties. (If $\tilde{A}, \tilde{E}, \tilde{B}, \tilde{C}$ are all zero, a *static output feedback controller* is defined by (10) and if additionally, $C = I_n$ and $D = 0$, then this yields a *state feedback* control law.)

There exist various approaches to derive the controller (10), see any textbook on control theory, e.g., [4, 67, 77, 89, 104]. Most modern methods like LQR/LQG design, H_2 or H_∞ control lead to controllers with $N \geq n$. If n is large, then this leads to impractical controllers as real-time constraints as well as fragility considerations pose restrictions on the allowable N . Usually, in control engineering, $N \leq 10$ or at most $N = \mathcal{O}(10)$ is desired. Thus, in order to compute a controller of acceptable order, either the plant model (1) or the controller (10) has to be reduced. The latter approach is often called *controller reduction* [79]. A preferred alternative would be to go directly from the high-order plant to a low-order controller. Although there exist approaches in this direction, none of the ones known to the author extend to the large-scale problems (at least $n > 1000$) that we are interested in here — see [79, Section 3.1] for a detailed discussion of this issue. Hence, we will focus on methods that can be used for plant and controller reduction in the context of feedback control design and that are applicable to large-scale problems arising, e.g., in the context of plant models given by spatially discretized instationary PDEs.

2.3 Inverse Problems

Assuming $m = p$, i.e., the number of inputs in (1) equals the number of outputs, so that the transfer function $G(s)$ in (3) is square, and that $D \in \mathbb{R}^{m \times m}$ is invertible, the inverse of the transfer function exists and can be realized as (see, e.g., [97] and references therein)

$$G^{-1}(s) = -D^{-1}C(sE - (A - BD^{-1}C))^{-1}BD^{-1} + D^{-1}. \quad (11)$$

This may be used for certain inverse problems where it is desired to reconstruct the input function from measured outputs: given $Y(s)$, the Laplace transform of $y(t)$, we can then compute

$$U(s) = G^{-1}(s)Y(s). \quad (12)$$

If $u(t)$ is desired in time domain, this can then be computed using the inverse Laplace transformation. An example of the application of system inversion is reported in [71], where heat flux fluctuations at the surface are to be determined in a pool boiling process.

System inversion can also be used to define a control input function $u(t)$: if a desired reference output trajectory $y_{ref}(t)$ is given, the corresponding control function can be computed from (12). It should be noted, though, that this may be an ill-conditioned procedure and using $u(t)$ as feedforward control requires a feedback controller in the control loop in order to compensate possible deviations from the nominal reference trajectory.

¹See www.slicot.org

When using a reduced-order model to compute $U(s)$ in (12), one should be careful due to the possible ill-conditioning of the inverse problem. Often, D is chosen as regularization parameter by setting $D = \epsilon I_m$ (if $D = 0$). This is to be balanced versus the accuracy required for the reduced-order model which makes an error bound for $G^{-1} - \hat{G}^{-1}$ necessary. Fortunately, such bounds can be obtained for some of the MOR methods discussed in the next section.

3 Balanced Truncation and Family

In the following, we will provide the necessary background material for MOR methods based on balancing. For this purpose we assume $E = I_n$ in the following as this simplifies the presentation. For invertible E , this representation of the system can be obtained by simply multiplying the differential part of (1) by E^{-1} from the left, leading to a new realization of the LTI system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), & t > 0, & \quad x(0) = x^0, \\ y(t) &= Cx(t) + Du(t), & t \geq 0, & \end{aligned} \quad (13)$$

where $A \leftarrow E^{-1}A$, $B \leftarrow E^{-1}B$. Note that in practical computations, of course we never compute E^{-1} or form $E^{-1}A, E^{-1}B$. It will turn out that the numerical algorithms described in Section 4 can be implemented based on the formulation (13), where $E^{-1}A, E^{-1}B$ are used just as operators, details will be given below.

Extensions of balancing-related methods to descriptor systems, i.e., systems in the form (1) where E is singular, are possible, see [23, 35, 73, 91, 85].

3.1 The Basic Principle

Inspired by the error bound (6), many system-theoretic model reduction methods for control systems design aim at minimizing $\|G - \hat{G}\|_\infty$, although for a given r , finding \hat{G} that minimizes $\|G - \hat{G}\|_\infty$ is an open problem even in the scalar case [7]. We focus here on model reduction based on balanced truncation and related methods and how these method can be applied to large-scale problems. Even though they usually do not lead to a best approximation, fairly tight computable error bounds are available.

The basic principle of all MOR methods based on balancing is to diagonalize two positive definite matrices $P, Q \in \mathbb{R}^{n \times n}$ via a so-called contragredient transformation, i.e., P is *balanced* vs. Q . This is achieved using a nonsingular matrix $T \in \mathbb{R}^{n \times n}$ chosen such that

$$TPT^T = T^{-T}QT^{-1} = \text{diag}(\sigma_1, \dots, \sigma_n),$$

where $\sigma_1 \geq \dots \geq \sigma_n > 0$. Note that the σ_j 's are the square roots of the eigenvalues of PQ as

$$TPQT^{-1} = \text{diag}(\sigma_1^2, \dots, \sigma_n^2). \quad (14)$$

(Generalizations to positive semidefinite matrices exist, leading to only partially balanced $TPT^T, T^{-T}QT^{-1}$ [92].) Using T as state-space transformation $x \mapsto Tx$ yields a new realization

$$(A, B, C, D) \mapsto (TAT^{-1}, TB, CT^{-1}, D) = \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, [C_1 \quad C_2], D \right) \quad (15)$$

of (13), where $A_{11} \in \mathbb{R}^{r \times r}$, and TB and CT^{-1} are partitioned conformably. With $T = [T_l^T, L_l^T]^T \in \mathbb{R}^{n \times n}$ and $T^{-1} = [T_r, L_r]$, $T_l \in \mathbb{R}^{r \times n}$, $T_r \in \mathbb{R}^{n \times r}$, the reduced-order model is given by the projections

$$\hat{A} := T_l A T_r = A_{11}, \quad \hat{B} := T_l B = B_1, \quad \hat{C} := C T_r = C_1, \quad \hat{D} := D. \quad (16)$$

For given r , the problem now is to find T_l, T_r such that $\|G - \hat{G}\|_\infty$ is small. For this purpose, several suitable choices of P, Q are suggested in the literature, the most prominent one, called *balanced truncation (BT)*, uses the controllability and observability Gramians of (13) [75]. We will describe several of these techniques, including BT, in the following subsections.

As P, Q are assumed to be positive (semi-)definite, they can be factored as $P = S^T S$, $Q = R^T R$, e.g., using Cholesky factorizations. From a numerical point of view, the observation that (16) can be computed using the product SR^T instead of the product PQ as in (14), is a key ingredient of a reliable and efficient implementation of MOR methods based on balancing. The resulting *square-root (SR) algorithm* [65, 92] avoids working with PQ since the condition number of this product is the square of the condition number of the product of the Cholesky factors. Given the Cholesky factors, the singular value decomposition (SVD)

$$SR^T = [U_1 \ U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}, \quad \begin{aligned} \Sigma_1 &= \text{diag}(\sigma_1, \dots, \sigma_r), \\ \Sigma_2 &= \text{diag}(\sigma_{r+1}, \dots, \sigma_n) \end{aligned} \quad (17)$$

is computed, where

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} \geq \sigma_{r+2} \geq \dots \geq \sigma_n \geq 0. \quad (18)$$

For a successful model reduction, r should be chosen to give a natural separation of the states, i.e., one should search for a large gap $\sigma_r \gg \sigma_{r+1}$. Finally, the matrices T_l and T_r yielding the reduced-order model (16) for the balancing state-space transformation are determined by

$$T_l = \Sigma_1^{-1/2} V_1^T R \quad \text{and} \quad T_r = S^T U_1 \Sigma_1^{-1/2}. \quad (19)$$

The possibly ill-conditioned balancing step can be avoided using so-called *balancing-free SR (BFSR) methods* [94]. We refrain here from providing further details on this but note that all methods that we present, also in the large-scale settings, can be implemented using the BFSR approach. Detailed discussions of balancing-related MOR methods and the square-root methods for implementing them can be found in [6, 98]. The implementations described in [98] are contained in SLICOT [27] and the associated SLICOT Model and Controller Reduction Toolbox for MATLAB².

Unfortunately, using Cholesky factors of P, Q , the SVD in (17) requires $\mathcal{O}(n^3)$ flops and $\mathcal{O}(n^2)$ workspace. This will be way too expensive for systems of order $n \gg 1000$. So for the moment, we will focus on reducing the required resources for this computational step by employing low-rank factorizations. This approach will turn out to be the key to the success of the sparse MOR algorithms. The basic idea is to replace the Cholesky factors of the Gramians with low-rank factors, resulting in a smaller arithmetic cost and workspace requirement. So far, we have assumed that the Cholesky factors S and R of the Gramians are square $n \times n$ matrices. For non-minimal systems, we have $\text{rank}(S) < n$ and/or $\text{rank}(R) < n$. Hence, rather than working with the singular Cholesky factors, we may use *full-rank factors* of P, Q . Since P, Q are positive semidefinite, there exist matrices $\hat{S} \in \mathbb{R}^{n_c \times n}$, $\hat{R} \in \mathbb{R}^{n_o \times n}$, such that $P = \hat{S}^T \hat{S}$, $Q = \hat{R}^T \hat{R}$, and

$$n_c := \text{rank}(\hat{S}) = \text{rank}(S) = \text{rank}(P), \quad n_o := \text{rank}(\hat{R}) = \text{rank}(R) = \text{rank}(Q).$$

Although the full-rank factors \hat{S}, \hat{R} can in principle be obtained from S and R , it is more efficient to compute \hat{S} and \hat{R} directly. The matrices U_1, V_1, Σ_1 in (17) that are needed to compute the reduced-order model can then be obtained directly from the SVD of $\hat{S}\hat{R}^T$. This technique yields a significant savings in workspace and computational cost. Using complexity estimates from [47], (17) requires $22n^3$ flops and workspace for $2n^2$ real numbers if U, V are formed explicitly, whereas the SVD of $\hat{S}\hat{R}^T$ requires only $14n_c n_o^2 + 8n_o^3$ flops and workspace for $n_c^2 + n_o^2$ real numbers. In practice, for large-scale dynamical systems, the *numerical rank* of P, Q and \hat{S}, \hat{R} in balancing-related methods is often much smaller than n ; see [9, 32, 48, 82]. This forms the basis for the balancing-related model reduction methods for large-scale problems discussed here.

3.2 Balanced Truncation

Here, we will assume that A from (13) is a stable matrix, i.e., the spectrum of A is contained in the open left half plane. This implies that the system (13) is stable, that is, all the poles of the associated transfer function $G(s)$ have strictly negative real parts. Hence, the model reduction procedure should also yield a stable matrix \hat{A} and stable transfer function $\hat{G}(s)$. Note that not all model reduction techniques automatically lead to a stable reduced-order model. In particular, this is an issue for the abovementioned Padé and Padé-type approximations based on Krylov subspace methods.

The most common approach to truncation-based model reduction involves balancing the controllability Gramian P_{BT} and the observability Gramian Q_{BT} of the system (13) given as the solutions of the Lyapunov equations

$$AP + PA^T + BB^T = 0, \quad A^T Q + QA + C^T C = 0. \quad (20)$$

Using $P_{\text{BT}}, Q_{\text{BT}}$ in place of P, Q in the balancing and truncation procedure described above yields balanced truncation (BT) [75]. BT does not generally yield the best r th order approximant of G in the H_∞ norm, but the following error bound is proven in [45]:

$$\sigma_{r+1}^{\text{BT}} \leq \|G - \hat{G}\|_\infty \leq 2 \sum_{k=r+1}^n \sigma_k^{\text{BT}}, \quad (21)$$

where the σ_j^{BT} 's are square roots of $P_{\text{BT}} Q_{\text{BT}}$ and are the *Hankel singular values* of (13). This a priori error bound makes BT attractive since it allows an adaptive choice of the order r of \hat{G} . Because of this error bound and its ability to preserve important system properties like stability, it is desirable to apply BT to large-scale models. However, Schur vector methods like the Bartels-Stewart algorithm [15] or Hammarling's method [56] for the solution of the Lyapunov equations in (20) require $\mathcal{O}(n^3)$ flops and $\mathcal{O}(n^2)$ workspace. Thus, the key ingredient to a successful application of BT to large-scale systems is a numerical algorithm for solving Lyapunov equations that can be implemented at a computational cost proportional to the cost of solving linear systems

²See <http://www.slicot.org> for further details.

of equations with coefficient matrix A . That is, if A is sparse with nz nonzero entries, the complexity should be $\mathcal{O}(nz)$. Moreover, such an algorithm needs to compute low-rank approximate factors $\hat{S}_{\text{BT}}, \hat{R}_{\text{BT}}$ without ever forming $P_{\text{BT}}, Q_{\text{BT}}$. This is achieved by the method described in Section 4.1. For further information on efficient BT methods see [10, 17, 20, 30, 53, 68, 84].

3.3 Balancing-Related Methods

There is a large-variety of balancing-related model reduction techniques, see, e.g., [6, 52, 79]. The main idea of all balancing-related methods is to replace the Gramians $P_{\text{BT}}, Q_{\text{BT}}$ defined in (20) by other symmetric, positive semidefinite matrices P, Q . Usually, P, Q define controllability and observability Gramians of LTI systems in some way related to (13). For detailed discussions of choices of various Gramians see, e.g., [79, 90]. Here, we will focus only on three approaches that are useful for the main application fields considered here and for which we have developed variants that can be applied to large-scale systems with sparse state matrices A .

3.3.1 LQG Balanced Truncation

If it comes to unstable systems, BT can no longer be applied in the form described in the previous subsection. There are extensions to unstable systems based on additive decomposition of the transfer function or frequency domain definitions of the Gramians, see, e.g., [14, 96, 105]. Unfortunately, none of them extends easily to large-scale systems as they require dense matrix algebra. New approaches that can deal with sparsity or sparse representations of A are under current investigation (a version employing formatted arithmetic for hierarchical matrices can be based on [16]), but here we will limit ourselves to a different balancing approach that can be employed in case of unstable systems. Moreover, the approach can also be efficiently be used to obtain a reduced-order controller. Despite the fact that such a low-order controller can also be computed based on a BT reduced-order model (if $G(s)$ is stable), there is a subtle problem with this approach, in particular in the context of infinite-dimensional systems such as control of parabolic PDEs: controllers based on the BT reduced-order model may not be robust when applied to the original infinite-dimensional problem. Therefore, in [40] it is suggested to use instead of BT a technique called LQG balancing [59]. It is then proven in [40] that a robust controller can be based on the reduced-order model computed by the truncated LQG balanced system. The basic idea of LQG balanced truncation (LQGBT) is to replace the Gramians P_{BT} and Q_{BT} from (20) by the stabilizing solutions of the dual algebraic Riccati equations (AREs)

$$\begin{aligned} 0 &= AP + PA^T - PC^T C P + BB^T, \\ 0 &= A^T Q + QA - QBB^T Q + C^T C, \end{aligned} \quad (22)$$

related to the regulator and filter AREs used in linear-quadratic Gaussian (LQG) control design. By *stabilizing solutions* we mean that $A - PC^T C$ and $A - BB^T Q$ have all their eigenvalues in the open left half plane \mathbb{C}^- . It is well-known (see [62] and references therein) that P_{LQG} and Q_{LQG} are positive semidefinite. Like the solutions of (20), often they can be approximated by low-rank factorizations $\hat{S}_{\text{LQG}}^T \hat{S}_{\text{LQG}}$ and $\hat{R}_{\text{LQG}}^T \hat{R}_{\text{LQG}}$. In order to use this approach for model reduction of large-scale systems, we thus need numerical algorithms for large-scale AREs that compute $\hat{S}_{\text{LQG}}, \hat{R}_{\text{LQG}}$ directly without ever forming $P_{\text{LQG}}, Q_{\text{LQG}}$. As we will see in Section 4, such a method can be based on Newton's method for AREs if a corresponding solver for the Lyapunov equations (20) is available. With such an ARE solver we arrive at implementations of LQGBT that can be applied to the very large-scale systems we are interested in.

Like BT, LQGBT comes with a computable error bound that becomes available once $P_{\text{LQG}}, Q_{\text{LQG}}$ or its factors are known. As the original and reduced-order systems are in general not stable, the H_∞ -norms of G, \hat{G} and $G - \hat{G}$ are usually not defined, thus the error bound is derived in a slightly different way than for standard BT. First note that if the considered LTI system is stabilizable and detectable without uncontrollable poles on the imaginary axis, its transfer function can be factored as $G = M^{-1}N$ where M, N are stable, rational transfer functions with $M \in \mathbb{R}^{p \times p}[s]$ and $N \in \mathbb{R}^{p \times m}[s]$. (This is called a *left coprime factorization*.) The augmented transfer function $[N, M] \in \mathbb{R}^{p \times m+p}[s]$ is stable and thus can be measured in the H_∞ -norm. With this, the following error bound is obtained in [59]:

$$\| [N \ M] - [\hat{N} \ \hat{M}] \|_\infty \leq 2 \sum_{j=r+1}^n \frac{\sigma_j^{\text{LQG}}}{\sqrt{1 + (\sigma_j^{\text{LQG}})^2}}, \quad (23)$$

where $G = M^{-1}N, \hat{G} = \hat{M}^{-1}\hat{N}$ are left coprime factorizations of G, \hat{G} and σ_j^{LQG} are the singular values obtained in (17) if S, R are replaced by the corresponding factors of the stabilizing solutions $P_{\text{LQG}}, Q_{\text{LQG}}$ of (22).

Summarizing, LQGBT is useful in simulation of unstable systems and in control design. Note that the stabilizing solutions of the LQG AREs corresponding to the obtained reduced-order model and thus the LQG controller for the reduced-order model are readily available as by-product of the computation of the LQGBT reduced-order model. Furthermore, it can be shown that this controller exponentially stabilizes even infinite-dimensional plant models under certain assumptions on the spatial discretization [76]. Nevertheless, this controller may not be robust which partially motivates the work in [40] mentioned above.

3.3.2 Positive-Real Balanced Truncation

Positive-real balanced truncation (PRBT) is mostly known as a method that preserves passivity in the reduced-order model. An LTI system is *passive* if

$$\int_{-\infty}^t u(\tau)^T y(\tau) d\tau \geq 0$$

for all $t \in \mathbb{R}$ and all $u \in L_2(\mathbb{R}, \mathbb{R}^m)$. In practice this means that the system cannot generate energy. It is a classical result of network theory [5] that a system is passive if and only if its transfer function $G(s)$ is *positive real*, that is,

1. G is analytic in $\mathbb{C}^+ := \{s \in \mathbb{C} \mid \text{Re}(s) > 0\}$,
2. $G(s) + G^T(\bar{s}) \geq 0$ for all $s \in \mathbb{C}^+$.

This shows that in order to retain passivity in the reduced-order model, we need its transfer function \hat{G} to be positive real. This can be achieved by truncating a *positive real balanced realization* of the system. Such a realization is obtained if the two positive-real Gramians $P_{\text{PR}}, Q_{\text{PR}}$, defined as the minimal positive (semi-)definite solutions of the AREs

$$\begin{aligned} 0 &= (A - BR^{-1}C)P + P(A - BR^{-1}C)^T + PC^T R^{-1}CP + BR^{-1}B^T, \\ 0 &= (A - BR^{-1}C)^T Q + Q(A - BR^{-1}C) + QBR^{-1}B^T Q + C^T R^{-1}C, \end{aligned} \quad (24)$$

where $R := D + D^T$, are diagonal and equal. A PRBT reduced-order model is then computed analogously to classical BT, just the Gramians (or their factors) are replaced by $P_{\text{PR}}, Q_{\text{PR}}$ (or their factors) from (24). It can then be shown [78] that the resulting reduced-order model is stable and passive.

Thus, the core computation of PRBT is the solution of the two coupled AREs in (24) similarly to LQGBT. A slight difference arises from the opposite signs of the quadratic terms in (22) and (24) which makes the solution of (24) with the method described in Section 4 slightly more expensive. Still, we will see that we are able to compute (approximate) low-rank factors of $P_{\text{PR}}, Q_{\text{PR}}$ without ever forming them explicitly.

The diagonal entries $\sigma_1^{\text{PR}} \geq \sigma_2^{\text{PR}} \geq \dots \geq \sigma_n^{\text{PR}} \geq 0$ of the balanced positive real Gramians are called the *positive real Hankel singular values* and can be used to derive error bounds for the approximation error. Using the error bound (where $\|\cdot\|$ denotes the spectral norm for matrices)

$$\|G_D^{-1} - \hat{G}_D^{-1}\|_{\infty} \leq 2\|R\|^2 \sum_{k=r+1}^n \sigma_k^{\text{PR}} \quad (25)$$

for $G_D(s) := G(s) + D^T$, $\hat{G}_D(s) := \hat{G}(s) + D^T$ given in [52], the following error bound is obtained in [21]:

$$\|G - G_r\|_{\infty} \leq 2\|R\|^2 \|\hat{G}_D\|_{\infty} \|G_D\|_{\infty} \sum_{k=r+1}^n \sigma_k^{\text{PR}}. \quad (26)$$

Note that in [52], PRBT is also slightly modified so that an error bound without the $\|\hat{G}_D\|_{\infty} \|G_D\|_{\infty}$ term on the right-hand side can be derived. The resulting method is not applicable to all positive real systems, though, for details see [6, 52].

The error bound (26), though sometimes quite pessimistic, distinguishes PRBT from all passivity-preserving model reduction methods based on Padé and Padé- type approximation like PRIMA or SyPVL (see [44] and the references therein). It also has the advantage to be applicable to general passive systems while the aforementioned methods require additional structure arising in RLC circuits. On the other hand, PRBT is only applicable if $D + D^T$ is positive definite. This condition can be relaxed by considering the Lur'e (positive real) equations [85], but so far no numerical method for large-scale problems is known for this. On the other hand, computational experience shows that for systems with $D = 0$, PRBT usually yields satisfactory results if applied to a regularized system with $D = \varepsilon I$, where the regularization term is then again set to zero in the reduced-order model.

Thus, PRBT is a useful method for simulation of passive LTI systems and therefore particularly interesting in VLSI design, circuit simulation, and other areas of microelectronics and microwave theory.

PRBT is also interesting for the inverse problems described in Section 2.3 due to the error bound (25) if $\|D\|$ is small.

3.3.3 Balanced Stochastic Truncation

Often it is desirable that the reduced-order system has uniform approximation properties over the whole frequency range $0 \leq \omega \leq \infty$ or gives a particularly good approximation at prescribed frequencies. For example, this is the case if the LTI system describes a high-order controller that should perform well at practically relevant frequencies. This requirement can be satisfied by relative error methods. They attempt to minimize the relative error $\|\hat{\Delta}\|_{\infty}$, defined implicitly by $G - \hat{G} = G\hat{\Delta}$. Among these, balanced stochastic truncation (BST) [42, 50, 99] is particularly popular.

BST is a model reduction method based on truncating a *balanced stochastic realization* of (13). Such a realization can be achieved if we assume that $0 < p \leq m$ and $\text{rank}(D) = p$ which implies that $G(s)$ must not be strictly proper. For strictly proper systems, the method can be applied when introducing an ε -regularization by adding an artificial matrix $D = \begin{bmatrix} \varepsilon I_p & 0 \end{bmatrix}$ [46]. We also need to assume that $G(s)$ has no zeros on the imaginary axis.

A balanced stochastic realization is obtained as follows. Define the power spectrum $\Phi(s) = G(s)G^T(-s)$ of $G(s)$, and let W be a *square minimum phase right spectral factor* of Φ , satisfying $\Phi(s) = W^T(-s)W(s)$. As D has full row rank, DD^T is positive definite and a minimal state-space realization (A_W, B_W, C_W, D_W) of W is given by (see [2, 3])

$$A_W = A, \quad B_W = BD^T + P_{\text{BST}}C^T, \quad C_W = (DD^T)^{-\frac{1}{2}}(C - B_W^T Q_{\text{BST}}), \quad D_W = (DD^T)^{\frac{1}{2}}.$$

Here, P_{BST} is the controllability Gramian of $G(s)$ given by the solution of the first Lyapunov equation in (20), while Q_{BST} is the observability Gramian of $W(s)$ obtained as the stabilizing solution of the ARE

$$(A - B_W(DD^T)^{-1}C)^T Q + Q(A - B_W(DD^T)^{-1}C) + Q_B W(DD^T)^{-1} B_W^T Q + C^T (DD^T)^{-1} C = 0. \quad (27)$$

In the balancing and truncation procedures described above, we now use $P = P_{\text{BST}}$, $Q = Q_{\text{BST}}$. Again, for practical purposes, in particular large-scale settings, we compute (approximate) low-rank factors of $P_{\text{BST}}, Q_{\text{BST}}$ directly and use them in (17) as well as the subsequent computations instead of S, R .

If (13) is stable and minimal, then the reduced-order model computed by BST is stable and has the following properties [51, 79]:

Proposition 1 *If $G(s)$ is square, minimal and stable, $\hat{G}(s)$ computed by BST satisfies the relative error bound*

$$\sigma_{k+1}^{\text{BST}} \leq \|\hat{\Delta}\|_{\infty} = \|G^{-1}(G - \hat{G})\|_{\infty} \leq \prod_{j=r+1}^n \frac{1 + \sigma_j^{\text{BST}}}{1 - \sigma_j^{\text{BST}}} - 1, \quad (28)$$

where the σ_j^{BST} denote the square roots of the eigenvalues of $P_{\text{BST}}Q_{\text{BST}}$.

This error bound can be extended to nonsquare systems using certain modifications [79, 99]. It should be noted that the *stochastic Hankel singular values* σ_j^{BST} satisfy $|\sigma_j^{\text{BST}}| \leq 1$. Moreover, the number of σ_j^{BST} 's with $|\sigma_j^{\text{BST}}| = 1$ equals the number of zeros of $G(s)$ in \mathbb{C}^+ . (Hence, r needs to be at least as large as this number!) Thus, for minimum-phase systems, $|\sigma_j^{\text{BST}}| < 1$ for all $j = 1, \dots, n$. Due to its nature as relative error bound, (28) implies that as desired, BST reduced-order models tend to have an evenly distributed approximation error in contrast to BT where the approximation at high frequencies is often significantly better than for low frequencies. Another advantage over BT is that there also exists a bound for the phase error [103].

The uniform approximation properties make BST a preferable MOR method for simulation if good approximation qualities over the whole frequency range or long time intervals are desired. As is pointed out in the literature, relative-error methods are also useful for controller reduction, see, e.g., [79]. Also, BST possesses a certain robust stability property: certain controllers for the BST reduced-order plant model can be proven to stabilize the full-order plant [86].

Thus, the method is interesting in two out of the three main application areas of MOR discussed in Section 2. But it turns out that BST also possesses interesting properties regarding the third area, namely inverse problems: starting out from the definition of the relative error,

$$\hat{\Delta}(s) = G^{-1}(s)(G(s) - \hat{G}(s)),$$

simple algebra (assuming \hat{G}^{-1} exists) leads to

$$G^{-1}(s) - \hat{G}^{-1}(s) = \hat{\Delta}(s)\hat{G}^{-1}(s).$$

This yields an H_{∞} error bound if \hat{G}^{-1} is stable. The latter property can be guaranteed for *minimum-phase systems*, i.e., systems that have no zeros in \mathbb{C}^+ due to the following property of BST [50, 79]:

Proposition 2 *Zeros of $G(s)$ are preserved in $\hat{G}(s)$, i.e., if $G(s)$ is rank-deficient, so is $\hat{G}(s)$. Moreover, if $G(s)$ is minimum-phase, then $\hat{G}(s)$ is minimum-phase, too.*

As the zeros of a transfer function become the poles of its inverse and due to minimality, we have no pole-zero cancellations, \hat{G}^{-1} computed using the BST reduced-order model is stable if G is minimum-phase. Thus, we have the following result:

Theorem 3.1 *If $G(s)$ is square, minimal, stable, minimum-phase, and nonsingular on the imaginary axis, then for $\hat{G}(s)$ computed by BST, \hat{G}^{-1} exists and is stable. Moreover, we have the following error bound:*

$$\|G^{-1} - \hat{G}^{-1}\|_{\infty} \leq \left(\prod_{j=r+1}^n \frac{1 + \sigma_j^{\text{BST}}}{1 - \sigma_j^{\text{BST}}} - 1 \right) \|\hat{G}^{-1}\|_{\infty}. \quad (29)$$

Note that the evaluation of $\|\hat{G}^{-1}\|_\infty$ is fairly cheap for a reduced-order model, so that (29) is a useful error bound that can be used in practical computations employing system inversion.

There is another beneficial property of minimum-phase systems: the solution of the ARE (27) can be avoided as it is shown in [79] that with R_{BST} solving the Lyapunov equation

$$(A - BD^{-1}C)^T R + R(A - BD^{-1}C) + C^T(DD^T)^{-1}C = 0,$$

the BST reduced-order model is obtained by balancing P_{BST} versus R_{BST} instead of Q_{BST} . The stochastic Hankel singular values are then

$$\sigma_j^{\text{BST}} = \frac{\alpha_j}{\sqrt{1 + \alpha_j^2}},$$

where the α_j 's are the square roots of the eigenvalues of $P_{\text{BST}}R_{\text{BST}}$. This makes the computation of the BST reduced-order model essentially as efficient as computing the BT reduced-order model.

4 Numerical Algorithms for Large-Scale Matrix Equations

We will focus on methods for solving the Lyapunov equations (20) and AREs (22), (24), (27) that compute approximate low-rank factors \hat{S}, \hat{R} of the solutions directly. Furthermore, we will only treat one type of methods which is based on the ADI iteration [100] for Lyapunov equations and the Newton-Kleinman iteration [60] for AREs. The methods are derived and discussed in detail in [19, 20, 25, 69, 81]. There are several new ideas [26, 58, 88] that can improve the performance of the methods, but due to space limitations we skip a discussion of all these approaches.

4.1 Low-Rank ADI for Lyapunov Equations

In this section we consider the Lyapunov equation

$$FX + XF^T + WW^T = 0, \quad A \in \mathbb{R}^{n \times n}, \quad W \in \mathbb{R}^{n \times w}, \quad (30)$$

where A is stable. The latter assumption is equivalent to (30) having a unique solution [63]. Let $X \in \mathbb{R}^{n \times n}$ be this unique solution. When applied to (20), $F \in \{A, A^T\}$ and $W \in \{B, C^T\}$.

The ADI iteration for solving Lyapunov equations (30) can be written as follows [100]:

$$\begin{aligned} (F + \mu_j I)X_{(j-1)/2} &= -W - X_{j-1}(F^T - \mu_j I), \\ X_j(F^T + \bar{\mu}_j I) &= -W - (F - \bar{\mu}_j I)X_{(j-1)/2}. \end{aligned}$$

If the shift parameters μ_j are chosen appropriately, then $\lim_{j \rightarrow \infty} X_j = X$ with $X_0 = 0$. Note that each iteration step requires the solution of a system of linear equations with a shifted version of F as coefficient matrix. Thus, this can be implemented efficiently if F is sparse. Still, in the formulation above, the full solution X is computed after convergence and in each iteration, the number of right-hand sides in the systems of linear equations is n . Thus, even if F is sparse, the ADI iteration would be computationally too expensive if applied in this form. In the following, we will therefore describe how the ADI iteration can be modified to yield an (approximate) low-rank solution factor directly. First of all, this is desired for the balancing-related MOR methods as discussed in Section 3. Second, it will turn out that the number of right-hand sides in the systems of linear equations comes down to m, p , or $m + p$, depending on the MOR method chosen.

Starting the ADI iteration with $X_0 = 0$ and observing that for stable F , X is positive semidefinite, we can assume that $X_j = Y_j Y_j^T$ for some $Y_j \in \mathbb{R}^{n \times r_j}$. Inserting this into the above iteration, re-arranging terms and combining two iteration steps, we obtain the following *factored ADI iteration*:

$$\begin{aligned} V_1 &\leftarrow \sqrt{-2\text{Re}(\mu_1)}(F^T + \mu_1 I)^{-1}W, \quad Y_1 \leftarrow V_1 \\ \text{FOR } j &= 2, 3, \dots \\ V_j &\leftarrow \frac{\sqrt{\text{Re}(\mu_j)}}{\sqrt{\text{Re}(\mu_{j-1})}}(V_{j-1} - (\mu_j + \bar{\mu}_{j-1})(F^T + \mu_j I)^{-1}V_{j-1}), \\ Y_j &\leftarrow \begin{bmatrix} Y_{j-1} & V_j \end{bmatrix}. \\ \text{END FOR} \end{aligned}$$

It should be noted that all V_j 's have the same number of columns as $W \in \mathbb{R}^{n \times w}$, i.e., at each iteration j , we have to solve w linear systems of equations with the same coefficient matrix $F^T + \mu_j I$. Hence, if convergence with respect to a suitable stopping criterion is achieved after j_{max} steps, $Y_{j_{\text{max}}} = [V_1 \ \dots \ V_{j_{\text{max}}}] \in \mathbb{R}^{n \times j_{\text{max}}w}$. For large n and small w we can therefore expect that $r_j := j_{\text{max}}w \ll n$. In that case, we have computed a low-rank approximation $Y_{j_{\text{max}}}$ to a factor Y of the solution, i.e., $X = YY^T \approx Y_{j_{\text{max}}} Y_{j_{\text{max}}}^T$. In case $w \cdot j$ becomes too large during the iteration, we apply a column compression technique to Y_j based on a rank-revealing QR factorization (RRQR) [47] as suggested first in [31] for a sign function-based Lyapunov solver. Usually, the number of columns of $Y_{j_{\text{max}}}$

is then reduced significantly without adding a significant error. A different column compression technique based on a computationally more involved SVD is suggested in [54].

If $F^T + \mu_j I_n$ is a banded matrix or can be re-ordered to become banded, then a sparse direct solver can be employed. If workspace permits, it is then desirable to compute a factorization of $F^T + \mu_j I_n$ for each different shift parameter beforehand (usually, very few parameters are used). These factorizations can then be used in each iteration step of the ADI iteration. In particular, if F is symmetric positive definite as will be the case in many applications from PDE constraint optimal control problems, and can be re-ordered to a (narrow) band matrix, then each factorization is $\mathcal{O}(n)$ and the total cost $\mathcal{O}(k_{\max} \max(j_{\max})n)$ scales with n as desired. The same is also true for other than band patterns such as arrowhead matrices. If iterative solvers are employed for the linear systems, it should be noted that due to shift-invariance properties, only one Krylov space needs to be computed (see [69] for details) and hence we obtain again an efficient implementation of the factored ADI iteration.

For an implementation of the factored ADI method, we need a strategy to select the shift parameters. We will not treat this issue here in depth; see [29, 101, 102] for a detailed discussion. We only note that usually, a finite number of shifts is computed in advance and applied cyclically if the ADI method needs more iterations than the number of available shifts. For complex shifts, a real version of the factored ADI iteration is derived in [25]. Another approach in this case is to use only the real parts of the shifts obtained by the heuristic shift selection strategy proposed in [80], see [29] for further discussion of this issue. The convergence of the method, in particular in the case of badly chosen shifts, can significantly be improved when combined with the Galerkin projection approach suggested in [57]; see [26] for details.

One important issue in the implementation of the factored ADI iteration arises in the application to problems resulting from an LTI system (1) with $E \neq I_n$. Typically, this situation is encountered when (1) is obtained from a spatial finite element discretization of an instationary PDE. If we then re-write the system as in (13), we obtain $F = E^{-1}A$ and $W = E^{-1}B$. Considering (30) in this situation, we need to avoid forming F explicitly as A will usually be a dense matrix even if A, E are sparse. There are several approaches that can be used to treat this problem. For instance, if we want to solve a linear system of the form $(F + pI)v = b$ during the ADI iteration, the coefficient matrix has the form $E^{-1}A + pI_n = E^{-1}(A + pE)$. Thus, at the additional cost of a sparse matrix-vector multiply, we get the solution v from the linear system of equations $(A + pE)v = Eb$ [20]. Further details of the corresponding implementation of the factored ADI iteration can be found in [37].

Note that with the described approach for solving Lyapunov equations, we can implement an efficient version of BT using $\hat{S}_{\text{BT}} = Y_{j_{\max}}^T$ obtained from applying the factored ADI iteration to $F = A$, $W = B$, $\hat{R}_{\text{BT}} = Y_{j_{\max}}^T$ obtained with $F = A^T$, $W = C^T$, and the small-size SVD of $\hat{S}_{\text{BT}}\hat{R}_{\text{BT}}^T$ as described in Section 3.1. Some advantage can be taken from the fact that all the coefficient matrices in both iterations (for $\hat{S}_{\text{BT}}, \hat{R}_{\text{BT}}$) only differ by the choice of shifts and are transposes of each other.

The factored ADI iteration is also used for solving the Lyapunov equation(s) encountered in BST and during the iteration steps when solving the AREs needed for LQGBT, PRBT, BST with Newton's method as described in the following subsection.

4.2 Low-Rank Newton-ADI for Algebraic Riccati Equations

In this section we consider the ARE

$$0 = \mathcal{R}(Z) := C^T C + A^T Z + ZA - ZBB^T Z, \quad (31)$$

where A, B, C can denote A, B, C as in (13), but may also represent the coefficient matrices in the AREs (22), (24), or (27). In all situations encountered here, we are interested in computing the stabilizing solution Z_* of (31) i.e., all eigenvalues of $A - BB^T Z_*$ are in the open left half plane. Under the assumptions required for any of the balancing-related MOR methods considered here, Z_* is positive semidefinite.

The ARE (31) is a nonlinear system of equations. Hence, it is quite natural to apply Newton's method to find its solutions. This approach has been investigated, e.g., in [60, 72, 62, 87]. With the Frechét derivative of $\mathcal{R}(Z)$ at Z ,

$$\mathcal{R}'_Z : N \rightarrow (A - BB^T Z)^T N + N(A - BB^T Z)$$

and the Newton-Kantorovich method for operator-valued functions, $Z_{k+1} = Z_k - (\mathcal{R}'_{Z_k})^{-1} \mathcal{R}(Z_k)$, $k = 0, 1, 2, \dots$, we obtain *Newton's method for AREs* for a given starting matrix Z_0 :

- FOR $k = 0, 1, 2, \dots$
1. $A_k \leftarrow A - BB^T Z_k$.
 2. Solve the Lyapunov equation $A_k^T N_k + N_k A_k = -\mathcal{R}(Z_k)$.
 3. $Z_{k+1} \leftarrow Z_k + N_k$.
- END FOR k

Assume that A_0 is stable, i.e., a stabilizing Z_0 is given. Then all A_k are stable and $\lim_{k \rightarrow \infty} Z_k = Z_*$ quadratically.

In order to make the Newton iteration work for large-scale problems, we need a Lyapunov equation solver which employs the structure of A_k as “sparse + low-rank perturbation” by avoiding to form A_k explicitly and which computes a low-rank approximation to the solution of the Lyapunov equation. A method that can be employed here is derived in detail in [80, 25] and will be described in short in the following.

As suggested originally in [60], we can re-write Newton’s method for AREs to the *Newton-Kleinman form* such that the next iterate is computed directly from the Lyapunov equation in Step 2,

$$A_k^T Z_{k+1} + Z_{k+1} A_k = -C^T C - Z_k B B^T Z_k =: -W_k W_k^T. \quad (32)$$

If we assume that $Z_k = Y_k Y_k^T$ for $\text{rank}(Y_k) \ll n$ and observe that $\text{rank}(W_k) \leq m + p \ll n$, we see that all we need is a numerical method to solve Lyapunov equations having a low-rank right hand side which returns a low-rank approximation to the (Cholesky) factor of its solution. As A_k is stable for all k we can apply the factored ADI iteration described in Section 4.1 to (32). Note that then, $W = \begin{bmatrix} C^T & Y_k(Y_k^T B) \end{bmatrix}$ and hence, $w = m + p$, so that usually, $w \ll n$. If the factored ADI iteration is applied to $F = A_k^T$ from (32), we have to deal with the situation that A_k is a shifted sparse matrix plus a low-rank perturbation. If we can solve for the shifted linear system efficiently, the low-rank perturbation can be dealt with using the Sherman-Morrison-Woodbury formula [47] in the following way: let k be the index of the Newton iterates and let j be the index of the ADI iteration used to solve the k th Lyapunov equation, respectively, and set $K_k := (B^T Y_k) Y_k^T$, then

$$\left(F^T + \mu_j^{(k)} I_n\right)^{-1} = \left(A + \mu_j^{(k)} I_n - B K_k\right)^{-1} = \left(I_n + L_k (I_m - K_k L_k)^{-1} K_k\right) \left(A + \mu_j^{(k)} I_n\right)^{-1},$$

where $L_k := (A + \mu_j^{(k)} I_n)^{-1} B$. Hence, all linear systems of equations to be solved in one iteration step have the same coefficient matrix $A + \mu_j^{(k)} I_n$.

The factored Newton-ADI method can be applied to the LQG AREs (22) directly. For the AREs (24) and (27) encountered in PRBT and BST, we need a slight modification due to the fact that the quadratic term there is positive semidefinite rather than negative definite as in (31). As a consequence, the right-hand side in (32) will in general not be semidefinite and can thus not be written as $-W_k W_k^T$. For instance, consider the second ARE in (24). If we denote the Newton iterates in this case by Q_j , the right-hand side in the Lyapunov equation in the j th Newton step becomes

$$-C^T R^{-1} C + Q_j B R^{-1} B^T Q_j =: -\tilde{W}_j \tilde{W}_j^T + \hat{W}_j \hat{W}_j^T.$$

(Note that R is positive definite and thus a Cholesky factorization of R^{-1} can be used to obtain this representation of the right-hand side.) As the Lyapunov equation

$$A_j^T Q_{j+1} + Q_{j+1} A_j = -\tilde{W}_j^T \tilde{W}_j + \hat{W}_j^T \hat{W}_j$$

is a nonsingular linear system of equations, its unique symmetric solution can be written as the difference of two positive semidefinite matrices:

$$A_j^T (\tilde{Q}_{j+1} - \hat{Q}_{j+1}) + (\tilde{Q}_{j+1} - \hat{Q}_{j+1}) A_j = -\tilde{W}_j \tilde{W}_j^T - (-\hat{W}_j \hat{W}_j^T),$$

where $\tilde{Q}_{j+1}, \hat{Q}_{j+1}$ are the unique positive semidefinite solutions of the Lyapunov equations

$$A_j^T \tilde{Q}_{j+1} + \tilde{Q}_{j+1} A_j = -\tilde{W}_j \tilde{W}_j^T, \quad A_j^T \hat{Q}_{j+1} + \hat{Q}_{j+1} A_j = -\hat{W}_j \hat{W}_j^T.$$

Hence, we can apply the factored ADI iteration to both Lyapunov equations in parallel, exploiting that the linear systems of equations to be solved in each ADI iteration step share the same coefficient matrix. The additional expense comes from the fact that now, we need to save two sequences of low-rank factors \tilde{Y}_j, \hat{Y}_j . But if the same (sparse) factorization is used for both linear systems of equations in each ADI iteration, the computational cost is the same as for the LQG case as the number of right-hand sides is the same.

At convergence of the Newton iteration, we obtain

$$Q_{\text{PR}} := \tilde{Q}_{k_{\text{max}}} - \hat{Q}_{k_{\text{max}}} = \tilde{Y}_{k_{\text{max}}, j_{\text{max}}} \tilde{Y}_{k_{\text{max}}, j_{\text{max}}}^T - \hat{Y}_{k_{\text{max}}, j_{\text{max}}} \hat{Y}_{k_{\text{max}}, j_{\text{max}}}^T,$$

where $\tilde{Y}_{k_{\text{max}}, j_{\text{max}}}, \hat{Y}_{k_{\text{max}}, j_{\text{max}}}$ are the two low-rank factors obtained by the two parallel factored (j_{max}) ADI iterations in the last (k_{max}) Newton step. As we know that Q_{PR} is positive semidefinite, we can obtain a factorization $Q_{\text{PR}} = Y_{k_{\text{max}}} Y_{k_{\text{max}}}^T$ similar as in [95, 99] as the full-rank factor of the solution of the Lyapunov equation

$$A^T (Y Y^T) + (Y Y^T) A + W^T W = 0$$

where

$$W = R^{-\frac{1}{2}}C - R^{-\frac{1}{2}}B \begin{bmatrix} \tilde{Y}_{k_{\max},j_{\max}} \\ \hat{Y}_{k_{\max},j_{\max}} \end{bmatrix} \begin{bmatrix} \tilde{Y}_{k_{\max},j_{\max}}^T \\ -\hat{Y}_{k_{\max},j_{\max}}^T \end{bmatrix}.$$

This Lyapunov equation can again be solved using the factored ADI iteration.

Similar considerations lead to factored Newton ADI iterations for solving the first ARE in (24) and the BST ARE (27).

In summary, we have described iterative methods to solve the large-scale Lyapunov and Riccati equations arising in balancing-related MOR methods with a cost mainly proportional to solving linear systems of equations with coefficient matrix A . Although the details of these algorithms are quite involved and their execution is certainly more expensive than just forming Krylov subspaces as in Padé(-type) MOR methods, they can be applied to systems of similar size as all other MOR methods. We provide some numerical evidence for this claim in the following section, see also [20, 22, 37] for further experiments.

5 Numerical Results

The results reported in this section are obtained using either Lyapack³ [83] or the *Sparse model Reduction* subroutine library SpaRed⁴ [10, 11]. Lyapack is a collection of MATLAB functions for solving Lyapunov equations and AREs using variants of the factored ADI and Newton-ADI iterations as described in Section 4. LQGBT, PRBT and BST implementations can be derived based on the available subroutines while BT is even contained in the package. Note that the implementations in Lyapack are not strictly following the descriptions in Section 4 as we have described some variations of the ADI methods that were not available when Lyapack was implemented. SpaRed uses parallel algorithms based on message-passing that can run on any computing platform that provides PBLAS and ScaLAPACK [39]. Sparse linear systems of equations in the ADI iterations can be solved using band matrix solvers provided by ScaLAPACK or sparse direct solvers like SuperLU [41] or MUMPS [1].

Further experiments based on MESS⁵ (Matrix Equations Sparse Solver package) can be found in [37] and will be contained in upcoming publications. MESS can be considered as a successor to Lyapack. Besides many other new features of MESS like simplified argument lists that make the provided MATLAB functions easier to use than the ones in Lyapack, it also implements the ADI iteration for generalized state-space systems (1) with $E \neq I_n$ as described in Section 4, employs column compression based on the RRQR as in [31], and has improved stopping and ADI parameter selection criteria. Altogether, we expect all balancing-related MOR methods to benefit from the improved efficiency of the matrix equation solvers in MESS.

We have discussed BT for large and sparse problems in several publications and provided numerical evidence of the efficiency of these BT implementations there. We will briefly summarize these results here, but will not provide details and in the following, we will show a few examples that have not been reported elsewhere. ADI-based BT using Lyapack in MATLAB for sparse problems is compared to modal truncation and Padé-via-Lanczos methods in [22] where we use two examples from the Oberwolfach benchmark collection⁶ [61]. The numerical results show superiority of BT for a microthruster model using different discretizations (orders of the resulting LTI systems: $n = 4,257$ and $n = 11,445$, leading to reduced-order models of order $r = 21$ and $r = 28$, both with absolute H_∞ error of the transfer function approximation less than 10^{-3}). The same conclusions can be drawn for an optimal cooling problem from [36]. In [22] we report on experiments with BT applied to a finite element discretized model, leading to an LTI system of order $n = 20,209$, $m = 7$, $p = 6$, in the form (13) with E being equal to the mass matrix of the finite element basis. The reduced-order model of order $r = 8$ used there is computed by prescribing a tolerance of 10^{-4} in (21). A finer discretization of the same optimal cooling problem leads to an LTI system with $n = 79,841$. Results computed by the BT implementation in SpaRed applied to this model are discussed in [12], other results obtained with SpaRed can also be found in [10, 11].

In the following, we will report first on the efficiency of the matrix equation solvers which form the bottleneck in all implementations of balancing-related MOR methods. We will also discuss the usage of BT reduced-order models in control design, as well as results obtained with LQGBT. Some experiments with PRBT can be found in [24, 34]. Note that in the figures, often G_r is used instead of \hat{G} for denoting the TFM of the reduced-order system.

Example 1 In our first example, we show the efficiency of the matrix equation solvers discussed in Section 4. For this purpose, we have discretized the linear 2D heat equation with homogeneous Dirichlet boundary conditions and unit heat conductivity on $[0, 1] \times [0, 1]$. The input function $u(t)$ corresponds to a constant heat source in the domain $\Omega_u = [0.2, 0.6] \times [0.2, 0.25]$ and the output is defined by the averaged temperature in $\Omega_{y,1} = [0.2, 0.25] \times [0.2, 0.25]$. We employ a finite differences discretization on a uniform 150×150 grid as provided in the Lyapack testing

³Available from <http://www.slicot.org>.

⁴Available from <http://www.pscm.uji.es/modred/SpaRedW3/SpaRed.html>.

⁵Available from http://www.tu-chemnitz.de/mathematik/industrie_technik/software/mess.php.

⁶Available from <http://www.imtek.de/simulation/benchmark>.

environment. The resulting LTI system is of the form (13) with $D = 0$, $n = 22,500$ and $m = p = 1$. For any ADI iteration, we select 10 shifts using the heuristic provided in Lyapack.

Convergence curves for the factored Newton-ADI iteration applied to the second ARE from (22) are shown in Figure 1 together with a histogram displaying the number of required ADI iterations for each Lyapunov equation solved during the Newton steps. Convergence of the Newton-ADI iteration is measured by relative changes in the feedback gain $F_k := (B^T Y_k) Y_k^T$, where Y_k is the low-rank factor of the current Newton approximate to the ARE solution.

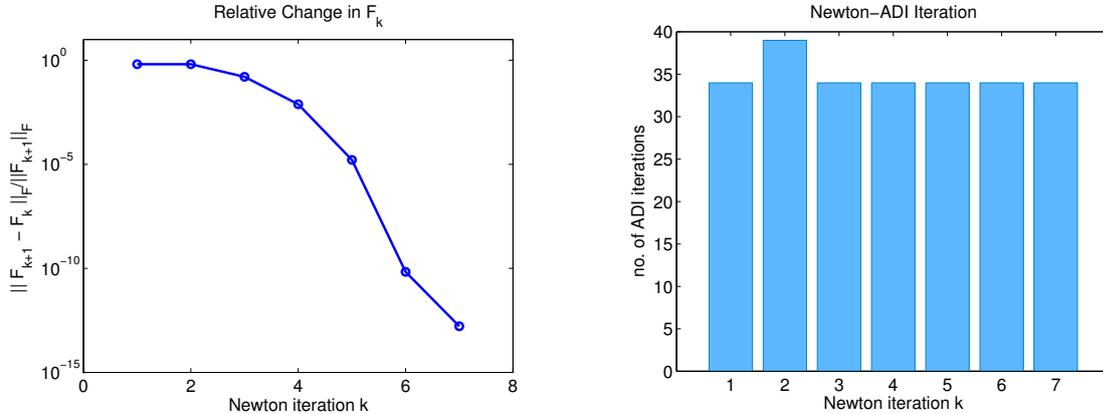


Figure 1: Convergence history of factored Newton-ADI (left) and number of factored ADI iterations needed to solve the Lyapunov equations in the Newton steps (right). The Newton-ADI iteration is applied to the second ARE in (22) corresponding to a control problem of order $n = 22,500$ for a semi-discretized 2D heat equation as described in Example 1.

The discretization of the heat equation with Dirichlet boundary conditions leads to a symmetric negative definite matrix A . In order to show that we can also achieve useful results for nonsymmetric A , we have applied the factored Newton-ADI iteration again to the second ARE from (22) for an LTI system corresponding to the finite differences discretization of a convection-diffusion equation. That is, we have added convection in x_2 -direction to the heat equation used above, i.e., the discretized spatial differential operator becomes $\Delta + \frac{\partial}{\partial x_2}$. The input is chosen as above, as a second output we add the averaged temperature in $\Omega_{y,2} = [0.8, 0.85] \times [0.2, 0.85]$. In Table 1, we show the number of Newton iterations needed for varying meshsizes (uniform grids with $2^k \times 2^k$ grid points, $k = 3, \dots, 7$). The number of unknowns in the full ARE solution matrix Q_{LQG} (exploiting symmetry) is shown in the second column of the table. We terminate the Newton iteration if the residual of the ARE has been reduced to $\approx 1/n$ relative to the initial residual (obtained for $Q_0 = 0$), where n is the number of inner grid points. The obtained normalized residuals are shown together with the number of Newton steps (and maximum number of ADI steps) needed to achieve this in the third and fourth columns of Table 1, while the last column shows the required CPU time for solving the ARE using MATLAB R14SP2 on a Windows XP notebook with Intel Pentium M CPU at 1.1 GHz and 1.25 GB RAM. Obviously, the two largest of the problems considered in the table were not solvable in this computing environment if Q_j were to be formed explicitly. On the other hand, using the factored Newton-ADI iteration makes the solution of such a large task easily accessible using standard hard- and software.

grid	no. of unknowns	$\frac{\ \mathcal{R}(Q)\ _F}{\ Q\ _F}$	it. (ADI it.)	CPU time (sec.)
8×8	2,080	$4.7\text{e-}7$	2 (8)	0.47
16×16	32,896	$1.6\text{e-}6$	2 (10)	0.49
32×32	524,800	$1.8\text{e-}5$	2 (11)	0.91
64×64	8,390,656	$1.8\text{e-}5$	3 (14)	7.98
128×128	134,225,920	$3.7\text{e-}6$	3 (19)	79.46

Table 1: Performance results of the factored Newton-ADI iteration applied to the second ARE in (22) corresponding to a control problem for a semi-discretized 2D convection-diffusion equation with varying meshsizes. The largest problem solved is of order $n = 16,384$.

Example 2 We now test BT for use in control. For this purpose, we employ the same discretized control problem for a 2D heat equation as described in the previous example. Applying the factored ADI iteration to the Lyapunov equations from (20), we obtain low-rank factors $\hat{S}_{BT}, \hat{R}_{BT}$ with 31 and 26 rows, respectively. The computed reduced-order model has order $r = 6$ with $\sigma_7^{BT} = 5.8 \cdot 10^{-4}$, and BT error bound $\delta = 1.7 \cdot 10^{-3}$ (right-hand side of (21)). The frequency responses of the transfer functions G and \hat{G} as well as the pointwise absolute approximation error (Bode magnitude plot) are displayed in Figure 2

In order to show the performance of the reduced-order model for control, we solve the linear-quadratic regulator (LQR) problem (see, e.g., [4]) for the original and reduced-order problems. The solution using the full-order

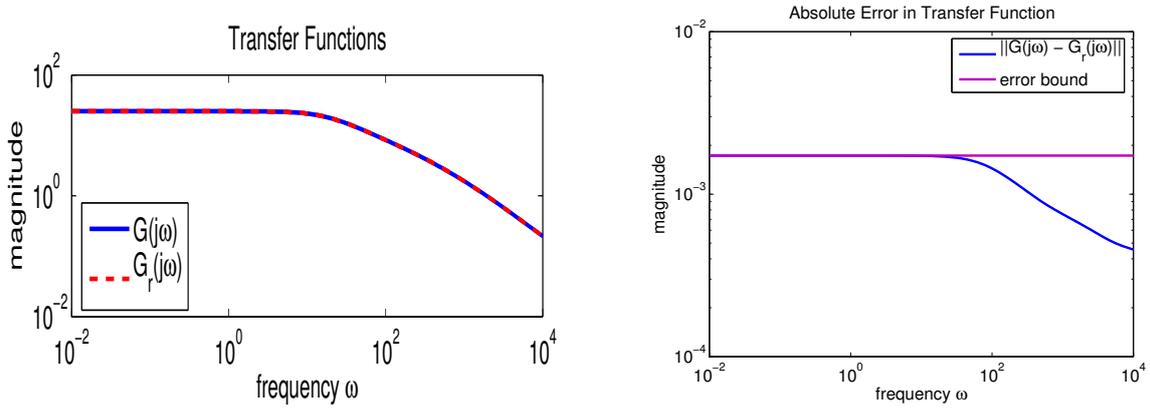


Figure 2: Transfer functions of original and reduced-order models (left) as well as absolute error $|G(i\omega) - \hat{G}(i\omega)|$ (right) for the discretized 2D heat problem from Example 2. Here $n = 22,500$, $m = p = 1$.

system is the state feedback control law

$$u(t) = Fx(t), \quad F = -B^T Q_{LQG} \quad (33)$$

(compare Section 2.2 for definition of controllers), while the reduced-order model yields the following control law:

$$\hat{u}(t) = \hat{F}\hat{x}(t), \quad \hat{F} = -\hat{B}^T \hat{Q}_{LQG}. \quad (34)$$

Here, Q_{LQG}, \hat{Q}_{LQG} are the stabilizing solutions to the second ARE in (22) for the original and reduced-order systems, respectively. Note that the full-order LQR problem can be solved with the help of the factored Newton-ADI iteration as well; see, e.g., [20] for details.

We observe in Figure 3 that there is no visible difference between the original control and the one computed based on the BT reduced-order model and between the resulting outputs. (For the numerical time domain simulation we use the backwards Euler method.)

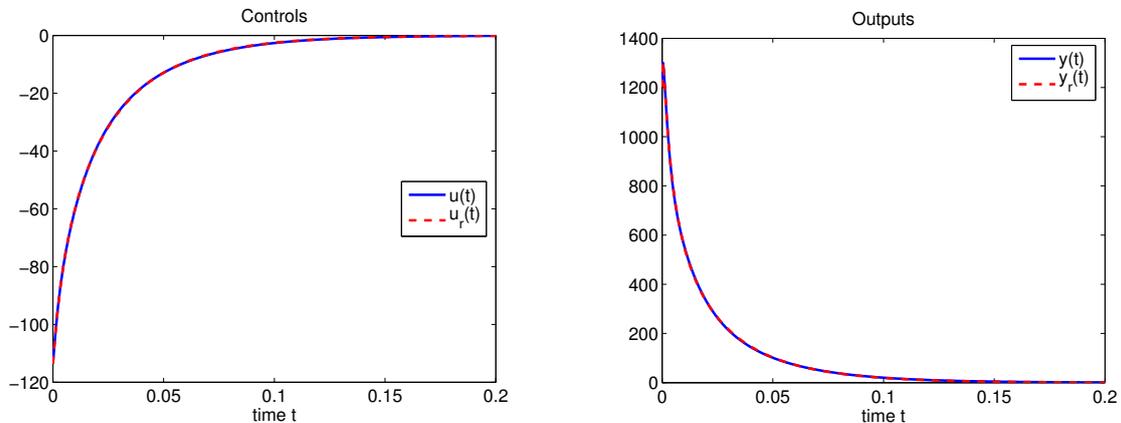


Figure 3: Controls (33), (34) (left) and outputs obtained by these controls (right) when applied to the discretized 2D heat problem from Example 2 and the corresponding BT reduced-order model.

Thus, the control computed from the reduced-order model can be applied without sacrificing the closed-loop performance of the system. The error in the computed controls and outputs is illustrated in Figure 4. Note that the larger errors towards the end of the interval are due to the fact that there, the states and thus outputs and controls are already very close to zero and the backwards Euler discretization on a uniform time grid does not lead to sufficiently small absolute errors to compensate for this.

Example 3 Next, we compare BT and LQGBT for Example HF2D3 from the CONstraint Matrix-optimization Problem library *COMPl_eib*⁷ [66]. The LTI model of the form (13) with order $n = 4,496$ is obtained from a finite differences discretization of a boundary control problem for 2D heat flow in copper on a rectangular domain. The control acts on two sides via Robins boundary conditions and is constant on either side so that $m = 2$. The heat is measured by sensors at 4 locations, hence $p = 4$.

⁷Available from www.compleib.de.

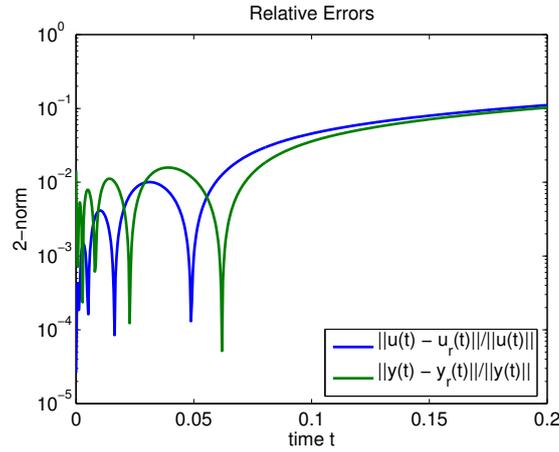


Figure 4: Error in controls (left) and outputs (right) computed by applying the full-order and reduced-order control laws (33) and (34) to the discretized 2D heat problem from Example 2.

We compute the low-rank approximate solution factors of the Lyapunov equation (20) using the factored ADI iteration from Section 4.1 and the corresponding low-rank solution factors of the LQG AREs (22) with the factored Newton-ADI iteration from Section 4.2. The factors $\hat{S}_{BT}, \hat{R}_{BT}$ have 68 and 124 rows, respectively, while $\hat{S}_{LQG}, \hat{R}_{LQG}$ have 210 rows each. In order to compare BT and LQGBT, we compute reduced-order models with $r = 10$ using both approaches.

The frequency responses of the original and reduced-order transfer functions as well as the pointwise absolute approximation errors (Bode magnitude plots) for both reduced-order models are displayed in Figure 5, where the computed error bounds (21) and (23) are shown as straight lines. The error curves are slightly different, but both are below the error bound as expected. This shows that for the example considered, the additional functionality of LQGBT is not traded for worse approximation quality. This observation is valid for a series of other examples tested.

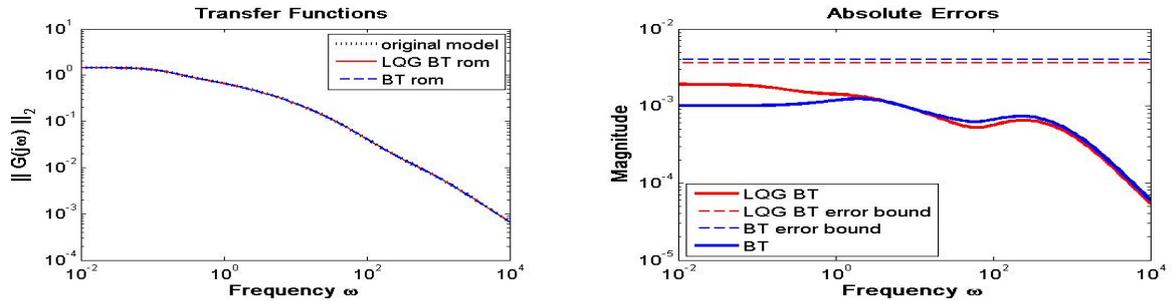


Figure 5: Transfer functions of original and reduced-order models (left) as well as absolute errors (right) for BT and LQGBT applied to the discretized boundary control problem described in Example 3.

6 Conclusions

We have presented model reduction methods based on balancing that can be used in simulation and control of large-scale dynamical systems, as well as for certain inverse problems. The advantage of these methods over other model reduction approaches is that they preserve important system properties like stability, passivity, and minimum phase (depending on the chosen method). Moreover, there are computable error bounds that allow an adaptive choice of the order of the reduced model with respect to a given accuracy tolerance. The methods rely on the availability of efficient methods to solve large-scale matrix equations. Such methods have been developed in the last decade. We have described approaches based on the factored ADI iteration for Lyapunov equations and on the factored Newton-ADI iteration for algebraic Riccati equations. These methods are available in software packages like Lyapack [83] and its successor MESS (Matrix Equations Sparse Solver). Numerical experiments demonstrate the efficiency of the resulting MOR methods. More numerical tests based on MESS, in particular for BST and its application to inverse problems, as well as for other balancing-related BT methods are planned for the near future. Recently, also other methods for large-scale Lyapunov equations have been suggested [26, 58, 88] that may further improve the efficiency of implementations of balancing-related MOR methods. These new methods need further study and will be the focus of future research.

Acknowledgments The work reported in this paper was supported by the German Federal Ministry of Education and Research (BMBF), grant no. 03BEPAE1, the EU project O-MOORE-NICE! funded with the Marie Curie Host Fellowships for the Transfer of Knowledge (ToK) Industry-Academia Partnership Scheme, and *Deutsche Forschungsgemeinschaft*, grants BE 2174/7-1, BE 2174/9-1. Responsibility for the contents of this publication rests with the author.

7 References

- [1] P.R. Amestoy, I.S. Duff, J. Koster, and J.-Y. L'Excellent. MUMPS: a general purpose distributed memory sparse solver. In *Proc. PARA2000, 5th International Workshop on Applied Parallel Computing*, pages 122–131, 2000.
- [2] B.D.O. ANDERSON. An algebraic solution to the spectral factorization problem. *IEEE TRANS. AUTOMAT. CONTROL*, AC-12:410–414, 1967.
- [3] B.D.O. ANDERSON. A system theory criterion for positive real matrices. *SIAM J. CONT.*, 5:171–182, 1967.
- [4] B.D.O. Anderson and J.B. Moore. *Optimal Control – Linear Quadratic Methods*. PRENTICE-HALL, Englewood Cliffs, NJ, 1990.
- [5] B.D.O. Anderson and B. Vongpanitlerd. *Network Analysis and Synthesis. A Modern Systems Approach*. PRENTICE-HALL, Englewood Cliffs, NJ, 1972.
- [6] A.C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM PUBLICATIONS, Philadelphia, PA, 2005.
- [7] A.C. Antoulas and A. Astolfi. H_∞ -norm approximation. In V.D. Blondel and A. Megretski, editors, *2002 MTNS Problem Book, Open Problems on the Mathematical Theory of Networks and Systems*, pages 73–76. 2002. Available online from <http://www.nd.edu/~mtns/OPMTNS.pdf>.
- [8] A.C. Antoulas, D.C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *CONTEMP. MATH.*, 280:193–219, 2001.
- [9] A.C. Antoulas, D.C. Sorensen, and Y. Zhou. On the decay rate of Hankel singular values and related issues. *SYS. CONTROL LETT.*, 46(5):323–342, 2002.
- [10] R.M. Badía, P. Benner, R. Mayo, and E.S. Quintana-Ortí. Parallel algorithms for balanced truncation model reduction of sparse systems. In J.J. Dongarra, K. Madsen, and J. Wasniewski, editors, *Applied Parallel Computing: 7th International Conference, PARA 2004, Lyngby, Denmark, June 20-23, 2004*, number 3732 in Lecture Notes in Computer Science, pages 267–275. Springer-Verlag, Berlin, Heidelberg, New York, 2006.
- [11] R.M. Badía, P. Benner, R. Mayo, E.S. Quintana-Ortí, G. Quintana-Ortí, and A. Remón. Balanced truncation model reduction of large and sparse generalized linear systems. Chemnitz Scientific Computing Preprints 06-04, TU Chemnitz, Fakultät für Mathematik, <http://www.tu-chemnitz.de/mathematik/csc/2006/csc06-04.pdf>, November 2006.
- [12] R.M. Badía, P. Benner, R. Mayo, E.S. Quintana-Ortí, G. Quintana-Ortí, and J. Saak. Parallel order reduction via balanced truncation for optimal cooling of steel profiles. In J.C. Cunha and P.D. Medeiros, editors, *Euro-Par 2005 Parallel Processing: 11th Intl. Euro-Par Conf, Lisbon, Portugal, August 30 – September 2, 2005*, number 3648 in Lecture Notes in Computer Science, pages 857–866. Springer-Verlag, Berlin, Heidelberg, New York, 2005.
- [13] Z. Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Appl. Numer. Math.*, 43(1–2):9–44, 2002.
- [14] S. Barrachina, P. Benner, and E. Quintana-Ortí. Parallel algorithms for balanced truncation of large-scale unstable systems. In *Proc. 44rd IEEE Conf. Decision Contr. and European Contr. Conf. ECC2005*, pages 2248–2253. Omnipress, Madison, WI, 2004. CD Rom.
- [15] R.H. Bartels and G.W. Stewart. Solution of the matrix equation $AX + XB = C$: Algorithm 432. *COMM. ACM*, 15:820–826, 1972.
- [16] U. Baur and P. Benner. Efficient solution of algebraic Bernoulli equations using H-matrix arithmetic. In K. Kunisch, G. Of, and O. Steinbach, editors, *Numerical Mathematics and Advanced Applications (Proc. ENUMATH 2007)*, pages 127–134, Heidelberg, 2008. SPRINGER-VERLAG.
- [17] U. Baur and P. Benner. Gramian-based model reduction for data-sparse systems. *SIAM J. SCI. COMPUT.*, 31(1):776–798, 2008.
- [18] U. Baur, P. Benner, and L. Feng. Mathematical aspects of model order reduction: Some new results and open problems. In preparation.
- [19] P. Benner. Efficient algorithms for large-scale quadratic matrix equations. *Proc. Appl. Math. Mech.*, 1(1):492–495, 2002.
- [20] P. Benner. Solving large-scale control problems. *IEEE Control Systems Magazine*, 14(1):44–59, 2004.
- [21] P. Benner. Passivitätserhaltende Modellreduktion mit Balancierungstechniken. In B. Lohmann, editor, *Tagungsband "Workshop am Bostalsee, 28.–30.9.2005"*, pages 152–168. GMA-Fachausschuss 1.30 "Modellbildung, Identifikation und Simulation in der Automatisierungstechnik", 2005. (In German).
- [22] P. Benner. Numerical linear algebra for model reduction in control and simulation. *GAMM Mitt.*,

- 29(2):275–296, 2006.
- [23] P. Benner. Advances in balancing-related model reduction for circuit simulation. In *Scientific Computing in Electrical Engineering – SCEE 2008*, Mathematics in Industry. SPRINGER-VERLAG, 2009, to appear.
- [24] P. Benner and H. Faßbender. Numerische Methoden zur passivitätserhaltenden Modellreduktion. *at-Automatisierungstechnik*, 54(4):153–160, 2006. (In German).
- [25] P. Benner, J.-R. Li, and T. Penzl. Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems. *NUMER. LIN. ALG. APPL.*, 15(9):755–777, 2008.
- [26] P. Benner, R.-C. Li, and N. Truhar. On the ADI method for Sylvester equations. Preprint, Fakultät für Mathematik, TU Chemnitz, D-09107 Chemnitz, Germany, August 2008.
- [27] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. SLICOT - a subroutine library in systems and control theory. In B.N. Datta, editor, *Applied and Computational Control, Signals, and Circuits*, volume 1, chapter 10, pages 499–539. BIRKHÄUSER, Boston, MA, 1999.
- [28] P. Benner, V. Mehrmann, and D. Sorensen, editors. *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin/Heidelberg, Germany, 2005.
- [29] P. Benner, H. Mena, and J. Saak. On the parameter selection problem in the Newton-ADI iteration for large-scale Riccati equations. *ELECTR. TRANS. NUM. ANAL.*, 29:136–149, 2008.
- [30] P. Benner and E.S. Quintana-Ortí. Model reduction based on spectral projection methods. Chapter 1 (pages 5–48) of [28].
- [31] P. Benner and E.S. Quintana-Ortí. Solving stable generalized Lyapunov equations with the matrix sign function. *NUMER. ALGORITHMS*, 20(1):75–100, 1999.
- [32] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Balanced truncation model reduction of large-scale dense systems on parallel computers. *Math. Comput. Model. Dyn. Syst.*, 6(4):383–405, 2000.
- [33] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Parallel algorithms for model reduction of discrete-time systems. *Int. J. Syst. Sci.*, 34(5):319–333, 2003.
- [34] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Computing passive reduced-order models for circuit simulation. In *Proc. Intl. Conf. Parallel Comp. in Elec. Engrg. PARELEC 2004*, pages 146–151. IEEE Computer Society, Los Alamitos, CA, 2004.
- [35] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Parallel model reduction of large-scale linear descriptor systems via Balanced Truncation. In M. Daydé, J.J. Dongarra, V. Hernández, and J.M.L.M. Palma, editors, *High Performance Computing for Computational Science – VECPAR 2004*, number 3402 in *Lecture Notes in Computer Science*, pages 340–353, Berlin/Heidelberg, 2005. SPRINGER-VERLAG.
- [36] P. Benner and J. Saak. A semi-discretized heat transfer model for optimal cooling of steel profiles. Chapter 19 (pages 353–356) of [28].
- [37] P. Benner and J. Saak. Efficient balancing based MOR for second order systems arising in control of machine tools. In I. Troch and F. Breiteneker, editors, *Proc. MATHMOD 2009, Vienna (Austria), February 11–13, 2009*, volume 35 of *ARGESIM Reports*, 2009.
- [38] J.K. Bennighof and R.B. Lehoucq. An automated multilevel substructuring method for eigenspace computation in linear elastodynamics. *SIAM J. SCI. COMPUT.*, 25(6):2084–2106, 2004.
- [39] L.S. Blackford, J. Choi, A. Cleary, E. D’Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R.C. Whaley. *ScaLAPACK Users’ Guide*. SIAM, Philadelphia, PA, 1997.
- [40] R.F. Curtain. On model reduction for control design for distributed parameter systems. In R. Smith and M. Demetriou, editors, *Research Directions in Distributed Parameter Systems*, pages 95–121, Philadelphia, PA, 2003. SIAM PUBLICATIONS.
- [41] J.W. Demmel, J.R. Gilbert, and X.S. Li. *SuperLU Users’ Guide*.
- [42] U.B. Desai and D. Pal. A transformation approach to stochastic model reduction. *IEEE TRANS. AUTOMAT. CONTROL*, AC–29:1097–1100, 1984.
- [43] R. Freund. Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation. In B.N. Datta, editor, *Applied and Computational Control, Signals, and Circuits*, volume 1, chapter 9, pages 435–498. BIRKHÄUSER, Boston, MA, 1999.
- [44] R. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerica*, 12:267–319, 2003.
- [45] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞ norms. *INTERNAT. J. CONTROL*, 39:1115–1193, 1984.
- [46] K. Glover. Multiplicative approximation of linear multivariable systems with L_∞ error bounds. In *Proc. American Control Conf.*, pages 1705–1709, 1986.
- [47] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.
- [48] L. Grasedyck. Existence of a low rank or H -matrix approximant to the solution of a Sylvester equation. *NUMER. LIN. ALG. APPL.*, 11:371–389, 2004.
- [49] L. Grasedyck and W. Hackbusch. Construction and arithmetics of \mathcal{H} -matrices. *Computing*, 70(4):295–334, 2003.
- [50] M. Green. Balanced stochastic realization. *LINEAR ALGEBRA APPL.*, 98:211–247, 1988.

- [51] M. Green. A relative error bound for balanced stochastic truncation. *IEEE TRANS. AUTOMAT. CONTROL*, AC-33(10):961–965, 1988.
- [52] S. Gugercin and A.C. Antoulas. A survey of model reduction by balanced truncation and some new results. *INTERNAT. J. CONTROL*, 77(8):748–766, 2004.
- [53] S. Gugercin and J.-R. Li. Smith-type methods for balanced truncation of large systems. Chapter 2 (pages 49–82) of [28].
- [54] S. Gugercin, D.C. Sorensen, and A.C. Antoulas. A modified low-rank Smith method for large-scale Lyapunov equations. *Numer. Algorithms*, 32(1):27–55, 2003.
- [55] M. Günther, U. Feldmann, and E.J.W. ter Maten. Modelling and discretization of circuit problems. In W.H.A. Schilders and E.J.W. ter Maten, editors, *Numerical Methods in Electromagnetics*, volume 15 of *Handbook of Numerical Analysis*, pages 523–659. Elsevier, 2005.
- [56] S.J. Hammarling. Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA J. NUMER. ANAL.*, 2:303–323, 1982.
- [57] I.M. Jaimoukha and E.M. Kasenally. Krylov subspace methods for solving large Lyapunov equations. *SIAM J. NUMER. ANAL.*, 31:227–251, 1994.
- [58] K. Jbilou. ADI preconditioned Krylov methods for large Lyapunov matrix equations. Preprint, L.M.P.A, Université du Littoral, F-62228 Calais Cedex, France, November 2008.
- [59] E.A. Jonckheere and L.M. Silverman. A new set of invariants for linear systems—application to reduced order compensator. *IEEE TRANS. AUTOMAT. CONTROL*, AC-28:953–964, 1983.
- [60] D.L. Kleinman. On an iterative technique for Riccati equation computations. *IEEE TRANS. AUTOMAT. CONTROL*, AC-13:114–115, 1968.
- [61] J.G. Korvink and E.B. Rudnyi. Oberwolfach benchmark collection. Chapter 11 (pages 311–315) of [28].
- [62] P. Lancaster and L. Rodman. *The Algebraic Riccati Equation*. Oxford University Press, Oxford, 1995.
- [63] P. Lancaster and M. Tismenetsky. *The Theory of Matrices*. ACADEMIC PRESS, Orlando, 2nd edition, 1985.
- [64] A.J. Laub. Efficient multivariable frequency response computations. *IEEE TRANS. AUTOMAT. CONTROL*, AC-26:407–408, 1981.
- [65] A.J. Laub, M.T. Heath, C.C. Paige, and R.C. Ward. Computation of system balancing transformations and other application of simultaneous diagonalization algorithms. *IEEE TRANS. AUTOMAT. CONTROL*, 34:115–122, 1987.
- [66] F. Leibfritz. *COMPlib: CONstraint Matrix-optimization Problem library – a collection of test examples for nonlinear semidefinite programs, control system design and related problems*. http://www.friedemann-leibfritz.de/COMPlib_Data/COMPlib_Main_Paper.pdf, 2004.
- [67] W.S. Levine, editor. *The Control Handbook*. CRC Press, 1996.
- [68] J.-R. Li and J. White. Reduction of large circuit models via low rank approximate gramians. *Int. J. Appl. Math. Comp. Sci.*, 11(5):1151–1171, 2001.
- [69] J.-R. Li and J. White. Low rank solution of Lyapunov equations. *SIAM J. MATRIX ANAL. APPL.*, 24(1):260–280, 2002.
- [70] The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760, USA. *The MATLAB Control System Toolbox, Version 8.2*, 2008.
- [71] M. Buchholz, T. Lüttich, H. Auracher, and W. Marquardt. Pool boiling at high heat fluxes (part I): Local temperature & heat flux fluctuations at the heater surface. In D. Gorenflo and A. Luke, editors, *Proc. Int. Refrig. Conf. Comm. B1, Paderborn, Germany, Oct. 3–5, 2001*, 2001.
- [72] V. Mehrmann. *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*. Number 163 in Lecture Notes in Control and Information Sciences. SPRINGER-VERLAG, Heidelberg, July 1991.
- [73] V. Mehrmann and T. Stykel. Balanced truncation model reduction for large-scale systems in descriptor form. Chapter 3 (pages 83–115) of [28].
- [74] G.B. Moler and C.F. Van Loan. 19 dubious ways to compute the exponential of a matrix. *SIAM REV.*, 29:801–837, 1978.
- [75] B.C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE TRANS. AUTOMAT. CONTROL*, AC-26:17–32, 1981.
- [76] K. Morris. Design of finite-dimensional controllers for infinite-dimensional systems by approximation. *J. Math. Syst., Estim., and Control*, 4:1–30, 1994.
- [77] A.G.O. Mutambara. *Design and Analysis of Control Systems*. CRC Press, Boca Raton, FL, 1999.
- [78] R. Ober. Balanced parametrizations of classes of linear systems. *SIAM J. CONT. OPTIM.*, 29:1251–1287, 1991.
- [79] G. Obinata and B.D.O. Anderson. *Model Reduction for Control System Design*. Communications and Control Engineering Series. SPRINGER-VERLAG, London, UK, 2001.
- [80] T. Penzl. *Numerische Lösung großer Lyapunov-Gleichungen*. Logos-Verlag, Berlin, Germany, 1998. Dissertation, Fakultät für Mathematik, TU Chemnitz, 1998.
- [81] T. Penzl. A cyclic low rank Smith method for large sparse Lyapunov equations. *SIAM J. SCI. COMPUT.*, 21(4):1401–1418, 2000.

- [82] T. Penzl. Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case. *SYS. CONTROL LETT.*, 40:139–144, 2000.
- [83] T. Penzl. LYAPACK Users Guide. Technical Report SFB393/00-33, Sonderforschungsbereich 393 *Numerische Simulation auf massiv parallelen Rechnern*, TU Chemnitz, 09107 Chemnitz, FRG, 2000. Available from <http://www.tu-chemnitz.de/sfb393/sfb00pr.html>.
- [84] T. Penzl. Algorithms for model reduction of large dynamical systems. *LINEAR ALGEBRA APPL.*, 415(2–3):322–343, 2006. (Reprint of Technical Report SFB393/99-40, TU Chemnitz, 1999.).
- [85] T. Reis and T. Stykel. Positive real and bounded real balancing for model reduction of descriptor systems. Preprint 25-2008, Institute of Mathematics, Technische Universität Berlin, available online at <http://www.math.tu-berlin.de/preprints/abstracts/Report-25-2008.rdf.html>, 2008.
- [86] M.G. Safonov and R.Y. Chiang. Model reduction for robust control: A Schur relative error method. *Int. J. Adapt. Cont. and Sign. Proc.*, 2:259–272, 1988.
- [87] V. Sima. *Algorithms for Linear-Quadratic Optimization*, volume 200 of *Pure and Applied Mathematics*. Marcel Dekker, Inc., New York, NY, 1996.
- [88] V. Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM J. CONT. OPTIM.*, 29(3):1268–1288, 2007.
- [89] E.D. Sontag. *Mathematical Control Theory*. SPRINGER-VERLAG, New York, NY, 2nd edition, 1998.
- [90] D.C. Sorensen and A. C. Antoulas. Dimension reduction of large-scale systems. Chapter 4 (pages 117–130) of [28].
- [91] T. Stykel. Gramian-based model reduction for descriptor systems. *MATH. CONTROL, SIGNALS, SYS.*, 16(4):297–319, 2004.
- [92] M.S. Tombs and I. Postlethwaite. Truncated balanced realization of a stable non-minimal state-space system. *INTERNAT. J. CONTROL*, 46(4):1319–1330, 1987.
- [93] I. Troch, P.C. Müller, and K.-H. Fasol. Modellreduktion für Simulation und Reglerentwurf. *Automatisierungstechnik*, 40(2/3/4):45–53/93–99/132–141, 1992.
- [94] A. Varga. Efficient minimal realization procedure based on balancing. In *Prepr. of the IMACS Symp. on Modelling and Control of Technological Systems*, volume 2, pages 42–47, 1991.
- [95] A. Varga. On computing high accuracy solutions of a class of Riccati equations. *Control–Theory and Advanced Technology*, 10(4):2005–2016, 1995.
- [96] A. Varga. Model reduction routines for SLICOT. NICONET Report 1999–8, The Working Group on Software (WGS), June 1999. Available from www.slicot.org.
- [97] A. Varga. Computing generalized inverse systems using matrix pencil methods. *Int. J. Appl. Math. Comp. Sci.*, 11(5):1055–1068, 2001.
- [98] A. Varga. Model reduction software in the SLICOT library. In B.N. Datta, editor, *Applied and Computational Control, Signals, and Circuits*, volume 629 of *The Kluwer International Series in Engineering and Computer Science*, pages 239–282. Kluwer Academic Publishers, Boston, MA, 2001.
- [99] A. Varga and K. H. Fasol. A new square–root balancing–free stochastic truncation model reduction algorithm. In *Prepr. 12th IFAC World Congress*, volume 7, pages 153–156, Sydney, Australia, 1993.
- [100] E.L. Wachspress. Iterative solution of the Lyapunov matrix equation. *Appl. Math. Letters*, 107:87–90, 1988.
- [101] E.L. Wachspress. The ADI model problem, 1995. Available from the author.
- [102] E.L. Wachspress. ADI iteration parameters for the Sylvester equation, 2000. Available from the author.
- [103] W. Wang and M.G. Safonov. A tighter relative-error bound for balanced stochastic truncation. *SYS. CONTROL LETT.*, 14:307–314, 1990.
- [104] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. PRENTICE-HALL, Upper Saddle River, NJ, 1996.
- [105] K. Zhou, G. Salomon, and E. Wu. Balanced realization and model reduction for unstable systems. *Int. J. Robust Nonlinear Control*, 9(3):183–198, 1999.