Balancing-Related Model Reduction for Large-Scale Systems

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

Abstract

Model reduction is an ubiquitous tool in analysis and simulation of dynamical systems, control design, circuit simulation, structural dynamics, CFD, etc. In the past decades many approaches have been developed for reducing the order of a given model. Here, we will focus on balancing-related model reduction techniques that have been developed since the early 80ies in control theory. The core computations for all the described techniques consist of solving large-scale matrix equations. In the first lecture, we will give special emphasis to new techniques from numerical linear algebra that enable us to solve these large-scale matrix equations and to apply balanced truncation to large-scale systems arising from various application areas.

In the second lecture, specific balancing-related techniques for (optimal) control of parabolic differential equations will be discussed. The methods considered here are based on spatial semi-discretization of the PDE followed by balanced truncation techniques applied to the resulting large-scale system of ordinary differential equations. Different choices of the system Gramians that are used for balancing will be presented. Specifically we will discuss open-loop and closed-loop techniques that allow to preserve system properties important for controller design. Furthermore we will discuss how FEM and model reduction error bounds can be combined to compute a reduced-order model of suitable order.

Keywords: Model reduction; Balanced truncation; Lyapunov equation; Algebraic Riccati equation

AMS Subject Classification: 65F30,93B11,41A20,65F50

Introduction

In our lectures, we will mostly consider linear, time-invariant (LTI) systems of the form

$$\begin{array}{rcll} \dot{x}(t) & = & Ax(t) + Bu(t), & t > 0, \qquad x(0) = x^0, \\ y(t) & = & Cx(t) + Du(t), & t \geq 0, \end{array} \tag{1}$$

*Supported by *Deutsche Forschungsgemeinschaft*, grant BE 2174/7-1.

where $A \in \mathbb{R}^{n \times n}$ is the state matrix, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$, 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:

$$Y(s) = (C(sI - A)^{-1}B + D)U(s),$$

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)*

$$G(s) = C(sI - A)^{-1}B + D. \tag{2}$$

is a real-rational matrix-valued function. Note that any *state-space transformation* $x \mapsto Tx$, $T \in \mathbb{R}^{n \times n}$ nonsingular, yielding a new system description via

$$(A, B, C, D) \mapsto (TAT^{-1}, TB, CT^{-1}, D) \tag{3}$$

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

$$(CT^{-1})(sI - TAT^{-1})^{-1}(TB) + D = C(sI_n - A)^{-1}B + D = G(s).$$

Therefore, there exist infinitely many matrix quadruples (A, B, C, D) 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,

$$\begin{array}{rcllll} \dot{\hat{x}}(t) & = & \hat{A}\hat{x}(t) + \hat{B}u(t), & t > 0, & \hat{x}(0) = \hat{x}^0, \\ \hat{y}(t) & = & \hat{C}\hat{x}(t) + \hat{D}u(t), & t \geq 0, & \end{array} \tag{4}$$

of order r, $r \ll n$, with the same number of inputs m, the same number of outputs p, and associated TFM $\hat{G}(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B} + \hat{D}$, 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

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

the approximation error can be quantified as

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

Here, $\| \,.\, \|$ 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 Z- instead of Laplace transformations; see, e.g., [29, 13].

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 partial differential equations, 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 [1, 3, 9, 5, 18, 29, 35]. 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., [5, 15, 17, 18]. 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 [31, 28, 21]) or in $\mathcal{O}(n \log^2(n))$ (see [6]) if A is approximated by a hierarchical matrix [20]. In our presentations, we will focus on these new techniques.

In the following, we will provide the necessary background material on balanced truncation. We will assume that A from (1) is a stable matrix, i.e., the spectrum of A is contained in the open left half plane. This implies that the system (1) 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.

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 [2]. We focus here on model reduction based on balanced truncation and related methods and how these method can be applied to large-scale problems. The proposed methods attempt to minimize the absolute error $\|G - \hat{G}\|_\infty$. Even though they usually do not lead to a best approximation, fairly tight computable error bounds are available.

Balanced truncation belongs to the class of model reduction methods that rely on truncating state-space transformations defined by means of a nonsingular matrix

$T \in \mathbb{R}^{n\times n}$, so that

$$TAT^{-1} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad TB = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad CT^{-1} = \begin{bmatrix} C_1 & C_2 \end{bmatrix},$$

where $A_{11} \in \mathbb{R}^{r\times r}$, and TB and CT^{-1} are conformably partitioned. With $T = \left[T_l^T, L_l^T\right]^T \in \mathbb{R}^{n\times n}$ 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

$$\hat{A} := T_l A T_r = A_{11}, \quad \hat{B} := T_l B = B_1, \quad \hat{C} := C T_r = C_1, \quad \hat{D} := D. \tag{7}$$

For given r, the problem now is to find T_l, T_r such that $\|G - \hat{G}\|_\infty$ is small.

The most common approach to truncation-based model reduction involves balancing the controllability Gramian W_c and the observability Gramian W_o of the system (1) given as the solutions of the Lyapunov equations

$$AW_c + W_c A^T + BB^T = 0, \qquad A^T W_o + W_o A + C^T C = 0. \tag{8}$$

Since W_c and W_o are positive semidefinite, they can be factored as $W_c = S^T S$ and $W_o = R^T R$. When the factors $S, R \in \mathbb{R}^{n\times n}$ are chosen to be triangular, they are the Cholesky factors of the Gramians.

From a numerical point of view, the observation that a balanced truncation approximation can be achieved using the product SR^T, instead of the product of the Gramians themselves, is a key ingredient of a reliable implementation of balanced model reduction. The resulting *square-root (SR) algorithms* avoid working with the Gramians since their condition number is the square of the condition numbers of the Cholesky factors. In these algorithms, the equations (8) are initially solved for the Cholesky factors without forming the Gramians explicitly. The Cholesky factor computation can be achieved, for example, by Hammarling's method, see [34] and references therein, or an algorithm described in [11]. Then the singular value decomposition (SVD)

$$SR^T = [U_1\ U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}, \quad \begin{array}{rcl} \Sigma_1 & = & \operatorname{diag}(\sigma_1, \dots, \sigma_r), \\ \Sigma_2 & = & \operatorname{diag}(\sigma_{r+1}, \dots, \sigma_n) \end{array} \tag{9}$$

is computed, where

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} \geq \sigma_{r+2} \geq \ldots \geq \sigma_n \geq 0. \tag{10}$$

If $\sigma_r > 0$ and $\sigma_{r+1} = 0$, i.e., $\Sigma_2 = 0$, then r is the McMillan degree of the given LTI system, i.e., $r = \hat{n}$. 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 (7) for the balancing state-space transformation are determined by

$$T_l = \Sigma_1^{-1/2} V_1^T R \quad \text{and} \quad T_r = S^T U_1 \Sigma_1^{-1/2}. \tag{11}$$

It is known that for every choice of r such that $\sigma_r > \sigma_{r+1}$ in (10), using T_l, T_r from (11) in (7) yields a stable, minimal, and balanced reduced model. The Gramians corresponding to the resulting transfer function $\hat{G}(s)$ are both equal to Σ_1. Detailed discussions of balanced truncation and the square-root methods for implementing them can be found in [1, 36].

Serial implementations of balanced truncation algorithms and other balancing-related model reduction techniques are described in the survey article [36]. The described implementations are contained in SLICOT [8] and the associated SLICOT Model Reduction Toolbox for Matlab[1].

[1] See `http://www.slicot.org` for further details.

Though balanced truncation does not generally yield the best rth order approximant of G in the H_∞ norm, we obtain the error bounds

$$\sigma_{r+1} \leq \|G - \hat{G}\|_\infty \leq 2 \sum_{k=r+1}^{n} \sigma_k. \qquad (12)$$

This a priori error bound makes balanced truncation 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 balanced truncation to large-scale models. However, Schur vector solutions of the Lyapunov equation require $\mathcal{O}(n^3)$ flops and $\mathcal{O}(n^2)$ workspace. Even if these requirements could be reduced, the SVD in (9) requires $\mathcal{O}(n^3)$ flops and $\mathcal{O}(n^2)$ workspace. 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 and parallel model reduction 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 $\operatorname{rank}(S) < n$ and/or $\operatorname{rank}(R) < n$. Hence, rather than working with the singular Cholesky factors, we may use *full-rank factors* of W_c, W_o. Since W_c and W_o 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 $W_c = \hat{S}^T\hat{S}$, $W_o = \hat{R}^T\hat{R}$, and

$$n_c := \operatorname{rank}(\hat{S}) = \operatorname{rank}(S) = \operatorname{rank}(W_c), \quad n_o := \operatorname{rank}(\hat{R}) = \operatorname{rank}(R) = \operatorname{rank}(W_o).$$

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 (9) 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 [19], (9) 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 W_c, W_o and $\hat{S}$, $\hat{R}$ is often much smaller than n; see [32, 12]. This forms the basis for the balancing-related model reduction methods described in the two lectures.

The contents of the two lectures will be summarized below, for further information on efficient balanced truncation methods see [4, 6, 7, 10, 21, 27, 28, 30, 31].

1. Balancing-related model reduction methods for large-scale systems

After reviewing some aspects of system theory necessary for the further understanding of both lectures, we will review the basics of balanced truncation as summarized above. Then we will focus on methods for solving the Lyapunov equation (8) that compute the low-rank factors $\hat{S}, \hat{R}$ directly.

The first approach discussed is based on spectral projection methods, in particular on the sign function method which allows to compute skew projectors onto the stable and anti-stable invariant subspaces of a matrix $Z \in \mathbb{R}^{n \times n}$, i.e., onto the subspaces corresponding to all eigenvalues in the open left and right, respectively, half complex planes. Let Z have no eigenvalues on the imaginary axis and $Z = S \begin{bmatrix} J^- & 0 \\ 0 & J^+ \end{bmatrix} S^{-1}$ be its Jordan decomposition, where the Jordan blocks in $J^- \in \mathbb{R}^{k \times k}$ and $J^+ \in \mathbb{R}^{(n-k)\times(n-k)}$ contain, respectively, the stable and unstable parts of the spectrum of Z. The *matrix sign function* of Z is defined as $\operatorname{sign}(Z) := S \begin{bmatrix} -I_k & 0 \\ 0 & I_{n-k} \end{bmatrix} S^{-1}$. Note that $\operatorname{sign}(Z)$ is unique and independent of the order of the eigenvalues in the Jordan decomposition of Z, see, e.g., [25]. Many other definitions of the sign function can be given; see [23] for an overview. Some important properties of the matrix sign function are summarized in the following proposition.

Proposition 1 *Let $Z \in \mathbb{R}^{n\times n}$ have no purely imaginary eigenvalues. Then:*

a) $(\operatorname{sign}(Z))^2 = I_n$*, i.e.,* $\operatorname{sign}(Z)$ *is a square root of the identity matrix;*

b) $\operatorname{sign}\left(T^{-1}ZT\right) = T^{-1}\operatorname{sign}(Z)\,T$ *for all nonsingular* $T \in \mathbb{R}^{n\times n}$*;*

c) $\operatorname{sign}\left(Z^T\right) = \operatorname{sign}(Z)^T$.

d) Let p_+ and p_- be the numbers of eigenvalues of Z with positive and negative real part, respectively. Then

$$p_+ = \frac{1}{2}(n + \operatorname{tr}(\operatorname{sign}(Z))), \quad p_- = \frac{1}{2}(n - \operatorname{tr}(\operatorname{sign}(Z))).$$

e) Let Z be stable, then $\operatorname{sign}(Z) = -I_n$*,* $\operatorname{sign}(-Z) = I_n$*.*

Applying Newton's root-finding iteration to $Z^2 = I_n$, where the starting point is chosen as Z, we obtain the Newton iteration for the matrix sign function:

$$Z_0 \leftarrow Z, \quad Z_{j+1} \leftarrow \frac{1}{2}(\frac{1}{\gamma_j}Z_j + \gamma_j Z_j^{-1}), \quad j = 0, 1, 2, \ldots, \tag{13}$$

where γ_j is an acceleration parameter. Under the given assumptions, the sequence $\{Z_j\}_{j=0}^{\infty}$ converges with an ultimately quadratic convergence rate and

$$\operatorname{sign}(Z) = \lim_{j\to\infty} Z_j;$$

see [33]. It is easy to see that

$$P_- := \frac{1}{2}(I_n - \operatorname{sign}(Z)), \tag{14}$$

is a spectral projector onto the stable Z-invariant subspace and $P_+ := (I_n + \operatorname{sign}(Z))/2$ is a spectral projector onto the Z-invariant subspace corresponding to the eigenvalues in the open right half plane. Now consider the Lyapunov equation

$$AX + XA^T + W = 0, \quad A, W \in \mathbb{R}^{n\times n}, \tag{15}$$

where A is stable. The latter assumption is equivalent to (15) having a unique solution [26]. Let $X \in \mathbb{R}^{n\times m}$ be this unique solution. Then the straightforward calculation

$$\begin{bmatrix} I_n & 0 \\ X & I_m \end{bmatrix}\begin{bmatrix} A & 0 \\ W & -A^T \end{bmatrix}\begin{bmatrix} I_n & 0 \\ -X & I_m \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & -A^T \end{bmatrix} \tag{16}$$

reveals that the columns of $\left[\begin{smallmatrix} I_n \\ -X_* \end{smallmatrix}\right]$ span the stable invariant subspace of $Z := \left[\begin{smallmatrix} A & 0 \\ W & -A^T \end{smallmatrix}\right]$. In principle, this subspace, and after an appropriate change of basis, also the solution matrix X, can be computed from a spectral projector onto this Z-invariant subspace. The sign function is an appropriate tool for this as P_- from (14) is the required spectral projector. A closer inspection of (13) applied to Z shows that we do not even have to form P_- in this case, as the solution can be directly read off the matrix $\operatorname{sign}(Z)$: Using (16) and Proposition 1 reveals that

$$\operatorname{sign}(Z) = \operatorname{sign}\left(\begin{bmatrix} A & 0 \\ W & -A^T \end{bmatrix}\right) = \begin{bmatrix} -I_n & 0 \\ 2X & I_m \end{bmatrix}$$

so that the solution of (15) is given as the lower left block of the limit of (13), divided by 2. Moreover, the block-triangular structure of Z allows to decouple (13) as

$$\begin{array}{l}
A_0 \quad \leftarrow \quad A, \quad W_0 \quad \leftarrow \quad W, \\
\texttt{for } j = 0, 1, 2, \ldots \\
\qquad A_{j+1} \;\leftarrow\; \dfrac{1}{2\gamma_j}\left(A_j + \gamma_j^2 A_j^{-1}\right), \\
\qquad W_{j+1} \;\leftarrow\; \dfrac{1}{2\gamma_j}\left(W_j + \gamma_j^2 A_j^{-1} W_j A_j^{-T}\right).
\end{array} \tag{17}$$

with $X_* = \frac{1}{2}\lim_{j\to\infty} W_j$. We will then show how this iteration can be modified to compute a low-rank factor of X directly. This iteration, being significantly less expensive than the original one, is still $\mathcal{O}(n^3)$ due to the A_j-iteration. But as it is composed of basic linear algebra operations only, it parallelizes well so that the resulting method can be used to solve fairly large Lyapunov equations on parallel computers, see [12, 14].

As the sign function-based approach does not respect any sparsity structure of A, we will in the last part of the first lecture describe an approach for large-scale and sparse Lyapunov equations based on the ADI iteration. For solving Lyapunov equations (15), the ADI iteration can be written as follows [37]:

$$\begin{aligned}
(A + p_j I) X_{(j-1)/2} &= -W - X_{j-1}(A^T - p_j I), \\
X_j (A^T + \overline{p_j} I) &= -W - (A - \overline{p_j} I) X_{(j-1)/2}.
\end{aligned}$$

If the shift parameters p_j are chosen appropriately, then $\lim_{j\to\infty} X_j = X$ with $X_0 = 0$. Again, we will describe how this iteration can be modified to yield an (approximate) low-rank solution factor directly.

Note that with both the described approaches for solving Lyapunov equations, we can implement an efficient version of balanced truncation based on the small-size SVD of $\hat{S}\hat{R}^T$ as described above.

2. Control-oriented model reduction for parabolic control systems

In the second lecture we will discuss the application of balancing-related model reduction to (optimal) control problems for parabolic partial differential equations (PDEs). The methods considered here are based on spatial semi-discretization of the PDE followed by balanced truncation techniques applied to the resulting large-scale system of ordinary differential equations. Obviously, the methods described above apply to resulting linear systems. As the semi-discretization of the parabolic PDE leads to a system (1) with an A matrix that can be interpreted as the negative of a discretized elliptic differential operator, we can use the hierarchical matrix format [20] to store it. This allows to re-write (17) in the formatted arithmetic for hierarchical matrices so that the overall complexity comes down to $\mathcal{O}(n\log^2(n))$, see [6]. Thus, we obtain a specific implementation of the sign function-based balanced truncation method for parabolic control systems.

A disadvantage of balanced truncation in the context of infinite-dimensional systems such as the considered control of parabolic PDEs is that controllers based on the the reduced-order model may not be robust when applied to the original infinite-dimensional problem. Therefore, in [16] it is suggested to use instead of balanced truncation a technique called LQG balancing [22]. It is then proved in [16] that a robust controller can be based on reduced-order model computed by the truncated LQG balanced system. The basic idea of LQG balanced truncation is to replace the Gramians W_c and W_o from (8) by the stabilizing solutions of the dual algebraic Riccati equations (AREs)

$$\begin{aligned}
0 &= AP + PA^T - PC^T CP + B^T B, \\
0 &= A^T Q + QA - QBB^T Q + C^T C,
\end{aligned}$$

related to the regulator and filter AREs used in linear-quadratic Gaussian (LQG) control design. Again, P and Q are positive semidefinite and can often be approximated by low-rank factorizations $\hat{S}\hat{S}^T$ and $\hat{R}\hat{R}^T$. We will show that Newton's method applied to the above AREs (as proposed in [24]) leads to a sequence of approximations P_j and Q_j that are computed as solutions of Lyapunov equations. Any of the methods for Lyapunov equations described above can be applied here, resulting in an efficient way to compute the approximate low-rank factors of P and Q directly. Hence we arrive at implementations of LQG balanced truncation that can be applied to the very large-scale systems resulting from semi-discretized parabolic control problems.

In the last part of our lecture, we will discuss how FEM and model reduction error bounds available for the balancing-related methods can be combined to compute a reduced-order model of suitable order. Several numerical examples will be used to demonstrate the proposed model reduction techniques. Furthermore, we give an interpretation of (LQG) balanced truncation as a Petrov-Galerkin projection method and show how the computed truncation operators T_l and T_r in (11) can be used to define ansatz functions for the Galerkin projection. Some relations to POD (proper orthogonal decomposition) will also be discussed.

References

[1] A. Antoulas. *Approximation of Large-Scale Dynamical Systems.* SIAM Publications, Philadelphia, PA, 2005.

[2] A. Antoulas and A. Astolfi. H_∞-norm approximation. In V. 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`.

[3] A. Antoulas, D. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemp. Math.*, 280:193–219, 2001.

[4] R. Badía, P. Benner, R. Mayo, and E. Quintana-Ortí. Parallel algorithms for balanced truncation model reduction of sparse systems. In J. Dongarra, K. Madsen, and J. Wasniewski, editors, *Applied Parallel Computing: 7th International Conference, PARA 2004, Lyngby, Denmark, June 20-23, 2004. Revised Selected Papers*, Lecture Notes in Computer Science, volume 3732, pages 267–275. Springer-Verlag, 2006.

[5] Z. Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Appl. Numer. Math*, 43(1–2):9–44, 2002.

[6] U. Baur and P. Benner. Factorized solution of the Lyapunov equation by using the hierarchical matrix arithmetic. *Proc. Appl. Math. Mech.*, 4(1):658–659, 2004.

[7] P. Benner. Solving large-scale control problems. *IEEE Control Systems Magazine*, 14(1):44–59, 2004.

[8] P. Benner, V. Mehrmann, V. Sima, S. V. Huffel, and A. Varga. SLICOT - a subroutine library in systems and control theory. In B. Datta, editor, *Applied and Computational Control, Signals, and Circuits*, volume 1, chapter 10, pages 499–539. Birkhäuser, Boston, MA, 1999.

[9] 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.

[10] P. Benner and E. Quintana-Ortí. Model reduction based on spectral projection methods. Chapter 1 (pages 5–48) of [9].

[11] P. Benner and E. Quintana-Ortí. Solving stable generalized Lyapunov equations with the matrix sign function. *Numer. Algorithms*, 20(1):75–100, 1999.

[12] P. Benner, E. 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.

[13] P. Benner, E. Quintana-Ortí, and G. Quintana-Ortí. Parallel algorithms for model reduction of discrete-time systems. *Int. J. Syst. Sci.*, 34(5):319–333, 2003.

[14] P. Benner, E. Quintana-Ortí, and G. Quintana-Ortí. State-space truncation methods for parallel model reduction of large-scale systems. *Parallel Comput.*, 29:1701–1722, 2003.

[15] J. Bennighof and R. Lehoucq. An automated multilevel substructuring method for eigenspace computation in linear elastodynamics. *SIAM J. Sci. Comput.*, 25(6):2084–2106, 2004.

[16] R.F. Curtain. On model reduction for control design for distributed parameter systems. In R. Smith and M. Demetriou, editors, *Research Directions in Distributer Parameter Systems*, pages 95–121, SIAM, Philadelphia, 2003.

[17] R. Freund. Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation. In B. Datta, editor, *Applied and Computational Control, Signals, and Circuits*, volume 1, chapter 9, pages 435–498. Birkhäuser, Boston, MA, 1999.

[18] R. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerica*, 12:267–319, 2003.

[19] G. Golub and C. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, third edition, 1996.

[20] L. Grasedyck and W. Hackbusch. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70(4):295–334, 2003.

[21] S. Gugercin and J.-R. Li. Smith-type methods for balanced truncation of large systems. Chapter 2 (pages 49–82) of [9].

[22] E. Jonckheere and L. Silverman. A new set of invariants for linear systems—application to reduced order compensator. *IEEE Trans. Automat. Control*, AC-28:953–964, 1983.

[23] C. Kenney and A. Laub. The matrix sign function. *IEEE Trans. Automat. Control*, 40(8):1330–1348, 1995.

[24] D. Kleinman. On an iterative technique for Riccati equation computations. *IEEE Trans. Automat. Control*, AC-13:114–115, 1968.

[25] P. Lancaster and L. Rodman. *The Algebraic Riccati Equation.* Oxford University Press, Oxford, 1995.

[26] P. Lancaster and M. Tismenetsky. *The Theory of Matrices.* Academic Press, Orlando, 2nd edition, 1985.

[27] 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.

[28] J.-R. Li and J. White. Low rank solution of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 24(1):260–280, 2002.

[29] G. Obinata and B. Anderson. *Model Reduction for Control System Design.* Communications and Control Engineering Series. Springer-Verlag, London, UK, 2001.

[30] T. Penzl. Algorithms for model reduction of large dynamical systems. Technical Report SFB393/99-40, Sonderforschungsbereich 393 *Numerische Simulation auf massiv parallelen Rechnern*, TU Chemnitz, 09107 Chemnitz, FRG, 1999. Available from `http://www.tu-chemnitz.de/sfb393/sfb99pr.html`.

[31] T. Penzl. A cyclic low rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418, 2000.

[32] T. Penzl. Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case. *Sys. Control Lett.*, 40:139–144, 2000.

[33] J. Roberts. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *Internat. J. Control*, 32:677–687, 1980. (Reprint of Technical Report No. TR-13, CUED/B-Control, Cambridge University, Engineering Department, 1971).

[34] V. Sima. *Algorithms for Linear-Quadratic Optimization*, volume 200 of *Pure and Applied Mathematics.* Marcel Dekker, Inc., New York, NY, 1996.

[35] I. Troch, P. Müller, and K.-H. Fasol. Modellreduktion für Simulation und Reglerentwurf. *Automatisierungstechnik*, 40(2/3/4):45–53/93–99/132–141, 1992.

[36] A. Varga. Model reduction software in the SLICOT library. In B. 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.

[37] E. Wachspress. Iterative solution of the Lyapunov matrix equation. *Appl. Math. Letters*, 107:87–90, 1988.