Numerical Linear Algebra for Model Reduction in Control and Simulation

Peter Benner[*1]

[1] Fakultät für Mathematik, Technische Universität Chemnitz, D-09107 Chemnitz, Germany.

Received 28 June 2005, revised 26 July 2005, accepted 26 July 2005
Published online 27 July 2005

Key words model reduction, modal truncation, Padé approximation, balanced truncation, Lyapunov equation, Krylov subspace method.
MSC (2000) 65F30,93B11,41A20,65F50

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. Often these methods have been derived in parallel in different disciplines with particular applications in mind. We will discuss some of the most prominent methods used for linear systems, compare their properties and highlight similarities. In particular, we will emphasize the role of recent developments in numerical linear algebra in the different approaches. Efficiently using these new techniques, the range of applicability of some of the methods has considerably widened.

1 Introduction

Consider the linear, time-invariant (LTI) system

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

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 be 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.

* e-mail: benner@mathematik.tu-chemnitz.de, Phone: +49 371 531 8367, Fax: +49 371 531 2657

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. The associated transfer function matrix (TFM) then is

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

Note that any *state-space transformation* $x \mapsto Tx$, $T \in \mathbb{R}^{n\times n}$ nonsingular, 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 throughout this paper, we will use several specific realizations of LTI systems.

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

$$\begin{array}{rcll} \dot{\hat{x}}(t) & = & \hat{A}\hat{x}(t) + \hat{B}\hat{u}(t), & t > 0 \qquad \hat{x}(0) = \hat{x}^0, \\ \hat{y}(t) & = & \hat{C}\hat{x}(t) + \hat{D}\hat{u}(t), & t \geq 0, \end{array} \tag{3}$$

of order r, $r \ll n$, and associated TFM $\hat{G}(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B} + \hat{D}$, so that for the same input function $u(t) \in L_2([0, \infty), \mathbb{R}^m)$, we have $y(t) \approx \hat{y}(t)$. Model reduction of discrete-time LTI systems can be formulated in an analogous manner; see, e.g., [55].

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 even application-specific even though they share many common features and origins. See the recent monographs and surveys [2, 3, 11, 6, 28, 55, 65]. In case of linear systems, it seems that three approaches play the most prominent role, these are

1. modal truncation and the related techniques of substructuring and static condensation,
2. Padé and Padé-type approximations, and
3. 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 millions. We will discuss the basic principles of all three approaches and the related techniques from numerical linear algebra. It is well-known that the first two approaches listed above can be applied to very large-scale problems, see, e.g., [6, 17, 26, 28]. In contrast to common belief, it is also possible to apply balanced truncation techniques for large-scale problems. As this is based on relatively new developments, we will spend a little more effort to explain the underlying new techniques from numerical linear algebra. Further information on these developments can be found in [5, 9, 10, 12, 40, 46, 59, 58].

The rest of this paper is structured as follows: in Sections 2, 3, and 4, we review the three approaches listed above, i.e., model truncation, Padé approximation, and balanced truncation. A comparison of the three concepts from a theoretical point of view is given in Section 5 while Section 6 provides several numerical examples for the application of the different approaches to some benchmark problems from several application areas. Moreover, a fairly extensive bibliography is provided which can be used as a basis for further reading.

2 Modal Approaches

Modal methods are based on catching the behavior of the dominant modes of the system in the reduced-order model. In the language of linear algebra, this means that the state-space is projected on the subspace spanned by the eigenvectors ("eigenforms") corresponding to specified eigenvalues of the state matrix A (or poles of $G(s)$). Modal truncation is probably one of the oldest model reduction techniques [21, 24, 49] and still in use, mainly in structural dynamics. It is for instance available in the popular software packages ANSYS [1] or Nastran [52]. The eigenvalues and -vectors can be computed by subspace iteration methods which is often used in structural dynamics packages. Usually, more efficient algorithms for large-scale problems result from utilizing modern methods of sparse numerical linear algebra like Krylov subspace method (Lanczos or Arnoldi iterations) of the Jacobi-Davidson algorithm [7, 17, 45, 66].

For state matrices A that are not unitarily diagonalizable, the eigenvector basis may be ill-conditioned. Therefore, it is advisable to use an orthogonal basis for the A-invariant subspace spanned by the desired eigenvectors.

An obvious, though certainly not always optimal choice of dominant modes is to select those eigenvalues of A having nonnegative or small negative real values. Basically, these eigenvalues dominate the long-term dynamics of the solution of the linear ordinary differential equation describing the dynamics of (1)—solution components corresponding to large negative real parts decay rapidly and mostly play a less important (negligible) role in vibration analysis or control design. This viewpoint is rather naive as it neither takes into account the transient behavior of the dynamical system nor the oscillations caused by large imaginary parts or the sensitivity of the eigenvalues with respect to small perturbations. Nevertheless, this approach is often successful when A comes from an FEM analysis of an elliptic operator as the ones arising in linear elasticity or heat transfer processes. Here, a large range of eigenvalues are insignificant as they result only from discretizing the differential operator and approximate a continuous part of its spectrum. The corresponding modes can safely be neglected as they have no impact on the dynamic response of the system.

An advantage of modal truncation is that the poles of the reduced-order system are poles of the original system. This is important in applications like vibration analysis as the modes correspond to the resonance frequencies of the original system; the most important resonances are then retained in the reduced-order model. On the other hand, the exact location of eigenvalues plays a less important role in the analysis of dynamical systems.

The error of modal truncation can easily be quantisized. The case of a diagonal matrix A is treated in [36, Lemma 9.2.1] which is trivially generalized to the following result.

Theorem 2.1 *Suppose A in (1) is stable and unitarily diagonalizable as*

$$A = U \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix} U^H,$$

where $U \in \mathbb{C}^{n\times n}$ is unitary and $D_1 \in \mathbb{C}^{r\times r}$, $D_2 \in \mathbb{C}^{n-r\times n-r}$ are diagonal matrices with D_1 containing the desired eigenvalues $\lambda_1, \ldots, \lambda_r$. If $U =: \begin{bmatrix} U_1 & U_2 \end{bmatrix}$ is partitioned accordingly and the reduced-order model (3) is computed as

$$\hat{A} = U_1^T A U_1, \quad \hat{B} = U_1^T B, \quad \hat{C} = C U_1, \quad \hat{D} = D,$$

then the reduced-order model satisfies

$$\|G - \hat{G}\|_\infty \leq \|C_2\|_2 \|B_2\|_2 \max_{\lambda\in\{\lambda_{r+1},\ldots,\lambda_n\}} \frac{1}{|\mathrm{Re}(\lambda)|}, \tag{4}$$

where $B_2 = U_2^T B$, $C_2 = C U_2$.

P r o o f. With $G(s) = CU(sI - U^H A U)^{-1} U^H B + D$ and the construction of the reduced-order model it follows immediately that

$$G(s) - \hat{G}(s) = C_2 (sI - D_2)^{-1} B_2.$$

Now, as A is assumed to be stable, we have no purely imaginary poles so that

$$\begin{aligned} \|G - \hat{G}\|_\infty &= \max_{\omega\in\mathbb{R}} \|C_2(\jmath\omega I - D_2)^{-1} B_2\|_2 \\ &\leq \|B_2\|_2 \|C_2\|_2 \max_{\omega\in\mathbb{R}} \|(\jmath\omega I - D_2)^{-1}\|_2 \\ &= \|B_2\|_2 \|C_2\|_2 \max_{\omega\in\mathbb{R}} \frac{1}{\min_{r+1\leq j\leq n} |\jmath\omega - \lambda_j|}, \end{aligned}$$

where the 2-norms in the right-hand side are the matrix 2-norms (or spectral norms). The error bound now follows by observing that the maximum of the right-hand side is taken for the eigenvalue closest to the imaginary axis. □

The error bound (4) is computable if the full matrix U (or all eigenvectors of A) is (are) available. Otherwise, the following bound can be used if the eigenvalues in D_1 are selected as the eigenvalues of A with smallest real parts and ordered as

$$|\mathrm{Re}(\lambda_1)| \leq |\mathrm{Re}(\lambda_2)| \leq \ldots \leq |\mathrm{Re}(\lambda_r)| \leq |\mathrm{Re}(\lambda_{r+\ell})| \ \forall \ell = 1, \ldots, n-r.$$

By noting that $\|B_2\|_2 \leq \|B\|_2$, $\|C_2\|_2 \leq \|C\|_2$, and $|\mathrm{Re}(\lambda_j)| \geq |\mathrm{Re}(\lambda_r)|$, we get

$$\|G - \hat{G}\|_\infty \leq \|C\|_2 \|B\|_2 \frac{1}{|\mathrm{Re}(\lambda_r)|}. \tag{5}$$

In case A is not unitarily diagonalizable, the situation becomes slightly more complicated. If A is diagonizable by a non-unitary similarity transformation with $T \in \mathbb{C}^{n\times n}$ nonsingular, then the same argument as in the proof of Theorem 2.1 yields

$$\|G - \hat{G}\|_\infty \leq \mathrm{cond}_2(T) \, \|C_2\|_2 \|B_2\|_2 \max_{\lambda \in \{\lambda_{r+1},\ldots,\lambda_n\}} \frac{1}{|\mathrm{Re}(\lambda)|}, \tag{6}$$

where $T^{-1}AT = D$ is the spectral decomposition of A and $\mathrm{cond}_2(T)$ is the spectral norm condition number of the eigenvector matrix T.

This brief error analysis shows that minimization of the error bounds for modal truncation corresponds to keeping the eigenvalues with small imaginary parts in the reduced-order model which is in full accordance with the engineering significance of the corresponding modes.

In order to better approximate the static behavior of a system, it is often advised to include static solutions in the reduced-order model. The static response of the LTI system (1) corresponds to setting $\dot{x}(t) = 0$. If A is invertible, we therefore get

$$x(t) = -A^{-1}Bu(t).$$

In the model reduction process this can be taken into account by *base enrichment*. One possibility is to simply add $\tilde{U} := -A^{-1}B$ to U_1 so that the reduced-order model becomes

$$\hat{A} = V^T A V, \quad \hat{B} = V^T B, \quad \hat{C} = CV, \quad \hat{D} = D,$$

where $V = \begin{bmatrix} U_1 & \tilde{U} \end{bmatrix}$. Somewhat more sophisticated combinations of static reduction methods (also called *Guyan reduction* [41]) and modal truncation lead to the Craig-Bampton method [21] which is the basis of modern *substructuring* and *component mode synthesis* methods, see, e.g., [17, 62] and the references therein.

3 Padé Approximation

In this section we consider systems of the form

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) \tag{7}$$

with corresponding transfer function $G(s) = C(sE - A)^{-1}B$. This formulation slightly generalizes (1) in allowing an non-identity matrix $E \in \mathbb{R}^{n\times n}$ in front of the derivative $\dot{x}$ which is not necessarily invertible. (Note that modal truncation as well as balanced truncation can easily be adapted to this situation, too.) On the other hand, we assume $D = 0$. The latter is not a restriction as otherwise, the system can be re-written in the form (1) with a slightly enlarged generalized state-space, see, e.g., [23].

For $s_0 \notin \Lambda(A, E)$, we can re-write the transfer function of (7) and expand it into a power (Laurent) series as follows:

$$\begin{aligned} G(s) &= C((s_0E - A) + (s - s_0)E)^{-1}B \\ &= C\left(I - (s - s_0)(s_0E - A)^{-1}E\right)^{-1}(s_0E - A)^{-1}B \\ &= M_0 + M_1(s - s_0) + M_2(s - s_0)^2 + \ldots \end{aligned}$$

For $s_0 = 0$, the coefficient matrices in the power series are given by $M_j := C(A^{-1}E)^j A^{-1}B$ and are called *moments*. In case $s_0 = \infty$ we have $M_j := C(E^{-1}A)^j E^{-1}B$, called the *Markov parameters* of the system.

As reduced-order model we can now use the rth Padé approximate $\hat{G}$ to G, defined by:

$$G(s) = \hat{G}(s) + \mathcal{O}((s - s_0)^{2r}),$$

i.e., $M_j = \widehat{M}_j$ for $j = 0, \ldots, 2r-1$. The corresponding computational problem is called *moment matching* if $s_0 < \infty$ and *partial realization* if $s_0 = \infty$.

For $m = p = 1$, the solution of the moment matching problem is given by

$$\hat{G}(s) = \frac{\alpha_{r-1}s^{r-1} + \alpha_{r-2}s^{r-2} + \ldots + \alpha_1 s + \alpha_0}{\beta_r s^r + \beta_{r-1}s^{r-1} + \ldots + \beta_1 s + 1},$$

where the coefficients of the enumerator and denominator polynomials can be computed via the so-called *Asymptotic Waveform Evaluation (AWE)* procedure: solve the linear system of equations

$$M \begin{bmatrix} \beta_r \\ \vdots \\ \beta_1 \end{bmatrix} = \begin{bmatrix} m_r \\ \vdots \\ m_{2r-1} \end{bmatrix},$$

with the *Hankel matrix*

$$M = \begin{bmatrix} m_0 & m_1 & m_2 & \ldots & m_{r-1} \\ m_1 & m_2 & & \iddots & m_r \\ m_2 & & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & \iddots & \\ m_{r-1} & m_r & \ldots & & m_{2r-2} \end{bmatrix}.$$

Then, with $\beta_0 := 1$ we get $\alpha_j = \sum_{k=0}^{j} m_k \beta_{j-k}$. Despite the fact that there are efficient methods for solving linear systems for Hankel matrices, see, e.g., [19], AWE is numerically not feasible due to the unavoidable large errors caused by explicitly computing the moments M_j.

A breakthrough for Padé approximation methods was the observation that the moments need not be computed explicitly as moment matching is equivalent to projecting the state-space onto

$$\mathcal{V} = \operatorname{span}\{\tilde{B}, \tilde{A}\tilde{B}, \ldots, \tilde{A}^{r-1}B) =: \mathcal{K}(\tilde{A}, \tilde{B}, r\}, \tag{8}$$

where $\tilde{A} = (s_0E - A)^{-1}E,\ \tilde{B} = (s_0E - A)^{-1}B$) along

$$\mathcal{W} = \operatorname{span}\{C^H, \tilde{A}^H C^H, \ldots, (\tilde{A}^H)^{r-1}C^H\} =: \mathcal{K}(\tilde{A}^H, C^H, r). \tag{9}$$

Here, $\mathcal{K}(M, b, \ell)$ denotes the *Krylov subspace*

$$\operatorname{span}\{b, Mb, M^2b, \ldots, M^{\ell-1}b\}.$$

Bi-orthogonal bases of these two (block-)Krylov subspaces can be computed via the unsymmetric (block, band) Lanczos method [28, 29, 30]. In addition, a realization of the reduced-order model is computed as a by-product of the unsymmetric Lanczos process. It can be shown that for $n \geq m, p$, the corresponding reduced-order model satisfies

$$G(s) - \hat{G}(s) = \mathcal{O}((s - s_0)^{q(n)}), \quad q(n) \geq \left\lfloor \frac{n}{m} \right\rfloor + \left\lfloor \frac{n}{p} \right\rfloor.$$

The computation of a reduced-order model based on the moment matching idea implemented as unsymmetric Lanczos process is called the *(matrix) Padé-via-Lanczos ((M)PVL) method* [29, 28]. PVL is popular in circuit simulation and micro electronics as it can be applied without changes to descriptor systems which quite commonly arise in these applications if $s_0 \notin \Lambda(A, E)$ and $s_0 < \infty$. (By a descriptor system we denote a system in the form (7) where E is a singular matrix.) There exist several variants of PVL and MPVL adapted to special situations arising in circuit simulation.

Similarly, the ideas described above can be applied for matching the Markov parameters, i.e., $s_0 = \infty$. This problem is also called *minimal realization* and the implementation as model reduction method using projection on Krylov subspaces for non-singular E is discussed in [37, 43] while the situation for $s_0 = \infty$ is treated in [16].

Still there are several difficulties with the PVL method.

1. Despite several efforts ([8, 39]), so far there exists in general no computable error estimate or bound for $\|y - \hat{y}\|_2$.
2. The choice of the expansion points s_0 is usually based on heuristic criteria. (In circuit simulation often the operating frequency of the circuit can be used which explains the particular success in this application area.)
3. The reduced-order model provides a good approximation quality only locally.
4. The preservation of physical properties like stability or passivity can only be shown in very special cases; usually some post processing which (partially) destroys the moment matching properties is required. Such a method based on PVL or MPVL for RCL circuits is discussed in [27]; the resulting SyPVL/SyMPVL algorithm computes stable and passive reduced-order models and matches at least half as many moments as the PVL and MPVL methods.

Padé-type methods are also based on the moment matching property, but the approximations need not match the maximum possible number of moments, that is for approximating the transfer function it is only required that

$$G(s) - \hat{G}(s) = \mathcal{O}((s - s_0)^{\tilde{q}(n)}, \quad \tilde{q}(n) \leq q(n).$$

In this sense, the SyPVL and SyMPVL methods are Padé-type methods. Other Padé-type methods result from computing bases of the Krylov subspaces (8) and (9) with the Arnoldi process rather than the Lanczos method. This has the advantage of producing orthogonal bases and therefore of improved numerical stability properties. A band (block) Arnoldi process for model reduction which is based on (8) is devised in [27, 28]. It is also shown there that

$$\tilde{q}(n) \geq \left\lfloor \frac{n}{m} \right\rfloor.$$

A special case of this procedure for RCL circuits is known under the name PRIMA in circuit simulation [56]. Due to the symmetry properties of the realizations describing RCL circuits, this procedure is mathematically equivalent to SyMPVL [27, 28]. Other Krylov subspace methods for model reduction include [38, 42, 43]. A two-sided Arnoldi method that computes bases for both Krylov subspaces in (8)–(9) is suggested in [22]; this approach can be interpreted as a Padé approximation method.

Current research in Padé-type approximation methods include aspects of (multi-point) rational and tangential interpolation [3, 31, 32, 33, 39] which allow to obtain good local approximation properties at several frequencies $s_0, \ldots, s_k$. The projection matrices needed to compute reduced-order models by these approximation techniques can be computed from shift-and-invert, or, more generally, rational Krylov subspace methods [63].

Summarizing this section, the development of efficient Krylov subspace methods based on the Lanczos and Arnoldi algorithms in the past two decades has led to new model reduction techniques based on rational approximation. Without this numerical linear algebra machinery, there would be no way to apply Padé or Padé-like methods to the large-scale problems for which reduced-order models are needed.

4 Balanced Truncation

The basic idea of balanced truncation is to compute a special realization of an LTI system which is called a *balanced realization*. Such a realization is defined by the property that the controllability and the observability Gramians W_c and W_o, respectively, of the LTI systems are diagonal and equal. Note that these Gramians are given by the solutions of the two dual *Lyapunov equations*

$$AW_c + W_cA^T + BB^T = 0, \quad A^TW_o + W_oA + C^TC = 0, \tag{10}$$

with A, B, C as in (1). After computing a balanced realization using a state-space transformation

$$(TAT^{-1}, TB, CT^{-1}, D) = \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, D \right), \tag{11}$$

where $A_{11} \in \mathbb{R}^{r\times r}$, $B_1 \in \mathbb{R}^{r\times m}$, $C_1 \in \mathbb{R}^{p\times r}$ with $r \leq \hat{n}$ (with $\hat{n}$ being the McMillan degree of the system), the basic idea is then to use as the reduced-order model the truncated realization

$$(\hat{A}, \hat{B}, \hat{C}, \hat{D}) = (A_{11}, B_1, C_1, D). \tag{12}$$

This idea goes essentially back to [51, 53]. Collecting results from [51, 34, 64], the following result summarizes the properties of balanced truncation.

Proposition 4.1 *Let (A, B, C, D) be a realization of a stable LTI system with transfer function $G(s)$ and $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$ with associated transfer function $\hat{G}$ be computed as in (11)–(12). Then the following holds:*

a) *The reduced-order system $\hat{G}$ is balanced, minimal, and stable. Its Gramians are*

$$\tilde{P} = \tilde{Q} = \tilde{\Sigma} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}.$$

b) *The absolute error bound*

$$\|G - \tilde{G}\|_\infty \leq 2 \sum_{k=r+1}^{\hat{n}} \sigma_k =: \delta_{BT}. \tag{13}$$

holds and implies

$$\|y - \hat{y}\|_2 = \|Y - \hat{Y}\|_2 \leq \delta_{BT} \|U\|_2 = \delta_{BT} \|u\|_2,$$

where $\|\,.\,\|_2$ *denotes the* L_2*-norm in either state-space or frequency domain and* $\hat{Y}$ *the Laplace-transform of the output* $\hat{y}$ *of the reduced-order model.*

c) *If* $r = \hat{n}$*, then (12) is a minimal realization of the LTI system (1) and* $G = \hat{G}$*.*

Of particular importance is the error bound (13) as it allows an adaptive choice of the order of the reduced-order model based on a prescribed tolerance threshold for the approximation quality. (The error bound (13) can be improved in the presence of Hankel singular values with multiplicity greater than one—they need to appear only once in the sum on the right-hand side.)

It is easy to check that for a controllable and observable (minimal) system, i.e., a system with nonsingular Gramians,

$$T = \Sigma^{\frac{1}{2}} U^T R^{-T} \tag{14}$$

provides a balancing state-space transformation. Here $W_c = R^T R$ and $R W_o R^T = U \Sigma^2 U^T$ is a singular value decomposition. A nice observation in [64, 44] allows to compute (12) also for non-minimal systems without the need to compute the full matrix T. The first part of this observation is that for $W_o = S^T S$,

$$S^{-T}(W_c W_o) S^T = (SR^T)(SR^T)^T = (U \Sigma V^T)(V \Sigma U^T) = U \Sigma^2 U^T$$

so that U, Σ can be computed from an SVD of SR^T,

$$SR^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}, \qquad \Sigma_1 = \operatorname{diag}(\sigma_1, \ldots, \sigma_r). \tag{15}$$

The second part needed is the fact that computing

$$T_l = \Sigma_1^{-1/2} V_1^T R, \quad T_r = S^T U_1 \Sigma_1^{-1/2}, \tag{16}$$

and

$$\hat{A} := T_l A T_r, \quad \hat{B} := T_l B, \quad \hat{C} := C T_r \tag{17}$$

is equivalent to first computing a minimal realization of (1), then balancing the system as in (11) with T as in (14), and finally truncating the balanced realization as in (12). In particular, the realizations obtained in (12) and (17) are the same, T_l contains the first r rows of T and T_r the first r columns of T^{-1}—those parts of T needed to compute A_{11}, B_1, C_1 in (11). Also note that the product $T_r T_l$ is a projector onto an r-dimensional subspace of the state-space

and model reduction via (17) can therefore be seen as projecting the dynamics of the system onto this subspace.

The algorithm resulting from (17) is often referred to as the *SR method* for balanced truncation. In [44, 64] and all textbooks treating balanced truncation, S and R are assumed to be the (square, triangular) Cholesky factors of the system Gramians. In [13] it is shown that everything derived so far remains true if instead of Cholesky factors, full-rank factors of the system Gramians are used. This yields a much more efficient implementation of balanced truncation whenever $\hat{n} \ll n$ (numerically). Low numerical rank of the Gramians usually results in/from a rapid decay of their eigenvalues as shown in Figure 1 and implies a rapid decay of the Hankel singular values. Partial analysis of this phenomenom is given in [4, 60]. An algorithm exploiting this fact by directly computing numerical full-rank factors is derived in [13], see also [12, 14].

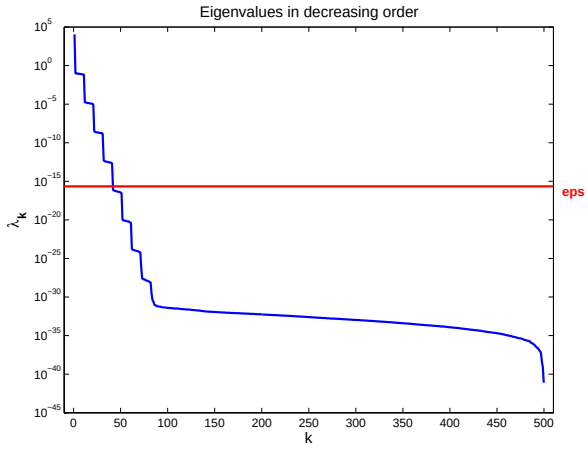

Fig. 1 Decay of the numerical rank of the controllability Gramian of a random LTI system with $n = 500$, $m = 10$, and stability margin ≈ 0.055.

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 [59, 47, 40]) or in $\mathcal{O}(n \log^2(n))$ (see [9]) if A is approximated by a hierarchical matrix [35].

We will briefly describe an implementation of balanced truncation for large-scale sparse systems. We follow here the description given in [10]. For this purpose, we note that the first step of any balanced truncation procedure consists in solving the Lyapunov equations (10). Here we want to go for low-rank approximations of factors of W_c, W_o. Hence, we start by considering the solution of Lyapunov equations of the form

$$FQ + QF^T = -WW^T,$$

where F is stable. The ADI iteration for solving such equations can be written as follows [67]:

$$\begin{aligned}(F + p_j I)Q_{(j-1)/2} &= -WW^T - Q_{j-1}(F^T - p_j I),\\ Q_j(F^T + \overline{p_j} I) &= -WW^T - (F - \overline{p_j} I)Q_{(j-1)/2}.\end{aligned}$$

If the shift parameters p_j are chosen appropriately, then $\lim_{j\to\infty} Q_j = Q$ with a superlinear convergence rate. Starting this iteration with $Q_0 = 0$, the ADI iteration can be re-written as fixed-point iteration

$$\begin{aligned}Q_j &= (F - \overline{p_j} I)(F + p_j I)^{-1} Q_{j-1} (F^T + \overline{p_j} I)^{-1}(F^T - p_j I)\\ &\quad -2\mathrm{Re}(p_j)(F + p_j I)^{-1} WW^T (F^T + \overline{p_j} I)^{-1}.\end{aligned} \tag{18}$$

Observing that for stable F, Q is positive semidefinite, we can assume that $Q_j = Y_j Y_j^T$ for some $Y_j \in \mathbb{R}^{n\times r_j}$. Inserting this into (18), re-arranging terms and combining two iteration steps, we obtain the following *factored ADI iteration*:

$V_1 \leftarrow \sqrt{-2\mathrm{Re}\,(p_1)}(F + p_1 I)^{-1} W, \qquad Y_1 \leftarrow V_1$

FOR $j = 2, 3, \ldots$

$\quad V_j \leftarrow \frac{\sqrt{\mathrm{Re}(p_j)}}{\sqrt{\mathrm{Re}(p_{j-1})}} \left(I - (p_j + \overline{p_{j-1}})(F + p_j I)^{-1}\right) V_{j-1},$

$\quad Y_j \leftarrow \left[\begin{array}{cc} Y_{j-1} & V_j \end{array}\right].$

END FOR

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 $F + p_j I$. Hence, if convergence with respect to a suitable stopping criterion is achieved after $j_{\max}$ steps, $Y_{j_{\max}} = \left[\begin{array}{ccc} V_1 & \ldots & V_{j_{\max}} \end{array}\right] \in \mathbb{R}^{n\times j_{\max} w}$. For large n and small w we can therefore expect that $r_j = j_{\max} w \ll n$. In that case, we have computed a low-rank approximation $Y_{j_{\max}}$ to a factor Y of the solution, i.e., $Q = YY^T \approx Y_{j_{\max}} Y_{j_{\max}}^T$. For an implementation of this method, we need a strategy to select the shift parameters. We will not discuss this here any further; see [68, 69] for a detailed discussion of this problem and an optimal choice in case of symmetric negative definite F. A numerically inexpensive, heuristic parameter selection that gives good performance in practice can be found in [57]. 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.

Also note that the above algorithm can be implemented in real arithmetic by combining two steps, even if complex shifts need to be used (which may be the case if F is nonsymmetric). A complexity analysis of the factored ADI method is hard to give. It depends on the method used to solve the linear systems in each iteration step. In any case, all linear systems of equations to be solved in one iteration step have the same coefficient matrix $F + p_j I_n$. If this is a banded matrix or can be re-ordered to become banded, then a direct solver can be employed. This remains true for some other sparsity patterns that allow a sparse Cholesky or LU factorization, see [18, 50]. If workspace permits, it is desirable to compute all the factorizations of $F + p_j I_n$ for each different shift parameter beforehand (usually, very few parameters are used). These factorizations can then be used in the iteration steps of the ADI iteration so that only forward

and backward substitution are needed to compute the new V_j. In particular, if F is symmetric negative definite as will be the case in many applications, and can be re-ordered to a (narrow) band matrix, then each factorization is $\mathcal{O}(n)$ and the total cost $\mathcal{O}(k_{\max} j_{\max} n)$ scales with n as desired. On the other hand, if iterative solvers are employed for the linear systems, it should be noted that due to the shift invariance of Krylov subspaces [25], only one Krylov space needs to be computed from which all linear systems of equations can be solved (see [47] for details). Hence, we obtain again an efficient implementation of the factored ADI iteration.

Stopping criteria for the factorized ADI iteration can be based either on the fact that $\|V_j\| \to 0$ very rapidly or on the residual norm $\|FY_jY_j^T + Y_jY_j^TF^T + WW^T\|$; see [60] for an efficient way to compute the Frobenius norms of the residuals. Alternatively, the relative change in the iterates

$$\frac{\|Q_j - Q_{j-1}\|_F}{\|Q_j\|_F} = \frac{\|Z_jZ_j^T - Z_{j-1}Z_{j-1}^T\|_F}{\|Z_jZ_j^T\|_F} = \frac{\|V_jV_j^T\|_F}{\|Z_jZ_j^T\|_F} = \frac{\|V_j^TV_j\|_F}{\|Z_j^TZ_j\|_F},$$

where the latter equality (based on $\|MM^T\|_F = \|M^TM\|_F$ for any matrix M) is cheap to evaluate whenever the number of columns in Z remains small with respect to n.

It should also be noted that for balanced truncation, the two factored ADI iterations needed to compute the solution factors of the Gramians can be coupled. As the iteration parameters are only based on the spectrum of F (with $F = A, A^T$ and A from (10)), they will be the same for both iterations and thus need only be computed once. Moreover, solving a linear system of equations with coefficient matrix $A^T + p_kI_n$ as needed in the iteration for the observability Gramian can be cheaply implemented using the factorization of $A + p_kI_n$ computed in the iteration for the controllability Gramian. Suppose we have computed a permuted LU decomposition (with the permutation based on pivoting and fill-in minimization)

$$A + p_kI_n = PLUQ,$$

with P, Q permutation matrices, L lower triangular with unit diagonal, U upper triangular, then

$$(A + p_jI)^{-1}V_{j-1}^B = (PLUQ)^{-1}V_{j-1}^B$$

is computed by permutation according to P, forward substitution with respect to L, backward substitution with respect to U and permutation with respect to Q, while

$$(A^T + p_jI)^{-1}V_{j-1}^C = (Q^TU^TL^TP^T)^{-1}V_{j-1}^C$$

can be implemented in the order: permutation with respect to Q^T, forward substitution with respect to U^T, backward substitution with respect to L^T, and permutation with respect to P^T.

Summarizing, new and efficient algorithms for Lyapunov equations based on the (approximate) low-rank nature of solutions yield balanced truncation algorithms that can be applied to problems of the same complexity as modal and Padé(-type) methods. In Section 6 it will be shown that such a balanced truncation implementation is more than competitive with the other methods.

5 Comparison

A joint feature of most model reduction methods is that they can be interpreted as *Galerkin* or *Petrov-Galerkin* type projection of the state-space onto a low-dimensional subspace $\mathcal{V}$ along the orthogonal complement of another subspace $\mathcal{W}$: assume

$$x \approx VW^T x =: \tilde{x},$$

where

$$\operatorname{range}(V) = \mathcal{V}, \quad \operatorname{range}(W) = \mathcal{W},$$

and W is chosen such that $W^T V = I_r$. Then, with $\hat{x} = W^T x$, we obtain $x \approx V\hat{x}$ and

$$\|x - \tilde{x}\| = \|x - V\hat{x}\|.$$

For modal truncation as described in Section 2, $\mathcal{V}$ is simply the eigenspace corresponding to the chosen eigenvalues, i.e., $V = U_1$, and $\mathcal{W} = \mathcal{V}$ for unitarily diagonalizable A while

$$\mathcal{W} = \operatorname{span}\{w_1, \ldots, w_r\},$$

where

$$\begin{bmatrix} w_1 & \ldots & w_n \end{bmatrix} = T^{-T}$$

with T as in Section 2. In any case, modal truncation has to choose $\mathcal{V}$ (for diagonalizable A) from the $\binom{n}{r}$ many eigenspaces of A.

Padé-via-Lanczos methods can be interpreted analogously with $\mathcal{V}$ and $\mathcal{W}$ as in (8) and (9), respectively. That is, once s_0 is determined, PVL has no choice to pick the subspace $\mathcal{V}$ in any other way than in (8).

For balanced truncation, the subspaces $\mathcal{V}$ and $\mathcal{W}$ are determined during the computation via

$$\mathcal{V} = \operatorname{range}(T_l), \quad \mathcal{W} = \operatorname{range}(T_r).$$

As the subspaces are not fixed prior to the computation, balanced truncation can—in principle—choose $\mathcal{V}$ from the *Grassman manifold*

$$\mathcal{G}(n, r) := \{\mathcal{V} \subset \mathbb{R}^n \ : \ \dim \mathcal{V} = r\}.$$

If we now consider the minimization problem

$$\min_{V \in \mathcal{G}(n,r)} \|x - V\hat{x}\| \,!,$$

it becomes clear that the larger variety in the selection of the projection subspace makes it likely that balanced truncation produces a model with smaller approximation error.

Of course, this is quite a simple discussion (see also [20] for similar observations). It does not take into account any modifications of modal truncation like base enrichment or the fact that an informed choice on the frequency s_0 in PVL methods will necessarily lead to good approximation qualities in a desired region. Thus, the comparison is not always favorable for balanced truncation, but examples in the next section demonstrate that often, a better approximation is achieved if the same reduced-order is used for the methods under discussion or, for the same error tolerance, a model of smaller order is often sufficient when using balanced truncation.

6 Numerical Examples

In the numerical experiments reported in this section, we have used the balanced truncation implementation provided in Lyapack [61] which closely follows the decription in Section 4, a simple implementation of modal truncation employing MATLAB's `eigs` which is based on ARPACK [45], and a PVL implementation[1] for single-input, single-output (SISO) systems ($m = p = 1$) corresponding to [6]. All experiments were performed using MATLAB Release 14 (SP2).

Example 1. Our first example is a MEMS device, called a microthruster. This device co-integrates solid fuel with a silicon micromachined system and is used for steering "nano-satellites", but also for gas generation or as high-energy actuator. To produce an impulse, fuel is ignited by heating a resistor at the top of a particular microthruster.

The design problem is to reach the ignition temperature within the fuel, but at the same time staying below the critical temperature at neighboring microthrusters. Moreover, the resistor temperature during the heating pulse should not become too high as this leads to the destruction of the membrane. Figure 2 shows a microthruster array, further information is available in [48, 54].

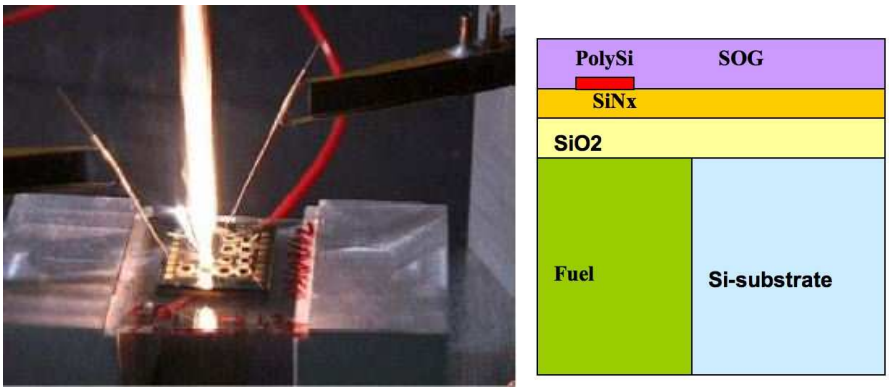

Fig. 2 A microthruster array at ignition (left) and schematic view of a cross-section (right). The array is about the size of a thumbnail. (By courtesy of C. Rossi, LAAS-CNRS.)

Here, we use two different 2D thermo-electrical models of a single microthruster as provided at the Oberwolfach Benchmark site [54]. A finite-element discretization using linear elements leads to $n = 4,257$ while using quadratic elements results in $n = 11,445$. The model has one input and seven outputs. As the PVL implementation used here can only treat SISO systems, we only use the first output in the experiments. That is, all frequency responses shown correspond to the transfer function from the input to the first output.

Figure 3 shows the frequency response (Bode magnitude plot) and the pointwise relative errors in the transfer function, i.e.,

$$\frac{\left|G(\jmath\omega) - \hat{G}(\jmath\omega)\right|}{|G(\jmath\omega)|},$$

[1] Available at `http://www.cs.ucdavis.edu/~bai/ROMmatlab/`

for the full $n = 4257$ system as well as reduced-order models of order $r = 21$ computed by balanced truncation, PVL with $s_0 = 1$ and $s_0 = 10^4$, as well as modal truncation. Obviously, the model computed by balanced truncation gives a satisfactory approximation over the whole frequncy range while PVL shows good approximation errors in a frequency range around the selected expansion point s_0. Close to s_0, the relative approximation error of PVL is even better than that of balanced truncation, but grows to more than 100% in frequency ranges far from s_0. The balanced truncation approximation error stays below 0.001% over the whole frequency range! The $r = 21$ model computed by modal truncation is quite useless except for very tiny frequencies. It should be noted that for this example, base enrichment using static solutions does not improve the accuracy of modal truncation—in contrast, the approximation error even grows and is therefore not displayed. Note that the ADI based balanced truncation implementation used six different paramters p_j in the ADI iteration for solving the Lyapunov equations, i.e., the essential cost of balanced truncation comes here from six sparse matrix factorizations as compared to only one sparse matrix factorization in the PVL and modal truncation methods. Thus the computation time needed for balanced truncation is roughly six times that of PVL and modal truncation.

This example demonstrates the claim from Section 5 that usually, balanced truncation yields better approximation for the same order r of the reduced-order models than the other approaches discussed in this paper. This is also supported by the very similar situation in the $n = 11,445$ case, displayed in Figure 4.

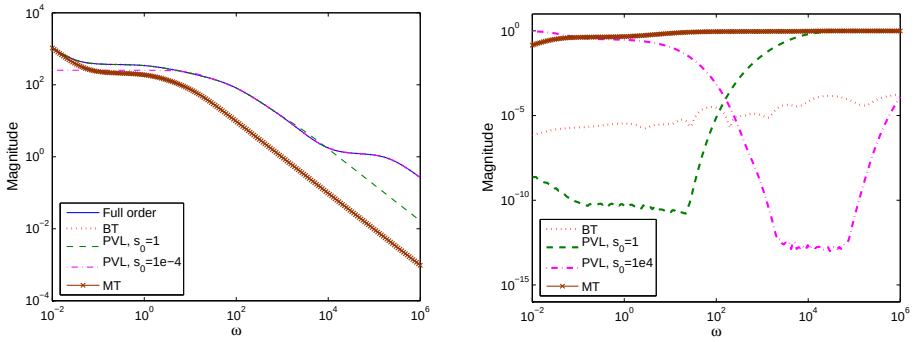

Fig. 3 Example 1 ($n = 4,257$), frequency response (left) and pointwise relative errors (right) of the reduced-order models.

Figure 5 demonstrates that in order to get a reduced-order model matching the approximation quality of balanced truncation, we need to more than double the order for the PVL-computed system even if we go up to $r = 1,000$ for modal truncation, tehre is still a significant error in the high frequency range. (The full order model, the BT $r = 21$ and the PVL $r = 50$ frequency response are on top of each other in Figure 5 (left).)

Example 2. Here we consider an LTI system of the form (7) coming from an optimal cooling problem for steel profiles. This problem arises in a rolling mill when different steps in the production process require different temperatures of the raw material. To achieve a high production rate, economical interests suggest to reduce the temperature as fast as possible to the required level before entering the next production phase. At the same time, the cooling

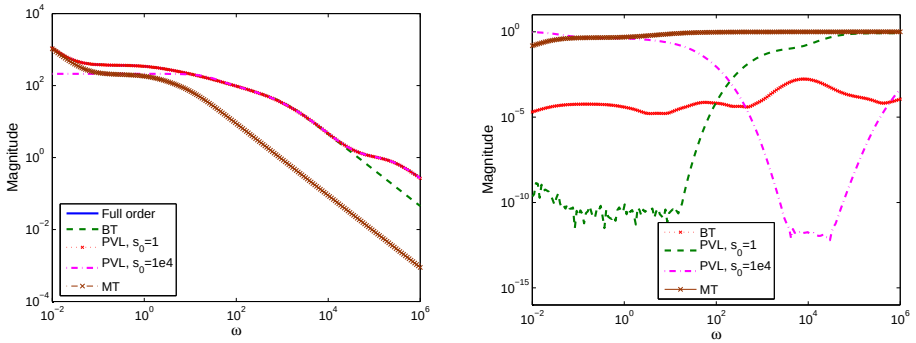

Fig. 4 Example 1 ($n = 11,445$), frequency response (left) and pointwise relative errors (right) of the reduced-order models.

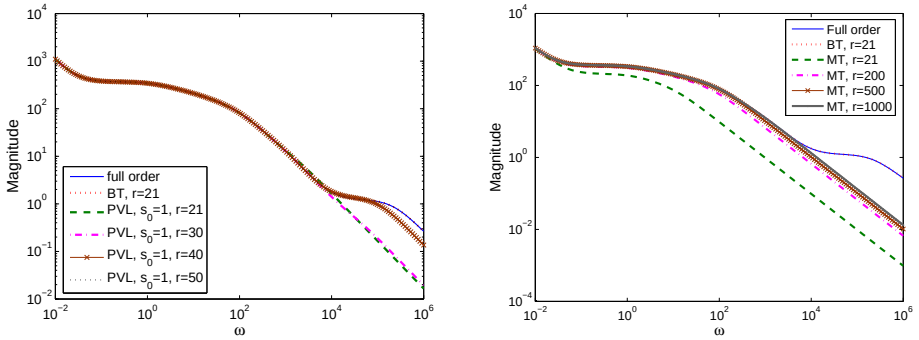

Fig. 5 Example 1, frequency response of the full order model and reduced-order models of several orders computed by PVL (left) and modal truncation (right).

process, which is realized by spraying cooling fluids on the surface, has to be controlled so that material properties do not degrade. Therefore, a primary design goal is to obtain a preferably even temperature distribution. For a picture of such a cooling plant see Figure 6.

We assume an infinitely long steel profile so that we may restrict ourselves to a 2D model. Exploiting the symmetry of the workpiece, the computational domain $\Omega \subset \mathbb{R}^2$ is chosen as the half of a cross section of the rail profile. The heat distribution is modeled by the instationary linear heat equation on Ω:

$$\begin{aligned} c\varrho\partial_t x(t,\xi) - \lambda\Delta x(t,\xi) &= 0 && \text{in } \mathbb{R}_{>0}\times\Omega, \\ x(0,\xi) &= x_0(\xi) && \text{in } \Omega, \\ \lambda\partial_\nu x(t,\xi) &= g_i && \text{on } \mathbb{R}_{>0}\times\Gamma_i, \partial\Omega = \bigcup_i \Gamma_i, \end{aligned} \tag{19}$$

where x is the temperature distribution ($x \in H^1([0,\infty], X)$ with $X := H^1(\Omega)$ being the state space), c the specific heat capacity, λ the heat conductivity and ϱ the density of the rail profile. We split the boundary into several parts Γ_i on which we have different boundary functions g_i,

allowing us to vary the controls on different parts of the surface. By ν we denote the outer normal of the boundary.

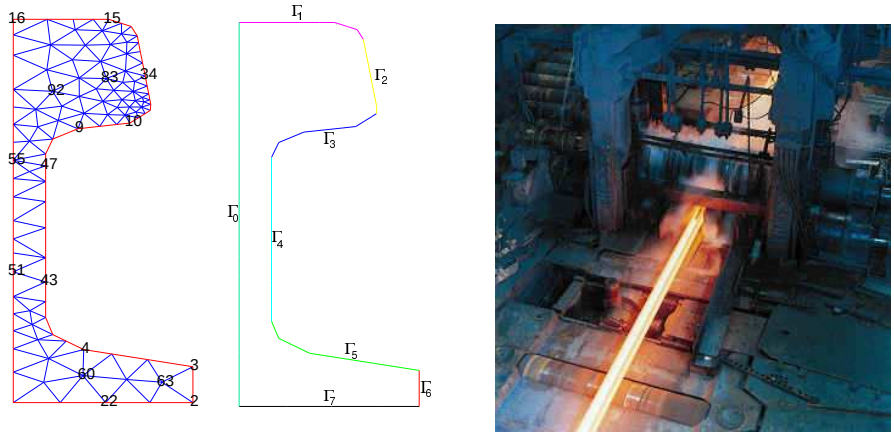

Fig. 6 Initial mesh, partitioning of the boundary, and a picture of a cooling plant.

Applying a standard Galerkin approach for discretizing the heat transfer model in space using a linear finite element basis results in a first-order ordinary differential equation. By refining the initial Delaunay triangulation shown in Figure 6 by bisection thrice, we obtain an LTI model with $n = 20,209$, corresponding to a maximum meshsize $h \approx 1.38 \cdot 10^{-2}$. For more information, see [15] and [54].

For comparing the model reduction algorithms, we choose the transfer function from the sixth input to the second output (the original model has 7 inputs and 6 outputs). This is motivated by the geometric vicinity of the corresponding input (spraying nozzle) and output (measured temperature in the area close to Γ_6). Figure 7 demonstrates that the situation here is similar to the one in Example 1. Here, prescribing a tolerance of 10^{-4} for the error bound (13), balanced truncation computes a reduced-order model of order $r = 8$. Again, among the models of order $r = 8$ we get the best global approximation by balanced truncation, while the PVL models exhibit the expected good local approximation quality and this time are only slightly off in the other regions. This is due to the fact that the given transfer function is very smooth and therefore easy to approximate. The model computed by modal truncation for $r = 8$ is completely useless; even the $r = 100$ model (including static modes) does only resemble the original transfer function for very low frequencies.

It should be noted that again, the balanced truncation model is computed using six shift parameters; hence the main computational cost in the ADI-based balanced truncation implementation is six sparse matrix factorizations and therefore about six times the cost of PVL and modal truncation.

Summarizing this section, the presented examples suggest that balanced truncation usually produces the best reduced-order model for a given order in the sense that the approximation error is globally the best. It is also competitive from the point of view of computational cost as it has basically the same complexity as the other methods, though the time to compute a reduced-order model is higher by some linear factor determined mostly by the number of shift

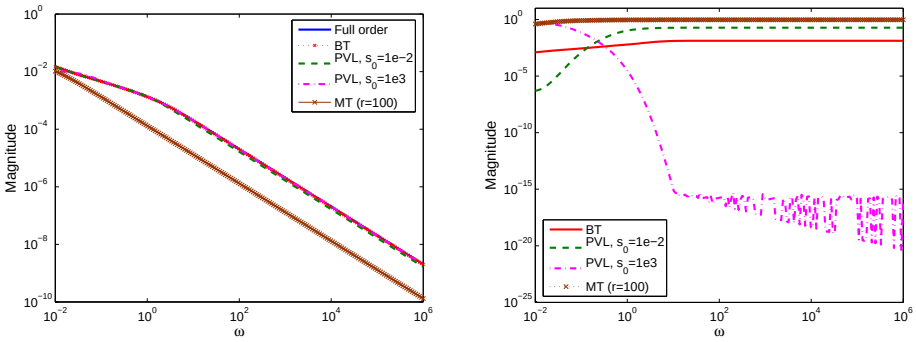

Fig. 7 Example 2, frequency response of the full order model and several reduced-order models (left); pointwise relative error of the reduced-order models (right).

parameters in the underlying ADI iteration. This is well compensated if the reduced-order has to be used in control or optimization or for many simulation runs.

7 Conclusions

Advances of numerical linear algebra techniques for sparse matrix computations during the past decades have lead to new and powerful tools for model reduction of linear systems. The Padé and Padé-type approximation techniques have set a standard for computing reduced-order models in circuit simulation and beyond. Balanced truncation has become an alternative for large and sparse problems as the $\mathcal{O}(n^3)$ barrier has been broken by several implementations of balanced truncation that make use of these new sparse matrix techniques.

Research directions in model reduction now shift more and more towards algorithms for parametric and nonlinear models in which parameters and nonlinearities, respectively, are to be preserved. Again, the efficient use of numerical linear algebra techniques will greatly improve the efficiency of these methods.

References

[1] ANSYS, Inc., http://www.ansys.com. *ANSYS*.
[2] A. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM Publications, Philadelphia, PA, 2005.
[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] A. Antoulas, D. Sorensen, and Y. Zhou. On the decay rate of Hankel singular values and related issues. *Sys. Control Lett.*, 46(5):323–342, 2002.
[5] R. Badía, P. Benner, R. Mayo, and E. Quintana-Ortí. Parallel algorithms for balanced truncation model reduction of sparse systems. In *Proc. of PARA'04 – Workshop on State-of-the-Art in Scientific Computing*, Lecture Notes in Computer Science. Springer-Verlag, Berlin, Heidelberg, New York, to appear.
[6] Z. Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Appl. Numer. Math*, 43(1–2):9–44, 2002.

[7] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, volume 11 of *Software - Environments - Tools*. SIAM Publications, Philadelphia, PA, 2000.
[8] Z. Bai and Q. Ye. Error estimation on the Padé approximation of transfer functions via the Lanczos process. *Electr. Trans. Num. Anal.*, 7:1–17, 1998.
[9] 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.
[10] P. Benner. Solving large-scale control problems. *IEEE Control Systems Magazine*, 14(1):44–59, 2004.
[11] 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.
[12] P. Benner and E. Quintana-Ortí. Model reduction based on spectral projection methods. Chapter 1 (pages 5–48) of [11].
[13] 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.
[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] P. Benner and J. Saak. A semi-discretized heat transfer model for optimal cooling of steel profiles. Chapter 19 (pages 353–356) of [11].
[16] P. Benner and V. Sokolov. Partial realization of descriptor systems. Preprint 2005-05, Fakultät für Mathematik, TU Chemnitz, 09107 Chemnitz, FRG, 2005. Available from `http://www.tu-chemnitz.de/sfb393`.
[17] 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.
[18] M. Bollhöfer and O. Schenk. Sparse matrix factorizations. *GAMM Mitteilungen*, ?(?):?–?, to appear.
[19] A. Böttcher and K. Rost. Topics in the numerical linear algebra of Toeplitz and Hankel matrices. *GAMM Mitteilungen*, 27(2):174–188, 2004.
[20] C. Brezinski. Model reduction by state space projection. *Comm. Appl. Anal.*, 8(3):323–333, 2004.
[21] R. Craig and M. Bampton. Coupling of substructures for dynamic analysis. *AIAA J.*, 6:1313–1319, 1968.
[22] J. Cullum and T. Zhang. Two-sided Arnoldi and nonsymmetric Lanczos algorithms. *SIAM J. Matrix Anal. Appl.*, 24(2):303–319, 2002.
[23] L. Dai. *Singular Control Systems*. Number 118 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Berlin, 1989.
[24] E. Davison. A method for simplifying linear dynamic systems. *IEEE Trans. Automat. Control*, AC–11:93–101, 1966.
[25] R. Freund. Solution of shifted linear systems by quasi-minimal residual iterations. In L. Reichel et al., editor, *Numerical Linear Algebra*, pages 101–121. Walter de Gruyter, Berlin, 1993.
[26] 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.
[27] R. Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *J. Comput. Appl. Math.*, 123(1–2):395–421, 2000.
[28] R. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerica*, 12:267–319, 2003.
[29] R. W. Freund and P. Feldmann. Small-signal circuit analysis and sensitivity computations with the PVL algorithm. *IEEE Transactions on Circuits and and Systems—II: Analog and Digital Signal Processing*, 43:577–585, 1996.
[30] K. Gallivan, E. Grimme, and P. Van Dooren. Asymptotic waveform evaluation via a Lanczos method. *Appl. Math. Lett.*, 7(5):75–80, 1994.

[31] K. Gallivan, E. Grimme, and P. Van Dooren. A rational Lanczos algorithm for model reduction. *Numer. Algorithms*, 12:33–63, 1996.
[32] K. Gallivan, A. Vandendorpe, and P. Van Dooren. Model reduction of MIMO systems via tangential interpolation. *SIAM J. Matrix Anal. Appl.*, 26(2):328–349, 2004.
[33] K. Gallivan, A. Vandendorpe, and P. Van Dooren. Sylvester equations and projection-based model reduction. *J. Comput. Appl. Math.*, 162(1):213–229, 2004.
[34] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞ norms. *Internat. J. Control*, 39:1115–1193, 1984.
[35] L. Grasedyck and W. Hackbusch. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70(4):295–334, 2003.
[36] M. Green and D. Limebeer. *Linear Robust Control*. Prentice-Hall, Englewood Cliffs, NJ, 1995.
[37] E. Grimme. *Krylov projection methods for model reduction*. PhD thesis, Univ. of Illinois at Urbana-Champaign, USA, 1997.
[38] E. Grimme, D. Sorensen, and P. Van Dooren. Model reduction of state space systems via an implicitly restarted Lanczos method. *Numer. Algorithms*, 12:1–31, 1996.
[39] S. Gugercin and A. Antoulas. Model reduction of large-scale systems by least squares. *Linear Algebra Appl.*, 2005. In press.
[40] S. Gugercin and J.-R. Li. Smith-type methods for balanced truncation of large systems. Chapter 2 (pages 49–82) of [11].
[41] R. Guyan. Reduction of stiffness and mass matrices. *AIAA J.*, 3:380, 1968.
[42] I. Jaimoukha and E. Kasenally. Krylov subspace methods for solving large Lyapunov equations. *SIAM J. Numer. Anal.*, 31:227–251, 1994.
[43] I. Jaimoukha and E. Kasenally. Implicitly restarted Krylov subspace methods for stable partial realizations. *SIAM J. Matrix Anal. Appl.*, 18(3):633–652, 1997.
[44] A. Laub, M. Heath, C. Paige, and R. Ward. Computation of system balancing transformations and other application of simultaneous diagonalization algorithms. *IEEE Trans. Automat. Control*, 34:115–122, 1987.
[45] R. Lehoucq, D. Sorensen, and C. Yang. *ARPACK User's Guide*, volume 6 of *Software - Environments - Tools*. SIAM Publications, Philadelphia, PA, 1998.
[46] 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.
[47] J.-R. Li and J. White. Low rank solution of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 24(1):260–280, 2002.
[48] J. Lienemann, E. Rudnyi, and J. Korvink. MST MEMS compact modeling meets model order reduction: Requirements and benchmarks. *Linear Algebra Appl.*, 2005. In press.
[49] S. Marschall. An approximate method for reducing the order of a linear system. *Contr. Eng.*, 10:642–648, 1966.
[50] G. Meurant. Gaussian elimination for the solution of linear systems of equations. In P. Ciarlet and J. Lions, editors, *Handbook of Numerical Analysis*, volume 7, pages 3–170. North-Holland/ Elsevier, Amsterdam, 2000.
[51] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, AC-26:17–32, 1981.
[52] MSC.Software Corporation, http://www.mscsoftware.com. *MSC.Nastran*.
[53] C. Mullis and R. Roberts. Synthesis of minimum roundoff noise fixed point digital filters. *IEEE Trans. Circuits and Systems*, CAS-23(9):551–562, 1976.
[54] Oberwolfach model reduction benchmark collection. Available at `http://www.imtek.de/simulation/benchmark/`.
[55] G. Obinata and B. Anderson. *Model Reduction for Control System Design*. Communications and Control Engineering Series. Springer-Verlag, London, UK, 2001.
[56] A. Odabasioglu, M. Celik, and L. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. In *Tech. Dig. 1997 IEEE/ACM Intl. Conf. Computer-Aided Design*, pages 58–65. IEEE Computer Society Press, Los Alamitos, CA, 1997.

[57] T. Penzl. *Numerische Lösung großer Lyapunov-Gleichungen*. Logos–Verlag, Berlin, Germany, 1998. (Dissertation, Fakultät für Mathematik, TU Chemnitz, 1998.).
[58] 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`.
[59] T. Penzl. A cyclic low rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418, 2000.
[60] T. Penzl. Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case. *Sys. Control Lett.*, 40:139–144, 2000.
[61] 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`.
[62] D. Rixen. A dual Craig–Bampton method for dynamic substructuring. *J. Comput. Appl. Math.*, 168(1–2):383–391, 2004.
[63] A. Ruhe. Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils. *SIAM J. Sci. Comput.*, 19(5):1535–1551, 1998.
[64] M. Tombs and I. Postlethwaite. Truncated balanced realization of a stable non-minimal state-space system. *Internat. J. Control*, 46(4):1319–1330, 1987.
[65] 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.
[66] H. van der Vorst. Computational methods for large eigenvalue problems. In P. Ciarlet and J. Lions, editors, *Handbook of Numerical Analysis*, volume 8, pages 3–179. North-Holland/ Elsevier, Amsterdam, 2002.
[67] E. Wachspress. Iterative solution of the Lyapunov matrix equation. *Appl. Math. Letters*, 107:87–90, 1988.
[68] E. Wachspress. The ADI model problem, 1995. Available from the author.
[69] E. Wachspress. ADI iteration parameters for the Sylvester equation, 2000. Available from the author.