

# Advances in balancing-related model reduction for circuit simulation

Peter Benner

**Abstract** We discuss algorithms for balanced truncation (BT) based model reduction of linear systems. BT is known to have good global approximation properties and to preserve important system properties. A computable error bound allows to choose the order of the reduced-order model adaptively. We will emphasize those aspects that makes the application of BT to models arising in circuit simulation a non-straightforward task. In recent years, these issues have been addressed by several authors. We will survey some of these developments and demonstrate that BT is now suitable for linear descriptor systems encountered in circuit simulation.

## 1 Introduction

Model order reduction (MOR) is an indispensable tool in the design and analysis of integrated circuits (ICs) and circuit simulation in general. This is due to the fact that on the one hand, almost all IC design relies heavily on simulation and on the other hand, the complexity of the mathematical models used to replicate the behavior of an actual electronic circuit is growing more rapidly than computing resources. This is caused by the increased packing density and multi-layer technology which nowadays requires the modeling of thermic and other parasitic effects caused by the interconnect. In many situations, only the use of MOR techniques allows the numerical simulation of the usually very large systems of ordinary differential and differential-algebraic equations used to describe (parts of) complex circuit layouts.

---

Peter Benner

Mathematik in Industrie und Technik, Fakultät für Mathematik, TU Chemnitz, D-09107 Chemnitz, Germany, e-mail: benner@mathematik.tu-chemnitz.de. The work reported in this paper was supported by the German Federal Ministry of Education and Research (BMBF), grant no. 03BEPAE1, and the EU project O-MOORE-NICE! funded with the Marie Curie Host Fellowships for the Transfer of Knowledge (ToK) Industry-Academia Partnership Scheme. Responsibility for the contents of this publication rests with the author.

MOR has been particularly successful in reducing the complexity of large linear subcircuits modeling parasitic effects of interconnect and in small signal analysis, and it is becoming an increasingly useful tool also in other areas of circuit design [1].

Linear circuit models can be described by linear descriptor systems of the form

$$E\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t), \quad (1)$$

where  $A, E \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$ ,  $D \in \mathbb{R}^{p \times m}$ , and  $x(t) \in \mathbb{R}^n$ ,  $y(t) \in \mathbb{R}^p$ ,  $u(t) \in \mathbb{R}^m$  denote generalized states, outputs, inputs, respectively. The corresponding transfer function

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

results from describing the input-to-output map  $u \rightarrow y$  in frequency domain.<sup>1</sup> One difficulty for balancing-related model reduction methods arises from  $E$  being singular as it is usually the case in circuit simulation. In this paper we will mainly focus on advances made for resolving this issue.

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

$$\hat{E}\dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{B}u(t), \quad \hat{y}(t) = \hat{C}\hat{x}(t) + \hat{D}u(t), \quad (3)$$

of order  $r$ ,  $r \ll n$ , with the same numbers of inputs ( $m$ ) and outputs ( $p$ ), i.e.,  $\hat{A}, \hat{E} \in \mathbb{R}^{r \times r}$ ,  $\hat{B} \in \mathbb{R}^{r \times m}$ ,  $\dots$ , and associated transfer function  $\hat{G}(s) = \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B} + \hat{D}$ , so that for the same input function  $u \in L_2(0, \infty; \mathbb{R}^m)$ , we have  $y(t) \approx \hat{y}(t)$ .

The most popular MOR methods in circuit simulation are Padé(-type) approximations, also known as moment-matching methods. The  $r$ th Padé approximant  $\hat{G}$  of  $G$  is defined by the property  $G(s) = \hat{G}(s) + \mathcal{O}((s - s_0)^{2r})$ , i.e.,  $M_j = \hat{M}_j$  for  $j = 0, \dots, 2r - 1$ , where the *moments*  $M_j, \hat{M}_j$  are the coefficients in a power (Laurent) expansion of  $G, \hat{G}$ , respectively, about some expansion point  $s_0 \notin \Lambda(A, E)^2$ . Moment-matching and Padé approximation properties are obtained for methods based on the unsymmetric Lanczos process, called the *(matrix) Padé-via-Lanczos ((M)PVL) method* [2–4]. *Padé-type methods* are also based on the moment matching property, but the approximations need not match the maximum possible number of moments. One such method is PRIMA [5] which employs the Arnoldi process to compute the reduced-order model. PRIMA is a success story in MOR for circuit simulation as besides having moment-matching properties, it preserves stability and passivity of RLC circuit models.

Despite the success with Padé(-type) approximation techniques based on the moment-matching properties of Krylov subspace methods, some major difficulties of this approach persist:

1. So far there exists in general no computable error estimate or bound for  $\|y - \hat{y}\|$  in some appropriate norm.
2. The reduced-order model provides good approximation quality only locally.

<sup>1</sup> Note that frequently in the area of circuit simulation, different notation is used: there  $E, A$ , and  $C$  become  $C, G$ , and  $L^T$ , respectively. The notation used here is standard in systems theory.

<sup>2</sup>  $\Lambda(A, E)$  denotes the set of generalized eigenvalues of the matrix pencil  $A - \lambda E$ .

3. 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.

There are many recent advances with respect to items 1.–3. discussed in the recent literature, see, e.g., [6, 7], but due to space limitations we can not discuss all these new developments here.

All the above problems of moment-matching methods are avoided when using balanced truncation (BT) or its relatives. Computable error bounds or estimates exist and come essentially for free as by-product of the computational procedures for obtaining the reduced-order model. The methods have good global approximation properties and thus, the reduced-order models can serve as surrogate for a large frequency range. Stability of the linear system is preserved for all variants of BT, other properties like passivity (which is important for passive devices) can be preserved by a variant of BT called *positive-real BT (PRBT)* (see, e.g., [8, 9] and references therein). Note that the error estimate for PRBT given, e.g., in [9], needs a good estimate of the  $\mathcal{H}_\infty$ -norm (defined below) of  $G(s) + D^T$  and thus is not as cheap to evaluate as, e.g., the BT error bound (7) below. On the other hand, for any reasonable approximation this quantity can be replaced without significant loss of information by the  $\mathcal{H}_\infty$ -norm of  $\hat{G}(s) + D^T$  which can be computed at moderate cost.

It has been common belief until recently that BT-related methods are not applicable in circuit simulation due to the  $\mathcal{O}(n^3)$  complexity required by matrix equation solvers used to solve the underlying Lyapunov or algebraic Riccati equations. But advances in numerical linear algebra nowadays allow to compute solutions to those Lyapunov and Riccati equations arising in BT-related methods for linear systems at a computational cost that scales with the cost for solving linear systems of equations with coefficient matrix  $A + s_0E$ . Thus, these methods can now be applied to systems of order  $\mathcal{O}(10^6)$ . Moreover, most of the difficulties resulting from a singular  $E$  matrix have now also been overcome. Many of these developments are discussed in [6, 10] and references therein.

In the main part of this paper (Section 2), we will focus on one possibility to extend BT to descriptor systems. A parallel implementation of an earlier version of this algorithm is already described in [11]. This method does not make use of possible sparsity of the system matrices and can thus be applied in order to reduce fairly small linear subcircuits with up to a few thousands elements. We will comment briefly on extensions to the case of large-scale, sparse matrices in Section 3. Some further issues like sparsification of the reduced-order model and passivity preservation using balancing-related methods will also be discussed in Section 3.

## 2 A Balanced Truncation Algorithm for Descriptor Systems

The method described in this section is based on two stages. In the first stage, we decompose the transfer function of the descriptor systems into a part corresponding to all finite poles and a polynomial part. Standard BT can then be applied to the

first part while the transfer function of the polynomial part is preserved, but may be realized by a system of smaller order.

First, we briefly explain how we apply BT to the part corresponding to the finite poles, then we present a method to compute the required decomposition of the transfer function. In subsection 2.3 we combine these algorithms to a BT algorithm for descriptor systems and some numerical results are reported in subsection 2.4.

## 2.1 Balanced Truncation for Generalized State-Space Systems

In this section, we briefly describe BT for systems of the form (1) when  $E$  is nonsingular. Such systems will be called *generalized state-space (GSS) systems* in the following. For more thorough descriptions and in particular the mathematical background of the method in case  $E = I_n$  see [8, 12, 13].

Throughout this and the following sections, we always assume  $\lambda E - A$  to be stable, i.e., to have all its (finite) eigenvalues in the open left half of the complex plane. We call a GSS system, realized by  $(A, B, C, D, E)$  as in (1) with  $E$  nonsingular *balanced*, if the solutions  $P, Q$  of the dual Lyapunov equations

$$APE^T + EPA^T + BB^T = 0, \quad A^T QE + E^T QA + C^T C = 0, \quad (4)$$

satisfy

$$P = E^T QE = \text{diag}(\sigma_1, \dots, \sigma_n) \quad \text{with} \quad \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n > 0. \quad (5)$$

The  $\sigma_j$  are the Hankel singular values (HSVs) of the GSS system.

*Remark 1.*  $P, E^T QE$  are the *controllability* and *observability Gramians* of the linear time-invariant system  $\dot{x}(t) = E^{-1}Ax(t) + E^{-1}Bu(t)$ ,  $y(t) = Cx(t) + Du(t)$ , which is equivalent to (1). As our method is equivalent to applying BT to this standard state-space system, this definition appears to be quite natural here. Our algorithm to solve (4) computes  $E^T QE$  directly rather than  $Q$  — this has a certain advantage over using  $Q$  as observability Gramian as in [14, 15]. On the other hand, the definition used in [14, 15] yields Gramians directly for the descriptor system (1) and turns out to be the appropriate approach in this case. Note also that in case  $E$  is singular, in contrast to common belief in many references in the literature, BT can not be directly based on (4) as the Lyapunov equations may or may not have solutions [14, 15]. The BT method for descriptor systems developed in [14, 15] therefore makes use of so-called *projected Lyapunov equations*. It turns out that Algorithm 3 below is mathematically equivalent to this approach, but solves the projected Lyapunov equations only implicitly.

A balanced realization of a minimal GSS system can be computed via a *system equivalence transformation*

$$\begin{aligned} \mathcal{T} : (A, B, C, D, E) &\mapsto (LAT, LB, CT, D, LET) \\ &= \left( \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, [C_1 \ C_2], D, \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \right), \end{aligned} \quad (6)$$

where  $L, T$  are nonsingular matrices so that (5) is true for the transformed system. Such a transformation always exists which easily follows from the theory for standard systems [12].

Now if  $\sigma_r > \sigma_{r+1}$  and the partitioning in (6) is chosen according to  $r$ , simple truncation leads to the reduced-order model  $(\hat{A}, \hat{B}, \hat{C}, \hat{D}, \hat{E}) = (A_{11}, B_1, C_1, D, E_{11})$  with some benign properties: first, it can be shown that the reduced-order model is again stable, and second, the error bound

$$\|G - \hat{G}\|_{\mathcal{H}_\infty} \leq 2 \sum_{j=r+1}^n \sigma_j, \quad (7)$$

holds. Here,  $\|\cdot\|_{\mathcal{H}_\infty}$  denotes the  $\mathcal{H}_\infty$ -norm, i.e., the 2-induced Hardy operator norm of real rational matrix functions having no poles in the right half plane (see, e.g., [8] and references therein). Due to its nature as 2-induced operator norm, the bound (7) implies (using the Paley-Wiener theorem)

$$\begin{aligned} \|y - \hat{y}\|_{L_2(0, \infty; \mathbb{R}^p)} &= \|y - \hat{y}\|_{\mathcal{H}_2^p} = \|Gu - \hat{G}u\|_{\mathcal{H}_2^p} \leq \|G - \hat{G}\|_{\mathcal{H}_\infty} \|u\|_{\mathcal{H}_2^m} \\ &\leq 2 \left( \sum_{j=r+1}^n \sigma_j \right) \|u\|_{\mathcal{H}_2^m} = 2 \left( \sum_{j=r+1}^n \sigma_j \right) \|u\|_{L_2(0, \infty; \mathbb{R}^m)}, \end{aligned}$$

where  $\mathcal{H}_2^q$  is the frequency domain equivalent of  $L_2(0, \infty; \mathbb{R}^q)$  obtained by the (normalized) Laplace transform. Thus, the output error in both, frequency and time domain, can be bounded. The existence of this bound is considered to be the main advantage of BT over other MOR methods, in particular as it can be computed as a by-product of the BT procedure without additional cost and allows to adaptively choose the order of the reduced-order model if it is requested that  $\|y - \hat{y}\| \leq \tau \|u\|$  for a given tolerance  $\tau$  and either one of the 2-norms in frequency or time domain.

It remains to show how to solve (4) and how to compute  $L, T$  as in (6). First we note that it is actually not necessary to compute  $P, Q$  and  $L, T$  explicitly. Following the ideas for standard systems from [13], one can show that the reduced-order model can be computed (even for non-minimal systems) by the following procedure: as  $P, Q$  are positive semidefinite, there exist matrices  $S \in \mathbb{R}^{r_p \times n}, R \in \mathbb{R}^{r_q \times n}$  (by  $r_p, r_q$  we denote the ranks of  $P, Q$ , respectively) so that  $P = S^T S$  and  $E^T Q E = R^T R$ . Now compute a singular value decomposition (SVD)

$$SR^T = [U_1, U_2] \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}, \quad \Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_r)$$

and set  $\hat{L} = \Sigma_1^{-1/2} V_1 R E^{-1} \in \mathbb{R}^{r \times n}$ ,  $\hat{T} = S^T U_1 \Sigma_1^{-1/2} \in \mathbb{R}^{n \times r}$ . Then it is easy to verify that  $\hat{L}$  and  $\hat{T}$  are the first  $r$  rows and columns of  $L, T$  from (6) and thus the reduced-

order model can equivalently be computed by

$$(\hat{A}, \hat{B}, \hat{C}, \hat{D}, \hat{E}) = (\hat{L}\hat{A}\hat{T}, \hat{L}\hat{B}, \hat{C}\hat{T}, D, \hat{L}\hat{E}\hat{T}).$$

Note that  $\hat{E} = I_r$  and thus the corresponding computations can be saved. Also observe that  $\Sigma_1^{-1/2}V_1R \in \mathbb{R}^{r \times n}$  and thus  $\hat{L}$  can be obtained as the solution of the linear system of equations  $\hat{L}E = \Sigma_1^{-1/2}V_1R$  with only  $r$  right-hand sides so that  $E^{-1}$  needs not be formed explicitly.

In many cases, the numerical ranks of  $P, Q$  are small ( $r_P, r_Q \ll n$ ) and thus it is desirable to compute  $S, R$  as above directly without first computing Cholesky factors of  $P$  and  $E^TQE$  as it is done in Hammarling's method for (4) [14–16]. A very efficient method to get  $S, R$  directly can be based on the sign function method, for details see [17, 18]. The resulting algorithm is given in Algorithm 1. There, the scalar  $c_j$  is a scaling factor used to accelerate convergence of this iteration (which is ultimately quadratic). The full-rank factors are computed using rank-revealing LQ/QR factorizations (RRLQ/RRQR) with respect to a tolerance  $\varepsilon$  for rank determination, without accumulation of orthogonal factors which makes their computation fairly cheap with a computational complexity bounded by  $4n \max\{r_P, r_Q\}^2$  operations per iteration step. For details on the scaling parameter  $c_j$  and the column/row compression step see [17]. As  $\lim_{j \rightarrow \infty} A_j = -E$ , the iteration can easily be stopped as soon as  $\|A + E\| \leq \tau \cdot \|E\|$  for an appropriate convergence tolerance  $\tau$  and an easy to compute matrix norm. After convergence, we obtain the desired full-rank factors of the Gramians as  $S = \frac{1}{\sqrt{2}}(E^{-1} \lim_{j \rightarrow \infty} S_j)^T$ ,  $R = \frac{1}{\sqrt{2}} \lim_{j \rightarrow \infty} R_j$ . Note that again, there is no need to compute  $E^{-1}$  as  $S$  can be obtained by solving a system of linear equations with  $r_P$  right-hand sides. In [17] a variant of this iteration is discussed that employs the  $R$ -factor of the QR factorization of  $E$  in the iteration instead of  $E$  itself. In this way, each iteration step becomes a lot cheaper and furthermore, the QR factorization can be used to solve the required linear systems of equations with coefficient matrix  $E$  (when computing  $\hat{L}$  and  $S$ ) just by application of the transposed orthogonal factor of  $E$  and backward substitution.

## 2.2 Additive Decomposition of the Transfer Function

In this section we show how to compute an explicit additive decomposition of the transfer function  $G(s)$  as in (2) so that  $G(s) = G_f(s) + G_\infty(s)$ , where  $G_f(s)$  and  $G_\infty(s)$  have exclusively finite and infinite poles, respectively. Such an algorithm was already proposed in [19]. Here, we suggest a method which employs different computational kernels to achieve this decomposition. The required computations are particularly efficient on computer architectures where matrix multiplication can be performed (almost) at peak performance.

The additive decomposition is achieved by computing nonsingular matrices  $U, V \in \mathbb{R}^{n \times n}$  that block-diagonalize  $\lambda E - A$ , i.e.,

**Algorithm 1** Coupled Newton Iteration for Dual Lyapunov Equations.

---

 INPUT:  $(A, B, C, E) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m} \times \mathbb{R}^{p \times n} \times \mathbb{R}^{n \times n}$  as in (4); a convergence tolerance  $\tau$ .

 OUTPUT: Numerical full-rank factors such that  $P = S^T S$ ,  $E^T Q E = R^T R$ , where  $P, Q$  are the solutions of (4).

- 1:  $A_0 \leftarrow A, S_0 \leftarrow B, R_0 \leftarrow C, j = 0$ .
  - 2: **while**  $\|A_j + E\|_1 > \tau$  **do**
  - 3:   Determine scaling factor  $c_j$ .
  - 4:    $S_{j+1} \leftarrow$  full-rank factor of  $\frac{1}{\sqrt{2c_j}} \begin{bmatrix} S_j & c_j E A_j^{-1} S_j \end{bmatrix}$ .
  - 5:    $R_{j+1} \leftarrow$  full-rank factor of  $\frac{1}{\sqrt{2c_j}} \begin{bmatrix} R_j \\ c_j R_j A_j^{-1} E \end{bmatrix}$ .
  - 6:    $A_{j+1} \leftarrow \frac{1}{2c_j} (A_j + c_j^2 E A_j^{-1} E)$ .
  - 7:    $j \leftarrow j + 1$ .
  - 8: **end while**
  - 9: Solve  $S E^T = S_j^T$  for  $S$  and set  $R := R_j$ .
- 

$$\lambda \hat{E} - \hat{A} := U(\lambda E - A)V = \lambda \begin{bmatrix} E_0 & 0 \\ 0 & E_\infty \end{bmatrix} - \begin{bmatrix} A_f & 0 \\ 0 & A_\infty \end{bmatrix},$$

and setting  $\hat{B} := UB =: \begin{bmatrix} B_f \\ B_\infty \end{bmatrix}$ ,  $\hat{C} := CV =: [C_f \ C_\infty]$ ,  $\hat{D} := D$ . Then

$$\begin{aligned} G(s) &= C(sE - A)^{-1}B + D = \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B} + \hat{D} \\ &= [C_f \ C_\infty] \begin{bmatrix} sE_f - A_f & \\ & sE_\infty - A_\infty \end{bmatrix}^{-1} \begin{bmatrix} B_f \\ B_\infty \end{bmatrix} + D \\ &= \underbrace{C_f(sE_f - A_f)^{-1}B_f}_{=: G_f(s)} + \underbrace{C_\infty(sE_\infty - A_\infty)^{-1}B_\infty + D}_{=: G_\infty(s)}. \end{aligned} \quad (8)$$

Thus, we can apply balanced truncation as described in the previous subsection to  $G_f$  in order to obtain a reduced-order system with transfer function  $\hat{G}_f$ .

The block-diagonalization is achieved using a two stage process. First, a block-triangularization of  $\lambda E - A$  is computed using the disk function method as described next, then a block diagonalization is achieved by solving a certain generalized Sylvester equation.

**Block-triangularization using the disk function method.** The algorithm discussed here is adapted from [20], and is based on earlier work by Malyshev [21]. This algorithm is referred to as *disk function method* as it can be used to compute the disk function of a matrix pencil, for details see [22]. We also make use of improvements suggested in [23] to reduce its cost.

Given a regular matrix pencil  $\lambda E - A$  having all finite eigenvalues inside the unit circle, Algorithm 2 provides an implementation of the disk function method which computes  $\tilde{U}, \tilde{V}$  such that

**Algorithm 2** Disk Function Method.

---

 INPUT: A matrix pencil  $\lambda E - A$ ,  $E, A \in \mathbb{R}^{n \times n}$  with no eigenvalues on the unit circle.

 OUTPUT: Orthogonal  $\tilde{U}, \tilde{V} \in \mathbb{R}^{n \times n}$  that block-triangularize  $\lambda E - A$ .
1: Set  $E_0 = E, A_0 = A$ .2: **for**  $j = 0, 1, \dots$  until convergence **do**3:  $\begin{bmatrix} E_j \\ -A_j \end{bmatrix} \rightarrow \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} R_j \\ 0 \end{bmatrix}$  (QR factorization),4:  $A_{j+1} \leftarrow Q_{12}^T A_j$  and  $E_{j+1} \leftarrow Q_{22}^T E_j$ ,5:  $s = j + 1$ .6: **end for**7: Use the subspace extraction procedure from [23] in order to compute  $\tilde{U}, \tilde{V}$ .

---

$$\tilde{U}(\lambda E - A)\tilde{V} = \lambda \begin{bmatrix} E_f & W_E \\ 0 & E_\infty \end{bmatrix} - \begin{bmatrix} A_f & W_A \\ 0 & A_\infty \end{bmatrix}, \quad (9)$$

where  $E_f \in \mathbb{R}^{n_f \times n_f}, A_\infty \in \mathbb{R}^{n_\infty \times n_\infty}$  are nonsingular,  $n_f$  is the number of eigenvalues inside the unit circle (here, this equals the number of finite eigenvalues),  $n_\infty := n - n_f$  is the number of infinite eigenvalues, and  $E_\infty \in \mathbb{R}^{n_\infty \times n_\infty}$  is of nilpotency index  $\nu$  which is the index of  $\lambda E - A$ . (Note that in general, if there are also finite eigenvalues outside the unit circle, a block-triangularization is achieved where  $\lambda E_\infty - A_\infty$  contains the finite eigenvalues of modulus larger than 1 and the infinite eigenvalues.)

Algorithm 2 is based on a generalized power iteration (see [23, 24] for more details) and the fact that (see [21, 24])

$$\lim_{j \rightarrow \infty} (A_j + E_j)^{-1} E_j = \mathcal{P}^0, \quad \lim_{j \rightarrow \infty} (A_j + E_j)^{-1} A_j = \mathcal{P}^\infty,$$

where  $\mathcal{P}^0$  and  $\mathcal{P}^\infty$  are projectors onto the right deflating subspaces of  $A - \lambda E$  corresponding to the eigenvalues inside and outside the unit disk  $\mathcal{D}_1(0)$ . Convergence of the algorithm is usually checked based on the relative change in  $R_j$ . Note that the QR decomposition in Step 1 is unique if we choose positive diagonal elements as  $\begin{bmatrix} E_j^T & -A_j^T \end{bmatrix}^T$  has full rank in all steps [25]. The convergence rate of the iteration in Algorithm 2 is globally quadratic [20] with deferred convergence in the presence of eigenvalues very close to the unit circle and stagnation in the limiting case of eigenvalues on the unit circle. Also, the method is proven to be numerically backward stable in [20]. Again, accuracy problems are related to eigenvalues close to the unit circle due to the fact that the spectral decomposition problem becomes ill-conditioned in this case.

It should be noted that for our purposes, neither the disk function nor the projectors  $\mathcal{P}^0$  nor  $\mathcal{P}^\infty$  need to be computed explicitly. All we need are the related matrices  $\tilde{U}, \tilde{V}$  from (9). This only requires orthogonal bases for the range and nullspace of these projectors. These can be obtained using a clever subspace extraction technique proposed in [23]. Due to space limitations, we can not provide further details here.

In order to separate finite from infinite eigenvalues using the disk function method for a stable matrix pencil  $\lambda E - A$ , Algorithm 2 is applied to  $(A, \alpha E)$ , where

$\alpha$  is the radius of a circle, centered at the origin, enclosing the finite eigenvalues of  $\lambda E - A$ . Sometimes,  $\alpha$  can be estimated from the physical background, otherwise a generalization of the Geršgorin circles to matrix pencils [26, 27] may be employed.

**Block-diagonalization.** After block-triangularization as described above, the matrix pencil  $\lambda E - A$  has the form (9). A block-diagonal form can now be obtained using the solution matrices  $Y, Z$  of the generalized Sylvester equation

$$A_f Y + Z A_\infty + W_A = 0, \quad E_f Y + Z E_\infty + W_E = 0. \quad (10)$$

Then

$$\begin{aligned} \lambda \hat{E} - \hat{A} &:= U(\lambda E - A)V := \begin{bmatrix} I & Z \\ 0 & I \end{bmatrix} \tilde{U}(\lambda E - A) \tilde{V} \begin{bmatrix} I & Y \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} I & Z \\ 0 & I \end{bmatrix} \left( \lambda \begin{bmatrix} E_f & W_E \\ 0 & E_\infty \end{bmatrix} - \begin{bmatrix} A_f & W_A \\ 0 & A_\infty \end{bmatrix} \right) \begin{bmatrix} I & Y \\ 0 & I \end{bmatrix} = \lambda \begin{bmatrix} E_f & 0 \\ 0 & E_\infty \end{bmatrix} - \begin{bmatrix} A_f & 0 \\ 0 & A_\infty \end{bmatrix}. \end{aligned} \quad (11)$$

A significant simplification can be observed for matrix pencils of index  $\nu = 1$ : in this case,  $E_\infty = 0$  so that (10) boils down to the subsequent solution of the two linear systems of equations

$$E_f Y = W_E, \quad Z A_\infty = -(W_A + A_f Y). \quad (12)$$

Otherwise, i.e., for  $\nu > 1$ , one can use an appropriate solver for generalized Sylvester equations, e.g., the Fortran 77 subroutine SB04OD from the Subroutine Library in Control Theory (SLICOT)<sup>3</sup> or its MATLAB gateway function `slgesg` from the SLICOT Basic Systems and Control Toolbox in order to solve (10).

### 2.3 *Balanced Truncation for Descriptor Systems*

In this section, we combine the algorithms from the previous two sections in order to derive a method for balanced truncation for descriptor systems. The resulting algorithm is mathematically equivalent to an algorithm proposed in [14, 15], but differs in the underlying computational routines employed. Our method may be more efficient in computing environments where matrix multiplication is very fast compared to the fine-grain computations required in the GUPTRI algorithm [28] employed in [14, 15], while our method may suffer from wrong rank decisions in situations when it is difficult to numerically distinguish finite and infinite eigenvalues of  $\lambda E - A$ .

Employing a minimal realization of  $G_\infty$ , the reduced-order descriptor system becomes  $\hat{G}(s) = \hat{G}_f(s) + G_\infty(s)$ . In [14, 15] it is shown that the order  $\hat{n}_\infty$  of a minimal realization of  $G_\infty$  satisfies  $\hat{n}_\infty \leq \min\{\nu m, \nu p, n_\infty\}$ . In case of  $\nu = 1$ , we get

<sup>3</sup> See [www.slicot.org](http://www.slicot.org).

**Algorithm 3** BT algorithm for descriptor systems.

---

 INPUT: A stable descriptor system realized by  $(A, B, C, D, E)$  as in (1).

 OUTPUT: A stable reduced-order model  $(\hat{A}, \hat{B}, \hat{C}, \hat{D}, \hat{E})$  of order  $r$  satisfying the error bound (7).

- 1: {Compute the additive decomposition of the transfer function.}
  - 2: Compute  $\alpha > 0$  so that  $\Lambda(A, \alpha E) \subset \mathcal{D}_1(0)$ .
  - 3: Apply Algorithm 2 to  $(A, \alpha E)$  in order to block-triangularize  $\lambda E - A$  as in (9).
  - 4: **if**  $\nu = 1$  **then**
  - 5: Solve the linear systems of equations (12).
  - 6: **else**
  - 7: Solve the generalized Sylvester equation (10).
  - 8: **end if**
  - 9: Compute the block-diagonalization as in (11).
  - 10: Apply the resulting system equivalence transformation in order to obtain (8).
  - 11: {Compute the reduced-order model.}
  - 12: Apply BT as described in subsection 2.1 to  $G_f$  and obtain  $\hat{G}_f$ .
  - 13: **if**  $\nu = 1$  **then**
  - 14: Set  $\hat{D} := D - C_\infty A_\infty^{-1} B_\infty$  and  $\hat{G}(s) = \hat{G}_f + \hat{D}$ .
  - 15: **else**
  - 16: Compute a minimal realization of  $G_\infty$  and set  $\hat{G}(s) = \hat{G}_f + G_\infty$ .
  - 17: **end if**
- 

$$G_\infty(s) \equiv \hat{D} := D - C_\infty A_\infty^{-1} B_\infty.$$

In case no feed-through term (“D term”) is allowed in the simulation software for which the reduced-order model is generated,  $\hat{G}(s)$  can then be realized as

$$\hat{G}(s) = [\hat{C}, \hat{D}] \left( s \begin{bmatrix} \hat{E} & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} \hat{A} & 0 \\ 0 & -I_m \end{bmatrix} \right)^{-1} \begin{bmatrix} \hat{B} \\ I_m \end{bmatrix}.$$

Procedures for computing a minimal realization of  $G_\infty$  in case of index  $\nu > 1$  can be found in [14, 15] and amount to applying discrete-time balanced truncation with zero error to the polynomial part. The cost of this procedure is in general  $\mathcal{O}(n_\infty^3)$ . It can be reduced if the corresponding discrete Lyapunov equations can be solved for their low-rank factors directly similar to Algorithm 1, see, e.g., [29].

As  $G_\infty$  is not reduced, just a different realization of possibly smaller order is employed so that  $G(s) - \hat{G}(s) = G_f(s) - \hat{G}_f(s)$ , the error bound (7) applies.

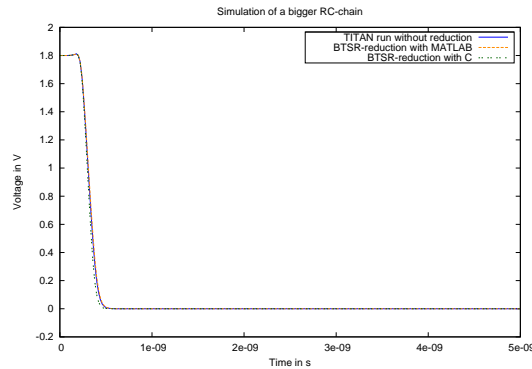
The resulting BT algorithm for descriptor systems is summarized in Algorithm 3.

## 2.4 Numerical Examples

Algorithm 3 was implemented as C subroutine in the circuit simulator TITAN<sup>4</sup> [30]. In the following, we will present simulation results for two examples provided by

---

<sup>4</sup> Copyrighted software, developed by Qimonda AG, München



**Fig. 1** TITAN simulation results for small nonlinear circuit, 1 linear subcircuit ( $n = 297$ ) replaced by reduced-order model ( $r = 31$ ).

Qimonda AG, München, obtained by using reduced-order circuits computed by this subroutine within TITAN.

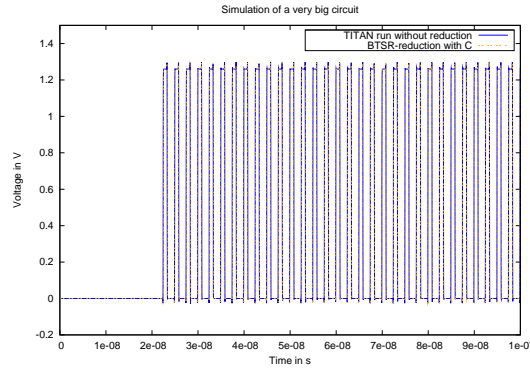
In the first example, a small nonlinear circuit model, designed for testing and verification of algorithms, is used. The circuit consists of 297 resistors, 268 capacitors, 4 voltage sources, and 8 MOSFETs. A linear subcircuit of order  $n = 297$  was extracted and replaced by a model of order  $r = 31$  computed by Algorithm 3. Simulation results are shown in Fig. 1. The figure shows results obtained by a MATLAB implementation of Algorithm 3 developed by the author and the C implementation from [30]. A slightly larger error results from using the C version which hints to an unresolved bug in the software. This is under current investigation.

In the second example, an industrial example with 14,677 resistors, 15,404 capacitors, 14 voltage sources, and 4,800 MOSFETs was investigated. The analysis showed that 14 subcircuit of varying order could be extracted and reduced. Simulation results for the original circuit and for a model where the 14 linear subcircuits were replaced by BT reduced-order models are shown in Fig. 2. Here we see again that the reduced-order model behaves well in time domain simulation.

### 3 Further developments

Besides the aspect of singular  $E$  often arising in circuit simulation, a number of further issues need to be addressed when applying BT for MOR in this area.

**Large-scale, sparse systems.** Large-scale Lyapunov equations can nowadays be solved by using, e.g., the low-rank ADI method, at a cost that scales with the cost of solving linear systems of equations with coefficient matrices  $A - \mu E$ ; see the surveys [10, 31] and references therein. Thus, BT can be implemented at a cost proportional to Krylov-subspace based methods. Usually more sparse factorizations



**Fig. 2** TITAN simulation results for industrial circuit, 14 linear subcircuits are reduced.

have to be computed using ADI methods, but the resulting MOR method has the advantageous properties of BT<sup>5</sup>. The ADI method for Lyapunov equations can also be extended to descriptor systems, see [32].

**Sparsification of reduced-order models.** BT is often criticized for producing dense reduced-order models. (Note: this is also true for most moment-matching methods like PRIMA, except for PVL-like methods.) Mostly, reduced-order models are used when solving linear systems of equations of the form

$$(\iota\omega\hat{E} - \hat{A})x = b \text{ in frequency-domain analysis,} \quad (13)$$

$$(\hat{E} - h_k\hat{A})x_{k+1} = \hat{E}x_k + \dots \text{ in implicit integrators (transient analysis,...).} \quad (14)$$

The cost for solving the linear systems may not benefit from the smaller order, if efficient sparse direct solvers for the full-size sparse system matrices are available.

A significant reduction can be achieved by transforming  $(\hat{A}, \hat{E})$  to Hessenberg-triangular form [25, Algorithm 7.7.1], i.e., compute orthogonal  $Q, Z$  such that

$$Q(\lambda\hat{E} - \hat{A})Z = \lambda \begin{bmatrix} \diagdown & & \\ & \diagdown & \\ & & \diagdown \end{bmatrix} - \begin{bmatrix} \diagdown & & \\ & \diagdown & \\ & & \diagdown \end{bmatrix} \equiv \begin{bmatrix} \diagdown & & \\ & \diagdown & \\ & & \diagdown \end{bmatrix}.$$

The new reduced-order system is then  $(Q\hat{A}Z, Q\hat{B}, \hat{C}Z, \hat{D}, Q\hat{E}Z)$ , the linear systems of equations (13) and (14) then have Hessenberg form, and can thus be solved using  $r - 1$  Givens rotations only! This only requires the introduction of a dedicated solver for Hessenberg systems in the simulation software.

**Passivity preservation.** An important physical property in circuit simulation is passivity as, e.g., RLC circuits only contain passive devices. Thus, the reduced-order model should preserve this property. For symmetric transfer functions as they are usually encountered in RLC circuit models, BT automatically preserves passivity.

<sup>5</sup> Despite unavoidable errors, loss of the theoretical properties of BT is usually not observed in practice.

Other possibilities are balancing-related methods such as PRBT, see [8, 9] and references therein. A number of recent papers deal with the efficient implementation of PRBT, see [9] for a review. Current efforts are directed towards extending the method to large-scale descriptor systems with sparse coefficient matrices and employing the structure of circuit matrices more efficiently so that explicit computation of projectors can be avoided<sup>6</sup>.

**Synthesis.** BT variants based on split-congruence transformations as in [33] are under current investigation. As *split-congruence BT* preserves reciprocity of the transfer function, this allows synthesis of the reduced-order model as circuit. The basic idea here is to exploit the structure of RLC circuits, leading to a “symmetric” transfer function with (for networks without voltage sources)

$$sE - A = s \begin{bmatrix} E_1 & 0 \\ 0 & E_2 \end{bmatrix} + \begin{bmatrix} A_1 & A_2^T \\ -A_2 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} = C^T, \quad D = 0,$$

where  $A_1, E_1 \geq 0$ ,  $E_2 > 0$ . This structure can be preserved in the reduced-order model if the BT truncation matrices  $\hat{L}, \hat{T}$  are embedded in a so-called split-congruence transformation [33]. The mathematical properties of this approach are not clear yet; we will report on this BT variant in the future.

**Acknowledgements** The author thanks René Günzel for providing some of the simulation results and graphics contained in Section 2.4 and Dr. Georg Denk from Qimonda AG, München, Germany, for providing the circuit models used in these simulations.

## References

1. Tan, S., He, L.: Advanced Model Order Reduction Techniques in VLSI Design. Cambridge University Press, New York (2007)
2. Freund, R.: Model reduction methods based on Krylov subspaces. *Acta Numerica* **12**, 267–319 (2003)
3. Feldmann, P., Freund, R.: Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **14**, 639–649 (1995)
4. Gallivan, K., Grimme, E., Van Dooren, P.: Asymptotic waveform evaluation via a Lanczos method. *Appl. Math. Lett.* **7**(5), 75–80 (1994)
5. Odabasioglu, A., Celik, M., Pileggi, L.: PRIMA: passive reduced-order interconnect macro-modeling algorithm. In: *Tech. Dig. 1997 IEEE/ACM Intl. Conf. CAD*, pp. 58–65. IEEE Computer Society Press (1997)
6. Benner, P., Mehrmann, V., Sorensen, D. (eds.): Dimension Reduction of Large-Scale Systems, *Lecture Notes in Computational Science and Engineering*, vol. 45. Springer-Verlag, Berlin/Heidelberg, Germany (2005)
7. Schilders, W., van der Vorst, H., J.Rommers: Model Order Reduction: Theory, Research Aspects and Applications. Springer-Verlag, Berlin, Heidelberg (2008)
8. Antoulas, A.: Approximation of Large-Scale Dynamical Systems. SIAM Publications, Philadelphia, PA (2005)

<sup>6</sup> See <http://www.math.tu-berlin.de/~stykel> for a number of recent preprints dealing with this issue.

9. Benner, P., Faßbender, H.: Numerische Methoden zur passivitätserhaltenden Modellreduktion. *at-Automatisierungstechnik* **54**(4), 153–160 (2006)
10. Benner, P.: Numerical linear algebra for model reduction in control and simulation. *GAMM Mitt.* **29**(2), 275–296 (2006)
11. Benner, P., Quintana-Ortí, E., Quintana-Ortí, G.: Parallel model reduction of large-scale linear descriptor systems via Balanced Truncation. In: M. Daydé, J. Dongarra, V. Hernández, J. Palma (eds.) *High Performance Computing for Computational Science – VECPAR 2004*, no. 3402 in LNCS, pp. 340–353. Springer-Verlag, Berlin/Heidelberg (2005)
12. Moore, B.: Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Control* **AC-26**, 17–32 (1981)
13. Tombs, M., Postlethwaite, I.: Truncated balanced realization of a stable non-minimal state-space system. *Internat. J. Control* **46**(4), 1319–1330 (1987)
14. Stykel, T.: Analysis and numerical solution of generalized Lyapunov equations. Dissertation, TU Berlin (2002)
15. Stykel, T.: Gramian-based model reduction for descriptor systems. *Math. Control Signals Systems* **16**(4), 297–319 (2004)
16. Penzl, T.: Numerical solution of generalized Lyapunov equations. *Adv. Comp. Math.* **8**, 33–48 (1997)
17. Benner, P., Quintana-Ortí, E.: Solving stable generalized Lyapunov equations with the matrix sign function. *Numer. Algorithms* **20**(1), 75–100 (1999)
18. Benner, P., Claver, J., Quintana-Ortí, E.: Efficient solution of coupled Lyapunov equations via matrix sign function iteration. In: A. Dourado et al. (ed.) *Proc. 3<sup>rd</sup> Portuguese Conf. on Automatic Control CONTROLO'98*, Coimbra, pp. 205–210 (1998)
19. Kågström, B., Van Dooren, P.: A generalized state-space approach for the additive decomposition of a transfer matrix. *J. Numer. Linear Algebra Appl.* **1**(2), 165–181 (1992)
20. Bai, Z., Demmel, J., Gu, M.: An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems. *Numer. Math.* **76**(3), 279–308 (1997)
21. Malyshev, A.: Parallel algorithm for solving some spectral problems of linear algebra. *Linear Algebra Appl.* **188/189**, 489–520 (1993)
22. Benner, P., Byers, R.: Disk functions and their relationship to the matrix sign function. In: *Proc. European Control Conf. ECC 97*, Paper 936. BELWARE Information Technology, Waterloo, Belgium (1997). CD-ROM
23. Sun, X., Quintana-Ortí, E.: Spectral division methods for block generalized Schur decompositions. *Mathematics of Computation* **73**, 1827–1847 (2004)
24. Benner, P.: Contributions to the Numerical Solution of Algebraic Riccati Equations and Related Eigenvalue Problems. Logos-Verlag, Berlin, Germany (1997). *Also*: Dissertation, Fakultät für Mathematik, TU Chemnitz-Zwickau, 1997.
25. Golub, G., Van Loan, C.: *Matrix Computations*, third edn. Johns Hopkins University Press, Baltimore (1996)
26. Stewart, G.: Gershgorin theory for the generalized eigenvalue problem  $Ax = \lambda Bx$ . *Math. Comp.* **29**, 600–606 (1975)
27. Kostić, V.: Eigenvalue localization for matrix pencils. Presented at workshop *Applied Linear Algebra — in honor of Ivo Marek. Novi Sad, April 28–30, 2008*
28. Demmel, J., Kågström, B.: The generalized Schur decomposition of an arbitrary pencil  $A - \lambda B$ : Robust software with error bounds and applications. Part I: Theory and algorithms. *ACM Trans. Math. Software* **19**, 160–174 (1993)
29. Benner, P., Quintana-Ortí, E., Quintana-Ortí, G.: Parallel algorithms for model reduction of discrete-time systems. *Int. J. Syst. Sci.* **34**(5), 319–333 (2003)
30. Günzel, R.: Balanced truncation for descriptor systems arising in interconnect modeling. Diplomarbeit, Fakultät für Mathematik, TU Chemnitz, D-09107 Chemnitz, FRG (2008)
31. Gugercin, S., Li, J.R.: Smith-type methods for balanced truncation of large systems. Chapter 2 (pages 49–82) of [6]
32. Stykel, T.: Low-rank iterative methods for projected generalized Lyapunov equations. *Electr. Trans. Num. Anal.* **30**, 187–202 (2008)
33. Kerns, K., Yang, A.: Preservation of passivity during RLC network reduction via split congruence transformations. *IEEE Trans. CAD Integr. Circuits Syst.* **17**(7), 582–591 (1998)