11-1-2000

Efficient Balance-and-Truncate Model Reduction for Large Scale Systems

Venkataramanan Balakrishnan
Purdue University School of Electrical and Computer Engineering

Qing Su
Purdue University School of Electrical and Computer Engineering

Cheng-Kok Koh
Purdue University School of Electrical and Computer Engineering

EFFICIENT BALANCE-AND-TRUNCATE MODEL REDUCTION FOR LARGE SCALE SYSTEMS

VENKATARAMANAN BALAKRISHNAN
QING SU
CHENG-KOK KOH

TR-ECE 00-15
NOVEMBER 2000

SCHOOL OF ELECTRICAL
AND COMPUTER ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE, INDIANA 47907-1285

Efficient Balance-and-Truncate Model Reduction for Large Scale Systems

Venkataramanan Balakrishnan
Qing Su
Cheng-Kok Koh

School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907–1285
Email: {ragu, qsu,chengkoh)@ecn.purdue.edu

Abstract

We present efficient implementations of the balance-and-truncate model reduction technique for large-scale systems. The key observation that distinguishes our approach is that Krylov subspace methods (Arnoldi and Lanczos) directly yield approximate low-rank square roots[1] of the system Gramians; the balancing transformation can be then constructed from these square roots, obviating the need for solving any Lyapunov equations. In addition, the order of the reduced model is not fixed a ***priori*** as with some existing methods, but is determined from the problem data. Numerical simulations show that our approach performs very well over a range of examples, and offers considerable savings in practice.

'We use the term "square root" to mean the not necessarily symmetric square root of a matrix: If $M = M^T = NN^T$, we say N is the square root of M.

1 Introduction

As engineering systems become more and more complex, so do the mathematical models describing them. This is true, for instance, when an engineering system is modeled as an interconnection of a large number of sub-systems, as with VLSI circuit models. Computer-aided tools are typically used to model the sub-systems; the resulting model of the overall system can involve thousands of variables.

The analysis and design of large-scale systems can stretch the limits of computing resources. Indeed, the mere simulation of a large-scale system can require an unacceptably long computation time. A standard practice that addresses such issues is that of model reduction: The objective is to find an approximate model of the original system with far fewer variables. Our objective is to present efficient algorithms for the model reduction of large-scale linear time-invariant (LTI) state-space models.

Model reduction of LTI systems is a well-studied topic. One approach is to expand the transfer function as a power series around a suitable point in the complex plane, and obtain a lower order model whose power series coefficients match the first few original coefficients ("moment-matching"). A well-known example of such an approach is Pad6 approximation, which can be shown to be optimal in a certain sense. Another model-reduction approach involves truncating the state space to the principal controllable subspace, or the principal observable subspace. Well-conditioned and efficient implementation of these two techniques has been the subject of considerable investigation. Model reduction through moment-matching is the subject of the asymptotic waveform evaluation (AWE) technique due to Huang, Pillage. and Rohrer [Hua90, PR90]. Pad6 approximations through moment matching are described in [CN92]; a discussion of the numerical properties of these algorithms can be found in [CC87]. For state-space models, Krylov subspace computation techniques such as the Arnoldi and the Lanczos iterations have proved to be very attractive. Krylov methods require only matrix-vector multiplications, and are therefore particularly useful for large-scale systems. For Pad6 approximation, the use of Lanczos iterations is discussed in [FF95, GGD94], and the use of Arnoldi iterations is discussed in [SKW95]. Krylov methods have been used to efficiently compute a basis for the principal controllable and observable subspaces; see [Fre98]. The reduced-order model can be then obtained by projecting the state-space on these subspaces [GG97, GN99].

A third technique, one that underlies the approach presented in this paper, is the balance-and-truncate method (see for example, [Moo81, GL95]). The idea here is to find a state-space coordinate transformation in which the input-to-state map and the state-to-output map are "aligned". (This is the so-called balanced realization.) In the balanced coordinates, state variables are ordered by the ease with which they are simultaneously reachable from the input and observable from the output. Thus, state-variables that are not easy to reach and not easily observed can the omitted (or the model truncated). An attractive feature of the balance-and-truncate method is that the approximation error can be shown to be bounded [Enn84, Glo84]. While the balance-and-truncate method is theoretically attractive and also yields excellent approximate models in practice, its use for the model-reduction of large-scale systems has been hampered by its quickly growing computational demand: Two large-size Lyapunov equations need to be solved, followed by a large-size eigen-decomposition. One approach towards addressing this issue is to obtain low-rank approximate solutions to the large-size Lyapunov equations, for instance, the "Alternate Direction Iteration" or ADI approach [LW91] and its modifications [LWW99, LW99]. The drawback of the ADI approach is the requirement that the original system matrix be tridiagonalized first; this step is both computationally demanding and possibly numerically ill-conditioned [GL89, §9.3.6]. Another prevalent approach for balance-and-truncate model reduction is to use Krylov subspace computation methods to first find the principal controllable or observable subspace, and then solve reduced-order Lyapunov equations to proceed with balancing and truncating; see, for example, [JKL92, JK94], and the reference therein.

Our contribution is an algorithm for balance-and-truncate model reduction, using Krylov methods, where no Lyapunov equations need solution. The key observation is the following: *Krylov methods have thus far been used only to obtain a basis for the principal controllable and/or observable subspace; however, it turns out that the Arnoldi and Lanczos iterations directly yield approximate low-rank square roots of the system Gramians; the balancing transformation can be then constructed from these square roots.* In addition, the order of the reduced model can be determined from the problem data. In Section 2, we introduce the mathematical framework underlying our approach, including an analysis of the approximation error. We also provide the description of the algorithm. In Section **3,** we present a few examples that illustrate that our approach requires greatly reduced computation.

2 Mathematical Framework

2.1 Balanced Transformation and Truncation

Consider the linear system described by the state-space equations

$$\dot{x} = Ax + Bu, \tag{1a}$$

$$y = Cx + Du, \tag{1b}$$

where $x(t) \in$ 1R$^{\mathrm{N}}$, $u(t) \in \mathbb{R}$ and $y(t) \in \mathbb{R}$, and A, B, C and D are real matrices of appropriate sizes. (We will consider only single-input single-output systems in this paper; the extension of the results presented herein to multi-input multi-output systems is straightforward.) We will use ordered quadruple $(A, \mathrm{B}, \mathrm{C}, D)$ to denote the state-space realization of the system. We assume that A is stable, i.e., all of its eigenvalues have negative real part, and that the realization is minimal. The objective in model reduction is to obtain another linear system

$$\dot{x}_{\mathrm{red}} = A_{\mathrm{red}} x_{\mathrm{red}} + B_{\mathrm{red}} u, \tag{2a}$$

$$y_{\mathrm{red}} = C_{\mathrm{red}} x_{\mathrm{red}} + Du, \tag{2b}$$

where $x_{\mathrm{red}}(t) \in \mathbb{R}^n$, with n $\ll$ N, and with the mapping from u to y well-approximated by the mapping from u to y_{red}.

Balanced truncation is one well-known model reduction scheme. The first step is to compute the controllability and observability Gramians, denoted W_c and W_o respectively, and defined as

$$W_c = \int_0^{W} e^{At}BB^Te^{A^Tt}\,dt, \qquad W_o = \int_0^{\infty} e^{A^Tt}C^TCe^{At}\,dt.$$

The Gramians satisfy the Lyapunov equations

$$AW_c + W_cA^T + BB^T = 0, \qquad A^TW_o + W_oA + C^TC = 0.$$

The eigenvalues and eigenvectors of W_c can be shown to be the square of the singular values and the corresponding right singular vectors of the mapping from input u to state x. Therefore, with the eigenvalues sorted in descending order, the eigenvectors of W_c yield directions in state-space that are increasingly hard to reach with the input u. The eigenvalues and eigenvectors of W_o can be shown to be the square of the singular values and the corresponding left singular vectors of the mapping from state x to output y. Therefore, with the eigenvalues sorted in descending order, the eigenvectors of W_o yield directions in state-space that are increasingly hard to observe from the output y.

Let $W_c = \mathrm{XX}^\mathrm{T}$ and $W_o = \mathrm{YY}^\mathrm{T}$ be square root decompositions. Let

$$X^TY = U\Sigma V^T$$

be a singular value decomposition (SVD), so that U, V $\in \mathbb{R}^{N\times N}$ are orthogonal, and $\Sigma > 0$ is diagonal, with the diagonal entries in descending order. The diagonal entries of Σ are called the Hankel singular values of the system.

Define

$$T_b = XU\Sigma^{-\frac{1}{2}} = \left(\Sigma^{-\frac{1}{2}}V^TY^T\right)^{-1}\cdot$$

In the new coordinates $\bar{x} = T_b^{-1}x$, the state-space realization is $(T_b^{-1}AT_b, T_b^{-1}\mathrm{B}, CT_b, \mathrm{D}) \triangleq (\bar{A}, \bar{B}, \bar{C}, \mathrm{D})$. It is easily verified that the corresponding controllability and observability Gramians are

$$\bar{W}_c = \bar{W}_o = \Sigma.$$

Thus, in the new state-space coordinates, the state components are as reachable from the input as they are observable at the output. Moreover, when a diagonal value of Σ is large (respectively, small), the corresponding state component is both very (respectively, not very) reachable from the input, and very (respectively, not very) observable at the output. This motivates the next step, that of "truncation" of the state-vector, i.e., simply "throwing away" state components for which the corresponding diagonal entry a; of Σ is small. If

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \gg \sigma_{n+1} \geq \cdots \geq \sigma_N,$$

the balance-and-truncate model reduction corresponds to that is,

$$\bar{x}_{\text{red}} = \begin{bmatrix} I_n & 0_{n\times(N-n)} \end{bmatrix}\bar{x} = \begin{bmatrix} I_n & 0_{n\times(N-n)} \end{bmatrix}T_b^{-1}x.$$

The approximation error with the balance-and-truncate model reduction is well understood, and we describe it next. The state-space realization of the reduced order model is given by $(A_{\text{red}}, B_{\text{red}}, C_{\text{red}}, D)$, where

$$A_{\text{red}} = \begin{bmatrix} I_n & 0_{n\times(N-n)} \end{bmatrix} \bar{A} \begin{bmatrix} I_n \\ 0_{(N-n)\times n} \end{bmatrix}, \ \text{Bred} = \begin{bmatrix} I_n & 0_{n\times(N-n)} \end{bmatrix} \bar{B},\ C_{\text{red}} = \bar{C} \begin{bmatrix} I_n \\ 0_{(N-n)\times n} \end{bmatrix}.$$

Let $H(s) = C(sI - A)^{-1}B +$ D and $H_{\text{red}}(s) = C_{\text{red}}(sI - A_{\text{red}})^{-1}B_{\text{red}} +$ D denote the transfer functions of the original and the reduced-order system respectively. Then, it can be shown that

$$\sup_{\omega\in\mathbb{R}} \|H(j\omega) - H_{\text{red}}(j\omega)\| \le 2\sum_{n+1}^{N} \sigma_i.$$

While the approximation properties of the balance-and-truncate model reduction algorithm are excellent, its use for large-scale systems is limited by the heavy computational demand: Two large-size Lyapunov equations need to be solved, followed by one large-size SVD computation. In many practical implementations, the balance-and-truncate step forms a second step of an overall model reduction algorithm: First, the dimension of the state-space is reduced, for instance, by projecting the state-space on the principal controllable and/or the principal observable subspace. Specifically, if Q_k is an N × k matrix with orthogonal columns spanning the principal k-dimensional controllable subspace, the idea is to obtain a kth order state-space realization $(Q_k^T A Q_k, Q_k^T B, C Q_k, D)$. Then this is followed by a small-size balance-and-truncate model reduction. In contrast, the approach that we propose herein is a direct approach: We will describe algorithms that directly compute low-rank square roots of the Gramians. We will then show how these factors can be combined to yield "approximate" balancing transformations that automatically truncate the state space. The idea of computing low-rank approximations can be found in [Fre98, JK94, LW91]. However, these approximations have only been used to find the (approximate) principal controllable and/or principal observable subspace, and not to find the (approximate) balanced coordinates; this is one of the distinguishing features of our work.

2.2 Approximate Balanced Truncation

We first describe the idea behind an approximate balance-and-truncate method that relies on low-rank square roots of the Gramians. (We will defer a careful analysis of the approximation error to Section 2.4.) Suppose that we have approximate low-rank square roots of the Gramians, i.e., we have full rank $X_k, Y_k \in \mathbb{R}^{N\times k}$ such that

$$W_c \approx X_k X_k^T, \quad W_o \approx Y_k Y_k^T.$$

Let

$$X_k^T Y_k = \hat{U}\hat{\Sigma}\hat{V}^T$$

be the k × k SVD. Then, the diagonal entries $\hat{\sigma}_i$ of $\hat{\Sigma}$ approximate the first k Hankel singular values of the system. Suppose that

$$\hat{\sigma}_1 \ge \hat{\sigma}_2 \ge \cdots \ge \hat{\sigma}_n \gg \hat{\sigma}_{n+1} \ge \cdots \ge \hat{\sigma}_k.$$

Define

and

$$\tilde{T_b}^{\ddagger} = \left[\begin{array}{cc} I_n & 0_{n\times(N-n)}\end{array}\right] \hat{\Sigma}^{-\frac{1}{2}} \hat{V}^T Y_k^T.$$

Note that $\tilde{T_b} \in$ Kt$^{\mathrm{Nxn}}$ and $\tilde{T_b}^{\ddagger} \in$ Kt$^{\mathrm{nxN}}$, and that $\tilde{T_b}^{\ddagger}\tilde{T_b} = \mathrm{I}$.

Consider the nth order system with state-space realization $(\tilde{T_b}^{\ddagger}A\tilde{T_b}, \tilde{T_b}^{\ddagger}B, C\tilde{T_b}, D)$. It can be verified that the controllability and observability Gramians for this realization are

$$\bar{W}_c \approx \mathbf{diag}(\hat{\sigma}_1, \ldots, \hat{\sigma}_n) \approx \bar{W}_o.$$

Thus, the matrices $\tilde{T_b}$ and $\tilde{T_b}^{\ddagger}$ directly provide for an approximate balance-and-truncate state-space model reduction.

When k $\ll$ N, the major computational task underlying the implementation of the forementioned approach is that of efficient calculation of low-rank square roots X_k and Y_k for the Gramians W_c and **W,.** Our next contribution is to describe an efficient algorithm for directly computing these low-rank square roots using Krylov methods.

2.3 Square Roots of the Gramians via Krylov Methods

For every real scalar p < 0, the equation

$$AW_c + W_cA^T + BB^T = 0$$

is equivalent to

$$(pI+A)^{-1}(pI-A)W_c(pI-A)^T(pI+A)^{-T} - W_c - 2p(pI+A)^{-1}BB^T(pI+A)^{-T} = 0.$$

Defining $A_p \triangleq (pI + A)^{-1}(pI -$ A), and $B_p \triangleq \sqrt{(-2p)}(pI + A)^{-1}B$, we then have

$$W_c = \Sigma_{j=0}^{\infty} A_p^j B_p B_p^T (A_p^T)^j \tag{3}$$

Similarly, with $C_p \triangleq \sqrt{(-2p)}C(pI + A)^{-1}$, we have

$$W_o = \Sigma_{j=0}^{\infty} (A_p^T)^j C_p^T C_p A_p^j. \tag{4}$$

One interpretation of these steps is that we have derived a discrete-time system with state-space realization (A_p, B_p, C_p, D_p), that has the same Gramians as the continuous system, using the conformal mapping $z \mapsto$ (p − s) / (∔ s) that maps the the transfer function $H(s) = C(sI -$ A)−'B + D to $H(z) = C_p(zI - A_p)^{-1}B_p + D_p$. Then, equations **(3)** and (4) are simply the expressions for the Gramians for the discrete-time system. (Note that $D_p =$ D − $C(pI +$ A)−'B.)

Equations (3) and (4) suggest a natural way of obtaining low-rank approximations to the square roots of W_c and W_o:

$$\begin{aligned} W_c &\approx \Sigma_{j=0}^{k-1} A_p^j B_p B_p^T (A_p^T)^j \\ &= \mathcal{K}(A_p, B_p, k)\mathcal{K}(A_p, B_p, k)^T \end{aligned}$$

where

$$\mathcal{K}(A_p,B_p,k) \triangleq [B_p \quad A_pB_p \quad A_p^2B_p \quad \cdots \quad A_p^{k-1}B_p]$$

is the kth order Krylov matrix. Similarly,

$$\begin{array}{rcl} W_o & \approx & \Sigma_{j=0}^{k-1}(A_p^T)^jC_p^TC_pA_p^j \\ & = & \mathcal{K}(A_p^T,C_p^T,k)\mathcal{K}(A_p^T,C_p^T,k)^T. \end{array}$$

The direct computation of the N x k matrices $\mathcal{K}(A_p,B_p,k)$ and $\mathcal{K}(A_p^T,C_p^T,k)$ is ill-conditioned, as the columns of these matrices quickly converge to the dominant eigenvector of A, and A_p^T respectively. Krylov methods are natural tools for well-conditioned computation of $\mathcal{K}(A_p,B_p,k)$ and $\mathcal{K}(A_p^T,C_p^T,k)$.

The quality of the approximation of $\mathcal{K}(A_p,B_p,$k) and $\mathcal{K}(A_p^T,C_p^T,k)$ as square roots of the Gramians depends critically on how fast A_p^k goes to zero with k, or on the spectral radius (i.e., the maximum magnitude of the eigenvalues) $\rho(A_p)$ of A,. The choice of p to make $\rho(A_p)$ as small as possible is a well-studied problem; see for example [LW91]. The key here is that the eigenvalues of A and A, are related by $\lambda_i(A_p) =$ (p − $\lambda_i(A))/(p + \lambda_i(A))$. For every i, the value of p that minimizes $|(p-\lambda_i(A))/(p+\lambda_i(A))|$ is p = $|\lambda_i(A)|$. Of course, we need to choose p to minimize

$$\max_i |(p-\lambda_i(A))/(p+\lambda_i(A))|.$$

As discussed in [LW91], a good choice for p is simply $-\sqrt{(\max_i|\lambda_i(A)|)\,(\text{mini}\,|\lambda_i(A)|)}$.

In our implementation, we used ten power iterations to compute $\max_i|\lambda_i(A)|$ and ten inverse power iterations to compute $\min_i|\lambda_i(A)|$. The implementation of the power iterations is straightforward, and requires only matrix-vector multiplications. To implement the inverse power iterations, we began with an LU factorization of A; then every inverse power iteration required the solution of two triangular systems of linear equations.

We next discuss the Arnoldi and Lanczos iterations that compute the Krylov matrices $\mathcal{K}(A_p,B_p,k)$ and $\mathcal{K}(A_p^T,C_p^T,k)$ in an efficient and well-conditioned manner.

2.3.1 Arnoldi Method

The Arnoldi iterations [GL89, Ch. 9] can be used to iteratively compute the columns of matrices Q_k and P_k, such that

- $Q_k^TQ_k = I = P_k^TP_k$;
- the columns of Q_k and P_k span the range of $\mathcal{K}(A_p,B_p,k)$ and $\mathcal{K}(A_p^T,C_p^T,k)$;
- $Q_k^TA_pQ_k = H_k$ and $P_k^TA_p^TP_k = F_k$ are Hessenberg matrices;
- $$\begin{array}{rcl} \mathcal{K}(A_p,B_p,k) & = & [B_p \quad A_pB_p \quad A_p^2B_p \quad \cdots \quad A_p^{k-1}B_p] = Q_kR_k, \\ \mathcal{K}(A_p^T,C_p^T,k) & = & [C_p^T \quad A_p^TC_p^T \quad (A_p^T)^2C_p^T \quad \ldots \quad (A_p^T)^{k-1}C_p^T] = P_kS_k \end{array}$$
  are QR factorizations.

The algorithm that we present here is adapted from [GL89, Ch. 9] to compute a QR factorization of both $\mathcal{K}(A_p, \boldsymbol{B},, k)$ and $\mathcal{K}(A_p^T, C_p^T, \mathrm{k})$ inside one iterative loop; in addition, we show explicitly the construction of the QR factors.

```
$j = 1;$
$q_1 = B_p/\|B_p\|_2;\ \beta = 1;\ Q_1 = q_1;\ R_1 = \|B_p\|_2;$
$p_1 = C_p^T/\|C_p\|_2;\ \gamma = 1;\ P_1 = p_1;\ S_1 = \|C_p\|_2;$
repeat while stopping criterion is not met {
        for $i = 1 : j$
                $h_{ij} = q_i^T A_p q_j;$
                $f_{ij} - p_i^T A_p^T p_j;$
        end
        $r_{j+1} = A_p q_j - \Sigma_{i=1}^{j} h_{ij} q_i;$

        $s_{j+1} = A_p^T p_j - \Sigma_{i=1}^{j} f_{ij} p_i;$

        $H_j = \left[\begin{array}{cc} H_{j-1} & \begin{bmatrix} h_{1j} \\ \vdots \\ h_{j-1,j} \end{bmatrix} \\ [0 \ \cdots \ \beta] & h_{jj} \end{array}\right]; \quad R_j = \left[\begin{array}{c|c} \frac{R_{j-1}}{[0\cdots 0]} & H_j \begin{bmatrix} R_{j-1}(:, j-1) \\ 0 \end{bmatrix} \end{array}\right],$
        $F_j = \left[\begin{array}{cc} F_{j-1} & \begin{bmatrix} f_{1j} \\ \vdots \\ f_{j-1,j} \end{bmatrix} \\ [0 \ \cdots \ \gamma] & f_{jj} \end{array}\right]; \quad S_j = \left[\begin{array}{c|c} \frac{S_{j-1}}{[0\cdots 0]} & F_j \begin{bmatrix} S_{j-1}(:, j-1) \\ 0 \end{bmatrix} \end{array}\right];$
        $\beta = \|r_{j+1}\|_2; \quad \gamma = \|s_{j+1}\|_2;$
        $q_{j+1} = r_{j+1}/\beta; \quad p_{j+1} = s_{j+1}/\gamma;$
        $Q_{j+1} = [Q_j \quad q_{j+1}]; \quad P_{j+1} = [P_j \quad p_{j+1}];$
        $j{=}j{+}1;$
}
```

In the above algorithm, matrix-vector multiplications such as $A_p q_j = (pI + A)^{-1}(pI - A)q_j$ can be implemented by first performing an LU factorization of $(pI + A)$; then the product $A_p q_j$ can be computed by performing one matrix-vector multiplication, followed by the solution of two triangular systems of linear equations.

At the end of the kth iteration, we have an approximate kth order square root of W_c, given as $X_k = \mathcal{K}(A_p, B_p, \mathrm{k}) = Q_k R_k$, and an approximate kth order square root of W_o, given as $Y_k = \mathcal{K}(A_p^T, C_p^T, \mathrm{k}) = P_k S_k$. An nth order approximately balance-and-truncate model can be obtained through the following steps:

1. Find the SVD
$$X_k^T Y_k = R_k^T Q_k^T P_k S_k = \hat{U}\hat{\Sigma}\hat{V}^T.$$

2. Determine n, the number of significant state components, from the approximate Hankel singular values

$$\hat{\sigma}_1 \geq \hat{\sigma}_2 \geq \cdots \geq \hat{\sigma}_n \gg \hat{\sigma}_{n+1} \geq \cdots \geq \hat{\sigma}_k.$$

3. Form

$$\tilde{T}_b = X_k \hat{U} \hat{\Sigma}^{-\frac{1}{2}} \begin{bmatrix} I_n \\ 0_{(N-n)\times n} \end{bmatrix}, \text{ and } \tilde{T}_b^{\ddagger} = \begin{bmatrix} I, & 0_{n\times(N-n)} \end{bmatrix} \hat{\Sigma}^{-\frac{1}{2}} \hat{V}^T Y_k^T.$$

4. Compute the nth-order state-space realization $(\tilde{T}_b^{\ddagger} A \tilde{T}_b, \tilde{T}_b^{\ddagger} B, C\tilde{T}_b, D)$.

2.3.2 Lanczos Method

The Lanczos method for non-symmetric matrices is another technique for the computation of a basis for Krylov subspaces. While the Lanczos method involves less computation than the Arnoldi method, its numerical properties are suspect [GL89, §9.3.6]. The Lanczos algorithm reduces the square matrix A, to tridiagonal form using a general similarity transformation. The algorithm interactively computes the bi-orthonormal columns of P_k and Q_k (i.e., with $P_k^T Q_k = I$) such that $P_k^T A_p Q_k = T_k$, with T_k a tridiagonal matrix. Here Q_k spans the range of the Krylov matrix $\mathcal{K}(A_p, B_p, k)$, P_k spans the range of the Krylov matrix $\mathcal{K}(A_p^T, C_p^T, k)$. Also, during the iterations, we can construct the upper-triangular matrices R_k and S_k, such that

$$\mathcal{K}(A_p, B_p, k) = [B_p \quad A_p B_p \quad \cdots \quad A_p^{k-1} B_p] = Q_k R_k$$

and

$$\mathcal{K}(A_p^T, C_p^T, k) = [C_p^T \quad A_p^T C_p^T \quad \cdots \quad (A_p^T)^{k-1} C_p^T] = P_k S_k.$$

(Note that these are not QR factorizations, unlike with the Arnoldi method.)

The algorithm is:

```
$j = 1;$
$q_1 = r_1 = B_p/\|B_p\|_2;\ \beta = 1;\ Q_1 = q_1;\ R_1 = \|B_p\|_2;$
$p_1 = s_1 = C_p^T/\|C_p\|_2;\ \gamma = s_1^T r_1/\beta;\ P_1 = p_1;\ S_1 = \|C_p\|_2;$
repeat while stopping criterion is not met {
    $\alpha = p_j^T A_p q_j;$
    $t_{jj} = \alpha;$
    $T_j = \begin{bmatrix} T_{j-1} & \begin{bmatrix} 0 \\ \vdots \\ \gamma \end{bmatrix} \\ [0 \ \cdots \ \beta] & t_{jj} \end{bmatrix};$
    $R_j = \left[ \begin{array}{c|c} \dfrac{R_{j-1}}{[0 \cdots 0]} & T_j \begin{bmatrix} R_{j-1}(:, j-1) \\ 0 \end{bmatrix} \end{array} \right];$
    $S_j = \left[ \begin{array}{c|c} \dfrac{S_{j-1}}{[0 \cdots 0]} & T_j^T \begin{bmatrix} S_{j-1}(:, j-1) \\ 0 \end{bmatrix} \end{array} \right];$
    $r_{j+1} = (A_p - \alpha I) q_j - \gamma q_{j-1};$
```

```
        $s_{j+1} = (A_p - \alpha I)^T p_j - \beta p_{j-1};$
        $\beta = \|r_{j+1}\|_2;\ \gamma = s_{j+1}^T r_{j+1}/\beta;$
        $q_{j+1} = r_{j+1}/\beta;\ p_{j+1} = s_{j+1}/\gamma;$
        $Q_{j+1} = [Q_j \quad q_{j+1}];\ P_{j+1} = [P_j \quad p_{j+1}];$
        $j = j + 1;$
    }
```

As with the Arnoldi method, we have an approximate lcth order square root of W_c, given as $X_k = \mathcal{K}(A_p, B_p, k) = Q_k R_k$, and an approximate kth order square root of W_o, given as $Y_k = \mathcal{K}(A_p^T, C_p^T, k) = P_k S_k$. Approximate balanced truncation can proceed as described at the end of Section 2.3.1. Note that with the Lanczos iteration $X_k^T Y_k = R_k^T Q_k^T P_k S_k = R_k^T S_k$, or the Lanczos iterations yield approximate rank-lc LU factors for the product $X^T Y$; in this sense, the Lanczos method is a more natural method for use with balancing.

2.4 Error Analysis, Stopping Criterion and Flop Counts

We first analyze the error in approximating the Gramians. Consider the error $E_{k,c} = W_c - X_k X_k^T$. Clearly $E_{k,c} \geq 0$ for all k, so that $\mathbf{Tr}\,(E_{k,c})$ serves as a norm of the error. Now,

$$\begin{array}{rcl} \mathbf{Tr}\,(E_{k,c}) & = & \mathbf{Tr}\,\left(\sum_{i=k}^{\infty} A_p^i B_p B_p^T (A_p^T)^i\right) \\ & = & \sum_{i=k}^{\infty} \|A_p^i B_p\|_2^2. \end{array}$$

Thus, the error converges monotonically to zero with k. Moreover,

$$\begin{array}{rcl} \sum_{i=k}^{\infty} \|A_p^i B_p\|_2^2 & \leq & K_c \rho(A_p)^k \sum_{i=0}^{\infty} \|A_p^i B_p\|_2^2 \\ & = & K_c \rho(A_p)^k \mathbf{Tr}\,(W_c) \end{array}$$

for some constant K_c. Thus,

$$\mathbf{Tr}\,(W_c - X_k X_k^T) \leq K_c \rho(A_p)^k \mathbf{Tr}\,(W_c), \text{ and similarly } \mathbf{Tr}\,(W_o - Y_k Y_k^T) \leq K_o \rho(A_p)^k \mathbf{Tr}\,(W_o),$$

or the relative error in the approximation depends critically on the spectral radius of **A,.**

Finally, we may derive an expression for the error in the approximation of the Hankel singular values themselves. Recall that with $W_c = \boldsymbol{XX^T}$ and $W_o = YY^T$, the Hankel singular values are simply the singular values σ_i of $\boldsymbol{X^T Y}$. Our algorithm yields k approximate Hankel singular values $\hat{\sigma}_i$, via an SVD of $X_k^T Y_k$. Then,

$$\|X^T Y\|_F^2 = \mathbf{Tr}\,(Y^T X X^T Y) = \sum_{i=1}^{N} \sigma_i^2, \quad \text{and} \quad \|X_k^T Y_k\|_F^2 = \mathbf{Tr}\,(Y_k^T X_k X_k^T Y_k) = \sum_{i=1}^{k} \hat{\sigma}_i^2.$$

Moreover,

$$\begin{array}{rcl} \mathbf{Tr}\,(Y^T X X^T Y - Y_k^T X_k X_k^T Y_k) & = & \mathbf{Tr}\,\left(\left(\sum_{i=k}^{\infty} A_p^i B_p B_p^T (A_p^T)^i\right)\left(\sum_{i=0}^{k-1} (A_p^T)^i C_p^T C_p A_p^i\right)\right) \\ & & +\mathbf{Tr}\,\left(\left(\sum_{i=k}^{\infty} (A_p^T)^i C_p^T C_p A_p^i\right)\left(\sum_{i=0}^{k-1} A_p^i B_p B_p^T (A_p^T)^i\right)\right) \\ & & +\mathbf{Tr}\,\left(\left(\sum_{i=k}^{\infty} (A_p^T)^i C_p^T C_p A_p^i\right)\left(\sum_{i=k}^{\infty} A_p^i B_p B_p^T (A_p^T)^i\right)\right) \\ & & \\ & \leq & K \rho(A_p)^{2k} \|W_c\|_F \|W_o\|_F \end{array}$$

for some constant K. Once again, the approximation error depends critically on the spectral radius of **A,.**

Stopping Criterion

One practical stopping criterion with both the Arnoldi and Lanczos iterations is to monitor the Frobenius norm of the product $X_k^T Y_k$, and to stop when the change is smaller than some tolerance. The quantity $\|X_k^T Y_k\|_F$ can be computed iteratively:

$$\begin{aligned}
&\|X_k^T Y_k\|_F \\
&= \left(\begin{array}{l} \mathbf{Tr}\ \left(X_{k-1}X_{k-1}^T Y_{k-1} Y_{k-1}^T + X_{k-1}X_{k-1}^T (A_p^T)^{k-1} C_p^T C_p A_p^{k-1}\right) \\ +\mathbf{Tr}\ \left(A_p^{k-1} B_p B_p^T (A_p^T)^{k-1} Y_k Y_k^T\right) + \|A_p^{k-1} B_p\|_2^2 \|(A_p^T)^{k-1} C_p^T\|_2^2 \end{array} \right)^{\frac{1}{2}} \\
&= \sqrt{\|X_{k-1}^T Y_{k-1}\|_F^2 + \|X_{k-1}^T (A_p^T)^{k-1} C_p^T\|_2^2 + \|Y_{k-1}^T A_p^{k-1} B_p\|_2^2 + \|A_p^{k-1} B_p\|_2^2 \|(A_p^T)^{k-1} C_p^T\|_2^2}.
\end{aligned}$$

The latter three terms require only matrix-vector multiplications.

Flop Counts

The flop counts of the major steps involved in the various implementations of the balance-and-truncate method are listed below. The number of states in the full-order model is N, and we assume that the reduced-order model has n states. We assume that k iterations are performed with the Arnoldi- and Lanczos-based approximate balance-and-truncate implementations.

Technique	Flop count
Standard balance-and-truncate	$(30 + \frac{2}{3})N^3 + 6\mathrm{n}\mathrm{N}^2 + 6\mathrm{n}^2\mathrm{N} + 4\mathrm{n}\mathrm{N}$
Balance-and-truncate with Arnoldi	$\frac{4}{3}N^3 + (40 + 10k + 2\mathrm{n})\mathrm{N}^2 + (10\mathrm{k}^2 + 2\mathrm{n}^2 + 4\mathrm{k} + 4\mathrm{n} + 4kn)N$ $+4\mathrm{n}^2\mathrm{k} + (22 + \frac{4}{3})k^3 + 6\mathrm{k}^2 - \frac{7}{3}k$
Balance-and-truncate with Lanczos	$\frac{4}{3}N^3 + (40 + 10k + 2\mathrm{n})\mathrm{N}^2 + (6k^2 + 2\mathrm{n}^2 + 10k + 4\mathrm{n} + 4kn)N$ $4\mathrm{n}^2\mathrm{k} + 22k^3 + 2\mathrm{k}^2 - k$

3 Numerical Results

We now demonstrate the performance of the model reduction schemes described thus far on some numerical examples. For the first set of test cases, we considered randomly generated stable LTI systems with twenty pairs of eigenvalues with a real part of -1, twenty pairs of eigenvalues with a real part of -2, with the remaining eigenvalues having smaller (more negative) real parts. We considered full-order models of three different sizes: 100,200 and 400. For each size, we generated thirty different test cases, and applied our model reduction schemes to obtain an approximately balanced-and-truncated reduced order model. Table 1 shows statistics describing the performance of our algorithm. It is evident that with our algorithm, considerable computational savings accrue as compared with the standard balance-and-truncate model reduction.

In order to illustrate the error in approximation, we consider a typical test case of a full-order model with 100 states. Our algorithm yielded a reduced-order model with 21 states. Figure 1(a) shows the relative approximation error of the 21-state reduced-order models obtained with the standard balance-and-truncate method, balance-and-truncate with Arnoldi iterations, and balance-and-truncate with Lanczos iterations respectively. It is evident that error performance of our algorithms are comparable with that of the standard balance-and-truncate method. Figure 1(b) show the magnitude and phase of the system response of the

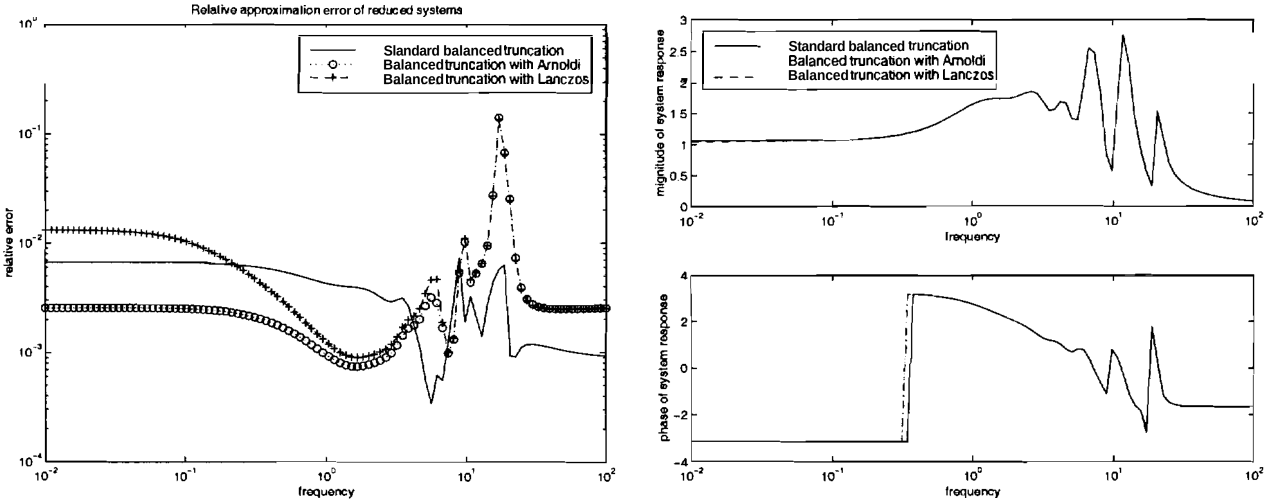
Original model order	100	200	400
Average flop count with the standard balance-truncate method (in millions)	225	1930	16400
Average savings with Arnoldi	46	108	173
Average savings with Lanczos	47	109	174
Maximum savings with Arnoldi	105	193	241
Maximum savings with Lanczos	107	194	241
Minimum savings with Arnoldi	23	48	119
Minimum savings with Lanczos	23	50	121

Table 1: Comparison of flop counts. For each of our algorithms, the term "savings" in the above table is the ratio of the flop count of the standard balance and truncate model reduction scheme to the flop count of our algorithm. All simulations were performed with MATLAB.

original system, and that of the reduced-order systems, once again illustrating that the reduced-order model obtained from our algorithms are virtually indistinguishable from those obtained by the standard balance-and-truncate method.

Recall that the analysis of the approximation error in Section 2.4 revealed that the quality of our low-rank approximation of the square root of the Gramian depended critically on how small the spectral radius $\rho(A_p) = \rho((pI + A)^{-1}(pI - A))$ is. When the eigenvalues of A are well-damped, as with the test cases presented thus far, the spectral radius of $\rho(A_p)$ can be made significantly less than one with an appropriate choice of p. This is the reason for the remarkably good performance of our approximate balance-and-truncate schemes. For very lightly damped systems, for every choice of p, the value of $\rho(A_p)$ will be very close to one, implying that the quality of approximation with our methods should be poor. To explore this issue further, we considered a second set of test cases, where we randomly generated stable LTI systems with twenty pairs of eigenvalues with a real part of $-.001$, twenty pairs of eigenvalues with a real part of $-.002$, with the remaining eigenvalues having smaller (more negative) real parts. We collected the same statistics as with the earlier test cases; these are shown in Table 2. It is immediately noticeable that as expected, the computational savings due to our algorithms, although quite significant, are not as high as with the reduction of heavily-damped models.

In order to examine the error in approximation, we consider a typical test case of a (full-order) system with 100 states. Our algorithm yielded a reduced-order model with 44 states. An examination of the quality of approximation, shown in Figure 2(a), reveals the remarkable fact that over a large range of frequencies, our approximate balance-and-truncate schemes perform better than the standard balance-and-truncate scheme. A possible explanation for this is that for very lightly-damped systems, the Gramians themselves are ill-conditioned (the Lyapunov operator $\mathcal{L}(\cdot) \triangleq A^T(\cdot) + (\cdot)A$ is close to being singular), and therefore numerical errors lead to the poor performance of the standard balance-and-truncate method. In contrast our algorithms, especially the Arnoldi method, are numerically more stable. Figures 2(b), 2(c) and 2(d) show the magnitude and phase of the system response of the reduced systems. From these plots, it is once again evident that our algorithms perform better than the standard balanced truncation method.



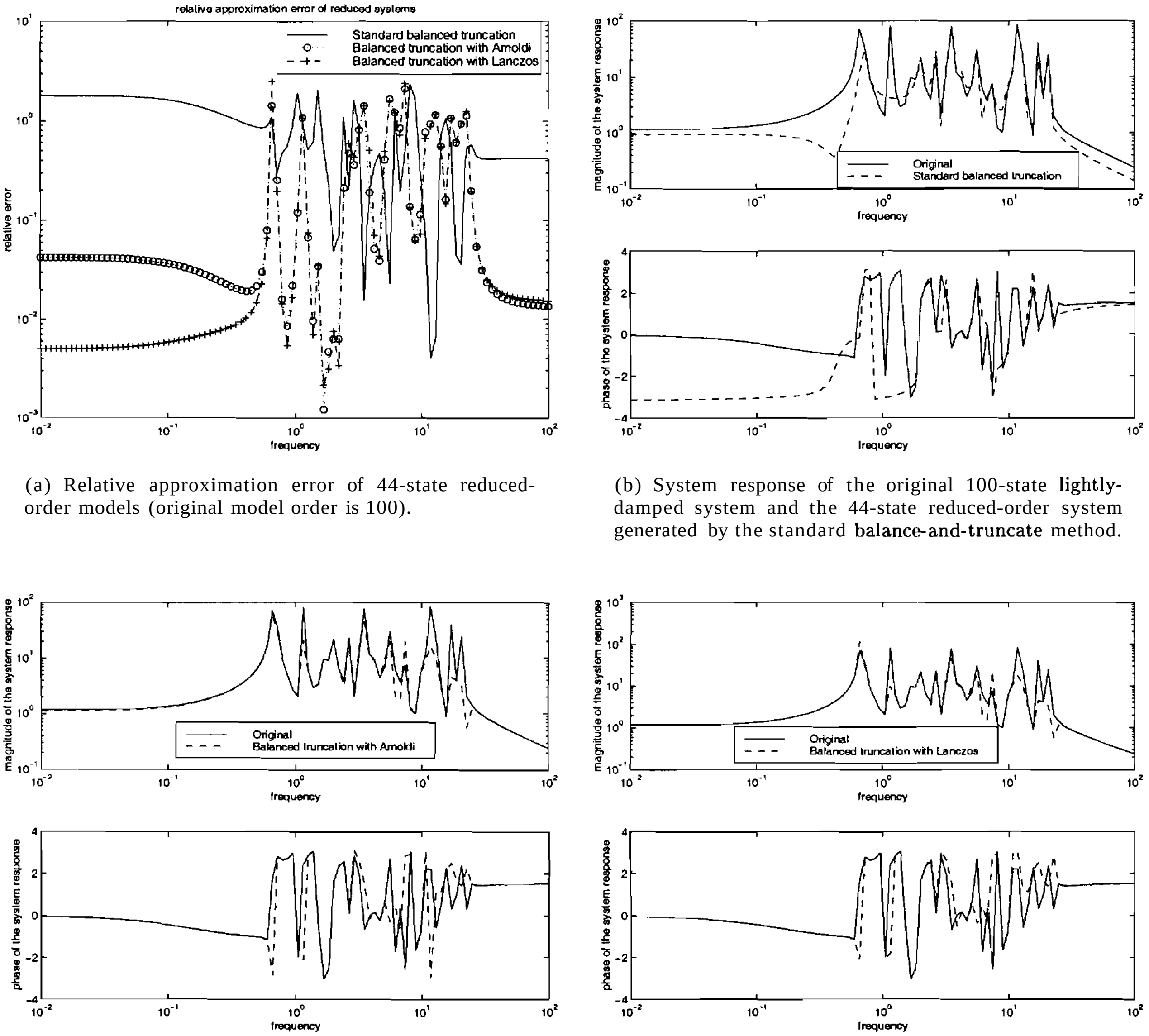
(a) Relative approximation error of 21-state reduced-order models (original model order is 100).

(b) System response of the original 100-state system and the 21-state reduced-order systems generated by different implementations of the balance-and-truncate method.

Figure **1:** A typical test case of a full-order model with **100** states. Our algorithm yielded a reduced-order model with **21** states.

Original model order	100	200	400
Average flop count with the standard balance-truncate method (in millions)	230	1900	16000
Average savings with Arnoldi	22	25	26
Average savings with Lanczos	22	25	25
Maximum savings with Arnoldi	24	28	27
Maximum savings with Lanczos	24	26	25
Minimum savings with Arnoldi	20	23	24
Minimum savings with Lanczos	21	23	25

Table **2:** Comparison of flop counts. For each of our algorithms, the term "savings" in the above table is the ratio of the flop count of the standard balance and truncate model reduction scheme to the flop count of our algorithm. All simulations were performed with MATLAB.



(a) Relative approximation error of 44-state reduced-order models (original model order is 100).

(b) System response of the original 100-state lightly-damped system and the 44-state reduced-order system generated by the standard balance-and-truncate method.

(c) System response of the original 100-state lightly-damped system and the 44-state reduced-order system generated by balance-and-truncate with Arnoldi.

(d) System response of the original 100-state lightly-damped system and the 44-state reduced-order system generated by balance-and-truncate with Lanczos.

Figure 2: A typical test case of a lightly-damped system with 100 states. Our algorithm yielded a reduced-order model with 44 states.

4 Conclusion

We have presented efficient implementations of the balance-and-truncate model reduction technique for large-scale systems, using Krylov subspace methods. The two distinguishing features of our algorithms are: (i) We directly compute state coordinate transformations that approximately balance-and-truncate the state vector. (ii) The coordinate transformations are computed directly from Krylov subspace methods and a small-size SVD, without the need for solving any Lyapunov equations. Numerical simulations show that our approach holds promise in the balance-and-truncate model reduction of large-scale systems.

References

[CC87] M. Celik and A. C. Cangellaris. Simulation of multiconductor transmission lines using Krylov subspace order-reduction techniques. IEEE Trans. Computer-Aided Design, 16(5):485–496, May 1987.

[CN92] E. Chiprout and M. Nakhla. Generalized moment-matching methods for transient analysis of interconnect networks. In Proc. 29th *ACM/IEEE* Design Automation Conference, pages 201–206, Anaheim, CA, June 1992.

[Enn84] D. F. Enns. Model reduction with balanced realizations: An error bound and a frequency weighted generalization. In Proc. IEEE Conf. on Decision and Control, pages 127–132, Las Vegas, NV, December 1984.

[FF95] P. Feldmann and R.. W. Freund. Efficient linear circuit analysis by Pad6 approximation via the Lanczos process. IEEE Trans. Computer-Aided Design, 14:639–649, May 1995.

[Fre98] R.. W. Freund. Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation. Technical Report 98-3-02, Numerical Analysis Manuscript, Bell Laboritories, Murray Hill, New Jersey, February 1998.

[GG97] E. Grimme and K. Gallivan. Approximate solves in Krylov-based modeling methods. In Proc. IEEE Conf. on Decision and Control, pages 3849–3854, December 1997.

[GGD94] K. Gallivan, E. Grimme, and P. Van Dooren. Pad6 approximation of large-scale dynamic systems with Lanczos methods. In Proc. IEEE Conf. on Decision and Control, pages 443–448, December 1994.

[GL89] G. Golub and C. Van Loan. *Matrix* Computations. Johns Hopkins Univ. Press, Baltimore, second edition, 1989.

[GL95] M. Green and D. J. N. Limebeer. Linear Robust Control. Information and System sciences. Prentice Hall, Englewood Cliffs, NJ, 1995.

[Glo84] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their $\mathbf{L}_\infty$-error bounds. Int. J. Control, 39(6):1115–1193, 1984.

[GN99] P. K. Gunupudi and M. S. Nakhla. Model-reduction of nonlinear circuits using Krylov-space techniques. In Proc. 29th *ACM/IEEE* Design Automation Conference, pages 13–16, 1999.

[Hua90] X. Huang. *Padé* Approximation of *Linear(ized)* Circuit Responses. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1990.

[JK94] I. M. Jaimoukha and E. M. Kasenally. Krylov subspace methods for solving large Lyapunov equations. SIAM J. on Numerical Analysis, 31(1):227–251, February 1994.

[JKL92] I. M. Jaimoukha, E. M. Kasenally, and D. J. N. Limebeer. Numerical solution of large scale Lyapunov equations using Krylov subspace methods. In Proc. IEEE Conf. on Decision and Control, pages 1927–1932, December 1992.

[LW91] A. Lu and E. L. Wachspress. Solution of Lyapunov equations by alternating direction implicit iteration. Computers Math. Appl., 21(9):43–58, 1991.

[LW99] J. Li and J. White. Efficient model reduction of interconnnect via approximate system Gramians. In Proc. Int. Conf. on Computer-Aided Design, pages 380–383, 1999.

[LWW99] J. Li, F. Wang, and J. White. An efficient Lyapunov equation-based approach for generating reduced-order models of interconnect. In Proc. Design Automation Conference, pages 1–6, 1999.

[Moo81] B. C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. IEEE Trans. Aut. Control, AC-26(1):17–32, February 1981.

[PR90] L. T. Pillage and R. A. Rohrer. Asymptotic waveform evaluation for timing analysis. IEEE Trans. Computer-Aided Design, 9:352–366, 1990.

[SKW95] L. M. Silveira, M. Kamon, and J. White. Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3D interconnect structures. In Proc. 32nd *ACM/IEEE* Design Automation Conference, pages 376–380, June 1995.