Electronic Transactions on Numerical Analysis.
Volume 7, 1998, pp. 1-17.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

ERROR ESTIMATION OF THE PADÉ APPROXIMATION OF TRANSFER FUNCTIONS VIA THE LANCZOS PROCESS*

ZHAOJUN BAI† AND QIANG YE‡

Abstract. Krylov subspace based moment matching algorithms, such as PVL (Padé approximation Via the Lanczos process), have emerged as popular tools for efficient analyses of the impulse response in a large linear circuit. In this work, a new derivation of the PVL algorithm is presented from the matrix point of view. This approach simplifies the mathematical theory and derivation of the algorithm. Moreover, an explicit formulation of the approximation error of the PVL algorithm is given. With this error expression, one may implement the PVL algorithm that adaptively determines the number of Lanczos steps required to satisfy a prescribed error tolerance. A number of implementation issues of the PVL algorithm and its error estimation are also addressed in this paper. A generalization to a multiple-input-multiple-output circuit system via a block Lanczos process is also given.

Key words. Transfer function, Padé approximation, Lanczos process, linear system.

AMS subject classifications. Primary 41A21, 65F15, 65F10.

1. Introduction. In early 1990s, the AWE (Asymptotic Waveform Evaluation) algorithm [25] based on Padé approximations was the method of choice for efficient analyses of large linear systems. Its success in tackling real-world problems attracted signifigant interest and spawned a substantial amount of related research and applications [6]. However, the AWE algorithm suffers from a number of fundamental numerical limitations for high order approximation. In the recent work of Feldman and Freund [10], a new efficient and numerically stable algorithm, called the Padé approximation via the Lanczos process (PVL), is proposed which produces more accurate results with higher-order approximations compared with AWE and its derivatives. In [12], Gallivan, Grimme and Van Dooren also propose to use the Lanczos process to implicitly construct the Padé approximation in the AWE technique.

The idea of the Padé approximation using the Lanczos process can be traced back to a number of fundamental papers in the numerical analysis community from 1950's to 1970's. In particular, in the 1970's, Gragg was the first to use the Lanczos process to interpret the Padé approximation and related continued fraction algorithms [15]. This work was further developed in [16]. This connection was also discussed in [18]. In [22], a general mathematical framework for the model reduction of a transfer function based on any tridiagonalization process, which includes the Lanczos process, was established.

However, most discussions in the literature are concerned with the order of approximation for the Padé approximation of a transfer function. It is important in engineering applications to have an explicit formulation on the approximation error of the transfer function, [6]. Without such error estimation, it is difficult to determine the order of approximation required (or a stopping criterion in the Lanczos process) to achieve a desired accuracy in practical applications of the moment matching AWE or the PVL techniques.

In [20, 17], the authors derive expressions for the error in terms of residuals of the associated linear systems involved in the transfer function. These general expressions do not particularly exploit the Lanczos method used to derive the PVL approximation. As results, intrinsic properties associated with Padé approximation and the convergence of the PVL method may not be revealed. In this work, we present a new derivation of the PVL algorithm from a ma-

*Received January 27, 1998. Accepted for publication August 24, 1998. Recommended by R. Lehoucq. This work was supported in part by NSF grant ASC-9313958 and by a research grant from NSERC of Canada.
†Department of Mathematics, University of Kentucky, Lexington, KY40506, U.S.A. (bai@ms.uky.edu).
‡Department of Applied Mathematics, University of Manitoba, Winnipeg, Manitoba, R3T 2N2, CANADA (ye@gauss.amath.umanitoba.ca).

trix point of view. With this approach, the mathematical derivation of the Padé approximation used in the AWE and PVL techniques is significantly simplified. Moreover, the approximation error of the Padé approximation of the transfer function is explicitly given. In contrast to the work of [20, 17], our approximation error expression transfers their formal linear system residual based error expressions into a detailed and essentially computable expression which explicitly reveals convergence properties of the PVL method. With this error expression, it is possible to implement the PVL algorithm that adaptively determines the number of iterative steps in the Lanczos process required in order to reduce the error to satisfy a prescribe tolerance value. Furthermore, we have also had a further understanding of why the PVL algorithm will work for a large range of frequency by increasing order of the approximation. A number of implementation issues in the implementation of the PVL algorithm and its error estimation are also addressed in this current work. Another advantage of this matrix approach is its straightforward generalization to a multiple-input-multiple-output circuit system via the block Lanczos process, which will also be given.

In Section 2, we outline the Lanczos process and its governing equations. We will also recall some basic properties associated with a tridiagonal matrix. In Section 3, we present a straightforward derivation of the PVL algorithm and its error estimation for linear circuit systems with single input and single output. Implementation issues of the PVL algorithm and error estimation are discussed in Section 4. Numerical results are presented in section 5 to demonstrate the effectiveness of the error bound. Section 6 is devoted to a generalization of the PVL algorithm and the error bound for multiple input and multiple output linear circuit systems.

2. Lanczos Process. Given an $N \times N$ matrix A and initial vectors $p_1, q_1 \in \mathbf{C}^N$ with $p_1^H q_1 = 1$, the Lanczos process is embodied by the following three-term recurrences

$$b_{j+1} p_{j+1}^H = \hat{p}_{j+1}^H = p_j^H A - a_j p_j^H - c_j p_{j-1}^H, \tag{2.1}$$

$$q_{j+1} c_{j+1} = \hat{q}_{j+1} = A q_j - q_j a_j - q_{j-1} b_j, \tag{2.2}$$

for $j = 1, 2, \ldots, k$ with $p_0 = q_0 = 0$. If we let

$$P_k = [\, p_1, p_2, \ldots, p_k \,], \qquad Q_k = [\, q_1, q_2, \ldots, q_k \,]$$

and

$$T_k = \begin{bmatrix} a_1 & b_2 & & \\ c_2 & a_2 & \ddots & \\ & \ddots & \ddots & b_k \\ & & c_k & a_k \end{bmatrix}, \tag{2.3}$$

then the matrix forms of the three-term vector recurrences (2.1) and (2.2) are

$$A^H P_k = P_k T_k^H + [\, 0 \;\; 0 \;\; \cdots \;\; 0 \;\; \hat{p}_{k+1} \,], \tag{2.4}$$

$$A Q_k = Q_k T_k + [\, 0 \;\; 0 \;\; \cdots \;\; 0 \;\; \hat{q}_{k+1} \,]. \tag{2.5}$$

Furthermore, the computed Lanczos vectors $\{p_j\}$ and $\{q_j\}$ satisfy the *bi-orthonormality* property

$$P_k^H Q_k = I_k \tag{2.6}$$

and $P_k^H \hat{q}_{k+1} = 0$ and $\hat{p}_{k+1}^H Q_k = 0$. Therefore we have

$$P_k^H A Q_k = T_k. \tag{2.7}$$

There is also freedom to normalize either the left or right Lanczos vectors and so we assume $\|p_j\| = 1$.

An alternative scaling scheme can also be used to normalize all Lanczos vectors $\{p_j\}$ and $\{q_j\}$, where $P_k^H Q_k = \Omega_k$, where Ω_k is a $k \times k$ diagonal matrix [11, 22]. It can be shown that these two scaling schemes are similar under diagonal transformation [3].

The Lanczos process can break down prematurely, namely, when a division by zero occurs. There are several methods available to deal with this problem [3, 11, 24, 30]. We point out that simply restarting the algorithm with different initial vectors is not an option in this application because the left and right initial vectors are fixed by the vectors that define the input and output relation.

Assume that the Lanczos process can be carried out to step N (the last step). (If break down occurs and one of the break down recovery schemes must be used, T_N below will not be tridiagonal but similar properties still hold.) Then P_N and Q_N are $N \times N$ square matrices and $P_N^H Q_N = I_N$. Therefore,

$$Q_N^{-1} A Q_N = T_N, \tag{2.8}$$

where T_N is an $N \times N$ tridiagonal matrix. Clearly, T_k is the leading $k \times k$ principal submatrix of T_N for any $k \leq N$.

Before we discuss applications of the Lanczos process to the approximation of transfer functions, we first present the following properties associated with the tridiagonal structure of T_N. The lemma below was derived in [29, Lemma 3.1 and Theorem 3.3] to analyze convergence of the Lanczos algorithm for eigenvalue computation.

LEMMA 2.1. *For any* $0 \leq j \leq 2k-1$,

$$e_1^H T_N^j e_1 = e_1^H T_k^j e_1$$

and for $j = 2k$,

$$e_1^H T_N^{2k} e_1 = e_1^H T_k^{2k} e_1 + b_2 b_3 \cdots b_k b_{k+1} \cdot c_{k+1} c_k \cdots c_3 c_2.$$

Furthermore,

$$e_1^H T_k^j e_k = \begin{cases} 0, & 0 \leq j < k-1 \\ b_2 b_3 \cdots b_k, & j = k-1 \end{cases}$$

and

$$e_k^H T_k^j e_1 = \begin{cases} 0, & 0 \leq j < k-1 \\ c_k \cdots c_3 c_2, & j = k-1; \end{cases}$$

here and hereafter, e_j *denotes the standard* j*th coordinate vector, i.e., a vector with all zeros except for the* j*th component which is* 1. *The dimension of* e_j *will conform with other vectors and matrices involved.*

3. Approximation of Transfer Functions and Error Estimation. In this section, we present a derivation of using the Lanczos process for the approximation of the transfer function of a linear system with single input and single output. The resulting method is essentially the same as the PVL algorithm of Feldman and Freund [10], which is also theoretically equivalent to the AWE (moment matching) method. The present derivation is straightforward and, more importantly, it provides an error estimation of the PVL approximation.

ETNA
Kent State University
etna@mcs.kent.edu

Consider the following state space formulation of a single input and single output linear system

$$C\dot{x}(t) = -Gx(t) + b\,u(t), \tag{3.1}$$

$$y(t) = l^H x(t) + d\,u(t), \tag{3.2}$$

where $x(t) \in \mathbf{R}^N$ is the state vector, $u(t) \in \mathbf{R}$ is the input and $y \in \mathbf{R}$ is the output of interest. In the electronic circuits applications, $x(t)$ represents the circuit variables at time t, and the terms $b\,u(t)$ and $d\,u(t)$ represent excitation from independent sources. $N \times N$ matrices G and C represent the contribution of memoryless elements (such as resistors) and memory elements (such as capacitors and inductors), respectively. $l \in \mathbf{R}^N$. For the sake of simplicity, we can assume the initial condition $x(0) = 0$.

The standard way of relating the input and output signals is to use a transfer function (or impulse response in the time domain) of the linear circuit. Applying the Laplace transform to equations (3.1) and (3.2), we obtain

$$sCX(s) = -GX(s) + b\,U(s), \tag{3.3}$$

$$Y(s) = l^H X(s) + d\,U(s), \tag{3.4}$$

where $X(s), Y(s)$ and $U(s)$ are the Laplace transforms of $x(t), y(t)$ and $u(t)$, respectively. Then the transfer function of the linear system is the following rational function

$$H(s) = \frac{Y(s)}{U(s)} = l^H(sC + G)^{-1}b + d. \tag{3.5}$$

To determine the impulse-response in the circuit analysis and other applications, it is important to compute the transfer function over a wide range of the frequency parameter s. Direct computations of $H(s)$ becomes inefficient or even prohibitive for a large scale system, namely, when the order N of the matrices C and G is large.

To compute $H(s)$, we follow [10] and expand $H(s)$ about some point s_0, i.e., write $s = s_0 + \sigma$. Then

$$H(s_0 + \sigma) = l^H(I - \sigma A)^{-1}r + d,$$

where

$$A = -(s_0C + G)^{-1}C \qquad \text{and} \qquad r = (s_0C + G)^{-1}b.$$

Since the crucial computation of $H(s_0 + \sigma)$ lies in the first part, we can assume without loss of generality that $d = 0$.

Let

$$l = p_1 b_1 \qquad \text{and} \qquad r = q_1 c_1$$

with $p_1^H q_1 = 1$. Then $l^H r = b_1 c_1$ and

$$H(s_0 + \sigma) = (l^H r) \cdot p_1^H(I - \sigma A)^{-1}q_1.$$

Applying k steps of the Lanczos process with the triplet $\{A, p_1, q_1\}$, we obtain the equations (2.4) and (2.5). Then a reduced-order transfer function

$$H_k(s_0 + \sigma) = (l^H r) \cdot e_1^H(I - \sigma T_k)^{-1}e_1 \tag{3.6}$$

can be defined as an approximation of $H(s_0+\sigma)$. Indeed using the Neumann series expansion, we have

$$H_k(s_0+\sigma) = (l^H r)\cdot \sum_{j=0}^{\infty} \sigma^j e_1^H T_k^j e_1$$

provided that $|\sigma| < 1/\rho(T_k)$, where $\rho(M)$ denotes the spectral radius of a matrix M, and by (2.8)

$$\begin{aligned} H(s_0+\sigma) &= (l^H r)\cdot p_1^H (I-\sigma Q_N T_N Q_N^{-1})^{-1} q_1 \\ &= (l^H r)\cdot p_1^H Q_N (I-\sigma T_N)^{-1} Q_N^{-1} q_1 \\ &= (l^H r)\cdot e_1^H (I-\sigma T_N)^{-1} e_1 \\ &= (l^H r)\cdot \sum_{j=0}^{\infty} \sigma^j e_1^H T_N^j e_1, \end{aligned}$$

provided that $|\sigma| < 1/\rho(A)$, where we have also used the facts $p_1^H Q_N = e_1^H$ and $P_N^H q_1 = e_1$. Thus by Lemma 1 in Section 2,

$$\begin{aligned} H(s_0+\sigma) - H_k(s_0+\sigma) &= (l^H r)\cdot \sum_{j=2k}^{\infty} \sigma^j (e_1^H T_N^j e_1 - e_1^H T_k^j e_1) \\ &= (l^H r)\cdot \sigma^{2k}\cdot b_2 b_3\cdots b_k b_{k+1}\cdot c_2 c_3\cdots c_k c_{k+1} + O(\sigma^{2k+1}). \end{aligned}$$

Therefore, the approximation $H_k(s_0+\sigma)$ has the order $2k$ of σ that defines the Padé approximation of the transfer function (or AWE reduction method of moments matching up to the order of $2k-1$ as it would be called in the interconnect analysis community). The coefficient of the leading σ^{2k} term of the approximation is given explicitly. But note that since the frequency parameter σ of interest may be large, the leading term generally does not indicate the correct order of approximation in that case. Fortunately, the following theorem gives an explicit expression of the approximation error.

THEOREM 3.1. *If $I-\sigma A$ and $I-\sigma T_k$ are invertible, then*

$$\begin{aligned} &|H(s_0+\sigma) - H_k(s_0+\sigma)| \\ &= |l^H r|\,|\sigma^2 \tau_{1k}(\sigma)\tau_{k1}(\sigma)|\,|\hat{p}_{k+1}^H (I-\sigma A)^{-1}\hat{q}_{k+1}| \qquad (3.7) \\ &= |\sigma|^{2k} |l^H r| \frac{|b_2 b_3\cdots b_k c_2 c_3\cdots c_k|}{|\det(I-\sigma T_k)^2|} |\hat{p}_{k+1}^H (I-\sigma A)^{-1}\hat{q}_{k+1}| \qquad (3.8) \end{aligned}$$

where

$$\tau_{1k}(\sigma) = e_1^H (I-\sigma T_k)^{-1} e_k \qquad \textit{and} \qquad \tau_{k1}(\sigma) = e_k^H (I-\sigma T_k)^{-1} e_1,$$

i.e., they are the $(1,k)$ and $(k,1)$ entries of the inverse of the tridiagonal matrix $I-\sigma T_k$. Furthermore, for $|\sigma| < 1/\|A\|$, where $\|\cdot\|$ can be any induced matrix norm, we have

$$|H(s_0+\sigma) - H_k(s_0+\sigma)| \le |l^H r|\,|\sigma^2 \tau_{1k}(\sigma)\tau_{k1}(\sigma)|\,\frac{\|\hat{p}_{k+1}\|\,\|\hat{q}_{k+1}\|}{1-|\sigma|\,\|A\|}. \qquad (3.9)$$

Proof. From (2.4) and (2.5), it is easy to see that

$$(I-\sigma T_k)^{-1} P_k^H = P_k^H (I-\sigma A)^{-1} - \sigma (I-\sigma T_k)^{-1} e_k \hat{p}_{k+1}^H (I-\sigma A)^{-1} \qquad (3.10)$$

and

$$Q_k(I-\sigma T_k)^{-1} = (I-\sigma A)^{-1}Q_k - \sigma(I-\sigma A)^{-1}\hat{q}_{k+1}e_k^H(I-\sigma T_k)^{-1}. \tag{3.11}$$

Multiplying the equation (3.11) on the left by p_1^H and on the right by e_1, we have

$$e_1^H(I-\sigma T_k)^{-1}e_1 = p_1^H(I-\sigma A)^{-1}q_1 - \sigma p_1^H(I-\sigma A)^{-1}\hat{q}_{k+1}e_k^H(I-\sigma T_k)^{-1}e_1,$$

where we have used the fact that $p_1^H Q_k = e_1^H$. Multiplying the equation (3.10) on the left by e_1^H and on the right by $\hat{q}_{k+1}$, we have

$$0 = p_1^H(I-\sigma A)^{-1}\hat{q}_{k+1} - \sigma e_1^H(I-\sigma T_k)^{-1}e_k\hat{p}_{k+1}^H(I-\sigma A)^{-1}\hat{q}_{k+1},$$

where we used the fact $P_k^H\hat{q}_{k+1} = 0$. Combining the above two equations, we obtain

$$H(s_0+\sigma)-H_k(s_0+\sigma) = (l^Hr)\sigma^2 e_1^H(I-\sigma T_k)^{-1}e_k\hat{p}_{k+1}^H(I-\sigma A)^{-1}\hat{q}_{k+1}e_k^H(I-\sigma T_k)^{-1}e_1.$$

Note that the inverse matrix of $I-\sigma T_k$ can be expressed as

$$(I-\sigma T_k)^{-1} = \frac{\operatorname{adj}(I-\sigma T_k)}{\det(I-\sigma T_k)},$$

where $\operatorname{adj}(I-\sigma T_k)$ stands for the classical adjoint matrix made up of $(k-1)\times(k-1)$ cofactors of $I-\sigma T_k$. Thus, we have

$$e_k^H(I-\sigma T_k)^{-1}e_1 = \frac{\sigma^{k-1}(c_2c_3\cdots c_k)}{\det(I-\sigma T_k)},$$

and

$$e_1^H(I-\sigma T_k)^{-1}e_k = \frac{\sigma^{k-1}(b_2b_3\cdots b_k)}{\det(I-\sigma T_k)}.$$

Therefore

$$H(s_0+\sigma)-H_k(s_0+\sigma) = \sigma^{2k}(l^Hr)\,\frac{b_2b_3\cdots b_kc_2c_3\cdots c_k}{\det(I-\sigma T_k)^2}\,\hat{p}_{k+1}^H(I-\sigma A)^{-1}\hat{q}_{k+1}.$$

This completes the proof of (3.7) and (3.8). Finally, (3.9) is proved by applying the Cauchy-Schwarz inequality

$$|\hat{p}_{k+1}^H(I-\sigma A)^{-1}\hat{q}_{k+1}| \le \|\hat{p}_{k+1}^H\|\,\|(I-\sigma A)^{-1}\|\,\|\hat{q}_{k+1}\|$$

and the simple inequality

$$\|(I-\sigma A)^{-1}\| \le \frac{1}{1-|\sigma|\,\|A\|} \tag{3.12}$$

provided that $|\sigma| < 1/\|A\|$. $\square$

Note that in Theorem 5.1 of Grimme [17], an error expression is given in terms of two residual vectors of a general Rational Krylov-subspace-based approximation of transfer function. Since these residuals can be formulated in terms of the Lanczos vectors for the PVL method, a similar error expression of (3.7) can also be derived based on Theorem 5.1 of [17].

As pointed out in [17], it is important to monitor the trends in the residual behavior as s and k vary. Direct attempting to gauge these trends is very expensive in practice. Fortunately, for the general Rational Krylov algorithm, the residual expression can be simplified [17, Lemma 5.1]. For the PVL algorithm we are concerned with, our bound (3.8) shows that there are two factors in the modeling error: $|\sigma^2 \tau_{1k}(\sigma)\tau_{k1}(\sigma)|$ and $|\hat{p}_{k+1}^H (I-\sigma A)^{-1}\hat{q}_{k+1}|$. As shown in (3.8), the quantity $|\sigma^2 \tau_{1k}(\sigma)\tau_{k1}(\sigma)|$ is of order σ^{2k}. Numerical experiments indicate that the terms $|\tau_{1k}(\sigma)|$ and $|\tau_{k1}(\sigma)|$ decrease steadyly as the number of Lanczos iterations increases while the values of the term $|\hat{p}_{k+1}^H (I-\sigma A)^{-1}\hat{q}_{k+1}|$ essentially remain on the same order of magnitude near convergence. $|\tau_{1k}(\sigma)|$ and $|\tau_{k1}(\sigma)|$ are the primary contributors to the convergence of the PVL approximation. This phenomenon can be explained by the following theorem. Note that it is also related to the discussion on decay rate of the elements of the inverse of a tridiagonal matrix, see [8, 21].

THEOREM 3.2. *If $I-\sigma T_k$ is invertible, then*

$$\frac{|\tau_{k1}(\sigma)|}{\|e_k^H (I-\sigma T_k)^{-1}\|_2} \le \sqrt{N}\,\epsilon_k$$

where

$$\epsilon_k = \min_{h\in\Pi_{k-1},h(0)=1} \|h(I-\sigma A)q_1\|_2$$

and Π_k denotes the set of all polynomials of degree k. A similar inequality holds for $|\tau_{1k}(\sigma)|$.

Proof. For any polynomial $h\in\Pi_{k-1}$ with $h(0)=1$, we write $h(x)=1+xg(x)$ with $g\in\Pi_{k-2}$. Thus using Lemma 1 again, we have

$$\begin{aligned}
|\tau_{k1}(\sigma)| &= |e_k^H (I-\sigma T_k)^{-1} e_1| \\
&= |e_k^H \left[(I-\sigma T_k)^{-1} + g(I-\sigma T_k)\right] e_1| \\
&= |e_k^H (I-\sigma T_k)^{-1} h(I-\sigma T_k) e_1| \\
&\le \|e_k^H (I-\sigma T_k)^{-1}\|_2 \|h(I-\sigma T_k)e_1\|_2 \\
&= \|e_k^H (I-\sigma T_k)^{-1}\|_2 \|h(I-\sigma T_N)e_1\|_2 \\
&= \|e_k^H (I-\sigma T_k)^{-1}\|_2 \|P_N^H h(I-\sigma A) Q_N e_1\|_2 \\
&\le \|e_k^H (I-\sigma T_k)^{-1}\|_2 \|P_N\|_2 \|h(I-\sigma A)q_1\|_2.
\end{aligned}$$

With the normalization $\|p_j\|_2=1$, $\|P_N\|_2\le\sqrt{N}$. This proves the theorem. ☐

The theorem shows that $|\tau_{k1}(\sigma)|$, relative to the absolute values of other entries in the kth row of $(I-\sigma T_k)^{-1}$, is bounded by ϵ_k, which decreases as the order of approximation k increases. However, the decreasing rate of ϵ_k depends on the spectral distribution of A. This has been widely studied in the convergence theory of Krylov subspace based algorithms (see [27], for example).

Finally, we show that under certain assumption, the term $|\sigma^2\tau_{1k}(\sigma)\tau_{k1}(\sigma)|$ is a monotonic function of $|\sigma|$. Therefore, for a certain range of σ of interest, we only need to compute the bound for the value of σ at the extremal point of the interested range. Specially, if we are interested in σ with $\sigma=|\sigma|e^{\mathrm{i}\theta}$ for a fixed θ (e.g. the upper pure imaginary axis if $\theta=\pi/2$), then from Theorem 1,

$$\begin{aligned}
|\sigma^2\tau_{1k}(\sigma)\tau_{k1}(\sigma)| &= \frac{|\sigma|^{2k}|b_2b_3\cdots b_kc_2c_3\cdots c_k|}{|\det(I-\sigma T_k)^2|} \\
&= \frac{|b_2b_3\cdots b_kc_2c_3\cdots c_k|}{|(|\sigma|^{-1}-e^{\mathrm{i}\theta}\lambda_1)|^2\cdots|(|\sigma|^{-1}-e^{\mathrm{i}\theta}\lambda_k)|^2} \\
&= \frac{|b_2b_3\cdots b_kc_2c_3\cdots c_k|}{[(|\sigma|^{-1}-\alpha_1)^2+\beta_1^2]\cdots[(|\sigma|^{-1}-\alpha_k)^2+\beta_k^2]},
\end{aligned}$$

where λ_j (for $1 \leq j \leq k$) are the eigenvalues of T_k and we have denoted $e^{\mathbf{i}\theta}\lambda_j = \alpha_j + \mathbf{i}\beta_j$. It is easy to see from the last expression that $|\sigma^2\tau_{1k}(\sigma)\tau_{k1}(\sigma)|$ is monotonic increasing in $|\sigma|$ in the range $0 \leq |\sigma| \leq |\sigma_0| < 1/\|T_k\|$ for a fixed $\sigma_0 = |\sigma_0|e^{\mathbf{i}\theta}$. Thus for all σ in that range, the bound need only be computed at one point σ_0.

4. Comments on Implementation. In this section, we discuss some implementation details for the PVL algorithm with error estimation.

A. Estimation of $\|A\|$: The norm used in Theorem 1 can be any of the operator induced norms, e.g. the 1-norm or the ∞-norm. The 1-norm or ∞-norm of the matrix

$$A = -(s_0C + G)^{-1}C$$

can be easily estimated using just the operations of matrix-vector products Az and A^Hz, which are the same operations used during the implementation of Lanczos process (see Section 2). A detailed description of the algorithm can be found in [19]. An implementation is provided in LAPACK [2].

B. Computing $e_1^T(I - \sigma T_k)^{-1}e_1$, $e_1^T(I - \sigma T_k)^{-1}e_k$ and $e_k^T(I - \sigma T_k)^{-1}e_1$: These values are needed in computing the values of the reduced-order transfer function $H_k(s_0 + \sigma)$ and the error bound (3.9). A common practice is to first compute the eigendecomposition of T_k:

$$T_k = S_k\Lambda_kS_k^{-1} = S_k\mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_k)S_k^{-1}$$

and set

$$f = S_k^He_1, \qquad g = S_k^{-1}e_1.$$

Then we obtain the so-called poles and residues representation of the reduced-order transfer function

$$\begin{aligned} H_k(s_0 + \sigma) &= (l^Hr) \cdot e_1^H(I - \sigma T_k)^{-1}e_1 \\ &= \sum_{j=1}^{k} \frac{l^Hr \cdot f_jg_j}{1 - \sigma\lambda_j} \\ &= \rho_\infty + \sum_{j=1, \lambda_j \neq 0}^{k} \frac{-l^Hr \cdot f_jg_j/\lambda_j}{\sigma - 1/\lambda_j}, \end{aligned}$$

where f_j and g_j are the components of the vectors f and g and where the term ρ_∞ may result if one of the eigenvalues of T_k is zero. This form is also suitable for computing the inverse Laplace transform.

Similarly, the quantities $\tau_{1k}(\sigma) = e_1^T(I - \sigma T_k)^{-1}e_k$ and $\tau_{k1}(\sigma) = e_k^T(I - \sigma T_k)^{-1}e_k$ used in the error estimation can also be easily computed using the eigen-decomposition of T_k. However, we point out that all of these quantities are just the $(1,1)$, $(1,k)$ $(k,1)$ elements of the inverse of the tridiagonal matrix $I - \sigma T_k$. They can be computed incrementally in the Lanczos process with very few additional operations. References [21, 13, 4] give more details (see also Theorem 3 in section 6.2).

It should also be pointed out that if some eigenvalues or eigenvectors of T_k are ill-conditioned, the computation via the eigendecomposition may be numerically unstable. In such a case, a direct computation (solving a linear system of T_k for each frequency value) might be preferred.

C. Balancing in computing the transfer function: The entries of the matrix

$$A = -(s_0 C + G)^{-1} C$$

(and C and G) could be poorly scaled because of physical units involved. A poorly scaled matrix could cause numerical difficulty in the implementation of Lanczos process. In addition, a large norm of the matrix A could limit the applicability of the error bound in the desired range of frequency. Therefore, a procedure for *balancing* a matrix and reducing its norm should be considered. For any nonsingular matrix D,

$$\begin{aligned} H(s_0+\sigma) &= l^H (I-\sigma A)^{-1} r \\ &= l^H D D^{-1} (I-\sigma A)^{-1} D D^{-1} r \\ &= l^H D \left(I - \sigma (D^{-1} A D)\right)^{-1} D^{-1} r. \end{aligned}$$

For example, one may choose a proper nonsingular matrix D such that

- the columns and rows of $D^{-1}AD$ are properly balanced and
- $\|D^{-1}AD\|$ is as small as possible.

An algorithm to do this has been widely used in computing the eigendecomposition of nonsymmetric matrices (for example, it is used in EISPACK, MATLAB and LAPACK). It was originally proposed by Parlett and Reinsch [23]. Although the scheme has been mainly used only in the dense matrix eigensolvers, recently Chen and Demmel [5] have implemented this scheme for sparse matrices, provided that the nonzero entries of a sparse matrix can be accessed directly. Unfortunately, since the entries of the matrix A are not available explicitly (i.e., the matrix $A = -(s_0 C + G)^{-1} C$ is never formed explicitly), a balancing scheme which uses the matrix-vector products is desirable here. In [5], a sparse matrix balancing algorithm uses only matrix-vector multiplication is also presented.

Note that we may approach our particular problem by balancing C and G, respectively. The entries of these matrices are immediately available. Specifically, for any nonsingular matrices D_1 and D_2, let

$$\widetilde{C} = D_1^{-1} C D_2, \qquad \widetilde{G} = D_1^{-1} G D_2.$$

Then

$$D_2^{-1} A D_2 = D_2^{-1} (s_0 C + G)^{-1} D_1^{-1} D_1 C D_2 = (s_0 \widetilde{C} + \widetilde{G})^{-1} \widetilde{C}.$$

With this balancing idea, the question becomes how to choose the nonsingular matrices D_1 and D_2 so that the rows and columns of $D_2^{-1} A D_2$ are properly balanced and the norm $\|D_2^{-1} A D_2\|$ is as small as possible. There is a scheme for balancing a pair of matrices used in the generalized eigenvalue problem [28], which is implemented in LAPACK. The applicability of the scheme to our current problem is the subject of further study.

D. Stopping Criterion and an Adaptive Implementation: With the availability of an error expression on the reduced-order transfer function $H_k(s_0+\sigma)$, one may develop an adaptive scheme to increase the number of matching moments (i.e., the number of Lanczos iterations) until a prescribed tolerance value of error is satisfied. Since the number k of Lanczos iterations is small in practice, $k \ll N$, T_k is a $k \times k$ tridiagonal matrix, the cost of computing the term $|\sigma^2 \tau_{1k}(\sigma)\tau_{k1}(\sigma)|$, which contributes to the convergence of the PVL approximation, is only a small fraction of the cost of Lanczos iterations. The major concern is on the estimation of the term $|\hat{p}_{k+1}^H (I-\sigma A)^{-1} \hat{q}_{k+1}|$. For small σ, namely, $|\sigma| < 1/\|A\|$, the upper bound in (3.9) can be used for stopping criterion. For $|\sigma| \geq 1/\|A\|$, however, it is no longer valid because of the inequality (3.12). Heuristically, the term $|\hat{p}_{k+1}^H(I-\sigma A)^{-1}\hat{q}_{k+1}|$ remains on the same order of magnitude after first few iteration. Since only an estimate of the magnitude of this quantity is required, we may simply use the quantity $|\hat{p}_{k+1}^H\hat{q}_{k+1}|$ to replace the term $|\hat{p}_{k+1}^H(I-\sigma A)^{-1}\hat{q}_{k+1}|$. Numerical experiments, some to be presented in next section, show that it matches the actually error very well. Unfortunately, there is no theoretical justification of this strategy at this time and whether it works for more general problems is still an open question.

5. Numerical Examples. In this section, we present three numerical examples to demonstrate the effectiveness of the error estimation of PVL algorithm discussed above.

Example 1. The first example is from a lumped-pi model of a three-conductor transmission line arranged to study near-end crosstalk [7]. The order of matrices C and G is 11. The expansion point $s_0 = 0$ and $\|A\| = \mathcal{O}(10^{-8})$. We examine the convergence of $H_k(s_0+\sigma)$ for $\sigma = s - s_0 = 2\pi jf$ for frequency f up to 1GHz. Numerical results of the approximation and error estimation for 4 and 5 PVL iterations are plotted in Figures 5.1 and 5.2, respectively.

In the top plots of both figures, the solid lines are the absolute values of original transfer function $H(s_0+\sigma)$. The dash lines are the reduced-order transfer function $H_k(s_0+\sigma)$. The bottom plots are the exact error $|H(s_0+\sigma)-H_k(s_0+\sigma)|$ (solid line), and its estimates (dash-dot and dash lines). Specifically, the dash-dot lines are plotted according to the estimator

$$|l^H r|\,|\sigma^2\tau_{1k}(\sigma)\tau_{k1}(\sigma)|\,\frac{\|\hat{p}_{k+1}\|\,\|\hat{q}_{k+1}\|}{|1-|\sigma|\,\|A\||} \tag{5.1}$$

for all $|\sigma|$, including when $|\sigma| \geq 1/\|A\|$. The dash lines are based on the following estimator

$$|l^H r|\,|\sigma^2\tau_{1k}(\sigma)\tau_{k1}(\sigma)|\,|\hat{p}_{k+1}^H\hat{q}_{k+1}|. \tag{5.2}$$

From the figures, we see that overall, the error estimation by formula (5.1) gives an upper bound of the actual error, but the estimates by (5.2) are much closer to the actual error. In the figures, there is a frequency interval where the actual errors are larger than either of estimators (5.1) and (5.2). This is due to the moderate ill-conditioning of $I-\sigma A$ and the roundoff errors in the presence of finite precision arithmetic when computing the original transfer function $H(s_0+\sigma)$. See the error analysis of linear system solvers based on the Gaussian elimination in [14].

Example 2. The second example is from an extracted RLC circuit. The matrix pair (C, G) is of order 52. The numerical results of 25 PVL iterations are plotted in Figure 5.3. All plots use the same legend as described in Example 1. The formula (5.2) is a very good error estimator. However, the estimator (5.2) severely under-estimates the error at high order range of frequencies. Again, the discrepancies between the computed exact errors and the error estimators given by (5.1) and (5.2) at low frequency are due to the moderate ill-conditioning of matrix $I-\sigma A$ and the roundoff errors in the presence of finite precision arithmetic.

Example 3. This is the same example used in [10] from a three-dimensional electromagnetic problem model via PEEC (partial element equivalent circuit) [26]. It is regarded as a benchmark and difficult test problem. The order of the matrix pair (C, G) is 306. The expansion point $s_0 = 2\pi\sqrt{-1}\times 10^9$ and $\|A\| = \mathcal{O}(10^{-8})$. The numerical results of 80 PVL iterations are plotted in Figure 5.4. All plots use the same legend as described in Example 1. Again, the formula (5.2) is a very good error estimator. The error estimation by (5.1) is over-estimated for high frequencies.

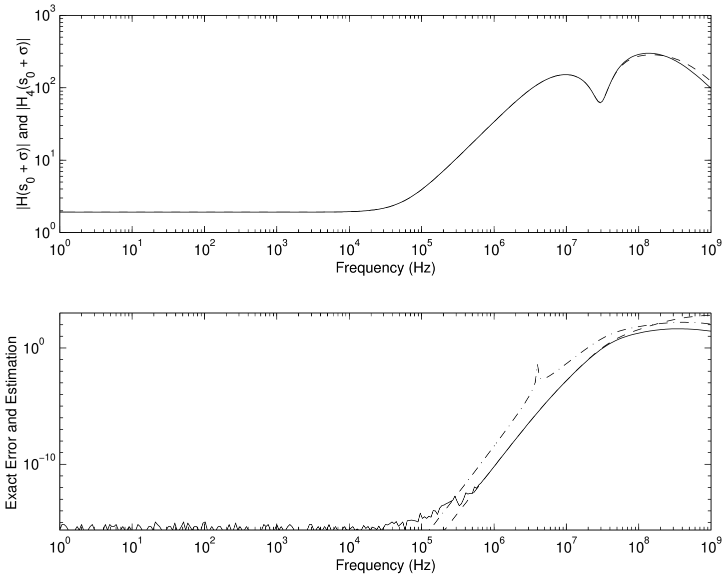

FIG. 5.1. *Numerical results of 4 PVL iterations of Example 1. Top: $|H(s)|$ (solid line) and its approximation $|H_4(s)|$ (dash line). Bottom: Exact error $|H(s) - H_4(s)|$ (solid line) and its estimation by (5.1) (dash-dot line) and by (5.2) (dash line)*

6. Block Lanczos Process and MIMO Transfer Function Approximation. In this section, we present a generalizations of error estimation to a matrix-valued transfer function of a linear multiple input multiple output system. The matrix approach makes this generalization straightforward and we therefore omit most of the details in our presentation.

A matrix Padé approximation of a matrix-valued transfer function has been presented in [9] using a general Lanczos-type process with multiple starting vectors [1]. It is called MPVL algorithm. We will only use a simple version of the block Lanczos process for illustrating our approach. The complete discussion of error estimation of the MPVL algorithm is beyond the scope of the present paper.

6.1. Block Lanczos Process. We first present a simple block Lanczos process for non-Hermitian matrices and discuss its basic properties. An adaptive blocksize implementation for the robustness and efficiency can be found in [3].

The block Lanczos process starts with two initial blocks of vectors P_1 and $Q_1 \in \mathbb{R}^{N\times p}$ and implements the three-term recurrences

$$B_{j+1}P_{j+1}^H = \widehat{P}_{j+1}^H = P_j^H A - A_j P_j^H - C_j P_{j-1}^H \tag{6.1}$$

$$Q_{j+1}C_{j+1} = \widehat{Q}_{j+1} = AQ_j - Q_j A_j - Q_{j-1}B_j. \tag{6.2}$$

As in the unblock case, if we let

$$\mathcal{P}_k = [\,P_1, P_2, \ldots, P_k\,], \qquad \mathcal{Q}_k = [\,Q_1, Q_2, \ldots, Q_k\,]$$

and

$$T_k = \begin{bmatrix} A_1 & B_2 & & \\ C_2 & A_2 & \ddots & \\ & \ddots & \ddots & B_k \\ & & C_k & A_k \end{bmatrix}, \tag{6.3}$$

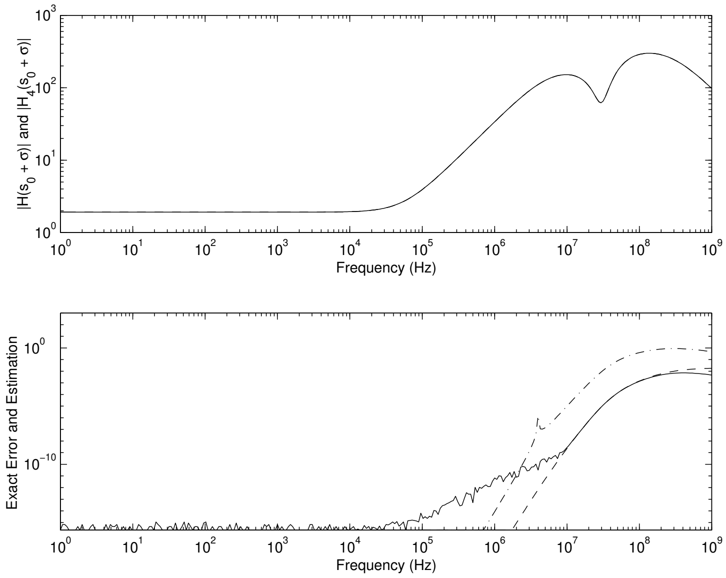

FIG. 5.2. *Numerical results of 5 PVL iterations of Example 1. Top:* $|H(s)|$ *(solid line) and its approximation* $|H_5(s)|$ *(dash line). Bottom: Exact error* $|H(s) - H_5(s)|$ *(solid line) and its estimation by (5.1) (dash-dot line) and by (5.2) (dash line)*

then the three-term recurrences (6.1) and (6.2) have the matrix forms

$$\mathcal{P}_k^H A = T_k \mathcal{P}_k^H + E_k \widehat{P}_{j+1}^H, \tag{6.4}$$

$$A\mathcal{Q}_k = \mathcal{Q}_k T_k + \widehat{Q}_{j+1} E_k^H, \tag{6.5}$$

where E_k is a $kp \times p$ tall thin block matrix whose entries are zero except the bottom square which is the $p \times p$ identity matrix. Furthermore, the computed Lanczos vectors P_k and Q_k satisfy the *bi-orthonormality* property

$$\mathcal{P}_k^H \mathcal{Q}_k = I. \tag{6.6}$$

From the above equations, we have

$$\mathcal{P}_k^H A \mathcal{Q}_k = T_k. \tag{6.7}$$

When the blocksize is $p = 1$, this is just the unblocked case that we discussed in Section 2.

If the above algorithm is carried to the end with n being the last step, then $\mathcal{P}_n$ and $\mathcal{Q}_n$ are $N \times N$ square matrices and $\mathcal{P}_n^H \mathcal{Q}_n = I$. Thus

$$\mathcal{Q}_n^{-1} A \mathcal{Q}_n = T_n, \tag{6.8}$$

where T_n is a block tridiagonal matrix. T_k is a leading principal submatrix of T_n for any $k \leq n$.

The following lemma is a generalization of Lemma 1 to the block case. The proofs are similar to those in [29, Lemma 3.1, Theorem 3.3] and are therefore omitted here.

LEMMA 6.1. *For any* $0 \leq j \leq 2k-1$,

$$E_1^H T_n^j E_1 = E_1^H T_k^j E_1$$

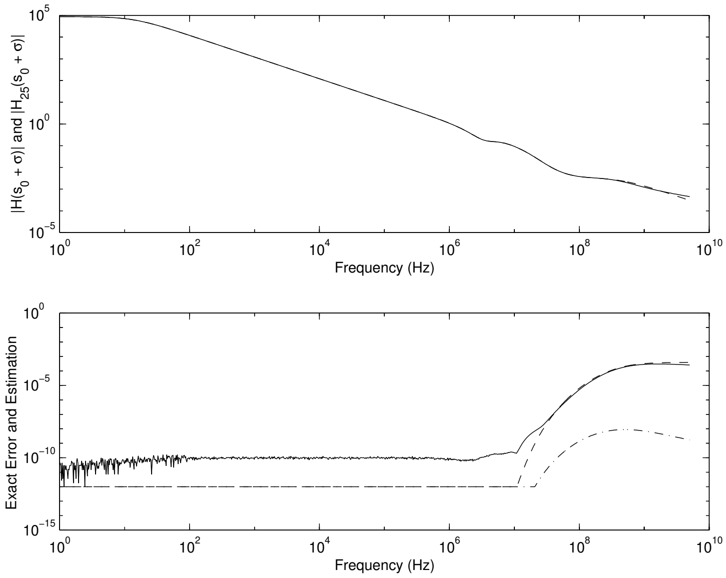

FIG. 5.3. *Numerical results of 25 PVL iterations of Example 2. Top: $|H(s)|$ (solid line) and its approximation $|H_{25}(s)|$ (dash line). Bottom: Exact error $|H(s) - H_{25}(s)|$ (solid line) and its estimation by (5.1) (dash-dot line) and by (5.2) (dash line).*

and for $j = 2k$,

$$E_1^H T_n^{2k} E_1 = E_1^H T_k^{2k} E_1 + B_2 \cdots B_k B_{k+1} C_{k+1} C_k \cdots C_2.$$

Furthermore,

$$E_1^H T_k^j E_k = \begin{cases} 0, & 0 \le j \le k-1 \\ B_2 B_3 \cdots B_k, & j = k-1. \end{cases}$$

and

$$E_k^H T_k^j E_1 = \begin{cases} 0, & 0 \le j \le k-1 \\ C_k \cdots C_3 C_2, & j = k-1. \end{cases}$$

Hereafter, E_j denotes a tall thin matrix whose bottom square is an identity matrix and is zero elsewhere. The dimension of E_j will conform with other vectors and matrices involved.

6.2. MIMO systems. We consider the same type of linear systems as in Section 3 but with a multiple input $u(t)$ and a multiple output $y(t)$. The state space formulation of a multiple input multiple output linear system is given as follows

$$C\dot{x}(t) = -Gx(t) + Bu(t) \tag{6.9}$$
$$y(t) = L^H x(t) + Du(t) \tag{6.10}$$

where $x(t) \in \mathbf{R}^N$ is the state vector, $u(t) \in \mathbf{R}^\ell$ is the input vector, $y(t) \in \mathbf{R}^m$ is the output vector and $C, G \in \mathbf{R}^{N\times N}$, $B \in \mathbf{R}^{N\times\ell}$, $L \in \mathbf{R}^{N\times m}$ and $D \in \mathbf{R}^{m\times\ell}$. Note that the dimensions m and ℓ for the input and output need not be the same. For the sake of simplicity, we assume again the initial condition $x(0) = 0$.

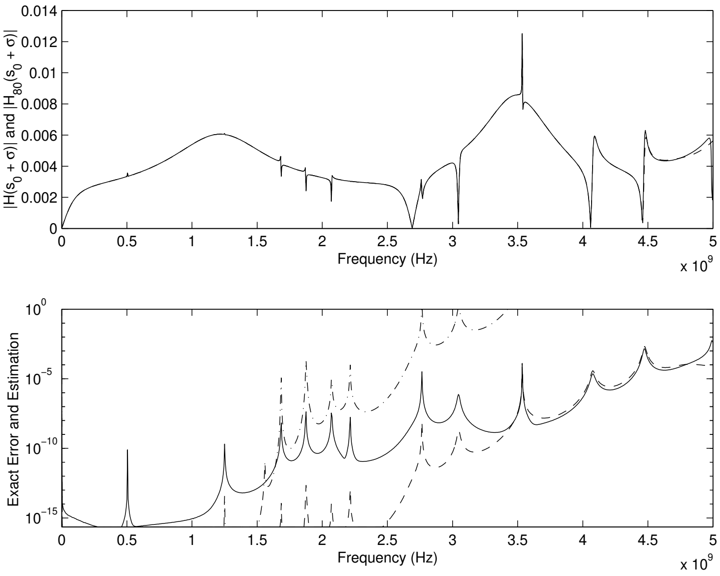

FIG. 5.4. *Numerical results of 80 PVL iterations of Example 3. Top: $|H(s)|$ (solid line) and its approximation $|H_{80}(s)|$ (dash line). Bottom: Exact error $|H(s) - H_{80}(s)|$ (solid line) and its estimation by (5.1) (dash-dot line) and by (5.2) (dash line).*

Applying the Laplace transform to the system (6.9) and (6.10), we obtain

$$sCX(s) = -GX(s) + BU(s) \tag{6.11}$$

$$Y(s) = L^H X(s) + DU(s) \tag{6.12}$$

where $X(s), Y(s)$ and $U(s)$ are the Laplace transforms of $x(t), y(t)$ and $u(t)$, respectively. Then the transfer function of the system is given by the $m \times \ell$ rational matrix function

$$H(s) = L^H (sC + G)^{-1} B + D \tag{6.13}$$

and $Y(s) = H(s)U(s)$. For computing $H(s)$, again we assume $D = 0$ and expand $H(s)$ about some point s_0, i.e., write $s = s_0 + \sigma$. Then

$$H(s) = H(s_0 + \sigma) = L^H (I - \sigma A)^{-1} R$$

where

$$A = -(s_0 C + G)^{-1} C \qquad \text{and} \qquad R = (s_0 C + G)^{-1} B.$$

In the following, we will simply denote $H(s_0 + \sigma)$ as $H(\sigma)$.

Let $p = \max\{m, \ell\}$ and $P_1, Q_1 \in \mathbf{R}^{N \times p}$ be orthogonal matrices such that

$$L = P_1 B_1 \quad \text{and} \quad R = Q_1 C_1,$$

where B_1 and C_1 are upper triangular. This this can be obtained by appropriately appending vectors, say random vectors, to L or R and then computing the QR-factorization. Thus

$$H(\sigma) \equiv H(s_0 + \sigma) = C_1^H P_1^H (I - \sigma A)^{-1} Q_1 B_1.$$

ETNA
Kent State University
etna@mcs.kent.edu

For ease of notation, we shall assume $B_1 = C_1 = I$ and therefore

$$H(\sigma) = P_1^H (I - \sigma A)^{-1} Q_1,$$

Now applying the block Lanczos process to the triplet $\{A, P_1, Q_1\}$, we have the equations (6.4) and (6.5). Then

$$H(\sigma) = Q_1^H (I - \sigma A)^{-1} P_1 = E_1^H (I - \sigma T_n)^{-1} E_1$$

and we can define the following reduced-order transfer function (a rational matrix function)

$$H_k(\sigma) = E_1^H (I - \sigma T_k)^{-1} E_1 \tag{6.14}$$

to approximate $H(\sigma)$. This is justified by the following moment matching property (under similar conditions on σ as in Section 3):

$$\begin{aligned} H(\sigma) - H_k(\sigma) &= \Sigma_{j=0}^{\infty} \sigma^j (E_1^H T_n^j E_1 - E_1^H T_k^j E_1) \\ &= \Sigma_{j=2k}^{\infty} \sigma^j (E_1^H T_n^j E_1 - E_1^H T_k^j E_1) \\ &= \sigma^{2k} B_2 \cdots B_k B_{k+1} C_{k+1} C_k \cdots C_2 + O(\sigma^{2k+1}), \end{aligned}$$

where we have used Neumann series expansion and Lemma 2. Furthermore, the following theorem gives an upper bound on the rational matrix approximation error.

THEOREM 6.2. *If $I - \sigma A$ and $I - \sigma T_k$ are invertible, then*

$$\begin{aligned} &\|H(\sigma) - H_k(\sigma)\| \le |\sigma|^2 \|\Gamma_{1k}(\sigma)\| \, \|\Gamma_{k1}(\sigma)\| \, \|\widehat{P}_{k+1}^H (I - \sigma A)^{-1} \widehat{Q}_{k+1}\| \\ &= |\sigma|^{2k} \|D_1 B_2 D_2 \cdots B_k D_k\| \, \|D_k C_k \cdots D_2 C_2 D_1\| \, \|\widehat{P}_{k+1}^H (I - \sigma A)^{-1} \widehat{Q}_{k+1}\| \end{aligned}$$

where

$$\Gamma_{1k}(\sigma) = E_1^H (I - \sigma T_k)^{-1} E_k, \quad \text{and} \quad \Gamma_{k1}(\sigma) = E_k^H (I - \sigma T_k)^{-1} E_1$$

and $D_k = (I - \sigma A_k)^{-1}$,

$$D_{j-1} = (I - \sigma A_{j-1} - \sigma^2 B_j D_j C_j)^{-1} \quad \text{for} \quad j = k, k-1, \cdots, 2.$$

Furthermore, for $|\sigma| < 1/\|A\|$, we have

$$\|H(\sigma) - H_k(\sigma)\| \le |\sigma|^2 \|\Gamma_{1k}(\sigma)\| \, \|\Gamma_{k1}(\sigma)\| \, \frac{\|\widehat{P}_{k+1}\| \, \|\widehat{Q}_{k+1}\|}{1 - |\sigma| \, \|A\|}.$$

Proof. Most of the proof is similar to that of Theorem 1. The main difference is in the second equality. Here we provide the proof of $E_k^H (I - \sigma T_k)^{-1} E_1 = \sigma^{k-1} D_k C_k \cdots D_2 C_2 D_1$ only. Using the Schur complement, it can be shown that

$$E_k^H (I - \sigma T_k)^{-1} E_1 = \sigma D_k C_k E_{k-1}^H (I - \sigma \widehat{T}_{k-1})^{-1} E_1 \tag{6.15}$$

where $I - \sigma \widehat{T}_{k-1}$ is the block tridiagonal matrix whose blocks are identical to $I - \sigma T_{k-1}$ except the $(k-1, k-1)$ block entry, which is $I - \sigma A_{k-1} - \sigma^2 B_k D_k C_k$. Now applying (6.15) recursively to itself, we obtain the desired equality. $\square$

The discussion on the computation of bounds and implementation issues in Section 3 is also applicable here. Also the same kind of bounds for $\|E_1^H (I - \sigma T_k)^{-1} E_k\|$ and $\|E_k^H (I -$

$\sigma T_k)^{-1}E_1\|$ as in Theorem 2 can be given. In general, the same kind of behavior as in the unblocked case would be expected.

Acknowledgement We are grateful for simulating discussions with William Smith and Rodney Slone during the course of this work. We would also like to thanks the referees and the editor R. B. Lehoucq for their valuable comments on the manuscript.

REFERENCES

[1] J. I. ALIAGA, D. L. BOLEY, R. W. FREUND AND HERNÁNDEZ, *A Lancos-type algorithm for multiple starting vectors*, Numerical Analysis Manuscript 95–11, AT& T Bell Laboratories, 1995.
[2] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, S. OSTROUCHOV AND D. SORENSEN, *LAPACK Users' Guide*, 2nd Ed., SIAM, Philadelphia, PA, 1995.
[3] Z. BAI, D. DAY AND Q. YE, *ABLE: An adaptive block lanczos method for non-hermitian eigenvalue problemsr*, Department of Mathematics Research Report 95-04, University of Kentucky, May 1995. To appear in SIAM J. Mat. Anal. Appl.
[4] Z. BAI, M. FAHEY AND G. GOLUB, *Some large scale matrix computation problems*, J. Comput. Appl. Math., 74:71–89, 1996.
[5] T. CHEN AND J. DEMMEL, *Balancing sparse matrices for computing eigenvalues*, Submitted to Lin. Alg. Appl. 1998
[6] E. CHIPROUT AND M. S. NAKHLA, *Asymptotic Waveform Evaluation*, Kluwer Academic Publishers, 1994.
[7] S. DAS, *Analysis of skin effect loss and incident field coupling in electrical interconnects using asymptotic waveform evaluation*, Ph.D. thesis, University of Kentucky, 1995.
[8] S. DEMKO, *Inverses of banded matrices and local convergence of spline projects*, SIAM J. Numer. Anal., 14:616–619, 1977.
[9] P. FELDMAN AND R. W. FREUND, *Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm*, Numerical Analysis Manuscript 94–11, AT& T Bell Laboratories, October 1994.
[10] P. FELDMAN AND R. W. FREUND, *Efficient linear circuit analysis by Padé approximation via the Lanczos process*, IEEE Trans. Comput.-Aided Design, CAD-14:639–649, 1995.
[11] R. W. FREUND, M. H. GUTKNECHT, AND N. M. NACHTIGAL, *An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices*, SIAM J. Sci. Comput., 14:137–158, 1993.
[12] K. GALLIVAN, E. GRIMME, AND P. VAN DOOREN, *Asymptotic waveform evaluation via a Lanczos method*, Appl. Math. Lett., 7:75–80, 1994.
[13] G. GOLUB AND G. MEURANT, *Matrices, moments and quadrature*, Report SCCM-93-07, Computer Science Department, Stanford University, September 1993.
[14] G. GOLUB AND C. VAN LOAN,. *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996.
[15] W. B. GRAGG, *Matrix interpretations and applications of the continued fraction algorithm*, Rocky Mountain J. Math., 5:213–225, 1974.
[16] W. B. GRAGG AND A. LINDQUIST, *On the partial realization problem*, Linear Algebra Appl., 50:227–319, 1983.
[17] E. J. GRIMME, *Krylov projection methods for model reduction*, Ph.D. dissertation, Univ. of Illinois at Urbana-Champaign, 1997.
[18] M. H. GUTKNECHT, *A completed theory of the unsymmetric Lanczos process and related algorithms, Part I and II*, SIAM J. Matrix Anal. Appl. Part I, 13(1992), pp.594–639, Part II, 15(1994), pp.15–58.
[19] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, PA, 1996.
[20] I. M. JAIMOUKHA AND E. M. KASENALLY, *Oblique projection methods for large scale model reductions*, SIAM J. Matrix Anal. Appl., 16:602–627, 1995.
[21] G. MEURANT, *A review of the inverse of symmetric tridiagonal and block tridiagonal matrices*, SIAM J. Matrix Anal. Appl., 13:707–728, 1992.
[22] B. PARLETT, *Reduction to tridiagonal form and minimal realizations*, SIAM J. Matrix Anal. Appl., 13(2):567–593, 1992.
[23] B. PARLETT AND C. REINSCH, *Balancing a matrix for calculation of eigenvalues and eigenvectors*, Numer. Math., 13:293–304, 1969.
[24] B. N. PARLETT, D. R. TAYLOR AND Z. A. LIU, *A look-ahead Lanczos algorithm for unsymmetric matrices*, Math. Comp., 44:105–124, 1985.
[25] L. T. PILLAGE AND R. A. ROHRER, *Asymptotic waveform evaluation for timing analysis*, IEEE Trans. Comput.-Aided Design, 9:353–366, 1990.

[26] A. E. RUEHLI, *Equivalent circuit models for three-dimensional multi-conductor systems*, IEEE Trans. Microwave Theory Tech., 22:216–221, 1974.

[27] Y. SAAD, *Numerical methods for large eigenvalue problems*, Halsted Press, Div. of John Wiley & Sons, Inc., New York, NY, 1992.

[28] R. C. WARD, *Balancing the generalized eigenvalue problem*, SIAM J. Sci. Statist. Comp., 2:141–152, 1981.

[29] Q. YE, *A convergence analysis for nonsymmetric Lanczos algorithms*, Math. Comp., 56:677–691, 1991.

[30] ——— *A breakdown-free variation of the nonsymmetric Lanczos algorithms*, Math. Comp., 62:179–207, 1994.