Efficient Linear Circuit Analysis by Padé Approximation via the Lanczos Process

Peter Feldmann, *Member, IEEE*, and Roland W. Freund

Abstract—**In this paper, we introduce PVL, an algorithm for computing the Padé approximation of Laplace-domain transfer functions of large linear networks via a Lanczos process. The PVL algorithm has significantly superior numerical stability, while retaining the same efficiency as algorithms that compute the Padé approximation directly through moment matching, such as AWE [1], [2] and its derivatives. As a consequence, it produces more accurate and higher-order approximations, and it renders unnecessary many of the heuristics that AWE and its derivatives had to employ. The algorithm also computes an error bound that permits to identify the true poles and zeros of the original network. We present results of numerical experiments with the PVL algorithm for several large examples.**

I. Introduction

CIRCUIT simulation tasks, such as the accurate prediction of interconnect effects at the board and chip level, or analog circuit analysis with full accounting of parasitic elements, may require the solution of large linear networks. These networks can become extremely large, especially when circuits are automatically extracted from layout, or contain models of distributed elements, such as transmission lines, ground planes, antennas, and other three-dimensional structures. The use of SPICE-like simulators, which are based on the integration of nonlinear ordinary differential equations, would be inefficient or even prohibitive for such large problems.

In recent years, the **A**symptotic **W**aveform **E**valuation (AWE) algorithm [1]–[3] based on Padé approximation [4] has emerged as the method of choice for the efficient analysis of large linear circuits. Its success in tackling real-world problems, the most notorious of which being the verification of the clock-distribution network on the DEC Alpha chip, attracted a lot of interest and spawned a substantial amount of related research.

AWE is based on approximating the Laplace-domain transfer function of a linear network by a reduced-order model, containing only a relatively small number of dominant poles and zeros. Such reduced-order models can be used to predict the time-domain or frequency-domain response of the linear network over a predetermined range of excitation frequencies. They are also useful in the simulation of nonlinear circuits containing large linear subnetworks. The nonlinear simulation can be made significantly more efficient with little loss of accuracy when the large linear subnetworks are replaced by reduced-order models [5].

Manuscript received June 13, 1994; revised November 15, 1994. This paper was recommended by Associate Editor J. White.

The authors are with AT&T Bell Laboratories, Murray Hill, NJ 07974 USA.

IEEE Log Number 9409640.

Despite its spectacular success, AWE suffers from a number of fundamental numerical limitations. In particular, each run of AWE produces only a fairly small number of accurate poles and zeros. The proposed remedial techniques, such as scaling, frequency shifting, and complex frequency hopping, are sometimes heuristic, hard to apply automatically, and may be computationally expensive. Another shortcoming of AWE is the absence of a theoretically solid procedure to predict the accuracy of the approximating reduced-order model [3], [6], [7].

In this paper, we introduce a new, numerically stable algorithm that computes the Padé approximation of a linear circuit via the Lanczos process [8]. This algorithm, called PVL (**P**adé **V**ia **L**anczos), can be used to generate an arbitrary number of poles and zeros (even all of them) with little numerical degradation. Moreover, PVL computes a quality measure for the poles and zeros it produces. The computational cost per order of approximation is practically the same as for AWE.

The remainder of the paper is organized as follows. In Section II, we demonstrate the numerical limitations of algorithms that compute the Padé approximation from time-domain moments of the frequency response, as it is done in AWE. In Section III, we derive the PVL algorithm and present some of its properties. In Section IV, we discuss practical aspects of the PVL algorithm. In Section V, we present results of numerical experiments with PVL for a variety of examples. In Section VI, we make some concluding remarks.

II. Limitations of Current Algorithms

A. System Reduction by Padé Approximation

Using any circuit-equation formulation method such as modified nodal analysis, sparse tableau, etc. [9], a lumped, linear, time-invariant circuit can be described by the following system of first-order differential equations:

$$\begin{aligned} \mathbf{C}\dot{\mathbf{x}} &= -\mathbf{G}\mathbf{x} + \mathbf{b}u, \\ y &= \mathbf{l}^T\mathbf{x} + du. \end{aligned} \tag{1}$$

Here, the vector $\mathbf{x}$ represents the circuit variables, the matrix $\mathbf{G}$ represents the contribution of memoryless elements, such as resistors, $\mathbf{C}$ represents contribution from memory elements, such as capacitors and inductors, y is the output of interest, and the terms $\mathbf{b}u$ and du represent excitations from independent sources.

We are interested in determining the impulse-response of the linear circuit with zero initial-conditions, which, in turn,

can be used to determine the response to any excitation. We apply the Laplace transform to the system (1), assuming zero initial conditions. In order to simplify the notation, we ignore the term du. Then, from (1), we obtain

$$\begin{aligned} s\mathbf{C}\mathbf{X} &= -\mathbf{G}\mathbf{X} + \mathbf{b}U, \\ Y &= \mathbf{l}^T\mathbf{X}, \end{aligned} \tag{2}$$

where $\mathbf{X}$, U, and Y denote the Laplace transforms of $\mathbf{x}$, u, and y, respectively. It follows from (2) that the Laplace-domain impulse response, defined as $H(s) = Y(s)/U(s)$, is given by

$$H(s) = \mathbf{l}^T(\mathbf{G} + s\mathbf{C})^{-1}\mathbf{b}. \tag{3}$$

The function $H(s)$ is called the *frequency response or transfer function* of the circuit.

Let $s_0 \in \mathbb{C}$ be an arbitrary, but fixed expansion point such that the matrix $\mathbf{G} + s_0\mathbf{C}$ is nonsingular. Using the change of variables $s = s_0 + \sigma$ and setting

$$\mathbf{A} = -(\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{C}, \quad \mathbf{r} = (\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{b}, \tag{4}$$

we can rewrite (3) as follows:

$$\begin{aligned} H(s_0 + \sigma) &= \mathbf{l}^T(\sigma\mathbf{C} + \mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{b} \\ &= \mathbf{l}^T(\mathbf{I} - \sigma\mathbf{A})^{-1}\mathbf{r}. \end{aligned}$$

Assuming that the matrix $\mathbf{A}$ is diagonalizable, we obtain

$$H(s_0 + \sigma) = \underbrace{\mathbf{l}^T\mathbf{S}}_{=\mathbf{f}^T}(\mathbf{I} - \sigma\Lambda)^{-1}\underbrace{\mathbf{S}^{-1}\mathbf{r}}_{=\mathbf{g}}, \tag{5}$$

where $\mathbf{A} = \mathbf{S}\Lambda\mathbf{S}^{-1}, \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_N)$ is a diagonal matrix whose diagonals elements are the eigenvalues of $\mathbf{A}$, and the matrix $\mathbf{S}$ contains the corresponding eigenvectors as columns. From (5), we get

$$H(s_0 + \sigma) = \sum_{j=1}^{N} \frac{f_j g_j}{1 - \sigma\lambda_j}, \tag{6}$$

where f_j and g_j are the components of the vectors $\mathbf{f}$ and $\mathbf{g}$.

The numerical computation of all eigenvectors and eigenvalues of the matrix $\mathbf{A}$ becomes prohibitively expensive as soon as its size reaches a few hundreds, and therefore, the only practical way to obtain an expression for the frequency response is through an approximation.

For each pair of integers $p, q \geq 0$, the Padé approximation (of type (p/q)) to the network frequency response $H(s_0 + \sigma)$ is defined as the rational function

$$H_{p,q}(s_0 + \sigma) = \frac{b_p\sigma^p + \cdots + b_1\sigma + b_0}{a_q\sigma^q + \cdots + a_1\sigma + 1} \tag{7}$$

whose Taylor series about $\sigma = 0$ agrees with the Taylor series of $H(s_0 + \sigma)$ in at least the first $p + q + 1$ terms, i.e.,

$$H_{p,q}(s_0 + \sigma) = H(s_0 + \sigma) + \mathcal{O}(\sigma^{p+q+1}).$$

The coefficients $a_1, \cdots, a_q, b_0, b_1, \cdots, b_p$ of the Padé approximation (7) are uniquely determined by the first $p + q + 1$ Taylor coefficients of the frequency response. The roots of the denominator and numerator polynomials in (7) represent the dominant poles and zeros of the system, respectively.

In view of (6), the function $H(s_0 + \sigma)$ itself is rational, with numerator and denominator polynomials of degree at most $N - 1$ and N, respectively. Thus, in the context of frequency-response approximations, it is very natural to choose $p = q - 1$ in (7), so that the Padé approximation is of the same form as the original frequency response. In the following, we always assume that $p = q - 1$, and we set $H_q := H_{q-1,q}$. We refer to H_q as the qth *Padé approximant* to the frequency response H. By using a partial-fraction decomposition, we can write H_q in the form

$$H_q(s_0 + \sigma) = k_\infty + \sum_{j=1}^{q'} \frac{k_j}{\sigma - p_j}, \tag{8}$$

where $q' \leq q$.

The Taylor coefficients necessary for the Padé approximant H_q result from the following expansion of $H(s)$ about s_0:

$$H(s_0 + \sigma) = \mathbf{l}^T(\mathbf{I} + \sigma\mathbf{A} + \sigma^2\mathbf{A}^2 + \cdots)\mathbf{r} = \sum_{k=0}^{\infty} m_k\sigma^k, \tag{9}$$

where

$$m_k = \mathbf{l}^T\mathbf{A}^k\mathbf{r}, \quad k = 0, 1, \cdots. \tag{10}$$

Note that in the case when the expansion point in (9) is chosen as the origin, i.e., $s_0 = 0$, the coefficients $m_k, k = 0, 1, \cdots$, are, up to a constant factor, the time-domain moments of the circuit response. Indeed, we have

$$\begin{aligned} H(s) &= \int_0^\infty h(t)e^{-st}\,dt \\ &= \int_0^\infty h(t)\left[1 - st + \frac{1}{2}s^2t^2 - \cdots\right]dt \\ &= \int_0^\infty h(t)\,dt - s\int_0^\infty th(t)\,dt \\ &\quad + s^2\frac{1}{2}\int_0^\infty t^2h(t)\,dt - \cdots. \end{aligned}$$

Because of this analogy, we will always refer to the Taylor coefficients (10) as the *moments* of the frequency-response function $H(s_0 + \sigma)$.

B. Implementation of the Padé Approximation in AWE

In AWE, the Padé approximant H_q is obtained via explicit computation of the leading $2q$ moments $m_0, m_1, \cdots, m_{2q-1}$ of H.

To this end, one first generates the vectors $\mathbf{u}_0 = \mathbf{r}, \mathbf{u}_1 = \mathbf{A}\mathbf{r}, \mathbf{u}_2 = \mathbf{A}^2\mathbf{r}, \cdots, \mathbf{u}_{2q-1} = \mathbf{A}^{2q-1}\mathbf{r}$ by recursive solution of the linear systems

$$(\mathbf{G} + s_0\mathbf{C})\mathbf{u}_k = -\mathbf{C}\mathbf{u}_{k-1}, \quad k = 1, 2, \cdots, 2q - 1, \tag{11}$$

with the initial vector

$$\mathbf{u}_0 = (\mathbf{G} + s_0\mathbf{C})^{-1}\mathbf{b}.$$

Observe that the recursive computation of the vectors $\mathbf{u}_k$ can be performed very efficiently. The matrix $\mathbf{G} + s_0\mathbf{C}$ is LU-factored exactly once. Then each vector $\mathbf{u}_k$ is obtained by re-solving the system with a different right-hand side at the

cost of only one forward-backward substitution. The moments are then computed as $m_k = \mathbf{l}^T \mathbf{u}_k$, $k = 0, 1, 2, \cdots, 2q-1$.

As the next step, AWE computes the coefficients of the denominator polynomial of the representation (7) of H_q via solution of the linear system

$$\mathbf{M}_q \begin{bmatrix} a_q \\ a_{q-1} \\ \vdots \\ a_1 \end{bmatrix} = - \begin{bmatrix} m_q \\ m_{q+1} \\ \vdots \\ m_{2q-1} \end{bmatrix}. \tag{12}$$

The coefficient matrix $\mathbf{M}_q$ of (12) is the so-called *moment matrix* given by

$$\mathbf{M}_q = [m_{j+k-2}]_{j,k=1,2,\cdots,q} = \begin{bmatrix} m_0 & m_1 & \cdots & m_{q-1} \\ m_1 & \iddots & \iddots & \vdots \\ \vdots & \iddots & \iddots & m_{2q-3} \\ m_{q-1} & \cdots & m_{2q-3} & m_{2q-2} \end{bmatrix}.$$

The poles p_j of H_q in (8) are then obtained as the roots of the equation

$$a_q \sigma^q + a_{q-1}\sigma^{q-1} + \cdots + a_1 \sigma + 1 = 0.$$

Finally, the constant k_∞ and the residues k_j in (8) are computed by solving another linear system of order q, see, e.g., [10] for details.

As q is increased, one would expect more and more accurate approximations H_q of the exact frequency response H. Unfortunately, this is not the case when the Padé approximant H_q is generated with AWE. Indeed, typically H_q improves only for values of q up to about $q = 10$, and after that the process stagnates. Fig. 1 illustrates this behavior. Here, we tried to simulate the voltage gain of a filter with our own implementation of AWE written in MATLAB^{TM} [11], which is based on double-precision arithmetic with about 16 decimal digits. We show the exact voltage gain and the approximations generated by AWE for $q = 2, 5, 8$. The results for $q > 8$ were virtually identical with the curve for $q = 8$, and the process never converged to the exact voltage gain.

The reason for the stagnation of AWE is the particular computation of the Padé approximation used in AWE, and not the Padé approximation itself. More specifically, the explicit use of the moments results in extremely ill-conditioned numerical computations. This is especially the case for the numerical solution of the linear system (12). The condition number, $\text{cond}(\mathbf{M}_q)$, of the coefficient matrix $\mathbf{M}_q$ of (12) is a measure for how round-off error affects the accuracy of the numerically computed solution of (12). Each increase of $\text{cond}(\mathbf{M}_q)$ by a factor of 10 signals the loss of one decimal digit of accuracy in the computed solution. In particular, if double-precision with 16 decimal digits is used, then the computed solution must be expected to be meaningless if $\text{cond}(\mathbf{M}_q) = \mathcal{O}(10^{16})$. In the first column of Table I, we list $\text{cond}(\mathbf{M}_q)$ for the moment matrices corresponding to the simulation of the voltage gain of a filter. Clearly, the moment matrices are extremely ill-conditioned.

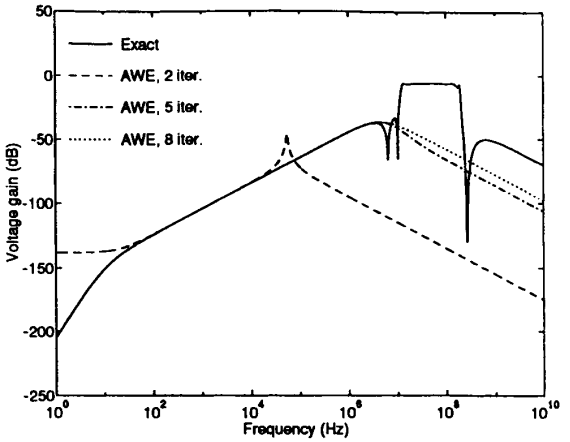

Fig. 1. Results for simulation of voltage gain with AWE.

TABLE I
CONDITION NUMBERS OF MOMENT MATRICES FOR SIMULATION OF VOLTAGE GAIN WITH AWE

q	$\text{cond}(\mathbf{M}_q)$	$\text{cond}(\mathbf{M}_q^{(1)})$	$\text{cond}(\mathbf{M}_q^{(2)})$	$\text{cond}(\mathbf{M}_q^{(3)})$
2	9.59e+05	2.62e+00	2.62e+00	1.07e+00
3	2.91e+15	5.14e+03	5.12e+03	2.71e+01
4	3.18e+26	1.13e+09	1.13e+09	1.30e+07
5	2.16e+35	2.34e+12	2.33e+12	3.01e+10
6	3.68e+46	2.72e+17	2.38e+17	5.99e+15
7	1.97e+53	2.96e+17	6.19e+17	7.05e+15
8	3.34e+58	1.25e+18	1.78e+18	2.67e+16
9	1.66e+65	1.18e+18	1.52e+18	7.86e+15
10	1.30e+72	1.57e+18	7.50e+17	5.38e+16
20	5.41e+130	1.27e+18	1.92e+18	9.01e+16
30	2.87e+191	7.26e+18	3.51e+18	5.23e+17

The problem of ill-conditioning in AWE was noticed early on, and as a remedy, it was proposed to use scaling [3]. The idea is to replace the original moments m_k by $\xi^k m_k$, $k = 0, 1, \cdots, 2q-1$, where $\xi > 0$ is a suitably chosen scaling factor. Three natural choices are

$$\xi_1 = \frac{1}{\|\mathbf{A}\|_2}, \quad \xi_2 = \frac{|m_0|}{|m_1|}, \quad \text{and} \quad \xi_3 = \left(\frac{|m_0|}{|m_{2q-1}|}\right)^{1/(2q-1)}. \tag{13}$$

The first choice in (13) corresponds to scaling the matrix $\mathbf{A}$ in (10) to have Euclidean norm 1. The second choice, proposed in [3], is such that the two first scaled moments, m_0 and $\xi_2 m_1$, have the same magnitude. The third choice, suggested in [10], is such that, after scaling, the first and the last computed moment, m_0 and $\xi_3^{2q-1} m_{2q-1}$, have the same magnitude. While scaling reduces the ill-conditioning somewhat, the scaled moment matrices

$$\mathbf{M}_q^{(l)} = [\xi_l^{j+k-2} m_{j+k-2}]_{j,k=1,2,\cdots,q}, \quad l = 1, 2, 3, \tag{14}$$

for all three strategies (13) still reach the critical value 10^{16} for fairly small values of q. In Table I, we also list the condition numbers of the scaled moment matrices (14) for the case of the simulation of the voltage gain of a filter (*cf.* Fig. 1). In this case, the third scaling strategy in (13) yields the smallest condition numbers. However, we still have

$\mathrm{cond}(\mathbf{M}_q^{(3)}) = \mathcal{O}(10^{16})$ after only $q = 8$ iterations, and after that AWE stagnates completely. In fact, the curves shown in Fig. 1 were computed with AWE, using the optimal third scaling strategy.

The severe numerical problems with the AWE approach of obtaining Padé approximants via direct computation of the moments can be explained as follows. The generation of the vectors $\mathbf{u}_k = \mathbf{A}^k\mathbf{r}$ in (11) corresponds to vector iteration with the matrix $\mathbf{A}$, and this process converges rapidly to an eigenvector corresponding to an eigenvalue of $\mathbf{A}$ with largest absolute value. As a result, the numerically computed vector $\mathbf{u}_k$ and also the moment $m_k = \mathbf{l}^T\mathbf{u}_k$ generated from it practically contain only information corresponding to **one** eigenvalue of $\mathbf{A}$, even for fairly small values of k. On the other hand, in view of (6), the function H to be approximated clearly depends on **all** eigenvalues of $\mathbf{A}$. Therefore, in general, for the accuracy of the numerically computed Padé approximant H_q to improve with increasing q, an algorithm must be able to recover information about more than one eigenvalue of $\mathbf{A}$. As we just explained, this is not the case for the algorithm used in AWE.

III. The PVL Algorithm

We now describe the PVL algorithm that exploits the intimate connection [12] between Padé approximation and the Lanczos process to elude the direct computation of the moments.

A. The Lanczos Algorithm

The classical Lanczos process [8] is an iterative procedure for the successive reduction of a square matrix $\mathbf{A}$ to a sequence of tridiagonal matrices. First, we briefly recall this algorithm and some of its key properties.

Algorithm 1 (Lanczos algorithm [8])

0) Set $\rho_1 = \|\mathbf{r}\|_2, \eta_1 = \|\mathbf{l}\|_2, \mathbf{v}_1 = \mathbf{r}/\rho_1$, and $\mathbf{w}_1 = \mathbf{l}/\eta_1$. Set $\mathbf{v}_0 = \mathbf{w}_0 = 0$ and $\delta_0 = 1$.

For $n = 1, 2, \cdots, q$ do:

1) Compute $\delta_n = \mathbf{w}_n^T\mathbf{v}_n$.

2) Set

$$\alpha_n = \frac{\mathbf{w}_n^T\mathbf{A}\mathbf{v}_n}{\delta_n}, \beta_n = \frac{\delta_n}{\delta_{n-1}}\eta_n, \gamma_n = \frac{\delta_n}{\delta_{n-1}}\rho_n. \quad (15)$$

3) Set

$$\mathbf{v} = \mathbf{A}\mathbf{v}_n - \mathbf{v}_n\alpha_n - \mathbf{v}_{n-1}\beta_n,$$
$$\mathbf{w} = \mathbf{A}^T\mathbf{w}_n - \mathbf{w}_n\alpha_n - \mathbf{w}_{n-1}\gamma_n.$$

4) Set $\rho_{n+1} = \|\mathbf{v}\|_2, \eta_{n+1} = \|\mathbf{w}\|_2$, and

$$\mathbf{v}_{n+1} = \frac{\mathbf{v}}{\rho_{n+1}}, \quad \mathbf{w}_{n+1} = \frac{\mathbf{w}}{\eta_{n+1}}.$$

We remark that in Algorithm 1, a breakdown, triggered by division by 0, will occur if one encounters $\delta_n = 0$ in (15). Furthermore, division by a nonzero yet small number $\delta_n \approx 0$ in (15) may result in numerical instabilities. However, these problems can be remedied by using a so-called look-ahead variant of the Lanczos algorithm, see [13] and [14]. In fact, in our implementation of the PVL algorithm we employ the look-ahead Lanczos algorithm described in [14].

The quantities generated by Algorithm 1 have the following properties:

- The vectors $\{\mathbf{v}_n\}_{n=1}^{q+1}$ and $\{\mathbf{w}_n\}_{n=1}^{q+1}$ are biorthogonal:

$$\mathbf{w}_j^T\mathbf{v}_k = \begin{cases}\delta_j, & \text{if } j = k,\\ 0, & \text{if } j \neq k,\end{cases} \qquad j, k = 1, 2, \cdots, q+1.$$

- The tridiagonal matrices

$$\mathbf{T}_q = \begin{bmatrix}\alpha_1 & \beta_2 & 0 & \cdots & 0\\ \rho_2 & \alpha_2 & \beta_3 & \ddots & \vdots\\ 0 & \rho_3 & \ddots & \ddots & 0\\ \vdots & \ddots & \ddots & \ddots & \beta_q\\ 0 & \cdots & 0 & \rho_q & \alpha_q\end{bmatrix} \quad (16)$$

and

$$\tilde{\mathbf{T}}_q = \begin{bmatrix}\alpha_1 & \gamma_2 & 0 & \cdots & 0\\ \eta_2 & \alpha_2 & \gamma_3 & \ddots & \vdots\\ 0 & \eta_3 & \ddots & \ddots & 0\\ \vdots & \ddots & \ddots & \ddots & \gamma_q\\ 0 & \cdots & 0 & \eta_q & \alpha_q\end{bmatrix}$$

have the following relation to the original matrix $\mathbf{A}$:

$$\mathbf{A}\mathbf{V}_q = \mathbf{V}_q\mathbf{T}_q + [0 \quad \cdots \quad 0 \ \mathbf{v}_{q+1}]\rho_{q+1},$$
$$\mathbf{A}^T\mathbf{W}_q = \mathbf{W}_q\tilde{\mathbf{T}}_q + [0 \quad \cdots \quad 0 \ \mathbf{w}_{q+1}]\eta_{q+1}, \quad (17)$$

where

$$\mathbf{V}_q = [\mathbf{v}_1 \ \mathbf{v}_2 \ \cdots \ \mathbf{v}_q]$$
and
$$\mathbf{W}_q = [\mathbf{w}_1 \ \mathbf{w}_2 \ \cdots \ \mathbf{w}_q] \quad (18)$$

are matrices whose columns are just the Lanczos vectors.

- The matrices $\tilde{\mathbf{T}}_q$ and $\mathbf{T}_q$ satisfy

$$\tilde{\mathbf{T}}_q^T = \mathbf{D}_q\mathbf{T}_q\mathbf{D}_q^{-1}, \quad (19)$$

where

$$\mathbf{D}_q = \mathbf{W}_q^T\mathbf{V}_q = \mathrm{diag}(\delta_1, \delta_2, \cdots, \delta_q). \quad (20)$$

- One key property of Algorithm 1 is that it produces a very useful small-dimensional approximation, namely the matrix $\mathbf{T}_q$, to the (usually large-dimensional) matrix $\mathbf{A}$. The matrix $\mathbf{T}_q$ is—in the sense of an oblique projection process—the best $q \times q$ approximation to $\mathbf{A}$ that can be obtained using only information from the two *Krylov subspaces*

$$K_q(\mathbf{v}_1, \mathbf{A}) = \mathrm{span}\{\mathbf{v}_1, \mathbf{A}\mathbf{v}_1, \cdots, \mathbf{A}^{q-1}\mathbf{v}_1\}$$

and

$$K_q(\mathbf{w}_1, \mathbf{A}^T) = \mathrm{span}\{\mathbf{w}_1, \mathbf{A}^T\mathbf{w}_1, \cdots, (\mathbf{A}^T)^{q-1}\mathbf{w}_1\}.$$

We remark that the vectors $\{\mathbf{v}_n\}_{n=1}^q$ and $\{\mathbf{w}_n\}_{n=1}^q$ just span the spaces $K_q(\mathbf{v}_1, \mathbf{A})$ and $K_q(\mathbf{w}_1, \mathbf{A}^T)$, respectively. It can be shown that $\mathbf{T}_q$ is the result of the projection of $\mathbf{A}$ onto $K_q(\mathbf{v}_1, \mathbf{A})$ and orthogonally to $K_q(\mathbf{w}_1, \mathbf{A}^T)$. We stress that the resulting approximation

$\mathbf{T}_q$ is very good even if q is much smaller than the order N of the $N \times N$ matrix $\mathbf{A}$. In fact, the Lanczos algorithm is mostly applied to very large matrices, and typically, $q \ll N$. In the next section, we will demonstrate that the matrix $\mathbf{T}_q$ is also a best approximation to $\mathbf{A}$ in the sense of matching the maximal number of moments; this is just the Padé connection.

B. The Padé Connection

Next, we demonstrate the connection of the Lanczos process to Padé approximation. Using the first relation in (17) and the fact that $\mathbf{T}_q$ is tridiagonal, one can show that

$$\begin{aligned}\mathbf{A}^j\mathbf{r} &= \rho_1\mathbf{A}^j\mathbf{v}_1 \\ &= \rho_1\mathbf{A}^j\mathbf{V}_q\mathbf{e}_1 \\ &= \rho_1\mathbf{V}_q\mathbf{T}_q^j\mathbf{e}_1, \quad j = 0, 1, \cdots, q-1, \end{aligned} \tag{21}$$

where $\mathbf{e}_1 = [1 \ 0 \ \cdots \ 0]^T \in \mathbb{R}^q$ is the first unit vector in $\mathbb{R}^q$. Similarly, from the second relation in (17), one can deduce that

$$\begin{aligned}\mathbf{l}^T\mathbf{A}^j &= \eta_1\mathbf{e}_1^T(\tilde{\mathbf{T}}_q^T)^j\mathbf{W}_q^T \\ &= \eta_1\delta_1\mathbf{e}_1^T\mathbf{T}_q^j\mathbf{D}_q^{-1}\mathbf{W}_q^T, \quad j = 0, 1, \cdots, q-1. \end{aligned} \tag{22}$$

We note that the second equation in (22) follows from (19).

On the other hand, by (10), each moment m_k can be written as follows:

$$m_k = \mathbf{l}^T\mathbf{A}^k\mathbf{r} = \left(\mathbf{l}^T\mathbf{A}^{k'}\right)\left(\mathbf{A}^{k''}\mathbf{r}\right), \tag{23}$$

where $k = k' + k''$. If $k \leq 2q-2$, we can find k' and k'' with $0 \leq k', k'' \leq q-1$, and from (21)–(23) it follows that

$$\begin{aligned}m_k &= \left(\eta_1\delta_1\mathbf{e}_1^T\mathbf{T}_q^{k'}\mathbf{D}_q^{-1}\mathbf{W}_q^T\right)\left(\rho_1\mathbf{V}_q\mathbf{T}_q^{k''}\mathbf{e}_1\right) \\ &= \eta_1\rho_1\delta_1\mathbf{e}_1^T\mathbf{T}_q^{k'}\mathbf{D}_q^{-1}\underbrace{\mathbf{W}_q^T\mathbf{V}_q}_{=\mathbf{D}_q}\mathbf{T}_q^{k''}\mathbf{e}_1. \end{aligned} \tag{24}$$

Note that $\eta_1\rho_1\delta_1 = \mathbf{l}^T\mathbf{r}$, and thus, from (24), we get

$$m_k = (\mathbf{l}^T\mathbf{r}) \cdot (\mathbf{e}_1^T\mathbf{T}_q^k\mathbf{e}_1) \quad \text{for all } k = 0, 1, \cdots, 2q-2. \tag{25}$$

Furthermore, it can be shown that the relation (25) also holds for $k = 2q-1$. Thus, by (25), we have

$$\begin{aligned}\mathbf{l}^T\mathbf{r} \cdot \mathbf{e}_1^T(\mathbf{I} - \sigma\mathbf{T}_q)^{-1}\mathbf{e}_1 &= \mathbf{l}^T\mathbf{r}\sum_{k=0}^{\infty}\mathbf{e}_1^T\mathbf{T}_q^k\mathbf{e}_1\sigma^k \\ &= \sum_{k=0}^{2q-1} m_k\sigma^k + \mathcal{O}(\sigma^{2q}), \end{aligned}$$

and consequently,

$$H_q(s_0 + \sigma) = \mathbf{l}^T\mathbf{r} \cdot \mathbf{e}_1^T(\mathbf{I} - \sigma\mathbf{T}_q)^{-1}\mathbf{e}_1 \tag{26}$$

is just the qth Padé approximant of H.

In analogy to the representation (6) of the exact frequency response H, we can rewrite the expression (26) of H_q in terms of the eigendecomposition $\mathbf{T}_q = \mathbf{S}_q\Lambda_q\mathbf{S}_q^{-1}$ of the Lanczos matrix $\mathbf{T}_q$:

$$\begin{aligned}H_q(s_0 + \sigma) &= \mathbf{l}^T\mathbf{r} \cdot \underbrace{\mathbf{e}_1^T\mathbf{S}_q}_{=\mu^T}(\mathbf{I} - \sigma\Lambda_q)^{-1}\underbrace{\mathbf{S}_q^{-1}\mathbf{e}_1}_{=\nu} \\ &= \sum_{j=1}^{q}\frac{\mathbf{l}^T\mathbf{r} \cdot \mu_j\nu_j}{1 - \sigma\lambda_j}. \end{aligned} \tag{27}$$

Here, $\Lambda_q = \operatorname{diag}(\lambda_1, \lambda_2, \cdots, \lambda_q)$ contains the eigenvalues of $\mathbf{T}_q$, and μ_j and ν_j are the components of the vectors μ and ν. Finally, from (27), we immediately obtain the pole/residue representation of the Padé approximant:

$$H_q(s_0 + \sigma) = k_\infty + \sum_{\substack{j=1 \\ \lambda_j \neq 0}}^{q}\frac{-\mathbf{l}^T\mathbf{r} \cdot \mu_j\nu_j/\lambda_j}{\sigma - 1/\lambda_j}. \tag{28}$$

Note that the term k_∞ in (28) may result if one of the eigenvalues of $\mathbf{T}_q$ is zero.

Recall that, in PVL, we actually use the look-ahead version [14] of the Lanczos Algorithm 1. We stress that the Lanczos-Padé connection holds true also for the look-ahead case. The look-ahead Lanczos algorithm [14] again generates a $q \times q$ approximation $\mathbf{T}_q$ to the matrix $\mathbf{A}$, where $\mathbf{T}_q$ is now block tridiagonal, with a few small blocks and mostly blocks of size one. Similarly, it produces a matrix $\mathbf{D}_q$ that is now block diagonal. In terms of these matrices $\mathbf{T}_q$ and $\mathbf{D}_q$, the qth Padé approximant H_q of H is then given by

$$H_q(s_0 + \sigma) = \|\mathbf{l}\| \cdot \|\mathbf{r}\| \cdot (\mathbf{D}_q^T\mathbf{e}_1)^T(\mathbf{I} - \sigma\mathbf{T}_q)^{-1}\mathbf{e}_1. \tag{29}$$

If no look-ahead steps occur, then $\mathbf{T}_q$ is just the tridiagonal matrix defined in (16). Furthermore, we have

$$\mathbf{D}_q^T\mathbf{e}_1 = \frac{\mathbf{l}^T\mathbf{r}}{\|\mathbf{l}\| \cdot \|\mathbf{r}\|}\mathbf{e}_1,$$

and thus (29) just reduces to (26) for the no-look-ahead case. To keep the exposition as simple as possible, we always use the no-look-ahead formula (26) throughout this paper.

We stress that the computational costs for the look-ahead Lanczos algorithm [14] are essentially the same as for the classical Lanczos Algorithm 1. In fact, the number of matrix-vector products, which dominate the computational costs, are identical for the look-ahead and the classical algorithm. The remaining computations are inner products of vectors of length N (the order of the $N \times N$ matrix $\mathbf{A}$) and SAXPY's of vectors of length N. A SAXPY operation is $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{y}\alpha$, where $\mathbf{x}$ and $\mathbf{y}$ are vectors and α is a scalar. The look-ahead and the classical algorithm involve the same number of inner products. The only overhead of the look-ahead algorithm is the computation of a few extra SAXPY's in the case that a look-ahead step is performed.

C. Computing Poles, Residues, and Zeros

The derivation in the previous section shows that the pole/residue representation (28) of the Padé approximant H_q can be obtained by running the Lanczos algorithm and by computing an eigendecomposition of the Lanczos matrix $\mathbf{T}_q$. The resulting computational procedure is the PVL algorithm.

Algorithm 2. (Sketch of the PVL algorithm):

1) Run q steps of the Lanczos process (Algorithm 1) to obtain the tridiagonal matrix $\mathbf{T}_q$.
2) Compute an eigendecomposition

$$\mathbf{T}_q = \mathbf{S}_q \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_q)\mathbf{S}_q^{-1} \tag{30}$$

of $\mathbf{T}_q$, and set

$$\mu = \mathbf{S}_q^T \mathbf{e}_1 \quad \text{and} \quad \nu = \mathbf{S}_q^{-1}\mathbf{e}_1.$$

3) Compute the poles and residues of H_q by setting

$$p_j = 1/\lambda_j \quad \text{and} \quad k_j = \frac{\mathbf{l}^T\mathbf{r} \cdot \mu_j \nu_j}{\lambda_j} \tag{31}$$

for all $j = 1, 2, \cdots, q$ with $\lambda_j \neq 0$. If there is at least one $\lambda_j = 0$, then set

$$k_\infty = \sum_{\substack{j=0 \\ \lambda_j = 0}}^{q} \mathbf{l}^T\mathbf{r} \cdot \mu_j \nu_j.$$

To compute the eigendecomposition (30) of $\mathbf{T}_q$, we use the standard QR algorithm, see, e.g., [15]. This algorithm, as a by-product, first computes the so-called Schur decomposition

$$\mathbf{T}_q = \mathbf{U}_q \mathbf{R}_q \mathbf{U}_q^{-1} \tag{32}$$

of $\mathbf{T}_q$. Here, $\mathbf{U}_q$ is a unitary matrix, and $\mathbf{R}_q$ is an upper triangular matrix whose diagonal elements are just the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_q$ of $\mathbf{T}_q$. Using (32), the Padé approximant H_q can be represented as follows:

$$H_q(s_0 + \sigma) = \mathbf{l}^T\mathbf{r} \cdot (\mathbf{U}_q^T\mathbf{e}_1)^T(\mathbf{I} - \sigma\mathbf{R}_q)^{-1}\mathbf{U}_q^{-1}\mathbf{e}_1. \tag{33}$$

The formula (33) is a useful alternative to the pole/residue representation (28) of H_q. In fact, (33) is more general and more stable than (28), which assumes that $\mathbf{T}_q$ is diagonalizable.

We remark that the PVL Algorithm 2 and AWE require roughly the same amount of computational work. As in AWE, the dominating cost is the computation of the LU factorization

$$\mathbf{G} + s_0\mathbf{C} = \mathbf{L}\mathbf{U}, \tag{34}$$

which needs to be computed only once. Based on (34), the vectors $\mathbf{A}\mathbf{v}_n$ and $\mathbf{A}^T\mathbf{w}_n$ required in step 3) of Algorithm 1 are obtained as follows. First, using forward-backward substitution, we solve

$$\mathbf{L}\mathbf{U}\mathbf{z} = -\mathbf{C}\mathbf{v}_n \quad \text{and} \quad \mathbf{U}^T\mathbf{L}^T\mathbf{y} = -\mathbf{w}_n$$

for $\mathbf{z}$ and $\mathbf{y}$, and then, we set

$$\mathbf{A}\mathbf{v}_n = \mathbf{z} \quad \text{and} \quad \mathbf{A}^T\mathbf{w}_n = \mathbf{C}^T\mathbf{y}.$$

Hence, the PVL algorithm involves $2q$ forward-backward substitutions to generate the qth Padé approximant, which is the same as in AWE.

As a first example, we reran the simulation of the voltage gain of the filter, now with the PVL algorithm instead of AWE. The results of three runs with $q = 2, 8, 28$ are shown in Fig. 2.

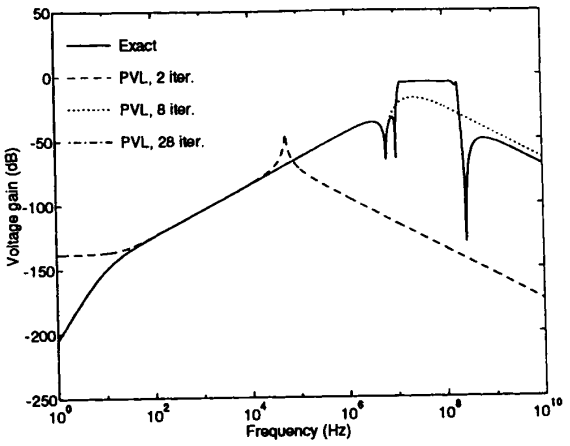

Fig. 2. Results for simulation of voltage gain with PVL.

In contrast to the simulation with AWE (*cf.* Fig. 1), the PVL-generated Padé approximant for $q = 28$ gives a perfect match of the exact voltage gain.

Finally, we remark that the zeros of the reduced-order model H_q can also be computed easily from the Lanczos matrix $\mathbf{T}_q$. In fact, it can be shown that

$$H_q(s_0 + \sigma) = \mathbf{l}^T\mathbf{r}\frac{\det(\mathbf{I} - \sigma\mathbf{T}_q')}{\det(\mathbf{I} - \sigma\mathbf{T}_q)}, \tag{35}$$

where

$$\mathbf{T}_q' := \begin{bmatrix} \alpha_2 & \beta_3 & 0 & \cdots & 0 \\ \rho_3 & \alpha_3 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_q \\ 0 & \cdots & 0 & \rho_q & \alpha_q \end{bmatrix}$$

is the matrix obtained from $\mathbf{T}_q$ by deleting the first row and column. By (35), the zeros of H_q are just the inverses of the eigenvalues of $\mathbf{T}_q'$.

IV. Practical Aspects

In this section, we discuss some practical aspects of the PVL algorithm.

A. A Bound for the Pole-Approximation Error

First, we briefly sketch how the PVL algorithm can be used to obtain bounds for the pole-approximation error.

Recall that, by (31), the poles of the qth Padé approximant H_q are just the inverses of the eigenvalues of the Lanczos matrix $\mathbf{T}_q$. On the other hand, in view of (6), the poles of the exact frequency response H are the inverses of the eigenvalues of $\mathbf{A}$. Therefore, a quality measure for the poles of H_q can be obtained by checking how well the eigenvalues of $\mathbf{T}_q$ approximate the eigenvalues of $\mathbf{A}$.

Let $\lambda_j \in \mathbb{C}$ be any eigenvalue of $\mathbf{T}_q$ with corresponding eigenvector $\mathbf{s}_j \in \mathbb{C}^q$. Thus we have

$$\mathbf{T}_q\mathbf{s}_j = \lambda_j\mathbf{s}_j. \tag{36}$$

Next we set

$$\mathbf{z}_j = \mathbf{V}_q \mathbf{s}_j, \tag{37}$$

where $\mathbf{V}_q$ is the matrix of Lanczos vectors defined in (18). Using (36), (37), and the first relation from (17), we obtain

$$\mathbf{A}\mathbf{z}_j - \lambda_j \mathbf{z}_j = \mathbf{v}_{q+1}\rho_{q+1}s_{qj}, \tag{38}$$

where s_{qj} is the last component of the vector $\mathbf{s}_j$. By taking norms in (38) and since $\|\mathbf{v}_{q+1}\|_2 = 1$, it follows that

$$\frac{\|\mathbf{A}\mathbf{z}_j - \lambda_j \mathbf{z}_j\|_2}{\|\mathbf{A}\|_2 \cdot \|\mathbf{s}_j\|_2} = \frac{\rho_{q+1}|s_{qj}|}{\|\mathbf{A}\|_2 \cdot \|\mathbf{s}_j\|_2}. \tag{39}$$

Furthermore, we can estimate the (usually unknown) norm $\|\mathbf{A}\|_2$ in (39) as follows:

$$\|\mathbf{A}\|_2 \geq \max_{n=1,2,\cdots,q} \{\|\mathbf{A}\mathbf{v}_n\|_2, \|\mathbf{A}^T\mathbf{w}_n\|_2\} = n(\mathbf{A}). \tag{40}$$

Combining (39) and (40) gives

$$\frac{\|\mathbf{A}\mathbf{z}_j - \lambda_j \mathbf{z}_j\|_2}{\|\mathbf{A}\|_2 \cdot \|\mathbf{s}_j\|_2} \leq \frac{\rho_{q+1}|s_{qj}|}{n(\mathbf{A}) \cdot \|\mathbf{s}_j\|_2} = Q_j. \tag{41}$$

Thus the number Q_j represents a measure of how well the pair $(\lambda_j, \mathbf{z}_j)$ approximates an eigenpair of the matrix $\mathbf{A}$, and consequently a quality measure of the approximate pole $1/\lambda_j$ produced by the PVL algorithm. Note that Q_j can easily be computed from the quantities generated by PVL. Finally, we refer the reader to [16] and [17] for a more detailed discussion of eigenvalue bounds of the type (41).

In the PVL algorithm, we use the numbers Q_j as the basis of a pole-screening procedure to check if an approximate pole $1/\lambda_j$ has converged to a true pole of the transfer function H.

B. Choice of the Expansion Point

Another important practical issue is the choice of the expansion point s_0 in (4). The obvious goal here is to select s_0 so that the convergence of the poles generated by the PVL algorithm will be fastest for those poles near the frequency range of interest. Finding such an optimal point s_0 is not practical, since it would require the knowledge of all poles of the transfer function H itself. Instead we developed a simple heuristic for the choice of s_0, based on the assumption that the circuit in stable, i.e., all its poles have negative real parts.

The frequency range of interest is usually of the form $f_{\min} \leq f \leq f_{\max}$ where $f_{\min} < f_{\max}$ and $f_{\max} > 0$, i.e., one is interested in the approximation $H_q(s)$ to $H(s)$ for $s = 2\pi i f, f_{\min} \leq f \leq f_{\max}$. Here $i = \sqrt{-1}$. Roughly speaking, the poles that determine the convergence behavior are the ones closest to the endpoints $2\pi i f_{\max}$ and $2\pi i f_{\min}$, and therefore, the expansion point s_0 should be chosen such that these "extremal" points converge as fast as possible. After making some simplifying assumptions on the convergence behavior of the Lanczos process, this problem can be solved approximately, and the resulting expansion point is as follows:

$$s_0 = (f_{\max} - f_{\min})\pi + (f_{\max} + f_{\min})\pi i. \tag{42}$$

Note that s_0 is a point in the right-half plane, and that its imaginary part is just the midpoint of the complex interval $[2\pi i f_{\min},\ 2\pi i f_{\max}]$.

Often, the frequency range of interest is of the form $0 \leq f \leq f_{\max}$. In view of $H(-2\pi f i) = \overline{H(2\pi f i)}$ and $H_q(-2\pi i f) = \overline{H_q(2\pi i f)}$, this case is equivalent to the frequency range $-f_{\max} \leq f \leq f_{\max}$, i.e., we can formally set $f_{\min} = -f_{\max}$. Thus, from (42), we obtain the expansion point

$$s_0 = 2\pi f_{\max},$$

which we recommend to use for frequency ranges of the form $0 \leq f \leq f_{\max}$.

Finally, we stress that the PVL algorithm is fairly insensitive to the choice of the expansion point s_0, as long as it is chosen as a point in the right-half plane whose distance to the imaginary axis is of the same order as the length of the frequency range of interest.

C. Positive Poles and Reduced-Order Models

The Padé approximation, and hence the PVL algorithm, may produce some approximate poles $1/\lambda_j$ with positive real parts. If the corresponding quality measure Q_j of such a "positive" pole is sufficiently small, then this pole has converged to a true pole of the transfer function, and thus the circuit is unstable. In particular, the PVL algorithm can be used to detect if the design of a circuit is unstable. In Section V, we present an example (the bipolar power transistor) for which the PVL algorithm found converged positive poles after 25 PVL steps.

Even if the circuit itself is stable, it cannot be excluded that the PVL algorithm produces a few positive poles in the early stages of the iteration. If one is only interested in the approximation H_q to the frequency-response function H in some predetermined frequency range and if H_q has converged to H in this frequency range, then such positive poles have no influence on the quality of the approximation, and they need not be eliminated from the pole/residue representation of H_q. Indeed, we found that, if positive poles occurred, either the corresponding residues are negligible, or the positive pole itself is far from the frequency range of interest.

However, if one is interested in using the pole/residue representation of H_q to construct a reduced-order model of the circuit and if one is certain that the circuit itself is stable, then the positive poles in (28) of H_q should be and can be safely deleted. Hence, we simply use

$$\tilde{H}_q(s_0 + \sigma) = k_\infty + \sum_{\substack{j=1 \\ \mathrm{Re}\lambda_j < 0}}^{q} \frac{-\mathbf{l}^T\mathbf{r} \cdot \mu_j \nu_j / \lambda_j}{\sigma - 1/\lambda_j}$$

as the reduced-order model, instead of the full Padé approximation H_q. Indeed, one can show that, if the circuit is stable and if H_q has converged to H in the frequency range of interest, then positive poles either have negligible residues, or they are far from the frequency range of interest. A rigorous proof of this statement is beyond the scope of this paper, and it will be presented elsewhere. In either case, a positive pole does not contribute significantly to H_q, and this justifies our recommendation to simply delete positive poles.

TABLE II
CONVERGED POLES FOR THE PEEC CIRCUIT AFTER 60 PVL ITERATIONS

Poles		Residues		Quality
-7.2630e+08		-2.2501e+06		5.7597e-14
-3.0658e+04	±7.8601e+09	-4.2143e+02	±2.0498e+02	6.7545e-13
-1.7983e+05	±3.1735e+09	-6.2671e+02	∓1.3765e+02	1.5401e-11
-2.5253e+09	±7.6292e+09	9.5132e+05	∓1.6717e+07	3.6946e-11
-1.1024e+06	±1.0577e+10	-1.0766e+03	±9.7822e+03	1.3244e-10
6.4776e-01	±3.9498e+09	-1.0340e-11	∓1.0085e-11	1.0834e-09
-3.3591e+06	±1.1772e+10	8.1455e+03	±1.2229e+04	1.1470e-09
1.2332e-01	±6.3398e+09	-4.7593e-10	∓1.9771e-10	1.0857e-08
-9.4922e+06	±1.3018e+10	3.1855e+04	∓6.6068e+03	1.1875e-08
-7.6129e+02	±1.3908e+10	-2.5802e+00	±1.1633e+02	3.6991e-08
-5.1607e+08	±1.6965e+10	1.8783e+06	∓3.9085e+05	4.2577e-07
-6.1254e+06	±1.7370e+10	-3.6332e+03	±2.7965e+04	1.1356e-06
-5.5966e+07	±1.9140e+10	2.5456e+05	∓1.0583e+05	3.4859e-06
-3.3079e+05	±3.1734e+09	6.5198e-14	±7.8626e-14	6.5304e-06
-2.4118e+04	±9.7937e+09	-2.8771e-10	∓3.5741e-10	9.8450e-06
-9.7108e+07	±2.5611e+10	-5.1745e+05	±2.8947e+05	4.8035e-05
-1.7241e+09	±2.1928e+10	-1.1550e+06	±1.6343e+07	5.6229e-05
-6.1598e+07	±2.8064e+10	-2.6352e+05	±3.0426e+05	2.3356e-04
-2.0255e+10	±3.1453e+11	2.1169e+01	±4.1566e+02	3.9872e-04

V. DISCUSSION AND EXAMPLES

The Padé approximation generates poles that correspond to the dominant poles of the original system and a few poles that do not correspond to poles in the original system, but account for the effects of the remaining original poles. The true poles can be identified using the bound presented in Section IV-A and by comparing the poles obtained at consecutive iterations of the algorithm. The true poles that have converged should not change significantly between iterations.

The first example is a lumped-element equivalent circuit for a three-dimensional electromagnetic problem modeled via PEEC [18] (partial element equivalent circuit). The circuit consists of 2100 capacitors, 172 inductors, and 6990 inductive couplings, resulting in a 306 × 306 fairly dense MNA matrix. The Padé approximation generated by AWE, reproduces the transfer function of the equivalent circuit accurately up to 1 GHz [19].

In [20], Chiprout *et al.*, through the use of multipoint moment matching, obtain a sufficient number of accurate poles and zeros to extend the validity of the approximation up to 5 GHz. However, their method is almost two orders of magnitude more expensive computationally. It involves one complex circuit matrix factorization for each of the 12 complex points used to generate moments, compared to only one real circuit matrix factorization required by AWE.

We applied our PVL algorithm to the same circuit and, after 60 iterations, obtained a reduced-order system with a better match up to 5 GHz than the one obtained from multi-point moment matching (Fig. 3). Moreover, since PVL requires only one real circuit matrix factorization, the cost of the computation is similar to AWE. We also show the result of the PVL algorithm after 30 iterations (Fig. 4).

In Table II, we list the converged poles, their residues, and the corresponding quality measure Q_j from (41), which were obtained after 60 PVL iterations. Note that two of the poles still have a small (compared to their imaginary part) positive real part. However, the residues of both these poles are tiny, and therefore, they can be safely ignored.

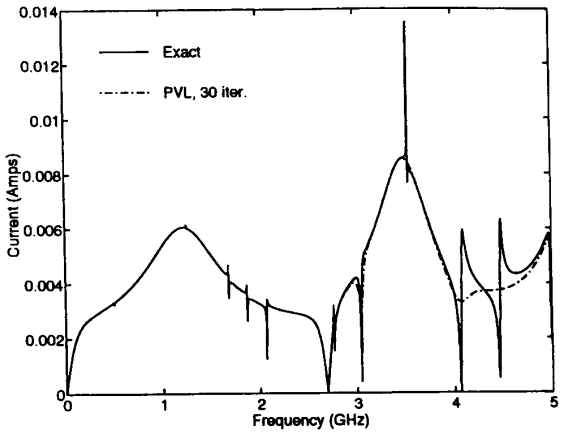

Fig. 3. Results for the PEEC circuit, 30 PVL iterations.

The next example is a very large bipolar power transistor the layout of which covers an area of 3.5 mm × 3.5 mm. It is obvious that interconnect parasitics play an important role in the behavior of the transistor. This transistor was fabricated, but failed to function in the desired regime. The following analysis done with the PVL algorithm explains the cause.

The network was simulated at the desired bias condition of 3 V between the collector and emitter and 0.95 V between base and emitter. The transistor is modeled for simulation purposes as 480 extended Gummel-Poon devices connected in parallel

TABLE III
CONVERGED POLES FOR THE TRANSISTOR AFTER 25 PVL ITERATIONS

Poles		Residues		Quality
1.6786e+09		3.6654e+07		1.9158e-15
-9.5188e+08		-5.6509e+09		1.4963e-08
-2.0437e+09	±5.5215e+09	2.2042e+09	∓7.2306e+08	1.5484e-06
-3.7803e+09		-3.4399e+08		5.4052e-06
5.2577e+08	±8.1671e+09	1.0976e+09	∓8.7793e+08	6.4962e-06
-1.7022e+09		-6.2806e+04		6.4523e-05
-1.0032e+09		-1.4626e+06		1.5880e-04
-1.4595e+09		-8.1321e+03		3.6377e-04
-1.0155e+09		-1.3627e+07		7.0845e-04

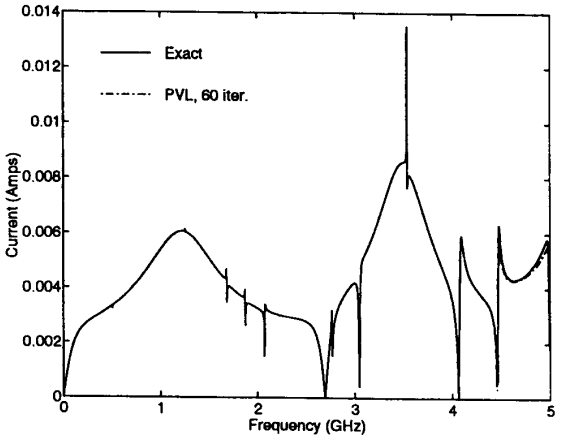

Fig. 4. Results for the PEEC circuit, 60 PVL iterations.

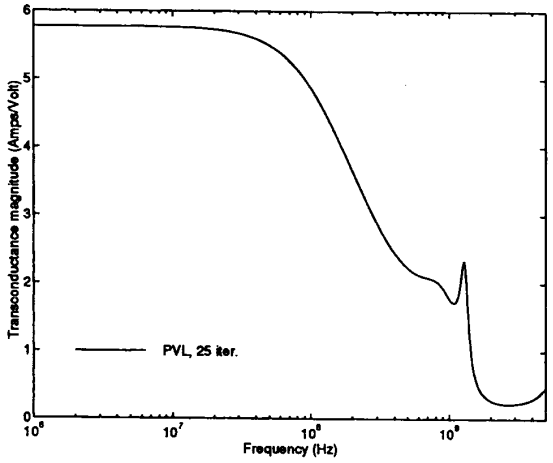

Fig. 5. Results for the transistor—magnitude, 25 PVL iterations.

through a detailed interconnect network extracted from the actual layout. The resulting MNA matrix has a size of 8346 × 8346. The resulting linearized network was analyzed with PVL and several positive half-plane poles were identified, indicating that the operating point was dynamically unstable. Table III lists the converged poles of the network obtained after 25 PVL iterations. In this case, the pole-screening procedure identified both the positive real pole and the complex-conjugate pair with positive real part as genuine. In retrospect, the presence of the instability due to multiple feedback paths through the parasitic interconnect network, was accepted as the most plausible cause of the failure.

Figs. 5 and 6 show the magnitude and phase of the frequency response of the transistor and is identical to that predicted through complex phasor analysis. Phasor analysis, however, besides being orders of magnitude more expensive computationally than PVL, is not capable of diagnosing instability. In fact, no other analysis tool would have been able to detect the instability. If pole analysis with PVL were available during the design of the device a very expensive experiment could have been avoided.

The last example models a complete power grid for a standard cell mixed signal ASIC [21], including some of the substrate contacts and substrate coupling/decoupling, as described in [22]. The model contains 1074 power bus segments,

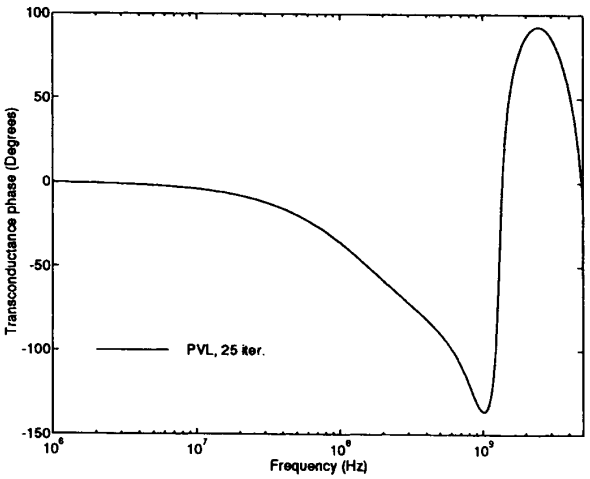

Fig. 6. Results for the transistor—phase, 25 PVL iterations.

36 models for cells, and a coarse, 10 × 10 × 1 substrate grid. The resulting MNA matrix has a size of 1766 × 1766. We are interested to determine the effects of the switching current in cells on the VDD and GND rails. Figs. 7 and 8 show the magnitude and phase of the corresponding transfer function

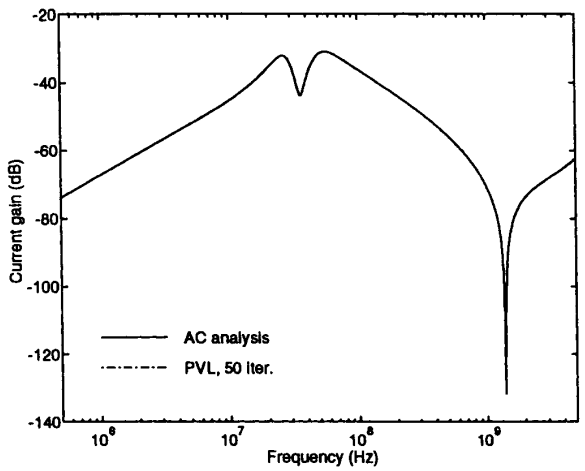

Fig. 7. Results for the mixed-signal circuit—magnitude, 50 PVL iterations.

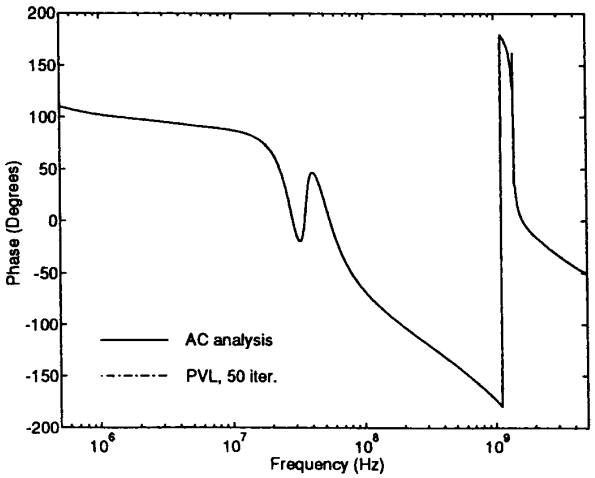

Fig. 8. Results for the mixed-signal circuit—phase, 50 PVL iterations.

produced by the PVL algorithm in 50 iterations compared to the same transfer function produced by an AC sweep. The agreement is excellent.

VI. Concluding Remarks

In this paper, we have introduced the PVL algorithm for stably computing the Padé approximation of Laplace-domain transfer functions of large linear networks via a Lanczos process. This paper argues that, due to its robustness and efficiency, PVL should become the algorithm of choice for the analysis of large linear(ized) electrical circuits. The advantages of the PVL algorithm are not limited to superior accuracy and efficiency, but include new capabilities difficult to implement with existing methods.

The PVL algorithm can also be used for sensitivity computations. In fact, in [23], we extend the PVL algorithm to compute sensitivities of network transfer functions, their poles, and zeros, with respect to arbitrary circuit parameters, with minimal additional computational cost.

The PVL algorithm proposed in this paper is for single-input single-output systems of the form (1). It is natural to ask whether PVL can be generalized to compute matrix Padé approximants to the matrix-valued transfer functions that describe multiple-input multiple-output systems, i.e., systems of the type (1) where the vectors b and l are replaced by matrices. This is indeed possible, and we have developed with the MPVL algorithm [24] an extension of PVL to general multiple-input multiple-output systems. The MPVL algorithm computes a matrix Padé approximation to the matrix-valued transfer function of the multiple-input multiple-output system, using a novel Lanczos-type algorithm for multiple starting vectors [25].

Acknowledgment

We would like to thank R. Melville and S. Moinian from Bell Labs., R. Rutenbar from CMU, and H. Heeb and B. Stanisic from IBM for providing us with interesting circuit examples. We also acknowledge the original AWE developers, L. Pillage, X. Huang, and R. Rohrer, who demonstrated the usefulness of the Padé approximation in the analysis of large linear circuits.

References

[1] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352–366, Apr. 1990.

[2] V. Raghavan, R. A. Rohrer, L. T. Pillage, J. Y. Lee, J. E. Bracken, and M. M. Alaybeyi, "AWE-inspired," *Proc. IEEE Custom Integrated Circuits Conf.*, May 1993.

[3] X. Huang, "Padé approximation of linear(ized) circuit responses," Ph.D. dissertation, Dept. Electrical and Computer Eng., Carnegie Mellon Univ., Pittsburgh, PA, Nov. 1990.

[4] G. A. Baker, Jr. and P. Graves-Morris, *Padé Approximants, Part I: Basic Theory*. Reading, MA: Addison-Wesley, 1981.

[5] V. Raghavan, J. E. Bracken, and R. A. Rohrer, "AWESpice: A general tool for the accurate and efficient simulation of interconnect problems," *Proc. 29th ACM/IEEE Design Automation Conf.*, June 1992.

[6] E. Chiprout and M. Nakhla, "Generalized moment-matching methods for transient analysis of interconnect networks," *Proc. 29th ACM/IEEE Design Automation Conf.*, June 1992.

[7] M. M. Alaybeyi, J. Y. Lee, and R. A. Rohrer, "Numerical integration algorithms and asymptotic waveform evaluation (AWE)," *Tech. Dig. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1992.

[8] C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *J. Res. Nat. Bur. Standards*, vol. 45, pp. 255–282, 1950.

[9] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New York: Van Nostrand Reinhold, 1983.

[10] E. Chiprout and M. S. Nakhla, *Asymptotic Waveform Evaluation and Moment Matching for Interconnect Analysis*. Norwell, MA: Kluwer Academic, 1994.

[11] *MATLAB User's Guide*, The Math Works, Inc., Natick, MA, 1992.

[12] W. B. Gragg, "Matrix interpretations and applications of the continued fraction algorithm," *Rocky Mountain J. Math.*, vol. 4, pp. 213–225, 1974.

[13] R. W. Freund, "The look-ahead Lanczos process for large non-symmetric matrices and related algorithms," in *Linear Algebra for Large Scale and Real-Time Applications*. Dordrecht, The Netherlands: Kluwer Academic, 1993.

[14] 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.*, vol. 14, pp. 137–158, Jan. 1993.

[15] G. H. Golub and C. F. Van Loan, *Matrix Computations*, second ed. Baltimore, MD: The Johns Hopkins University Press, 1989.

[16] W. Kahan, B. N. Parlett, and E. Jiang, "Residual bounds on approximate eigensystems of nonnormal matrices," *SIAM J. Numer. Anal.*, vol. 19, pp. 470–484, June 1982.
[17] J. Cullum and R. A. Willoughby, "A practical procedure for computing eigenvalues of large sparse nonsymmetric matrices," in *Large Scale Eigenvalue Problems*. Amsterdam, The Netherlands: North-Holland, 1986.
[18] A. E. Ruehli, "Equivalent circuit models for three-dimensional multiconductor systems," *IEEE Trans. Microwave Theory Tech.*, vol. 22, pp. 216–221, Mar. 1974.
[19] H. Heeb, A. E. Ruehli, J. E. Bracken, and R. A. Rohrer, "Three-dimensional circuit oriented electromagnetic modeling for VLSI interconnects," *Proc. Int. Conf. Computer Design: VLSI in Computers & Processors*, Oct. 1992.
[20] E. Chiprout, H. Heeb, M. S. Nakhla, and A. E. Ruehli, "Simulating 3-D retarded interconnect models using complex frequency hopping (CFH)," *Tech. Dig. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1993.
[21] T. J. Schmerbeck, R. A. Richetta, and L. D. Smith, "A 27 MHz mixed analog/digital magnetic recording channel DSP using partial response signalling with maximum likelihood detection," *Tech. Dig. IEEE Int. Solid-State Circuit Conf.*, Feb. 1991.
[22] B. R. Stanisic, R. A. Rutenbar, and L. R. Carley, "Mixed-signal noise-decoupling via simultaneous power distribution design and cell customization in rail," *Proc. IEEE Custom Integrated Circuits Conf.*, 1994.
[23] R. W. Freund and P. Feldmann, "Efficient small-signal circuit analysis and sensitivity computations with the PVL algorithm," *Numerical Analysis Manuscript 94-09*, AT&T Bell Labs., Murray Hill, NJ, Aug. 1994.
[24] P. Feldmann and R. W. Freund, "Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm," *Numerical Analysis Manuscript*, AT&T Bell Labs., Oct. 1994.
[25] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernández, "A Lanczos-type algorithm for multiple starting vectors," *Numerical Analysis Manuscript*, AT&T Bell Labs., Murray Hill, NJ, Apr. 1995.

Peter Feldmann (S'87–M'91) was born in Timişoara, Romania. He received the B.Sc. degree (*summa cum laude*) in computer engineering in 1983 and the M.Sc. degree in electrical engineering in 1987, both from the Technion, Israel, and the Ph.D. degree in 1991 from Carnegie Mellon University.

From 1985 through 1987 he worked for Zoran Microelectronics in Haifa, Israel, on the design of digital signal processors. Currently, he is a member of technical staff in the Computing Systems Technology Research Department of AT&T Bell Laboratories, Murray Hill, NJ. His research interests include CAD for VLSI circuits, more specifically, circuit-level simulation, optimization, and statistical design.

Roland W. Freund received the Ph.D. degree (*summa cum laude*) in mathematics from the University of Würzburg, Germany, in 1983.

He held positions at the University of Würzburg and at the Research Institute for Advanced Computer Science at NASA Ames Research Center. He also spent one year with the Computer Science Department of Stanford University. In 1992 he joined AT&T Bell Laboratories, Murray Hill, NJ, where he is a member of technical staff in the Scientific Computing Research Department. His research interests are in scientific computing, numerical linear algebra, large-scale optimization, and algorithms for circuit simulation.

Dr. Freund won the SIAM SIAG Best Linear Algebra Paper Prize in 1994 (with N. Nachtigal), and in 1989, he was awarded the Heinz-Maier-Leibnitz Prize for outstanding publications in Applied Mathematics.