Efficient linear circuit analysis by Padé approximation via the Lanczos process

Peter Feldmann
AT&T Bell Laboratories
Room 2C–217
600 Mountain Avenue
Murray Hill, NJ 07974–0636, USA

Roland W. Freund
AT&T Bell Laboratories
Room 2C–420
600 Mountain Avenue
Murray Hill, NJ 07974–0636, USA

Abstract

This paper describes a highly efficient and numerically stable algorithm for the iterative computation of dominant poles and zeros of large linear networks. The algorithm is based on a new implementation of the Padé approximation via the Lanczos process. This implementation has considerably superior numerical properties, while maintaining the same computational efficiency as its predecessors. In addition, the algorithm provides a bound on the pole approximation error.

1 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 the last few years, the **A**symptotic **W**aveform **E**valuation algorithm (AWE) [1, 2, 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].

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 [2, 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 2, we demonstrate the numerical limitations of algorithms that compute the Padé approximation from time-domain moments of the impulse response, as it is done in AWE. In Section 3, we derive the new PVL algorithm and discuss some of its properties. In Section 4, we present results of numerical experiments with PVL for a variety of examples. Section 5 contains the conclusion.

2 Limitations of current algorithms

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

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

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

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

we can rewrite (3) as follows:

$$\begin{aligned} H(s_0 + \sigma) &= \mathbf{l}^T\left(\sigma\mathbf{C} + \mathbf{G} + s_0\mathbf{C}\right)^{-1}\mathbf{b} \\ &= \mathbf{l}^T\left(\mathbf{I} - \sigma\mathbf{A}\right)^{-1}\mathbf{r}. \end{aligned} \tag{5}$$

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

$$H(s_0 + \sigma) = \underbrace{\mathbf{l}^T\mathbf{S}}_{=\mathbf{f}^T}\left(\mathbf{I} - \sigma\mathbf{\Lambda}\right)^{-1}\underbrace{\mathbf{S}^{-1}\mathbf{r}}_{=\mathbf{g}}, \tag{6}$$

where $\mathbf{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}$, $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$ 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 (6), we get

$$H(s_0 + \sigma) = \sum_{j=1}^{N} \frac{f_j g_j}{1 - \sigma\lambda_j}, \tag{7}$$

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 impulse response is through an approximation.

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

$$H_{p,q}(s_0 + \sigma) = \frac{b_p \sigma^p + \cdots + b_1 \sigma + b_0}{a_q \sigma^q + \cdots + a_1 \sigma + 1} \tag{8}$$

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

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

The coefficients $a_1, \ldots, a_q, b_0, b_1, \ldots, b_p$ of the Padé approximation (8) are uniquely determined by the first $p + q$ Taylor coefficients of the impulse response. The roots of the denominator and numerator polynomials in (8) represent the dominant poles and zeros of the system, respectively.

In view of (7), 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 impulse-response approximations, it is very natural to choose $p = q - 1$ in (8), so that the Padé approximation is of the same form as the original impulse 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 impulse response H. By using a partial-fraction decomposition, we can write H_q in the form

$$H_q(s_0 + \sigma) = k_\infty + \sum_{j=1}^{q'} \frac{k_j}{\sigma - p_j}, \tag{9}$$

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:

$$H(s_0 + \sigma) = \mathbf{l}^T \left(\mathbf{I} + \sigma \mathbf{A} + \sigma^2 \mathbf{A}^2 + \cdots \right) \mathbf{r} = \sum_{k=0}^{\infty} m_k \sigma^k, \tag{10}$$

where

$$m_k = \mathbf{l}^T \mathbf{A}^k \mathbf{r}, \quad k = 0, 1 \ldots . \tag{11}$$

Note that in the case when the expansion point in (10) is chosen as the origin, i.e., $s_0 = 0$, the coefficients m_k, $k = 0, 1, \ldots$, are, up to a constant factor, the time-domain moments of the circuit response. Indeed, we have

$$
\begin{aligned}
H(s) &= \int_0^\infty h(t)e^{-st}\,dt \\
&= \int_0^\infty h(t)\left[1 - st + \frac{1}{2}s^2t^2 - \cdots\right] dt \\
&= \int_0^\infty h(t)\,dt - s\int_0^\infty th(t)\,dt + s^2\frac{1}{2}\int_0^\infty t^2h(t)\,dt - \cdots .
\end{aligned} \tag{12}
$$

Because of this analogy, we will always refer to the Taylor coefficients (11) as the *moments* of the impulse-response function $H(s_0 + \sigma)$.

2.2 Implementation of the Padé approximation in AWE

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

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

$$(\mathbf{G} + s_0\mathbf{C})\,\mathbf{u}_k = \mathbf{C}\mathbf{u}_{k-1}, \quad k = 1, 2, \ldots, 2q-1, \tag{13}$$

with the initial vector

$$\mathbf{u}_0 = (\mathbf{G} + s_0\mathbf{C})^{-1}\,\mathbf{b}. \tag{14}$$

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 $m_k = \mathbf{l}^T\mathbf{u}_k$, $k = 0, 1, 2, \ldots, 2q-1$.

As the next step, AWE computes the coefficients of the denominator polynomial of the representation (8) of H_q via solution of the linear system

$$
\begin{bmatrix}
m_0 & m_1 & m_2 & \cdots & m_{q-1} \\
m_1 & \iddots & & \iddots & \vdots \\
m_2 & & \iddots & & \vdots \\
\vdots & \iddots & & & m_{2q-3} \\
m_{q-1} & \cdots & \cdots & m_{2q-3} & m_{2q-2}
\end{bmatrix}
\cdot
\begin{bmatrix}
a_q \\ a_{q-1} \\ \vdots \\ a_2 \\ a_1
\end{bmatrix}
= -
\begin{bmatrix}
m_q \\ m_{q+1} \\ \vdots \\ m_{2q-2} \\ m_{2q-1}
\end{bmatrix}. \tag{15}
$$

The poles p_j of H_q in (9) are then obtained as the roots of the equation

$$a_q\sigma^q + a_{q-1}\sigma^{q-1} + \cdots + a_1\sigma + 1 = 0. \tag{16}$$

Finally, the constant k_∞ and the residues k_j in (9) are computed by solving another linear system of order q, see, e.g., [10] for details.

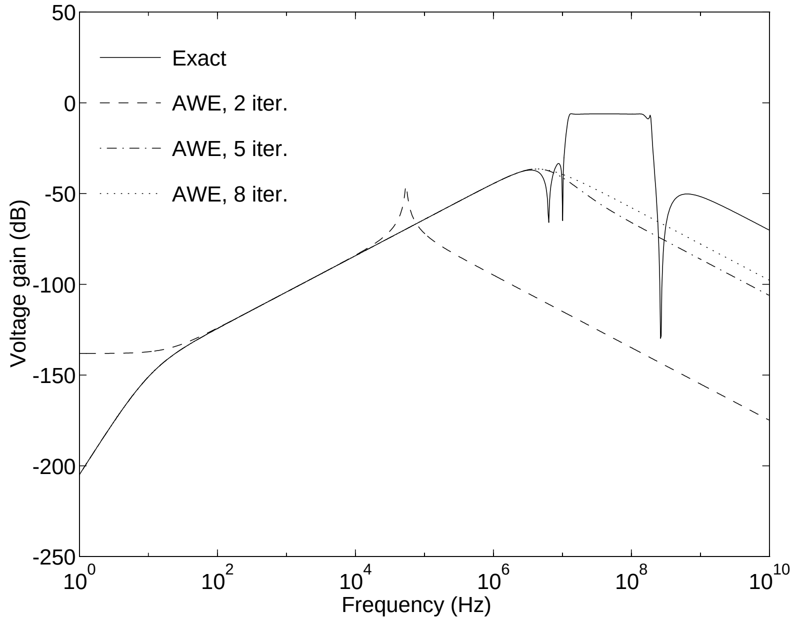

Figure 1: Results for simulation of voltage gain with AWE

As q is increased, one would expect more and more accurate approximations H_q of the exact impulse 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. Figure 1 illustrates this behavior. Here, we tried to simulate the voltage gain of a filter with our own implementation of AWE written in MATLAB™ [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 (15). Let us denote by

$$\mathbf{M}_q = [\, m_{j+k-2} \,]_{j,k=1,2,\ldots,q} \tag{17}$$

the coefficient matrix of (15); we will refer to $\mathbf{M}_q$ as the *moment matrix*. The condition number, $\mathrm{cond}(\mathbf{M}_q)$, of the matrix $\mathbf{M}_q$ is a measure for how round-off error affects the accuracy of the numerically computed solution of (15). Each increase of $\mathrm{cond}(\mathbf{M}_q)$ 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 $\mathrm{cond}(\mathbf{M}_q) = \mathcal{O}(10^{16})$. In the first column of Table 1, we list $\mathrm{cond}(\mathbf{M}_q)$

q	cond($\mathbf{M}_q$)	cond($\mathbf{M}_q^{(1)}$)	cond($\mathbf{M}_q^{(2)}$)	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

Table 1: Condition numbers of moment matrices for simulation of voltage gain with AWE

for the moment matrices corresponding to the simulation of the voltage gain of a filter. Clearly, the moment matrices are extremely ill-conditioned.

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

$$\xi_1 = \frac{1}{\|\mathbf{A}\|_2}, \quad \xi_2 = \frac{|m_0|}{|m_1|}, \quad \text{and} \quad \xi_3 = \left(\frac{|m_0|}{|m_{2q-1}|}\right)^{1/(2q-1)}. \tag{18}$$

The first choice in (18) corresponds to scaling the matrix $\mathbf{A}$ in (11) to have Euclidean norm 1. The second choice, proposed in [2], 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

$$\mathbf{M}_q^{(l)} = [\, \xi_l^{j+k-2} m_{j+k-2} \,]_{j,k=1,2,\ldots,q}\,, \quad l = 1, 2, 3, \tag{19}$$

for all three strategies (18) still reach the critical value 10^{16} for fairly small values of q. In Table 1, we also list the condition numbers of the scaled moment matrices (19) for the case of the simulation of the voltage gain of a filter (cf. Figure 1). In this case, the third scaling strategy in (18) yields the smallest condition numbers. However, we still have 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 Figure 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 (13) 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 (7), 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.

3 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.

3.1 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 $n = 0, 1, \ldots, q$ do:

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

2) Set
$$\alpha_n = \frac{\mathbf{w}_n^T \mathbf{A} \mathbf{v}_n}{\delta_n}, \quad \beta_n = \frac{\delta_n}{\delta_{n-1}} \eta_n, \quad \gamma_n = \frac{\delta_n}{\delta_{n-1}} \rho_n. \tag{20}$$

3) Set
$$\begin{aligned} \mathbf{v} &= \mathbf{A}\mathbf{v}_n - \mathbf{v}_n \alpha_n - \mathbf{v}_{n-1} \beta_n, \\ \mathbf{w} &= \mathbf{A}^T \mathbf{w}_n - \mathbf{w}_n \alpha_n - \mathbf{w}_{n-1} \gamma_n. \end{aligned} \tag{21}$$

4) Set $\rho_{n+1} = \|\mathbf{v}\|_2$, $\eta_{n+1} = \|\mathbf{w}\|_2$, and
$$\mathbf{v}_{n+1} = \frac{\mathbf{v}}{\rho_{n+1}}, \quad \mathbf{w}_{n+1} = \frac{\mathbf{w}}{\eta_{n+1}}. \tag{22}$$

We remark that in Algorithm 1 a breakdown, triggered by division by 0, will occur if one encounters $\delta_n = 0$ in (20). Furthermore, division by a nonzero yet small number $\delta_n \approx 0$ in (20) 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, 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:

$$\mathbf{w}_j^T \mathbf{v}_k = \begin{cases} \delta_j, & \text{if } j = k, \\ 0, & \text{if } j \neq k, \end{cases} \qquad \text{for all} \quad j, k = 1, 2, \ldots, q+1. \tag{23}$$

- The tridiagonal matrices

$$\mathbf{T}_q = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ \rho_2 & \alpha_2 & \beta_3 & \ddots & \vdots \\ 0 & \rho_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_q \\ 0 & \cdots & 0 & \rho_q & \alpha_q \end{bmatrix}, \quad \tilde{\mathbf{T}}_q = \begin{bmatrix} \alpha_1 & \gamma_2 & 0 & \cdots & 0 \\ \eta_2 & \alpha_2 & \gamma_3 & \ddots & \vdots \\ 0 & \eta_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \gamma_q \\ 0 & \cdots & 0 & \eta_q & \alpha_q \end{bmatrix}, \tag{24}$$

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

$$\begin{aligned} \mathbf{A}\mathbf{V}_q &= \mathbf{V}_q \mathbf{T}_q + [\,0 \quad \cdots \quad 0 \quad \mathbf{v}_{q+1}\,]\, \rho_{q+1}, \\ \mathbf{A}^T \mathbf{W}_q &= \mathbf{W}_q \tilde{\mathbf{T}}_q + [\,0 \quad \cdots \quad 0 \quad \mathbf{w}_{q+1}\,]\, \eta_{q+1}, \end{aligned} \tag{25}$$

  where

$$\mathbf{V}_q = [\,\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_q\,] \quad \text{and} \quad \mathbf{W}_q = [\,\mathbf{w}_1 \quad \mathbf{w}_2 \quad \cdots \quad \mathbf{w}_q\,] \tag{26}$$

  are matrices whose columns are just the Lanczos vectors.

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

$$\tilde{\mathbf{T}}_k^T = \mathbf{D}_q \mathbf{T}_q \mathbf{D}_q^{-1}, \tag{27}$$

  where

$$\mathbf{D}_q = \mathbf{W}_q^T \mathbf{V}_q = \operatorname{diag}(\delta_1, \delta_2, \ldots, \delta_q). \tag{28}$$

3.2 The Padé connection

Next, we demonstrate the connection of the Lanczos process to Padé approximation. Using the first relation in (25) and the fact that $\mathbf{T}_q$ is tridiagonal, one can show that

$$\mathbf{A}^j \mathbf{r} = \rho_1 \mathbf{A}^j \mathbf{v}_1 = \rho_1 \mathbf{A}^j \mathbf{V}_q \mathbf{e}_1 = \rho_1 \mathbf{V}_q \mathbf{T}_q^j \mathbf{e}_1, \quad j = 0, 1, \ldots, q-1, \tag{29}$$

where $\mathbf{e}_1 = [1 \quad 0 \quad \cdots \quad 0]^T \in \mathbb{R}^q$ is the first unit vector in $\mathbb{R}^q$. Similarly, from the second relation in (25), one can deduce that

$$\mathbf{l}^T \mathbf{A}^j = \eta_1 \mathbf{e}_1^T \left(\tilde{\mathbf{T}}_q^T\right)^k \mathbf{W}_q^T = \eta_1 \delta_1 \mathbf{e}_1^T \mathbf{T}_q^k \mathbf{D}_q^{-1} \mathbf{W}_q^T, \quad j = 0, 1, \ldots, q-1. \tag{30}$$

We note that the second equation in (30) follows from (27).

On the other hand, by (11), each moment m_k can be written as follows:

$$m_k = \mathbf{l}^T \mathbf{A}^k \mathbf{r} = \left(\mathbf{l}^T \mathbf{A}^{k'}\right) \left(\mathbf{A}^{k''} \mathbf{r}\right), \tag{31}$$

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 (29)–(31) it follows that

$$m_k = \left(\eta_1 \delta_1 \mathbf{e}_1^T \mathbf{T}_q^{k'} \mathbf{D}_q^{-1} \mathbf{W}_q^T\right) \left(\rho_1 \mathbf{V}_q \mathbf{T}_q^{k''} \mathbf{e}_1\right) = \eta_1 \rho_1 \delta_1 \mathbf{e}_1^T \mathbf{T}_q^{k'} \mathbf{D}_q^{-1} \underbrace{\mathbf{W}_q^T \mathbf{V}_q}_{= \mathbf{D}_q} \mathbf{T}_q^{k''} \mathbf{e}_1. \tag{32}$$

Note that $\eta_1 \rho_1 \delta_1 = \mathbf{l}^T \mathbf{r}$, and thus, from (32), we get

$$m_k = (\mathbf{l}^T \mathbf{r}) \cdot \left(\mathbf{e}_1^T \mathbf{T}_q^k \mathbf{e}_1\right) \quad \text{for all} \quad k = 0, 1, \ldots, 2q-2. \tag{33}$$

Furthermore, it can be shown that the relation (33) also holds for $k = 2q-1$. Thus, by (33), we have

$$\begin{aligned} \mathbf{l}^T \mathbf{r} \cdot \mathbf{e}_1^T \left(\mathbf{I} - \sigma \mathbf{T}_q\right)^{-1} \mathbf{e}_1 &= \mathbf{l}^T \mathbf{r} \sum_{k=0}^{\infty} \mathbf{e}_1^T \mathbf{T}_q^k \mathbf{e}_1 \, \sigma^k \\ &= \sum_{k=0}^{2q-1} m_k \, \sigma^k + \mathcal{O}(\sigma^{2q}), \end{aligned} \tag{34}$$

and consequently,

$$H_q(s_0 + \sigma) = \mathbf{l}^T \mathbf{r} \cdot \mathbf{e}_1^T \left(\mathbf{I} - \sigma \mathbf{T}_q\right)^{-1} \mathbf{e}_1 \tag{35}$$

is just the qth Padé approximant of H.

In analogy to the representation (7) of the exact impulse response H, we can rewrite the expression (35) of H_q in terms of the eigendecomposition $\mathbf{T}_q = \mathbf{S}_q \mathbf{\Lambda}_q \mathbf{S}_q^{-1}$ of the Lanczos matrix $\mathbf{T}_q$:

$$H_q(s_0 + \sigma) = \mathbf{l}^T \mathbf{r} \cdot \underbrace{\mathbf{e}_1^T \mathbf{S}_q^T}_{= \mu^T} \left(\mathbf{I} - \sigma \mathbf{\Lambda}_q\right)^{-1} \underbrace{\mathbf{S}_q^{-1} \mathbf{e}_1}_{= \nu} = \sum_{j=1}^{q} \frac{\mathbf{l}^T \mathbf{r} \cdot \mu_j \nu_j}{1 - \sigma \lambda_j}. \tag{36}$$

Here, $\mathbf{\Lambda}_q = \operatorname{diag}(\lambda_1, \lambda_1, \ldots, \lambda_q)$ contains the eigenvalues of $\mathbf{T}_q$, and μ_j and ν_j are the components of the vectors μ and ν. Finally, from (36), we immediately the pole/residue representation of the Padé approximant:

$$H_q(s_0 + \sigma) = k_\infty + \sum_{\substack{j=1 \\ \lambda_j \neq 0}}^{q} \frac{-\mathbf{l}^T \mathbf{r} \cdot \mu_j \nu_j / \lambda_j}{\sigma - 1/\lambda_j} \tag{37}$$

Note that the term h_∞ in (37) may result if one of the eigenvalues of $\mathbf{T}_q$ is zero.

3.3 Computing poles, residues, and zeros

The derivation in the previous section shows that the pole/residue representation (37) 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

$$\mathbf{T}_q = \mathbf{S}_q \operatorname{diag}(\lambda_1, \lambda_1, \ldots, \lambda_q) \mathbf{S}_q^{-1} \tag{38}$$

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

$$\mu = \mathbf{S}_q^T \mathbf{e}_1 \quad \text{and} \quad \nu = \mathbf{S}_q^{-1} \mathbf{e}_1. \tag{39}$$

3) Compute the poles and residues of H_q by setting

$$\begin{aligned} p_j &= 1/\lambda_j \quad \text{and} \quad k_j = \frac{\mathbf{l}^T \mathbf{r} \cdot \mu_j \nu_j}{\lambda_j} \quad \text{for all} \quad j = 1, 2, \ldots, q \quad \text{with} \quad \lambda_j \neq 0, \\ k_\infty &= \sum_{\substack{j=0 \\ \lambda_j = 0}}^{q} \mathbf{l}^T \mathbf{r} \cdot \mu_j \nu_j. \end{aligned} \tag{40}$$

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

$$\mathbf{G} + s_0 \mathbf{C} = \mathbf{L}\mathbf{U}, \tag{41}$$

which needs to be computed only once. Based on (41), 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

$$\mathbf{L}\mathbf{U}\mathbf{z} = -\mathbf{C}\mathbf{v}_n \quad \text{and} \quad \mathbf{U}^T \mathbf{L}^T \mathbf{y} = -\mathbf{w}_n \tag{42}$$

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

$$\mathbf{A}\mathbf{v}_n = \mathbf{z} \quad \text{and} \quad \mathbf{A}^T \mathbf{w}_n = \mathbf{C}^T \mathbf{y}. \tag{43}$$

Hence, the PVL algorithm involve $2q$ 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 Figure 2. In contrast to the simulation with AWE (cf. Figure 1), the PVL-generated Padé approximant for $q = 28$ gives a perfect match of the exact voltage gain.

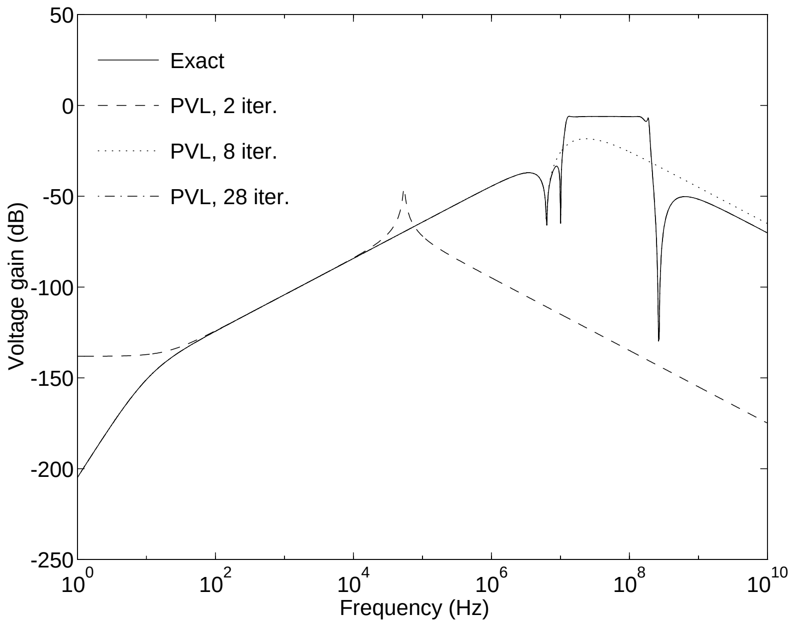

Figure 2: Results for simulation of voltage gain with PVL

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

$$H_q(s_0+\sigma) = \mathbf{l}^T\mathbf{r}\frac{\det\left(\mathbf{I}-\sigma\mathbf{T}'_q\right)}{\det\left(\mathbf{I}-\sigma\mathbf{T}_q\right)}, \tag{44}$$

where

$$\mathbf{T}'_q := \begin{bmatrix} \alpha_2 & \beta_3 & 0 & \cdots & 0 \\ \rho_3 & \alpha_3 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_q \\ 0 & \cdots & 0 & \rho_q & \alpha_q \end{bmatrix} \tag{45}$$

is the matrix obtained from $\mathbf{T}_q$ by deleting the first row and column. By (44), the zeros of H_q are just the inverses of the eigenvalues of $\mathbf{T}'_q$.

3.4 A bound for the pole approximation error

In this section, we briefly sketch how the PVL algorithm can be used to obtain bounds for the pole approximation error.

Recall that, by (40), 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 (7), the poles of the exact impulse 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

$$\mathbf{T}_q \mathbf{s}_j = \lambda_j \mathbf{s}_j. \tag{46}$$

Next we set

$$\mathbf{z}_j = \mathbf{V}_q \mathbf{s}_j, \tag{47}$$

where $\mathbf{V}_q$ is the matrix of Lanczos vectors defined in (26). Using (46), (47), and the first relation from (25), we obtain

$$\mathbf{A}\mathbf{z}_j - \lambda_j \mathbf{z}_j = \mathbf{v}_{q+1} \rho_{q+1} s_{qj}, \tag{48}$$

where s_{qj} is the last component of the vector $\mathbf{s}_j$. By taking norms in (48) and since $\|\mathbf{v}_{q+1}\|_2 = 1$, it follows that

$$\frac{\|\mathbf{A}\mathbf{z}_j - \lambda_j \mathbf{z}_j\|_2}{\|\mathbf{A}\|_2 \cdot \|\mathbf{s}_j\|_2} = \frac{\rho_{q+1}|s_{qj}|}{\|\mathbf{A}\|_2 \cdot \|\mathbf{s}_j\|_2}. \tag{49}$$

Furthermore, we can estimate the (usually unknown) norm $\|\mathbf{A}\|_2$ in (49) as follows:

$$\|\mathbf{A}\|_2 \geq \max_{n=1,2,\ldots,q} \{\|\mathbf{A}\mathbf{v}_n\|_2, \|\mathbf{A}^T\mathbf{w}_n\|_2\} = n(\mathbf{A}). \tag{50}$$

Combining (49) and (50) gives

$$\frac{\|\mathbf{A}\mathbf{z}_j - \lambda_j \mathbf{z}_j\|_2}{\|\mathbf{A}\|_2 \cdot \|\mathbf{s}_j\|_2} \leq \frac{\rho_{q+1}|s_{qj}|}{n(\mathbf{A}) \cdot \|\mathbf{s}_j\|_2} = Q_j. \tag{51}$$

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 [15] for a more detailed discussion of eigenvalue bounds of the type (51).

4 Discussion and examples

The Padé approximation generates poles that correspond to the dominant poles of the original system and a few poles which do not have correspondence 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 the previous section 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 three-dimensional electromagnetic problem modeled via PEEC [16] (partial element equivalent circuit). The circuit

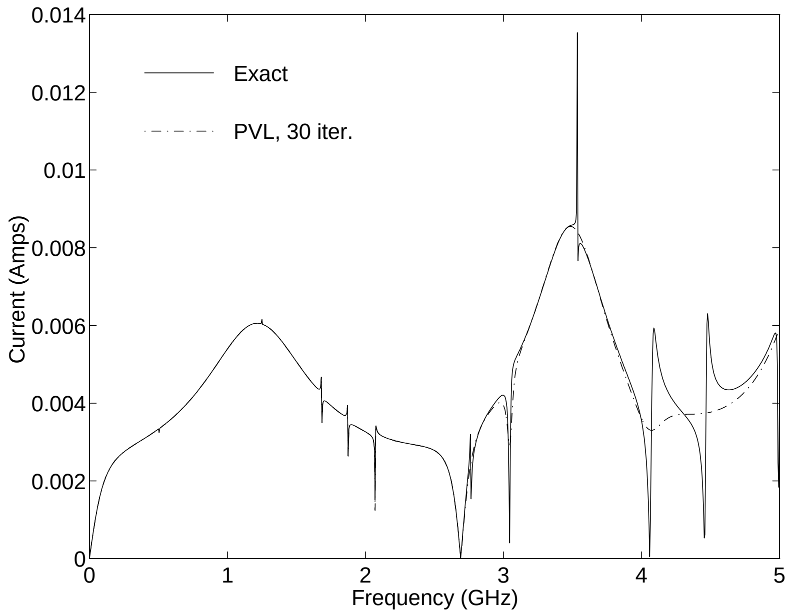

Figure 3: Results for the PEEC circuit, 30 PVL iterations

consists of 2100 capacitors, 172 inductors and 6990 inductive couplings. The Padé approximation generated by AWE, reproduces the transfer-function of the equivalent circuit accurately up to 1GHz [17].

In [6], Chiprout and Nakhla, through the use of multi-point moment matching, obtain a sufficient number of accurate poles and zeros to extend the validity of the approximation up to 5GHz. 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 5GHz than the one obtained from multi-point moment matching (Figure 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 (Figure 4).

The next example is a linear network of 594 elements representing the linearization of a 2x switched-capacitor amplifier with a wiring error that is causing positive feedback. We ran our MATLAB™ implementation of AWE on this circuit with various scaling strategies. Figure 5 shows the phase of the transfer function for three cases. The first graph shows that after 4 iterations the phase of the reduced model is correct up to 100 MHz. By this time, the condition of the moment matrix (17) is so bad that increasing the order of the approximation to 8 does not improve the result significantly, as shown by the second graph.

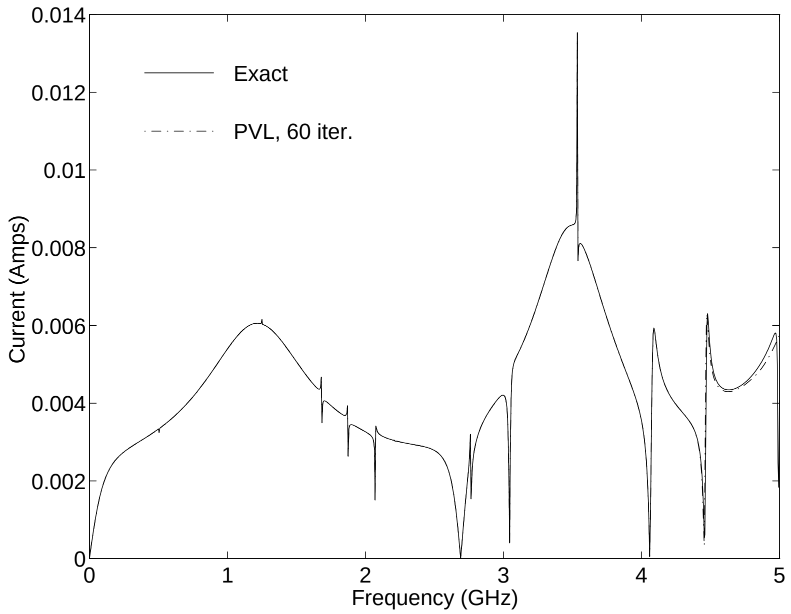

Figure 4: Results for the PEEC circuit, 60 PVL iterations

This result is obtained using the best scaling strategy among (18), which in this example turned out to be the third choice ξ_3. The third graph shows that with a slightly worse scaling strategy, namely the second choice ξ_2 in (18), the results become meaningless. In contrast, Figure 6 indicates that PVL does not suffer from numerical degradation. The results after 4 iterations are similar to AWE, after 8 iterations the matching of the frequency response is extended to 500MHz and for 18 iterations the match is perfect up to 5GHz.

The last example models a complete power grid for a standard cell mixed signal ASIC [18], including some of the substrate contacts and substrate coupling/decoupling, as described in [19]. The model contains 1074 power bus segments, 36 models for cells and a coarse, $10\times10\times1$ substrate grid. We are interested to determine the effects of the switching current in cells on the VDD and GND rails. Figure 7 and 8 show the magnitude and phase of the corresponding transfer function produced by the PVL algorithm in 50 iterations compared to the same transfer function produced by an AC sweep. The agreement is excellent.

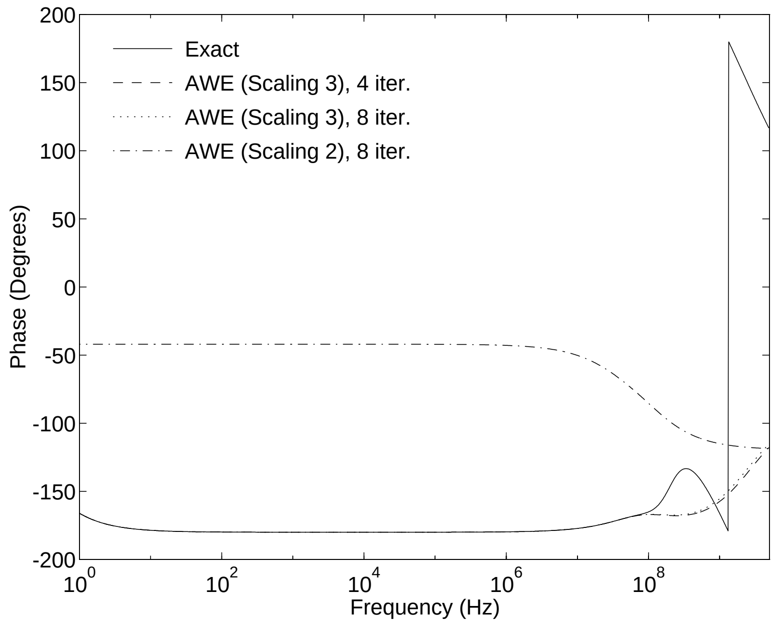

Figure 5: Results for the 2x switched-cap amplifier phase with AWE

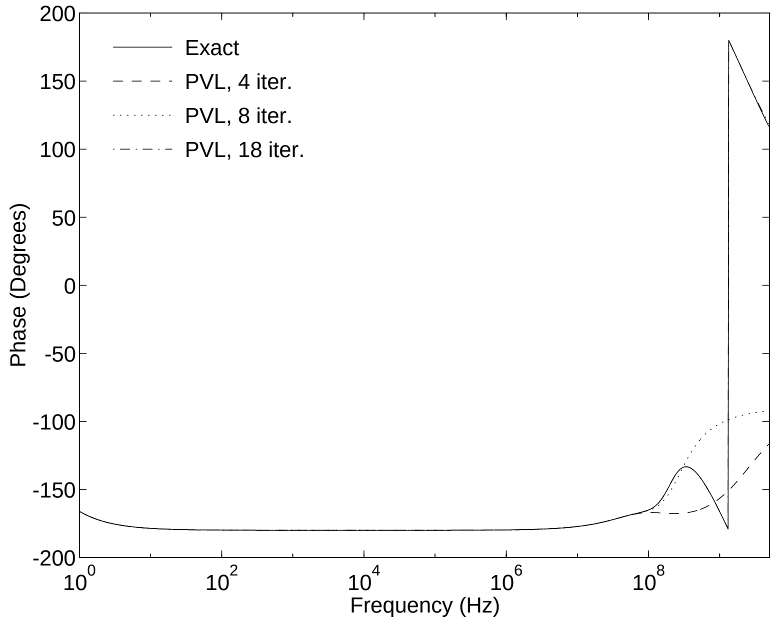

Figure 6: Results for the 2x switched-cap amplifier phase with PVL

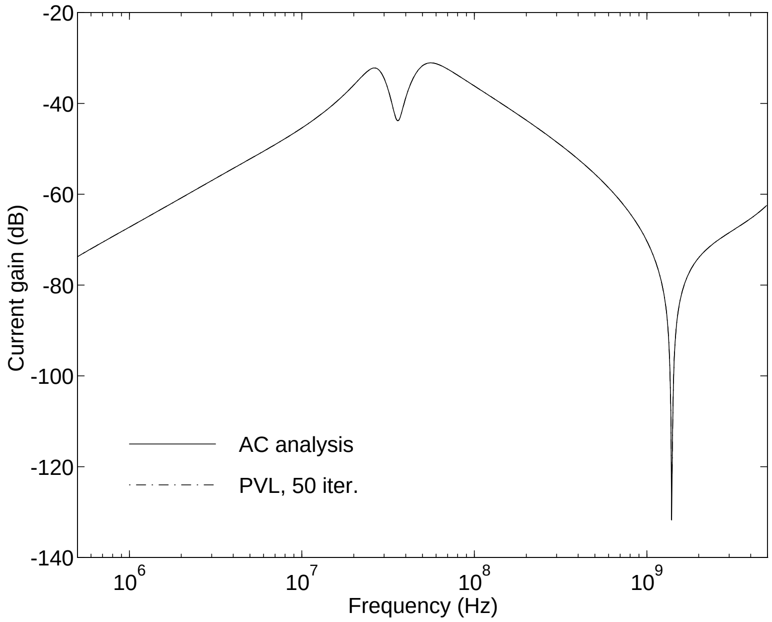

Figure 7: Results for the mixed-signal circuit—magnitude, 50 PVL iterations

5 Conclusions

This paper introduced, a new algorithm for computing the Padé approximation of Laplace-domain transfer function of large linear networks via a Lanczos process. This algorithm has significantly superior numerical stability, while retaining the same efficiency as its predecessors. As a consequence, it can produce more accurate and higher order approximations and render many of the heuristics, that other approaches had to employ, unnecessary. The algorithm also computes an error bound which permits to identify the true poles and zeros of the original network. The paper discusses the application of the algorithm on several large examples.

6 Acknowledgment

We would like to acknowledge the help of Bob Melville, Haideh Khorramabadi from Bell Labs, Rob Rutembar, Emil Ochotta, from CMU, Hansruedi Heeb and Bob Stanisic from IBM for providing us with interesting circuit examples. We also acknowledge the original AWE developers, Larry Pillage, Xiaoli Huang, and Ron Rohrer, who demonstrated the usefulness of the Padé approximation in the analysis of large linear circuits.

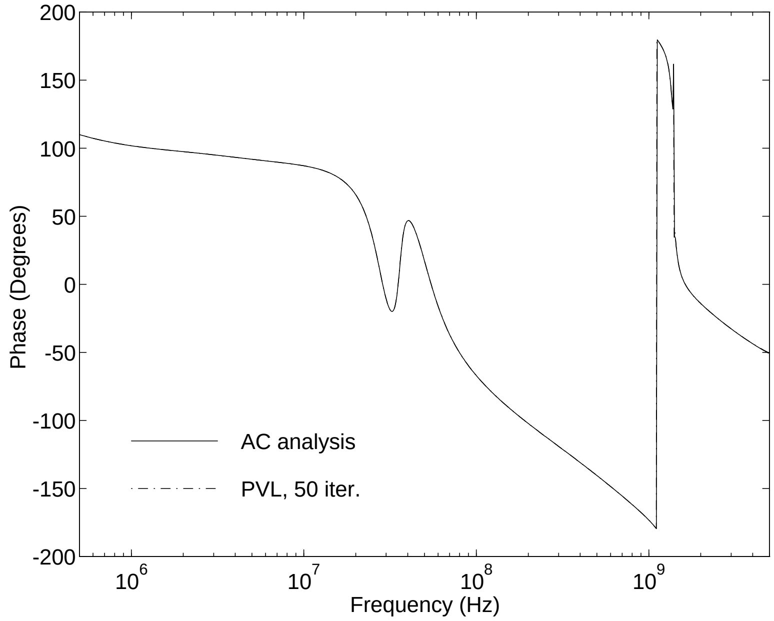

Figure 8: Results for the mixed-signal circuit—phase, 50 PVL iterations

References

[1] L.T. Pillage and R.A. Rohrer. Asymptotic Waveform Evaluation for Timing Analysis. IEEE Trans. Computer-Aided Design, Vol. CAD-9, pp. 352–366, April, 1990.

[2] X. Huang. Padé Approximation of Linear(ized) Circuit Responses. Ph.D. Dissertation, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, November, 1990.

[3] V. Raghavan, R.A. Rohrer, L.T. Pillage, J.Y. Lee, J.E. Bracken, and M.M. Alaybeyi. AWE–Inspired. Proceedings of the IEEE 1993 Custom Integrated Circuits Conference, May, 1993.

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

[5] V. Raghavan, J.E. Bracken, and R.A. Rohrer. AWESpice: A General Tool for the Accurate and Efficient Simulation of Interconnect Problems. Proceedings of the 29th ACM/IEEE Design Automation Conference, pp. 87–92, June, 1992.

[6] E. Chiprout and M. Nakhla. Generalized Moment-Matching Methods for Transient Analysis of Interconnect Networks. Proceedings of the 29th ACM/IEEE Design Automation Conference, pp. 201–206, June, 1992.

[7] M.M. Alaybeyi, J.Y. Lee, and R.A. Rohrer. Numerical Integration Algorithms and Asymptotic Waveform Evaluation (AWE). Technical Digest of the 1992 IEEE/ACM International Conference on Computer-Aided Design, pp. 76–79, November, 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. Van Nostrand Reinhold, New York, N.Y., 1983.

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

[11] MATLAB User's Guide. The MathWorks, 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 Nonsymmetric Matrices and Related Algorithms. Linear Algebra for Large Scale and Real-Time Applications, pp. 137–163. Kluwer Academic Publishers, Dordrecht, The Netherlands, 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, 1993.

[15] 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, 1982.

[16] A.E. Ruehli. Equivalent Circuit Models for Three-Dimensional Multiconductor Systems. IEEE Trans. Microwave Theory and Tech., Vol. MTT-22, pp. 216–221, March, 1974.

[17] H. Heeb, A.E. Ruehli, J.E. Bracken, and R.A. Rohrer. Three Dimensional Circuit Oriented Electromagnetic Modeling for VLSI Interconnects. Proceedings of the 1992 IEEE International Conference on Computer Design: VLSI in Computers & Processors, pp. 218–221, October 1992.

[18] T.J. Schmerbeck, R.A. Richetta, and L.D. Smith. A 27MHz Mixed Analog/Digital Magnetic Recording Channel DSP Using Partial Response Signalling with Maximum Likelihood Detection. Technical Digest of the 1991 IEEE International Solid-State Circuit Conference, pp. 136–137, February, 1991.

[19] B.R. Stanisic, R.A. Rutenbar, and L.R. Carley. Mixed-Signal Noise-Decoupling via Simultaneous Power Distribution Design and Cell Customization in Rail. Proceedings of the 1994 IEEE Custom Integrated Circuits Conference.