Efficient Linear Circuit Analysis by Padé Approximation via the Lanczos Process

P. Feldmann
AT&T Bell Laboratories
Murray Hill, NJ 07974–0636

R. W. Freund
AT&T Bell Laboratories
Murray Hill, NJ 07974–0636

Abstract

This paper describes a highly efficient 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 superior numerical properties, maintains the same computational efficiency as its predecessors, and provides a bound on the 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, require the solution of very large linear networks. The use of SPICE-like simulators, 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] based on Padé approximation [3] has emerged as the method of choice for the efficient analysis of large linear circuits. 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.

AWE, however, suffers from a number of fundamental numerical limitations. In particular, each run of AWE produces only a fairly small number of accurate poles. The proposed remedial techniques are sometimes heuristic, hard to apply automatically, and may be computationally expensive. Another shortcoming of AWE is its inability to estimate the accuracy of the approximating reduced-order model [2, 4].

In this paper, we introduce a new, numerically stable algorithm that computes the Padé approximation of a linear circuit via the Lanczos process [5]. This algorithm, called PVL (**P**adé **V**ia **L**anczos), can be used to generate an arbitrary number of poles and zeros 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 paper is organized as follows. In Section 2, we review system-order reduction by Padé approximation. In Section 3, we demonstrate the numerical limitations of existing algorithms. In Section 4, we derive the new PVL algorithm and discuss some of its properties. In Section 5, we present results of numerical experiments with PVL for a variety of examples.

2 System Reduction by Padé Approx.

Using any standard circuit-equation formulation method such as modified nodal analysis, sparse tableau, etc. [6], a lumped, linear, time-invariant circuit can be described by the following system of first-order differential equations:

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

Here, the vector $\mathbf{x}$ represents the circuit variables, the matrix $\mathbf{G}$ represents memoryless elements, such as resistors, $\mathbf{C}$ represents memory elements, such as capacitors and inductors, y is the output of interest, and $\mathbf{b}u$ represents 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) and 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

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

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

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

we can rewrite (3) as follows:

$$H(s_0 + \sigma) = \mathbf{l}^T (\mathbf{I} - \sigma\mathbf{A})^{-1} \mathbf{r}. \quad (5)$$

When matrix $\mathbf{A}$ is diagonalizable, we obtain

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

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

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

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

The numerical computation of all eigenvectors and eigenvalues of the matrix $\mathbf{A}$ becomes prohibitive 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

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

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

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

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 $p + q + 1$ 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 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 refer to $H_q := H_{q-1,q}$ as the qth *Padé approximant* to the impulse response H. By using a partial-fraction decomposition, we can write H_q in the form

$$H_q(s_0 + \sigma) = \sum_{j=1}^{q} \frac{k_j}{\sigma - p_j}. \quad (9)$$

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

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

where

$$m_k = \mathbf{l}^T\mathbf{A}^k\mathbf{r}, \quad k = 0, 1 \ldots . \quad (10)$$

Note that in the case when the expansion point is chosen as the origin, i.e., $s_0 = 0$, the coefficients m_k, are, up to a constant factor, the time-domain moments of the circuit response. Because of this analogy, we will always refer to the Taylor coefficients (10) as the *moments* of the impulse-response function $H(s_0 + \sigma)$.

3 Limitations of Current Algorithms

In AWE, the Padé approximant H_q is obtained via explicit computation of the leading $2q$ moments, m_k, 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

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

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 because the matrix $\mathbf{G} + s_0\mathbf{C}$ is LU-factored exactly once. 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

$$\mathbf{M}_q \begin{bmatrix} a_q \\ a_{q-1} \\ \vdots \\ a_1 \end{bmatrix} = - \begin{bmatrix} m_q \\ m_{q+1} \\ \vdots \\ m_{2q-1} \end{bmatrix}, \quad (12)$$

where $\mathbf{M}_q = [\, m_{j+k-2} \,]_{j,k=1,2,\ldots,q}$ is the so-called *moment matrix*. The poles p_j of H_q in (9) are then obtained as the roots of the denominator polynomial. Finally, the residues k_j in (9) are computed by solving another linear system of order q, see, e.g., [4].

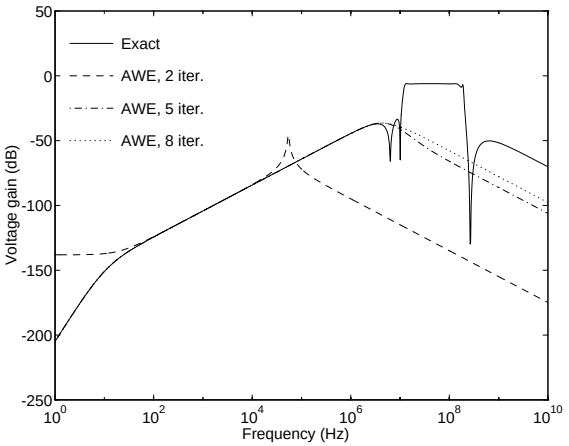

Figure 1: Results for simulation of voltage gain with AWE

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 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™ [7]. We show the exact voltage gain and the approximations generated by AWE for $q = 2, 5, 8$. The results for $q > 8$ showed no further improvement.

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 moments $m_k = \mathbf{l}^T\mathbf{u}_k$ contain only information corresponding to **one** eigenvalue of $\mathbf{A}$, even for fairly small values of k. As a result, the moment matrix $\mathbf{M}_q$ becomes rapidly ill-conditioned. The condition number, $\kappa(\mathbf{M}_q)$, of the matrix $\mathbf{M}_q$ is a measure for how round-off error affects the accuracy of the numerically computed solution of (12). Each increase of $\kappa(\mathbf{M}_q)$ by a factor of 10 signals the loss of one decimal digit of accuracy in the computed solution.

In the first column of Table 1, we list $\kappa(\mathbf{M}_q)$ for the moment matrices corresponding to the simulation of the voltage gain of a filter. Clearly, the moment matrices are extremely ill-conditioned. Scaling of moments, proposed as remedy to this problem, in [2] and [4], provides only a modest improvement in the matrix condition number. In Table 1, we also list the condition numbers of the scaled moment matrices using three proposed scaling strategies.

q	$\kappa(\mathbf{M}_q)$	$\kappa(\mathbf{M}_q^{(1)})$	$\kappa(\mathbf{M}_q^{(2)})$	$\kappa(\mathbf{M}_q^{(3)})$
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
8	3.34e+58	1.25e+18	1.78e+18	2.67e+16
10	1.30e+72	1.57e+18	7.50e+17	5.38e+16

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

4 The PVL Algorithm

We now describe the PVL algorithm that exploits the intimate connection [8] between Padé approximation and the Lanczos process to elude the direct computation of the moments. First, we recall the classical Lanczos process [5] and some of its key properties.

Algorithm 1 (Lanczos algorithm [5])

0) Set $\mathbf{v} = \mathbf{r}$, $\mathbf{w} = \mathbf{l}$, $\mathbf{v}_0 = \mathbf{w}_0 = 0$, and $\delta_0 = 1$.

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

1) Compute $\rho_{n+1} = \|\mathbf{v}\|_2$ and $\eta_{n+1} = \|\mathbf{w}\|_2$.
   If $\rho_{n+1} = 0$ or $\eta_{n+1} = 0$, then stop.

2) Set

$$\mathbf{v}_{n+1} = \frac{\mathbf{v}}{\rho_{n+1}}, \quad \mathbf{w}_{n+1} = \frac{\mathbf{w}}{\eta_{n+1}}, \tag{13}$$

$$\delta_n = \mathbf{w}_n^T\mathbf{v}_n, \quad \alpha_n = \frac{\mathbf{w}_n^T\mathbf{A}\mathbf{v}_n}{\delta_n}, \tag{14}$$

$$\beta_n = \eta_n\frac{\delta_n}{\delta_{n-1}}, \quad \gamma_n = \rho_n\frac{\delta_n}{\delta_{n-1}}, \tag{15}$$

$$\mathbf{v} = \mathbf{A}\mathbf{v}_n - \mathbf{v}_n\alpha_n - \mathbf{v}_{n-1}\beta_n, \tag{16}$$

$$\mathbf{w} = \mathbf{A}^T\mathbf{w}_n - \mathbf{w}_n\alpha_n - \mathbf{w}_{n-1}\gamma_n. \tag{17}$$

We remark that in Algorithm 1 a breakdown will occur if one encounters $\delta_n = 0$ or even $\delta_n \approx 0$ in (14). Therefore, our implementation of the PVL algorithm employs the look-ahead Lanczos algorithm described in [9] that remedies the breakdown problem.

The quantities generated by Algorithm 1 have the following properties:

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

$$\mathbf{w}_j^T \mathbf{v}_k = \begin{cases} \delta_j, & \text{if } j = k, \\ 0, & \text{if } j \neq k, \end{cases} \tag{18}$$

for all $j, k = 1, 2, \ldots, q+1$. Thus, setting

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

we have $\mathbf{D}_q = \mathbf{W}_q^T \mathbf{V}_q = \mathrm{diag}(\delta_1, \delta_2, \ldots, \delta_q)$.

2. The tridiagonal matrices $\tilde{\mathbf{T}}_q$ and $\mathbf{T}_q$ defined by

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

satisfy $\tilde{\mathbf{T}}_q^T = \mathbf{D}_q \mathbf{T}_q \mathbf{D}_q^{-1}$.

3. The matrices $\tilde{\mathbf{T}}_q$ and $\mathbf{T}_q$ have the following relation to the matrix $\mathbf{A}$:

$$\begin{aligned} \mathbf{A}\mathbf{V}_q &= \mathbf{V}_q \mathbf{T}_q + [\, 0 \cdots 0 \ \mathbf{v}_{q+1} \,] \rho_{q+1}, \\ \mathbf{A}^T \mathbf{W}_q &= \mathbf{W}_q \tilde{\mathbf{T}}_q + [\, 0 \cdots 0 \ \mathbf{w}_{q+1} \,] \eta_{q+1}. \end{aligned} \tag{19}$$

Next, we demonstrate the connection of the Lanczos process to Padé approximation. Using the relations in (19) and the fact that $\mathbf{T}_q$ is tridiagonal, one can show that, for all $j = 0, 1, \ldots, q-1$,

$$\begin{aligned} \mathbf{A}^j \mathbf{r} &= \rho_1 \mathbf{A}^j \mathbf{v}_1 = \rho_1 \mathbf{A}^j \mathbf{V}_q \mathbf{e}_1 = \rho_1 \mathbf{V}_q \mathbf{T}_q^j \, \mathbf{e}_1, \\ \mathbf{l}^T \mathbf{A}^j &= \eta_1 \delta_1 \mathbf{e}_1^T \mathbf{T}_q^j \, \mathbf{D}_q^{-1} \, \mathbf{W}_q^T, \end{aligned} \tag{20}$$

where $\mathbf{e}_1 = [1 \quad 0 \quad \cdots \quad 0]^T$ is the first unit vector.

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

$$m_k = \mathbf{l}^T \mathbf{A}^k \mathbf{r} = \left(\mathbf{l}^T \mathbf{A}^{k'} \right) \left(\mathbf{A}^{k''} \mathbf{r} \right), \tag{21}$$

where $k = k' + k''$. If $k \leq 2q-2$, we can find $0 \leq k', k'' \leq q-1$, and from (20)–(21) it follows that

$$m_k = \eta_1 \rho_1 \delta_1 \mathbf{e}_1^T \mathbf{T}_q^{k'} \, \mathbf{D}_q^{-1} \underbrace{\mathbf{W}_q^T \mathbf{V}_q}_{= \mathbf{D}_q} \mathbf{T}_q^{k''} \mathbf{e}_1. \tag{22}$$

Note that $\eta_1 \rho_1 \delta_1 = \mathbf{l}^T \mathbf{r}$, and thus, from (22), we get

$$m_k = (\mathbf{l}^T \mathbf{r}) \cdot \left(\mathbf{e}_1^T \mathbf{T}_q^k \mathbf{e}_1 \right), \quad k = 0, 1, \ldots, 2q-2. \tag{23}$$

Furthermore, it can be shown that the relation (23) also holds for $k = 2q-1$. Thus, by (23), we have

$$\mathbf{l}^T \mathbf{r} \sum_{k=0}^{\infty} \mathbf{e}_1^T \mathbf{T}_q^k \mathbf{e}_1 \, \sigma^k = \sum_{k=0}^{2q-1} m_k \, \sigma^k + \mathcal{O}(\sigma^{2q}), \tag{24}$$

and consequently,

$$H_q(s_0 + \sigma) = \mathbf{l}^T \mathbf{r} \cdot \mathbf{e}_1^T \left(\mathbf{I} - \sigma \mathbf{T}_q \right)^{-1} \mathbf{e}_1 \tag{25}$$

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 (25) 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 $\mathbf{T}_q$:

$$\begin{aligned} H_q(s_0 + \sigma) &= \mathbf{l}^T \mathbf{r} \cdot \underbrace{\mathbf{e}_1^T \mathbf{S}_q}_{= \mu^T} \left(\mathbf{I} - \sigma \mathbf{\Lambda}_q \right)^{-1} \underbrace{\mathbf{S}_q^{-1} \mathbf{e}_1}_{= \nu} \\ &= \sum_{j=1}^{q} \frac{\mathbf{l}^T \mathbf{r} \cdot \mu_j \nu_j}{1 - \sigma \lambda_j}. \end{aligned} \tag{26}$$

Here, $\mathbf{\Lambda}_q = \mathrm{diag}(\lambda_1, \lambda_1, \ldots, \lambda_q)$ contains the eigenvalues of $\mathbf{T}_q$, and μ_j and ν_j are the components of the vectors μ and ν. Finally, from (26), we obtain the pole/residue representation of the Padé approximant:

$$H_q(s_0 + \sigma) = \sum_{\substack{j=1 \\ \lambda_j \neq 0}}^{q} \frac{-\mathbf{l}^T \mathbf{r} \cdot \mu_j \nu_j / \lambda_j}{\sigma - 1/\lambda_j}. \tag{27}$$

The previous derivation shows that 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

$$\mathbf{T}_q = \mathbf{S}_q \, \mathrm{diag}(\lambda_1, \lambda_1, \ldots, \lambda_q) \mathbf{S}_q^{-1} \tag{28}$$

of $\mathbf{T}_q$, and set $\mu = \mathbf{S}_q^T \mathbf{e}_1$ and $\nu = \mathbf{S}_q^{-1} \mathbf{e}_1$.

3) Compute the poles and residues of H_q by setting

$$p_j = 1/\lambda_j \quad \text{and} \quad k_j = \frac{\mathbf{l}^T \mathbf{r} \cdot \mu_j \nu_j}{\lambda_j} \quad \text{for all } j = 1, 2, \ldots . q \tag{29}$$

We remark that the PVL algorithm and AWE require roughly the same amount of computational work. Similar to AWE, the dominating cost is the computation of the LU factorization of the matrix $\mathbf{G} + s_0 \mathbf{C}$ which needs to be computed only once. The vectors $\mathbf{A}\mathbf{v}_n$ and $\mathbf{A}^T \mathbf{w}_n$ required in step 2) of Algorithm 1

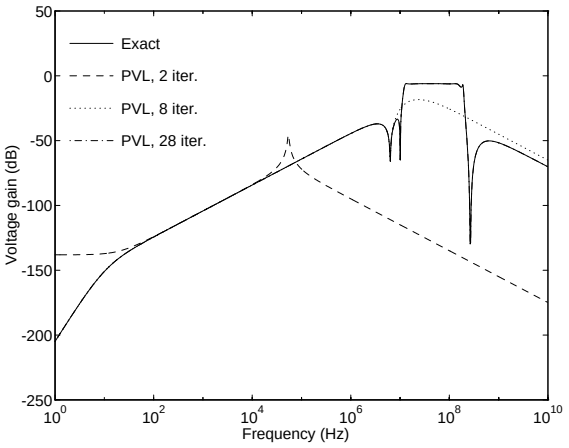

Figure 2: Results for simulation of voltage gain with PVL

are then obtained using forward-backward substitutions. PVL requires $2q$ substitutions to generate the qth Padé approximant, exactly as 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.

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

Finally, we briefly sketch how the PVL algorithm can be used to obtain bounds for the pole approximation error. Recall that, by (29), 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 $\mathbf{A}$.

It can be shown that $\mathbf{A}\mathbf{z}_j - \lambda_j \mathbf{z}_j = \mathbf{v}_{q+1}\rho_{q+1}s_{qj}$, where s_{qj} is the last component of the vector $\mathbf{s}_j$. By taking norms, and since $\|\mathbf{v}_{q+1}\|_2 = 1$, it follows that

$$\frac{\|\mathbf{A}\mathbf{z}_j - \lambda_j \mathbf{z}_j\|_2}{\|\mathbf{A}\|_2 \cdot \|\mathbf{s}_j\|_2} \leq \frac{\rho_{q+1}|s_{qj}|}{n(\mathbf{A}) \cdot \|\mathbf{s}_j\|_2} = Q_j . \qquad (30)$$

where $n(\mathbf{A})$ is an estimate of $\|\mathbf{A}\|_2$, easily obtained from the algorithm. 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.

5 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 4 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 [10] (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 1GHz [11]. In [12], Chiprout et al., 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 involves several complex circuit matrix factorizations and, therefore, is significantly more expensive computationally.

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.

The second example models a complete power grid for a standard cell mixed signal ASIC, including some of the substrate contacts and substrate coupling/decoupling, as described in [13]. The model contains 1074 power bus segments, 36 models for cells, and a coarse, $10 \times 10 \times 1$ 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. Figure 4 shows the magnitude 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 perfect.

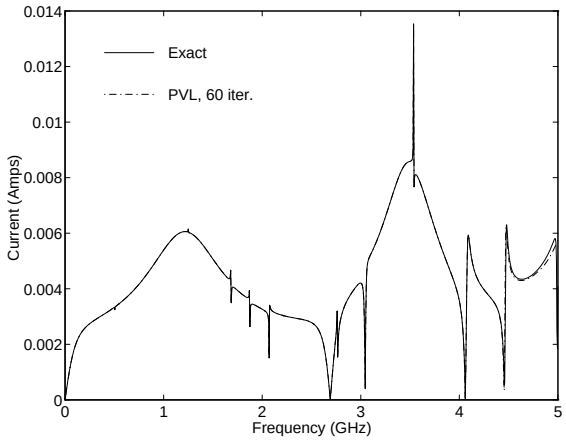

Figure 3: Results for the PEEC circuit, 60 PVL iterations.

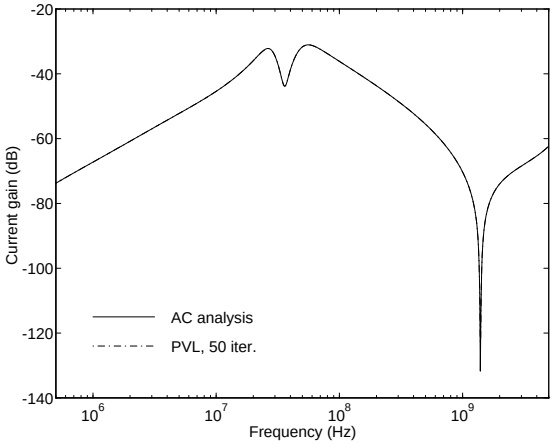

Figure 4: Results for the mixed-signal circuit—magnitude, 50 PVL iterations.

6 Conclusions

This paper introduces PVL, a novel algorithm for linear circuit analysis based on the connection between the Lanczos process and the Padé approximation. Its superior numerical stability allows the computation of more accurate and higher order Padé approximations with no sacrifice in efficiency. As a consequence the usefulness of Padé approximation based linear circuit analysis techniques is considerably extended.

Acknowledgments

We would like to thank R. Melville, R. Rutenbar, H. Heeb, and B. Stanisic 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] X. Huang, "Padé approximation of linear(ized) circuit responses," Ph.D. diss., Dept. of Electrical and Computer Engineering, Carnegie Mellon Univ., Pittsburgh, PA, Nov. 1990.

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

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

[5] 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.

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

[7] *MATLAB User's Guide*. The MathWorks, Inc., Natick, MA, 1992.

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

[9] 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.

[10] A.E. Ruehli, "Equivalent circuit models for three-dimensional multiconductor systems," *IEEE Trans. Microwave Theory and Tech.*, vol. 22, pp. 216–221, Mar. 1974.

[11] H. Heeb, A.E. Ruehli, J.E. Bracken, and R.A. Rohrer, "Three dimensional circuit oriented electromagnetic modeling for VLSI interconnects," in *Proc. Int. Conf. Computer Design: VLSI in Computers & Processors*, Oct. 1992.

[12] E. Chiprout, H. Heeb, M.S. Nakhla, and A.E. Ruehli, "Simulating 3-D retarded interconnect models using complex frequency hopping (CFH)," in *Tech. Dig. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1993.

[13] B.R. Stanisic, R.A. Rutenbar, and L.R. Carley, "Mixed-signal noise-decoupling via simultaneous power distribution design and cell customization in rail," in *Proc. IEEE Custom Integrated Circuits Conf.*, 1994.