Small-Signal Circuit Analysis and Sensitivity Computations with the PVL Algorithm

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

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

Abstract. We describe the application of the PVL algorithm to the small-signal analysis of circuits, including sensitivity computations. The PVL algorithm is based on the efficient computation of the Padé approximation of the network transfer function via the Lanczos process. The numerical stability of the algorithm permits the computation of the Padé approximation to any accuracy over a certain frequency range. We extend the algorithm to compute sensitivities of network transfer functions, their poles, and their zeros, with respect to arbitrary circuit parameters, with minimal additional computational cost. We demonstrate the implementation of our algorithm on circuit examples.

1 Introduction

The process of analyzing analog circuits with full accounting of parasitic elements, interconnect analysis at the board or chip level, and numerous other circuit-simulation tasks often require the analysis 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 time-domain differential-equation integration or complex phasor analysis, as implemented in SPICE-like simulators, would be inefficient or even prohibitive for such large problems.

Recently, in [1, 2], we have introduced a new algorithm for the efficient analysis of large linear networks. This algorithm, called PVL (**P**adé **V**ia **L**anczos), computes in a numerically stable way the Padé approximation of a linear circuit response via the Lanczos process [3]. A key feature of the PVL algorithm is the fact that the accuracy of the approximation can be guaranteed for a specified frequency range. In addition, the efficiency of the algorithm is such that it allows computation of the circuit response over an entire frequency range with similar computational effort as classical phasor analysis would require for a single frequency point. Therefore, PVL can be used to replace complex phasor analysis altogether. In Figure 1, we

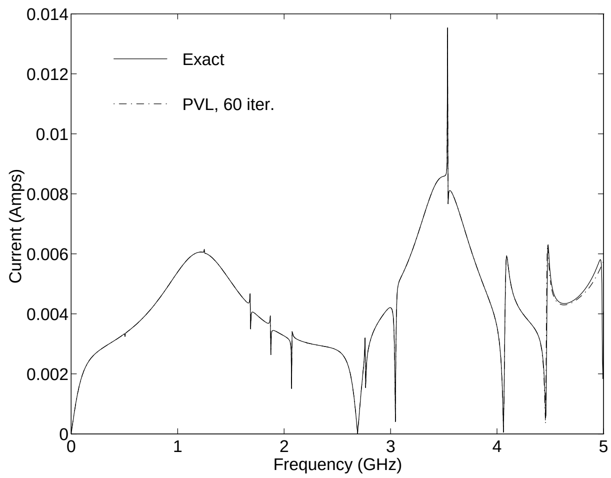

Figure 1: Results for the PEEC circuit, 60 PVL iterations

illustrate this point with an example taken from [2]. Here, we simulate a lumped-element equivalent circuit for a three-dimensional electromagnetic problem modeled via PEEC [4] (partial-element equivalent circuit). Using a single Padé approximant (of order 60), PVL accurately simulates the circuit response over a frequency range of 5 GHz. One would need to carry out a high-resolution complex phasor analysis involving a large number of frequency points to capture all the resonance frequencies. The response shown in Figure 1 was obtained with the PVL algorithm at a cost that is smaller than phasor analysis for a single frequency point.

In addition to efficiency and accuracy, the PVL algorithm has other features that make it useful in the analysis of large linear circuits. One important application is the computation of sensitivities of the frequency response or other measures related to it, such as the location of its poles or zeros. In this paper, we describe the application of the PVL algorithm to the small-signal analysis of circuits, including sensitivity computations.

The remainder of the paper is organized as follows. In Section 2, we briefly review the small-signal analysis of circuits to establish our notation, discuss circuit sensitivities, and recall the Padé-Lanczos connection on which PVL is based. In Section 3, we present the extension of the Lanczos algorithm needed to implement sensitivity computations. In Section 4, we describe an extended PVL algorithm for the computation of circuit sensitivities. In Section 5, we present results of numerical experiments with this extended version of PVL. In Section 6, we make some concluding remarks.

2 Small-Signal Circuit Equations and Sensitivities

2.1 Circuit Equations

In order to establish notation, we briefly review the small-signal analysis of circuits. All of the standard circuit-equation formulation methods, such as MNA, sparse tableau, etc. [5]

lead to a system of differential equations of the form

$$\mathbf{f}(\mathbf{z}) + \frac{d}{dt}\mathbf{q}(\mathbf{z}) - \mathbf{e}(t) = 0. \tag{1}$$

Here, $\mathbf{z} = \mathbf{z}(t)$ is the vector of circuit variables at time t, the term $\mathbf{f}(\mathbf{z})$ represents the contribution of nonreactive elements such as resistors, sources, etc., the term $\frac{d}{dt}\mathbf{q}(\mathbf{z})$ represents the contribution of reactive elements such as capacitors and inductors, and $\mathbf{e}(t)$ is the excitation applied to the circuit. We assume that, in (1), the excitation $\mathbf{e}(t)$ is "small", and thus all circuit variables $\mathbf{z}$ will be "close" to equilibrium values, $\mathbf{z}_0$, which represent the DC operating point of the circuit. Therefore, the solution of the circuit equations (1) will be of the form $\mathbf{z}(t) = \mathbf{z}_0 + \mathbf{x}(t)$, where $\mathbf{z}_0$ is a constant (DC operating point) component, and $\mathbf{x}(t)$ is a variable (small-signal) component. Note that the DC operating point $\mathbf{z}_0$ is found as the solution of the system of nonlinear equations $\mathbf{f}(\mathbf{z}_0) = 0$.

By expanding the terms $\mathbf{f}(\mathbf{z})$ and $\mathbf{q}(\mathbf{z})$ in the circuit equations (1) into Taylor series about the operating point $\mathbf{z}_0$, we obtain

$$\left[\mathbf{f}(\mathbf{z}_0) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{z}}\right|_{\mathbf{z}_0} \mathbf{x}(t) + \cdots\right] + \frac{d}{dt}\left[\mathbf{q}(\mathbf{z}_0) + \left.\frac{\partial \mathbf{q}}{\partial \mathbf{z}}\right|_{\mathbf{z}_0} \mathbf{x}(t) + \cdots\right] - \mathbf{e}(t) = 0. \tag{2}$$

Since we assumed that all the deviations from the equilibrium values are "small", the second and higher-order terms of the Taylor expansion can be neglected in (2). Using $\mathbf{f}(\mathbf{z}_0) = 0$ and $\frac{d}{dt}\mathbf{q}(\mathbf{z}_0) = 0$, and setting

$$\mathbf{G} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{z}}\right|_{\mathbf{z}_0} \quad \text{and} \quad \mathbf{C} = \left.\frac{\partial \mathbf{q}}{\partial \mathbf{z}}\right|_{\mathbf{z}_0}, \tag{3}$$

we obtain from (2) the linear dynamic system

$$\mathbf{G}\,\mathbf{x}(t) + \mathbf{C}\,\frac{d}{dt}\mathbf{x}(t) - \mathbf{e}(t) = 0 \tag{4}$$

that needs to be solved to obtain the $\mathbf{x}(t)$ component of the solution.

Currently, this system is most often analyzed in frequency domain, using complex phasor analysis (the AC analysis of SPICE-type simulators). Phasor analysis is performed by sweeping the frequency over the entire range of interest and computing the response of the circuit at each frequency. In this paper, we present a more accurate and efficient alternative, namely the computation of the Laplace-domain frequency response that can be used to predict the response of the circuit to any excitation. The frequency response, as well as its poles, zeros, and residues, are very important to the circuit designer and, therefore, a circuit simulator must be capable to compute them numerically.

Since the system (4) is linear, the superposition law holds, and hence all sources of excitation can be treated separately. Therefore, without loss of generality, we can assume that there is only one source of excitation, $\mathbf{e}(t) = \mathbf{b}u(t)$. Then, the small-signal linear time-invariant dynamic system (4) is described by the following system of first-order differential equations:

$$\begin{aligned} \mathbf{C}\,\frac{d}{dt}\mathbf{x} &= -\mathbf{G}\,\mathbf{x} + \mathbf{b}\,u(t), \\ y &= \mathbf{l}^T\mathbf{x}. \end{aligned} \tag{5}$$

Here, $\mathbf{x} = \mathbf{x}(t)$ represents the small-signal circuit variables, and y is the output of interest. Applying the Laplace transform to the system (5) and assuming zero initial conditions, we obtain

$$\begin{aligned} s\,\mathbf{C}\,\mathbf{X} &= -\mathbf{G}\,\mathbf{X} + \mathbf{b}\,U, \\ Y &= \mathbf{l}^T\mathbf{X}, \end{aligned} \tag{6}$$

where $\mathbf{X}$, U, and Y denote the Laplace transform of $\mathbf{x}$, u, and y, respectively. It follows from (6) that the Laplace-domain *frequency response*, or *transfer function*, defined as $H(s) = Y(s)/U(s)$, is given by

$$H(s) = \mathbf{l}^T \left(\mathbf{G} + s\mathbf{C}\right)^{-1} \mathbf{b}. \tag{7}$$

The frequency response is in fact a rational function in the complex frequency variable s, and it can be expressed in the familiar pole-zero or pole-residue forms

$$H(s) = a\frac{(s-z_1)(s-z_2)\cdots(s-z_{N-1})}{(s-p_1)(s-p_2)\cdots(s-p_N)} = k_\infty + \sum_{j=1}^{N} \frac{k_j}{s-p_j}.$$

Here, N denotes the length of the vectors $\mathbf{b}$ and $\mathbf{l}$. The complex numbers p_j, k_j, and z_j, denote the poles, residues, and zeros, respectively, of the frequency response.

2.2 Padé Approximation of the Frequency Response and PVL

Unfortunately, the exact computation of the frequency response and related quantities involves the eigendecomposition of the circuit matrices, a task that quickly becomes impractical as the size of the circuit increases. Even if this computation were possible, the designer would be overwhelmed by hundreds or even thousands of poles, residues, and zeros, of which only a small fraction is relevant. Therefore, it is important to develop approximate methods that compute the frequency response in terms of only a small number of dominant poles. Efficient methods of this type can be obtained by using Padé approximations of the frequency response. This was first demonstrated by Pillage and Rohrer [6] with their asymptotic waveform evaluation (AWE) approach; for a detailed description of AWE, we refer the reader to [7, 8] and the references given there.

The basic idea of Padé approximation of the frequency response (7) is as follows. 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. Here, $\mathbf{G}$ and $\mathbf{C}$ are the matrices from the small-signal linear system (5). We now approximate the frequency response (7) in a region in the complex plane about the point s_0. To this end, we use the change of variables $s = s_0 + \sigma$, and we set

$$\mathbf{A} = -\left(\mathbf{G} + s_0\mathbf{C}\right)^{-1}\mathbf{C}, \quad \mathbf{r} = \left(\mathbf{G} + s_0\mathbf{C}\right)^{-1}\mathbf{b}. \tag{8}$$

Then, from (7), we get

$$\begin{aligned} H(s_0+\sigma) &= \mathbf{l}^T \left(\mathbf{G} + s_0\mathbf{C} + \sigma\mathbf{C}\right)^{-1} \mathbf{b} \\ &= \mathbf{l}^T \left(\mathbf{I} - \sigma\mathbf{A}\right)^{-1} \mathbf{r}, \end{aligned} \tag{9}$$

where $\mathbf{I}$ denotes the identity matrix. By expanding the function (9) into a Taylor series about $\sigma = 0$, we obtain

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

Here, $m_k = \mathbf{l}^T \mathbf{A}^k \mathbf{r}$, $k = 0, 1, \ldots$, are the so-called *moments*.

For a given integer $q \geq 1$, a rational function of the form

$$H_q(s_0 + \sigma) = \frac{b_0 + b_1\sigma + \cdots + b_{q-1}\sigma^{q-1}}{1 + a_1\sigma + a_2\sigma^2 + \cdots + a_q\sigma^q}, \tag{10}$$

with coefficients $a_1, a_2, \ldots, a_q, b_0, b_1, \ldots, b_{q-1} \in \mathbb{C}$, is called a *Padé approximant* (of *order* q) to the frequency response $H(s_0 + \sigma)$ if the Taylor series of H and H_q about $\sigma = 0$ agree in at least the first $2q$ terms, i.e.,

$$\begin{aligned} H_q(s_0 + \sigma) &= \sum_{k=0}^{2q-1} m_k \sigma^k + \mathcal{O}(\sigma^{2q}) \\ &= H(s_0 + \sigma) + \mathcal{O}(\sigma^{2q}). \end{aligned} \tag{11}$$

Note that the condition (11) just represents $2q$ equations for the $2q$ free parameters $a_1, a_2, \ldots, a_q, b_0, b_1, \ldots, b_{q-1}$ in (10), which suggests that a Padé approximant H_q of order q always exists. Indeed, it can be shown that—except for certain degenerate cases—the function H_q exists and is unique; see, e.g., [9].

In AWE, the Padé approximant H_q is directly generated from the moments. Unfortunately, this procedure is inherently numerically unstable, as we illustrated in [2]. In fact, this is the reason why, in practice, AWE can be used only for fairly moderate values of q. In [1, 2], we proposed a different algorithm, called PVL, for computing H_q. The PVL algorithm bypasses the moments, and instead, PVL exploits the well-known connection [10] between the Padé approximant H_q and the Lanczos process [3]. This connection states that the Padé approximant H_q is simply given as

$$H_q(s_0 + \sigma) = \mathbf{l}^T\mathbf{r} \cdot \mathbf{e}_1^T \left(\mathbf{I} - \sigma\mathbf{T}_q\right)^{-1} \mathbf{e}_1. \tag{12}$$

Here, $\mathbf{e}_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T \in \mathbb{R}^q$ is the first unit vector in $\mathbb{R}^q$, and $\mathbf{T}_q$ is the tridiagonal matrix generated by the nonsymmetric Lanczos process; see Section 3.1 below. For a detailed description of the PVL algorithm and its properties, we refer the reader to [2].

2.3 Sensitivities

Information about the frequency response and its poles, zeros, and residues, becomes even more valuable to the designer if, in addition, sensitivities of these quantities with respect to various circuit and device parameters can also be computed. Sensitivities can be used to determine the robustness of the design to manufacturing variations, to asses the stability of the circuit, and in design optimization.

In the following, we consider the calculation of sensitivities of frequency-response quantities with respect to some parameter $p \in \mathbb{R}$. From now on, we always use $'$ to denote derivatives with respect to p. Moreover, we assume that the matrices $\mathbf{C}$, $\mathbf{G}$ and the vectors $\mathbf{b}$, $\mathbf{l}$ in (5) are differentiable with respect to p. This guarantees that the quantities $\mathbf{A}$, $\mathbf{l}$, and $\mathbf{r}$, which define the frequency response $H(s_0+\sigma)$ in (9), are also differentiable, and we denote by

$$\mathbf{A}' = \frac{\partial \mathbf{A}}{\partial p}, \quad \mathbf{l}' = \frac{\partial \mathbf{l}}{\partial p}, \quad \text{and} \quad \mathbf{r}' = \frac{\partial \mathbf{r}}{\partial p}$$

their derivatives.

In Section 3.2 below, we will see that the Lanczos tridiagonal matrix $\mathbf{T}_q$ is differentiable, and we give an extended Lanczos algorithm for the computation of $\mathbf{T}_q'$. In particular, by differentiating the representation (12) of H_q, we obtain the following formula for the sensitivity of the approximate frequency response:

$$\begin{aligned} H_q'(s_0+\sigma) &= \frac{\partial}{\partial p} H_q(s_0+\sigma) \\ &= \mathbf{e}_1^T \left(\mathbf{I} - \sigma \mathbf{T}_q\right)^{-1} \left[\left(\mathbf{l}^T \mathbf{r}' + \mathbf{r}^T \mathbf{l}' \right) \mathbf{e}_1 \; + \; \sigma \cdot \mathbf{l}^T \mathbf{r} \cdot \mathbf{T}_q' \left(\mathbf{I} - \sigma \mathbf{T}_q\right)^{-1} \mathbf{e}_1 \right]. \end{aligned} \tag{13}$$

Formulas for the sensitivities of the frequency-response poles and zeros will be given in Section 4.3 below.

3 An Extended Lanczos Algorithm

In this section, we first briefly review the classical nonsymmetric Lanczos process, and then we present an extended version for the computation of $\mathbf{T}_q'$.

3.1 The Classical Lanczos Algorithm

The classical nonsymmetric Lanczos algorithm (applied to the matrix $\mathbf{A}$ and the vectors $\mathbf{l}$, $\mathbf{r}$ from (9)) can be stated as follows; note that the matrix $\mathbf{A}$ is in general nonsymmetric.

Algorithm 1 (Lanczos algorithm [3])

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

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

1) Compute $\rho_n = \|\mathbf{v}\|_2$ and $\eta_n = \|\mathbf{w}\|_2$.
   If $\rho_n = 0$ or $\eta_n = 0$, then stop.

2) Set

$$\mathbf{v}_n = \frac{\mathbf{v}}{\rho_n}, \quad \mathbf{w}_n = \frac{\mathbf{w}}{\eta_n},$$

$$\delta_n = \mathbf{w}_n^T \mathbf{v}_n, \tag{14}$$

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

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

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

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, we stress that these problems can be remedied by using a so-called look-ahead variant of the Lanczos algorithm, see [11, 12]. In fact, in our implementation of the PVL algorithm we employ the look-ahead Lanczos algorithm described in [12].

The Lanczos algorithm has numerous interesting properties and applications; we refer the reader to [11, 12, 13] and the references given there. The key property of Algorithm 1 is that it produces a very useful small-dimensional approximation, called $\mathbf{T}_q$ in the sequel, to the (usually large-dimensional) matrix $\mathbf{A}$. More precisely, $\mathbf{T}_q$ is the $q \times q$ tridiagonal matrix

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

where ρ_n, α_n, and β_n are the quantities generated by Algorithm 1. The matrix $\mathbf{T}_q$ is—in some sense—the best $q \times q$ approximation to $\mathbf{A}$. We stress that this approximation 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, $q \ll N$. Note that $\mathbf{T}_q$ is just the matrix that appears in the important formula (12) on which PVL is based.

3.2 The Extended Algorithm for Sensitivity Computations

Recall from (13) that the sensitivity of the approximate frequency response $H_q(s_0 + \sigma)$ with respect to the parameter p involves the derivative, $\mathbf{T}'_q$, of the Lanczos tridiagonal matrix $\mathbf{T}_q$. Next, we show how to compute $\mathbf{T}'_q$.

First, we note that—since $\mathbf{A}$, $\mathbf{l}$, and $\mathbf{r}$ are differentiable—all quantities generated by the Lanczos Algorithm 1 are also differentiable, and thus, by (16), we have

$$\mathbf{T}'_q = \begin{bmatrix} \alpha'_1 & \beta'_2 & 0 & \cdots & 0 \\ \rho'_2 & \alpha'_2 & \beta'_3 & \ddots & \vdots \\ 0 & \rho'_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta'_q \\ 0 & \cdots & 0 & \rho'_q & \alpha'_q \end{bmatrix}.$$

Moreover, the entries ρ'_n, α'_n, and β'_n of $\mathbf{T}'_n$ can be computed by means of an "extended" version of the Lanczos algorithm that is obtained by differentiating all equations in Algorithm 1 and adding the differentiated relations to Algorithm 1; the resulting extended Lanczos process is stated as Algorithm 2 below. For example, differentiating (14) gives

$$\delta'_n = \mathbf{w}_n^T \mathbf{v}'_n + (\mathbf{w}'_n)^T \mathbf{v}_n, \tag{17}$$

and differentiating the first relation in (15) gives

$$\alpha_n' = \frac{(\mathbf{w}_n')^T \mathbf{A}\mathbf{v}_n + \mathbf{w}_n^T (\mathbf{A}\mathbf{v}_n)'}{\delta_n} - \mathbf{w}_n^T \mathbf{A}\mathbf{v}_n \frac{\delta_n'}{\delta_n^2}. \tag{18}$$

Note that, by means of (17) and the formula for α_n in (15), we can rewrite (18) in the form

$$\begin{aligned}\alpha_n' &= \frac{(\mathbf{w}_n')^T \mathbf{A}\mathbf{v}_n + \mathbf{w}_n^T (\mathbf{A}\mathbf{v}_n)' - \alpha_n \left(\mathbf{w}_n^T \mathbf{v}_n' + (\mathbf{w}_n')^T \mathbf{v}_n\right)}{\delta_n} \\ &= \frac{\mathbf{w}_n^T \left((\mathbf{A}\mathbf{v}_n)' - \mathbf{v}_n'\alpha_n\right) + (\mathbf{w}_n')^T (\mathbf{A}\mathbf{v}_n - \mathbf{v}_n\alpha_n)}{\delta_n},\end{aligned}$$

which is just the equation (20) in the following extended Lanczos algorithm.

Algorithm 2 (Extended Lanczos algorithm, including sensitivity computations)

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

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

1) Compute $\rho_n = \|\mathbf{v}\|_2$ and $\eta_n = \|\mathbf{w}\|_2$.
   If $\rho_n = 0$ or $\eta_n = 0$, then stop.

2) Set

$$\mathbf{v}_n = \frac{\mathbf{v}}{\rho_n}, \quad \mathbf{w}_n = \frac{\mathbf{w}}{\eta_n},$$

$$\rho_n' = \mathbf{v}_n^T \mathbf{v}', \quad \eta_n' = \mathbf{w}_n^T \mathbf{w}',$$

$$\mathbf{v}_n' = \frac{\mathbf{v}' - \mathbf{v}_n \rho_n'}{\rho_n}, \quad \mathbf{w}_n' = \frac{\mathbf{w}' - \mathbf{w}_n \eta_n'}{\eta_n},$$

$$\delta_n = \mathbf{w}_n^T \mathbf{v}_n, \quad \delta_n' = \mathbf{w}_n^T \mathbf{v}_n' + \mathbf{v}_n^T \mathbf{w}_n',$$

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

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

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

$$\beta_n' = \eta_n' \frac{\delta_n}{\delta_{n-1}} + \beta_n \left(\frac{\delta_n'}{\delta_n} - \frac{\delta_{n-1}'}{\delta_{n-1}}\right), \quad \gamma_n' = \rho_n' \frac{\delta_n}{\delta_{n-1}} + \gamma_n \left(\frac{\delta_n'}{\delta_n} - \frac{\delta_{n-1}'}{\delta_{n-1}}\right),$$

$$\mathbf{t}_1 = (\mathbf{A}\mathbf{v}_n)' - \mathbf{v}_n'\alpha_n, \quad \mathbf{t}_2 = \left(\mathbf{A}^T \mathbf{w}_n\right)' - \mathbf{w}_n'\alpha_n, \tag{19}$$

$$\alpha_n' = \frac{\mathbf{w}_n^T \mathbf{t}_1 + (\mathbf{w}_n')^T (\mathbf{A}\mathbf{v}_n - \mathbf{v}_n\alpha_n)}{\delta_n}, \tag{20}$$

$$\mathbf{v}' = \mathbf{t}_1 - \mathbf{v}_n \alpha_n' - \mathbf{v}_{n-1}\beta_n' - \mathbf{v}_{n-1}'\beta_n,$$

$$\mathbf{w}' = \mathbf{t}_2 - \mathbf{w}_n \alpha_n' - \mathbf{w}_{n-1}\gamma_n' - \mathbf{w}_{n-1}'\gamma_n.$$

4 Computing Sensitivities with PVL

In [14], the AWE algorithm was extended to compute sensitivities of poles and zeros of the frequency response. In this section, we show how the PVL algorithm can be used to generate in a numerically stable manner sensitivities of the frequency response and related quantities with respect to some parameter $p \in \mathbb{R}$. We note that the advantages of PVL over AWE are preserved in sensitivity computations, and our method is not restricted to low-order Padé approximants.

4.1 A Formula for the Approximate Frequency Response

When the representation (12) is used to evaluate the approximate frequency response $H_q(s_0+\sigma)$, a tridiagonal linear system with the coefficient matrix $\mathbf{I} - \sigma \mathbf{T}_q$ has to be solved for each value of σ. A more economic way is to precompute the eigendecomposition

$$\mathbf{T}_q = \mathbf{S}_q \operatorname{diag}(\lambda_1, \lambda_1, \ldots, \lambda_q) \mathbf{S}_q^{-1} \tag{21}$$

of $\mathbf{T}_q$. Here, $\lambda_1, \lambda_2, \ldots, \lambda_q$ are the eigenvalues of $\mathbf{T}_q$, and the columns of the matrix $\mathbf{S}_q$ are the corresponding eigenvectors. Then, setting

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_q \end{bmatrix} = \mathbf{S}_q^T \mathbf{e}_1 \quad \text{and} \quad \nu = \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_q \end{bmatrix} = \mathbf{S}_q^{-1} \mathbf{e}_1, \tag{22}$$

we obtain from (12) the formula

$$H_q(s_0+\sigma) = \mathbf{l}^T \mathbf{r} \cdot \underbrace{\mathbf{e}_1^T \mathbf{S}_q}_{=\mu^T} (\mathbf{I} - \sigma \mathbf{\Lambda}_q)^{-1} \underbrace{\mathbf{S}_q^{-1} \mathbf{e}_1}_{=\nu} = \sum_{j=1}^{q} \frac{\mathbf{l}^T \mathbf{r} \cdot \mu_j \nu_j}{1 - \sigma \lambda_j}. \tag{23}$$

From (23), we immediately get the pole-residue representation of the Padé approximant:

$$H_q(s_0+\sigma) = k_\infty + \sum_{\substack{j=1 \\ \lambda_j \neq 0}}^{q} \frac{-\mathbf{l}^T \mathbf{r} \cdot \mu_j \nu_j / \lambda_j}{\sigma - 1/\lambda_j} \tag{24}$$

The term k_∞ in (24) may result if one of the eigenvalues of $\mathbf{T}_q$ is zero.

4.2 A Formula for the Frequency-Response Sensitivity

In analogy to (23), we can also derive a computationally more economic formula for the frequency-response sensitivity (13). Indeed, by means of (21) and (22), it follows from (13) that

$$H_q'(s_0+\sigma) = \sum_{j=1}^{q} \frac{\mu_j}{1-\sigma\lambda_j} \left[\left(\mathbf{l}^T \mathbf{r}' + \mathbf{r}^T \mathbf{l}' \right) \nu_j \; + \; \sigma \cdot \mathbf{l}^T \mathbf{r} \cdot \sum_{k=1}^{q} \frac{\nu_k c_{jk}}{1 - \sigma \lambda_k} \right]. \tag{25}$$

Here, the coefficients c_{jk} are the entries of the matrix

$$\mathbf{C} = [\, c_{jk} \,]_{j,k=1,2,\ldots,q} = \mathbf{S}_q^{-1} \, \mathbf{T}_q' \, \mathbf{S}_q. \tag{26}$$

4.3 Sensitivities of Poles and Zeros of the Frequency Response

By (24), the poles p_j of H_q are given as $p_j = 1/\lambda_j$, where $\lambda_j \neq 0$, $j = 1, 2, \ldots, q$, are just the nonzero eigenvalues of $\mathbf{T}_q$. Therefore, the sensitivity p_j' of the jth pole p_j is given by

$$p_j' = -\frac{\lambda_j'}{\lambda_j^2}. \tag{27}$$

Thus, it only remains to show how to obtain the sensitivity λ_j' of the jth eigenvalue λ_j of $\mathbf{T}_q$. To this end, we use standard perturbation theory for eigenvalues; see, e.g., [15, 16]. First, for simplicity, we assume that λ_j is a simple eigenvalue of $\mathbf{T}_q$. Then from the results in [15] it follows that

$$\lambda_j' = \mathbf{y}_j^T \, \mathbf{T}_q' \, \mathbf{s}_j, \tag{28}$$

where $\mathbf{s}_j$ is the jth column of the eigenvector matrix $\mathbf{S}_q$ in (21) and $\mathbf{y}_j^T$ is the jth row of $\mathbf{S}_q^{-1}$. The case that λ_j is a multiple eigenvalue can be treated similarly, see [16]. The eigenvalue sensitivities are then given as the eigenvalues of the matrix $\left(\mathbf{Y}^{(j)}\right)^T \mathbf{T}_q' \, \mathbf{S}^{(j)}$. Here, $\mathbf{S}^{(j)}$ is the submatrix of the matrix $\mathbf{S}_q$ in (21) that contains all vectors associated with the multiple eigenvalue λ_j, and $\left(\mathbf{Y}^{(j)}\right)^T$ is the corresponding submatrix of $\mathbf{S}_q^{-1}$.

Finally, we note that the sensitivities of the zeros of H_q can be computed in an analogous manner, by using the fact (see [2]) that the zeros of H_q are just the inverse eigenvalues of the matrix

$$\widetilde{\mathbf{T}}_q = \begin{bmatrix} \alpha_2 & \beta_3 & 0 & \cdots & 0 \\ \rho_3 & \alpha_3 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_q \\ 0 & \cdots & 0 & \rho_q & \alpha_q \end{bmatrix} \tag{29}$$

obtained from $\mathbf{T}_q$ by deleting the first row and column.

4.4 A Sketch of the Extended PVL Algorithm

We now summarize our extended PVL algorithm for sensitivity computations.

Algorithm 3 (Sketch of the extended PVL algorithm)

1) Run q steps of the extended Lanczos process (Algorithm 2) to obtain the tridiagonal matrix $\mathbf{T}_q$ and its derivative $\mathbf{T}_q'$.

2) Compute an eigendecomposition

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

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

$$p_j = 1/\lambda_j \quad \text{and} \quad k_j = -\frac{\mathbf{l}^T\mathbf{r} \cdot \mu_j\nu_j}{\lambda_j} \quad \text{for all} \quad j = 1, 2, \ldots, q \quad \text{with} \quad \lambda_j \neq 0,$$

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

4) Compute the frequency-response sensitivity $H_q'(s_0+\sigma)$ by means of (25) and (26).
5) Compute the pole sensitivities p_j' by means of (27) and (28).
6) If desired, compute the zero sensitivities using an eigendecomposition of the matrix (29).

4.5 Computing Derivatives of Circuit Quantities

The extended Lanczos Algorithm 2 requires the computation of the derivatives of $\mathbf{A}$, $\mathbf{l}$, and $\mathbf{r}$. Recall from (8) that

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

where $\mathbf{b}$ and $\mathbf{l}$ are vectors that depend on the nature of excitation and of the output of interest, respectively.

We need, therefore, to compute matrices $\mathbf{G}'$, $\mathbf{C}'$, and the vectors $\mathbf{b}'$ and $\mathbf{l}'$. Again, we denote by p the parameter with respect to which we want to compute sensitivities. Then, from (3), we get

$$\mathbf{G}' = \frac{\partial}{\partial p}\left[\frac{\partial \mathbf{f}}{\partial \mathbf{z}}(\mathbf{z}_0(p), p)\right] = \frac{\partial^2 \mathbf{f}}{\partial \mathbf{z}^2}\frac{\partial \mathbf{z}_0}{\partial p} + \frac{\partial^2 \mathbf{f}}{\partial p\,\partial \mathbf{z}} \tag{30}$$

and

$$\mathbf{C}' = \frac{\partial^2 \mathbf{q}}{\partial \mathbf{z}^2}\frac{\partial \mathbf{z}_0}{\partial p} + \frac{\partial^2 \mathbf{q}}{\partial p\,\partial \mathbf{z}}. \tag{31}$$

The vector $\partial\mathbf{z}_0/\partial p$ represents the sensitivity of the operating point with respect to the parameter p and can be obtained by solving the linear system that results from the differentiation of the operating point equations

$$\mathbf{f}(\mathbf{z}_0(p), p) = 0,$$

$$\frac{\partial \mathbf{f}}{\partial \mathbf{z}}\frac{\partial \mathbf{z}_0}{\partial p} + \frac{\partial \mathbf{f}}{\partial p} = 0.$$

We observe that, in general, the computation of sensitivities requires the knowledge of second-order derivatives of the device modeling equations. This information can be provided by the device models in a way similar to how first-order derivatives are computed for the circuit Jacobian matrix. The implementation of second-order derivative computation can be significantly facilitated through the use of automatic differentiation [17, 18]. In cases where major modifications to model code is impractical, second-order derivatives must be computed by means of less accurate and computationally more expensive finite-difference methods.

In some important special cases, second-order derivative information is easier to obtain. For example, when the parameter affects only the reactive part of the circuit equations, then the operating point is not dependent on the parameter p and thus the first term in (30) and (31), respectively, vanishes. Moreover, when the circuit is linear, then $\mathbf{G}'$ and $\mathbf{C}'$ can be easily obtained using *stencils* or *stamps* [5].

The vectors $\mathbf{b}$ and $\mathbf{l}$ depend in most cases solely on the topology of the circuit, and therefore, their derivatives with respect to circuit parameters are zero. However, sometimes, the parameter p can affect the excitation of a circuit and cause $\mathbf{b}'$ to be nonzero, or an output function may result in a nonzero $\mathbf{l}'$.

Finally, we show how to obtain the vectors $(\mathbf{A}\mathbf{v}_n)'$ and $(\mathbf{A}^T\mathbf{w}_n)'$ in equation (19) of Algorithm 2. Recall from [2] that the vector $\mathbf{u} = \mathbf{A}\mathbf{v}_n$ is obtained by solving the linear system

$$(\mathbf{G} + s_0\mathbf{C})\,\mathbf{u} = -\mathbf{C}\mathbf{v}_n. \tag{32}$$

By differentiating (32), we see that the vector $\mathbf{u}' = (\mathbf{A}\mathbf{v}_n)'$ can be computed by solving the linear system

$$(\mathbf{G} + s_0\mathbf{C})\,\mathbf{u}' = -\mathbf{C}'\mathbf{v}_n - \mathbf{C}\mathbf{v}_n' - (\mathbf{G}' + s_0\mathbf{C}')\,\mathbf{u}. \tag{33}$$

A similar procedure is used to obtain $(\mathbf{A}^T\mathbf{w}_n)'$. Observe that both linear systems (32) and (33) have the same coefficient matrix $\mathbf{G} + s_0\mathbf{C}$, and therefore, no additional LU-factorization is required for sensitivity computations.

5 Numerical Examples

In this section, we present numerical results for two typical examples.

Our first example is the 741 operational amplifier shown in Figure 2. The same example was analyzed in [14]. The magnitude and phase response of the operational amplifier obtained with PVL in $q = 18$ iterations are shown in Figures 3 and 4, respectively. They match with 5 digits of accuracy the response predicted by complex phasor analysis.

Table 1 shows the poles that converged after 18 PVL iterations, together with their normalized sensitivities with respect to the compensation capacitor (30pF). A graphical representation of pole sensitivities is shown in Figure 5.

Figures 6 and 7 show the normalized sensitivities of the magnitude and phase of the amplifier frequency response computed directly with the PVL algorithm. The responses match very well results obtained by perturbation over the entire frequency range. In contrast, the results published in [14] show discrepancies at higher frequencies.

Note that in this example the sensitivity parameter affects only a linear element in the reactive part of the circuit equations, therefore computation of second order derivatives was not required.

Our second example is a simulation of a low-noise amplifier designed for a radio-frequency application and implemented in an advanced BiCMOS process. The netlist, extracted from

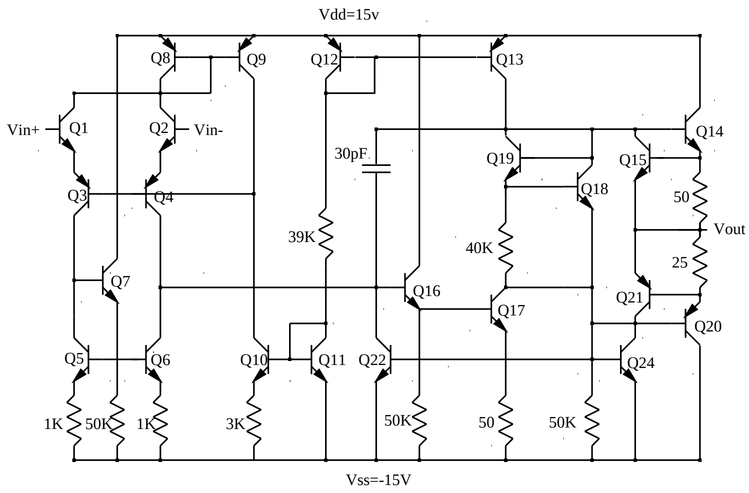

Figure 2: Schema of the 741 Operational Amplifier

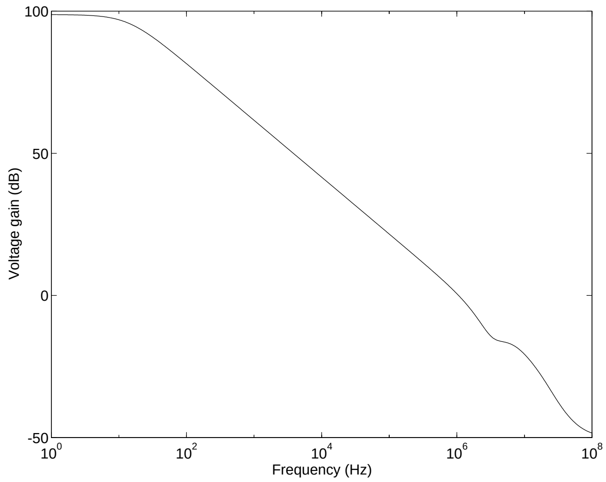

Figure 3: 741 Operational Amplifier magnitude response

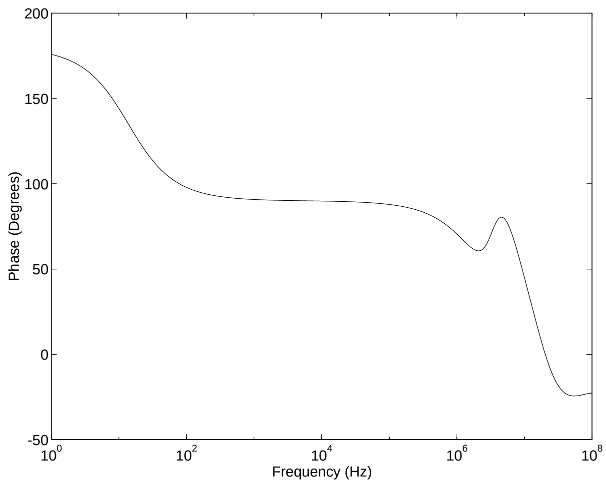

Figure 4: 741 Operational Amplifier phase response

Poles	Pole Sensitivities
$-8.68942 \cdot 10^1$	$8.66 \cdot 10^1$
$-6.46523 \cdot 10^5$	$1.433 \cdot 10^3$
$-2.05993 \cdot 10^7 \pm 6.07465 \cdot 10^6$	$-1.32 \cdot 10^5 \mp 1.861 \cdot 10^5$
$-3.50809 \cdot 10^7 \pm 2.76351 \cdot 10^7$	$6.7 \cdot 10^4 \pm 1.558 \cdot 10^5$
$-7.7302 \cdot 10^7$	$1.323 \cdot 10^5$
$-8.17495 \cdot 10^7$	$9.888 \cdot 10^3$
$-9.58123 \cdot 10^7 \pm 7.59525 \cdot 10^7$	$6.907 \cdot 10^5 \quad \pm 2.67 \cdot 10^7$

Table 1: Poles and Sensitivities for the 741 OpAmp, 18 PVL iterations

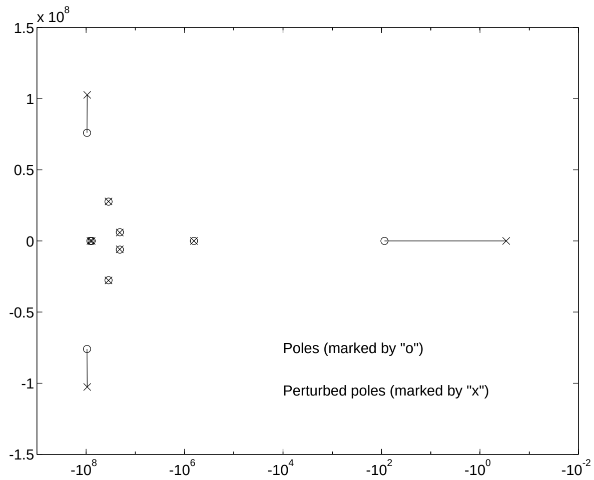

Figure 5: 741 Operational Amplifier pole sensitivities

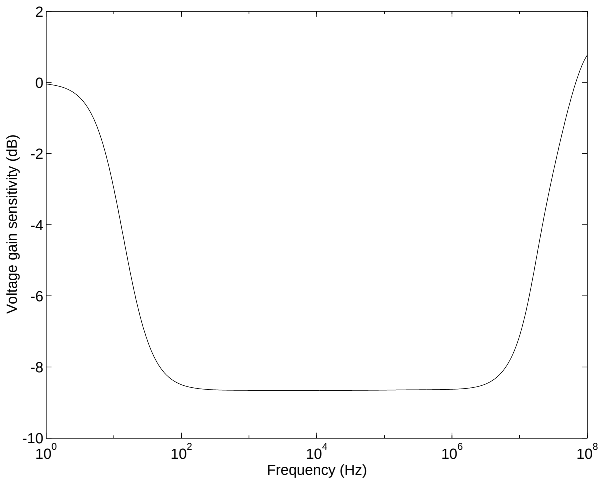

Figure 6: 741 Operational Amplifier magnitude sensitivities

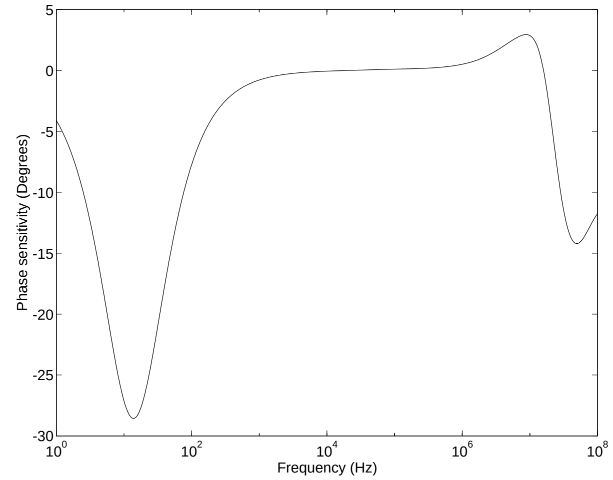

Figure 7: 741 Operational Amplifier phase sensitivities

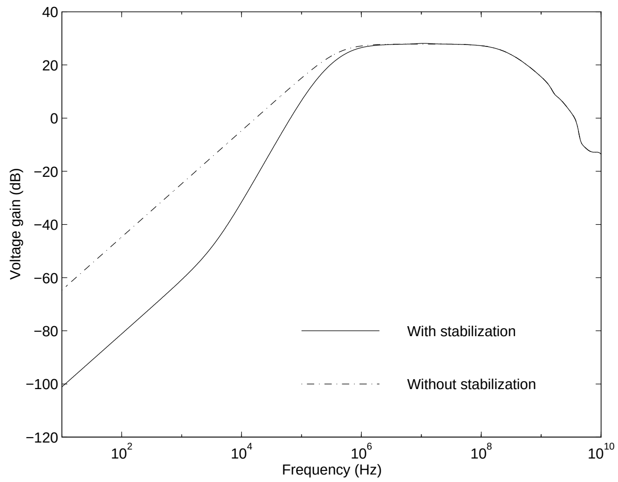

Figure 8: Low-Noise Amplifier magnitude responses

the actual layout with all the parasitics included, consists of 51 MOSFET devices, 26 bipolar transistors, 35 resistors, 6 inductors, and 381 capacitors. The circuit must operate over a range of temperatures and contains a sophisticated temperature-stabilization scheme. Therefore, the sensitivity of the frequency response with respect to temperature is of considerable interest.

We study the frequency-domain characteristics of the amplifier gain—with and without temperature stabilization—and their sensitivities with respect to temperature. The two variants of the circuit were linearized around their DC operating points, and the resulting small-signal equivalent networks were analyzed, using the extended PVL algorithm.

A Padé approximant of order $q = 60$ models correctly the behavior of the amplifier from 10Hz to 10GHz. Figure 8 shows the magnitude of the circuit frequency-response, as produced by the algorithm, with the temperature-stabilization circuitry turned successively on and off. Separately, we compared the Padé approximations to responses computed by complex phasor analysis and they were indistinguishable.

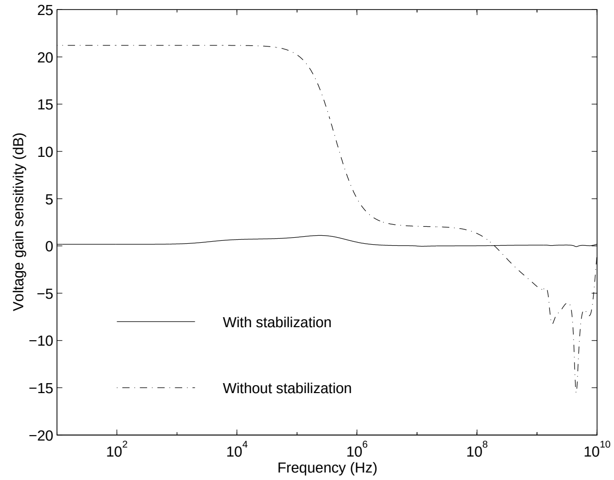

Figure 9: Low-Noise Amplifier magnitude sensitivities

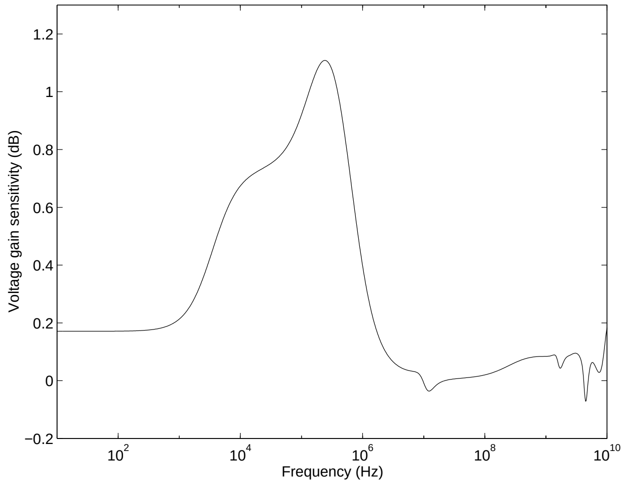

Figure 10: Low-Noise Amplifier magnitude sensitivity (stabilized)

Figure 9 shows the normalized sensitivities (in dB) of the frequency-response magnitudes for the two circuit variants. The plots confirm that the temperature stabilization performs as expected, significantly reducing the sensitivity. Finally, Figure 10 shows only the normalized sensitivity of the temperature-stabilized circuit at a more appropriate scale. Note that in this case the sensitivity must also account for the change in the operating point due to temperature. Since, in our present implementation, we cannot compute second-order derivatives, finite differences were used.

6 Conclusions

This paper argues that, due to its robustness and efficiency, PVL should become the algorithm of choice for the small-signal analysis of 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. In this paper we extend the PVL algorithm to compute the sensitivity of network transfer functions and of its poles and zeros.

References

[1] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," in *Proc. EURO-DAC '94 with EURO-VHDL '94*, Sep. 1994.

[2] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 639–649, May 1995.

[3] 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.

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

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

[6] 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.

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

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

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

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

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

[12] 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.

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

[14] J. Y. Lee, X. Huang, and R. A. Rohrer. "Pole and zero sensitivity calculation in asymptotic waveform evaluation," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 586–597, May 1992.

[15] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Second Edition. New York, N.Y.: Springer-Verlag, 1993.

[16] T. Kato, *A Short Introduction to Perturbation Theory for Linear Operators*. New York, N.Y.: Springer-Verlag, 1982.

[17] A. Griewank and G. F. Corliss (Eds.), *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*. Philadelphia, PA: SIAM, 1991.

[18] P. Feldmann, R. Melville, and S. Moinian, "Automatic differentiation in circuit simulation and device modeling," in *Tech. Dig. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1992.