# Analysis of Interconnect Networks Using Complex Frequency Hopping (CFH)

Eli Chiprout and Michel S. Nakhla, *Senior Member, IEEE*

***Abstract*— With increasing miniaturization and operating speeds, loss of signal integrity due to physical interconnects represents a major performance limiting factor of chip-, board- or system-level design. Moment-matching techniques using Padé approximations have recently been applied to simulating modelled interconnect network that include lossy coupled transmission lines and nonlinear terminations, giving a marked increase in efficiency over traditional simulation techniques. Nevertheless, moment-matching can be inaccurate in high-speed circuits due to critical properties of Padé approximations. Further, moment-generation for transmission line networks can be shown to have increasing numerical truncation error with higher order moments. These inaccuracies are reflected in both the frequency and transient response and there is no criterion for determining the limits of the error. In this paper, a *multipoint* moment-matching, or complex frequency hopping (CFH) technique is introduced which extracts accurate dominant poles of a linear subnetwork up to any predefined maximum frequency. The method generates a single transfer function for a large linear subnetwork and provides for a CPU/accuracy tradeoff. A new algorithm is also introduced for generating higher-order moments for transmission lines without incurring increasing truncation error. Several interconnect examples are considered which demonstrate the accuracy and efficiency in both the time and frequency domains of the new method.**

## I. Introduction

As circuit switching speeds continue to increase, the high frequency effects of previously negligible interconnects have become prominent [1]–[3]. Advances in fabrication methods has made interconnect analysis imperative for diverse technologies such as VLSI chips and packages, multichip models, printed circuit boards and backplanes. Shrinking device sizes coupled with increasing operating speeds are highlighting the effects of ringing, signal delay, distortion, and reflections on single interconnect lines, as well as crosstalk between adjacent lines. These phenomena can, in turn, have undesirable side-effects such as false or delayed switching of digital circuits.

At relatively higher speeds, interconnects can no longer be treated as lumped components. Instead, lossy coupled transmission line models with nonlinear terminations become necessary [4]. Several methods have been proposed to simulate these models accurately [5]–[9]. However, the proposed methods may require extensive CPU time, either in many frequency point analyses, numerical inversion(s), and/or convolution.

Recently, Padé moment-matching methods such as asymptotic waveform evaluation (AWE) [10] and the stepwis equivalent conductance (SWEC) method [11] have been proposed as new techniques to obtain the approximate transient response of interconnect networks, using considerably less CPU than conventional circuit simulators. Moment generation has been applied to networks containing both lumped elements and lossy coupled transmission lines [12], [13], and moment-matching techniques have been refined [14], [15] and applied to interconnects with nonlinear terminations [13], [16], [17], making moment methods useful for delay and crosstalk estimation in interconnect analysis.

Moment-matching approximates a linear subnetwork response from a Taylor series expansion of the frequency response in the complex $s$ plane. In the case of AWE the linear subnetwork can consist of any combination of lumped elements and distributed lossy, coupled transmission lines. The cost of an expansion is approximately one frequency point analysis. The moments (coefficients of the expansion) are matched to a low-order pole/residue transfer function using a rational Padé approximation. This transfer function approximates the true output transfer function which is computationally prohibitive to obtain. The moments are generally taken either from an $s = 0$ expansion (Maclaurin series) or from an $s = \infty$ expansion (Laurent series) or a combination from both [11], [13]. An expansion point along the real axis instead of the origin was also suggested [26] as a way to address stiff systems. We refer to this technique as a single Padé approximation method because a single matrix of moments is set up and solved for the approximate poles.

Single Padé approximations, using moments from the origin or infinity or combined moments from the origin and from infinity have been documented in the mathematical literature [18]–[20] and their properties are reasonably well-known. The approximation extracted will be accurate near the point of expansion in the complex $s$ plane and decrease in accuracy with increased distance from the point. In the case of using combined moments from two expansions, there will be regions in the complex plane between the two expansion points that will be inaccurate in the approximation. This poses a source of unpredictable error both in the frequency and time domain responses as some dominant poles may be approximated

Manuscript received January 24, 1994; revised June 28, 1994. This work was supported through the Computer-aided Engineering Chairs funded by Bell-Northern Research and the Natural Sciences and Engineering Research Council of Canada. This paper was recommended by Associate Editor R. A. Saleh.

The authors are with the Department of Electronics, Carleton University, Ottawa K1S 5B6, Canada.

IEEE Log Number 9404898.

poorly, or remain undetected, if they are relatively distant from a point of expansion. A single Padé approximation is generally limited to, at most, 8–10 accurate poles before generating spurious right-half plane poles for stable networks. Networks that contain transmission line effects such as ringing and crosstalk generally require more than this number for an accurate waveform extraction.

In general, there is no available method of error analysis or an error criteria for a single Padé approximation. An approximation that may be accurate for some networks, might not yield the same accuracy for another network, or for the same network with different frequencies of operation. If two sets of moments are used, it is not immediately clear what is the best combination of moments from each expansion point. More moments from one expansion will influence the approximation to be more accurate near that expansion in the frequency plane.

As an example, the analysis of linear 3-D interconnect models may require hundreds of closely-spaced frequency point analysis in order determine the resonant frequencies accurately [22]. These frequencies show up as sharp, narrow bands in the frequency response, making them difficult to detect with a widely spaced sweep. A single Padé approximation cannot guarantee that all resonant frequencies within a given band will be detected, and will consequently not be able to determine all the resonant frequencies accurately.

A further problem in moment-matching for interconnect networks is that the moments generated for transmission lines rely on the eigenvectors and eigenvalues of the transmission line propagation matrix [12], [13]. The eigenvalues and eigenvectors, which are propagated in the equations recursively, contain a small truncation error, and it is expected and empirically observed that truncation error will increase with the higher order moments due to the recursion. Padé aproximations are sensitive to truncation error in the moments, resulting in inaccurate poles, thereby yielding loss of accuracy in the response. This effect, coupled with the inaccuracy of a Padé approximations in regions of the $s$ plane far from an expansion point can have unpredictable effects on the time-domain accuracy.

In this paper, we introduce complex frequency hopping (CFH), a *multi-point* moment-matching technique. CFH ensures accuracy of an approximation for a complete frequency range, using multiple expansion points ("hops") and corresponding Padé approximations at the frequencies of interest. Additional hops are generated at an incremental CPU cost above the cost of the first. However, the number of hops typically needed ranges from 2–10, far less than full FFT analysis requiring hundreds of frequency point analyses.

CFH provides a new method of generating the exact eigenvalues (poles) of a linear network containing distributed and lumped elements. Eigenvalues are calculated only for the desired region of the complex plane, thereby reducing CPU cost. Accurate poles are generated for the entire frequency range, setting no limits to the number of dominant poles that can be extracted. CFH has enabled the generation of 50 or more accurate dominant poles from circuits whose response required these for an accurate waveform.

The properties of multiple Padé approximations are used to an advantage not possible with a single Padé approximation. The technique's accuracy is enhanced significantly by using a new method of generating the Taylor series moments of a transmission line network, which is introduced in this paper. Individual transmission lines are not approximated. Rather, CFH yields a single transfer function for the entire linear subnetwork consisting of mixed lumped and distributed elements. Accurate frequency or transient analysis is possible without numerical inversion or convolution and the accuracy range can be predetermined.

Four main contributions are included in this paper. First, an efficient binary search algorithm for the selection and minimization of hops in the complex frequency plane is developed. Second, a method for collection of dominant pole/residue information from each expansion point into a unified network transfer function is presented. This yields an error checking mechanism for the accuracy of pole/residue pairs not available in a single Padé approximation. Only the actual dominant poles are generated, not approximate poles. Consequently, the instability properties of Padé approximations [23] are bypassed and no unstable right-half plane poles result when applied to stable systems. Third, a tradeoff in accuracy versus CPU cost is provided. CFH can yield a low-frequency range accuracy at low CPU cost, or a larger frequency range accuracy at higher CPU cost. In the case of a low-frequency range, generally only two expansion points are necessary. In the case of a larger frequency sweep, more expansion points are used but always much less than the hundreds of expansions necessary for a full frequency analysis.

Finally, in order to apply CFH accurately, a new method for generating the moments of a set of lossy coupled transmission lines is introduced. The new moment generation method does not rely on the eigenmodes of the lines and therefore causes no increasing truncation error in the moments. The new method may be used either with AWE or with CFH.

In Section II of this paper we will review the concepts behind moment-matching, specifically as applied in AWE, as well as some of the properties of Padé approximations that will be useful in the method being introduced. In Section III we will introduce the algorithms used in complex frequency hopping and explain how they address the major concerns of moment-matching. In Section IV a new method for generating the moments of a transmission line network will be introduced enabling the accurate application of CFH. In Section V we will analyze the CPU expense required to carry out CFH algorithms and how they compare with other methods. We will illustrate the algorithm in Section VI using linear and nonlinear distributed interconnect examples and finally, in Section VI, we will draw conclusions and discuss future work.

## II. Overview of Moment Generation and Moment-Matching

Moment techniques in AWE generally consist of two main stages:

- Moment generation
- Moment matching

Moment-matching uses the coefficients (moments) of the expansion of a particular system output transfer function, $H(s)$, about a point in the complex $s$-plane. The moments can either be generated by expanding $H(s)$ in a Taylor series,

$$H(s) = m_0 + m_1 s + m_2 s^2 + \cdots \tag{1}$$

or in a Laurent series,

$$H(s) = m_{-1}s^{-1} m_{-2}s^{-2} + m_{-3}s^{-3} + \cdots. \tag{2}$$

Once the moments have been generated, they are matched using a Padé approximation to a low-order set of poles and residues. So while the original network may have a very large or even an infinite number of poles $N_p$, a smaller set of $q \ll N_p$ poles is used in the approximation.

### A. Moment Matching

Given input source(s) and a specified output $i$, the actual frequency response of output $i$ of a linear network can be described as

$$H_i(s) = c_i + \sum_{j=1}^{N_p} \frac{k_j}{s - p_j} \tag{3}$$

where the $p_j$ are the complex $N_p$ poles (eigenvalues) of the network and the $k_j$ the corresponding complex residues of the output. A direct coupling constant, $c_i$, between the inut and the output is oftentimes zero.

Using a Padé approximation, an approximate transfer function can be obtained,

$$H_i(s) \approx \frac{\hat{P}_L(s)}{\hat{Q}_M(s)} \tag{4}$$

where $\hat{P}_L(s) = a_0 + a_1 s + a_2 s^2 + \cdots + a_L s^L$ and $\hat{Q}_M(s) = b_0 + b_1 s + b_2 s^2 + \cdots + b_M s^M$. The denominator is derived from the moments by using,

$$\begin{bmatrix} m_{L-M+1} & m_{L-M+2} & \cdots & m_L \\ m_{L-M+2} & m_{L-M+3} & \cdots & m_{L+1} \\ \vdots & \vdots & & \vdots \\ m_L & m_{L+1} & \cdots & m_{L+M-1} \end{bmatrix} \begin{bmatrix} b_M \\ b_{M-1} \\ \vdots \\ b_1 \end{bmatrix} = - \begin{bmatrix} m_{L+1} \\ m_{L+2} \\ \vdots \\ m_{L+M} \end{bmatrix} \tag{5}$$

where $b_0 = 1$. The entire array of approximations that are possible with a different choice of $L$ and $M$ is known as the Padé table, and a particular approximation as an $[L/M]$ Padé approximation. $L$ and the $M$ must be carefully selected, but generally for a single Padé approximation a choice of $L = M - 1$ or $L = M$ is used. However the Padé table has other properties that make it useful.

An important property of Padé approximation is the convergence of the poles of the horizontal sequence (increasing $L$, fixed $M$) to the actual network poles nearest the point of expansion [18], [26], [24], [14]. Given a denominator polynomial $\hat{Q}_{[L/M]}(s) \equiv \hat{Q}_M(s)$ obtained from an $[L/M]$ Padé approximation,

*Theorem 1 (Horizontal sequence):*

$$\lim_{L\to\infty} [\hat{Q}_{[L/M]}(s)] \to \prod_{j=1}^{M} \left(1 - \frac{s}{p_j}\right) \tag{6}$$

where $p_j$ are the actual poles of the network arranged in order of increasing modulus: $|p_1| \le |p_2| \le \cdots \le |p_M| \le \cdots$ and $|p_M| \ne |p_{M+1}|$.

This theorem is useful in extracting accurate poles (eigenvalues) in the immediate neighborhood of an expansion point as will be demonstrated in Section III.

As is shown in [10] rather than obtaining the numerator directly, it is more convenient to generate a transient response estimation in terms of $q$ poles (in this case $q = M$), in the form of,

$$H_{i(s)} \approx \hat{c} + \sum_{i=1}^{q} \frac{\hat{k}_i}{s - \hat{p}_i}. \tag{7}$$

The advantage is that the domain impulse response is given in closed form by,

$$h_i(t) \approx \hat{c}_i \delta(t) + \sum_{j=1}^{q} \hat{k}_j e^{\hat{p}_j t}. \tag{8}$$

Similar closed forms are possible for a step response or ramp input. The $\hat{p}_j$ can be obtained from (5) by applying a root solving algorithm on the resulting $\hat{Q}_M$. Another matrix equation is used to calculate the residues $\hat{k}_j$. Once obtained, the poles/residues can be used in linear or nonlinear analysis [17].

A Taylor series expansion of a function has a radius of convergence $R$ beyond which the series does not yield a finite result. In a linear network $R$ is directly related to the pole of smallest modulus

$$R = \min(|p_j|) \quad (1 < j < N_p). \tag{9}$$

A Taylor series gives less information than the original function since the latter can be evaluated at all points in the complex plane except singularities. A Padé approximation however, extends information of the Taylor series beyond its radius of convergence at the nearest pole(s). Nevertheless, as with the case of a Taylor series, the accuracy of a Padé approximation decreases with increasing distance from the point of expansion. Therefore poles close to the point of expansion will be generated accurately while those farther away will be approximated poorly or remain completely undetected.

Poles that dominate the response of a network are relatively close to the imaginary axis and have large residues. These have the largest influence both on the frequency and time domain

responses. Poles that are extremely close to the imaginary axis cause sharp peaked resonant frequencies. This can be seen by inspecting (3). At frequencies $s = j\omega_{\text{res}}$, closest to a pole $\alpha + j\omega_{\text{res}}$ which is very near the imaginary axis, one of the denominator terms will become small making the sum large, yielding a large response.

If an approximation fails to obtain some or all of the dominant system poles accurately, the inaccuracy will be reflected in the frequency and transient responses. In order to ensure accuracy in the general case, any moment-method must extract the actual dominant poles of a network. This is the approach used in CFH. The set of dominant poles is generally much smaller than the entire set of system poles. The net result is that the simulation is accurate at all frequencies between dc and a pre-determined highest frequency.

Moment matching relies directly on the generated moments of a linear network. This method is described next.

### B. Moment Generation

An efficient algorithm for generating the moments for an RLC network was first proposed in [10]. The method was extended to networks that include distributed lossy coupled transmission lines in [12] or alternatively in [13]. We describe the general method briefly and refer the reader to more details in those references.

The modified nodal admittance (MNA)[25] network equations for a linear distributed transmission line network can then be written in simplified form as,

$$\mathbf{Y}(s)\mathbf{X}(s) = \mathbf{E} \tag{10}$$

where $\mathbf{Y}(s)$ is a matrix containing the set of modified nodal admittance stamps of each of the lumped or distributed elements, $\mathbf{X}(s)$ is the set of unknown network output variables, and $\mathbf{E}$ the vector of input sources. Solving for $\mathbf{X}(s)$,

$$\mathbf{X}(s) = [\mathbf{Y}^{-1}(s)]\mathbf{E}. \tag{11}$$

In order to obtain the moments of the system about $s$ =0, (11) is expanded in a Taylor series about the origin.

$$\mathbf{X}(s) = \sum_{n=0}^{\infty} M_n s^n \tag{12}$$

where,

$$\mathbf{M}_n = \frac{\left[\frac{\partial^n}{\partial_s{}^n}\mathbf{Y}^{-1}(s)\right]_{s=0}}{n!}\mathbf{E} \tag{13}$$

are the vector of moments for all the variables in $\mathbf{X}$. For an individual output variable $i$ the $i$-th row is selected giving the Taylor series of the $i$-th transfer function,

$$H_i(s) = \frac{P_L(s)}{Q_M(s)} = m_0 + m_1 s + m_2 s^2 + m_3 s^3 + \cdots. \tag{14}$$

The derivatives of $\mathbf{Y}^{-1}(s)$ are difficult to evaluate but a recursive relationship for $\mathbf{M}_n$ is possible in terms of the derivatives of $\mathbf{Y}(s)$. The relationship is given by,

$$[\mathbf{Y}(s)]_{s=0}\mathbf{M}_n = -\sum_{r=1}^{n} \frac{\left[\frac{\partial^r}{\partial_s r}\mathbf{Y}(s)\right]_{s=0}\mathbf{M}_{n-r}}{r!} \tag{15}$$

with,

$$[\mathbf{Y}(s)]_{s=0}\mathbf{M}_0 = \mathbf{E}. \tag{16}$$

In the case of a lumped element network $\pi$, the system matrix of $\pi$, $\mathbf{Y}_\pi(s)$, can be formulated in terms of two matrices,

$$\mathbf{Y}_\pi(s) = \mathbf{G}_\pi + s\mathbf{C}_\pi \tag{17}$$

giving $\mathbf{Y}_\pi(0) = \mathbf{G}_\pi$ and $\mathbf{Y}'_\pi(0) = \mathbf{C}_\pi$, so (15) and (16) become a simple,

$$\mathbf{G}_\pi\mathbf{M}_n = -\mathbf{C}_\pi\mathbf{M}_{n-1} \tag{18}$$

with,

$$\mathbf{G}_\pi\mathbf{M}_0 = \mathbf{E}. \tag{19}$$

When a network conaining lossy coupled transmission lines, the $s$-domain port formulations of the lines form stamps in the $\mathbf{Y}(s)$ matrix. Derivatives of these stamps will yield derivatives of the matrix. The stamps are derived from the Telegrapher's equations for the lines,

$$\frac{\partial}{\partial x}\mathbf{V}(x,s) = -(\mathbf{R} + s\mathbf{L})\mathbf{I}(x,s) = \mathbf{Z}_p(s)\mathbf{I}(x,s) \tag{20}$$

$$\frac{\partial}{\partial x}\mathbf{I}(x,s) = -(\mathbf{G} + s\mathbf{C})\mathbf{V}(x,s) = \mathbf{Y}_p(s)\mathbf{V}(x,s) \tag{21}$$

where $\mathbf{V}(x,s)$ are the line voltages as a function of position on the line, $\mathbf{I}(x,s)$ are the line currents, and $\mathbf{R}, \mathbf{G}, \mathbf{L}, \mathbf{C}$ are the per-unit-length resistive, conductive, inductive and capacitive line parameter matrices, respectively. $\mathbf{Z}_p(s)$ and $\mathbf{Y}_p(s)$ are then the per-unit-length impedance and admittance parameters.

In order to obtain the moments of $\mathbf{V}(s)$ and $\mathbf{I}(s)$ [12] and [13] proposed that the equations be solved in terms of the eigenmodes of the lines. One method may be to solve for (20) and (21) in terms of the eigenvectors and eigenvalues of $\mathbf{Z}_p(s)\mathbf{Y}_p(s)$ and calculate the derivatives with respect to $s$. Alternatively, one can solve the equations for the moments with respect to $s$ using the eigenvalues of the resistive parameters. Both methods yield identical results theoretically and numerically. Both these methods recursively propagate the eigenmodes of the transmission lines which may contain a small truncation error. Any truncation error however will be compounded recursively. This effect will not be significant for low orders of approximations, but at high orders it becomes critical. Further, Padé approximation are very sensitive to even the smallest error in the moments and any truncation error should be minimized. In this paper we introduce a techniqe that elminates this source of error.

## III. Complex Frequency Hopping (CFH)

We note that in the general case, it is possible to "hop" to any point in the complex $s$-plane and expand in a Taylor series there. This is the same a real frequency shift in the complex plane which was first suggested in [26] as a way to ensure greater accuracy in stiff systems. Additional hops will ensure that more information can be obtained in the neighborhood of each of the hops. However, hops not on the real axis or at $s = \infty$ result in complex moments and the calculations must be carried with complex numbers. Further, it is not clear how to choose other expansion points nor how to combine information from two or more hops into a unified whole.

Complex frequency hopping resolves these issues by utilizing the properties of Padé approximations. Poles in the neighborhood of an expansion are extracted by horizontal convergence of the Padé table. An algorithm is provided for incorporating information from each hop into a unified transfer function for the entire network. An efficient binary search strategy is incorporated for the selection and minimization of expansion points.

Choosing any arbitrary complex frequency point, $p = \alpha + j\beta$, a Taylor expansion of about $p$ is given by,

$$H_i(s) = m_0' + m_1'(s-p) + m_2'(s-p)^2 + \cdots. \tag{22}$$

If we substitute $s' = s - p$ then,

$$H_i(s') = m_0' + m_1's' + m_2's'^2 + \cdots \tag{23}$$

which is the Maclaurin expansion about $s' = 0$ or $s = p$. In other words, in order to use Padé approximations at another expansion point, the system function must be shifted to the new point and treated as an expansion about the origin. This moment-shifting technique has been applied to a real expansion point in [26] by expanding at $s = \alpha$ in order to replace and improve a Padé approximation about the origin.

If the network consists of lumped element components formulated in the MNA formulation, the system moments about $s = 0$ may be derived by (18) and (19).

In the case of an arbitrary expansion about a complex point, $p = \alpha + j\beta$, the moments for a lumped network are shifted by $p$ and are calculated using,

$$\mathbf{M_0'} = -(\mathbf{G}_\pi + p\mathbf{C}_\pi)^{-1}\mathbf{C}_\pi \tag{24}$$

$$(\mathbf{G}_\pi + p\mathbf{C}_\pi)\mathbf{M}_{i+1}' = -\mathbf{C}_\pi\mathbf{M}_i'. \tag{25}$$

Since the moments generated will be complex, all of the calculations including Padé matrix solutions in (5) and root finding to extract the poles, must be done using complex numbers.

In the case of lossy, coupled transmission lines, the generation of moments could be done by using the methods in [12] or in [13] and by shifting the point of expansion. The characteristic per-unit-length line impedance matrix $\mathbf{Z}_p = \mathbf{R} + s\mathbf{L}$ would be replaced by $\mathbf{Z}_p' = (\mathbf{R} + p\mathbf{L}) + s\mathbf{L}$ and $\mathbf{Y}_p = \mathbf{G} + s\mathbf{C}$ by $\mathbf{Y}_p' = (\mathbf{G} + p\mathbf{C}) + s\mathbf{C}$. However, truncation error in these methods make them difficult to use in Padé table convergence in (6). An alternative method is therefore proposed in Section IV.

A complex frequency hop is used to detect poles in a region of the complex plane not visible from another expansion point. If one expansion point is near enough to another, some of the same poles will be detected thereby giving confirmation of the accuracy of pole/residue pairs. This is an error check not available in moment-matching methods using one Padé approximation, and which contains only approximate poles.

In some circuits, two complex frequency hops or conventional AWE are sufficient for the estimated waveform to converge to the real waveform. CFH, however, provides a guarantee of the desired accuracy. As the circuit operating frequency is increased and transmission line effects become prominent, more hops are needed to accurately describe the waveform and the CPU cost is proportional to this complexity. CFH therefore provides an accuracy/CPU trade-off not available by one Padé approximation. Convergence to the actual dominant poles is always guaranteed within the frequency of interest.

### A. Pole Convergence

The strategy of CFH is to target and generate the actual dominant poles of the circuit rather than generating approximate poles. When one generates an approximating *function* to the original transfer function as in (4), a diagonal or near diagonal approximation is used. However, the poles generated from such an approximation will not necessarily reflect the actual system poles. To generate the actual system poles accurately, one needs to use (6). This will allow convergence to the actual system poles. $M$ is chosen large enough so that the sequence converges quickly and the number of moments needed is not excessive. A larger $M$ will means faster convergence but may mean that the matrix in (5) will be singular because $M$ poles may not be numerically prominent at that hop. It is necessary therefore to test the matrix for ill-conditioning. A smaller $M$ will mean that less poles are detected at the hop. Experimentally and typically we have found that $M = 7$ is a good choice and approximately 20–30 moments are used for convergence.

Using this method, the poles generated about any hop are accurate with respect to the actual poles in the neighborhood of the expansion. The closer the point of expansion to a pole the more accurate the extracted pole. Nevertheless, in practice, some numerical truncation error affects the convergence of (6). As a result some poles far from the point of expansion do not converge well and a further series of steps are needed to ensure the accuracy of the poles. We describe this first in Section III-B.

As well, a search strategy necessary both to select the location of the hops and minimize their number is covered in Section III-C. Since Padé approximations rely on the high accuracy of the moments, a new method of generating the moments of lossy, coupled transmission line networks will be introduced in Section III-D.

We note that CFH gives a general technique for calculating the dominant eigenvalues of a lumped or distributed linear network. This is significant that the poles of distributed systems are transcendental and infinite in number and there is no general algorithm for calculating them. Generally, known

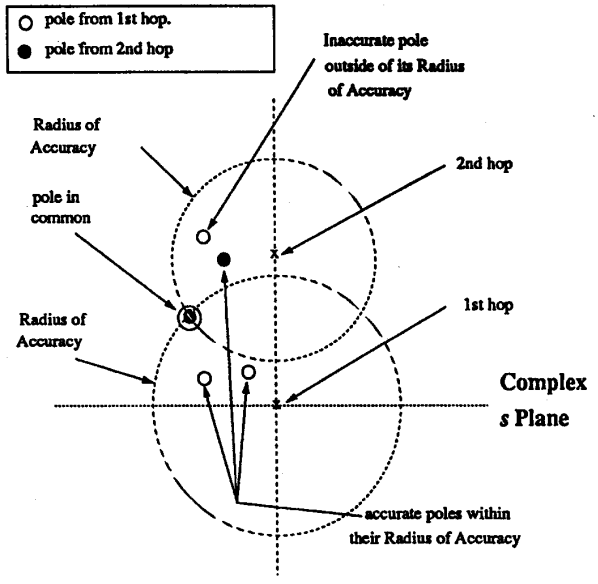



Fig. 1. Pole selection algorithm for two hops.

algorithms, such as the QR algorithm or the iteration (power) method with shifting [27] are used for lumped networks. The QR algorithm is useful for calculating all the eigenvalues of a network and is practical for only relatively small networks. CFH generates only the dominant eigenvalues within a given search area and is therefore far less CPU intensive. The iteration method with shifting generates one largest (smallest) eigenvalue at a given matrix decomposition. This makes it expensive for generating a set of eigenvalues. It also suffers from the weaknesses of not having a method for choosing the next shift, and not converging to two poles of the same modulus such as complex conjugate poles. CFH suffers no such problems.

### B. Pole Selection

With two or more complex frequency hops, pole information from each is combined into a set of poles and residues once convergence of each is achieved. The algorithm is based on a property of Padé approximation in which the accuracy of an approximation is good near the point of expansion but decreases with increasing distance from the point. This result has been noted from numerous empirical tests in the literature as well as by other researchers and our own tests. This property is a natural result of the fact that the moments of (3) is given by,

$$H_i(s) = \left(c_i + \sum_{j=1}^{N_p} \frac{k_j}{p_j}\right) + s\left(\sum_{j=1}^{N_p} \frac{k_j}{p_j^2}\right) + s^2\left(\sum_{j=1}^{N_p} \frac{k_j}{p_j^3}\right) + \cdots \tag{26}$$

Poles with the smallest modulus (nearest the point of expansion) will have less truncation error than poles with larger modulus (farther from the point of expansion), assuming same order of magnitude of residues.

If a pole generated in an approximation is confirmed to be accurate then all poles *closer* to the point of expansion are also tagged as accurate. The outermost pole known to be accurate consequently defines a radius of accuracy $R_{\text{accuracy}}$ for an error tolerance $\varepsilon_{\text{err}}$. All poles within the radius of accuracy are marked as accurate. Poles are initially marked as accurate when they are confirmed by at least two different hops to within $\varepsilon_{\text{err}}$. If the same pole is detected in two hops, then that pole is confirmed accurate and its distance from the point of expansion defines $R_{\text{accuracy}}$.

The following steps are taken to extract a set of accurate poles:

**Step 1:** Poles from each hop are converged using (6).

**Step 2:** Residues from each hop are also extracted from the moments. This requires using the equation,

$$\begin{bmatrix} \hat{p}_1^{-L-2} & \hat{p}_2^{-L-2} & \cdots & \hat{p}_q^{-L-2} \\ \hat{p}_1^{-L-3} & \hat{p}_2^{-L-3} & \cdots & \hat{p}_q^{-L-3} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{p}_1^{-L-M-1} & \hat{p}_2^{-L-M-1} & \cdots & \hat{p}_q^{-L-M-1} \end{bmatrix} \begin{bmatrix} \hat{k}_1 \\ \hat{k}_2 \\ \vdots \\ \hat{k}_q \end{bmatrix} = -\begin{bmatrix} m_{L+1} \\ m_{L+2} \\ \vdots \\ m_{L+M} \end{bmatrix}. \tag{27}$$

Equation (27) differs from the one used in [10]. It derives the actual residues of the (converged) poles rather than approximate residues.

**Step 3:** If the same poles (with the error tolerance $\varepsilon_{\text{err}}$) are detected in two different expansions, they are marked as accurate.

**Step 4:** The distance between a hop and its farthest confirmed accurate pole defines a radius of accuracy $R_{\text{accuracy}}$. All poles within $R_{\text{accuracy}}$ are also marked accurate.

**Step 5:** Residues of marked accurate poles are also marked accurate.

**Step 6:** Poles/residues not marked by steps above are rejected. There is no confirmation of their accuracy.

Assuming that the moments about two expansion points $a$ and $b$ have been generated, and that (6) has been used to collect a set of $q_a$ poles $\{p_i\}$ from hop $a$ and $q_b$ poles $\{p_j\}$ from hop $b$, as well as their respective residues $\{k_i\}$ and $\{k_j\}$ the algorithm for collecting poles/residues at hop $a$ is summarized as follows:

*Pole Selection Algorithm:*

**Un-Mark** all $p_i^{(a)}$ and $k_i^{(b)}$;
*if* $(\exists i, j$ such that $|p_i^{(a)} - p_j^{(a)}| < \varepsilon_{\text{err}})$
  $R_{\text{acc}}^{(a)} \leftarrow \max(|p_i^{(a)}|)$ for which $|p_i^{(a)} - p_j^{(b)}| < \varepsilon_{\text{err}}$
  for some $j$;
*end if for* $(i \leq q_a)$
  *if* $(|p_i^{(a)}| \leq R_{\text{acc}}^{(a)})$
    $p_i^{(a)} \leftarrow$ **accurate**
    $k_i^{(a)} \leftarrow$ **accurate**
  *end if*
*end for*

The pole selection algorithm is illustrated in Fig. 1.

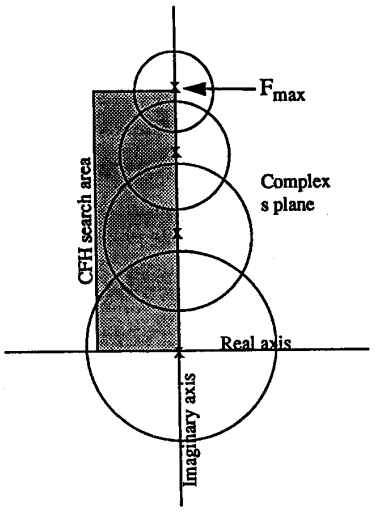



Fig. 2. Binary search and CFH search area in the complex $s$ plane.

### C. Search Strategy

Given a high enough uniform density of expansion points, CFH is guaranteed to find all the dominant network poles within a given search area. However, the CPU expense of such a strategy will be high. Any search strategy must therefore aim to extract accurate pole/residue information with a minimized number of hops. Several search strategies are possible.

A search strategy must be subject to several constraints if it is to be efficient:

**Constraint 1:** *The search should be confined to the upper left half complex plane.* Stable systems with interconnects contain only complex conjugate poles in the left half plane. For such systems a search strategy should be confined to one of the left hand quadrants (the upper one is chosen arbitrarily). Poles found here can be duplicated for the conjugate quadrant.

**Constraint 2:** *The search should be confined to or near the imaginary axis.* Poles are critical to the transient and frequency responses if they are close to the imaginary axis. In the time domain the influence of poles relatively far into the left hand plane is very transient. In the frequency domain poles far from the $j\omega$ axis do not have much influence on the response in (3).

**Constraint 3:** *The search should be confined to points below the highest frequency* $f_{\max}$. The expected operating frequency of the circuit, which depends on the input waveform and the type of circuit, limits the search up the frequency axis. Only information up to a limit operating frequency is needed. Poles higher up the imaginary axis will not be important. The $f_{\max}$ of a circuit will depend on the input ramp in the case of linear circuits [28], or on the type of circuit in the case of nonlinear circuits.

The above three constraints describe a CFH search area and drastically reduce the search space as illustrated in Fig 2. Using the above a search area an algorithm is developed which uses a binary search strategy.

**Step 1:** Expand at the origin ($s = 0$). Collect poles by convergence of (6).

**Step 2:** Determine the highest frequency of accuracy desired, $s = j\omega_{\max} = 2\pi f_{\max}$, and expand at that point. Collect poles by convergence.

**Step 3:** If common poles exist between $s = 0$ and $s = j\omega_{\max}$, STOP. Define a radius of accuracy for each expansion and collect poles and residues between the two points. Else, expand at the midpoint $s = j\omega_{\max}/2$ between the two end points.

**Step 4:** Continue the search in similar binary fashion between $s = 0$ and $s = j\omega_{\max}/2$ and between $s = j\omega_{\max}/2$ and $s = j\omega_{\max}$ defining and expanding at midpoints if no common poles are found. The algorithm is summarized below.

*Binary Search Algorithm:*

```
Orig Hop ← 0
Expand at OrigHop;
Converge to poles {p}^(OrigHop) in the neighborhood of
OrigHop;
ω_max ← highest frequency of interest;
EndHop ← jω_max
PreviousHop(EndHop) ← OrigHop;
Expand at EndHop;
Converge to poles {p}^(EndHop);
CommonPoleWithPreviousHop(EndHop) ← NO
SearchFinished ← NO
while (Not SearchFinished)
    SearchFinished ← YES
    SearchHop ← EndHop
    while (SearchHop ! = OrigHop)
        if (CommonPoleWithPreviousHop
          (SearchHop)= YES)
          SearchHop ← PreviousHop(SearchHop)
        else if (∃i, j such that p_i ∈ {p}^(SearchHop)
              p_j ∈ {p}^(PreviousHop(SearchHop))
              such that |p_i = p_j| < ε_err)
              CommonPoleWithPreviousHop
              (SearchHop) ← YES);
              SearchHop ← PreviousHop(SearchHop)
        else
              MidHop ← Midpoint
              (SearchHop,PreviousHop(SearchHop))
              PreviousHop(MidHop)
              = PreviousHop(SearchHop)
              PreviousHop(SearchHop) = MidHop
              Expand at MidHop;
              Converge to poles {p}^(MidHop);
              SearchFinished ← NO
              SearchHop ← EndHop
        end if
    end while
end while
```

Each hop must be near enough to the neighboring hop so as to confirm the same outer poles between the two expansions. Generally, in most low frequency circuits, the search bottoms out after two expansions. In some difficult

examples, of electrically long transmission lines with associated long delays, crosstalk and signal reflections, the search will need more hops. The CPU cost will be proportional to the ringing (number of dominant poles) in the circuit but the CFH method guarantees an accurate waveform within the specified frequency range. The moments generated for any circuit must be accurate for this type of search. Any small truncation error in the moments will mean that poles far from a point of expansion will not be accurate, necessitating more hops and increasing the CPU cost.

The binary search strategy and search area is illustrated in Fig 2.

### D. Nonlinear Analysis

Since CFH generates a single transfer function of a linear network in terms of the exact dominant poles and residues, it is possible to create a macromodel of the linear network in terms of lumped components. The method, first proposed in context of AWE [17], matched a network to approximate poles and as such was limited in its accuracy. In CFH however, the accuracy is greatly enhanced because exact poles are generated and the network matched up to any desired frequency. Essentially, the lumped component network will have the same dominant poles and residues as the dominant poles and residues of the linear subnetwork. This is a unique matching of eigenvalues of the networks. The resultant macromodel can then be combined with any nonlinear terminations such as drivers or receivers.

Because CFH generates the exact dominant poles at every output node, and because the poles are unique for the system, CFH generates the same poles for all output nodes analyzed. The only exception is if a residue at a particular output is negligible so as to make the pole undetectable. This is important, because when using a single Padé approximation, the choice of nodes becomes critical [14], [29].

In the case of nonlinear analysis, it is important that the linear subnetwork be accurately represented up to high frequencies, for while the input signal may be of low frequency, the output may contain high frequencies due to the nonlinear frequency spread. CFH can be used to create a macro-model for the linear subnetwork up to the highest operating frequency of the circuit. Once the linear network macromodel is created, it can then be simulated with the nonlinear terminations in SPICE [30] or outside of the context of SPICE [16]. In either case no numerical inversion or convolution is required.

## IV. Matrix Exponential Moments

As noted previously, published methods of generating moments for a transmission line system results in unwanted truncation error which increases with each new moment generated. This is due to the truncation error in the eigenmode solutions proposed in [12] and [13]. This can be verified in two ways. First, moments generated using eigenvalue methods do not converge as in (6). This is a direct result of increasing truncation error in the higher order moments. Second, accurate moments of a transmission line may be generated using extended precision arithmetic [31]. Extended precision arithmetic is a software simulation of high precision computers. While extremely inefficient, it allows the elimination of truncation errors up to any desired degree of accuracy. Extended precision programming done by us has also demonstrated that eigenvalue methods yield increasing truncation error in the moments.

CFH relies on high accuracy moments, including moments of high order, so that convergence of (6) can be achieved at each hop. For this purpose a special more accurate method of generating the moments is developed here.

Equations (20) and (21) can be reformulated as,

$$\frac{\partial}{\partial x}\Phi(x,s) = (\mathbf{A} + s\mathbf{B})\Phi(x,s) \tag{28}$$

where,

$$\Phi(x,s) = \begin{bmatrix} \mathbf{V}(s) \\ \mathbf{I}(s) \end{bmatrix} \tag{29}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & -\mathbf{R} \\ -\mathbf{G} & \mathbf{0} \end{bmatrix} \tag{30}$$

and,

$$\mathbf{B} = \begin{bmatrix} \mathbf{0} & -\mathbf{L} \\ -\mathbf{C} & \mathbf{0} \end{bmatrix} \tag{31}$$

with $\mathbf{0}$ an $n \times n$ matrix of zeros. The solution for (28) in terms of $x$ is a matrix exponential [32],

$$\Phi(x,s) = e^{(\mathbf{A}+s\mathbf{B})x}\Phi(0,s). \tag{32}$$

The far end voltages and currents are described as $\Phi(d,s)$ where $d$ is the length of the line. Therefore, in order to solve the far end voltages and currents in terms of the near end voltages and currents, we get,

$$\Phi(d,s) = e^{(\mathbf{A}+s\mathbf{B})d}\Phi(0,s). \tag{33}$$

An important property of matrix exponentiation is that while it parallels scalar exponentiation in most ways, it does not follow the normal exponentiation rule. That is,

$$(A \cdot B \neq B \cdot A) \leftrightarrow \left(e^{\mathbf{A}}e^{\mathbf{B}} \neq e^{\mathbf{A}+\mathbf{B}}\right). \tag{34}$$

The matrices must commute if they are to follow exponent addition. So in the general case of transmission parameters,

$$e^{(\mathbf{A}+s\mathbf{B})d} \neq e^{\mathbf{A}d}e^{s\mathbf{B}d} \tag{35}$$

$e^{(\mathbf{A}+s\mathbf{B})d}$ represents a complex function of $s$ even though it appears simple. It is the frequency domain stamp that is entered into the $\mathbf{Y}(s)$ matrix.

In order to get a Taylor expansion of the stamp we assume a series in the form,

$$e^{(\mathbf{A}+s\mathbf{B})d} = \mathbf{C}_0 + s\mathbf{C}_1 + s^2\mathbf{C}_2 + \cdots. \tag{36}$$

By property of matrix functions,

$$e^{\mathbf{A}} = \mathbf{U} + \mathbf{A} + \frac{1}{2!}\mathbf{A}^2 + \cdots + \frac{1}{n!}\mathbf{A}^n + \cdots. \quad (37)$$

Expanding (36) in an exponential series,

$$e^{(\mathbf{A}+s\mathbf{B})d} = \mathbf{U} + (\mathbf{A}+s\mathbf{B})d + \frac{(\mathbf{A}+s\mathbf{B})^2 d^2}{2!} + \cdots + \frac{(\mathbf{A}+s\mathbf{B})^n d^n}{n!} + \cdots. \quad (38)$$

Further expanding (38) and collecting terms we obtain,

$$\begin{aligned} e^{(\mathbf{A}+s\mathbf{B})d} = &\left[\mathbf{U} + \mathbf{A}d + \frac{1}{2!}\mathbf{A}^2 d^2 + \cdots\right] \\ &+ s\left[\mathbf{B}d + \frac{1}{2!}(\mathbf{AB}+\mathbf{BA})d^2 \right. \\ &\left. + \frac{1}{3!}(\mathbf{A}^2\mathbf{B} + \mathbf{ABA} + \mathbf{BA}^2)d^3 + \cdots\right] \\ &+ s^2\left[\frac{1}{2!}(\mathbf{B}^2)d^2 + \frac{1}{3!}(\mathbf{AB}^3 + \mathbf{BAB}^2 \right. \\ &\left. + \mathbf{B}^2\mathbf{AB} + \mathbf{B}^3\mathbf{A})d^3 + \cdots\right] + \cdots. \end{aligned} \quad (39)$$

Equating coefficients of (36) and (39) we obtain a compact expression for the moments in the form,

$$\mathbf{C}_i = \sum_{j=0}^{\infty} \mathbf{C}_{i,j} \quad (40)$$

where the $\mathbf{C}_{i,j}$ are given recursively by,

$$\mathbf{C}_{0,j} = \frac{\mathbf{A}\mathbf{C}_{0,j-1}d}{j} \quad (j > 1) \quad (41)$$

$$\mathbf{C}_{i,j} = \frac{(\mathbf{A}\mathbf{C}_{i,j-1} + \mathbf{B}\mathbf{C}_{i-1,j})d}{i+j} \quad (i > 1, j \geq 0) \quad (42)$$

with,

$$\mathbf{C}_{0,0} = \mathbf{U} \quad (43)$$

$$\mathbf{C}_{i,-1} = \mathbf{0}. \quad (44)$$

Using a series expansion for a matrix exponential can cause truncation error because the $\mathbf{A}^n$ can, for the first terms, grow large before the $n!$ can reduce it in (37) [32]. This problem can also result in slow convergence because $n$ must be large for $\mathbf{A}^n/n!$ to become insignificant.

In order to control this problem, we note that the growth of $\mathbf{A}^n$ depends on the eigenvalues of $\mathbf{A}$. If all the eigenvalues of $\mathbf{A}$ are inside the unit circle in the complex plane, then $\mathbf{A}^n$ will decay with increasing $n$ and (37) will converge quickly to the desired sum. The same is true of the series in (39). Quick convergence is therefore achieved by controlling $d$ in (38) and (39). With a small enough $d$, the eigenvalues of $\mathbf{A}d$ and $\mathbf{B}d$ will also be small enough so as not to cause truncation errors or slow convergence. Generally convergence is desirable using 20–30 terms and the smallest machine precision as criterion.

Reducing $d$ implies truncating the transmission line. For example with typical parameters of a single lossless line such as $L = 10^{-7}$ H/m and $C = 10^{-10}$ F/m, the shifted $\mathbf{A}$ matrix at 250 MHz will have eigenvalues inside the unit circle only if the length is less than approximately 0.2 meters. This is a generous length and we have found that only rarely does the line length need much truncation.

TABLE I
CPU COST (SUN SPARC II) OF CFH TRANSIENT ANALYSIS APPLIED TO SOME LARGE TRANSMISSION LINE NETWORKS

| Circuit | Number of nodes | $f_{max}$ | Number of hops | dominant poles | CPU time |
|---|---|---|---|---|---|
| 1 | 160 | 0.8 GHz | 2 | 5 | 1.47 |
| | | 6.4 GHz | 8 | 25 | 6.42 |
| 2 | 160 | 1.6 GHz | 3 | 10 | 2.12 |
| | | 16 GHz | 8 | 26 | 4.88 |
| | | 160 GHz | 13 | 50 | 7.72 |
| 3 | 240 | 0.2 GHz | 3 | 13 | 2.90 |
| | | 8 GHz | 5 | 29 | 4.27 |
| 4 | 2777 | 0.5 GHz | 5 | 50 | 49.3 |
| | | 1.5 GHz | 8 | 98 | 70.3 |

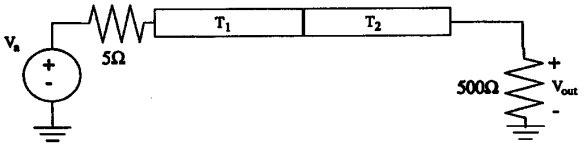



Fig. 3. Example circuit proposed by Chang.

Truncating $d$ can be achieved in one of two ways. Either by sub-dividing the line into $m$ sections and entering each section as a separate stamp in the $\mathbf{Y}(s)$ matrix, or, more efficiently, by noting that,

$$e^{\mathbf{A}d} = e^{\mathbf{A}\frac{d}{2}} e^{\mathbf{A}\frac{d}{2}}. \quad (45)$$

That is, the moments of a line can be generated by squaring the half-line moments. If $C_n^{(\frac{1}{2})}$ represent the half-line moments, then,

$$\begin{aligned} &\mathbf{C}_0 + s\mathbf{C}_1 + s^2\mathbf{C}_2 + \cdots \\ &\quad = \left(\mathbf{C}_0^{(\frac{1}{2})} + s\mathbf{C}_1^{(\frac{1}{2})} + s^2\mathbf{C}_2^{(\frac{1}{2})} + \cdots\right) \\ &\quad \times \left(\mathbf{C}_0^{(\frac{1}{2})} + s\mathbf{C}_1^{(\frac{1}{2})} + s^2\mathbf{C}_2^{(\frac{1}{2})} + \cdots\right) \end{aligned} \quad (46)$$

giving,

$$\mathbf{C}_n = \sum_{i=0}^{n} \mathbf{C}_i^{(\frac{1}{2})} \mathbf{C}_{n-i}^{(\frac{1}{2})}. \quad (47)$$

The line can be subdivided by a power of 2 and the moments of the smallest section calculated. From these the moments of 2 sections, 4 sections, 8 sections, etc. can be calculated. This process can increase the efficiency of moment calculation for a large network if the per-unit-length parameters of lines of different lengths are the same.

An example of the accuracy of the new method is given in example 1 below.

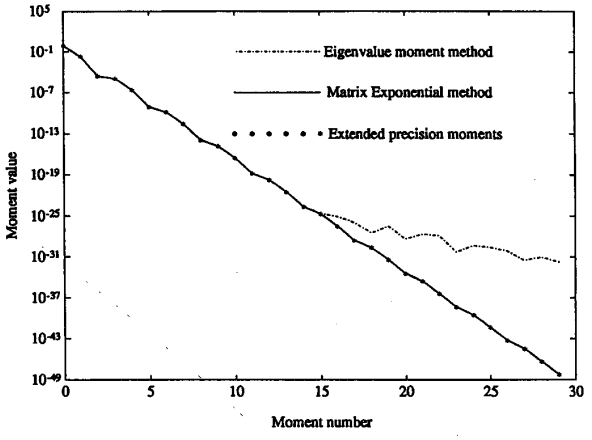



Fig. 4. Moment comparison of Chang's circuit (example 1) using eigenvalue methods and the matrix exponential method.

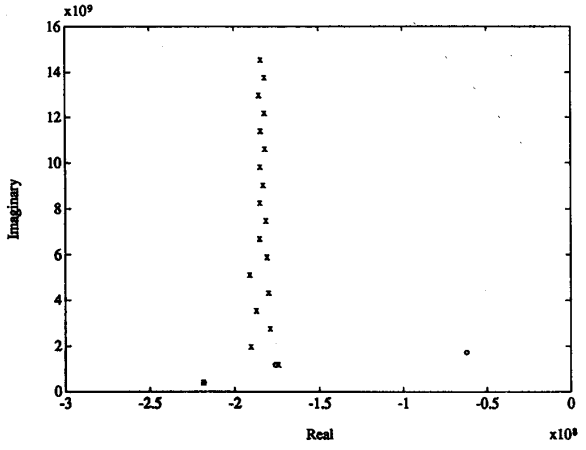



Fig. 5. Pole distribution ($x$) of Chang's circuit (example 1). Approximate poles obtained at the origin ($o$) are also indicated.

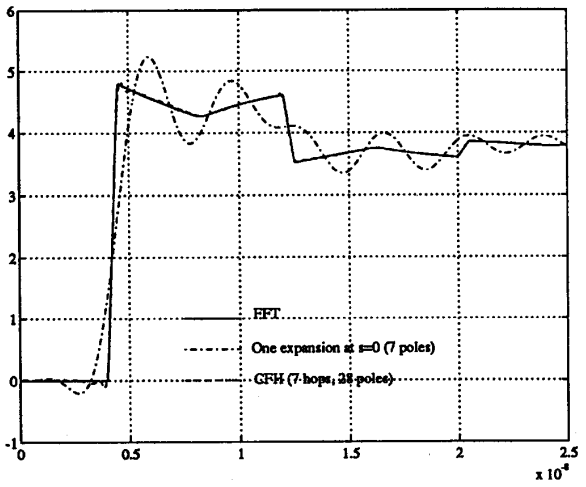



Fig. 6. Transient response comparison of Chang's circuit (example 1).

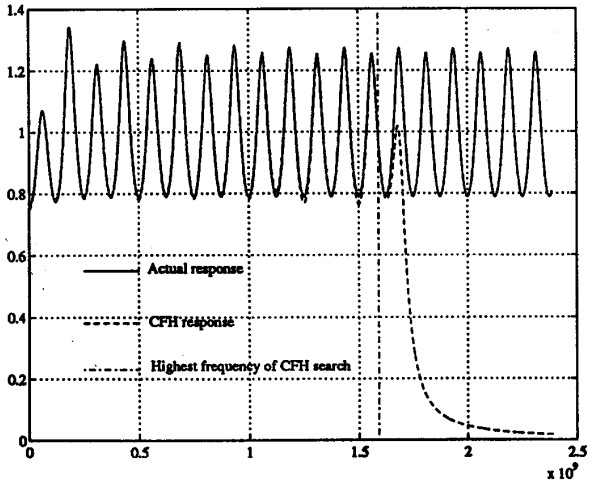



Fig. 7. Frequency response comparison of Chang's circuit (example 1).

## V. Implementation and CPU Analysis

Using the CFH method requires some calculations using complex numbers in contrast to expansions about the real axis or $s = \infty$ which require only real number processing. However, a dominant cost of an expansion is the LU factorization of the MNA matrix which is approximately the CPU cost of one frequency point analysis. A sparse matrix solution requires a matrix reordering before performing the LU factorization. A reordering need not be done again for any additional frequency point [33], provided all of the stamps of a nonzero frequency are accounted for when ordering, and provided that a frequency is not too close to a pole of the system. This alleviates the cost of additional hops. The reordering property of sparse systems has been used to advantage in SPICE for the calculation of frequency response at multiple frequencies.

It is to be noted that the matrix to be ordered, $\mathbf{G}_\pi$ with all of the stamps of $\mathbf{C}$ present, is generally more dense and therefore somewhat more expensive than the AWE matrix necessary to expand only at the origin, $\mathbf{G}_\pi$. However, AWE may require a shifted LU factorization about $s = \alpha$ for stiff systems [26] which may mean a new ordering. We therefore choose to factorize a fully stamped matrix at the origin both to reuse the ordering for frequency shifting and to reuse it in case of a stiff system.

The CPU cost at the origin (or on the real axis) is partly composed of a reordering, a (real) LU factorization and real moment generation and Padé table analysis. Every other expansion point consists of a complex LU factorization (without reordering), complex moment generation, and complex Padé table convergence. In general, a sparse matrix reordering consumes more CPU time than one LU factorization. For example, CFH was implemented using Berkeley Sparse routines [34]. A real LU factorization of a 240 node circuit about the

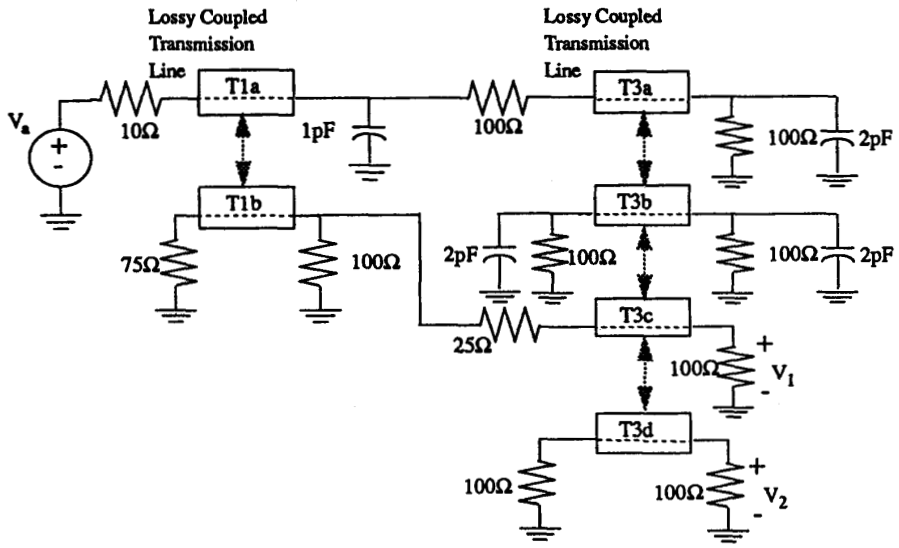



Fig. 8. A lossy coupled transmission line network (example 2).

origin took 0.22 CPU seconds on a SUN SPARC II, while a subsequent complex factorization of the same circuit took 0.02 s. The ratio was 13.83 s to 0.77 s for a 1143 node circuit. As another example, a large network containing 306 resistors, 2100 capacitors, 172 inductors and 6990 mutual inductances was analyzed to 1 GHz. The original hop about the origin consumed 96.2 CPU seconds which showed it to be a stiff system. Another real hop was performed which consumed 17.7 s to alleviate the stiffness. A final highest frequency complex hop took 26.3 s.

The additional CPU expense of Padé table convergence and moment generation is more expensive using complex numbers as compared to real numbers. The former however, is independent of circuit size and diminishes therefore in proportion to the size of the circuit, while the latter is generally 3–4 time more expensive than real moment generation. However since moment generation and Padé analysis is a small fraction of the cost of LU factorization in a large circuit [10], [12], the increase is minimal. Table I gives a sample of some CPU times of CFH applied to large linear interconnect networks.

At low-frequency operations, the binary search strategy bottoms out after two hops. This gives a guarantee of approximation accuracy within the specified frequency at the cost of approximately two frequency point analyses. The CPU cost increases almost linearly with the number of hops required. Some difficult waveforms will require more than two hops, but in this case, it is clear that one expansion will not have been enough to capture the details of the waveform accurately.

One important advantage to CFH is that is can be readily implemented in parallel. This is not generally true of a differential equation solver such as SPICE [30]. Parallelization may be achieved by assigning one hop to each compute node and assuring that the hops tile the imaginary axis in a regular pattern. The information from each hop may then be gathered at the central compute node and collected in terms of poles and residues of a single transfer function. The result of such a process would be an accurate simulation at the cost of a little more than one complex LU factorization.

CFH was implemented in a simulator called CoFHee (complex frequency hopping electrical evaluator (pronounced "coffee") at Carleton University and applied to diverse interconnect circuits some of which are given in the examples below.

## VI. Examples

*Example 1:* In Fig. 3 a small network proposed by Chang [35] is shown. Two transmission lines $(T_1, T_2)$ with different resistive parameters are cascaded. The parameters of $T_1$ are $Rd = 25\ \Omega$, $Gd = 0.005\ \Omega^{-1}$, $Ld = 0.1\ \mu$H, $Cd = 40$ pF, where $d$ is the length of the transmission line. The parameters of $T_2$ are the same except for $Rd = 3.125\ \Omega$. A 1 V step of 0.5 ns rise/fall time is placed at the input, and the output waveform was generated at the load resistance by CoFHee. Moments of one expansion point were generated about the origin using the matrix exponential method. These are compared in Fig. 4 to the accurate moments that were generated using multiple precision programming. They are almost identical. The same moments were generated by eigenvalue methods and show an increase in numerical truncation error (deviation from the high precision moments) with higher order moments. The same results can be obtained with expansions about other points, or with other networks.

CFH was applied to the network for analysis up to $\omega_{\max} = 1e10$ or $f_{\max} = 1.59$ GHz. The analysis generated 28 poles using 7 hops. The pole distribution of the circuit is shown in Fig. 5 by the '$x$'. The '$o$' represent the approximate poles of the best moment-matching Padé approximation about $s = 0$. Clearly, one Padé approximation fails to obtain many of the high frequency poles very close to the imaginary axis.

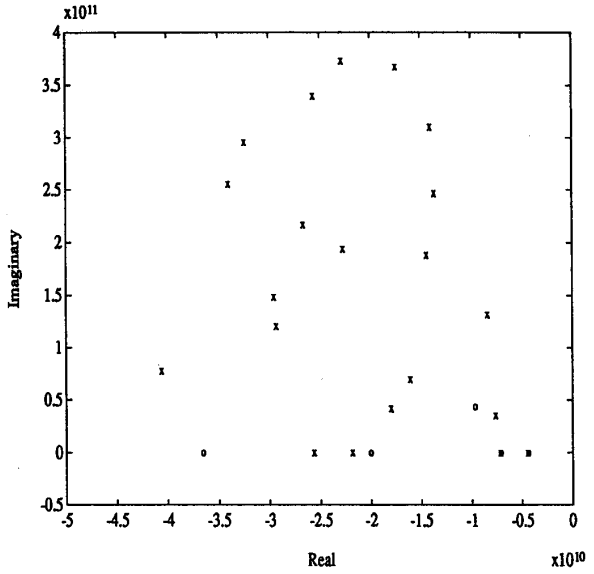



Fig. 9. Poles of the network in example 2.

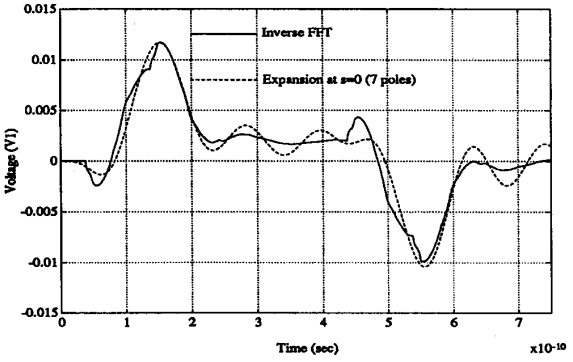



Fig. 10. Transient response comparison of $V_1$ in example 2 using one expansion at $s = 0$ which generated 7 poles.

In Fig. 6 we compare the accurate FFT generated transient response using 1024 frequency point analyses to the approximate transient response using one expansion point, as well as to the CFH response using multiple expansion points. The frequency response of CFH is compared to the full frequency response using hundreds of LU decompositions Fig. 7. The CFH frequency response matches the actual response up to $f_{\max}$ after which it drops off.

*Example 2:* We show a small lossy coupled transmission line network in Fig. 8. The network contains 2 coupled lines and 4 coupled lines. The network was analyzed at two outputs: $V_1$ and $V_2$. An input pulse of 0.1 ns rise/fall time and 0.3 ns duration was placed at the input. First, one expansion at the origin was used and then several additional incremental hops were generated. The pole distribution of the network is displayed in Fig. 9 by '$x$.' These poles were obtained by

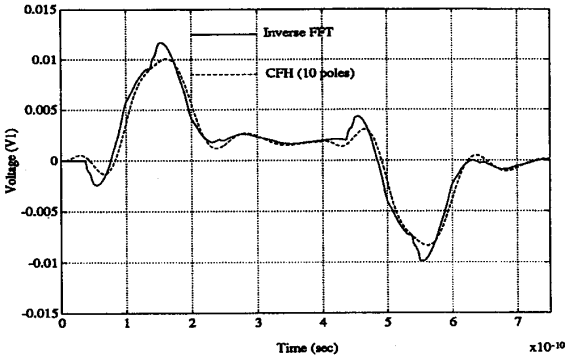



Fig. 11. Transient response comparison of $V_1$ in example 2 using two hops which generated 10 poles.

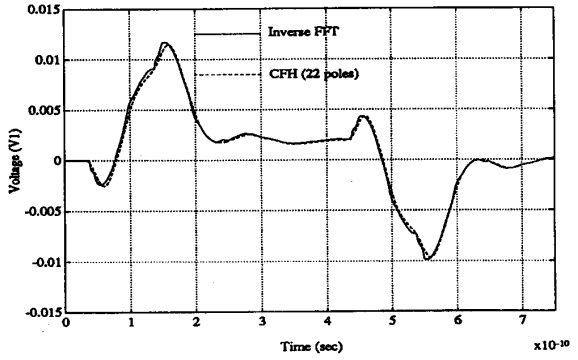



Fig. 12. Transient response comparison of $V_1$ in example 2 using four hops which generated 22 poles.

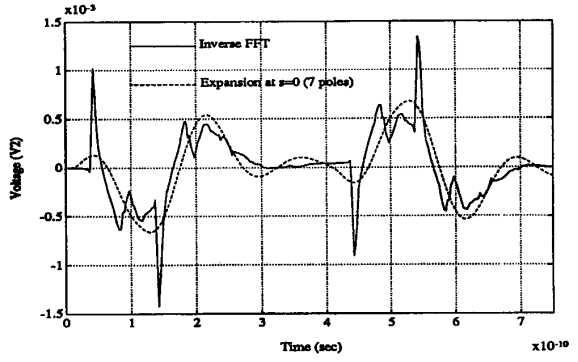



Fig. 13. Transient response comparison of $V_2$ in example 2 using one expansion at $s = 0$ which generated 7 poles.

multiple hops. The approximate poles generated at the origin are marked by '$o$.'

The waveforms at the nodes are compared with the accurate FFT generated one. At $V_1$ one expansion point at the origin produced the waveform displayed in Fig. 10. Another hop

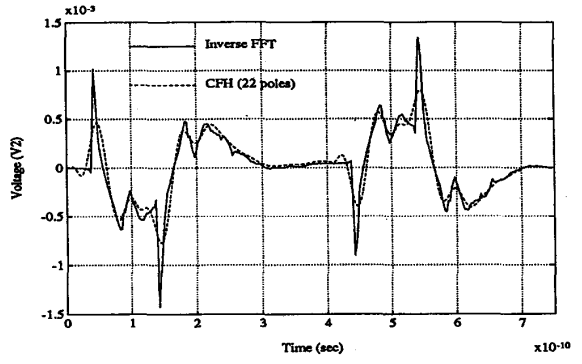



Fig. 14. Transient response comparison of $V_2$ in example 2 using four hops which generated 22 poles.

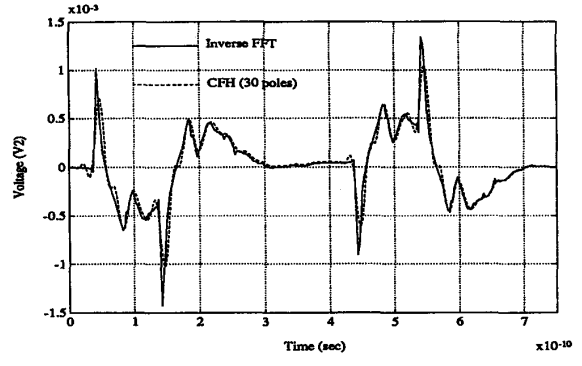



Fig. 15. Transient response comparison of $V_2$ in example 2 using six hops which generated 30 poles.

allowed the generation of 10 accurate poles and gave the waveform given in Fig. 11. Two more hops gave a total of 22 poles and gave the almost converged waveform in Fig. 12.

The waveform at $V_2$ was more difficult to obtain because it is 3 orders of magnitude smaller than the input voltage and contains many high frequency components. One expansion point at the origin produced the waveform displayed in Fig. 13. Four hops produced 22 poles and gave the waveform given in Fig. 14. Finally, the waveform is almost converged with six hops giving a total of 30 poles and the waveform in Fig. 15.

*Example 3:* A 160 node circuit with 96 capacitors, 84 inductors and 116 resistors as well as 8 lossy interconnect transmission lines, was simulated using CoFHee. The transmission lines were mismatched with respect to the terminations, giving rise to strong reflections. The voltage was sampled close to the source, immediately after the first transmission line. The input pulse was of 1 V, 10 ps rise/fall time, and 30 ps duration. The highest frequency searched was up to 159 GHz. 12 hops were required with a simulation time of 7.72 s and a resulting 50 poles/residues. The difficult signal is compared with the FFT generated one in Fig. 16. The delay is clear at the beginning of the waveform after which the pulse follows. Past 70 ps however, the reflection from the rest of the circuit clearly affects the signal. Frequency response of CFH is compared

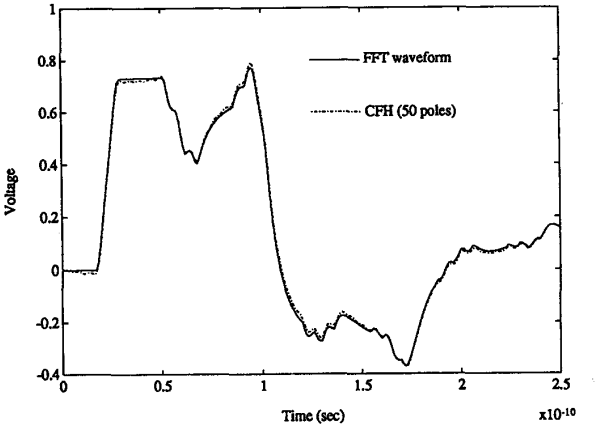



Fig. 16. Transient response comparison of CFH applied to 160 node lossy transmission line network (example 3).

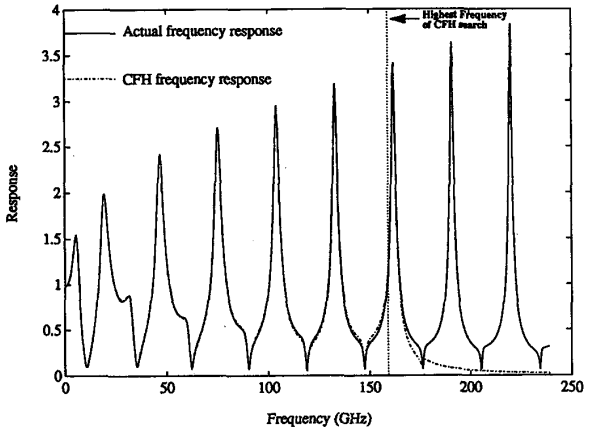



Fig. 17. Frequency response comparison of CFH applied to 160 node lossy transmission line network with a search up to 159 GHz (example 3).

to the actual frequency response in Fig. 17. Berkeley Spice3 was used for comparison. It could only simulate single lines without both nonzero $R$ and nonzero $G$ and we made $G = 0$ in the circuit. The CPU time on the same machine was 8.7 s.

*Example 4:* A clock tree circuit (Fig. 18) was simulated by CoFHee. It contained 73 lossy lines, 2895 resistors, and 2777 capacitors and was driven by two cascaded CMOS inverters. The output was generated on two of the clock pins chosen at random. A 5 V input signal with a rise time of 0.1 ns was placed on the input to the inverters. The resulting output at a clock line is shown in Fig. 19. The CPU time for the linear analysis was 33.4 s and the nonlinear analysis for 10 ns run with a step of 0.04 ns consumed 53.7 s on a SUN SPARC II. The result compared favourably with SWEC [11]. The CPU time using Spice 3f (and reducing $G$ to zero) was 20 671 s. The slowdown was attributed to the slow convergence due to the nonlinear elements and the large size of the linear subcircuit.

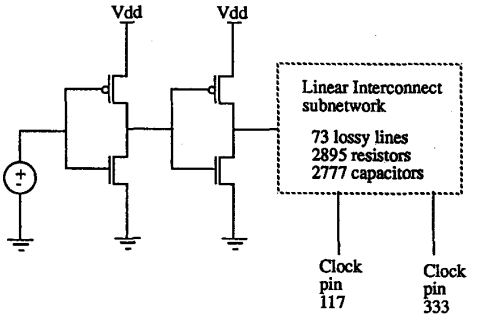



Fig. 18. A clock tree network containing lossy transmission lines and driven by 2 CMOS inverters (example 4).

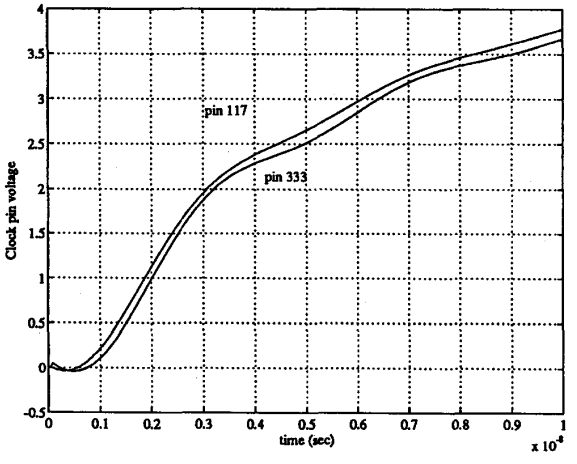



Fig. 19. Transient response of the clock tree network (example 4).

## VII. Summary and Future Work

We have introduced a novel method of simulating large networks which include lossy, coupled transmission lines, and nonlinear terminations, and for generating the dominant eigenvalues of a linear lumped or distributed network. The method relies on *multipoint* expansions and Padé approximations in the complex $s$ plane to generate the actual dominant poles of a circuit and a single transfer function for an entire network using the equivalent CPU of only a few frequency point analyses. CFH, as implemented in the CoFHee simulator, was demonstrated to be accurate and efficient. Future research should concentrate on further reducing the number of frequency hops required, and on implementing the algorithm in parallel, with one hop computed per compute node.

## Acknowledgment

The authors are grateful to E. Bracken of Carnegie Mellon University for discussions about transmission lines in AWE, to S. Lin of IBM T. J. Watson research center for providing the clock tree example, and to D. Xie and R. Griffith of Carleton University for implementing an efficient macromodel generator for SPICE using the poles and residues generated by CoFHee.

## References

[1] S. S. Gao, A. Y. Yang and S. M. Kang, "Modeling and simulation of interconnection delays and crosstalks in high-speed integrated circuits," *IEEE Trans. Circuits Syst.,* vol. 37, pp. 1–9, Jan. 1990.

[2] H. Hasegawa and S. Seki, "Analysis of interconnection delay on very high-speed LSI/VLSI chips using a microstrip line model," *IEEE Trans. Electron Devices,* vol. ED-31, pp. 1954–1960, Dec. 1984.

[3] W. W. M. Dai (Guest Editor), "Special issue on simulation, modeling, and electrical design of high-speed and high-density interconnects," *IEEE Trans. Circuits Syst.* vol. 39 no. 11, pp. 857–982, Nov. 1992.

[4] H. B. Bakoglu, *Circuits, Interconnections and Packaging for VLSI.* Reading, MA: Addison-Wesley, 1990.

[5] A. R. Djordjević, T. K. Sarkar, and R. F. Harrington, "Analysis of lossy transmission lines with arbitrary nonlinear terminal networks," *IEEE Trans. Microwave Theory Tech.,* vol. 34, pp. 660–666, June 1986.

[6] F. Y. Chang, "The generalized method of characteristics for waveform relaxation analysis of lossy coupled transmission lines," *IEEE Trans. Microwave Theory Tech.,* vol. 37, pp. 2028–2038, Dec. 1989.

[7] J. E Schutt-Aine and R. Mitra, "Nonlinear transient analysis of coupled transmission lines," *IEEE Trans. Circuits Syst.,* vol. 36, pp. 959–967, July 1989.

[8] R. Wang and O. Wing, "Transient analysis of dispersive VLSI interconnects terminated in nonlinear loads," *IEEE Trans. Computer- Aided Design,* vol. 11, pp. 1258–1277, Oct. 1992.

[9] R. Griffith and M. S. Nakhla, "Mixed frequency/time domain analysis of nonlinear circuits," *IEEE Trans. Computer-Aided Design,* vol. 10, pp. 1032–1043, Aug. 1992.

[10] 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.

[11] S. Lin and E. S. Kuh, "Transient simulation of lossy interconnects based on the recursive convolution formulation," *IEEE Trans. Circuits Syst.* vol. 39, no. 11, pp. 879–892, Nov. 1992.

[12] T. Tang and M. S. Nakhla, "Analysis of high-speed VLSI interconnect using the asymptotic waveform evaluation technique," *IEEE Trans. Computer Aided-Design,* vol. 11, Mar. 1992.

[13] J. E. Bracken, V. Raghavan, and R. A. Rohrer, "Interconnect simulation with asymptotic waveform evaluation (AWE)," *IEEE Trans. Circuits Syst.,* vol. 39, no. 11, pp. 869–878, Nov. 1992.

[14] D. F. Anastasakis, N. Gopal, S. Y. Kim, and L. T. Pillage, "On the stability of moment-matching approximations in asymptotic waveform evaluation," in *Proc. ACM/IEEE Design Automation Conf.,* June 1992, pp. 207–212.

[15] R. Griffith, E. Chiprout, Q. J. Zhang, and M. S. Nakhla, "A CAD framework for simulation and optimization of high-speed VLSI interconnections," *IEEE Trans. Circuits Syst.,* vol. 39, no. 1, pp. 893–906, Nov. 1992.

[16] E. Chiprout and M. S. Nakhla, "Fast nonlinear waveform estimation of large distributed networks," in *Proc. IEEE MTT-S Int. Microwave Symp.,* 1992, pp. 1341–1344.

[17] D. Xie and M. S. Nakhla, "Delay and crosstalk simulation of high speed VLSI interconnects with nonlinear terminations," *IEEE Trans. Computer-Aided Design,* vol. 12, no. 11, pp. 1198–1811, Nov. 1993.

[18] G. A. Baker Jr., *Essential of Padé Approximants.* New York: Academic, 1975.

[19] J. H. McCabe, "A formal extension of the Padé table to include two point Padé quotients," *J. Inst. Math. Applic.,* vol. 15, pp. 363–372, 1975.

[20] W. B. Jones and A. Magnus, "Computation of poles of two-point Padé approximants and their limits," *J. Computational App. Math.,* vol. 6, no. 2, pp. 105–119, 1980.

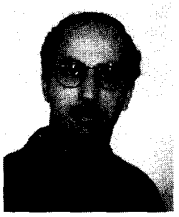
[21] G. A. Baker Jr. and P. Graves-Morris, "Padé approximants," in *Encyclopedia of Mathematics and Its Applications.*, G. C. Rota, Ed. Reading, MA: Addison-Wesley, 1981, vols. 13, 14.

[22] E. Chiprout, H. Heeb, M. S. Nakhla, and A. E. Ruehli, "Simulating 3-D retarded interconnect models using complex frequency hopping (CFH)," in *IEEE Int. Conf. Computer-Aided Design*, Nov. 1993, pp. 66–72.

[23] P. K. Chan, "Comments on 'asymptotic waveform evaluation for timing analysis,'" *IEEE Trans. Computer-Aided Design*, pp. 1078–1079, Aug. 1991.

[24] E. Chiprout and M. S. Nakhla, "Generalized moment matching methods for transient analysis of interconnect networks," in *Proc. ACM/IEEE Design Automation Conf.,* pp. 201–206, 1992.

[25] C. W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 504–509, June 1975.
[26] X. Huang, "Padé Approximation of linear(ized) circuit responses," Ph.D. dissertation, Carnegie Mellon Univ., Nov. 1990.
[27] D. S. Watkins, "Understanding the QR algorithm," *Siam Rev.*, vol. 24, no. 4, pp. 427–440, Oct. 1982.
[28] N. Gopal, D. P. Neikirk, and L. T. Pillage, "Evaluating on-chip RC-interconnect using moment-matching approximations," in *IEEE Int. Conf. Computer-Aided Design*, pp. 74–77, 1991.
[29] E. Chiprout and M. S. Nakhla, "Optimal pole selection in asymptotic waveform evaluation," in *Proc. IEEE Int. Symp. on Circuits and Syst.*, June 1992, pp. 1961–1964.
[30] L. W. Nagel, "SPICE2, a computer program to simulate semiconductor circuits," Tech. Rep., ERL-M520, U.C.-Berkeley, May 1975.
[31] S. Wolfarm, *Mathematica: A System for Doing Mathematics by Computer.* Reading, MA: Addison-Wesley, 1991.
[32] C. Moler and C. V. Loan, "Nineteen dubious ways to compute the exponential of a matrix," *Siam Rev.*, vol. 20, no. 4, pp. 801–836, Oct. 1978.
[33] K. S. Kundert, "Sparse matrix techniques," *Circuit Analysis, Simulation and Design*, A. E. Ruehli, Ed. Amsterdam: North-Holland, 1986, vol. 3, pt. 1.
[34] ——, "Sparse 1.3 user's guide," Tech. Rep., Univ. CA Berkeley, 1987.
[35] F. Y. Chang, "Waveform realaxation analysis of nonuniform lossy transmission lines characterized with frequency-dependent parameters," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 1484–1500, Dec. 1991.



**Eli Chiprout** received the B.Eng. degree in electrical engineering from McGill University, Montréal, Québec, Canada, in 1988, and the M.S. and Ph.D. degrees from Carlton University, Ottawa, Ontario, Canada, in 1992 and 1994, respectively.

In the past, he has been employed by IBM Canada as a systems engineer, and by Mitel Semiconductor as a VLSI CAD engineer. He is now with IBM Watson Research Center, Yorktown Heights, NY. He is co-author of *Asymptotic Waveform Evaluation* (Boston: Kluwer, 1993). His research interests include computer-aided design of high-speed systems and numerical algorithms.

**Michel S. Nakhla** (S'73–M'76–SM'88) received the B.Sc. degree in electronics and communications from Cairo University, Egypt, in 1967, and the M.A.Sc. and the Ph.D. degrees in 1973 and 1975, respectively.

In 1975, he was a Post-Doctoral Fellow at the University of Toronto, Ontario, Canada. In 1976, he joined Bell-Northern Research, Ottawa, Canada, as a member of the Scientific Staff, where he became manager of the simulation group in 1983. In 1988, he joined Carleton University, Ottawa, Canada, where he is currently a Professor with the Department of Electronics. His research interests include computer-aided design of VLSI and communications systems, high-frequency interconnects and synthesis of analog circuits.

Dr. Kakhla is the co-writer of *Asymptotic Waveform Evaluation* (Boston: Kluwer, 1993), and co-editor of *Modeling and Simulation of High-Speed VLSI Interconnects* (Boston: Kluwer, 1974). In 1984 and 1985, he was the recipient of the Bell Northern Research Outstanding Contribution Patent Award. Currently, he is the holder of the Computer-Aided Engineering Industrial Chair established at Carleton University by Bell Northern Research and the Natural Sciences and Engineering Research Council of Canada.