RECENT ADVANCES IN MODELING AND SIMULATION OF HIGH-SPEED INTERCONNECTS

Michel Nakhla and Ram Achar
Department of Electronics, Carleton University,
Ottawa, ON - K1S5B6, Canada
E-mail: msn@doe.carleton.ca

The rapid increase in operating speeds, density and complexity of modern integrated circuits has made interconnect analysis a requirement for all state-of-the-art circuit simulators. Interconnect effects such as ringing, signal delay, distortion and crosstalk can severely degrade signal integrity. Interconnections can be from various levels of design hierarchy. As the frequency of operations increases, the interconnect lengths become a significant fraction of the operating wavelength, and conventional lumped models become inadequate in describing the interconnect performance and transmission line models become necessary. This chapter describes some of the recent advances in transmission line simulation techniques. Application of mode-order-reduction algorithms to high-speed interconnect analysis is also presented.

1. Introduction

With the rapid developments in VLSI technology, design and CAD techniques, at both the chip and package level, the operating frequencies are fast reaching the vicinity of GHz and switching times are getting to the sub-nano second levels. The ever increasing quest for high-speed applications is placing higher demands on interconnect performance and highlighted the previously negligible effects of interconnects such as ringing, signal delay, distortion, reflections and crosstalk. Interconnects can exist at various levels of design hierarchy such as on-chip, packaging structures, multichip modules, printed circuit boards and backplanes. In addition, the trend in the VLSI industry towards miniature designs, low power consumption and increased integration of analog circuits with digital blocks has further complicated the issue of signal
integrity analysis. It is predicted that interconnects will be responsible for majority of signal degradation in high-speed systems\textsuperscript{1-21}. High-speed interconnect problems are not always handled appropriately by the conventional circuit simulators, such as SPICE\textsuperscript{22}. If not considered during the design stage, these interconnect effects can cause logic glitches which render a fabricated digital circuit inoperable, or they can distort an analog signal such that it fails to meet specifications. Since extra iterations in the design cycle are costly, accurate prediction of these effects is a necessity in high-speed designs. Hence it becomes extremely important for designers to simulate the entire design along with interconnect subcircuits as efficiently as possible while retaining the accuracy of simulation\textsuperscript{22-135}. Speaking on a broader perspective, a “high-speed interconnect” is the one in which the time taken by the propagating signal to travel between its end points can’t be neglected. An obvious factor which influences this definition is the physical extent of the interconnect, the longer the interconnect, more time the signal takes to travel between its end points. Smoothness of signal propagation suffers once the line becomes long enough for signal’s rise/fall times to roughly match its propagation time through the line. Then the interconnect electrically isolates the driver from the receivers, which no longer function directly as loads to the driver. Instead, within the time of signal’s transition between its high and low voltage levels, the impedance of interconnect becomes the load for the driver and also the input impedance to the receivers\textsuperscript{1,11}. This leads to various transmission line effects, such as reflections, overshoot, undershoot, crosstalk and modeling of these needs the blending of EM and circuit theory.

Alternatively, the term ‘high-speed’ can be defined in terms of the frequency content of the signal. At low frequencies an ordinary wire, in other words, an interconnect, will effectively short two connected circuits. However, this is not the case at higher frequencies. The same wire, which is so effective at lower frequencies for connection purposes, has too much inductive/capacitive effects to function as a short at higher frequencies. Faster clock speeds and sharper slew rates tend to add more and more high-frequency contents. An important criterion used for classifying interconnects is the electrical length of an interconnect. An interconnect is considered to be “electrically short”, if at the highest operating frequency of interest, the interconnect length is physically shorter than approximately one-tenth of the
wave-length (i.e., length of the interconnect/λ = 0.1, \( \lambda = v/f \)). Else the interconnect is referred as “electrically long”\(^1,11\). In most digital applications, the desired highest operating frequency (which corresponds to the minimum wavelength) of interest is governed by the rise/fall time of the of the propagating signal. For example, the energy spectrum of a trapezoidal pulse is spread over an infinite frequency range, however, most of the signal energy is concentrated near the low frequency region and decreases rapidly with increase in frequency. Hence ignoring the high-frequency components of the spectrum above a maximum frequency, \( f_{max} \), will not seriously alter the overall signal shape. Consequently, for all practical purposes, the width of the spectrum can be assumed to be finite. In other words, the signal energy of interest is assumed to be contained in the major lobes of the spectrum and the relationship between desired \( f_{max} \) with the \( t_r \) is the rise/fall time of the signal can be expressed as\( ^2, 3, 35, 64, 86 \)

\[
| \begin{align*}
f_{max} &= 0.35/t_r 
\end{align*} |
\]

This implies that, for example, for a rise time of 0.1ns, the maximum of frequency of interest is approximately 3GHz or the minimum wave-length of interest is 10cms. In some cases the limit can be more conservatively set as \( f_{max} \approx 1/t_r \) \( ^86 \).

In summary, the primary factors which influence the decision that, “whether high-speed signal distortion effects should be considered”, are interconnect length, cross-sectional dimensions, signal slew rate and the clock-speed. Other factors which also should be considered are logic levels, dielectric material and conductor resistance. Electrically short interconnects can be represented by lumped models where as electrically long interconnects need distributed or full-wave models.

**High-Speed Interconnect Models**

Depending on the operating frequency, signal rise times and nature of the structure, the interconnects can be modeled as lumped, distributed (frequency independent/dependent RLCG parameters, lossy, coupled) or full-wave models.
Lumped Models

At lower frequencies, the interconnect circuits could be modelled using lumped RC or RLC circuit models. RC circuit responses are monotonic in nature. However, in order to account for ringing in signal waveforms, RLC circuit models may be required. Usually lumped interconnect circuits extracted from layouts contain large number of nodes which make the simulation highly CPU intensive.

Distributed Transmission Line Models

At relatively higher signal-speeds, electrical length of interconnects becomes a significant fraction of the operating wavelength, giving rise to signal distorting effects that do not exist at lower frequencies. Consequently, the conventional lumped impedance interconnect models become inadequate and transmission line models based on quasi-TEM assumptions are needed. The TEM (Transverse Electromagnetic Mode) approximation represents the ideal case, where both $E$ and $H$ fields are perpendicular to the direction of propagation and it is valid under the condition that the line cross-section is much smaller than the wavelength. However, the inhomogeneities in practical wiring configurations, give rise to $E$ or $H$ fields in the direction of propagation. If the line cross-section or the extent of these nonuniformities remain a small fraction of the wavelength in the frequency range of interest, the solution to Maxwell’s equations are given by the so called quasi-TEM modes and are characterized by distributed $R, L, C, G$ per unit length parameters\textsuperscript{11}.

In practical situations, owing to complex interconnect geometries and varying cross-sectional areas, the interconnects may need to be modelled as nonuniform lines. In this case, the per unit length parameters are functions of distance, $x$\textsuperscript{91 - 93}.

Distributed Models with Frequency-Dependent Parameters

At low frequencies, the current in a conductor is distributed uniformly through out its cross section. However, as the operating frequency increases,
the current distribution gets uneven and it starts getting concentrated more and more near the surface or edges of the conductor. This phenomenon can be categorized as follows: skin, edge and proximity effects. The skin effect causes the current to concentrate in a thin layer near the conductor surface and this reduces the effective cross-section available for signal propagation. This leads to increase in the resistance to signal propagation and other related effects. The edge effect causes the current to concentrate near the sharp edges of the conductor. The proximity effect causes the current to concentrate in the sections of ground plane that are close to the signal conductor. To account for these effects, modelling based on frequency-dependent p.u.l. parameters may be necessary.

Rest of the notes is organized as follows. Section-2 provides a detailed analysis of transmission line equations and derivation of a generic multiconductor transmission line stamp, suitable for inclusion in a MNA analysis. In Section-3, a review of formulation of circuit equations in the presence of distributed elements and limitations of conventional simulators is given. Review of efficient techniques for discretization of Telegrapher’s equations is given in Section-4. Sections 5-7 give a detailed account of simulation of interconnects using model-reduction techniques. Section-8 provides references to related topics.

2. **Distributed Transmission Line Equations**

Transmission line characteristics are in general described by Telegrapher’s equations. Consider the transmission line system shown in Fig. 1a. Telegrapher’s equations for such a structure can be derived by discretizing the line into infinitesimal sections of length \( \Delta x \) and assuming uniform per-unit length (p.u.l.) parameters of resistance \( R \), inductance \( L \), conductance \( G \) and capacitance \( C \). Each section then includes a resistance \( R\Delta x \), inductance \( L\Delta x \), conductance \( G\Delta x \) and capacitance \( C\Delta x \) (Fig. 1b). Using Kirchoff’s current and voltage laws, one can write\(^{11}\),

\[
v(x + \Delta x, t) = v(x, t) - R\Delta x i(x, t) - L\Delta x \frac{di(x, t)}{dt}
\]  

or
\[
\frac{v(x + \Delta x, t) - v(x, t)}{\Delta x} = -Ri(x, t) - L\frac{\partial}{\partial t}i(x, t)
\]

(3)

Taking the limit \(\Delta x \to 0\), one gets
\[
\frac{\partial}{\partial x}v(x, t) = -Ri(x, t) - L\frac{\partial}{\partial t}i(x, t)
\]

(4)

Similarly, we can obtain the second transmission line equation in the form:
\[
\frac{\partial}{\partial x}i(x, t) = -Gv(x, t) - C\frac{\partial}{\partial t}v(x, t)
\]

(5)

Taking Laplace transform of equations (4) and (5) one can write
where \( Z \) and \( Y \) represent the p.u.l. impedance and admittances of the transmission line, given by

\[
Z = R + sL; \quad Y = G + sC
\]

(8)

The set of equations represented by (6) and (7) can be solved if they can be written in terms of one of the unknowns (either \( V(x, s) \) or \( I(x, s) \)) as follows

\[
\frac{d^2}{dx^2}V(x, s) = ZYV(x, s) = Y^2V(x, s)
\]

(9)

\[
\frac{d^2}{dx^2}I(x, s) = YZI(x, s) = Y^2I(x, s)
\]

(10)

where \( \Upsilon(s) \) is the complex propagation constant, given by

\[
\Upsilon(s) = \alpha + j\beta = \sqrt{Z\Upsilon} = \sqrt{(R + j\omega L)(G + j\omega C)}
\]

(11)

where \( \alpha \) represents the real part of the propagation constant and is known as attenuation constant, whose units are expressed in (nepers/m). \( \beta \) represents the imaginary part of the propagation constant and is known as phase constant, whose units are expressed in (radians/m). Solution of (9) and (10) is given as a combination of forward-reflected waves travelling on the line as

\[
V(x, s) = V_0e^{\pm\Upsilon(s)x}
\]

(12)

\[
I(x, s) = I_0e^{\pm\Upsilon(s)x}
\]

(13)

The phase shift and attenuation experienced by the travelling waves are given by \( e^{\pm j\beta s x} \) and \( e^{\pm\alpha s x} \), respectively. If the lines are lossless, the propagation constant is given by, \( \Upsilon(s) = j\beta = \sqrt{Z\Upsilon} = j\omega\sqrt{L/C} \). The line in this case represents a pure-delay element.
2.1 Multiconductor Transmission Line System

Consider the multiconductor transmission line (MTL) system, with $N$ coupled conductors, shown in Fig. 2.

Using the steps similar to the case of single transmission line, we can derive the multiconductor transmission line equations. Per-unit-length parameters ($R$, $L$, $G$ & $C$) in this case become matrices and voltage/current variables become vectors represented by $v$ and $i$, respectively. Noting these changes, we can re-write (4) and (5) as

Fig. 2. Multiconductor transmission line system
Recent Advances in Modeling and Simulation of High-Speed Interconnects

Taking Laplace transform of equation (14) one can write

\[ \frac{\partial}{\partial x} \begin{bmatrix} V(x, t) \\ I(x, t) \end{bmatrix} = - \begin{bmatrix} 0 & R \\ G & 0 \end{bmatrix} \begin{bmatrix} I(x, t) \\ V(x, t) \end{bmatrix} - \begin{bmatrix} 0 & L \\ C & 0 \end{bmatrix} \frac{2}{\partial x} \begin{bmatrix} I(x, t) \\ V(x, t) \end{bmatrix} \]  

(14)

Taking Laplace transform of equation (14) one can write

\[ \frac{\partial}{\partial x} V(x, s) = -ZI(x, s) \]  

(16)

\[ \frac{\partial}{\partial x} I(x, s) = -YV(x, s) \]  

(17)

where \( Z \) and \( Y \) represent the impedance and admittance matrices, given by

\[ Z = R + sL; \quad Y = G + sC \]  

(18)

The \( R, L, G \) & \( C \) matrices are obtained by a two-dimensional solution of Maxwell’s equations at appropriate positions, along the propagation axis. For this purpose, depending on the nature and geometry of the structure, and the desired accuracy, techniques based on quasi-static or full-wave approaches can be used. The \( R, L, G \) & \( C \) matrices are symmetric and positive definite\(^{11, 86}\).

### 2.2 Multiconductor Transmission Line Stamp

In this section, we derive a stamp relating the terminal currents and voltages of MTL structures, suitable for inclusion in SPICE-like simulators. The transmission line stamp\(^{61}\) is derived, through *de-coupling of MTL equations*.

From (16) and (17), we get two sets of coupled wave equations as

\[ \frac{\partial^2}{\partial x^2} V(x, s) = ZYV(x, s) \]  

(19)

\[ \frac{\partial^2}{\partial x^2} I(x, s) = YZI(x, s) \]  

(20)

Decoupling of equations in (19) or (20) can be achieved through the use of suitable modal transformation matrices\(^{10}\). For this purpose, introduce a transformation \( W \) relating the circuit voltages \( V \) and modal voltages \( \tilde{V} \) as
\[ V(x, s) = W \tilde{V}(x, s) \]  

(21)

Hence (19) can be re-written as (for simplicity, we omit the accompanying term \((x, s)\))

\[ \frac{d^2 \tilde{V}}{dx^2} = (F^{-1} Z Y W) \tilde{V} \]  

(22)

For effective de-coupling of equations to take place, the matrix product in parenthesis must lead to a diagonal matrix as,

\[ F^{-1} Z Y W = \begin{bmatrix} \Upsilon_1^2 & 0 & 0 \\ 0 & \ldots & 0 \\ 0 & 0 & \Upsilon_N^2 \end{bmatrix} \]  

(23)

where the diagonal matrix contains the eigenvalues of the product \(Z Y\), which corresponds to the roots of the characteristic equation

\[ \Upsilon_k^2 U - Z Y = 0; \quad k = 1, 2, \ldots, N \]  

(24)

where \(U\) represents the unity matrix (we assume the general case that, there exist \(N\) distinct eigenvalues).

Having obtained the propagation constants, solution of (22) can be written in the standard form as

\[ \tilde{V}_k(x) = e^{-\Upsilon_k x} c_{ki} + e^{\Upsilon_k x} c_{kr}; \quad k = 1, 2, \ldots, N \]  

(25)

where \(\tilde{V}_k(x)\) represents the \(k^{th}\) modal voltage and \(c_{ki}, c_{kr}\) are the corresponding constants, pertaining to incident and reflected waves, respectively. Eq. (25) can be written in the matrix form as
Define, \( E(x) = \text{diag} \left[ e^{-\gamma_1 x}, \ldots, e^{-\gamma_N x} \right] \) and premultiplying both sides of (26) by the modal transformation matrix \( F \) (from (21)), we can write (26) in terms of circuit voltages as

\[
\begin{bmatrix}
\tilde{V}_1(x) \\
\tilde{V}_2(x) \\
\vdots \\
\tilde{V}_N(x)
\end{bmatrix}
= 
\begin{bmatrix}
e^{-\gamma_1 x} & e^{-\gamma_2 x} & \cdots & e^{-\gamma_N x} \\
e^{-\gamma_1 x} & e^{-\gamma_2 x} & \cdots & e^{-\gamma_N x} \\
\vdots & \vdots & \ddots & \vdots \\
e^{-\gamma_1 x} & e^{-\gamma_2 x} & \cdots & e^{-\gamma_N x}
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2 \\
\vdots \\
c_N
\end{bmatrix}
= 
\begin{bmatrix}
e^{-\gamma_1 x} & e^{-\gamma_2 x} & \cdots & e^{-\gamma_N x} \\
e^{-\gamma_1 x} & e^{-\gamma_2 x} & \cdots & e^{-\gamma_N x} \\
\vdots & \vdots & \ddots & \vdots \\
e^{-\gamma_1 x} & e^{-\gamma_2 x} & \cdots & e^{-\gamma_N x}
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2 \\
\vdots \\
c_N
\end{bmatrix}
\quad (26)
\]

where \( C_1 \) and \( C_2 \) are constant vectors, which can be determined from the terminal currents and voltages (i.e. at \( x = 0 \) and \( x = d \)).

A relationship between the near-end (\( x=0 \)) and far-end (\( x=d \)) voltages can be derived using (27) as

\[
V(x) = W[E(x)]C_1 + W[E(x)]^{-1}C_2
\quad (27)
\]

Next, substituting (27) in (16), we have

\[
-W\Gamma[E(x)]C_1 + W[T[E(x)]]^{-1}C_2 = -ZI(x) ; \quad \Gamma = \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_N \end{bmatrix}
\quad (29)
\]

or

\[
I(x) = W_i[E(x)]C_1 - W_i[E(x)]^{-1}C_2 ; \quad W_i = Z^{-1}F\Gamma
\quad (30)
\]

A relationship between the near-end (\( x=0 \)) and far-end (\( x=d \)) can be derived using (30) as
Using (28) and (31) and eliminating the constants, $C_1$ and $C_2$, we get the representation in terms of y-parameters as

$$\begin{bmatrix} I(0) \\ I(d) \end{bmatrix} = \begin{bmatrix} W_i & -W_i \\ W_iE(d) & -W_i[E(d)]^{-1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

(31)

Assume that the multiconductor stamp is required in the standard form shown in Fig. 3. In this case, to account for the current $I(d)$ as flowing inwards, the expression for $I(d)$ in (32) must be multiplied by -1. Noting this and simplifying (32) further, we can write the MTL stamp in terms of y-parameters as

$$\begin{bmatrix} I(0) \\ I(d) \end{bmatrix} = \begin{bmatrix} W_i & -W_i \\ W_iE(d) & -W_i[E(d)]^{-1} \end{bmatrix} \begin{bmatrix} W & W \\ W[E(d)]^{-1} & W[E(d)]^{-1} \end{bmatrix}^{-1} \begin{bmatrix} V(0) \\ V(d) \end{bmatrix}$$

(32)

Assume that the multiconductor stamp is required in the standard form shown in Fig. 3. In this case, to account for the current $I(d)$ as flowing inwards, the expression for $I(d)$ in (32) must be multiplied by -1. Noting this and simplifying (32) further, we can write the MTL stamp in terms of y-parameters as

$$\begin{bmatrix} I(0) \\ I(d) \end{bmatrix} = \begin{bmatrix} W_iE_1W^{-1} & W_iE_2W^{-1} \\ W_iE_2W^{-1} & W_iE_1W^{-1} \end{bmatrix} \begin{bmatrix} V(0) \\ V(d) \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V(0) \\ V(d) \end{bmatrix}$$

(33)

where

$$E_1 = \text{diag}\left\{\frac{(-Y_{kd}) + (Y_{kd})}{2}\right\}, \quad E_2 = \text{diag}\left\{\frac{(-Y_{kd}) - (Y_{kd})}{2}\right\}; \quad (k = 1, 2, \ldots, N)$$

(34)

![Fig. 3. Multiconductor transmission line system](image-url)
Matrix Exponential Stamp

An alternative form of the MTL stamp is also quite popular and it has the matrix exponential form\(^70\), which is explained below. Equations (6) and (7) can be written in the hybrid form as

\[
\frac{d}{ds} \begin{bmatrix} V(x, s) \\ I(x, s) \end{bmatrix} = (D + sE) \begin{bmatrix} V(x, s) \\ I(x, s) \end{bmatrix}. \quad D = \begin{bmatrix} 0 & -R \\ -G & 0 \end{bmatrix}; \quad E = \begin{bmatrix} 0 & -L \\ -C & 0 \end{bmatrix} \tag{35}
\]

Solution of (35) can be written as

\[
\begin{bmatrix} V(d, s) \\ I(d, s) \end{bmatrix} = e^{(D + sE)d} \begin{bmatrix} V(0, s) \\ I(0, s) \end{bmatrix} \tag{36}
\]

A relationship between the forms represented by (33) and (36) can be obtained as follows: Define \( T(s) \) as

\[
T(s) = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} = e^{(D + sE)d} \tag{37}
\]

Using some algebraic manipulations, we can express the relationships between the hybrid parameters (36) and the y-parameters (33) as

\[
\begin{bmatrix} I(0) \\ I(d) \end{bmatrix} = \begin{bmatrix} -T_{12}^{-1}T_{11} & T_{12}^{-1} \\ T_{21}T_{22}T_{12}^{-1}T_{11} & T_{22}T_{12}^{-1} \end{bmatrix} \begin{bmatrix} V(0) \\ V(d) \end{bmatrix} \tag{38}
\]

\[
\begin{bmatrix} V(d) \\ I(d) \end{bmatrix} = \begin{bmatrix} Y_{12}^{-1}Y_{11} & -Y_{12}^{-1} \\ Y_{21} + Y_{22}Y_{12}^{-1}Y_{11} & -Y_{22}Y_{12}^{-1} \end{bmatrix} \begin{bmatrix} V(0) \\ I(0) \end{bmatrix} \tag{39}
\]

Similarly, another useful representation of the MTL stamp is in terms of ABCD parameters, which can be written as
In the next section, we will review a generic formulation of distributed interconnect circuit equations, suitable for general purpose circuit simulators.

3. Formulation of Circuit Equations

The MNA and output equations for lumped linear networks can be written using a generic notation as

\[
\begin{bmatrix}
T_{11} & -U \\
T_{21} & 0
\end{bmatrix}
\begin{bmatrix}
V(0) \\
V(d)
\end{bmatrix}
+ \begin{bmatrix}
T_{12} & 0 \\
T_{22} & -U
\end{bmatrix}
\begin{bmatrix}
I(0) \\
I(d)
\end{bmatrix}
= \begin{bmatrix}
0 \\
0
\end{bmatrix}
\]  \hspace{1cm} (40)

In the next section, we will review a generic formulation of distributed interconnect circuit equations, suitable for general purpose circuit simulators.

3. Formulation of Circuit Equations

The MNA and output equations for lumped linear networks can be written using a generic notation as

\[
Wx(t) + Gx(t) = Bu(t)
\]

\[
y = L^T u(t)
\]  \hspace{1cm} (41)

where \( B \) and \( L \) are selector matrices, with entries (0 or 1). The superscript ‘\( T \)’ denotes the transpose. Let \( b(t) = Bu(t) \). From (41), MNA equations in the frequency-domain can be written as

\[
(G + sW)X(s) = b(s)
\]

\[
Y(s) = L^T U(s)
\]  \hspace{1cm} (42)

For the case of nonlinear elements, MNA equations in (41) can be modified as

\[
Wx(t) + Gx(t) + F(x(t)) - b(t) = 0
\]

\[
y = L^T u(t)
\]  \hspace{1cm} (43)

where \( F(x(t)) \) is a nonlinear function of \( x \).

A. Formulation of linear subnetworks containing Distributed Elements
Consider a linear subnetwork \( \pi \) containing distributed elements. Using (33), the frequency-domain equations of a distributed subnetwork containing \( n_d \) coupled conductors can be written as \(^{61}\)

\[
Y_{d}(s)V_{d}(s) = I_{d}(s)
\]

(44)

where \( V_{d}(s) \) and \( I_{d}(s) \) represent the Laplace-domain terminal voltages and currents of the distributed element, respectively, \( Y_{d}(s) \) represents the admittance matrix having complex dependency on frequency, which are described in terms of line parameters. Equation (42) representing the lumped linear network can be combined with (44) as

\[
\begin{bmatrix}
G_{\pi} & L_{d}
\end{bmatrix}
\begin{bmatrix}
V_{\pi}(s)
\end{bmatrix}
= \begin{bmatrix}
W_{\pi}
\end{bmatrix}
\]

(45)

where

- \( W_{\pi} \in \mathbb{R}^{N_{\pi} \times N_{\pi}} \) are constant matrices describing the lumped memory and memoryless elements of subnetwork \( \pi \), respectively, \( \mathbb{R}^{N_{\pi}} \) is the node-space of subnetwork \( \pi \),
- \( L_{d} \) is the selector matrix which maps the terminal currents of the distributed subnetwork to the nodal space of the linear subnetwork \( \pi \).
- \( b_{\pi} \in \mathbb{R}^{N_{\pi}} \) is a constant vector with entries determined by independent voltage/current sources of subnetwork \( \pi \), \( V_{\pi}(s) \in \mathbb{R}^{N_{\pi}} \) is the vector of node voltage waveforms appended by independent voltage source currents, linear inductor current waveforms of linear subnetwork \( \pi \).

The equation (45) can be concisely written as

\[
\psi(s)X(s) = b(s)
\]

(46)
B. Generic formulation of nonlinear circuits with distributed elements

Consider a general network containing arbitrary number of nonlinear and linear (lumped and distributed) components. For simplicity, let the linear components be grouped into a single linear subnetwork $\pi$ as shown in Fig. 4.

Using (43), without loss of generality, the circuit equations for the network $\phi$ can be written as

$$W_\phi \frac{d}{dt} x_\phi(t) + G_\phi x_\phi(t) + L_\pi l_\pi(t) + F(x_\phi(t)) - b_\phi(t) = 0, \quad t \in [0, T]$$  \hspace{1cm} (47)

where

- $W_\phi, G_\phi \in \mathbb{R}^{N_\phi \times N_\phi}$ are constant matrices describing the lumped memory and memoryless elements of network $\phi$, respectively, $b_\phi \in \mathbb{R}^{N_\phi}$ is a constant vector with entries determined by the independent voltage and current source,
- $F(x_\phi)$ is a function describing the nonlinear elements of the circuit,
- $x_\phi(t) \in \mathbb{R}^{N_\phi}$ is the vector of node voltage waveforms appended by independent voltage source current, linear inductor current, nonlinear capacitor charge and nonlinear inductor flux waveforms, $N_\phi$ is the total number of variables in the MNA formulation and $n_\pi$ is the total number of ports for linear subnetwork $\pi$.
- $l_\pi = [l_{i,j}]$ with elements $l_{i,j} \in \{0, 1\}$ where $i \in \{1, \ldots, N_\phi\}, j \in \{1, \ldots, n_\pi\}$ with a maximum of one nonzero in each row or column, is a selector matrix that maps $l_\pi(t) \in \mathbb{R}^{n_\pi}$ the vector of currents entering the linear subnetwork $\pi$, into the node space $\mathbb{R}^{N_\phi}$ of the network $\phi$.

The linear multi-terminal subnetwork $\pi$ can be characterized in the frequency-domain by its terminal behavior as

$$Y_\pi(s) V_\pi(s) = I_\pi(s)$$  \hspace{1cm} (48)
where $Y_\pi(s)$ is the y-parameter matrix of subnetwork $\pi$, $V_\pi(s)$ is the vector of terminal voltage nodes that connect the subnetwork to the network $\phi$, $I_\pi(s)$ is the Laplace transform ($i_\pi(t)$).

Fig. 4. Nonlinear network $\phi$ containing linear subnetwork $\pi$ with distributed elements

### 3.1 Interconnect simulation issues

Simulation of large interconnect networks is associated with two major bottlenecks: Mixed frequency/Time problem and CPU expense.

**Mixed frequency/Time Problem**

The major difficulty in simulating high-frequency models such as distributed transmission lines is due to the fact that, while described in terms of partial
differential equations, they are best represented in the frequency-domain (48). As seen, they do not have a direct representation in the time-domain. On the other hand, nonlinear devices can only be described in the time-domain (47). These simultaneous formulations are difficult to handle by a traditional ordinary differential equation solver such as SPICE\textsuperscript{22, 136 - 147}.

CPU Expense

Frequency-domain simulation of large linear networks is conventionally done by solving (42) or (45) at each frequency point using LU decomposition and forward-backward substitution. For time-domain simulation, integration techniques are used to convert a set of time-domain differential equations into a set of difference equations. For example, application of trapezoidal rule to (43), leads to a nonlinear set of difference equation\textsuperscript{147}

\[
\left(G + \frac{2}{\Delta t} W\right)v(t + \Delta t) + F(v(t + \Delta t)) = \left(\frac{2}{\Delta t} W - G\right)v(t) + b(t) + b(t + \Delta t) - Fv(t)
\]  

(49)

To solve (49) at each time point, Newton iterations are required, which may need several LU decompositions. This causes (note that $W$ and $G$ matrices for interconnect networks are usually very large) the CPU cost of a time-domain analysis to be expensive.

The objectives of interconnect simulation algorithms are to address both mixed frequency/time problem as well as to handle large linear circuits without too much of CPU expense. There have been several algorithms proposed for this purpose, which are broadly classified into two main categories, as follows. (a) Approaches based on macromodelling each individual transmission line set. Techniques such as “method of characteristics” are grouped in this category and are discussed in Section 4. (b) Approaches based on model-order reduction (such as AWE, CFH, PRIMA) of the entire linear subnetwork containing lumped as well as distributed subnetworks and are discussed in Sections 5-7. It is to be noted that the second approach can also be used in conjunction with the first approach.

In this approach, transmission-line networks described by Telegrapher’s equations (partial differential equations) are translated into a set of ordinary differential equations (known as the macromodel), through some kind of discretization.

The conventional approach\cite{11,35} for discrete modelling of distributed interconnects is to divide the line into segments of length $\Delta x$, chosen to be small fraction of the wavelength. If each of these segments (assume that the line is discretized into $M$ segments) are electrically small at the frequencies of interest (i.e. $\Delta x = L/M \approx \lambda$), then each segment can be replaced by lumped models. Generally lumped structures used to discretize MTL equations contain, the series elements $L(x)\Delta x$ and $R(x)\Delta x$, and shunt elements: $G(x)\Delta x$ and $C(x)\Delta x$. ($L(x), R(x), G(x), C(x)$ are the per unit length inductance, resistance, conductance and capacitance of the line, respectively)

\textbf{Distributed v/s Lumped: Number of Lumped Segments Required}

It is often of practical interest to know how many lumped segments are required to reasonably approximate a distributed model. For the purpose of illustration, consider LC segments, which can be viewed as low pass filters. For a reasonable approximation, this filter must pass at least some multiples of the highest frequency $f_{max}$ of the propagating signal (say ten times, $f_0 \geq 10f_{max}$). In order to relate these\cite{2,3}, we make use of the 3-dB pass band of the LC filter given by

$$f_0 = \frac{1}{\pi \sqrt{LdCd}} = \frac{1}{\pi \tau d} \quad (50)$$
where \(d\) is the length of the line. From (1), we have \(f_{\text{max}} = 0.35/t_r\) and using (50), we can express the relation \(f_0 \geq 10f_{\text{max}}\) in terms of the delay of the line and the rise time as \(\frac{1}{\pi td} \geq 10 \times 0.35/t_r\), or

\[
t_r \geq 3.5(\pi \tau d) = 10 \tau d
\]  

(51)

In other words, delay allowed per segment is \(0.1t_r\). Hence the number of segments \((N)\) is given by

\[
N = (10\tau d)/t_r
\]  

(52)

In the case of RLC segments, in addition to satisfying (51), the series resistance of each segment must also be accounted. The series resistance \(R_d\) representing the ohmic drop should not lead to impedance mismatch which can result in excessive reflection within the segment.\(^2\)\(^3\)

However, one of the major drawbacks of the above conventional discretization is that it requires large number of sections, especially for circuits with high-operating speeds and sharper rise times. This leads to large circuit sizes and the simulation becomes CPU inefficient. In order to overcome these difficulties, several techniques for efficient discretization are proposed in the literature. These methods can be broadly classified, based on the passivity property as follows. (1) Macromodels with no guarantee of passivity: A sample of such techniques is the Method of Characteristics (2) Macromodels with guaranteed passivity by construction macromodels: A sample of such techniques are Integrated Congruent Transform and Exponential Padé based Matrix-Rational Approximation.

4.1 Method of Characteristics

The method of characteristics (MC)\(^4\)\(^1\)\(^-\)\(^3\) transforms partial differential equations of a transmission line into ordinary differential equations containing time-delayed controlled sources.
Consider the case of two conductor transmission lines, as shown in Fig. 5a. An analytical solution, in terms of $y$-parameters for (6) or (7) can be derived as

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = Y V = \frac{1}{Z_0(1 - e^{-2\gamma d})} \begin{bmatrix} 1 + e^{-2\gamma d} & -2e^{-\gamma d} \\ -2e^{-\gamma d} & 1 + e^{-2\gamma d} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$  \hspace{1cm} (53)$$

where $\gamma$ is the propagation constant on the line, and $Z_0$ is the characteristic impedance. $V_1$ and $I_1$ are the terminal voltage and current at the near end of the line, $V_2$ and $I_2$ are the terminal voltage and current at the far end of the line. The $y$-parameters of the transmission line, are complex functions of $s$, and in most cases cannot be directly transformed into an ordinary differential equation in the time domain. The MC succeeded in doing such a transformation, but only for lossless transmission lines. Although this method was originally developed in the time domain using what was referred to as characteristic curves (hence the name), a short alternative derivation in the frequency domain will be presented here. By re-arranging the terms in (53) we can write,

$$V_1 = Z_0 I_1 + e^{-\gamma d}[2V_2 - e^{-\gamma d}(Z_0 I_1 + V_1)]$$

$$V_2 = Z_0 I_2 + e^{-\gamma d}[2V_1 - e^{-\gamma d}(Z_0 I_2 + V_2)]$$  \hspace{1cm} (54)$$

Next, (54) can be re-written as

$$V_1 - Z_0 I_1 = W_1$$

$$V_2 - Z_0 I_2 = W_2$$  \hspace{1cm} (55)$$

where

$$W_1 = e^{-\gamma d}[2V_2 - e^{-\gamma d}(Z_0 I_1 + V_1)]$$

$$W_2 = e^{-\gamma d}[2V_1 - e^{-\gamma d}(Z_0 I_2 + V_2)]$$  \hspace{1cm} (56)$$

Using (54) and (56), a recursive relation for $W_1$ and $W_2$ can be obtained as
A lumped model of the transmission line can then be deduced from (54) and (57), as in Fig. 5b.

\[ W_1 = e^{-j\beta d}[2V_2 - W_2] \]
\[ W_2 = e^{-j\beta d}[2V_1 - W_1] \]  

(57)

If the lines were lossless (in which case the propagation constant is purely imaginary; \(\gamma = j\beta\)), the frequency domain expression (57) can be analytically converted into time-domain using inverse Laplace transform as

\[ w_1(t + \tau) = 2v_2(t) - w_2(t) \]
\[ w_2(t + \tau) = 2v_1(t) - w_1(t) \]  

(58)

### 4.2 Exponential Padé based Matrix-Rational Approximation

This algorithm directly converts partial differential equations into time-domain macromodels based on Padé rational approximations of exponential matrices. In this technique coefficients describing the macromodel are computed \textit{a priori} and analytically, using closed-form Padé approximant of exponential matrices. Since closed-form relations are used, this technique doesn’t suffer from the usual ill-conditioning experienced with the direct
application of Padé approximations. Hence it allows higher-order of approximation. Also it guarantees the passivity of the resulting Macromodel.

**Matrix-Rational Approximation**

Consider the exponential form of Telegrapher’s equations describing the multiconductor transmission lines, given by (36),

\[
\begin{bmatrix}
V(d, s) \\
I(d, s)
\end{bmatrix} = e^Z \begin{bmatrix}
V(0, s) \\
I(0, s)
\end{bmatrix}; \quad Z = (D + sE)d;
\]

\[
D = \begin{bmatrix}
0 & -R \\
-G & 0
\end{bmatrix}; \quad E = \begin{bmatrix}
0 & -L \\
-C & 0
\end{bmatrix}
\]

(59)

where \(d\) is the length of the line. The matrix \(e^Z\) is approximated using matrix-rational function as

\[
P_{N,M}(Z) e^Z = Q_{N,M}(Z)
\]

(60)

where \(P_{N,M}(Z)\) and \(Q_{N,M}(Z)\) are polynomial matrices expressed in terms of closed-form Padé rational functions\(^{37}\) as

\[
P_{N,M}(Z) = \sum_{j=0}^{N} \frac{(M+N-j)!N!}{(M+N)!j!(N-j)!} (-Z)^j
\]

\[
Q_{N,M}(Z) = \sum_{j=0}^{M} \frac{(M+N-j)!M!}{(M+N)!j!(M-j)!} Z^j
\]

(61)

Several recursive relationships exist for \(P_{N,M}(Z)\) and \(Q_{N,M}(Z)\) such as
The fact that the coefficients $P$ and $Q$ are known a priori in closed-form, provides substantial computational advantage for the proposed algorithm.

The Padé rational function of (61) for $M = N = n$ can be represented in terms of subsections, obtained by the pole-zero pairs as

$$P_{N,M}(Z) = P_{N-1,M}(Z) + Z\left(\frac{-M}{(M+N)(M+N-1)}\right)P_{N-1,M-1}(Z)$$

$$P_{N,M}(Z) = P_{N,M-1}(Z) + Z\left(\frac{N}{(M+N)(M+N-1)}\right)P_{N-1,M-1}(Z)$$

$$Q_{N,M}(Z) = Q_{N-1,M}(Z) + Z\left(\frac{-M}{(M+N)(M+N-1)}\right)Q_{N-1,M-1}(Z)$$

$$Q_{N,M}(Z) = Q_{N,M-1}(Z) + Z\left(\frac{N}{(M+N)(M+N-1)}\right)Q_{N-1,M-1}(Z)$$

The fact that the coefficients $P_{N,M}(Z)$ and $Q_{N,M}(Z)$ are known a priori in closed-form, provides substantial computational advantage for the proposed algorithm.

The Padé rational function of (61) for $M = N = n$ can be represented in terms of subsections, obtained by the pole-zero pairs as

$$[P_{n,n}(Z)]^{-1}Q_{n,n}(Z) = \prod_{i=1}^{\frac{n}{2}} [(a_iU - Z)(a_i^*U - Z)]^{-1}[(a_iU + Z)(a_i^*U + Z)]$$

for even values of $n$, and

$$[P_{n,n}(Z)]^{-1}Q_{n,n}(Z) = \prod_{i=1}^{\frac{(n-1)}{2}} [(a_iU - Z)(a_i^*U + Z)]^{-1}[(a_iU + Z)(a_i^*U + Z)]$$

for odd values of $n$, where $U$ represents the unity matrix, $a_i = x_i + jy_i$ are the complex roots for $i > 0$ and $a_0$ is a real root. The symbol $^*$ represents the complex conjugate operation. It is to be noted that the polynomial $P_{N,M}(Z)$ is a strict Hurwitz polynomial. This means that the coefficients $a_i$ and $a_0$ in (7) and (8) are positive definite constants. The hybrid stencil for such an $i^{th}$ subsection can be written as
Proof of passivity of the macromodel can be found in 37. Also the extension of the closed-form matrix-rational approximation based technique to handle frequency-dependent parameters can be found in 38.

\[
\begin{bmatrix}
P_{n,n}(Z)_{i+1}
\end{bmatrix} = \begin{bmatrix}
V^i_{i+1}
I^i_{i+1}
\end{bmatrix} = \begin{bmatrix}
Q_{n,n}(Z)
\end{bmatrix} \begin{bmatrix}
V^i
I^i
\end{bmatrix}
\] (65)

**Delay Extraction**

Although algorithms based on method of characteristics (MC) provide fast solutions for long low loss lines, they do not guarantee the passivity of resulting macromodels. This was substantiated by several numerical tests which showed that the MC can lead to non-passive macromodels. Moreover, it has been widely shown in the literature that, the transient analysis of a non-passive macromodel with other passive circuit elements may lead to spurious oscillations. On the other hand, algorithms based on matrix rational-function approximations (MRA) guarantee the passivity of macromodels. However, in the presence of large delay lines (e.g. long lines with small losses), this may require high-order approximations (to accurately capture the flat delay portion) leading to inefficient transient simulation.

To address the above issues, a new algorithm for passive macromodeling of transmission line subnetworks was recently introduced. It employs a mechanism for delay reduction prior to performing the matrix rational approximation. The new algorithm leads to significantly lower order macromodels for long lossy coupled lines. A brief discussion of the algorithm is given below.

Using perturbation and assuming that \(|A| \ll s_{max}B|\) (where \(s_{max}\) corresponds to the maximum frequency of interest), (36) can be approximated as

\[
e^{(A + iB)} \approx e^{B} \prod_{k=1}^{m} e^{C_k}; \quad (C_k = f(A, B))
\] (66)

where \(\|C_1\| > \|C_2\| > \ldots > \|C_m\|\). It is to be noted that, the Lie product formula.
provides a systematic alternative approach to obtain (66) with an error estimation \(\varepsilon_m\) and is given by

\[
e^{(sA + sB)} = \prod_{k=1}^{m} P_k + \varepsilon_m; \quad P_k = e^{sB/m} e^{A/m}; \quad |\varepsilon_m| \leq O\left(\frac{1}{m}\right)
\]  

(67)

Next, a theorem is introduced which enables more accurate delay extraction by modifying the Lie product formula given in (67).

**Theorem:** The product

\[
\prod_{k=1}^{m} Q_k + \varepsilon_m; \quad Q_k = e^{sB/m} e^{A/m} 
\]  

(68)

converges asymptotically to \(e^{(sA + sB)}\) as \(m \to \infty\). The associated error \(\varepsilon_m\) in this case is given by

\[
|\varepsilon_m| = \max_{0 \leq s \leq s_{\text{max}}} \left| e^{(sA + sB)} - \prod_{k=1}^{m} Q_k \right| \leq O\left(\frac{1}{m^2}\right)
\]  

(69)

Equation (68) henceforth is referred to as Modified Lie Formula - I. If \(|A| \ll |B|\) (which is the case for long low lossy lines), then an alternative form for (68) can be used with a reduced error, and is given by (referred to as Modified Lie Formula - II):

\[
\prod_{k=1}^{m} Q_k + \varepsilon_m; \quad Q_k = e^{sB/m} e^{A/m} 
\]  

(70)

Also it can be proved that, average of the approximations in (68) and (70) as given by (referred to as Modified Lie Formula - III):

\[
\prod_{k=1}^{m} Q_k + \varepsilon_m; \quad Q_k = \frac{1}{2} \left( e^{sB/m} e^{A/m} + e^{sB/m} e^{A/m} \right) 
\]  

(71)

further reduces the error (note that exact error estimates of each of these formulae are derived, however not given here due to lack of space). Fig. 6 demonstrates numerically, the accuracy comparisons of Lie’s formula and proposed approximations (Modified Lie Formula - I, II and III), for a typical set of line parameters.
Frequency-Dependent Parameters

In the case of frequency-dependent parameters, it is ideal to extract a maximum delay, without affecting the transmission line causality conditions. Let $B(s) = B(s) - B_{\text{max}}$. In general a logical choice for $B_{\text{max}}$ is $B(\infty)$. Next, (68), (70) can be modified as:

$$m \prod_{k=1} Q_k + e_m; \quad Q_k = e^{\frac{A(s) + s\hat{B}(s)}{2m} + \frac{sB_{\text{max}}}{e^m} + \frac{A(s) + s\hat{B}(s)}{2m}}$$

$$m \prod_{k=1} Q_k + e_m; \quad Q_k = e^{\frac{sB_{\text{max}}}{e^m} + \frac{A(s) + s\hat{B}(s)}{2m}}$$

**Implementation:** The products represented by (72) or (73) can be viewed as a cascade of $m$ transmission lines. In addition, each of the $k^{th}$ product term can be viewed as a cascade of lossy and lossless transmission lines. The lossy terms are macromodeled using the passive matrix rational approximation. The resulting macromodels are of significant lower orders (since a significant delay portion is already extracted from these terms). They are later combined with the lossless terms using the method of characteristics approach. For example, each $Q_k$ in (73) can be realized as shown in Fig. 7. It is to be noted that passivity of the entire macromodel is now guaranteed as the passivity of each sub-line in (73) is preserved.

Based on the knowledge of the norm of $A$ and $B$ line parameter matrices and the maximum frequency of interest, approximation represented by (72) or (73) is selected and required order ($m$) satisfying the pre-defined error tolerance can be determined using (69). It is to be noted that for relatively short lossy lines, delay extraction may reduce the efficiency of the MRA macromodel. Based on the knowledge of line parameters and error-estimates of MRA and modified Lie formulae, a criterion has been developed to select the appropriate macromodel (i.e., MRA or MRA with delay extraction).
5. Model-Reduction Based Simulation Algorithms

Interconnect networks generally tend to have large number of poles, spread over a wide frequency range. Even though majority of these poles would normally have very little effect on simulation results, however, they make the simulation to be CPU extensive by forcing the simulator to take smaller step sizes.

Dominant Poles

Dominant poles are in general those, which are close to the imaginary axis and significantly influence the time as well as the frequency characteristics.
of the system. The moment-matching techniques (MMTs)\textsuperscript{57-73} capitalize on the fact that irrespective of the presence of large number of poles in a system, only the dominant poles are sufficient to accurately characterize a given system.

A brief mathematical description of the underlying concepts of moment-matching techniques is given below. Consider a single input/single output system and let $H(s)$ be the transfer function. $H(s)$ can be represented in a rational form as

$$H(s) = \frac{P(s)}{Q(s)}$$  \hspace{1cm} (74)

where $P(s)$ and $Q(s)$ are polynomials in $s$. Equivalently, (74) can be written as

$$H(s) = c + \sum_{i=0}^{N_p} \frac{k_i}{s - p_i}$$  \hspace{1cm} (75)

where $P_j$ and $k_i$ are the $i^{th}$ pole-residue pair, $N_p$ is the total number system poles and $c$ is the direct coupling constant. The time-domain impulse response can be computed in a closed form using inverse Laplace transform as

$$h(t) = c \delta t + \sum_{i=0}^{N_p} k_i e^{p_i t}$$  \hspace{1cm} (76)

In case of large networks, $N_p$, the total number of poles can be of the order of thousands. Generating all the $N_p$ poles will be highly CPU intensive even for a small network and for large networks it is completely impractical. Model-reduction techniques address the above issue by deriving a reduced-order approximation $\hat{H}(s)$ in terms of dominant poles, instead of trying to compute all the poles of a system. Assuming that only $L$ dominant poles were extracted which give a reasonably good approximation to the original system, equation (74) and the corresponding approximate frequency and time responses can be written as
Moments of the response

Consider the Taylor series expansion of a given transfer-function, \( H(s) \), at point, \( s = 0 \),

\[
H(s) \approx \hat{H}(s) = \hat{H}(s) = \hat{H}(s) + s \frac{(H(s))^{(1)}}{1!} + s^2 \frac{(H(s))^{(2)}}{2!} + \ldots + s^n \frac{(H(s))^{(n)}}{n!}
\]

(79)

where the super-script \( (n) \) denotes the \( n^{th} \) derivative. Using a simpler notation, we can re-write(79) as

\[
H(s) \approx \hat{H}(s) = m_0 + m_1(s) + m_2s^2 + \ldots + m_n s^n = \sum_{i=0}^{n} m_i; \quad m_i = \frac{H(s)^{(i)}}{i!}
\]

(80)

The coefficients of Taylor series expansion, \( (m_i) \) are also identical to the time-domain moments of the impulse response \( h(t) \). This can be easily seen by using the inverse Laplace transform of \( h(t) \)

\[
H(s) = \int_0^\infty h(t)e^{-st}dt = \int_0^\infty h(t)[1-st+s\frac{t^2}{2!}+\ldots]dt
\]

\[
= \int_0^\infty h(t)dt + s\int_0^\infty (-1)^i h(t)dt + s^2\frac{1}{2!} \int_0^\infty h(t)dt + \ldots = \sum_{i=0}^{\infty} s^i \frac{(-1)^i}{i!} \int_0^\infty h(t)dt
\]

(81)

Due to this analogy, the coefficients of Taylor series expansion, \( (m_i) \), are generally referred as moments.

It has been shown that the moments provide an estimation of delay and rise times.\(^{55,56}\) Elmore delay,\(^{55}\) which approximates the mid-point of the
monotonic step response waveform by the mean of the impulse response, essentially matches the first moment of the response. This can be considered as one of the basic forms of approximation. However, in order to get accurate prediction of interconnect effects, it is essential that the reduced-order model must match (or preserve) as many moments as possible.

Several algorithms can be found in the literature for reduction of large interconnect subnetworks\textsuperscript{57-90}. They can be broadly classified into two categories: (1) approaches based on explicitly matching the moments to a reduced-order model, (2) approaches based on implicitly matching the moments. The techniques such as AWE belong to the first category and are discussed in Section 6. Techniques such as PVL, PRIMA, which are based on Krylov subspace formulation, belong to the second category and are discussed in Section 7.

6. Model-reduction based on explicit moment-matching

These techniques employ Padé approximation, based on explicit moment-matching to extract the dominant poles and residues of a given system\textsuperscript{57-61}.

6.1 Padé Approximation

Consider a system-transfer function $H(s)$ which is approximated by a rational function $\hat{H}(s)$ as

$$
H(s) \approx \hat{H}(s) = \frac{a_0 + a_1 s + a_2 s^2 + \ldots + a_L s^L}{1 + b_1 s + \ldots + b_M s^M} = \frac{P_L(s)}{Q_M(s)} \quad (82)
$$

where $a_0, \ldots, a_L, b_1, \ldots, b_M$ are the unknowns (total of $L + M + 1$ variables).

Consider the Taylor series expansion of $H(s)$ at $(s = 0)$, in terms of moments and match it to the rational function approximation given in (82) (hence is the name, moment-matching techniques (MMTs), which is also known as Padé approximation) as follows:

$$
\frac{a_0 + a_1 s + a_2 s^2 + \ldots + a_L s^L}{1 + b_1 s + \ldots + b_M s^M} = m_0 + m_1 s + m_2 s^2 + \ldots + m_{L+M} s^L \quad (83)
$$
Cross multiplying and equating the coefficients of similar powers of $s$ starting from $s^{L+1}$ to $s^{L+M}$ on both sides of (83), we can evaluate the denominator polynomial coefficients as

$$
\begin{bmatrix}
m_{L-M+1} & m_{L-M+2} & \cdots & m_L \\
m_{L-M+2} & \cdots & \cdots & m_{L+1} \\
\vdots & \vdots & \vdots & \vdots \\
m_L & m_{L+1} & \cdots & m_{L+M-1}
\end{bmatrix}
\begin{bmatrix}
b_M \\
b_{M-2} \\
\vdots \\
b_1 \\
b_0 
\end{bmatrix}
= 
\begin{bmatrix}
m_{L+1} \\
m_{L+2} \\
\vdots \\
m_{L+M} 
\end{bmatrix}
(84)
$$

The numerator coefficients can be found by equating the remaining powers of $s$ (from $s^0$ to $s^L$) as

$$a_0 = m_0$$

$$a_1 = m_1 + b_1 m_0$$

$$\vdots$$

$$a_L = m_L + \sum_{i=1}^{\min(L, M)} b_i m_{L-i}$$

Equations (84) and (85) yield an approximate transfer function in terms of rational polynomials. Alternatively, an equivalent pole-residue model can be found as follows. Poles $p_i$ are obtained by applying a root-solving algorithm on denominator polynomial $\hat{Q}(s)$. In order to obtain $k_i$, the approximate transfer function given by (77) is expanded using Maclaurin series as

$$\hat{H}(s) = \hat{c} - \sum_{n=0}^{\infty} \frac{L}{n!} \sum_{i=0}^{P_i} \frac{k_i}{\gamma^{n+1}}$$

Comparing $\hat{H}(s)$ from equations (80) and (86), we note that
Residues can be evaluated by writing the equations in (87) in a matrix form as

\[
\begin{bmatrix}
\hat{c} \\
p_1^{-1} \hat{c}^{-1} \\
p_2^{-2} \hat{c}^{-2} \\
\vdots \\
p_{L-1}^{-L-1} \hat{c}^{-L-1}
\end{bmatrix}
\begin{bmatrix}
p_1 & p_2 & \cdots & p_L \\
p_1^{-1} & p_2^{-1} & \cdots & p_L^{-1} \\
p_1^{-2} & p_2^{-2} & \cdots & p_L^{-2} \\
\vdots \\
p_1^{-L-1} & p_2^{-L-1} & \cdots & p_L^{-L-1}
\end{bmatrix}
\begin{bmatrix}
\hat{k} \\
\hat{k}_1 \\
\hat{k}_2 \\
\vdots \\
\hat{k}_{L-1}
\end{bmatrix}
= 
\begin{bmatrix}
M_0 \\
M_1 \\
M_2 \\
\vdots \\
M_{L-1} \\
M_L
\end{bmatrix}
\]  

(88)

In the above equations \( \hat{c} \) represents the direct coupling between input and output. There are more exact ways to compute \( \hat{c} \).  

### 6.2 Computation of Moments

Having outlined the concept of MMTs, we need to evaluate the moments of the system, which are required by (84) - (88). Consider the simple case of lumped circuits and the corresponding MNA equations represented by (42). Expanding the vector \( X(s) \) using Taylor series, we have

\[
\left[ G + sW \right] \left[ M_0 + M_1 s + M_2 s^2 + \ldots \right] = [h] 
\]

(89)

where \( M_i \) represents the \( i^{th} \) moment-vector. Equating coefficients of similar powers of \( s \) on both sides of (89) we obtain the following relationships
Above equations give a closed form relationship for the computation of moments. The moments of a particular output of interest (which are represented by \( m_i \) in equations (82) - (88), are picked from moment-vectors \( M_i \). As seen, (90) requires only one LU decomposition and few forward/backward substitutions during the recursive computation of higher order moments. Since the major cost involved in linear circuit simulation is due to LU decomposition, MMTs yield very high speed advantage (100 to 1000 times) compared to conventional simulators.

**Generalized Computation of Moments**

In the case of networks containing transmission lines, moment-computation is not straight-forward. A generalized relation for recursive computation of higher-order moments can be derived as follows. Consider (46), expanding \( \psi(s) \) and \( X(s) \) in Taylor series, at an expansion point \( s = \alpha \), we have

\[
\left[ \psi(\alpha) + \frac{\psi^{(1)}}{1!}(s - \alpha) + \ldots + \frac{\psi^{(n)}}{n!}(s - \alpha)^n \right] \times \left[ M_0 + M_1(s - \alpha) + \ldots + M_n(s - \alpha)^n \right] = [b]
\]

Where \( \psi^{(n)} \) denotes the \( n^{th} \) derivative of \( \psi(s) \), and \( M_n \) denotes the \( n^{th} \) moment of \( X(s) \) at \( s = \alpha \). Equating coefficients of similar powers of \( s = \alpha \) on both sides of (91), we have

\[
GM_0 = b \\
GM_i = -WM_{i-1} \quad \forall i > 0
\]
Recent Advances in Modeling and Simulation of High-Speed Interconnects

Generalizing (92), a recursive relation for any \( n \)th higher-order moment can be obtained as

\[
[\psi]M_0 = b \\
[\psi]M_n = -\sum_{r=1}^{n} \frac{(\psi^{(r)})M_{n-r}}{r!}
\]  \hspace{1cm} (92)

It can be seen that the coefficient on the left hand side of (93) does not change during higher-order moment computation. Hence it requires only one LU decomposition and \( n \) forward-backward substitutions to compute \( n \) moments. Also it is easy to note that, the lumped networks are a special case of (93) (where \( \psi^{(r)} = 0 \) for \( r \geq 2 \) in which case (93) reduces to the form given by (90)). Relation (93) requires the derivatives of \( \psi \). These can be obtained using (45) as

\[
[\psi]^{(1)} = \begin{bmatrix} W_\pi & 0 \\ Y_d & 0 \end{bmatrix}; \quad [\psi]^{(r)} = \begin{bmatrix} 0 & 0 \\ Y_d & 0 \end{bmatrix} \quad (r \geq 2)
\]  \hspace{1cm} (94)

Using (45), transmission line moments can be computed as

The derivatives \( Y_d^{(r)} \) can be obtained as a function of the derivatives of the entries on the RHS of (38) and proper application of Leibnitz’s theorem. However, this requires the derivatives of the exponential stamp represented (37). A brief review of computation of these derivatives is given below.
Transmission Line Moments

Consider the exponential stamp represented by (37). We wish to expand the exponential matrix in Taylor series, as follows:

\[ e^{(\mathbf{D} + s\mathbf{E})d} = F_0 + F_1 s + \ldots + F_n s^n \]  

(95)

From the property of matrix exponentiation of an arbitrary matrix \( \mathbf{A} \), we have

\[ e^{\mathbf{A}} = 1 + \frac{\mathbf{A}}{1!} + \frac{\mathbf{A}^2}{2!} + \ldots + \frac{\mathbf{A}^n}{n!} \]  

(96)

Let

\[ \mathbf{A} = (\mathbf{D} + s\mathbf{E})d \]  

(97)

Hence (96) can be re-written as

\[ e^{(\mathbf{D} + s\mathbf{E})d} = 1 + \frac{(\mathbf{D} + s\mathbf{E})d}{1!} + \frac{(\mathbf{D} + s\mathbf{E})d)^2}{2!} + \ldots + \frac{(\mathbf{D} + s\mathbf{E})d)^n}{n!} \]  

(98)

Expanding the RHS of (98) further, and collecting the terms in powers of \( s \), we have

\[ e^{(\mathbf{D} + s\mathbf{E})d} = \left[ \mathbf{0} + \frac{\mathbf{D}d}{1!} + \frac{\mathbf{D}^2 d^2}{2!} + \ldots \right] + \]  

\[ s \left[ \frac{\mathbf{E}d}{1!} + \frac{1}{2!} (\mathbf{DE} + \mathbf{ED}) d^2 + \frac{1}{3!} (\mathbf{D}^2 \mathbf{E} + \mathbf{D} \mathbf{E} \mathbf{D} + \mathbf{E} \mathbf{D}^2) d^3 + \ldots \right] + \]  

\[ s^2 \left[ \frac{1}{2!} \mathbf{E}^2 d^2 + \frac{1}{3!} (\mathbf{D}^3 \mathbf{E} + \mathbf{E} \mathbf{D}^2 \mathbf{E} + \mathbf{E}^2 \mathbf{D} \mathbf{E} + \mathbf{E}^3 \mathbf{D}) d^3 + \ldots \right] + \]  

(99)

Equating (95) with (99) gives
Recent Advances in Modeling and Simulation of High-Speed Interconnects

and so on. From the above results, a recursive relationship for generating transmission line moments can be obtained as

\[ F_0 = \frac{I_n}{0!} + \frac{D d}{1!} + \frac{1}{2!} D^2 d^2 + \ldots = F_{0,0} + F_{0,1} + F_{0,2} + \ldots \]

\[ F_1 = \frac{E d}{1!} + \frac{1}{2!}(DE + ED)d^2 + \frac{1}{3!}(D^2 E + DED + ED^2)d^3 + \ldots \]

\[ = F_{1,0} + F_{1,1} + F_{1,2} + \ldots \]

\[ F_2 = \frac{1}{2!} E^2 d^2 + \frac{1}{3!}(D^3 E + EDE^2 + E^2 DE + E^3 D)d^3 + \ldots \]

\[ = F_{2,0} + F_{2,1} + F_{2,2} + \ldots \] (100)

and so on. From the above results, a recursive relationship for generating transmission line moments can be obtained as

\[ F_i = \sum_{j=0}^{\infty} F_{i,j}; \quad F_{i,j} = \frac{(DF_{i,j-1} + EF_{i-1,j})d}{i+j}; \quad i \geq 0, j \geq 0, (i+j) \neq 0 \]

\[ F_{i,j} = 0 \quad (i < 0 \text{ or } j < 0); \quad F_{0,0} = I \] (101)

Convergence of (101), in practice requires 20-30 terms. It is to be noted that the convergence of the series represented by (96) can suffer, if for the first few terms, \( A^n \) grows quicker than \( n! \). In order to control this problem, note that the growth of \( A^n \) depends on its eigenvalues. If all the eigenvalues of \( A \) are within the unit circle in the complex plane, then \( A^n \) will decay with increasing \( n \), leading to fast convergence. From (96) one can see that, the eigenvalues of \( A \) can be controlled by varying the length \( d \). By restricting \( d \) to be small enough, such that the eigenvalues of \( (D + sE)d \) will also be small (over a given frequency range), so as not to cause truncation errors or slow convergence. This can be achieved efficiently, by noting that

\[ e^{(D + sE)d} = e^{\frac{(D + sE)d}{2}} e^{-\frac{(D + sE)d}{2}} \] (102)
In other words, moments of a line can be generated by squaring half-line moments. Let Φ represent the half-line moments, then

\[ e^{(D + sE)d} = F_0 + F_1s + \ldots + F_n s^n \]

\[ = (\Phi_0 + \Phi_1 s + \ldots + \Phi_n s^n)(\Phi_0 + \Phi_1 s + \ldots + \Phi_n s^n) \]

which will give,

\[ F_r = \sum_{i=0}^{r} \Phi_i \Phi_{r-i} \]  \hspace{1cm} (104)

The line can be subdivided by power of 2 (i.e, 2 sections, four sections, 8 sections...) and the moments of the smallest section that meets the convergence requirements are calculated. From these, the moments of the entire line can be recursively calculated with the help of (104).

### 6.3 Limitations of Single Expansion MMT Algorithms

Obtaining a lower-order approximation of the network transfer function using a single Padé expansion is commonly referred as *Asymptotic Waveform Evaluation (AWE)* in the literature. However, due to the inherent limitations of Padé approximants, MMTs based on single expansion often give inaccurate results. The following is a list of those properties which have the most impact on MMTs.

- The matrix in (84) (which is known as Toeplitz matrix) becomes increasingly ill-conditioned as its size increases. This implies that one can only expect to detect 6 to 8 accurate poles from a single expansion.
- Padé often produces unstable poles on the right hand side of the complex plane.
- Padé accuracy deteriorates as we move away from the expansion point.
- Padé provides no estimates for error bounds.

In addition, there is no guarantee that the reduced-model obtained as above is passive. Passivity implies that a network cannot generate more energy than it absorbs, and no passive termination of the network will cause the system to go unstable\(^{81-88}\). The loss of passivity can be a serious problem because
transient simulations of reduced networks may encounter artificial oscillations.

In systems containing distributed elements the number of dominant poles will be significantly higher, and it is very difficult to capture all of them with a single Padé expansion. This lead to the development of multi-point expansion techniques such as complex frequency hopping (CFH), which are summarized in the next section.

6.4 Complex Frequency Hopping

CFH extends the process of moment Matching to multiple expansion points (hops) in the complex plane near or on the imaginary axis using a binary search algorithm. With a minimized number of frequency point expansions, enough information is obtained to enable the generation of an approximate transfer function that matches the original function up to a predefined highest frequency of interest. Using the information from all the expansion points, CFH extracts a dominant pole set as illustrated in Fig. 8(b). In addition, CFH provides an error criterion for the selection of accurate poles and transfer functions.

Selection and Minimization of Hops in CFH

A Padé approximation is accurate only near the point of expansion and its accuracy decreases as we move away from the point of expansion (hop). In order to validate the accuracy of such an approximation, at least two expansion points are necessary. Accuracies of these two expansions can be verified by matching the poles generated at these two hops (referred as pole-matching based approach). Alternatively, the two hops can be verified for their accuracy by comparing the value of the transfer functions due to both these hops at a point intermediate to them (referred as transfer-function based approach). CFH relies on a binary search algorithm to determine the expansion points and to minimize the number of expansions. The steps
A. Transfer Function Based Approach

In this approach the transfer functions obtained at various hops (expansions) are used to ensure the accuracy of the reduced-order model up to the highest frequency of interest. Steps involved in the algorithm are given in Fig. 9 and Fig. 10. It is to be noted that the computational effort needed for a comparison as required by Step 5 is trivial as the responses can be computed in a closed-form using the transfer functions generated in Steps 2 & 3. Here $\varepsilon_{th}$ is a per-defined threshold for the relative error in the transfer functions. At the completion of the binary search algorithm, a set of transfer functions are generated. When evaluating the frequency response at a frequency point $\alpha$, only the transfer function which is valid in the region containing $\alpha$ is used. This is repeated for all other frequency points to obtain the frequency response of the system.
Recent Advances in Modeling and Simulation of High-Speed Interconnects

Fig. 9. Graphical illustration of transfer function based search algorithm

\[ f_H = f_{\text{max}} \]

\[ f_L = 0 \]

\[ f_{\text{mid}} \]

\[ f_{\text{mid}} \]

\[ f_{\text{mid}} \]

\[ H_H(s) \]

\[ H_{\text{mid}}(s) \]

\[ H_L(s) \]

\( jw \) (Imaginary axis)

Real axis

(1st hop)

(2nd hop)

(3rd hop)

midpoints
**B. Pole-Matching Based Approach**

In this approach poles of the transfer function are explicitly evaluated at each hop and the hops are verified for their accuracy by comparing the poles from two adjacent hops using a binary search algorithm. If a matching pole is found between two adjacent expansions, then the binary search is stopped. The distance between the matching pole and the expansion point under consideration defines the radius of accuracy for the corresponding expansion. All the poles which are within the radius of accuracy are treated as accurate poles and are retained in the final pole-set. The poles which are outside the radius of accuracy are considered as inaccurate poles and are
discarded. Once a set of dominant poles are obtained, residues of the system are obtained using (88). Further details of CFH and its search algorithms can be found in 70 and 71.

7. Model-Reduction Based on Krylov-subspace Techniques

The direct moment-matching techniques such as AWE have some disadvantages associated with them. First one among them is the ill-conditioning associated with the moment-matrix. Due to this difficulty, the number of good poles that could be extracted from any expansion point is generally fewer than 10 poles. The second major difficulty is that, they do not guarantee the passivity of reduced-order models. In order to address these difficulties, a parallel class of algorithms, which can be classified as indirect moment-matching techniques were developed 75-86.

These algorithms are based on what is known as Krylov-subspace formulation and Congruent transformation. One of the main features of these algorithms is that they construct the reduced-model based on the extraction of leading eigenvalues (those with the largest magnitude) of a given system (on the contrary, the reduced models from the CFH technique is based on extracting the dominant poles of a given system). In the rest of this section, we will describe the concept and important features of these algorithms.

7.1 Preliminaries

Recall from Section-3, the time-domain MNA and the corresponding output equations can be represented in the form:

\[ Cx(t) + Gx(t) = Bu(t); \quad C, G \in \mathbb{R}^{n \times n}; \quad B \in \mathbb{R}^{n \times 1}; \quad x \in \mathbb{R}^{n \times 1} \]

\[ w = L^Tx(t); \quad L \in \mathbb{R}^{n \times 1} \tag{105} \]

where \( n \) represents the total number of MNA variables. Pre-multiplying both-sides of (105) by \( G^{-1} \), we can write
Taking the Laplace transform of (106), we can write

\[ sAX(s) = X(s) - RU(s) \]

\[ W(s) = L^T X(s) \]  \hspace{1cm} (107)

Rearranging (107), we can write the transfer function \( Y(s) \) of the given system as

\[ Y(s) = \frac{W(s)}{U(s)} = L^T (I - sA)^{-1} R \]  \hspace{1cm} (108)

where \( I \) is an identity matrix.

**A. Why direct Padé based approximation (moment-matrix) is ill-conditioned?**

Consider the transfer-function of a system, as represented by (108). Expanding it in terms of Taylor series, we have

\[ Y(s) = L^T (I + sA + s^2 A^2 + s^3 A^3 + \ldots + s^q A^q) R \]

\[ = \sum_{k=0}^{q} s^k (L^T A^k R) \]

\[ = \sum_{k=0}^{q} s^k m_k \]

where \( m_k = L^T A^k R \)  \hspace{1cm} (109)

Ideally, increasing the order of the Padé approximation (which is equivalent to matching more number of moments), should have given us better approximation results. However, in practice, this is true only up to very limited order, beyond which Padé approximation will not yield any better results\textsuperscript{75, 156}. This can be explained by examining the nature of higher-order
moments, which are given by $m_k = L^T A^k R$. As can be seen, when successive moments are explicitly calculated, they are obtained as powers of $A$. With the increasing values of ‘$k$’, this process quickly converges to an eigenvector corresponding to an eigenvalue of $A$ with the largest magnitude. As a result, for relatively large values of ‘$k$’, the explicitly calculated moments $m_k, m_{k+1}, m_{k+2}, \ldots$ will not add any extra information to the moment-matrix, as all of them contain information only about the largest eigenvalue. In other words, the rows beyond ‘$k$’ of moment-matrix are almost identical (or parallel to each other) making the matrix ill-conditioned.

B. Relationship between eigenvalues and poles of the system

In this section we will show the correspondence between the leading eigenvalues and poles of the system. It is important to understand this concept as the Krylov-subspace based techniques obtain the reduced-models by extracting the leading eigenvalues of a given system. Consider (106), and assume that the matrix $A$ can be diagonalized in the form

$$A = F \lambda F^{-1}$$

(110)

where $\lambda = \text{diag} [\lambda_1, \lambda_2, \ldots, \lambda_n]$ is a diagonal matrix, whose diagonal elements represent the eigenvalues of matrix $A$. The matrix $F$ contains the eigenvectors of matrix $A$. Using (110), the transfer-function represented by (108) can be re-written as

$$Y(s) = L^T (I - sF \lambda F^{-1})^{-1} R$$

$$= L^TF(I - s\lambda)^{-1} F^{-1} R$$

$$= L^TF \begin{bmatrix}
\frac{1}{1-s\lambda_1} \\
& \ddots \\
& & \frac{1}{1-s\lambda_n}
\end{bmatrix} F^{-1} R$$

(111)
which can be simplified as

\[ Y(s) = \sum_{i} \frac{\eta_i}{1 - s \lambda_i} = \sum_{i} \frac{(\eta_i/\lambda_i)}{s - (1/\lambda_i)} = \sum_{i} \frac{k_i}{s - p_i} \]  

(112)

where \( \eta_i \) is a function of eigenvectors of matrix \( A \), \( k_i \) represent the residues. From (112), we can draw following inferences: (a) poles \( p_i \) are the reciprocal of eigenvalues of matrix \( A \); the leading eigenvalues (those with largest magnitudes) correspond to the poles closer to the origin, (b) the transfer function of \( Y(s) \) can be easily obtained in terms of poles and residues, once the eigenvalues and eigenvectors of \( A \) are available.

However, for large interconnect circuits, it would be impractical to compute all the eigenvalues and eigenvectors. Hence in the following sections, we will review some of the efficient techniques to extract leading eigenvalues.

**Computation of Eigenvalues of Matrix ‘A’**

In general, the numerical computation of all the eigenvalues and eigenvectors of a given matrix \( A \) becomes exceedingly expensive as its size gets above few hundreds. The general approach in such cases is to approximate \( A \) with a smaller matrix \( \tilde{A} \), such that the eigenvalues of \( \tilde{A} \) are reasonable approximation of the leading eigenvalues of \( A \). Due to the relatively small size of \( \tilde{A} \), finding its eigenvalues will be a much simpler problem, than finding the eigenvalues of \( A \). Next, we will review some of the basic matrix forms\(^\text{156}\), which would be helpful in understanding the eigenvalue computation algorithms presented in this section.

**Upper Hessenberg Matrix:** A matrix \( H \) is called *Upper Hessenberg* if \( H_{ij} = 0 \) for \( (i > j + 1) \). For example, consider an upper Hessenberg matrix of order \( q \), having the following form (which is known as *companion form*)
One of the important advantages of the above companion form is that, its characteristic polynomial, \( p(x) \), can be analytically computed and is given by

\[
H = \begin{bmatrix}
0 & 0 & 0 & \ldots & 0 & -c_1 \\
1 & 0 & 0 & \ldots & 0 & -c_2 \\
0 & 1 & 0 & \ldots & \vdots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ldots & 0 & \vdots & \vdots \\
\vdots & \vdots & \ldots & 1 & -c_d
\end{bmatrix}
\] (113)

The roots of \( p(x) \) give the eigenvalues of \( H \).

Orthogonal Matrices: A real square matrix \( Q \) is orthogonal if \( Q^{-1} = Q^T \).

Essentially this implies

\[
QQ^T = Q^T Q = I
\] (115)

All columns, \( q_i \) (or rows) of orthogonal matrices have unit two norms or \( \|q_i\|_2 = 1 \) (which implies that \( q_i^T q_i = 1 \)) and are orthogonal to one another (which means that \( q_i^T q_j = 0 \)).

QR decomposition: Let \( K \) be a \( m \times n \) matrix with \( m > n \). Suppose that \( K \) has full column rank. Then there exists unique \( m \times n \) orthogonal matrix \( Q \) and a unique upper triangular matrix \( R \) with positive diagonals \( (r_{ii} > 0) \) such that \( K = QR \). There are several techniques (such as *modified Gram-Schmidt orthogonalization* process) available in the literature, for this purpose 156.

Next, consider the circuit equations (106), and a simple similarity transformation as follows

\[
AK = KH_q
\] (116)
where the transformation matrix $K$ is defined as

$$ K = \begin{bmatrix} R & AR & \ldots & A^q R \end{bmatrix} $$

(117)

and $H_q$ has the upper Hessenberg companion form discussed above. Obviously, since $H_q$ is related to the matrix $A$ through a similarity transformation, its eigenvalues are the same as that of $A$. Although it looks straight-forward, this approach has the following limitations:

Computation of $H_q$ using the relation (116) ($H_q = K^{-1} AK$) requires the inverse of the matrix $K$. However, $K$ is a dense matrix and hence computation of its inverse will be expensive. Also, $K$ is likely to be ill-conditioned since the columns of $K$ are formed based on the sequence $A^j R$, which as shown in Section 7.1, quickly converges to the eigenvector corresponding to the largest eigenvalue. In the next section, we will describe general techniques to overcome these problems. These algorithms belong to a class of methods known as Krylov-subspace techniques.

### 7.2 Krylov-Subspace methods for Iterative Computation of Eigenvalues

We will start by replacing the matrix $K$ in (116) with an orthogonal matrix $Q$ such that for all $q$, the leading $q$ columns of $K$ and $Q$ span the same space. This space is called a Krylov subspace, and is denoted by $K(A, R, q)$. In other words, any vector which is a linear combination of the leading $q$ columns of $K$ can be expressed also as a linear combination of the leading $q$ columns of $Q$. Mathematically we will express this as

$$ K(A, R, q) = \text{ColumnSpace} \left( \begin{bmatrix} R & AR & \ldots & A^q R \end{bmatrix} \right) = \text{ColumnSpace}[Q] $$

(118)

In contrast to matrix $K$, the matrix $Q$ has the following advantages:

- $Q$ is well conditioned,
- It is easy to invert since $Q^{-1} = Q^T$,
Most importantly, we can compute only as many leading columns of \( Q \) as needed to get accurate solution (more details about this is covered later in this section).

The next question is, how do we get the matrix \( Q \)? This can be achieved, by performing QR decomposition on matrix \( K \). Writing \( K = QR \), we can modify (116) as

\[
H_q = K^{-1}AK \\
R = (QR)^{-1}A(QR) \\
= (R^{-1}Q^T)A(QR) \\
Q^T AQ = RH_q R^{-1} = H
\]

(119)

Since \( R \) and \( R^{-1} \) are both upper triangular and \( H_q \) is upper Hessenberg, it is easy to prove that the new matrix, \( H = RH_q R^{-1} \), is also upper Hessenberg. The implications of (119) is that we can reduce the matrix \( A \) of dimension \( n \times n \) to a smaller upper Hessenberg matrix \( H \) of dimension \( q \times q \) using orthogonal transformation. In addition, the eigenvalues of smaller system \( H \) are approximations of the first \( q \) leading eigenvalues of larger system represented by \( A \).

Next, we will show that the columns of \( Q \) can be computed one at a time giving us the advantage of computing only as many leading columns of \( Q \) as needed. One of the popular approaches used for partial reduction of a large matrix to a smaller upper Hessenberg matrix by computing \( Q \), is known in the literature as Arnoldi’s algorithm\(^{78-85,156}\). More details about this is given in the next section.

#### 7.3 Arnoldi Algorithm for (partial) Reduction

Assume \( Q = [q_1 \ q_2 \ \ldots \ q_k] \), where \( q_i \) represents the \( i^{th} \) column of matrix \( Q \). From (119) we have
\[ AQ = QH \] (120)

Recall that all columns, \( q_i \) (or rows) of orthogonal matrices have \( \|q_i\|_2^2 = 1 \)
(which implies that \( q_i^T q_i = 1 \)) and are orthogonal to one another (which
means that \( q_i^T q_j = 0 \)). Using this information, the first few steps in obtaining
the \( Q \) and \( H \) matrices are outlined below.

Since the \( \|q_1\|_2 = 1 \), an easy way to compute it, is to divide the vector \( R \) by
its magnitude \( \|R\|_2 \) (we get a unit vector in the direction of \( R \)). This step is
illustrated in Fig. 11a.

\[ q_1 = \frac{R}{\|R\|_2}. \] (121)

To determine \( q_2 \) and the first column of \( H \), we multiply \( A \) by the first
column of \( Q \). This gives us \( Aq_1 \), which is the first column on LHS of (120).
Equating it with the first column of RHS, we have

\[ Aq_1 = h_{11}q_1 + h_{21}q_2 \] (122)

Premultiplying both sides by \( q_1^T \) we have

\[ q_1^T Aq_1 = h_{11}q_1^T q_1 + h_{21}q_1^T q_2 \]

\[ q_1^T Aq_1 = h_{11} \] (123)

Knowing the value of \( h_{11} \) and using the fact that \( \|q_2\| = 1 \), we can compute
\( h_{21} \) from (122) as

\[ h_{21} = \| Aq_1 - h_{11}q_1 \|. \] (124)

The direction for \( q_2 \) can be obtained using (122) as (illustrated in Fig. 11b)

\[ q_2 = \frac{Aq_1 - h_{11}q_1}{h_{21}}. \] (125)
Similarly the rest of the columns of $Q$ and $H$ matrices can be obtained by generalizing the above steps.

Note that we didn’t need to explicitly compute the product $A^TR$. As a result, we were able to avoid the ill-conditioning problem arising due to the quick convergence of the sequence $[R, AR, A^2R, A^3R, \ldots]$ to the eigenvector of the largest eigenvalue.

The columns $q_i$ computed by Arnoldi algorithm are called Arnoldi vectors. The loop over $i$ updating $z$ corresponds to the modified Gram-Schmidt algorithm\footnote{156} which subtracts the components in the directions $q_1$ to $q_i$ away from $z$, leaving them orthogonal to $z$. Computing a total of $k$ Arnoldi vectors costs $k$ matrix-vector multiplications involving $A$, plus $O(k^2n)$ related cost.

There are several alternative methods available in the literature for finding the Krylov-subspace\footnote{83}. For example one can use multiple passes of orthogonalization, to increase the robustness of the modified Gram-Schmidt orthogonalization process.
To recap, we started with the circuit equations \( Cx(t) + Gx(t) = b(t) \) and \( w = L^T x(t) \). We formed the product \( A = G^{-1} C \). Using orthogonal transformation, we were able to determine the leading eigenvalues of \( A \) which correspond to the dominant poles of the transfer-function. In the following section, we will show how to use this information to perform circuit reduction.

### 7.4 Circuit Reduction Using Arnoldi Algorithm

Finding the reduced-order circuit equations can be explained by a change of variables in (105) by mapping the vector \( x \) of dimension \( n \) into a smaller vector \( \hat{x} \) of dimension \( q \) \( (q < n) \) using the orthogonal matrix \( Q \):

\[
x = Q\hat{x}
\]

Using (126) we can re-write Laplace-domain circuit equations in (106) as

\[
sAQ\hat{X}(s) = Q\hat{X}(s) - RU(s)
\]

\[
W(s) = L^T Q\hat{X}(s)
\]

Pre-multiplying both sides of (127) by \( Q^T \) and using the relation \( QQ^T = I \), we have

\[
\hat{X}(s) = (I - sQ^T AQ)^{-1} Q^T RU(s)
\]

\[
W(s) = L^T Q(I - sQ^T AQ)^{-1} Q^T RU(s)
\]

Hence the transfer-function of the reduced system can be written as

\[
\hat{Y}(s) = \frac{W(s)}{U(s)} = L^T Q(I - sQ^T AQ)^{-1} Q^T R
\]

\[
\hat{Y}(s) = L^T Q(I - sH)^{-1} Q^T R
\]

Comparing the original transfer-function \( Y(s) \) represented by (108) with the transfer-function \( \hat{Y}(s) \) of the reduced system represented by (129), we can draw the following conclusions. The eigenvalues of \( \hat{Y}(s) \) are given by the
eigenvalues of $H$. However, since the eigenvalues of $H$ are good approximation of the leading eigenvalues of $A$, we can conclude that the eigenvalues of the transfer function of the reduced system are good approximation of the poles of the original transfer function.

An important criterion during the above reduction is the accuracy of the response of the reduced system given by (129). The frequency response of the reduced system (129) is also a good approximation of the frequency-response of the original transfer function (108). An indicator for the accuracy of the response of the reduced system is the total number of moments it can preserve (match), for a given order of reduction ($q$). It can be proved that the reduced system (129) of order $q$ preserves the first $q$ moments of the original network. 82.

In essence, we are able to implicitly match the moments and obtain a reduced-model without the need to directly use the moments as in the AWE algorithm. Hence we will not suffer from the same numerical ill-conditioning which is associated with direct moment-matching algorithms. The accuracy of the Arnoldi approximation gradually increases as the order $q$ is increased since more moments of the original transfer function will be matched.

A question that may possibly arise here: how are the accuracies of Arnoldi based approximation and direct Padé based approximation are compared? It was shown in section-8 that a Padé approximation of order $q$ matches the first $2q$ moments. However, an Arnoldi based reduction of order $q$ matches only first $q$ moments. 82. Essentially, this means that, for a comparable accuracy, the reduced-model from Arnoldi will have double the size of the reduced model from direct Padé based approximation (in other words, direct Padé based models are more optimal). On the other hand, due to the ill-conditioning, direct Padé based approximation can’t achieve higher-order approximation, where as Arnoldi based approximation can.

8. Related Topics and Further Reading

In addition to the interconnection simulation algorithms discussed here, there are several related topics which may be of interest to the reader.
Passivity Preservation

Passivity implies that a network cannot generate more energy than it absorbs, and no passive termination of the network will make the system unstable. Passivity is an important property, because, stable but not passive macromodels can lead to unstable systems when connected to other passive systems. The loss of passivity can be a serious problem because transient simulations of nonpassive networks may encounter artificial oscillations. This is illustrated in Fig. 12, which represents the transient response of a reduced-order macromodel of a large linear RLC circuit, when connected to an external load of 50kΩ. Several algorithms were proposed in the literature for preservation of passivity during the reduction of interconnect networks.\textsuperscript{81 - 88}

![Fig. 12. Transient response of a nonpassive macromodel with passive terminations](image)

Full-Wave Models

At further sub-nano second rise times, the line cross-section or the nonuniformities become a significant fraction of the wavelength and field
components in the direction of propagation can no longer be neglected. Consequently, full-wave models which take into account all possible field components and satisfies all boundary conditions are required to give an accurate estimation of high-frequency effects. However, circuit simulation of full-wave models is highly involved. The information that is obtained through a full-wave analysis is in terms of field parameters such as propagation constant, characteristic impedance etc. A circuit simulator requires the information in terms of currents, voltages and circuit impedances. This demands a generalized method to combine modal results into circuit simulators in terms of a full-wave stamps. References 25-28,54,72, provide solution techniques and moment generation schemes for such cases.

**Measured data**

In practice, it may not be possible to obtain accurate analytical models for interconnects because of the geometric inhomogeneity and associated discontinuities. To handle such situations modeling techniques based on measured data have been proposed in the literature 97-109. In general, the behavior of high-speed interconnects can easily be represented by measured frequency-dependent scattering parameters or time-domain terminal measurements. However, handling measured data in circuit simulation is a tedious and a computationally expensive process.

**EMI Subnetworks**

Electrically long interconnects function as spurious antennas to pick up emissions from other nearby electronic systems. This makes susceptibility to emissions a major concern to current system designers of high-frequency product. Hence the availability of interconnect simulation tools including the effect of incident fields is becoming an important design requirement. References 115-135 provide analysis techniques for interconnects subjected to external EM interferences and also for radiation analysis of interconnects.

**Sensitivity Analysis**

Sensitivity analysis involving large interconnect subnetworks can be highly CPU intensive. Model-reduction based approaches provide an efficient means for this purpose 110-114.
References

17. A. E. Ruehli and P. A. Brennan, “Efficient capacitance calculations for three


sign, pp. 2107-2116 Mar. 92.


Recent Advances in Modeling and Simulation of High-Speed Interconnects


114. C. Jiao, A. C. Cangellaris, A. Yaghmour and J. L. Prince, “Sensitivity analysis of multiconductor transmission lines and optimization for high-speed in-


