1

PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Analysis of High Speed Interconnects
Using
Transform Methods

by

Linchao Lu, M. Eng.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of

Master of Engineering

Ottawa-Carleton Institute for Electrical Engineering
Department of Electronics
Carleton University
Ottawa, Ontario
September, 1991

©1991, Linchao Lu

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-70904-9

Canada

The undersigned hereby recommend to the Faculty of Graduate Studies and Research
acceptance of the thesis,

**“ANALYSIS OF HIGH SPEED INTERCONNECTS
USING
TRANSFORM METHODS”**

submitted by Linchao Lu, M.E.

in partial fulfillment of the requirements for
the degree of Master of Engineering

Prof. M. S. Nakhla

Thesis Supervisor

Prof. J. S. Wight

Chairman, Department of Electronics

Carleton University

September, 1991

Abstract

As circuit speed increases, delay and crosstalk due to interconnections become the dominant limiting factors for higher speed and higher performance systems. Thus, efficient and reliable numerical methods to evaluate interconnection delay and crosstalk become the essential tools for higher speed VLSI system design.

Numerical inversion of Laplace transform is an efficient approach for time analysis of linear lossy coupled transmission line networks. This method is more accurate and more efficient than FFT-based methods and is absolutely stable.

This thesis proposes two new methods based on Pade approximation. The first method is a high degree numerical inversion of Laplace transform which uses derivatives of output waveforms with respect to the Laplace operator s . The second method is a stepping algorithm which uses a resetting technique to improve the accuracy of the solutions for circuits with relatively long transients. While still maintaining the absolutely stable property of the conventional numerical inversion of Laplace transform, the new methods have proven to be more accurate and more efficient.

Acknowledgment

Firstly, I am grateful to my supervisor Dr. M. Nakhla for his invaluable advice in guiding this research work and completing this thesis, and also his endeavour in reviewing this thesis.

Furthermore, I wish thank Dr. Q.J. Zhang for useful discussions and Mr. D. Piamonte for his help in correcting the English writing of the thesis.

Finally, I want to acknowledge Mr. Richard Griffith, Mr. Roger Shum and Mr. Nelson Liu for their help in running some of the software utilities used in the thesis.

Table of Contents

Acceptance Sheet.....	ii
Abstract.....	iii
Acknowledgements.....	iv
Table of Contents.....	v
List of Tables.....	viii
List of Figures.....	ix
List of Symbols.....	xi
 Chapter 1	 Introduction 1
1.1 Problem Description.....	1
1.2 Thesis Objectives and Contributions.....	3
1.3 Outline of The Thesis.....	4
 Chapter 2.	 Methods for Time-Domain Analysis of Linear Distributed Networks 5
2.1 Introduction.....	5
2.2 The Equations of the Multiconductor Transmission Line.....	5
2.3 Fast Fourier Transform(FFT).....	8
2.4 Generalized Asymptotic Waveform Evaluation(GAWE).....	9
2.5 Numerical Inversion of Laplace Transform(NILT).....	13
(a) Description of NILT Method.....	14
(b) Griffith and Nakhla's Stamps of Transmission Lines for NILT0.....	17
2.6 Comparison.....	22
 Chapter 3.	 High Degree Numerical Inversion of Laplace Transform 24
3.1 Introduction.....	24
3.2. Derivation of The New method.....	25
3.3. The Truncation Error of NILTn.....	33

	3.4. Stability of NILTn.....	41
	3.5. Examples.....	49
Chapter 4	Applications of the High Degree Numerical Inversion of Laplace Transform for Transient Analysis of Distributed Networks	55
	4.1. Introduction.....	55
	4.2. Stamps of Transmission Lines for NILT1 and NILT2.....	56
	4.3. Implementations of NILTn for Distributed Networks.....	64
	4.4. Examples.....	64
	4.5. Comments on NILTn.....	93
Chapter 5	A Stepping Algorithm for the Numerical Inversion of Laplace Transform	95
	5.1. Introduction.....	95
	5.2. The Stepping Algorithm.....	96
	5.3. Extension of The Stepping Algorithm.....	97
	(a) Formulation of The Stepping Algorithm for Lossy Coupled Transmission Lines.....	97
	(b) Derivation of the Equivalent Sources Caused by the Initial Conditions of the kth Transmission Line.....	98
	5.4. Implementations.....	106
	(a) Computation of $g_k(v_k(x, t_0), i_k(x, t_0))$	106
	(b) Solutions of $v_k(x, t_0)$ and $i_k(x, t_0)$	108
	(c) Computation of The Integral Part of (5.4-8).....	109
	(d) CPU Time Considerations.....	111
	(e) Simulation Steps.....	111
	(f) Accuracy.....	112
	5.5. Examples.....	113
	5.6. Comments on The Stepping Algorithm.....	122

Chapter 6	Conclusions and Further Research	123
	6.1 Conclusions.....	123
	6.2 Further Research.....	124
References		125

Lists of Tables

Table 2.1	Comparison of FFT, GAWE and NILT0.....	23
Table 3.1	Comparison of truncation error coefficients of NILT0, NILT1 and NILT2.....	40

Lists of Figures

Fig. 2.1	The representation of a multiconductor transmission line.....6
Fig. 2.2	Voltages and currents of a distributed transmission line as a function of location x8
Figure 3.1	The stability region of NILTn.....48
Figure 3.2	Comparison of accuracy of NILT0, NILT1 and NILT2 for example 3.1.....49
Figure 3.3	RC tree circuit with widely varying time constant.....50
Figure 3.4	Responses by Hspice, NILT0, NILT1 and NILT2 for example 3.2.....51
Figure 3.5	The circuit for example 3.3.....52
Figure 3.6-3.8	Results for example 3.3.....53-54
Figure 4.1	One-single TL circuit for example 4.1.....65
Figure 4.2	Result for example 4.1.....66
Fig. 4.3	Comparison of NILT2 (8 poles) and NILT0 (22 poles) for example 4.1.....66
Figure 4.4	7 single transmission line circuit.....68
Figure 4.5-4.8	Results for example 4.2.....60-70
Fig. 4.9	Comparison of NILT2 (8 poles) and NILT0 (22 poles) for example 4.2.....71
Figure 4.10-4.13	Results for example 4.3.....72-73
Fig. 4.14	Comparison of NILT2 (8 poles) and NILT0 (22 poles) for example 4.3.....74
Figure 4.15	One multiconductor TL circuit for example 4.4.....75
Figure 4.16-4.19	Results for example 4.4.....75-77
Fig. 4.20	Comparison of NILT2 (8 poles) and NILT0 (22 poles) for example 4.4.....77
Figure 4.21-4.24	Results for example 4.5.....78-80
Fig. 4.25	Comparison of NILT2 (8 poles) and NILT0 (22 poles) for example 4.5.....80

Figure 4.26	2 multiconductor TL circuit for example 4.6.....	83
Figure 4.27-4.30	Results for example 4.6.....	84-85
Fig. 4.31	Comparison of NILT2 (8 poles) and NILT0 (22 poles) for example 4.6.....	86
Figure 4.32	3 multiconductor TL circuit for example 4.7.....	87
Figure 4.33-4.34	Results for example 4.7.....	88
Fig. 4.35	Comparison of NILT2 (8 poles) and NILT0 (22 poles) for example 4.7.....	89
Figure 4.36	35-transmission line circuit for example 4.8.....	90
Figure 4.37-4.40	Results for example 4.8.....	91-92
Fig. 4.41	Comparison of NILT2 (8 poles) and NILT0 (22 poles) for example 4.8.....	93
Figure 5.1	The flow of the simulations.....	112
Figure 5.2	Single-lossless transmission line circuit for example 5.1.....	114
Figure 5.3-5.4	Results for example 5.1.....	114-115
Figure 5.5-5.6	Results for example 5.2.....	116
Figure 5.7-5.8	Results for example 5.3.....	117-118
Figure 5.9-5.10	Results for example 5.4.....	119
Figure 5.11-5.13	Results for example 5.5.....	120-121

Lists of Symbols

A	Admittance matrix contributed by resistances and conductances of linear lumped subnetworks
B	Admittance matrix contributed by inductances and capacitances of linear lumped subnetworks
H_k	An admittance matrix contributed by the kth transmission line
m_k	Dimension of the kth coupled transmission line
N_T	Number of transmission lines
D_k	Length of the kth transmission line
F(s)	A frequency domain vector of waveforms of nodal voltages, independent voltage source currents and impedance currents
J(s)	A vector which represents independent current and voltage sources
f(t)	Time domain representation of F(s)
$\hat{f}(t)$	Time domain response obtained using Pade approximation
$\xi_{N,M}(z)$	Pade approximation of e^z
M	Order of the denominator polynomial of Pade approximation
N	Order of the numerator polynomial of Pade approximation
Ψ_0	Truncation error of NILTO
$v_k(x, t)$	Voltage vector of the kth transmission line in the time domain at location x
$i_k(x, t)$	Current vector of the kth transmission line in the time domain at location x
R, G, L and C	Transmission line parameter matrices
$V_k(x, s)$	Voltage vector of the kth transmission line in the frequency domain at location x
$I_k(x, s)$	Current vector of the kth transmission line in the frequency domain at location x
p_k	A selector matrix which maps the vector of currents entering the

	kth transmission line into the node space R^n of the linear distributed network
γ_k^2	Eigenvalue of matrix $Z_k Y_k$
S_v	Eigenvector of matrix ZY associated with γ^2
Γ	Eigenvalue matrix
ψ_n	Truncation error of NILTn
d_i	ith derivative of $f(t)$
$D_{N,M}$	$(N+M+1)$ th derivative of $\xi_{N,M}$
$s_k(v_k(x, t_0), i_k(x, t_0))$	A vector contributed by the initial values of the kth transmission line at $t = t_0$ and location x
Φ	A frequency domain vector consists of voltages and currents of the transmission line
ϕ	The time domain representation of Φ
β_k	Coefficient of the state space equations of the kth transmission line

CHAPTER 1

Introduction

1.1 Problem Description

The present trend towards increased density of packaging in digital circuits and the implementation of high speed devices has led to increasing demands for the characterization of interconnections. Multiconductors embedded in dielectric media have found numerous applications in microwave and digital networks. As frequency or pulse speed increases, the electromagnetic coupling between conductors which are most frequently embedded on print-circuit boards becomes more significant. Signal delays and rise times are increasingly limited by interconnection lengths rather than by device speed and represent a potential obstacle to the ultimate scaling on VLSI technology.

Design of interconnections is especially important after recent advances in integrated circuit technologies and packaging techniques, as they can increase the number of connections on a single board into the tens of thousands. Improperly designed interconnects can result in an increase of signal delay because of losses, inadvertent switching and noise as a result of crosstalk, false switching and ringing due to reflections.

This phenomenon can be observed at both chip and system levels, and the interconnected blocks could be analog, digital or mixed. With subnanosecond rise times, the electrical length of interconnects can become a significant fraction of a wavelength. Consequently, the conventional lumped-impedance interconnect model is not adequate in this situation. Instead, a distributed transmission line model should be used.

First, we are going to consider the lossy coupled transmission line to be uniform along its length. At the ends, the line is terminated by arbitrary linear networks. Secondly, we are going to assume that the line behavior can completely be described in terms of circuit-theory parameters, i.e., in terms of matrices of inductances, capacitances, resistances and conductances per unit length. These parameters are often taken to be constant. We are not, therefore, going to discuss their evaluation, i.e., these parameters are assumed to be known.

With these assumptions, the lossy coupled transmission lines can be described by a system of partial differential equations in the time domain or by a system of ordinary differential equations in the frequency domain.

Using numerical inversion of the Laplace transform (NILT) with Pade approximation, one can solve distributed or lumped and distributed mixed networks in frequency domain [8][9]. Then the time domain responses can be obtained by taking the Laplace inversion of the solutions. The method is accurate for small time intervals [8][9] and is absolutely stable [21].

The disadvantage of NILT is that the results can be inaccurate when time t approaches a certain value. The improvement of accuracy can theoretically be achieved by using more poles of Pade approximation. However, increasing of the number of the poles used is limited by computer roundoff errors which affect both calculation of the poles and

applications of the poles. To present, however, the maximum available number of poles has been 22. Further, using more poles implies taking more time consuming steps such as LU decomposition and backward-forward substitution.

1.2 Thesis Objectives and Contributions

This thesis proposes two new methods to improve the accuracy of the numerical inversion of Laplace transform. They are (1) a high degree of numerical inversion of Laplace transform which uses derivatives of output waveforms with respect to the Laplace operator s and (2) a stepping algorithm for numerical inversion of Laplace transform which uses a resetting technique.

The contributions offered by this thesis are enumerated below:

1. A derivation of a general formula for a high degree numerical inversion of Laplace transform (section 3.2).
2. Two formulae for the numerical inversion of Laplace transforms of degree 1 and degree 2 are derived (section 3.2).
3. A formula of the truncation error of the high degree numerical inversion of Laplace transform is derived, which proves that the high degree numerical inversion of Laplace transform is more accurate than the conventional inversion of Laplace transform (section 3.3).
4. A proof that a high degree numerical inversion of Laplace transform is absolutely stable (section 3.4).
5. Demonstrations that the high degree numerical inversion of Laplace

transform is more efficient than the conventional inversion of Laplace transform (section 4.5).

6. The stamps of the numerical inversion of Laplace transforms of degree 1 and degree 2 for solving distributed networks are derived (section 4.2).

7. A general formula of the stepping algorithm for the numerical inversion of Laplace transform for solving distributed networks is given (section 5.1).

8. An explicit formula for solving the term contributed by the initial conditions of the transmission line, which is required by the general formula, is derived (section 5.3).

1.3 Outline of The Thesis

In chapter 2, a brief review of previous major methods for solving distributed networks is given. Theoretical derivation, truncation error and stability studies of the high degree numerical inversion of Laplace transform are discussed in Chapter 3. Applications of this method for distributed networks are presented in chapter 4. In chapter 5, the derivation and applications of the stepping algorithm for the numerical inversion of Laplace transform are discussed. Finally, a summary and conclusions are given in chapter 6.

CHAPTER 2

Methods for Time-Domain Analysis of Linear Distributed Networks

2.1 Introduction

A number of methods for time analysis of linear distributed networks can be found in recent literature [3] [8][9][22][23]. In this chapter, a short description of the equations of coupled transmission lines are given in section 2.2, followed by a brief review of the fast Fourier transform and the generalized asymptotic waveform evaluation in section 2.3 and section 2.4, respectively. Finally, since the numerical inversion of Laplace transform is the basis of the proposed methods, a description of this method is given in detail in section 2.5.

2.2 The Equations of the Multiconductor Transmission Line

In chapter 1, we assumed that a lossy multiconductor transmission line with an arbitrary cross section is uniform along its length. The cross section with m_k signal conductors and a reference can be represented by following $m_k \times m_k$ matrices of line

parameters: the resistance per unit R , the conductance per unit G , the inductance per unit L and the capacitance per unit C .

Let x represent the length along the transmission line, with $x=0$ corresponding to the near end and $x=D$ corresponding to the far end, as show in Fig. 2.1 and Fig. 2.2.

In time domain, the transmission line can be described by a system of partial differential equations given by

$$\frac{\partial v(x, t)}{\partial x} = -(Ri(x, t) + L \frac{\partial i(x, t)}{\partial t}) \quad (2.2-1)$$

$$\frac{\partial i(x, t)}{\partial x} = -(Gv(x, t) + C \frac{\partial v(x, t)}{\partial t}) \quad (2.2-2)$$

where $v(x, t) \in R^{m_k}$ and $i(x, t) \in R^{m_k}$ are the time domain voltage and current vectors of the transmission line at location x , respectively; m_k is the dimension of the k th transmission line.

In frequency domain, the transmission line can be represented by a system of ordinary differential equations with consideration of its initial conditions given by

$$\frac{dV(x, s)}{dx} = -(R + sL)I(x, s) + Li(x, t) = -ZI(x, s) + Li(x, t) \quad (2.2-3)$$

$$\frac{dI(x, s)}{dx} = -(G + sC)V(x, s) + Cv(x, t) = -YV(x, s) + Cv(x, t) \quad (2.2-4)$$

where $V(x, s)$ and $I(x, s)$ are the frequency domain counterparts of $v(x, t)$ and $i(x, t)$,

$$Z = R + sL \quad (2.2-5)$$

$$Y = G + sC \quad (2.2-6)$$

Note that the voltages and currents on the transmission line are the function of location x as shown in Fig. 2.2.

2.3 Fast Fourier Transform (FFT)

The fast Fourier transform (FFT) can be used to compute the time domain responses of the networks containing lossy multiconductor transmission lines [3]. Using the Fourier transform, the time domain waveforms are initially transformed. Then the analysis of the system is performed in the frequency domain at a set of discrete frequencies. Finally, the inverse FFT is used to obtain the time domain waveforms.

This approach has a major difficulty when the analysis has to span a time interval of several line transient times. For example, the response of a lossless line with short-circuited ports is of an infinite duration. Consequently, it is impossible in this case to compute the response using FFT. Even for moderately lossy lines, the duration of the response exceeds many transit times of the transmission line networks. This makes the use of the inverse FFT technique inefficient, as a large number of points must be added to the analysis to avoid aliasing problems [9].

2.4 Generalized Asymptotic Waveform Evaluation(GAWE)

In 1990 Pillage and Rohrer introduced the Asymptotic Waveform Evaluation (AWE) technique to time response analysis[16] for lumped component networks. This is an s-domain simulator which models the n th order circuit by a lower order q -model approximated with $2q-1$ moments. The q dominant poles and residues are found efficiently by recursive D.C. analysis of the circuit. However, their method is only limited to lumped component circuits. Tang and Nahkla extended AWE into both lumped and distributed circuits [23], labelling it the Generalized Asymptotic Waveform Evaluation (GAWE) technique. They created s-domain stamps of transmission lines and added them into the Modified Nodal Admittance (MNA) equations of lumped components, which resulted in more efficient computation of distributed networks. The following is a brief summary of the GAWE method.

Consider a linear network which contains linear lumped components and linear distributed component subnetworks. The generalized MNA equations of the network with an impulse input excitation can be written as [23]:

$$W \frac{dZ(t)}{dt} + HZ(t) + \sum_{k=1}^{N_s} D_k i_k(t) - b\delta(t) = 0 \quad (2.4-1)$$

where

$Z(t) \in R^N$ is the vector of node voltage waveforms appended by independent voltage source current waveforms and linear inductor current waveforms,

$W \in R^{N \times N}$ and $H \in R^{N \times N}$ are constant matrices with entries determined by the lumped linear components,

$b \in R^N$ is a constant vector related with the independent voltage and current sources,

$D_k = [d_{m,j}]$, $d_{mj} \in \{0, 1\}$, $m \in \{1, 2, \dots, N\}$, and $j \in \{1, 2, \dots, n_k\}$ with a maximum of one nonzero in each row or column is a selector matrix that maps $i_k(t) \in R^{n_k}$, the vector of currents entering the linear subnetworks k , into the node space R^N of the network,

N_s is the number of linear subnetworks,

$\delta(t)$ is the unit impulse function.

The relation between terminal voltages and terminal currents of the k th distributed subnetwork is given by

$$A_k V_k(s) + B_k I_k(s) = 0 \quad (2.4-2)$$

where V_k and I_k represents the s-domain terminal voltages and currents of the subnetwork k , respectively. For linear distributed subnetwork k , A_k and B_k can be described in terms of the line parameters. The details can be found in [23][24].

Taking the Laplace transform of (2.4-1) will result in.

$$\begin{bmatrix} sW + H & D_1 & \dots & D_{N_s} \\ A_1 D_1^t & B_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ A_{N_s} D_{N_s}^t & 0 & \dots & B_{N_s} \end{bmatrix} \begin{bmatrix} Z(s) \\ I_1(s) \\ \vdots \\ I_{N_s}(s) \end{bmatrix} = \begin{bmatrix} b \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (2.4-3)$$

or

$$Y(s) X(s) = E \quad (2.4-4)$$

where

$$Y(s) = \begin{bmatrix} sW + H & D_1 & \dots & D_{N_s} \\ A_1 D_1^t & B_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ A_{N_s} D_{N_s}^t & 0 & \dots & B_{N_s} \end{bmatrix} \quad (2.4-5)$$

$$X(s) = [Z(s) \quad I_1 \quad \dots \quad I_{N_s}]^t \quad (2.4-6)$$

$$E = [b \quad 0 \quad \dots \quad 0]^t \quad (2.4-7)$$

Superscript t denotes transpose. The non-singularity of Y is a necessary requirement.

From (2.4-4) an explicit expression of X(s) can be written as follows.

$$X(s) = Y^{-1} E \quad (2.4-8)$$

X(s) can be approximated by its Maclaurin's series in the form

$$X(s) = \sum_{n=0}^{\infty} M_n s^n \quad (2.4-9)$$

where

$$M_n = \frac{\frac{\partial^n}{\partial s^n} Y^{-1}}{n!} E \bigg|_{s=0} \quad (2.4-10)$$

The moments $[M_n]_i$ ($n=0, 1, \dots, 2q-1$) of an output i are then matched to a lower order s -domain Pade approximation function given by

$$[\tilde{X}(s)]_i = \sum_{j=1}^q \frac{k_j}{s-p_j} = \frac{R_1(s)}{R_2(s)} \quad (2.4-11)$$

which gives the time domain solution

$$[\tilde{x}(t)]_i = \sum_{j=1}^q k_j e^{p_j t} \quad (2.4-12)$$

It is noted that in (2.4-11) R_2 is a q -order polynomial function and R_1 is specifically a $(q-1)$ -order polynomial function, respectively. p_j are poles and k_j are residues of $[\tilde{X}(s)]_i$, respectively.

Solutions of p_j are given by [16]

$$a_0 + a_1 p^{-1} + a_2 p^{-2} + \dots + p^{-q} = 0 \quad (2.4-13)$$

where coefficients a_i are found by solving [16]

$$\begin{bmatrix} m_0 & m_1 & \dots & m_{q-1} \\ m_1 & m_2 & \dots & m_q \\ \dots & \dots & \dots & \dots \\ m_{q-1} & m_q & \dots & m_{2q-2} \end{bmatrix} \begin{bmatrix} -a_0 \\ -a_1 \\ \dots \\ -a_{q-1} \end{bmatrix} = \begin{bmatrix} m_q \\ m_{q+1} \\ \dots \\ m_{2q-1} \end{bmatrix} \quad (2.4-14)$$

The corresponding residues are then obtained by solving [15][24]

$$\begin{bmatrix} p_1^{-1} & p_2^{-1} & \dots & p_q^{-1} \\ p_1^{-2} & p_2^{-2} & \dots & p_q^{-2} \\ \dots & \dots & \dots & \dots \\ p_1^{-q} & p_2^{-q} & \dots & p_q^{-q} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \dots \\ k_q \end{bmatrix} = \begin{bmatrix} m_0 \\ m_1 \\ \dots \\ m_{q-1} \end{bmatrix} \quad (2.4-15)$$

Comments on GAWE.

Since GAWE models a n-order circuit by a lower q-order model, it saves CPU time significantly. Therefore, GAWE is one of the most efficient computing methods for distributed networks. Compared with the numerical inversion of the Laplace transform method, which will be discussed after this section, GAWE is less accurate. Improvement of accuracy can be achieved by increasing the order of approximation. In [24] it is shown that, in general, a relatively low order GAWE approximation is adequate.

2.5 Numerical Inversion of Laplace Transform

Various numerical methods for numerical inversion of Laplace transform are available in literature. In 1975 Singhal and Vlach introduced the numerical inversion of Laplace transform (NILT) using Pade approximation into time response analysis for linear lumped networks[20][21]. The method does not requires the determination of the poles

and residues of the network functions. It is applicable to stiff systems and to systems with multiple poles and is equivalent to a high order, absolutely stable integration method. The results of NILT are more accurate and more efficient than previous FFT-based methods since it has no aliasing problem and requires only a one point computation for any time interval t . It is particularly useful in the analysis of mismatched coupled systems where reflections result in very long response time[8][9].

(a) Description of NILT Method

For a linear network with linear distributed subnetworks, the system MNA equations are given by

$$\left(A + sB + \sum_{k=1}^{N_T} P_k H_k P_k^t \right) F(s) = J(s) \quad (2.5-1)$$

where $A \in R^{n \times n}$ and $B \in R^{n \times n}$ are matrices contributed by lumped components,

n is the dimension of the system,

$H_k \in C^{2m_k \times 2m_k}$ is a matrix of the k th transmission line stamp, the details of

H_k will be given in section (b),

$P_k = [(p_k)_{ij}] \in R^{n \times 2m_k}$, $(p_k)_{ij} \in \{0, 1\}$, P_k is a selector matrix that

maps the vector of currents entering the k th transmission line into the node

space R^n of the linear distributed network,

m_k is the dimension of the k th transmission line,

N_T is the number of coupled transmission lines,

s is the Laplace operator,

$F(s) \in C^n$ is a vector related with the waveforms of nodal voltages, independent voltage source currents and impedance currents,

$J(s) \in C^n$ is a vector of current sources appended by independent voltage source waveforms.

The solution of (2.5-1) can be obtained by using LU decomposition. After solving (2.5-1) in the s-domain, numerical inversion of Laplace transform (NILT) is used to obtain time domain response, which is discussed below.

The NILT method is based on the Laplace transform inversion formula

$$f(t) = \frac{1}{2\pi j} \int_{(c-j\infty)}^{(c+j\infty)} F(s) e^{st} ds \quad (2.5-2)$$

where s is the Laplace operator and $f(t)$ is the time domain response with respect to $F(s)$. Exact inversion is possible only if the poles of $F(s)$ are known. Since Vlach and Singhal wished to avoid the root-finding procedure, they used the substitution

$$z = st \quad (2.5-3)$$

then (2.5-2) can be changed into

$$f(t) = \frac{1}{2\pi jt} \int_{(c'-j\infty)}^{(c'+j\infty)} F\left(\frac{z}{t}\right) e^z dz \quad (2.5-4)$$

Instead of calculating its exact solution, (2.5-4) can be approximated by

$$\hat{f}(t) = \frac{1}{2\pi jt} \int_{(c'-j\infty)}^{(c'+j\infty)} F\left(\frac{z}{t}\right) \xi_{N,M}(z) dz \quad (2.5-5)$$

where

$$\xi_{N,M}(z) = \frac{P_N(z)}{Q_M(z)} = \frac{\sum_{i=0}^N (M+N-i)! \binom{N}{i} z^i}{\sum_{i=0}^M (-1)^i (M+N-i)! \binom{M}{i} z^i} = \sum_{i=1}^M \frac{k_i}{z-z_i} \quad (2.5-6)$$

is the Pade approximation of e^z . $P_N(z)$ and $Q_M(z)$ are polynomials of order N and M , respectively. k_i are residues and z_i are poles of $\xi_{N,M}(z)$, respectively. In order that the paths along the infinite arc do not contribute to the integral of (2.5-5), N must be less than M . The integral (2.5-5) can be evaluated by residue calculus in the form

$$\hat{f}(t) = -\frac{1}{t} \sum_{i=1}^M k_i F(z_i) = -\frac{1}{t} \sum_{i=1}^{M'} \operatorname{Re} \left[\tilde{k}_i F\left(\frac{z}{t}\right) \right] \quad (2.5-7)$$

where

$$M' = \frac{M}{2}, \tilde{k}_i = 2k_i \text{ when } M \text{ is even,}$$

$$M' = \frac{(M+1)}{2} \text{ when } M \text{ is odd and in this case } \tilde{k}_i = k_i \text{ for the residue corresponding to the real pole,}$$

$\operatorname{Re}[\cdot]$ denotes real part of complex number. A collection of tables for the poles z_i and the residues k_i can be found in [21] [28] for different values of N and M .

The inversion method is equivalent to the integration of differential equations, with the order of integration equal to $M+N$ [18][21]. To distinguish this method from that of the next chapter, we will call it a numerical inversion of Laplace transform of degree 0 or simply NILT0 (0 means 0 degree of derivatives). It was proved [21] that the truncation

error of NILTO, which is caused by Pade approximation, is

$$\Psi_0(N, M) = \frac{d_{N+M+1} (1 - D_{N, M})}{(N + M + 1)!} t^{N+M+1} \quad (2.5-8)$$

where the definition of d_{N+M+1} and $D_{N, M}$ will be given in chapter 3. Note that since $\Psi_0(N, M)$ is the function of t^{N+M+1} , error goes up with time.

It is also noted that since k_i and z_i were already calculated, the main cost of NILTO is one LU decomposition plus a backward-forward substitution for each pair of conjugate poles of the Pade approximation function of (2.5-6).

(b) Griffith and Nakhla's Stamps of Transmission Lines for NILTO

It is known that the relationship between voltages and currents of the k th transmission line in time domain can be written as partial differential equations as follows:

$$\frac{\partial v_k(x, t)}{\partial x} = -(R_k i_k(x, t) + L_k \frac{\partial i_k(x, t)}{\partial t}) \quad (2.5-9)$$

$$\frac{\partial i_k(x, t)}{\partial x} = -(G_k v_k(x, t) + C_k \frac{\partial v_k(x, t)}{\partial t}) \quad (2.5-10)$$

where R_k , G_k , L_k and C_k are line parameter matrices of the k th transmission line, R_k is the resistance matrix, G_k is the conductance matrix, L_k is the inductance matrix and C_k is the capacitance matrix. $v_k(x, t)$ and $i_k(x, t)$ are voltage and current vector of the k th transmission line in time domain at location x , respectively.

Applying the Laplace transform to (2.5-9) and (2.5-10) with assuming zero initial conditions will result in

$$\frac{dV_k(x, s)}{dx} = -(R_k + sL_k)I_k(x, s) = -Z_k I_k(x, s) \quad (2.5-11)$$

$$\frac{dI_k(x, s)}{dx} = -(G_k + sC_k)V_k(x, s) = -Y_k V_k(x, s) \quad (2.5-12)$$

where $V_k(x, s)$ and $I_k(x, s)$ are the s-domain counterparts of $v_k(x, t)$ and $i_k(x, t)$, respectively.

$$Z_k = R_k + sL_k \quad (2.5-13)$$

$$Y_k = G_k + sC_k \quad (2.5-14)$$

Differentiating both (2.5-11) and (2.5-12) in terms of x gives

$$\frac{d^2}{dx^2} V_k(x, s) = Z_k Y_k V_k(x, s) \quad (2.5-15)$$

$$\frac{d^2}{dx^2} I_k(x, s) = Y_k Z_k I_k(x, s) \quad (2.5-16)$$

Assume that

$$\underline{V}_k(x, s) = S_{kvi} e^{\gamma_{ki} x} \quad (2.5-17)$$

is a solution of (2.5-15), where γ_{ki} and S_{kvi} are the eigenvalue and eigenvector pair of $Z_k Y_k$. Applying (2.5-17) in (2.5-15) gives

$$\gamma_{ki}^2 S_{kvi} e^{\gamma_{ki} x} = Z_k Y_k S_{kvi} e^{\gamma_{ki} x} \quad (2.5-18)$$

which is equivalent to

$$(\gamma_{ki}^2 U - Z_k Y_k) S_{kvi} = 0 \quad (2.5-19)$$

where $i=1, 2, \dots, m_k$, and m_k is the dimension of the k th coupled transmission line and U is the identity matrix.

Since both γ_{ki} and $-\gamma_{ki}$ are eigenvalues of $Z_k Y_k$, the general solution of (2.5-15) is given by

$$V_k(x, s) = S_{kv} E_k(x) K_{k1} + S_{kv} E_k^{-1}(x) K_{k2} \quad (2.5-20)$$

where

$$S_{kv} = [S_{kv1} \quad S_{kv2} \quad \dots \quad S_{kvm_k}] \quad (2.5-21)$$

$$E_k(x) = \begin{bmatrix} e^{-\gamma_{k1}x} & 0 & \dots & 0 \\ 0 & e^{-\gamma_{k2}x} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & e^{-\gamma_{km_k}x} \end{bmatrix} \quad (2.5-22)$$

K_{k1} and K_{k2} are constant vectors defined by terminal values of $V_k(x, s)$.

Differentiating (2.5-20) gives

$$\frac{dV_k(x, s)}{dx} = -S_{kv} \Gamma_k E_k(x) K_{k1} + S_{kv} \Gamma_k E_k^{-1}(x) K_{k2} \quad (2.5-23)$$

where

$$\Gamma_k = \begin{bmatrix} \gamma_{k1} & 0 & \dots \\ 0 & \gamma_{k2} & \dots \\ \dots & \dots & \dots \\ 0 & 0 & \dots & \gamma_{km_k} \end{bmatrix} \quad (2.5-24)$$

Combining (2.5-23) with (2.5-11) together gives

$$-S_{kv}\Gamma_k E_k(x)K_{k1} + S_{kv}\Gamma_k E_k^{-1}(x)K_{k2} = -Z_k I_k(x, s) \quad (2.5-25)$$

The solution of $I_k(x, s)$ can be obtained from (2.5-25) as follows

$$\begin{aligned} I_k(x, s) &= Z_k^{-1} S_{kv} \Gamma_k E_k(x) K_{k1} - Z_k^{-1} S_{kv} \Gamma_k E_k^{-1}(x) K_{k2} \\ &= S_{ki} E_k(x) K_{k1} - S_{ki} E_k^{-1}(x) K_{k2} \end{aligned} \quad (2.5-26)$$

where

$$S_{ki} = Z_k^{-1} S_{kv} \Gamma_k \quad (2.5-27)$$

Let $x=0$ and D_k (D_k is the length of the k th coupled transmission line) in (2.5-20), then

$$\begin{aligned} V_k(0, s) &= S_{kv} K_{k1} + S_{kv} K_{k2} \\ V_k(D_k, s) &= S_{kv} E_k(D_k) K_{k1} + S_{kv} E_k^{-1}(D_k) K_{k2} \end{aligned} \quad (2.5-28)$$

which gives the constant vectors K_{k1} and K_{k2} as below:

$$\begin{bmatrix} K_{k1} \\ K_{k2} \end{bmatrix} = \begin{bmatrix} S_{kv} & S_{kv} \\ S_{kv} E_k(D_k) & S_{kv} E_k^{-1}(D_k) \end{bmatrix}^{-1} \begin{bmatrix} V_k(0, s) \\ V_k(D_k, s) \end{bmatrix} \quad (2.5-29)$$

Similarly,

$$\begin{aligned}
I_k(0, s) &= S_{ki}K_{k1} - S_{ki}K_{k2} \\
I_k(D_k, s) &= S_{ki}E_k(D_k)K_{k1} - S_{ki}E_k^{-1}(D_k)K_{k2}
\end{aligned} \tag{2.5-30}$$

Applying (2.5-29) into (2.5-30) gives a matrix form as follows

$$\begin{aligned}
\begin{bmatrix} I_k(0, s) \\ I_k(D_k, s) \end{bmatrix} &= \begin{bmatrix} S_{ki} & -S_{ki} \\ S_{ki}E_k(D_k) & -S_{ki}E_k^{-1}(D_k) \end{bmatrix} \begin{bmatrix} K_{k1} \\ K_{k2} \end{bmatrix} \\
&= \begin{bmatrix} S_{ki} & -S_{ki} \\ S_{ki}E_k(D_k) & -S_{ki}E_k^{-1}(D_k) \end{bmatrix} \begin{bmatrix} S_{kv} & S_{kv} \\ S_{kv}E_k(D_k) & S_{kv}E_k^{-1}(D_k) \end{bmatrix}^{-1} \begin{bmatrix} V_k(0, s) \\ V_k(D_k, s) \end{bmatrix}
\end{aligned} \tag{2.5-31}$$

In order to match the MNA equation form, the reference direction of $I_k(D_k)$ is changed in (2.5-31), then (2.5-31) can be written as:

$$\begin{aligned}
\begin{bmatrix} I_k(0, s) \\ I_k(D_k, s) \end{bmatrix} &= \begin{bmatrix} S_{ki} & -S_{ki} \\ -S_{ki}E_k(D_k) & S_{ki}E_k^{-1}(D_k) \end{bmatrix} \begin{bmatrix} S_{kv} & S_{kv} \\ S_{kv}E_k(D_k) & S_{kv}E_k^{-1}(D_k) \end{bmatrix}^{-1} \begin{bmatrix} V_k(0, s) \\ V_k(D_k, s) \end{bmatrix} \\
&= \begin{bmatrix} S_{ki} & 0 \\ 0 & S_{ki} \end{bmatrix} \begin{bmatrix} U & -U \\ -E_k(D_k) & E_k^{-1}(D_k) \end{bmatrix} \begin{bmatrix} U & U \\ E_k(D_k) & E_k^{-1}(D_k) \end{bmatrix}^{-1} \begin{bmatrix} S_{kv} & 0 \\ 0 & S_{kv} \end{bmatrix}^{-1} \begin{bmatrix} V_k(0, s) \\ V_k(D_k, s) \end{bmatrix} \\
&= \begin{bmatrix} S_{ki} & 0 \\ 0 & S_{ki} \end{bmatrix} \begin{bmatrix} E_{k1}(D_k) & E_{k2}(D_k) \\ E_{k2}(D_k) & E_{k1}(D_k) \end{bmatrix} \begin{bmatrix} S_{kv}^{-1} & 0 \\ 0 & S_{kv}^{-1} \end{bmatrix} \begin{bmatrix} V_k(0, s) \\ V_k(D_k, s) \end{bmatrix} \\
&= \begin{bmatrix} S_{ki}E_{k1}(D_k)S_{kv}^{-1} & S_{ki}E_{k2}(D_k)S_{kv}^{-1} \\ S_{ki}E_{k2}(D_k)S_{kv}^{-1} & S_{ki}E_{k1}(D_k)S_{kv}^{-1} \end{bmatrix} \begin{bmatrix} V_k(0, s) \\ V_k(D_k, s) \end{bmatrix} = H_k(D_k) \begin{bmatrix} V_k(0, s) \\ V_k(D_k, s) \end{bmatrix}
\end{aligned} \tag{2.5-32}$$

where

$$E_{k1}(D_k) = \text{diag} \left(\frac{\exp(\gamma_{ki}D_k) + \exp(-\gamma_{ki}D_k)}{\exp(\gamma_{ki}D_k) - \exp(-\gamma_{ki}D_k)} \right) = \text{diag} \left(\frac{1 + \exp(-2\gamma_{ki}D_k)}{1 - \exp(-2\gamma_{ki}D_k)} \right) \tag{2.5-33}$$

$$E_{k2}(D_k) = \text{diag} \left(\frac{2}{\exp(-\gamma_{ki} D_k) - \exp(\gamma_{ki} D_k)} \right) = \text{diag} \left(\frac{2 \exp(-\gamma_{ki} D_k)}{\exp(-2\gamma_{ki} D_k) - 1} \right) \quad (2.5-34)$$

and

$$H_k(D_k) = \begin{bmatrix} S_{ki} E_{k1}(D_k) S_{kv}^{-1} & S_{ki} E_{k2}(D_k) S_{kv}^{-1} \\ S_{ki} E_{k2}(D_k) S_{kv}^{-1} & S_{ki} E_{k1}(D_k) S_{kv}^{-1} \end{bmatrix} \quad (2.5-35)$$

Here $H_k(D_k)$ is the k th transmission stamp required in equation (2.5-1).

Comments on NILT0

Compared to FFT, NILT0 is more accurate and more efficient. Particularly for the continuous input waveform, NILT0 has no aliasing problem. NILT0 is also faster than FFT because it can compute the response at any specific time t_c , whereas FFT has to calculate all of the responses from $t=t_0$ to $t = t_c$ for a set of points. A disadvantage of NILT0, on the other hand, is that the computing results will be inaccurate when delay time t increases.

2.4 Comparison

Comparisons are made of the three methods mentioned in this chapter for time analysis of distributed networks based on their accuracy and efficiencies, as shown in Table 2.1. Among them, GAWE is the most efficient and NILT0 is the most accurate.

Method	Accuracy	Efficiency
FFT	L	L
GAWE	M	H
NILTO	H	M

Table 2.1 Comparison of the three methods

Chapter 3

High Degree Numerical Inversion of Laplace Transform

3.1 Introduction

As described in the previous chapter, the numerical inversion of Laplace transform (NILT0) method is efficient in solving time responses of linear networks, especially for lossy multiconductor transmission line networks since conventional circuit simulators can not be used in these circumstances. The results have shown that the NILT0 method is accurate for certain time interval and is absolutely stable [21].

This chapter presents a high degree numerical inversion of Laplace transform method by using derivatives of output waveforms with respect to the Laplace operator s . The new methods are more accurate and more efficient than NILT0 for a given M and N . In addition, they are absolutely stable. The theoretical basis for the new methods is given in section 3.2 and their truncation error is proved in section 3.3. The stability of the new methods is discussed in section 3.4. Comparisons of results obtained by Hspice, NILT0

and the new methods for linear lumped networks are shown in section 3.5. Their efficiency and applications for distributed networks will be discussed in the next chapter.

3.2 Derivation of The New Method

To distinguish the new method from the method mentioned in section 2.5, the numerical inversion of Laplace transform of degree n will be denoted by NILT $_n$ ($n > 0$). Corresponding to (2.5-5), the equation for NILT $_n$ with Pade approximation is given by

$$\begin{aligned}\hat{f}_n(t) &= \frac{n+1}{2\pi jt} \int_{(c-j\infty)}^{(c+j\infty)} F\left(\frac{(n+1)z}{t}\right) \xi_{N,M}^{n+1}(z) dz \\ &= \frac{p}{2\pi jt} \int_{(c-j\infty)}^{(c+j\infty)} F\left(\frac{pz}{t}\right) \xi_{N,M}^p(z) dz\end{aligned}\quad (3.2-1)$$

where n is the highest degree of derivative used and $p = n + 1$. Note that (3.2-1) will be exactly the same as (2.5-5) for $n=0$. The following is its derivation.

Proof of (3.2-1)

Let n ($n = 1, 2, \dots$) be the highest degree of derivative used and $p = n + 1$. Replacing t by $p\tau$ in (2.5-2) results in

$$f_n(p\tau) = \frac{1}{2\pi j} \int_{(c-j\infty)}^{(c+j\infty)} F(s) e^{sp\tau} ds \quad (3.2-2)$$

Subscript n denotes the n degree derivative to be used for obtaining the time response.

Using the same substitution as used in 2.5(a), let

$$z = s\tau \quad (3.2-3)$$

Applying (3.2-3) into (3.2-2) gives

$$\hat{f}_n(p\tau) = \frac{1}{2\pi j\tau} \int_{(c'-j\infty)}^{(c'+j\infty)} F\left(\frac{z}{\tau}\right) e^{pz} dz \quad (3.2-4)$$

Substitute e^{pz} by pth-order power of Pade approximation $\xi_{N,M}^p(z)$, which can be obtained using (2.5-6)

$$\xi_{N,M}^p(z) = \left(\sum_{i=1}^M \frac{k_i}{z - z_i} \right)^p \quad (3.2-5)$$

from (3.2-4) and (3.2-5)

$$\hat{f}_n(p\tau) = \frac{1}{2\pi j\tau} \int_{(c'-j\infty)}^{(c'+j\infty)} F\left(\frac{z}{\tau}\right) \xi_{N,M}^p(z) dz \quad (3.2-6)$$

Substituting $\tau = \frac{t}{p}$ into (3.2-6) gives a general form of the numerical inversion of Laplace transform of degree n (NILTn).

□

Solving (3.2-1) is equivalent to solving for the residues of $F\left(\frac{pz}{t}\right) \xi_{N,M}^p(z)$.

The solutions of (3.2-1) with n=1 and n=2 are given below.

Numerical Inversion of Laplace Transform of Degree 1 (NILT1)

Using the first derivative of F(s) (n=1) with respect to s, the numerical inversion of Laplace transform of degree 1 (NILT1) is given by

$$\hat{f}_1(t) = -\frac{2}{t} \sum_{i=1}^M \operatorname{Re} \left[\frac{\tilde{k}_i}{t} F' \left(\frac{2z_i}{t} \right) + 2\tilde{k}_i A_i F \left(\frac{2z_i}{t} \right) \right] \quad (3.2-7)$$

where

$$A_i = \sum_{\substack{j=1 \\ j \neq i}}^M \frac{k_j}{z_i - z_j} \quad (3.2-8)$$

The symbols used in (3.2-7) are exactly the same as those in section 2.5.

The following is the derivation of (3.2-7).

Proof of (3.2-7)

Letting $p=n+1=2$ in (3.2-1) gives

$$\hat{f}_1(t) = \frac{1}{\pi j t} \int_{(c-j\infty)}^{(c+j\infty)} F \left(\frac{2z}{t} \right) \xi_{N,M}^2(z) dz \quad (3.2-9)$$

where

$$\begin{aligned} \xi_{N,M}^2(z) &= \left(\sum_{i=1}^M \frac{k_i}{z - z_i} \right)^2 = \sum_{i=1}^M \sum_{j=1}^M \frac{k_i k_j}{(z - z_i)(z - z_j)} \\ &= \sum_{i=1}^M \frac{k_i^2}{(z - z_i)^2} + \sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i}}^M \frac{k_i k_j}{(z - z_i)(z - z_j)} \end{aligned} \quad (3.2-10)$$

Substituting (3.2-10) into (3.2-9) results in

$$\begin{aligned}
\hat{f}_1(t) &= \frac{1}{\pi j t} \int_{(c'-j\infty)}^{(c'+j\infty)} \left(\sum_{i=1}^M \frac{k_i^2}{(z-z_i)^2} F\left(\frac{2z}{t}\right) + \sum_{\substack{i=1 \\ j \neq i}}^M \sum_{j=1}^M \frac{k_i k_j}{(z-z_i)(z-z_j)} F\left(\frac{2z}{t}\right) \right) dz \\
&= -\frac{2}{t} \left(\sum \text{residues of} \left(\sum_{i=1}^M \frac{k_i^2}{(z-z_i)^2} F\left(\frac{2z}{t}\right) \right) \right) \\
&\quad -\frac{2}{t} \left(\sum \text{residues of} \left(\sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i}}^M \frac{k_i k_j}{(z-z_i)(z-z_j)} F\left(\frac{2z}{t}\right) \right) \right)
\end{aligned} \tag{3.2-11}$$

The negative sign in (3.2-11) implies that the integral takes a clockwise route. The solutions of residues in (3.2-11) are given below.

$$\sum \text{residue of} \left(\sum_{i=1}^M \frac{k_i^2}{(z-z_i)^2} F\left(\frac{2z}{t}\right) \right) = \frac{2}{t} \sum_{i=1}^M k_i^2 F'\left(\frac{2z_i}{t}\right) \tag{3.2-12}$$

$$\sum \text{residues of} \left(\sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i}}^M \frac{k_i k_j}{(z-z_i)(z-z_j)} F\left(\frac{2z}{t}\right) \right) = 2 \sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i}}^M \frac{k_i k_j}{(z_i - z_j)} F\left(\frac{2z_i}{t}\right) \tag{3.2-13}$$

Therefore an explicit expression of (3.2-11) can be written as

$$\hat{f}_1(t) = -\frac{2}{t} \left(\frac{2}{t} \sum_{i=1}^M k_i^2 F'\left(\frac{2z_i}{t}\right) + 2 \sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i}}^M \frac{k_i k_j}{(z_i - z_j)} F\left(\frac{2z_i}{t}\right) \right) \tag{3.2-14}$$

Assuming M is even, the poles of the Pade approximation function always appear in conjugate pairs, which implies that the imaginary part of (3.2-14) will be zero. Let $M' = \frac{M}{2}$ and $\tilde{k}_i = 2k_i$ in (3.2-14). Thus the final solution of (3.2-1) for $n=1$ can be written as

$$\begin{aligned}
 \hat{f}_1(t) &= -\frac{2}{t} \left(\frac{1}{t} \sum_{i=1}^{M'} \operatorname{Re} \left[\tilde{k}_i F' \left(\frac{2z_i}{t} \right) \right] + 2 \sum_{i=1}^{M'} \sum_{\substack{j=1 \\ j \neq i}}^M \operatorname{Re} \left[\frac{\tilde{k}_i k_j}{(z_i - z_j)} F \left(\frac{2z_i}{t} \right) \right] \right) \\
 &= -\frac{2}{t} \sum_{i=1}^{M'} \operatorname{Re} \left[\frac{\tilde{k}_i}{t} F' \left(\frac{2z_i}{t} \right) + 2 \tilde{k}_i \sum_{\substack{j=1 \\ j \neq i}}^M \frac{k_j}{(z_i - z_j)} F \left(\frac{2z_i}{t} \right) \right]
 \end{aligned} \tag{3.2-15}$$

When M is odd, $M' = \frac{(M+1)}{2}$ and $\tilde{k}_i = k_i$ for the residue corresponding to the real pole.

□

Numerical Inversion of Laplace Transform of Degree 2 (NILT2)

Using up to the second degree of the derivative of $F(s)$ ($n=2$), the numerical inversion of Laplace transform of degree 2 (NILT2) is given by

$$\hat{f}_2(t) = -\frac{9}{t} \sum_{i=1}^{M'} \operatorname{Re} \left[\frac{3\tilde{k}_i^3}{8t^2} F'' \left(\frac{3z_i}{t} \right) + \frac{3\tilde{k}_i^2}{2t} A_i F' \left(\frac{3z_i}{t} \right) + D_i F \left(\frac{3z_i}{t} \right) \right] \tag{3.2-16}$$

where

$$D_i = C_i \tilde{k}_i - \frac{\tilde{k}_i^2 B_i}{2} \tag{3.2-17}$$

$$B_i = \sum_{\substack{j=1 \\ j \neq i}}^M \frac{k_j}{(z_i - z_j)^2} \tag{3.2-18}$$

$$C_i = \sum_{j=1}^M \sum_{\substack{v=1 \\ j \neq i, v \neq i}}^M \frac{k_i k_v}{(z_i - z_j)(z_i - z_v)} \quad (3.2-19)$$

The proof of (3.2-16) is the same as that of (3.2-7) except for choosing $n=2$ ($p=3$), which is given below.

Proof of (3.2-16)

Replacing $n=2$ into (3.2-1) gives

$$\hat{f}_2(t) = \frac{3}{2\pi j t} \int_{(c-j\infty)}^{(c+j\infty)} F\left(\frac{3z}{t}\right) \xi_{N,M}^3(z) dz \quad (3.2-20)$$

where

$$\begin{aligned} \xi_{N,M}^3(z) &= \left(\sum_{i=1}^M \frac{k_i}{z - z_i} \right)^3 \\ &= \sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i, v \neq i}}^M \sum_{v=1}^M \frac{k_i k_j k_v}{(z - z_i)(z - z_j)(z - z_v)} \\ &= \sum_{i=1}^M \frac{k_i^3}{(z - z_i)^3} + 3 \sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i}}^M \frac{k_i^2 k_j}{(z - z_i)^2 (z - z_j)} \\ &\quad + \sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i, v \neq i, j \neq v}}^M \sum_{v=1}^M \frac{k_i k_j k_v}{(z - z_i)(z - z_j)(z - z_v)} \end{aligned} \quad (3.2-21)$$

Substituting (3.2-21) into (3.2-20) results in

$$\begin{aligned}
\hat{f}_2(t) &= \frac{3}{2\pi jt} \int_{(c'-j\infty)}^{(c'+j\infty)} \sum_{i=1}^M \frac{k_i^3}{(z-z_i)^3} F\left(\frac{3z}{t}\right) dz \\
&+ \frac{3}{2\pi jt} \int_{(c'-j\infty)}^{(c'+j\infty)} 3 \sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i}}^M \frac{k_i^2 k_j}{(z-z_i)^2 (z-z_j)} F\left(\frac{3z}{t}\right) dz \\
&+ \frac{3}{2\pi jt} \int_{(c'-j\infty)}^{(c'+j\infty)} \sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i, v \neq i, j \neq v}}^M \sum_{v=1}^M \frac{k_i k_j k_v}{(z-z_i)(z-z_j)(z-z_v)} F\left(\frac{3z}{t}\right) dz \\
&= -\frac{3}{t} \sum \text{residues of} \left(\sum_{i=1}^M \frac{k_i^3}{(z-z_i)^3} F\left(\frac{3z}{t}\right) \right) \\
&- \frac{3}{t} \sum \text{residues of} \left(3 \sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i}}^M \frac{k_i^2 k_j}{(z-z_i)^2 (z-z_j)} F\left(\frac{3z}{t}\right) \right) \\
&- \frac{3}{t} \sum \text{residues of} \left(\sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i, v \neq i, j \neq v}}^M \sum_{v=1}^M \frac{k_i k_j k_v}{(z-z_i)(z-z_j)(z-z_v)} F\left(\frac{3z}{t}\right) \right)
\end{aligned} \tag{3.2-22}$$

The residues in (3.2-22) are given by the following.

$$\sum \text{residues of} \left(\sum_{i=1}^M \frac{k_i^3}{(z-z_i)^3} F\left(\frac{3z}{t}\right) \right) = \frac{1}{2} \sum_{i=1}^M \frac{9k_i^3}{t^2} F''\left(\frac{3z_i}{t}\right) \tag{3.2-23}$$

$$\begin{aligned}
& \sum \text{residues of} \left(3 \sum_{\substack{i=1 \\ j \neq i}}^M \sum_{j=1}^M \frac{k_i^2 k_j}{(z-z_i)^2 (z-z_j)} F\left(\frac{3z}{t}\right) \right) \\
&= 3 \sum_{\substack{i=1 \\ j \neq i}}^M \sum_{j=1}^M k_i^2 k_j \left(\frac{3F'\left(\frac{3z_i}{t}\right)}{t(z_i-z_j)} - \frac{F\left(\frac{3z_i}{t}\right)}{(z_i-z_j)^2} \right) \\
&\quad + 3 \sum_{\substack{i=1 \\ j \neq i}}^M \sum_{j=1}^M \frac{k_i k_j^2}{(z_i-z_j)^2} F\left(\frac{3z_i}{t}\right)
\end{aligned} \tag{3.2-24}$$

$$\begin{aligned}
& \sum \text{residues of} \left(\sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i, v \neq i, j \neq v}}^M \sum_{v=1}^M \frac{k_i k_j k_v}{(z-z_i)(z-z_j)(z-z_v)} F\left(\frac{3z}{t}\right) \right) \\
&= 3 \sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i, v \neq i, j \neq v}}^M \sum_{v=1}^M \frac{k_i k_j k_v}{(z_i-z_j)(z_i-z_v)} F\left(\frac{3z_i}{t}\right)
\end{aligned} \tag{3.2-25}$$

Substituting (3.2-23)-(3.2-25) into (3.2-22) gives an explicit expression of (3.2-20)

$$\begin{aligned}
\hat{f}_2(t) = & -\frac{27}{2t^3} \sum_{i=1}^M k_i^3 F''\left(\frac{3z_i}{t}\right) - \frac{9}{t} \sum_{\substack{i=1 \\ j \neq i}}^M \sum_{j=1}^M k_i^2 k_j \left(\frac{3F'\left(\frac{3z_i}{t}\right)}{t(z_i-z_j)} - \frac{F\left(\frac{3z_i}{t}\right)}{(z_i-z_j)^2} \right) \\
& - \frac{9}{t} \sum_{i=1}^M \sum_{\substack{j=1 \\ j \neq i, v \neq i}}^M \sum_{v=1}^M \frac{k_i k_j k_v}{(z_i-z_j)(z_i-z_v)} F\left(\frac{3z_i}{t}\right)
\end{aligned} \tag{3.2-26}$$

Substituting the relations $M' = \frac{M}{2}$ and $\bar{k}_i = 2k_i$ into (3.2-26), the final solution of (3.2-1) for $n=2$ (assuming M is even) can be written as

$$\begin{aligned}
\hat{f}_2(t) &= -\frac{27}{2t^3} \sum_{i=1}^{M'} Re \left[\frac{\tilde{k}_i^3}{4} F'' \left(\frac{3z_i}{t} \right) \right] - \frac{9}{t} \sum_{i=1}^{M'} \sum_{\substack{j=1 \\ j \neq i}}^M Re \left[\frac{\tilde{k}_i^2 k_j}{2} \left(\frac{3F' \left(\frac{3z_i}{t} \right)}{(z_i - z_j)t} - \frac{F \left(\frac{3z_i}{t} \right)}{(z_i - z_j)^2} \right) \right] \\
&\quad - \frac{9}{t} \sum_{i=1}^{M'} \sum_{\substack{j=1 \\ j \neq i, v \neq i}}^M \sum_{v=1}^M Re \left[\frac{\tilde{k}_i k_j k_v}{(z_i - z_j)(z_i - z_v)} F \left(\frac{3z_i}{t} \right) \right] \\
&= -\frac{9}{t} \sum_{i=1}^{M'} Re \left[\frac{3\tilde{k}_i^3 F'' \left(\frac{3z_i}{t} \right)}{8t^2} + \tilde{k}_i \sum_{j=1}^M \frac{k_j}{2} \left(\frac{3F' \left(\frac{3z_i}{t} \right)}{(z_i - z_j)t} - \frac{F \left(\frac{3z_i}{t} \right)}{(z_i - z_j)^2} \right) \right] \\
&\quad - \frac{9}{t} \sum_{i=1}^{M'} Re \left[\tilde{k}_i \sum_{\substack{j=1 \\ j \neq i, v \neq i}}^M \sum_{v=1}^M \frac{k_j k_v}{(z_i - z_j)(z_i - z_v)} F \left(\frac{3z_i}{t} \right) \right]
\end{aligned} \tag{3.2-27}$$

When M is odd, $M' = \frac{(M+1)}{2}$ and $\tilde{k}_i = k_i$ for the residue corresponding to the real pole.

□

Following the above derivations, we can have solutions of (3.2-1) with $n=3, n=4, \dots$

3.3 The Truncation Error of NILTn

Let $\Psi_n(N, M)$ be the truncation error of the numerical inversion of Laplace transform of degree n ($n > 0$). Compared to the truncation error of NILT0, the relative truncation error of NILTn with respect to NILT0 is given by

$$\frac{\Psi_n(N, M)}{\Psi_0(N, M)} = \frac{1}{(n+1)^{N+M}} \tag{3.3-1}$$

To prove (3.3-1), we need the following preparations.

Lemma 1:

$$(\xi_{N,M}^p(z))^{(k)} \Big|_{z=0} = p^k, 0 \leq k \leq N+M \quad (3.3-2)$$

Proof of Lemma 1:

It is noted [21] that the Pade approximation of e^z has the first $M+N+1$ terms of its Maclaurin's series expansion equal to the first $M+N+1$ terms of Maclaurin's series expansion of

$$e^z = \sum_{i=0}^{\infty} \frac{z^i}{i!} \quad (3.3-3)$$

which implies that

$$(\xi_{N,M}(z))^{(i)} \Big|_{z=0} = 1, 0 \leq i \leq N+M \quad (3.3-4)$$

The induction technique is used to prove (3.3-2) as follows:

(i) It is straightforward to show (3.3-2) is satisfied by $k=1$, i.e.

$$(\xi_{N,M}^p(z))^{(1)} \Big|_{z=0} = (p\xi_{N,M}^{p-1}(z) (\xi_{N,M}(z))^{(1)}) \Big|_{z=0} = p = p^1 \quad (3.3-5)$$

(ii) Assuming (3.3-2) is valid for $k=m-1$ ($0 < m \leq N+M$), i.e.

$$(\xi_{N,M}^p(z))^{(m-1)} \Big|_{z=0} = p^{m-1} \quad (3.3-6)$$

(iii) When $k=m$

$$\begin{aligned}
 (\xi_{N,M}^p(z))^{(m)} \Big|_{z=0} &= ((\xi_{N,M}^p(z))^{(1)})^{(m-1)} \Big|_{z=0} \\
 &= (p \xi_{N,M}^{p-1}(z) (\xi_{N,M}(z))^{(1)})^{(m-1)} \Big|_{z=0}
 \end{aligned} \tag{3.3-7}$$

Using Leibniz's formula

$$(hg)^{(k)} = \sum_{i=0}^k \binom{k}{i} h^{(i)} g^{(k-i)} \tag{3.3-8}$$

(3.3-7) can be written in the form

$$\begin{aligned}
 &(\xi_{N,M}^p(z))^{(m)} \Big|_{z=0} \\
 &= p \sum_{i=0}^{m-1} \binom{m-1}{i} (\xi_{N,M}^{p-1}(z))^{(i)} ((\xi_{N,M}(z))^{(1)})^{(m-1-i)} \Big|_{z=0} \\
 &= p \sum_{i=0}^{m-1} \binom{m-1}{i} (\xi_{N,M}^{p-1}(z))^{(i)} (\xi_{N,M}(z))^{(m-i)} \Big|_{z=0}
 \end{aligned} \tag{3.3-9}$$

Note that in (3.3-9), $0 \leq i \leq m-1 < N+M$. According to (3.3-6),

$$(\xi_{N,M}^{p-1}(z))^{(i)} \Big|_{z=0} = (p-1)^i \tag{3.3-10}$$

It is also noted that according to (3.3-4),

$$(\xi_{N,M}(z))^{(m-i)} \Big|_{z=0} = 1 \tag{3.3-11}$$

since $m-i \leq M+N$.

Using the binomial equation

$$(a+b)^k = \sum_{i=0}^k \binom{k}{i} a^i b^{k-i} \quad (3.3-12)$$

in (3.3-9) combined with (3.3-10) and (3.3-11) gives

$$\begin{aligned} (\xi_{N,M}^p(z))^{(m)} \Big|_{z=0} &= p \sum_{i=0}^{m-1} \binom{m-1}{i} (p-1)^i \bullet 1 \\ &= p(p-1+1)^{m-1} = p(p)^{m-1} = p^m \end{aligned} \quad (3.3-13)$$

□

Lemma 2:

$$(\xi_{N,M}^p(z))^{N+M+1} = p^{N+M+1} \left(1 - \frac{1 - D_{N,M}}{p^{N+M}} \right) \quad (3.3-14)$$

where

$$D_{N,M} = (\xi_{N,M}(z))^{(N+M+1)} \Big|_{z=0} \quad (3.3-15)$$

Proof of Lemma 2:

Write $(\xi_{N,M}^p(z))^{(N+M+1)} \Big|_{z=0}$ in the form

$$\begin{aligned}
(\xi_{N,M}^p(z))^{(N+M+1)} \Big|_{z=0} &= ((\xi_{N,M}^p(z))^{(1)})^{(N+M)} \Big|_{z=0} \\
&= (p \xi_{N,M}^{p-1}(z) (\xi_{N,M}(z))^{(1)})^{(N+M)} \Big|_{z=0} \\
&= p \sum_{i=0}^{N+M} \binom{N+M}{i} (\xi_{N,M}^{p-1}(z))^{(i)} (\xi_{N,M}(z))^{(N+M+1-i)} \Big|_{z=0}
\end{aligned} \tag{3.3-16}$$

Note that all terms of (3.3-16) have derivatives of $\xi_{N,M}^{p-1}(z)$ of degrees less than or equal to $N+M$ and have derivatives of $\xi_{N,M}(z)$ of degrees less than or equal to $N+M$, except for $i=0$. Using (3.3-4) and Lemma 1, $(\xi_{N,M}^p(z))^{(N+M+1)} \Big|_{z=0}$ can be written as

$$\begin{aligned}
&(\xi_{N,M}^p(z))^{(N+M+1)} \Big|_{z=0} \\
&= p \sum_{i=0}^{N+M} \binom{N+M}{i} (\xi_{N,M}^{p-1}(z))^{(i)} (\xi_{N,M}(z))^{(N+M+1-i)} \Big|_{z=0} \\
&= p \left(D_{N,M} + \sum_{i=1}^{N+M} \binom{N+M}{i} (p-1)^i \cdot 1 \right) \\
&= p (D_{N,M} + (p-1+1)^{N+M} - 1) \\
&= p^{N+M+1} \left(1 - \frac{1-D_{N,M}}{p^{N+M}} \right)
\end{aligned} \tag{3.3-17}$$

□

Using Lemma 1 and Lemma 2, the Maclaurin's series expansion of $\xi_{N,M}^p(z)$ represented by its first $N+M+2$ terms is given by

$$\xi_{N,M}^p(z) = \sum_{i=0}^{N+M} \frac{p^i}{i!} z^i + \frac{p^{N+M+1}}{(N+M+1)!} \left(1 - \frac{1-D_{N,M}}{p^{N+M}} \right) z^{N+M+1} + \dots \tag{3.3-18}$$

Proof of (3.3-1)

The definition of the truncation error of NILTn is

$$\Psi_n(N, M) = f(t) - \hat{f}_n(t) \quad (3.3-19)$$

Let d_i be the i th derivative of $f(t)$ at $t=0$, i.e.

$$d_i = f^{(i)}(t) \big|_{t=0} \quad (3.3-20)$$

Therefore the Maclaurin's series expansion of $f(t)$ represented by its first $N+M+2$ terms is

$$\begin{aligned} f(t) &= d_0 + d_1 t + \frac{d_2}{2!} t^2 + \dots + \frac{d_{N+M}}{(N+M)!} t^{N+M} + \frac{d_{N+M+1}}{(N+M+1)!} t^{N+M+1} + \dots \\ &= \sum_{i=0}^{N+M} \frac{d_i}{i!} t^i + \frac{d_{N+M+1}}{(N+M+1)!} t^{N+M+1} + \dots \end{aligned} \quad (3.3-21)$$

Taking the Laplace transform of (3.3-21), its s -domain expression is given by

$$F(s) = \sum_{i=0}^{N+M} \frac{d_i}{s^{i+1}} + \frac{d_{N+M+1}}{s^{N+M+2}} + \dots \quad (3.3-22)$$

Applying (3.3-22) and (3.3-18) into (3.2-1) with $s = \frac{(n+1)z}{t} = \frac{pz}{t}$ gives

$$\begin{aligned}
\hat{f}_n(t) &= \frac{p}{2\pi j t} \int_{(c-j\infty)}^{(c+j\infty)} F\left(\frac{pz}{t}\right) \xi_{N,M}^p(z) dz \\
&= \frac{p}{2\pi j t} \int_{(c-j\infty)}^{(c+j\infty)} \left(\sum_{i=0}^{N+M} \frac{d_i}{\left(\frac{pz}{t}\right)^{i+1}} + \frac{d_{N+M+1}}{\left(\frac{pz}{t}\right)^{N+M+2}} + \dots \right) \cdot \\
&\quad \cdot \left(\sum_{i=0}^{N+M} \frac{p^i}{i!} z^i + \frac{p^{N+M+1}}{(N+M+1)!} \left(1 - \frac{1-D_{N,M}}{p^{N+M}} \right) z^{N+M+1} + \dots \right) dz \quad (3.3-23) \\
&= \frac{p}{2\pi j t} \int_{(c-j\infty)}^{(c+j\infty)} \left(\sum_{i=0}^{N+M} \frac{d_i}{p z i!} t^{i+1} + \frac{d_{N+M+1} \left(1 - \frac{1-D_{N,M}}{p^{N+M}} \right)}{p z (N+M+1)!} t^{N+M+2} + \dots \right) dz \\
&= \frac{1}{2\pi j} \int_{(c-j\infty)}^{(c+j\infty)} \left(\sum_{i=0}^{N+M} \frac{d_i}{z i!} t^i + \frac{d_{N+M+1} \left(1 - \frac{1-D_{N,M}}{p^{N+M}} \right)}{z (N+M+1)!} t^{N+M+1} + \dots \right) dz
\end{aligned}$$

Solving (3.3-23) is equivalent to solving for the residues of

$$\left(\sum_{i=0}^{N+M} \frac{d_i}{z} t^i + \frac{d_{N+M+1} \left(1 - \frac{1-D_{N,M}}{p^{N+M}} \right)}{z (N+M+1)!} t^{N+M+1} + \dots \right). \text{ Note that all residues will be zero}$$

except for those z^{-1} terms. Thus (3.3-23) can be written as

$$\hat{f}_n(t) = \sum_{i=0}^{N+M} \frac{d_i t^i}{i!} + \frac{d_{N+M+1} \left(1 - \frac{1-D_{N,M}}{p^{N+M}} \right)}{(N+M+1)!} t^{N+M+1} + \dots \quad (3.3-24)$$

Applying (3.3-24) and (3.3-21) into (3.3-20), an explicit solution of $\Psi_n(N, M)$ can be written as

$$\begin{aligned}
\Psi_n(N, M) &= f(t) - \hat{f}_n(t) \\
&= \left(\sum_{i=0}^{N+M} \frac{d_i}{i!} t^i + \frac{d_{N+M+1}}{(N+M+1)!} t^{N+M+1} + \dots \right) \\
&\quad - \left(\sum_{i=0}^{N+M} \frac{d_i}{i!} t^i + \frac{d_{N+M+1} \left(1 - \frac{1 - D_{N,M}}{p^{N+M}} \right)}{(N+M+1)!} t^{N+M+1} + \dots \right) \\
&\approx \frac{d_{N+M+1} \frac{1 - D_{N,M}}{p^{N+M}}}{(N+M+1)!} t^{N+M+1} = \frac{\Psi_0(N, M)}{p^{N+M}}
\end{aligned} \tag{3.3-25}$$

□

To show the improvement of accuracy obtained by the new inversion formula of (3.2-1) compared with (2.5-4), the relative truncation error of (3.3-1) is calculated in Table 3.1 for some typical values of M and N with $n=1$ and $n=2$, respectively.

M/N	$n = 0$	$n = 1$	$n = 2$
4/2	1.0	1.6×10^{-3}	1.4×10^{-4}
6/4	1.0	9.8×10^{-4}	1.7×10^{-5}
8/6	1.0	6.1×10^{-5}	2.1×10^{-7}
10/8	1.0	3.8×10^{-6}	2.6×10^{-9}
Table 3.1: Comparison between truncation error coefficients of $n=0$, $n=1$ and $n=2$			

Table 3.1 shows that the new methods improve the accuracy significantly compared with

NILT0.

3.4 Stability of NILTn

In [21] NILT0 was proved to be absolutely stable. In this section it will be proved that NILTn has the same property for any n.

The stability property will be studied on an mth order linear system, written in the state variable form

$$y' = Ay \quad (3.4-1)$$

where $A \in C^{m \times m}$ is a constant matrix, for the sake of simplicity, we assume to have simple eigenvalues λ_i , $i=1, 2, \dots, m$. Taking the Laplace transform of (3.4-1)

$$(sU - A)Y = y_0 \quad (3.4-2)$$

where y_0 is the vector of initial conditions and U is the identity matrix. Since the poles are simple, there exists a non-singular matrix $T \in C^{m \times m}$ and

$$A = T^{-1}\Lambda T \quad (3.4-3)$$

where

$$\Lambda = \text{diag} \{ \lambda_1 \quad \lambda_2 \quad \dots \quad \lambda_m \} \quad (3.4-4)$$

Applying (3.4-3) into (3.4-2) gives

$$Y = T(sU - \Lambda)^{-1}T^{-1}y_0 \quad (3.4-5)$$

Let

$$F(s) = (sU - \Lambda)^{-1} \quad (3.4-6)$$

Hence,

$$F^{(1)}(s) = (-1)(sU - \Lambda)^{-2} \quad (3.4-7)$$

... ..

$$F^{(i)}(s) = (-1)^i i! (sU - \Lambda)^{-(i+1)} \quad (3.4-8)$$

where i is an integer and $i \geq 0$

Replacing s by $\frac{pz}{t}$ in (3.4-6)-(3.4-8) gives

$$F\left(\frac{pz}{t}\right) = \left(\frac{pz}{t}U - \Lambda\right)^{-1} \quad (3.4-9)$$

$$F^{(1)}\left(\frac{pz}{t}\right) = (-1)^1 \left(\frac{pz}{t}U - \Lambda\right)^{-2} \quad (3.4-10)$$

... ..

$$F^{(i)}\left(\frac{pz}{t}\right) = (-1)^i i! \left(\frac{pz}{t}U - \Lambda\right)^{-(i+1)} \quad (3.4-11)$$

Since

$$\frac{pz}{h}U - \Lambda = \text{diag} \left\{ \frac{pz - \lambda_j h}{h} \right\} = \frac{p}{h} \text{diag} \left\{ z - \frac{\lambda_j h}{p} \right\} \quad (3.4-12)$$

$$\left(\frac{pz}{h}U - \Lambda\right)^{-1} = \frac{h}{p} \text{diag} \left\{ \frac{1}{z - \frac{\lambda_j h}{p}} \right\} \quad (3.4-13)$$

$$\left(\frac{pz}{h}U - \Lambda\right)^{-2} = \left(\frac{h}{p}\right)^2 \text{diag} \left\{ \frac{1}{\left(z - \frac{\lambda_j h}{p}\right)^2} \right\} \quad (3.4-14)$$

... ..

$$\left(\frac{pz}{h}U - \Lambda\right)^{-i} = \left(\frac{h}{p}\right)^i \text{diag} \left\{ \frac{1}{\left(z - \frac{\lambda_j h}{p}\right)^i} \right\}, j = 1, 2, \dots, m \quad (3.4-15)$$

Combining (3.4-13)-(3.4-15) with (3.4-9)-(3.4-11), gives

$$F\left(\frac{pz}{t}\right) = \frac{h}{p} \text{diag} \left\{ \frac{1}{z - \frac{\lambda_j h}{p}} \right\} \quad (3.4-16)$$

$$F^{(1)}\left(\frac{pz}{t}\right) = -\left(\frac{h}{p}\right)^2 \text{diag} \left\{ \frac{1}{\left(z - \frac{\lambda_j h}{p}\right)^2} \right\} \quad (3.4-17)$$

... ..

$$F^{(i)}\left(\frac{pz}{t}\right) = (-1)^i i! \left(\frac{h}{p}\right) \text{diag} \left\{ \frac{1}{\left(z - \frac{\lambda_j h}{p}\right)^{i+1}} \right\} \quad (3.4-18)$$

Taking an inversion of (3.4-5) gives

$$y_1 = T(L^{-1}(sU - \Lambda)^{-1})T^{-1}y_0 \quad (3.4-19)$$

where L^{-1} denotes the inversion of the Laplace transform.

To solve (3.4-19) an inversion form of $F\left(\frac{pz}{t}\right)$ with using $n=P-1$ degree of derivatives is used as follows

$$\begin{aligned} \hat{f}(t) = & -\frac{p}{t} \left(\frac{\left(\frac{p}{t}\right)^0}{0!} \sum_{i=1}^M A_{1i} F^{(0)}\left(\frac{pz_i}{t}\right) + \frac{\left(\frac{p}{t}\right)^1}{1!} \sum_{i=1}^M A_{2i} F^{(1)}\left(\frac{pz_i}{t}\right) \right) \\ & - \dots - \frac{p}{t} \left(\frac{\left(\frac{p}{t}\right)^{p-1}}{(p-1)!} \sum_{i=1}^M A_{pi} F^{(p-1)}\left(\frac{pz_i}{t}\right) \right) \end{aligned} \quad (3.4-20)$$

The derivation of (3.4-20) will be given at the end of this section.

Applying the inversion form (3.4-20) into (3.4-19) with $t=h$

$$\begin{aligned}
 y_1 &= T(L^{-1}(sU - \Lambda))Ty_0 \\
 &= T \left(-\frac{p}{h} \left(\frac{\left(\frac{p}{h}\right)^0}{0!} \sum_{i=1}^M A_{1i} F^{(0)}\left(\frac{pz_i}{h}\right) + \frac{\left(\frac{p}{h}\right)^1}{1!} \sum_{i=1}^M A_{2i} F^{(1)}\left(\frac{pz_i}{h}\right) \right) \right) T^{-1}y_0 + \dots \\
 &\quad + T \left(-\frac{p}{h} \left(\frac{\left(\frac{p}{h}\right)^{p-1}}{(p-1)!} \sum_{i=1}^M A_{pi} F^{(p-1)}\left(\frac{pz_i}{h}\right) \right) \right) T^{-1}y_0
 \end{aligned} \tag{3.4-21}$$

Substituting (3.4-16) - (3.4-18) into (3.4-21) gives

$$\begin{aligned}
 y &= \\
 T \left[\begin{array}{ccc} \sum_{i=1}^M \frac{A_{1i}}{\lambda_1 h \frac{p}{p} - z_i} + \dots + \sum_{i=1}^M \frac{A_{pi}}{\lambda_1 h \frac{p}{p} - z_i} & \dots & 0 \\ \dots & \dots & \dots \\ 0 \dots & \sum_{i=1}^M \frac{A_{1i}}{\lambda_p h \frac{p}{p} - z_i} + \dots + \sum_{i=1}^M \frac{A_{pi}}{\lambda_p h \frac{p}{p} - z_i} & \dots \end{array} \right] T^{-1}y_0 \\
 &= TWT^{-1}y_0
 \end{aligned} \tag{3.4-22}$$

where

$$W = \left[\begin{array}{ccc} \sum_{i=1}^M \frac{A_{1i}}{\lambda_1 h \frac{p}{p} - z_i} + \dots + \sum_{i=1}^M \frac{A_{pi}}{\lambda_1 h \frac{p}{p} - z_i} & \dots & 0 \\ \dots & \dots & \dots \\ 0 \dots & \sum_{i=1}^M \frac{A_{1i}}{\lambda_p h \frac{p}{p} - z_i} + \dots + \sum_{i=1}^M \frac{A_{pi}}{\lambda_p h \frac{p}{p} - z_i} & \dots \end{array} \right] \tag{3.4-23}$$

It can be seen that taking k steps we obtain

$$y_k = TW^k T^{-1} y_0 \quad (3.4-24)$$

In order to keep the integration stable as $k \rightarrow \infty$ the entries of W^k must remain bounded which is equivalent to requiring for any eigenvalue λ

$$\left\| \sum_{i=1}^M \frac{A_{1i}}{\frac{\lambda h}{p} - z_i} + \dots + \sum_{i=1}^M \frac{A_{pi}}{\left(\frac{\lambda h}{p} - z_i\right)^p} \right\| \leq 1 \quad (3.4-25)$$

The equation (3.3-25) is the exactly same as (37) in [21] except for $z = \frac{\lambda h}{p}$.

Derivation of (3.4-20)

Expand the Pade approximation of e^{pz} to be the following form

$$\begin{aligned} \xi_{N,M}^p(z) &= \left(\sum_{i=1}^M \frac{k_i}{z - z_i} \right)^p \\ &= \sum_{i=1}^M \frac{A_{1i}}{z - z_i} + \sum_{i=1}^M \frac{A_{2i}}{(z - z_i)^2} + \dots + \sum_{i=1}^M \frac{A_{pi}}{(z - z_i)^p} \end{aligned} \quad (3.4-26)$$

where A_{ki} are coefficients of $(z - z_i)^{-k}$.

Then the approximation of $f(t)$ is given by

$$\begin{aligned}
 \hat{f}(t) &= \frac{p}{2\pi jt} \int_{(c'-j\infty)}^{(c'+j\infty)} F\left(\frac{pz}{t}\right) \xi_{N,M}^p(z) dz \\
 &= \frac{p}{2\pi jt} \int_{(c'-j\infty)}^{(c'+j\infty)} F\left(\frac{pz}{t}\right) \left(\sum_{i=1}^M \frac{A_{1i}}{z-z_i} + \sum_{i=1}^M \frac{A_{2i}}{(z-z_i)^2} + \dots + \sum_{i=1}^M \frac{A_{pi}}{(z-z_i)^p} \right) dz
 \end{aligned} \quad (3.4-27)$$

For stable networks all poles of $F(s)$ are at the left side of the imaginary axis. Choosing the direction of the integral of (3.4-27) to be clockwise we can have all poles of $F(s)$ at the outside of the integral loop. This implies that a negative sign will be chosen to solve the residues of $F\left(\frac{pz}{t}\right) \xi_{N,M}^p(z)$ given by

$$\begin{aligned}
 &-2\pi j \left(\frac{\left(\frac{p}{t}\right)^0}{0!} \sum_{i=1}^M A_{1i} F^{(0)}\left(\frac{pz_i}{t}\right) + \frac{\left(\frac{p}{t}\right)^1}{1!} \sum_{i=1}^M A_{2i} F^{(1)}\left(\frac{pz_i}{t}\right) \right) \\
 &- \dots - 2\pi j \left(\frac{\left(\frac{p}{t}\right)^{p-1}}{(p-1)!} \sum_{i=1}^M A_{pi} F^{(p-1)}\left(\frac{pz_i}{t}\right) \right)
 \end{aligned} \quad (3.4-28)$$

Combining (3.4-27) with (3.4-28), $\hat{f}(t)$ can be written as:

$$\begin{aligned}
 \hat{f}(t) &= -\frac{p}{t} \left(\frac{\left(\frac{p}{t}\right)^0}{0!} \sum_{i=1}^M A_{1i} F^{(0)}\left(\frac{pz_i}{t}\right) + \frac{\left(\frac{p}{t}\right)^1}{1!} \sum_{i=1}^M A_{2i} F^{(1)}\left(\frac{pz_i}{t}\right) \right) \\
 &- \dots - \frac{p}{t} \left(\frac{\left(\frac{p}{t}\right)^{p-1}}{(p-1)!} \sum_{i=1}^M A_{pi} F^{(p-1)}\left(\frac{pz_i}{t}\right) \right)
 \end{aligned} \quad (3.4-29)$$

□

Compared to (3.4-25), it can be concluded that the stability region given by the Pade approximation of (3.4-26) is

$$\|\xi_{N,M}^p(z)\| \leq 1 \quad (3.4-30)$$

The stability regions of Pade approximations for NILTn with $N=M-2$ are shown in Fig. 3.1. The curves are symmetric about the real axis.

3.5 Examples

Example 3.1.

This is a test example to show the accuracy of the high degree NILT compared to NILT0. Consider the function $F(s) = \frac{1}{(s^2 + 1)}$ which has the time domain representation $f(t) = \sin t$. Fig. 3.2 shows time responses of NILT0, NILT1 and NILT2 with $M=10$ and $N=8$, respectively. The results show that high degree NILT improve accuracy significantly.

Example 3.2

Consider the circuit shown in Fig. 3.3. Comparison of the results obtained by using Hspice, NILT0, NILT1 and NILT2 is shown in Fig. 3.4. In this example, all NILT methods are used with $M=4$ and $N=2$. The scale of the vertical axis in Fig. 3.4 is 1volt/unit.

Fig. 3.4 Responses by Hspice, NILT0, NILT1 and NILT2 ($M/N=4/2$)

Example 3.3

Consider the circuit shown in Fig. 3.5. Time responses obtained by Hspice, NILT0, NILT1 and NILT2 are shown in Fig. 3.6 -Fig. 3.8 for $M/N=4/2$, $M/N=6/4$ and $M/N=8/6$, respectively. The results show that the response by NILT2 with 8 poles is very close to that of Hspice. Scales of vertical axes in Fig. 3.6-Fig. 3.8 are 1 volt/unit.

Fig. 3.6 Responses of the circuit shown in Fig. 3.5 ($M/N=4/2$)

Fig. 3.7 Responses of the circuit shown in Fig. 3.5 ($M/N=6/4$)

Chapter 4

Applications of the High Degree Numerical Inversion of Laplace Transform for Transient Analysis of Distributed Networks

4.1 Introduction

The numerical inversion of Laplace transform (NILT0) is an efficient tool for simulating time responses of linear lumped, distributed and mixed networks. It is a very important method for solving lossy multiconductor transmission line networks since conventional simulators, such as SPICE, can not be used. In chapter 3 the numerical inversion of Laplace transforms of high degrees (NILTn, $n > 0$) were introduced. The results of their applications in linear lumped networks have shown that the NILTn are more accurate than the NILT0 method and are also absolutely stable. In this chapter the applications of the NILTn for the solution of distributed multiconductor transmission lines are presented. The modified nodal admittance matrix stamps of the coupled transmission lines are derived in section 4.2. Implementation issues are discussed in section 4.3.

Several examples are presented in section 4.4. Comments on NILTn and its efficiency will be discussed in section 4.5.

4.2 Stamps of Coupled Transmission Lines for NILT1 and NILT2

Stamps for NILT1

Consider the s-domain MNA equation of a linear network with distributed subnetworks (2.5-1) shown as below

$$\left(A + sB + \sum_{k=1}^{N_r} P_k H_k P_k' \right) F(s) = J(s)$$

Differentiating (2.5-1) gives

$$\left(A + sB + \sum_{k=1}^{N_r} P_k H_k P_k' \right) F'(s) = J'(s) - \left(B + \sum_{k=1}^{N_r} P_k H_k' P_k' \right) F(s) \quad (4.2-1)$$

To solve (4.2-1) we have to solve the first derivative of H_k . Use (2.5-35) shown as follows

$$H_k(D_k) = \begin{bmatrix} S_{ki} E_{k1}(D_k) S_{kv}^{-1} & S_{ki} E_{k2}(D_k) S_{kv}^{-1} \\ S_{ki} E_{k2}(D_k) S_{kv}^{-1} & S_{ki} E_{k1}(D_k) S_{kv}^{-1} \end{bmatrix}$$

Taking the first derivative of (2.5-35) with respect to s gives

$$H'_k(D_k) = \begin{bmatrix} h'_{k11} & h'_{k12} \\ h'_{k21} & h'_{k22} \end{bmatrix} \quad (4.2-2)$$

where

$$h'_{k11} = h'_{k22} = \frac{\partial S_{ki}}{\partial s} E_{k1}(D_k) S_{kv}^{-1} + S_{ki} \frac{\partial E_{k1}(D_k)}{\partial s} S_{kv}^{-1} + S_{ki} E_{k1}(D_k) \frac{\partial S_{kv}^{-1}}{\partial s} \quad (4.2-3)$$

$$h'_{k12} = h'_{k21} = \frac{\partial S_{ki}}{\partial s} E_{k2}(D_k) S_{kv}^{-1} + S_{ki} \frac{\partial E_{k2}(D_k)}{\partial s} S_{kv}^{-1} + S_{ki} E_{k2}(D_k) \frac{\partial S_{kv}^{-1}}{\partial s} \quad (4.2-4)$$

All notations above are the same as that in chapter 2 and chapter 3. (4.2-2)-(4.2-4) are the stamps required by NILT1. The following is the derivations of the terms in (4.2-2) -(4.2-4).

(a) *Computation of $\frac{\partial S_{kv}}{\partial s}$ and $\frac{\partial \gamma_{ki}}{\partial s}$*

$\frac{\partial S_{kv}}{\partial s}$ and $\frac{\partial \gamma_{ki}}{\partial s}$ are computed by differentiating (2.3-19) with respect to s as follows

$$(2\gamma_{ki} \frac{\partial \gamma_{ki}}{\partial s} U - \frac{\partial Z_k}{\partial s} Y_k - Z_k \frac{\partial Y_k}{\partial s}) S_{kvi} + (\gamma_{ki}^2 U - Z_k Y_k) \frac{\partial S_{kvi}}{\partial s} = 0 \quad (4.2-5)$$

i.e.

$$(2\gamma_{ki} \frac{\partial \gamma_{ki}}{\partial s} U - (L_k Y_k + Z_k C_k)) S_{kvi} + (\gamma_{ki}^2 U - Z_k Y_k) \frac{\partial S_{kvi}}{\partial s} = 0 \quad (4.2-6)$$

There are two variables in (4.2-6), which are $\frac{\partial S_{kv}}{\partial s}$ and $\frac{\partial \gamma_{ki}}{\partial s}$. To solve (4.2-6) another equation is required. Assume that the eigenvector S_{kvi} is normalized such that

$$S_{kvi}' \frac{\partial S_{kvi}}{\partial s} = 0 \quad (4.2-7)$$

(4.2-6) and (4.2-7) are written into a matrix form as follows

$$\begin{bmatrix} (\gamma_{ki}^2 U - Z_k Y_k) & 2\gamma_{ki} S_{kvi} \\ S'_{kvi} & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial S_{kvi}}{\partial s} \\ \frac{\partial \gamma'_{ki}}{\partial s} \end{bmatrix} = \begin{bmatrix} (L_k Y_k + Z_k C_k) S_{kvi} \\ 0 \end{bmatrix} \quad (4.2-8)$$

where $i=1, 2, \dots, m_k$ (m_k is the dimension of the k th coupled transmission line)

Solving (4.2-8) for $i=1, 2, \dots, m_k$ results in the two matrices

$$\frac{\partial S_{kv}}{\partial s} = \text{diag} \left(\frac{\partial S_{kv1}}{\partial s} \quad \frac{\partial S_{kv2}}{\partial s} \quad \dots \quad \frac{\partial S_{kv m_k}}{\partial s} \right) \quad (4.2-9)$$

$$\frac{\partial \Gamma_k}{\partial s} = \text{diag} \left(\frac{\partial \gamma_{k1}}{\partial s} \quad \frac{\partial \gamma_{k2}}{\partial s} \quad \dots \quad \frac{\partial \gamma_{k m_k}}{\partial s} \right) \quad (4.2-10)$$

(b) Computation of $\frac{\partial Z_k^{-1}}{\partial s}$

Let

$$T_k = Z_k^{-1} \quad (4.2-11)$$

Premultiplying (4.2-11) by Z_k gives

$$Z_k T_k = U \quad (4.2-12)$$

Differentiating (4.2-12) results in

$$\frac{\partial Z_k}{\partial s} T_k + Z_k \frac{\partial T_k}{\partial s} = 0 \quad (4.2-13)$$

which gives the explicit expression of $\frac{\partial Z_k^{-1}}{\partial s}$ as

$$\frac{\partial Z_k^{-1}}{\partial s} = \frac{\partial T_k}{\partial s} = -Z_k^{-1} \frac{\partial Z_k}{\partial s} Z_k^{-1} = -Z_k^{-1} L_k Z_k^{-1} \quad (4.2-14)$$

(c) Computation of $\frac{\partial S_{ki}}{\partial s}$

From (2.3-27)

$$\frac{\partial S_{ki}}{\partial s} = \frac{\partial Z_k^{-1}}{\partial s} S_{kv} \Gamma_k + Z_k^{-1} \frac{\partial S_{kv}}{\partial s} \Gamma_k + Z_k^{-1} S_{kv} \frac{\partial \Gamma_k}{\partial s} \quad (4.2-15)$$

(d) Computation of $\frac{\partial E_{k1}}{\partial s}$

Let

$$X_k = \frac{e^{\gamma_k D_k} + e^{-\gamma_k D_k}}{e^{\gamma_k D_k} - e^{-\gamma_k D_k}} \quad (4.2-16)$$

which gives

$$(e^{\gamma_k D_k} - e^{-\gamma_k D_k}) X_k = e^{\gamma_k D_k} + e^{-\gamma_k D_k} \quad (4.2-17)$$

Differentiating (4.2-17) gives

$$\begin{aligned} (D_k \frac{\partial \gamma_k}{\partial s} e^{\gamma_k D_k} + D_k \frac{\partial \gamma_k}{\partial s} e^{-\gamma_k D_k}) X_k + (e^{\gamma_k D_k} - e^{-\gamma_k D_k}) \frac{\partial X_k}{\partial s} \\ = D_k \frac{\partial \gamma_k}{\partial s} e^{\gamma_k D_k} - D_k \frac{\partial \gamma_k}{\partial s} e^{-\gamma_k D_k} \end{aligned} \quad (4.2-18)$$

in which $\frac{\partial X_k}{\partial s}$ can be written as:

$$\frac{\partial X_k}{\partial s} = \frac{D_k \frac{\partial \gamma_k}{\partial s} (e^{\gamma_k D_k} - e^{-\gamma_k D_k}) - D_k \frac{\partial \gamma_k}{\partial s} (e^{\gamma_k D_k} + e^{-\gamma_k D_k}) X_k}{e^{\gamma_k D_k} - e^{-\gamma_k D_k}} = D_k \frac{\partial \gamma_k}{\partial s} (1 - X_k^2) \quad (4.2-19)$$

From (4.2-19) $\frac{\partial E_{k1}}{\partial s}$ can be written as

$$\frac{\partial E_{k1}}{\partial s} = \text{diag} \left(D_k \frac{\partial \gamma_{ki}}{\partial s} (1 - E_{k1}^2(i, i)) \right) \quad (4.2-20)$$

Similarly,

$$\frac{\partial E_{k2}}{\partial s} = \text{diag} \left(-D_k \frac{\partial \gamma_{ki}}{\partial s} E_{k1}(i, i) E_{k2}(i, i) \right) \quad (4.2-21)$$

where $1 \leq i \leq m_k$

(e) *Computation of $\frac{\partial S_{kv}^{-1}}{\partial s}$*

Following the method of solving Z_k^{-1} , $\frac{\partial S_{kv}^{-1}}{\partial s}$ can be written as

$$\frac{\partial S_{kv}^{-1}}{\partial s} = -S_{kv}^{-1} \frac{\partial S_{kv}}{\partial s} S_{kv}^{-1} \quad (4.2-22)$$

Stamps for NILT2

In order to get the second derivative of $F(s)$ required by NILT2, we differentiate (4.2-1) again to obtain the following equation

$$\begin{aligned}
 & \left(A + sB + \sum_{k=1}^{N_T} P_k H_k P_k' \right) F''(s) \\
 & = J''(s) - 2 \left(B + \sum_{k=1}^{N_T} P_k H_k' P_k' \right) F'(s) - \left(\sum_{k=1}^{N_T} P_k H_k'' P_k' \right) F(s)
 \end{aligned} \tag{4.2-23}$$

Differentiate (4.2-2) and combine the results with (4.2-3) and (4.2-4)

$$H_k''(D_k) = \begin{bmatrix} h''_{k11} & h''_{k12} \\ h''_{k21} & h''_{k22} \end{bmatrix} \tag{4.2-24}$$

$$\begin{aligned}
 h''_{k11} = h''_{k22} = & \frac{\partial^2 S_{ki}}{\partial s^2} E_{k1}(D_k) S_{kv}^{-1} + S_{ki} \frac{\partial^2 E_{k1}(D_k)}{\partial s^2} S_{kv}^{-1} + S_{ki} E_{k1}(D_k) \frac{\partial^2 S_{kv}^{-1}}{\partial s^2} \\
 & + 2 \frac{\partial S_{ki}}{\partial s} \frac{\partial E_{k1}(D_k)}{\partial s} S_{kv}^{-1} + 2 S_{ki} \frac{\partial E_{k1}(D_k)}{\partial s} \frac{\partial S_{kv}^{-1}}{\partial s} + 2 \frac{\partial S_{ki}}{\partial s} E_{k1}(D_k) \frac{\partial S_{kv}^{-1}}{\partial s}
 \end{aligned} \tag{4.2-25}$$

$$\begin{aligned}
 h''_{k12} = h''_{k21} = & \frac{\partial^2 S_{ki}}{\partial s^2} E_{k2} S_{kv}^{-1} + S_{ki} \frac{\partial^2 E_{k2}}{\partial s^2} S_{kv}^{-1} + S_{ki} E_{k2} \frac{\partial^2 S_{kv}^{-1}}{\partial s^2} \\
 & + 2 \frac{\partial S_{ki}}{\partial s} \frac{\partial E_{k2}}{\partial s} S_{kv}^{-1} + 2 S_{ki} \frac{\partial E_{k2}}{\partial s} \frac{\partial S_{kv}^{-1}}{\partial s} + 2 \frac{\partial S_{ki}}{\partial s} E_{k2} \frac{\partial S_{kv}^{-1}}{\partial s}
 \end{aligned} \tag{4.2-26}$$

Equations (4.2-24) - (4.2-26) are the stamps required by NILT2. Following are the derivations of the terms in equations (4.2-24) - (4.2-26).

(a) Computation of $\frac{\partial^2 S_{kvi}}{\partial s^2}$ and $\frac{\partial^2 \gamma_{ki}}{\partial s^2}$

Differentiating (4.2-6) and (4.2-7) gives

$$\begin{aligned}
& \left(2 \left(\frac{\partial \gamma_{ki}}{\partial s} \right)^2 U + 2 \gamma_{ki} \frac{\partial^2 \gamma_{ki}}{\partial s^2} - 2 L_k C_k \right) S_{kvi} + \left(2 \gamma_{ki} \frac{\partial \gamma_{ki}}{\partial s} U - (L_k Y_k + Z_k C_k) \right) \frac{\partial S_{kvi}}{\partial s} \\
& + \left(2 \gamma_{ki} \frac{\partial \gamma_{ki}}{\partial s} U - (L_k Y_k + Z_k C_k) \right) \frac{\partial S_{kvi}}{\partial s} + \left(\gamma_{ki}^2 U - Z_k Y_k \right) \frac{\partial^2 S_{kvi}}{\partial s^2} = 0
\end{aligned} \quad (4.2-27)$$

$$\left(\frac{\partial S_{vi}}{\partial s} \right)' \frac{\partial S_{vi}}{\partial s} + S_{vi}' \frac{\partial^2 S_{vi}}{\partial s^2} = 0 \quad (4.2-28)$$

or

$$\begin{aligned}
& \begin{bmatrix} (\gamma_{ki}^2 U - Z_k Y_k) & 2 \gamma_{ki} S_{kvi} \\ S_{kvi}' & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial^2 S_{kvi}}{\partial s^2} \\ \frac{\partial^2 \gamma_{ki}}{\partial s^2} \end{bmatrix} \\
& = \begin{bmatrix} 2 \left(L_k C_k - \left(\frac{\partial \gamma_{ki}}{\partial s} \right)^2 U \right) S_{kvi} + 2 \left((L_k Y_k + Z_k C_k) - 2 \gamma_{ki} \frac{\partial \gamma_{ki}}{\partial s} U \right) \frac{\partial S_{kvi}}{\partial s} \\ - \left(\frac{\partial S_{kvi}}{\partial s} \right)' \frac{\partial S_{kvi}}{\partial s} \end{bmatrix}
\end{aligned} \quad (4.2-29)$$

It is to be noted that the coefficient matrices of the left hand sides of (4.2-8) and (4.2-29) are the same. This means that only one LU factorization is needed to solve both equations.

Solving (4.2-29) for $i=1,2,\dots, m_k$ results in the two matrices

$$\frac{\partial^2 S_{kv}}{\partial s^2} = \begin{pmatrix} \frac{\partial^2 S_{kv1}}{\partial s^2} & \frac{\partial^2 S_{kv2}}{\partial s^2} & \dots & \frac{\partial^2 S_{kv m_k}}{\partial s^2} \end{pmatrix} \quad (4.2-30)$$

$$\frac{\partial^2 \Upsilon_k}{\partial s^2} = \text{diag} \left(\frac{\partial^2 \gamma_{k1}}{\partial s^2}, \frac{\partial^2 \gamma_{k2}}{\partial s^2}, \dots, \frac{\partial^2 \gamma_{k m_k}}{\partial s^2} \right) \quad (4.2-31)$$

(b) Computation of $\frac{\partial^2 Z_k^{-1}}{\partial s^2}$

Taking the first derivative of (4.2-14) gives

$$\frac{\partial^2 Z_k^{-1}}{\partial s^2} = - \left(\frac{\partial Z_k^{-1}}{\partial s} L_k Z_k^{-1} + Z_k^{-1} L_k \frac{\partial Z_k^{-1}}{\partial s} \right) \quad (4.2-32)$$

(c) Computation of $\frac{\partial^2 S_{ki}}{\partial s^2}$

Differentiating (4.2-15) again gives

$$\begin{aligned} \frac{\partial^2 S_{ki}}{\partial s^2} &= \frac{\partial^2 Z_k^{-1}}{\partial s^2} S_{kv} \Gamma_k + Z_k^{-1} \frac{\partial^2 S_{kv}}{\partial s^2} \Gamma_k + Z_k^{-1} S_{kv} \frac{\partial^2 \Gamma_k}{\partial s^2} \\ &+ 2 \frac{\partial Z_k^{-1}}{\partial s} \frac{\partial S_{kv}}{\partial s} \Gamma_k + 2 \frac{\partial Z_k^{-1}}{\partial s} S_{kv} \frac{\partial \Gamma_k}{\partial s} + 2 Z_k^{-1} \frac{\partial S_{kv}}{\partial s} \frac{\partial \Gamma_k}{\partial s} \end{aligned} \quad (4.2-33)$$

(d) Computation of $\frac{\partial^2 E_{k1}}{\partial s^2}$ and $\frac{\partial^2 E_{k2}}{\partial s^2}$

Differentiating (4.2-20) and (4.2-21) again gives

$$\frac{\partial^2 E_{k1}}{\partial s^2} = D_k \text{diag} \left(\frac{\partial^2 \gamma_{ki}}{\partial s^2} (1 - E_{k1}^2(i, i)) - 2 \frac{\partial \gamma_{ki}}{\partial s} E_{k1}(i, i) \frac{\partial E_{k1}(i, i)}{\partial s} \right) \quad (4.2-34)$$

and

$$\begin{aligned} \frac{\partial^2 E_{k2}}{\partial s^2} &= -D_k \bullet \\ &\bullet \text{diag} \left(\frac{\partial^2 \gamma_{ki}}{\partial s^2} E_{k1}(i, i) E_{k2}(i, i) + \frac{\partial \gamma_{ki}}{\partial s} \frac{\partial E_{k1}(i, i)}{\partial s} E_{k2}(i, i) + \frac{\partial \gamma_{ki}}{\partial s} E_{k1}(i, i) \frac{\partial E_{k2}(i, i)}{\partial s} \right) \end{aligned} \quad (4.2-35)$$

(e) Computation of $\frac{\partial^2}{\partial s^2} S_{kv}^{-1}$

Differentiating (4.2-22) again gives

$$\frac{\partial^2}{\partial s^2} S_{kv}^{-1} = - \left(\frac{\partial S_{kv}^{-1}}{\partial s} \frac{\partial S_{kv}}{\partial s} S_{kv}^{-1} + S_{kv}^{-1} \frac{\partial^2 S_{kv}}{\partial s^2} S_{kv}^{-1} + S_{kv}^{-1} \frac{\partial S_{kv}}{\partial s} \frac{\partial S_{kv}^{-1}}{\partial s} \right) \quad (4.2-36)$$

4.3 Implementations of NILTn for Distributed Networks

The coefficient matrices of the left hand sides of (2.5-1), (4.2-1) and (4.2-23) are the same. This implies that compared with NILT0, the incremental cost of NILTn is n backward-forward substitutions. On the other hand, high degree NILT methods can not only provide more accurate results, but also save CPU time compared to NILT0 with more poles. For example, assume that 22-pole of NILT0 has the same accuracy of 8-pole of NILT2. The first method needs 11 LUs plus 11 backward-forwards, the latter needs 4 LUs plus 12 backward-forwards. In the next section, comparisons of accuracy between NILT0 with 22 poles and NILT2 with 8 poles are given for all examples. In section 4.5, CPU times are counted for NILT0 with 22 poles and NILT2 with 8 poles for computing a 35 single transmission line circuit. The results show that NILT2 with 8 poles can save two-thirds CPU time compared to NILT0 with 22 poles.

4.4 Examples

Example 4.1 One single transmission line circuit.

A single lossless transmission line circuit is shown in Fig. 4.1. Parameters of the line are $L=60\text{nH}$, $C=100\text{pF}$ and $D=0.5\text{m}$. The input pulse has 1.5ns rise and fall times and 10ns width. The response is computed with $M=4$ and $N=2$. Comparison of results of

Hspice with the results of NILT0, NILT1 and NILT2 is shown in Fig. 4.2. A comparison of the accuracy between NILT0 (22 poles) and NILT2 (8 poles) is shown in Fig. 4.3. Scales of vertical axes in Fig. 4.2 - Fig. 4.3 are 1volt/unit.

Fig.4.2 Comparison of results of Hspice, NILT0, NILT1 and NILT2 for circuit shown in Fig. 4.1.

Fig. 4.3 Comparison of the accuracy between NILT0 (22poles) and NILT2 (8poles) for the circuit of example 4.1

Example 4.2 Seven single lossless transmission line circuit.

Consider the circuit shown in Fig. 4.4. The transient response is calculated by using $M/N=4/2$, $6/4$, $8/6$ and $10/8$, respectively. The input pulse has zero rise and fall times and 10ns width. The results are shown in Fig. 4.5 - Fig. 4.8, which verify that the high degree methods lead to more accurate results in all cases. A comparison of accuracy between NILT0 (22 poles) and NILT2 (8 poles) is shown in Fig. 4.9. The scales of vertical axes in Fig. 4.5 - Fig. 4.9 are 1volt/unit.

Fig.4.7 Comparison of Hspice result with the results of NILT0, NILT1 and NILT2 for the circuit shown in Fig. 4.4 ($M/N=8/6$, lossless)

Fig.4.8 Comparison of Hspice result with the results of NILT0, NILT1 and NILT2 for the circuit shown in Fig. 4.4 ($M/N=10/8$, lossless)

Example 4.3 Seven single lossy transmission line circuit.

The circuit is the same as that of example 4.2 except for replacing the lossless lines with lossy ones with $R=75\Omega$ and $G=0.01 \text{ } 1/\Omega$. Since Hspice can not handle lossy transmission line networks, comparisons are made between NILT0, NILT1 and NILT2. Results are shown in Fig. 4.10 - Fig. 4.13 for $M/N=4/2, 6/4, 8/6$ and $10/8$, respectively. A comparison of the accuracy between NILT0 (22 poles) and NILT2 (8 poles) for this example is shown in Fig. 4.14. Scales of vertical axes in Fig. 4.10-Fig. 4.14 are 1volt/unit.

Example 4.4 *One multiconductor transmission line circuit.*

The circuit is shown in Fig. 4.15. The line parameters are given by

$$L = \begin{bmatrix} 494.6 & 63.3 \\ 63.3 & 494.6 \end{bmatrix} \text{ nH/m}$$

$$C = \begin{bmatrix} 62.8 & -4.9 \\ -4.9 & 62.8 \end{bmatrix} \text{ pF/m}$$

$$D = 0.3048m$$

The responses of the circuit are shown in Fig. 4.16- Fig.4.19 for node 1 - node 4 with $M=10$ and $N=8$. The input pulse has a 1.5ns rise and fall times and 4.5ns width. A comparison of the accuracy between NILT0 (22 poles) and NILT2 (8 poles) for the time response at node

4 is shown in Fig. 4.20. Scales of vertical axes in Fig. 4.16 - Fig. 4.20 are 1volt/unit.

Fig. 4.17 Time response of $v(2)$ of the circuit shown in Fig. 4.15 (lossless)

Fig. 4.18 Time response of $v(3)$ of the circuit shown in Fig. 4.15 (lossless)

Example 4.5 One lossy multiconductor transmission line circuit.

This example is the same as example 4.4 shown in Fig. 4.15, except that the lossless transmission line is replaced by lossy one with

$$R = \begin{bmatrix} 0.1 & 0.02 \\ 0.02 & 0.1 \end{bmatrix} \Omega/\text{m}$$

$$G = \begin{bmatrix} 0.1 & -0.01 \\ -0.01 & 0.1 \end{bmatrix} \text{ s/m}$$

The input pulse and poles used are the same as those of example 4.4. The results are shown in Fig.4.21 -Fig.4.24 for node 1 - node 4. A comparison of the accuracy between NILT0 (22 poles) and NILT2 (8 poles) for the response at node 4 is shown in Fig. 4.25.

Fig.4.21 Time response of $v(1)$ of the circuit shown in Fig.4.15 (lossy)

Fig. 4.24 Time response of $v(4)$ of the circuit shown in Fig. 4.15 (lossy)

Fig. 4.25 Comparison of the accuracy between $NILT0$ (22poles) and $NILT2$ (8poles) for the response $v(4)$ in the circuit of example 4.5 (lossy)

Example 4.6 Two multiconductor transmission line circuit.

The circuit is shown in Fig.4.26 with the following line parameters

For the Line 1# $D_1 = 0.1m$, and

$$R_1 = \begin{bmatrix} 75 & 15 \\ 15 & 75 \end{bmatrix} \Omega/m$$

$$G_1 = \begin{bmatrix} 0.1 & -0.01 \\ -0.01 & 0.1 \end{bmatrix} S/m$$

$$L_1 = \begin{bmatrix} 494.6 & 63.3 \\ 63.3 & 494.6 \end{bmatrix} nH/m$$

$$C_1 = \begin{bmatrix} 62.8 & -4.9 \\ -4.9 & 62.8 \end{bmatrix} pF/m$$

For Line 2# $D_2 = 0.1m$, and

$$R_2 = \begin{bmatrix} 50 & 10 & 1 & 0 \\ 10 & 50 & 10 & 1 \\ 1 & 10 & 50 & 10 \\ 0 & 1 & 10 & 50 \end{bmatrix} \Omega/m$$

$$G_2 = \begin{bmatrix} 0.1 & -0.01 & -0.001 & 0 \\ -0.01 & 0.1 & -0.01 & -0.001 \\ -0.001 & -0.01 & 0.1 & -0.01 \\ 0 & -0.001 & -0.01 & 0.1 \end{bmatrix} S/m$$

$$L_2 = \begin{bmatrix} 494.6 & 63.3 & 7.8 & 0 \\ 63.3 & 494.6 & 63.3 & 7.8 \\ 7.8 & 63.3 & 494.6 & 63.3 \\ 0 & 7.8 & 63.3 & 494.6 \end{bmatrix} nH/m$$

$$C_2 = \begin{bmatrix} 62.8 & -4.9 & -0.3 & 0 \\ -4.9 & 62.8 & -4.9 & -0.3 \\ -0.3 & -4.9 & 62.8 & -4.9 \\ 0 & -0.3 & -4.9 & 62.8 \end{bmatrix} \text{ pF/m}$$

The input pulse has 1ns rise and fall times and 3ns width. The comparisons of NILT0 - NILT2 are shown in Fig. 4.27- Fig. 4.30 for voltage outputs of node 3, 4, 7 and 15, respectively. The response is calculated with $M/N=10/8$. A comparison of the accuracy between NILT0 (22 poles) and NILT2 (8 poles) for the output of node 7 is shown in Fig. 4.31. Scales of vertical axes in Fig. 4.27 - Fig. 4.31 are 1volt/unit.

Fig. 4.26 Two multiconductor TL circuit for example 4.6.

2

of/de

2

PM-1 3½"x4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

Fig. 4.27 Time responses of $V(3)$ in Fig. 4.26. Fig. 4.28 Time responses of $V(4)$ in Fig. 4.26.

Fig.4.29 Time responses of $V(7)$ in Fig. 4.26. Fig.4.30 Time responses of $V(15)$ in Fig. 4.26

Example 4.7 Three multiconductor lossless transmission line circuit

Consider the circuit shown in Fig. 4.32. The input pulse has 1.5ns rise and fall time and 4.5ns width. The response is calculated with $M=10$ and $N=8$. Fig. 4.33 and Fig. 4.34 show the responses at nodes 2 and 5, respectively.

Transmission line parameters are described in the following. For both Line #1 and Line #2 $R_1 = G_1 = R_2 = G_2 = 0$ and $D_1 = D_2 = 0.3048m$

$$L_1 = L_2 = \begin{bmatrix} 494.6 & 63.3 \\ 63.3 & 494.6 \end{bmatrix} \text{ nH/m}$$

$$C_1 = C_2 = \begin{bmatrix} 62.8 & -4.9 \\ -4.9 & 62.8 \end{bmatrix} \text{ pF/m}$$

For Line #3 $R_3 = G_3 = 0$, $D_3 = 0.3048m$, $L_3 = 494.6nH/m$ and $C_3 = 62.8pF/m$.

A comparison of the accuracy between NILT0 (22 poles) and NILT2 (8 poles) for the output of node 2 is shown in Fig. 4.35. Scales of vertical axes in Fig. 4.33 - Fig. 4.35 are 1volt/unit.

Example 4.8 Thirty-five single transmission line circuit.

Consider the circuit shown in Fig. 4.36 which is obtained by cascading the circuit shown in Fig. 4.4. This example is calculated with $M/N=4/2$, $M/N=6/4$, $M/N=8/6$ and $M/N=10/8$ shown in Fig. 4.37 - Fig. 4.40, respectively. The input pulse has 1ns rise and fall times and 10ns width.

A comparison of the accuracy between NILT0 (22poles) and (NILT2 (8 poles) is shown in Fig. 4.41.

Fig. 4.36 Thirty-five single transmission line circuit for example 4.8.

4.5 Comments on NILTn

a. Accuracy

Chapters 3 and 4 have shown that the high degree numerical inversion of Laplace transforms improve the accuracy of the results significantly for linear lumped, distributed and mixed networks.

b. Efficiency

Not only do high degree numerical inversion of Laplace transforms improve accuracy, but they are also more efficient than NILT0 with comparable accuracy. With high degree and less poles, computer CPU time can be saved. In the last section, the results show that NILT2 with 8 poles has the same accuracy as NILT0 with 22 poles for all examples. To solve the 35 single transmission line circuit shown in Fig. 4.36, the relative CPU time cost of NILT2 with 8 poles with respect to NILT0 with 22 poles is found to be 0.3355 (68.09s/202.93s). To solve the system equation, the first method requires 4 LUs plus 12 backward-forwards and the second needs 11 LUs plus 11 backward-forwards. Each additional pair of conjugate poles requires one more LU decomposition which is the main cost of NILT method.

c. Stability

In chapter 3 it was proved that high degree numerical inversion of Laplace transforms are absolutely stable, i.e. the stability is the same as that of the NILT0.

Chapter 5

A Stepping Algorithm for the Numerical Inversion of Laplace Transform

5.1. Introduction

The NILT methods described in the previous chapters possess the disadvantage that accuracy decreases as time increases. This can be seen by looking at truncation error equations (2.5-8) and (3.3-26), where both Ψ_0 and Ψ_n are the functions of t^{N+M+1} . In [20][21], a solution was proposed, in which the NILT method was used with small time intervals where the accuracy was excellent. The problem was reset so that in the next evaluation the previous result at the resetting time was considered as a new reference point for the next step. Unfortunately, the reported solutions in the literature [20][21] were limited to lumped networks and rational functions.

In this chapter, the resetting principle is applied to networks which contain lossy multiconductor transmission lines. The proposed method will be proved more accurate than the standard NILT method. In section 5.2 a brief review of the stepping

algorithm for lumped networks is given. The extension of the stepping algorithm for distributed networks will be discussed in section 5.3. The concrete implementations of the method will be presented in section 5.4. To show that the new method can improve accuracy, several examples are presented in section 5.5. Finally, the comments on the stepping algorithm are given in section 5.6.

5.2. The Stepping Algorithm

The resetting technique was used [20][21] for linear lumped component networks, in which any time can be selected as a reference time by taking into account the voltages across the capacitors and the current through the inductors as equivalent independent sources. By using modified nodal formulation, one can simply change the s-domain MNA equation

$$(A + sB)F(s) = J(s) \quad (5.2-1)$$

into

$$(A + sB)F(s) = J(s) + Bf(t_0) \quad (5.2-2)$$

The definitions of the symbols used in (5.2-1) and (5.2-2) are the same as those used in the previous chapters. $f(t)$ is the time domain response related to $F(s)$; t_0 is the resetting time. Equation (5.2-2) is the formula for solving the time response after $t = t_0$ for linear lumped networks. The principle of the resetting technique is that, first, one starts using (5.2-1) to solve the time responses with zero initial conditions of capacitances and inductances within time interval t_0 and stops at $t = t_0$; second, the networks are solved by (5.2-2) within a new time interval, and all initial conditions of capacitances and inductances are replaced by independent voltage and current sources caused equivalently by voltages across capacitances and currents through inductances at $t = t_0$. These

equivalent independent sources are represented by $Bf(t_0)$ in (5.2-2). As long as the step size remains constant, the coefficient matrix in (5.2-2) will not change and one LU decomposition is sufficient. Therefore, one only needs to modify the right hand side of (5.2-2) and perform a backward-forward substitution at each resetting time.

5.3. Extension of The Stepping Algorithm

For distributed or mixed lumped and distributed networks, the resetting approach becomes more complicated, since the responses of linear distributed networks are two dimension functions of time t and line location x . The purpose of this section is to develop a formula of the resetting technique for distributed or mixed networks.

(a) Formulation of The Stepping Algorithm for Lossy Coupled Transmission Lines

For linear networks with lossy coupled transmission lines, the Generalized Modified Nodal Admittance equations in the s -domain (2.3-1) are as follows,

$$\left(A + sB + \sum_{k=1}^{N_T} P_k H_k P_k^t \right) F(s) = J(s) \quad (2.3-1)$$

Considering the initial conditions of both lumped elements and distributed elements at $t = t_0$, a generalized form of (2.3-1) is given by

$$\begin{aligned} & \left(A + sB + \sum_{k=1}^{N_T} P_k H_k P_k^t \right) F(s) \\ &= J(s) + Bf(t_0) + \sum_{k=1}^{N_T} P_k g_k(v_k(D_k, t_0), i_k(D_k, t_0)) \end{aligned} \quad (5.3-1)$$

where $g_k(v_k(D_k, t_0), i_k(D_k, t_0)) \in C^{2m_k}$ is a vector of equivalent independent sources contributed by the k th coupled transmission line,

$v_k(D_k, t_0)$ and $i_k(D_k, t_0)$ are terminal voltage and current vectors of the k th transmission line at time t_0 ,

D_k is the length of the k th transmission line,

In the section (b) $g_k(v_k(D_k, t_0), i_k(D_k, t_0))$ will be proved to be

$$\begin{aligned}
 & g_k(v_k(D_k, t_0), i_k(D_k, t_0)) \\
 = & - \int_0^{D_k} \begin{bmatrix} S_{ki} & 0 \\ 0 & S_{ki} \end{bmatrix} \begin{bmatrix} -E_k(D_k) & -U \\ E_k^{-1}(D_k) & U \end{bmatrix}^{-1} \begin{bmatrix} E_k(D_k - \tau) & 0 \\ 0 & E_k^{-1}(D_k - \tau) \end{bmatrix} \cdot \\
 & \begin{bmatrix} S_{kv}^{-1} & S_{ki}^{-1} \\ S_{kv}^{-1} & -S_{ki}^{-1} \end{bmatrix} \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \begin{bmatrix} v_k(\tau, t_0) \\ i_k(\tau, t_0) \end{bmatrix} d\tau
 \end{aligned} \quad (5.3-2)$$

The notations in (5.3-2) and all other symbols in (5.3-1) can be found in previous chapters.

(b) Derivation of the Equivalent Sources Caused by the Initial Conditions of the k th Transmission Line

To solve $g_k(v_k(D_k, t_0), i_k(D_k, t_0))$, it is necessary to start the derivation from the time domain partial differential equations of the k th transmission line given by

$$\frac{\partial v_k(x, t)}{\partial x} = -(R_k i_k(x, t) + L_k \frac{\partial i_k(x, t)}{\partial t}) \quad (5.3-3)$$

$$\frac{\partial i_k(x, t)}{\partial x} = -(G_k v_k(x, t) + C_k \frac{\partial v_k(x, t)}{\partial t}) \quad (5.3-4)$$

Apply the Laplace transform to these equations by assuming nonzero initial conditions,

$$\frac{dV_k(x, s)}{dx} = -(R_k + sL_k) I_k(x, s) = -Z_k I_k(x, s) + L_k i_k(x, t) \quad (5.3-5)$$

$$\frac{dI_k(x, s)}{dx} = -(G_k + sC_k) V_k(x, s) = -Y_k V_k(x, s) + C_k v_k(x, t) \quad (5.3-6)$$

where upper-case variables are defined in the s -domain, those of lower-case are in the time domain. $i_k(x, t_0)$ and $v_k(x, t_0)$ represent the k th transmission line current and voltage waveforms at time $t = t_0$ and at location x . Rewrite (5.3-5) and (5.3-6) in the form:

$$\begin{bmatrix} \frac{dV_k(x, s)}{dx} \\ \frac{dI_k(x, s)}{dx} \end{bmatrix} = \begin{bmatrix} 0 & -Z_k \\ -Y_k & 0 \end{bmatrix} \begin{bmatrix} V_k(x, s) \\ I_k(x, s) \end{bmatrix} + \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \begin{bmatrix} v_k(x, t_0) \\ i_k(x, t_0) \end{bmatrix} \quad (5.3-7)$$

Defining

$$\Phi_k(x, s) = \begin{bmatrix} V_k(x, s) \\ I_k(x, s) \end{bmatrix} \quad (5.3-8)$$

$$\beta_k = \begin{bmatrix} 0 & -Z_k \\ -Y_k & 0 \end{bmatrix} \quad (5.3-9)$$

$$b_k = \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \begin{bmatrix} v_k(x, t_0) \\ i_k(x, t_0) \end{bmatrix} = \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \phi_k(x, t_0) \quad (5.3-10)$$

where $\phi_k(x, t_0)$ is the Laplace inversion of $\Phi_k(x, s)$ at $t = t_0$. Then the simplified form of (5.3-7) can be written in the form

$$\Phi_k' = \beta_k \Phi_k + b_k \quad (5.3-11)$$

where $\Phi_k(x, s) \in C^{2m_k}$ is a complex vector whose elements are function of x and s . $b_k \in R^{2m_k}$ is a real number vector whose elements are functions of x and t_0 , and $\beta_k \in C^{2m_k \times 2m_k}$ is a complex constant matrix. Note that equation (5.3-11) is a normal linear state space equation.

The homogeneous form of (5.3-11) is given by

$$\Phi_k' = \beta_k \Phi_k \quad (5.3-12)$$

Before solving (5.3-11), the solution of its homogeneous form of (5.3-12) needs to be solved first.

Solution of Homogeneous Form of (5.3-12)

Assume that the solution of (5.3-12) can be written as

$$\Phi_k(x, s) = S_k \Lambda_k(x) K_k \quad (5.3-13)$$

where

$$\Lambda_k(x) = \text{diag}(e^{\gamma_{k1}x} \quad e^{\gamma_{k2}x} \quad \dots \quad e^{\gamma_{k2m_k}x}) \quad (5.3-14)$$

γ_{ki} are eigenvalues of β_k , $1 \leq i \leq 2m_k$

$$S_k = [S_{k1} \ S_{k2} \ \dots \ S_{k2m_k}] \quad (5.3-15)$$

S_{kj} are eigenvectors associated with the eigenvalues γ_{kj} , $1 \leq j \leq 2m_k$. K_k is a constant vector defined by the initial value of Φ_k .

In (5.3-13) let $x=0$

$$\Phi_k(0, s) = S_k U K_k \quad (5.3-16)$$

where U is the identity matrix. From (5.3-16) K_k can be written as

$$K_k = S_k^{-1} \Phi_k(0, s) \quad (5.3-17)$$

Applying (5.3-17) in (5.3-13), the solution of (5.3-12) can be written as

$$\Phi_k(x, s) = S_k \Lambda_k(x) S_k^{-1} \Phi_k(0, s) \quad (5.3-18)$$

In order to find eigenvalues of β_k , consider the following equation

$$(\gamma_k U - \beta_k) = \left(\gamma_k U - \begin{bmatrix} 0 & -Z_k \\ -Y_k & 0 \end{bmatrix} \right) = \begin{bmatrix} \gamma_k U & Z_k \\ Y_k & \gamma_k U \end{bmatrix} \quad (5.3-19)$$

Let $\det(\gamma_k U - \beta_k) = 0$, i.e.

$$\begin{vmatrix} \gamma_k U & Z_k \\ Y_k & \gamma_k U \end{vmatrix} = 0 \quad (5.3-20)$$

this yields

$$\gamma_k^2 U - Z_k Y_k = 0 \quad (5.3-21)$$

(5.3-21) is the same as (2.3-19). Using the solutions of (2.3-20) and (2.3-26) here

$$V_k(x, s) = S_{kv} E_k(x) K_{k1} + S_{kv} E_k^{-1}(x) K_{k2} \quad (5.3-22)$$

$$I_k(x, s) = S_{ki} E_k(x) K_{k1} - S_{ki} E_k^{-1}(x) K_{k2} \quad (5.3-23)$$

where all notations are the same as that in section 2.3. For a clear understanding, some explanations are repeated as follows

$$S_{kv} = \begin{bmatrix} S_{kv1} & S_{kv2} & \dots & S_{kv m_k} \end{bmatrix} \quad (5.3-24)$$

S_{kvj} are eigenvectors associated with γ_{kj}^2 which are the eigenvalues of $Z_k Y_k$, $1 \leq j \leq m_k$.

$$E_k(x) = \text{diag}(e^{-\gamma_{k1}x} \quad e^{-\gamma_{k2}x} \quad \dots e^{-\gamma_{km_k}x}) \quad (5.3-25)$$

$$S_{ki} = Z_k^{-1} S_{kv} \Gamma_k \quad (5.3-26)$$

$$\Gamma_k = \begin{bmatrix} \gamma_{k1} & 0 & \dots \\ 0 & \gamma_{k2} & \dots \\ \dots & \dots & \dots \\ 0 & 0 & \dots & \gamma_{km_k} \end{bmatrix} \quad (5.3-27)$$

K_{k1} and K_{k2} are constant vectors defined by the initial values of $V_k(x, s)$ and $I_k(x, s)$.

Rewrite (5.3-22) and (5.3-23) in matrix form

$$\begin{aligned} \begin{bmatrix} V_k(x, s) \\ I_k(x, s) \end{bmatrix} &= \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix} \begin{bmatrix} E_k(x) & 0 \\ 0 & E_k^{-1}(x) \end{bmatrix} \begin{bmatrix} K_{k1} \\ K_{k2} \end{bmatrix} \\ &= \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix} \begin{bmatrix} E_k(x) & 0 \\ 0 & E_k^{-1}(x) \end{bmatrix} \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix}^{-1} \begin{bmatrix} V_k(0, s) \\ I_k(0, s) \end{bmatrix} \end{aligned} \quad (5.3-28)$$

Compared to (5.3-28) the explicit expression of S_k and Λ_k in (5.3-18) can be written as

$$S_k = \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix} \quad (5.3-29)$$

$$\Lambda_k(x) = \begin{bmatrix} E_k(x) & 0 \\ 0 & E_k^{-1}(x) \end{bmatrix} \quad (5.3-30)$$

The k th transmission line admittance matrix H_k was already solved by Griffith and Nakhla [8] (also see section 2.3). Since the objective here is not to obtain matrix H_k , but to solve $g_k(v_k(D_k, t_0), i_k(D_k, t_0))$, a different procedure of deriving H_k is chosen.

In order to obtain an admittance form, changing the direction of $I_k(x, s)$ in (5.3-28) gives

$$\begin{bmatrix} V_k(x, s) \\ I_k(x, s) \end{bmatrix} = \begin{bmatrix} S_{kv} & S_{kv} \\ -S_{ki} & S_{ki} \end{bmatrix} \begin{bmatrix} E_k(x) & 0 \\ 0 & E_k^{-1}(x) \end{bmatrix} \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix}^{-1} \begin{bmatrix} V_k(0, s) \\ I_k(0, s) \end{bmatrix} \quad (5.3-31)$$

Rearrangement of (5.3-31) will be made in order to get the admittance form of the k th transmission line.

Premultiplying (5.3-31) by $\begin{bmatrix} S_{kv} & S_{kv} \\ -S_{ki} & S_{ki} \end{bmatrix}^{-1}$ gives

$$\begin{bmatrix} S_{kv} & S_{kv} \\ -S_{ki} & S_{ki} \end{bmatrix}^{-1} \begin{bmatrix} V_k(x, s) \\ I_k(x, s) \end{bmatrix} = \begin{bmatrix} E_k(x) & 0 \\ 0 & E_k^{-1}(x) \end{bmatrix} \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix}^{-1} \begin{bmatrix} V_k(0, s) \\ I_k(0, s) \end{bmatrix} \quad (5.3-32)$$

i.e.

$$\frac{1}{2} \begin{bmatrix} S_{kv}^{-1} & -S_{ki}^{-1} \\ S_{kv}^{-1} & S_{ki}^{-1} \end{bmatrix} \begin{bmatrix} V_k(x, s) \\ I_k(x, s) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} E_k(x) & 0 \\ 0 & E_k^{-1}(x) \end{bmatrix} \begin{bmatrix} S_{kv}^{-1} & S_{ki}^{-1} \\ S_{kv}^{-1} & -S_{ki}^{-1} \end{bmatrix} \begin{bmatrix} V_k(0, s) \\ I_k(0, s) \end{bmatrix} \quad (5.3-33)$$

Cancelling 1/2 at both sides of (5.3-33) gives

$$\begin{bmatrix} S_{kv}^{-1} & -S_{ki}^{-1} \\ S_{kv}^{-1} & S_{ki}^{-1} \end{bmatrix} \begin{bmatrix} V_k(x, s) \\ I_k(x, s) \end{bmatrix} = \begin{bmatrix} E_k(x) & 0 \\ 0 & E_k^{-1}(x) \end{bmatrix} \begin{bmatrix} S_{kv}^{-1} & S_{ki}^{-1} \\ S_{kv}^{-1} & -S_{ki}^{-1} \end{bmatrix} \begin{bmatrix} V_k(0, s) \\ I_k(0, s) \end{bmatrix} \quad (5.3-34)$$

Arrange all terms of (5.3-34) into the left hand side of the equation gives

$$\begin{bmatrix} -E_k(x) S_{ki}^{-1} & -S_{ki}^{-1} \\ E_k^{-1}(x) S_{ki}^{-1} & S_{ki}^{-1} \end{bmatrix} \begin{bmatrix} I_k(0, s) \\ I_k(x, s) \end{bmatrix} - \begin{bmatrix} E_k(x) S_{kv}^{-1} & -S_{kv}^{-1} \\ E_k^{-1}(x) S_{kv}^{-1} & -S_{kv}^{-1} \end{bmatrix} \begin{bmatrix} V_k(0, s) \\ V_k(x, s) \end{bmatrix} = 0 \quad (5.3-35)$$

Decomposing (5.3-35) results in

$$\begin{bmatrix} -E_k(x) & -U \\ E_k^{-1}(x) & U \end{bmatrix} \begin{bmatrix} S_{ki}^{-1} & 0 \\ 0 & S_{ki}^{-1} \end{bmatrix} \begin{bmatrix} I_k(0, s) \\ I_k(x, s) \end{bmatrix} - \begin{bmatrix} E_k(x) & -U \\ E_k^{-1}(x) & -U \end{bmatrix} \begin{bmatrix} S_{kv}^{-1} & 0 \\ 0 & S_{kv}^{-1} \end{bmatrix} \begin{bmatrix} V_k(0, s) \\ V_k(x, s) \end{bmatrix} = 0 \quad (5.3-36)$$

Rearranging (5.3-36) will give a final form as

$$\begin{aligned}
 \begin{bmatrix} I_k(0, s) \\ I_k(x, s) \end{bmatrix} &= \begin{bmatrix} S_{ki}^{-1} & 0 \\ 0 & S_{ki}^{-1} \end{bmatrix} \begin{bmatrix} -E_k(x) & -U \\ E_k^{-1}(x) & U \end{bmatrix}^{-1} \begin{bmatrix} E_k(x) & -U \\ E_k^{-1}(x) & -U \end{bmatrix} \begin{bmatrix} S_{kv}^{-1} & 0 \\ 0 & S_{kv}^{-1} \end{bmatrix} \begin{bmatrix} V_k(0, s) \\ V_k(x, s) \end{bmatrix} \\
 &= \begin{bmatrix} S_{ki}^{-1} & 0 \\ 0 & S_{ki}^{-1} \end{bmatrix} \begin{bmatrix} E_{k1}(x) & E_{k2}(x) \\ E_{k2}(x) & E_{k1}(x) \end{bmatrix} \begin{bmatrix} S_{kv}^{-1} & 0 \\ 0 & S_{kv}^{-1} \end{bmatrix} \begin{bmatrix} V_k(0, s) \\ V_k(x, s) \end{bmatrix} \quad (5.3-37) \\
 &= \begin{bmatrix} S_{ki} E_{k1}(x) S_{kv}^{-1} & S_{ki} E_{k2}(x) S_{kv}^{-1} \\ S_{ki} E_{k2}(x) S_{kv}^{-1} & S_{ki} E_{k1}(x) S_{kv}^{-1} \end{bmatrix} \begin{bmatrix} V_k(0, s) \\ V_k(x, s) \end{bmatrix} \\
 &= H_k(x) \begin{bmatrix} V_k(0, s) \\ V_k(x, s) \end{bmatrix}
 \end{aligned}$$

where

$$H_k(x) = \begin{bmatrix} S_{ki} E_{k1}(x) S_{kv}^{-1} & S_{ki} E_{k2}(x) S_{kv}^{-1} \\ S_{ki} E_{k2}(x) S_{kv}^{-1} & S_{ki} E_{k1}(x) S_{kv}^{-1} \end{bmatrix} \quad (5.3-38)$$

(5.3-38) is the same as (2.3-32) except for $x = D_k$ there. The importance of above derivation can be seen by solving the general state space equation (5.3-11), which is given as following.

Solution of the general equation of (5.3-11)

It is known that the solution of homogeneous form of (5.3-12) can be

$$\Phi_k(x, s) = e^{\beta_k x} \Phi_k(0, s) \quad (5.3-39)$$

Compared to (5.3-31) it can be concluded that

$$e^{\beta_k x} = \begin{bmatrix} S_{kv} & S_{kv} \\ -S_{ki} & S_{ki} \end{bmatrix} \begin{bmatrix} E_k(x) & 0 \\ 0 & E_k^{-1}(x) \end{bmatrix} \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix}^{-1} \quad (5.3-40)$$

The solution of the general state space equation (5.3-11) with changed direction of $I_k(x, s)$ is

$$\Phi_k(x, s) = e^{\beta_k x} \Phi_k(0, s) + \int_0^x e^{\beta_k(x-\tau)} b_k(\tau, t_0) d\tau \quad (5.3-41)$$

This is equivalent to

$$\begin{aligned} \begin{bmatrix} V_k(x, s) \\ I_k(x, s) \end{bmatrix} &= \begin{bmatrix} S_{kv} & S_{kv} \\ -S_{ki} & S_{ki} \end{bmatrix} \begin{bmatrix} E_k(x) & 0 \\ 0 & E_k^{-1}(x) \end{bmatrix} \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix}^{-1} \begin{bmatrix} V_k(0, s) \\ I_k(0, s) \end{bmatrix} \\ &+ \int_0^x \begin{bmatrix} S_{kv} & S_{kv} \\ -S_{ki} & S_{ki} \end{bmatrix} \begin{bmatrix} E_k(x-\tau) & 0 \\ 0 & E_k^{-1}(x-\tau) \end{bmatrix} \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix}^{-1} \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \begin{bmatrix} v_k(\tau, t_0) \\ i_k(\tau, t_0) \end{bmatrix} d\tau \end{aligned} \quad (5.3-42)$$

Following the procedures of (5.3-31)-(5.3-37), (5.3-42) can be changed into

$$\begin{aligned} \begin{bmatrix} I_k(0, s) \\ I_k(x, s) \end{bmatrix} &= \begin{bmatrix} S_{ki} E_{k1}(x) S_{kv}^{-1} & S_{ki} E_{k2}(x) S_{kv}^{-1} \\ S_{ki} E_{k2}(x) S_{kv}^{-1} & S_{ki} E_{k1}(x) S_{kv}^{-1} \end{bmatrix} \begin{bmatrix} V_k(0, s) \\ V_k(x, s) \end{bmatrix} \\ &+ \int_0^x \begin{bmatrix} S_{ki} & 0 \\ 0 & S_{ki} \end{bmatrix} \begin{bmatrix} -E_k(x) & -U \\ E_k^{-1}(x) & U \end{bmatrix}^{-1} \begin{bmatrix} E_k(x-\tau) & 0 \\ 0 & E_k^{-1}(x-\tau) \end{bmatrix} \cdot \\ &\quad \begin{bmatrix} S_{kv}^{-1} & S_{ki}^{-1} \\ S_{kv}^{-1} & -S_{ki}^{-1} \end{bmatrix} \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \begin{bmatrix} v_k(\tau, t_0) \\ i_k(\tau, t_0) \end{bmatrix} d\tau \end{aligned} \quad (5.3-43)$$

Let $x = D_k$ in (5.3-43), then

$$\begin{aligned}
\begin{bmatrix} I_k(0, s) \\ I_k(D_k, s) \end{bmatrix} &= \begin{bmatrix} S_{ki} E_{k1}(D_k) S_{kv}^{-1} & S_{ki} E_{k2}(D_k) S_{kv}^{-1} \\ S_{ki} E_{k2}(D_k) S_{kv}^{-1} & S_{ki} E_{k1}(D_k) S_{kv}^{-1} \end{bmatrix} \begin{bmatrix} V_k(0, s) \\ V_k(D_k, s) \end{bmatrix} \\
&+ \int_0^{D_k} \begin{bmatrix} S_{ki} & 0 \\ 0 & S_{ki} \end{bmatrix} \begin{bmatrix} -E_k(D_k) & -U \\ E_k^{-1}(D_k) & U \end{bmatrix}^{-1} \begin{bmatrix} E_k(D_k - \tau) & 0 \\ 0 & E_k^{-1}(D_k - \tau) \end{bmatrix} \cdot \\
&\quad \begin{bmatrix} S_{kv}^{-1} & S_{ki}^{-1} \\ S_{kv}^{-1} & -S_{ki}^{-1} \end{bmatrix} \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \begin{bmatrix} v_k(\tau, t_0) \\ i_k(\tau, t_0) \end{bmatrix} d\tau
\end{aligned} \quad (5.3-44)$$

Comparing (5.3-44) with general form (5.3-1), it can be concluded that

$$H_k(D_k) = \begin{bmatrix} S_{ki} E_{k1}(D_k) S_{kv}^{-1} & S_{ki} E_{k2}(D_k) S_{kv}^{-1} \\ S_{ki} E_{k2}(D_k) S_{kv}^{-1} & S_{ki} E_{k1}(D_k) S_{kv}^{-1} \end{bmatrix} \quad (5.3-45)$$

which is the same as that of Griffith and Nakhla's form, and the reminding part of (5.3-44) must be the contributions associated with the initial values of the k th transmission line, i.e., the negative of the reminding part is $g_k(v_k(D_k, t_0), i_k(D_k, t_0))$, which was given by (5.3-2).

It is noted that the initial conditions of the k th distributed transmission line is dependent upon the voltages and currents at all locations along the line.

5.4 Implementations

(a) Computation of $g_k(v_k(D_k, t_0), i_k(D_k, t_0))$

The Simpson's formula is used to solve the integral of (5.3-2) with cutting the length of D_k by n_{kd} , where n_{kd} is an integer and should be chosen large enough to guarantee the accuracy of the integral. Then (5.3-2) can be simplified as

$$\begin{aligned}
& g_k(v_k(D_k, t_0), i_k(D_k, t_0)) \\
&= - \begin{bmatrix} s_{ki} & 0 \\ 0 & s_{ki} \end{bmatrix} \int_0^{D_k} \begin{bmatrix} \Theta_{k11}(\tau) & \Theta_{k12}(\tau) \\ \Theta_{k21}(\tau) & \Theta_{k22}(\tau) \end{bmatrix} \begin{bmatrix} s_{kv}^{-1} & s_{ki}^{-1} \\ s_{kv}^{-1} & -s_{ki}^{-1} \end{bmatrix} \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \begin{bmatrix} v_k(\tau, t_0) \\ i_k(\tau, t_0) \end{bmatrix} d\tau \quad (5.4-1) \\
&= \begin{bmatrix} s_{ki} & 0 \\ 0 & s_{ki} \end{bmatrix} \mathfrak{R}_k(D_k)
\end{aligned}$$

where

$$\begin{bmatrix} \Theta_{k11}(\tau) & \Theta_{k12}(\tau) \\ \Theta_{k21}(\tau) & \Theta_{k22}(\tau) \end{bmatrix} = \begin{bmatrix} -E_k(D_k) & -U \\ E_k^{-1}(D_k) & U \end{bmatrix}^{-1} \begin{bmatrix} E_k(D_k - \tau) & 0 \\ 0 & E_k^{-1}(D_k - \tau) \end{bmatrix} \quad (5.4-2)$$

$$\begin{cases} \Theta_{k11}(\tau) = \text{diag} \left(\frac{e^{-\Upsilon_{ki}(D_k - \tau)}}{(e^{\Upsilon_{ki}D_k} - e^{-\Upsilon_{ki}D_k})} \right) = \text{diag} \left(\frac{e^{-\Upsilon_{ki}(2D_k - \tau)}}{(1 - e^{-2\Upsilon_{ki}D_k})} \right) \\ \Theta_{k12}(\tau) = \text{diag} \left(\frac{e^{\Upsilon_{ki}(D_k - \tau)}}{(e^{\Upsilon_{ki}D_k} - e^{-\Upsilon_{ki}D_k})} \right) = \text{diag} \left(\frac{e^{-\Upsilon_{ki}\tau}}{(1 - e^{-2\Upsilon_{ki}D_k})} \right) \\ \Theta_{k21}(\tau) = \text{diag} \left(-\frac{e^{\Upsilon_{ki}\tau}}{(e^{\Upsilon_{ki}D_k} - e^{-\Upsilon_{ki}D_k})} \right) = \text{diag} \left(-\frac{e^{-\Upsilon_{ki}(D_k - \tau)}}{(1 - e^{-2\Upsilon_{ki}D_k})} \right) \\ \Theta_{k22}(\tau) = \text{diag} \left(-\frac{e^{-\Upsilon_{ki}\tau}}{(e^{\Upsilon_{ki}D_k} - e^{-\Upsilon_{ki}D_k})} \right) = \text{diag} \left(-\frac{e^{-\Upsilon_{ki}(D_k + \tau)}}{(1 - e^{-2\Upsilon_{ki}D_k})} \right) \end{cases} \quad (5.4-3)$$

$\mathfrak{R}_k(D_k)$ is a $2m_k$ dimension vector which is the Simpson's approximation of the integral of

$$\int_0^{D_k} \begin{bmatrix} \Theta_{k11}(\tau) & \Theta_{k12}(\tau) \\ \Theta_{k21}(\tau) & \Theta_{k22}(\tau) \end{bmatrix} \begin{bmatrix} s_{kv}^{-1} & s_{ki}^{-1} \\ s_{kv}^{-1} & -s_{ki}^{-1} \end{bmatrix} \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \begin{bmatrix} v_k(\tau, t_0) \\ i_k(\tau, t_0) \end{bmatrix} d\tau \quad (5.4-4)$$

For convenience, copy the Simpson's formula here [14]

$$\int_0^{D_k} p_k(x) dx = \Delta x_k \left(\frac{17}{48} p_{k0} + \frac{59}{48} p_{k1} + \frac{43}{48} p_{k2} + \frac{49}{48} p_{k3} \right) + \Delta x_k (p_{k5} + p_{k6} + \dots + p_{kn_{kd}-1}) + \Delta x_k \left(\frac{49}{48} p_{kn_{kd}-3} + \frac{43}{48} p_{kn_{kd}-2} + \frac{59}{48} p_{kn_{kd}-1} + \frac{17}{48} p_{kn_{kd}} \right) \quad (5.4-5)$$

where $\Delta x_k = \frac{D_k}{n_{kd}}$ and

$$p_{kj} = \begin{bmatrix} \Theta_{k11}(j\Delta x_k) & \Theta_{k12}(j\Delta x_k) \\ \Theta_{k21}(j\Delta x_k) & \Theta_{k22}(j\Delta x_k) \end{bmatrix} \begin{bmatrix} s_{kv}^{-1} & s_{ki}^{-1} \\ s_{kv}^{-1} & -s_{ki}^{-1} \end{bmatrix} \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \begin{bmatrix} v_k(j\Delta x_k, t_0) \\ i_k(j\Delta x_k, t_0) \end{bmatrix} \quad (5.4-6)$$

$j=0, 1, 2, \dots, n_{kd}$.

(b) Solutions of $v_k(j\Delta x_k, t_0)$ and $i_k(j\Delta x_k, t_0)$

Replacing x by $j\Delta x_k$ in (5.3-42) without changing the sign of $I_k(x, s)$,

$$\begin{aligned} \begin{bmatrix} V_k(j\Delta x_k, s) \\ I_k(j\Delta x_k, s) \end{bmatrix} &= \begin{bmatrix} s_{kv} & s_{kv} \\ s_{ki} & -s_{ki} \end{bmatrix} \begin{bmatrix} E_k(j\Delta x_k) & 0 \\ 0 & E_k^{-1}(j\Delta x_k) \end{bmatrix} \begin{bmatrix} s_{kv} & s_{kv} \\ s_{ki} & -s_{ki} \end{bmatrix}^{-1} \begin{bmatrix} V_k(0, s) \\ I_k(0, s) \end{bmatrix} \\ &+ \int_0^{j\Delta x_k} \begin{bmatrix} s_{kv} & s_{kv} \\ s_{ki} & -s_{ki} \end{bmatrix} \begin{bmatrix} E_k(j\Delta x_k - \tau) & 0 \\ 0 & E_k^{-1}(j\Delta x_k - \tau) \end{bmatrix} \cdot \\ &\cdot \begin{bmatrix} s_{kv} & s_{kv} \\ s_{ki} & -s_{ki} \end{bmatrix}^{-1} \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \begin{bmatrix} v_k(\tau, t_{0-1}) \\ i_k(\tau, t_{0-1}) \end{bmatrix} d\tau \end{aligned} \quad (5.4-7)$$

where t_{0-1} means the previous resetting time.

In order to avoid the positive exponential, premultiply (5.4-7) by

$$\begin{bmatrix} E_k(j\Delta x_k) & 0 \\ 0 & E_k(j\Delta x_k) \end{bmatrix} \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix}^{-1}, \text{ i.e.}$$

$$\begin{aligned} & \begin{bmatrix} E_k(j\Delta x_k) & 0 \\ 0 & E_k(j\Delta x_k) \end{bmatrix} \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix}^{-1} \begin{bmatrix} V_k(j\Delta x_k, s) \\ I_k(j\Delta x_k, s) \end{bmatrix} \\ &= \begin{bmatrix} E_k(2j\Delta x_k) & 0 \\ 0 & U \end{bmatrix} \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix}^{-1} \begin{bmatrix} V_k(0, s) \\ I_k(0, s) \end{bmatrix} \\ &+ \int_0^{j\Delta x_k} \begin{bmatrix} E_k(2j\Delta x_k - \tau) & 0 \\ 0 & E_k(\tau) \end{bmatrix} \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix}^{-1} \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \begin{bmatrix} v_k(\tau, t_{0-1}) \\ i_k(\tau, t_{0-1}) \end{bmatrix} d\tau \end{aligned} \quad (5.4-8)$$

Solving (5.4-8) at each $j\Delta x_k$ by LU decomposition, the s-domain solution $\begin{bmatrix} V_k(j\Delta x_k, s) & I_k(j\Delta x_k, s) \end{bmatrix}^t$ at all $j\Delta x_k$ can be obtained. Then the Laplace inversion of its solution will give $v_k(j\Delta x_k, t_0)$ and $i_k(j\Delta x_k, t_0)$.

(c) Computation of The Integral Part of (5.4-8)

Since (5.4-8) involves another integral which is more complicated, the following procedures are taken

Let

$$\begin{aligned}
w_k(j\Delta x_k) &= \int_0^{j\Delta x_k} \begin{bmatrix} E_k(2j\Delta x_k - \tau) & 0 \\ 0 & E_k(\tau) \end{bmatrix} \cdot \\
&\cdot \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix}^{-1} \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \begin{bmatrix} v_k(\tau, t_{0-1}) \\ i_k(\tau, t_{0-1}) \end{bmatrix} d\tau \\
&= \int_0^{j\Delta x_k} h_{k1}(2j\Delta x_k - \tau, \tau) h_{k2}(\tau) d\tau
\end{aligned} \tag{5.4-9}$$

where

$$h_{k1}(2j\Delta x_k - \tau, \tau) = \begin{bmatrix} E_k(2j\Delta x_k - \tau) & 0 \\ 0 & E_k(\tau) \end{bmatrix} \tag{5.4-10}$$

$$h_{k2}(\tau) = \begin{bmatrix} S_{kv} & S_{kv} \\ S_{ki} & -S_{ki} \end{bmatrix}^{-1} \begin{bmatrix} 0 & L_k \\ C_k & 0 \end{bmatrix} \begin{bmatrix} v_k(\tau, t_{0-1}) \\ i_k(\tau, t_{0-1}) \end{bmatrix} \tag{5.4-11}$$

When $j=0$

$$w_k(0) = h_{k1}(0, 0) h_{k2}(0) \Delta x_k \tag{5.4-12}$$

When $j=1$, the *Trapezoidal* rule is used

$$w_k(\Delta x_k) = \frac{h_{k1}(2\Delta x_k, 0) h_{k2}(0) + h_{k1}(\Delta x_k, \Delta x_k) h_{k2}(\Delta x_k)}{2} \Delta x_k \tag{5.4-13}$$

When $2 \leq j \leq 7$, the *Extended Trapezoidal* rule is used

$$\begin{aligned}
w_k(j\Delta x_k) &= \frac{h_{k1}(2j\Delta x_k, 0) h_{k2}(0) + h_{k1}(j\Delta x_k, j\Delta x_k) h_{k2}(j\Delta x_k)}{2} \Delta x_k \\
&\quad + (h_{k1}((2j-1)\Delta x_k, \Delta x_k) h_{k2}(\Delta x_k) + \dots) \Delta x_k \\
&\quad + h_{k1}((j+1)\Delta x_k, (j-1)\Delta x_k) h_{k2}((j-1)\Delta x_k) \Delta x_k
\end{aligned} \tag{5.4-14}$$

When $j > 7$, the *Simpson* form of (5.4-5) is used.

(d) CPU Time Considerations

Considering the general form of (5.3-1), its CPU cost is almost the same as that of NILT0 when the term of $g_k(v_k(D_k, t_0), i_k(D_k, t_0))$ is not present. The implementation of $g_k(v_k(D_k, t_0), i_k(D_k, t_0))$ is complicated and time consuming. One has to consider the number of poles used, the number of coupled transmission lines, the dimension of each coupled transmission line and the number of steps used to solve the integrals of (5.4-1) and (5.4-9). Under each step one also needs to consider the number of poles and the dimensions of transmission lines.

The most efficient way to compute time responses using the numerical inversion of Laplace transform with the stepping algorithm is, first, to calculate $g_k(v_k(D_k, t_0), i_k(D_k, t_0))$ before going into the resetting, since the most information can be used for calculating it at $t = t_0$; and second, within the processing of resetting, it is better to prepare as much information as possible for the integrals and then to solve it. Thus h_{k1} and h_{k2} are calculated and then stored before computing w_k . Certainly, storing such matrices takes more memory space.

(e) Simulation steps

The simulation steps is shown in Fig 5.1.

(f) Accuracy

To save CPU time, h_{k1} and h_{k2} are calculated and stored in advance at fixed

distance interval. So this may cause inaccuracy of computing the integral (5.4-9) when j is small. Adding more points between 0 and j can improve the accuracy, but this will result in more complicated programming and more CPU time costs.

5.5. Examples

To demonstrate the stepping algorithm for the numerical inversion of Laplace transform several examples are given in this section. All examples in this chapter are calculated with the unit step input.

Example 5.1. Single lossless transmission line circuit.

Consider the circuit shown in Fig. 5.2. To show that this method improves accuracy with increasing time, comparisons of results of Hspice, NILT0 and the resetting technique are shown in Fig. 5.3 and Fig. 5.4, respectively. Both NILT0 and the new method are used with $M=4$ and $N=2$. Fig. 5.4 shows that the result of two step resettings is very close to the that of Hspice. In Fig. 5.3 the resetting starts at $t_0 = 2.7ns$. In Fig. 5.4 two step resettings are taken at $t_{01} = 2.7ns$ and $t_{02} = 6.3ns$, respectively. Scales of vertical axes in Fig. 5.3 and Fig. 5.4 are 1volt/unit.

Fig. 5.4 Comparison of Hspice, NILT0 and two step resettings for the circuit of Fig. 5.2

Example 5.2. Seven single lossless transmission line circuit.

The circuit is given in Fig. 4.3. Comparisons of Hspice, NILT0 and the new method are shown in Fig. 5.5 and Fig. 5.6, respectively. Both NILT0 and the new method are used with $M=4$ and $N=2$. In Fig. 5.5 one step resetting is taken at $t_0 = 0.84ns$. In Fig. 5.6 two step resettings are taken at $t_{01} = 0.84ns$ and $t_{02} = 1.3ns$, respectively. Scales of vertical axes in Fig. 5.5 and Fig. 5.6 are 1volt/unit.

Example 5.3 One lossless multiconductor transmission line circuit.

The circuit is given by Fig. 4.12 and all parameters are given in example 4.4. Since Hspice can not be used for a coupled transmission line circuit, comparison of NILT2, NILT0 and the new method are shown in Fig. 5.7 and Fig. 5.8, respectively. In Fig. 5.7 one step resetting is used at $t_0 = 1.5\text{ns}$ and in Fig. 5.8 two step resettings are taken at $t_{01} = 1.5\text{ns}$ and $t_{02} = 3.2\text{ns}$, respectively. All methods in this example are used with $M=6$ and $N=4$. Scales of vertical axes in those figures are 1 volt/unit.

Fig. 5.8 Responses of $v(1)$ by different methods for the circuit of Fig. 4.12 (two step reinitializations)

Example 5.4. Two multiconductor transmission line circuit.

The circuit is given in Fig. 4.21 and all parameters are given in example 4.6. Comparisons of NILT2, NILT0 and the new method are shown in Fig. 5.9 and Fig. 5.10, respectively. In Fig. 5.9 one step resetting is used at $t_0 = 0.45\text{ ns}$ and in Fig. 5.10 two step resettings are taken at $t_{01} = 0.45\text{ ns}$ and $t_{02} = 1.1\text{ ns}$, respectively. All methods are used with $M=6$ and $N=4$. Scales of vertical axes in those figures are 1volt/unit.

Example 4.5 Three multiconductor transmission line circuit.

The circuit is given in Fig. 4.26 and all parameters are given in example 4.7. Comparisons of NILT2, NILT0 and the new method are shown in Fig. 5.11, Fig. 5.12 and Fig. 5.13, respectively. In 5.11 one step resetting is used at $t_0 = 1.1\text{ns}$, in Fig. 5.12 two step resettings are taken at $t_{01} = 1.1\text{ns}$ and $t_{02} = 3.3\text{ns}$ respectively and in Fig. 5.13 three step resettings are taken at $t_{01} = 1.1\text{ns}$, $t_{02} = 3.3\text{ns}$ and $t_{03} = 5\text{ns}$, respectively. All methods are used with $M=6$ and $N=4$. Scales of vertical axes in those figures are 1volt/unit.

Fig. 5.11 Responses of $v(4)$ in the circuit of Fig. 4.26 (One step resetting)

5.6 Comments on The Stepping Algorithm

The resetting technique for solving distributed networks is more accurate for calculating long transients. The method bridges the gap between NILT and the integration approach for distributed networks. Theoretically, the stepping algorithm can get an accurate solution no matter how far t goes. Combined with high degree numerical inversion of Laplace transforms, the stepping algorithm can be even more powerful. The disadvantage of this method is that more points should be added for solving the integrals.

CHAPTER 6

Conclusions and Further Research

6.1 Conclusions

The numerical inversion of Laplace transform (NILT0) is efficient for analyzing the time responses of linear distributed networks. The method is equivalent to very high order integration method and is absolutely stable. Using derivatives of output waveforms with respect to the Laplace operator s , the high degree numerical inversion of Laplace transform (NILT n , $n > 0$) decreases the errors, caused by truncation errors of the Pade approximation, of results significantly. The new method is also more efficient since, in order to obtain the same accuracy, the methods using less poles and high degree derivatives require less CPU time than that of more poles without any derivative. The NILT n method has also been proved to be absolutely stable.

The stepping algorithm based on the numerical inversion of Laplace transform uses *the resetting* technique to improve the accuracy of solutions of linear distributed networks. The stepping algorithm bridges the gap between NILT and integration approaches. The stepping algorithm is also absolutely stable because at each time interval

NILT is used directly.

6.2 Further Research

Further research might take the following directions:

(i) Combining with the piecewise decomposition technique [9][10], the methods proposed in this thesis can also be extended to solve the coupled transmission line networks containing *nonlinear* terminations.

(ii) The proposed methods can be extended to solve non-uniform transmission line networks.

(iii) Combined with NILT_n, the stepping algorithm can obtain more accurate results than that of NILT_n itself with using the same number of poles of Pade approximation.

References

- [1] K. Adamiak, R. Allen, J. Poltz, C. Rebiaznt and A. Wexler, "System simulation of printed circuit boards including packages and connectors," *27th ACM/IEEE Design Automation Conference*, pp.413 - 418, 1990.
- [2] "ME28," *Harwell Subroutine Library*, April 1980.
- [3] A. R. Djordjevic and T. K. Sarkar, "Analysis of time response of lossy multiconductor transmission line networks," *IEEE Trans. on Microwave Theory and Techniques*, vol. MTT-35, pp. 898-908, October 1987.
- [4] A. R. Djordjevic, T. K. Sarkar and R. F. Harrington, "Time-domain response of multiconductor transmission lines," *Proceedings of IEEE*, vol. 75, pp. 743-764, June 1987.
- [5] D. F. Elliott and K. R. Rao, "Fast transforms, algorithm, analysis, applications," *Academic Press*, 1982.
- [6] B. Friedland, "Control system design," *McGraw-Hill*, 1986.
- [7] J. L. Morris, "Computational method in elementary numerical Analysis," *Wiley*, 1983.
- [8] R. Griffith and M. Nakhla, "Time-domain analysis of lossy coupled transmission lines," *IEEE Trans. on Microwave Theory and Tech.*, vol. MTT-38, pp. 1480-1487, October 1990.
- [9] R. Griffith and M. Nakhla, "Mixed frequency/time domain analysis of nonlinear circuits," submitted to *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*.
- [10] L. Lu, M. Nakhla and Q. J. Zhang, "A stepping algorithm for transient analysis of distributed networks," submitted to *IEEE ISCAS' 1992*.
- [11] S. Lum, M. Nakhla and Q. J. Zhang, "Sensitivity analysis of lossy coupled

- transmission lines," accepted for publication of *IEEE Trans. on Microwave Theory and Techniques*.
- [12] S. Lum, "Sensitivity analysis and optimization of high-speed VLSI interconnects," *M. Eng. Thesis, Department of Electronics, Carleton University*, 1991.
- [13] M. S. Nakhla, "Analysis of pulse propagation on high-speed VLSI chips," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 490-494, April 1990.
- [14] M. Nakhla and L. Lu, "Time-domain analysis of distributed networks," *Proceedings of IEEE Int. Symp. Circuits Syst.*, pp. 888-891, June 1991.
- [15] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352-366, April 1990.
- [16] L. T. Pillage, C. M. Wolff and R. A. Rohrer, "Dominant pole/zero analysis for analog circuit design," *IEEE 1989 Custom Integrated Circuits Conference*.
- [17] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, "Numerical recipes," *Cambridge University Press*, 1989.
- [18] J. E. Schutt-Aine and R. Mittra, "Analysis of pulse propagation in coupled transmission lines," *IEEE Trans. on Circuits and Systems*, vol. CAS-32, pp. 1214 - 1219, December 1985.
- [19] J. E. Schutt-Aine and R. Mittra, "Scattering parameter transient analysis of transmission lines loaded with nonlinear terminations," *IEEE Trans. on Microwave Theory and Techniques*, vol. 36, pp. 529, March 1988.
- [20] K. Singhal and J. Vlach, "Computation of time-domain response by numerical inversion of Laplace transform," *Journal of the Franklin Institute*, vol. 299, pp. 109-126, Feb. 1975.
- [21] K. Singhal, J. Vlach and M. Nakhla, "Absolutely stable, high order method for time domain solution of networks," *Archiv fur Elektronik und Uebertragungstechnik*, vol. 30, pp. 157-166, 1976.
- [22] T. K. Tang and M. S. Nakhla, "Analysis of high-speed VLSI interconnects using the asymptotic waveform evaluation technique," *Proceedings of IEEE International Conference on Computer-Aided Design (ICCAD)*, pp. 542-545, November 1990.

-
- [23] T. K. Tang and M. S. Nakhla, "Analysis of high-speed VLSI interconnects," accepted for publication in *IEEE Trans. on Computer-Aided Design*.
 - [24] T. K. Tang, "Analysis of interconnection delay and crosstalk in high-speed VLSI systems," *M. Eng. Thesis, Department of Electronics, Carleton University*, 1990.
 - [25] T. K. Tang, M. Nakhla and R. Griffith, "Analysis of lossy multiconductor transmission lines using the asymptotic waveform evaluation technique," accepted for publication by *IEEE Trans. on Microwave Theory and Techniques*.
 - [26] J. Vlach and K. Singhal, "Computer methods for circuit analysis and design," Van Nostrand Reinhold, 1983.
 - [27] D. H. Xie and M. Nakhla, "Delay and crosstalk simulation of high-speed VLSI interconnects with nonlinear terminations," *International Conference on Computer Aided Design (ICCAD)*, NOV. 1991.
 - [28] Mouna Nakhla, "On the numerical inversion of Laplace transform," *University of Waterloo*, 1974

END

28.05.92

FIN