

Removing Redundancy in Interconnect Simulation using Domain Decomposition Techniques

by

Velnamby Ambalavanar

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

Ottawa-Carleton Institute for Electrical Engineering
Department of Electronics
Faculty of Engineering
Carleton University
Ottawa, Canada
May 2008

© Velnamby Ambalavanar, 2008



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-40630-4
Our file Notre référence
ISBN: 978-0-494-40630-4

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

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

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

Abstract

A new, accurate and efficient coupled interconnect simulation algorithm is presented in this Thesis. This algorithm eliminates redundancies present in simulation of interconnect networks by exploiting the repetitive nature of segments in the conventional method with the aid of domain decomposition techniques. The proposed algorithm does not perform any approximations and hence, it provides exact solution of conventional lumped segmentation method, but with less computational effort. This algorithm was tested on practical interconnect networks and a considerable speed-up was obtained.

Acknowledgments

I would like to express my deepest gratitude and appreciation to my thesis supervisor, Prof. Pavan Gunupudi, for his guidance, helpful suggestions, encouragement and support. His ever friendly manner, motivation in difficult moments, patience, and good judgement made the journey through this thesis an enjoyable and unforgettable experience.

I would like to thank Prof. Ram Achar and Prof. Michel Nakhla for their expert advises on signal integrity and high-speed interconnect simulation.

I would also like to thank my colleagues: A. Jerome for research collaboration and helpful materials, B. Nouri for important helps and useful discussions; A. Charest for providing LATEX templates for my thesis; Dr. C. Chen, Dr. D. Saraswat, Y. Xu, Y. Zhou, A. Sridhar, and H. Dhindsa for all their help and making my time in the lab an enjoyable experience. I am also grateful to E. Ghias-Begloo for his contributions during his internship program.

I also extend my thanks to all staff of the department of electronics, especially, S. Bruce, P. Piccolo, and B. Power, for their support.

Finally, I would like to thank my parents for their strong support and encouragement throughout my studies.

Contents

1	Introduction	1
1.1	Background and Motivation	1
1.2	Contributions	4
1.3	Organization of the Thesis	5
2	Simulation of High-Speed Interconnect Networks	6
2.1	Overview of High Speed Interconnect Issues	6
2.2	High-speed Interconnect Effects	8
2.2.1	Propagation Delay	8
2.2.2	Reflection	9
2.2.3	Crosstalk	9
2.2.4	Attenuation	10
2.3	High-Speed Interconnect Models	10
2.4	Distributed Transmission Line Equations	12
2.5	General Formulation of Network Equations	16
2.6	Conventional Lumped Segmentation	19
2.6.1	Simulation of Interconnects Discretized with Lumped Segments	21

2.7	Efficient Macromodeling Approaches	23
2.7.1	Method of Characteristics	24
2.7.2	Model-Reduction Techniques	26
3	Proposed Interconnect Simulation Method	28
3.1	Circuit Partitioning	29
3.2	Simulation of Discretized Interconnects using Domain Decomposition	31
3.3	Analysis of Transient Simulation with Domain Decomposition	34
3.4	Removing Redundancy in Simulation of Interconnect Networks	38
3.5	Numerical Results	40
3.5.1	Example 1	40
3.5.2	Example 2	41
3.5.3	Example 3	45
4	Implementation Techniques for the Proposed Method	54
4.1	Efficient Computation of Schur Complement Matrix	55
4.1.1	Avoiding Computation of Matrices $\mathbf{Q}_i \mathbf{M}_2^{-1} \mathbf{P}_i$	55
4.1.2	Computation of Non-Zero Block Matrices Found in \mathbf{S}	58
4.2	Implementation Procedure of the Proposed Method	61
5	Estimation of Optimal Number of Domains	66
5.1	Estimation of Optimal Domains for Variable Step Size	67
5.1.1	Results	72
5.2	Estimation of Optimal Domains for Constant Step Size	75
5.2.1	Results	76

6	Conclusions and Future Work	79
6.1	Summary	79
6.2	Future Work	80

List of Figures

1.1	Interconnect problem	2
2.1	Frequency response of a trapezoidal pulse	7
2.2	Single transmission line system	13
2.3	A multi-conductor transmission line system	15
2.4	Lumped Segmentation	20
3.1	Longitudinal partitioning of an interconnect	29
3.2	Longitudinal partitioning and coupling between each discretized segment	30
3.3	Example of classification of MNA variables according to domain decomposition theory	32
3.4	Sparsity pattern of \mathbf{M} with natural node ordering.	33
3.5	Sparsity pattern of \mathbf{M} with domain decomposed ordering (6 domains).	34
3.6	Two coupled interconnect network for Example 1	41
3.7	Time-domain response of interconnect network in Example 1 with 120 lumped segments	42
3.8	Time-domain response of interconnect network in Example 1 with 1200 lumped segments	43

3.9	Nine coupled interconnect network from [64] for Example 2	44
3.10	Time-domain response of Example 2 ($d = 5cm$) for far-end of signal line and V_{out}	46
3.11	Time-domain response of Example 2 ($d = 5cm$) for near-end of first victim line and far-end of last line	47
3.12	Time-domain response of Example 2 ($d = 15cm$) for far-end of signal line and V_{out}	48
3.13	Time-domain response of Example 2 ($d = 15cm$) for near-end of first victim line and far-end of last line	49
3.14	Interconnect network comprised two and nine transmission lines for Example 3	50
3.15	Time-domain response for Example 3 (V_1 and V_{out})	51
3.16	Time-domain response for Example 3 (V_2 and V_3)	52
4.1	Node numbering approach	56
4.2	Sparsity pattern of $\mathbf{Q}_i\mathbf{M}_2^{-1}\mathbf{P}_i$ matrices ($i = 2, 5, 9$)	58
4.3	Interaction relationship for set of interface nodes lying on any given boundary	59
4.4	Sparsity pattern of \mathbf{M}_z matrix	59
4.5	Computed and shifted non-zeros of \mathbf{S}	61
5.1	Effects of number of domains in the CPU time for DD operations	67
5.2	Estimated and actual speed-up for two transmission line for the vari- able step size case	73

5.3	Estimated and actual speed-up for 10 transmission line for the variable step size case	73
5.4	Estimated and actual speed-up for 25 transmission line for the variable step size case	74
5.5	Estimated speed-up for two transmission line at six different number of lumped segments (K) for the variable step size case	74
5.6	Estimated speed-up for two transmission line at four different I values for the variable step size case	75
5.7	Estimated and actual speed-up for two transmission line for the constant step size case	77
5.8	Estimated and actual speed-up for 10 transmission line for the constant step size case	77
5.9	Estimated and actual speed-up for 25 transmission line for the constant step size case	78
5.10	Estimated speed-up for two transmission line at six different number of lumped segments (K) for the constant step size case	78

List of Tables

3.1	Computational results for Example 1 using variable step-size	41
3.2	Computational results for Example 1 using constant step-size	44
3.3	Computational results for Example 2 using variable step-size	45
3.4	Computational results for Example 2 using constant step-size	45
3.5	Computational results for Example 3	53
5.1	CPU time coefficients of large sparse matrix	69
5.2	CPU time coefficients of Schur Complement matrix	70
5.3	Comparison of optimal domains	72
5.4	Comparison of optimal domains obtained for constant step size	76

List of Acronyms

AWE	Asymptotic Waveform Evaluation
CFH	Complex Frequency Hopping
CPU	Central Processing Unit
FB	Forward-Backward
LU	Lower-Upper
MNA	Modified Modal Analysis
MTL	Multi-conductor Transmission Line
PRIMA	Passive Reduced-order Interconnect Macromodeling Algorithm
p.u.l	Per Unit Length
SPICE	Simulation Program with Integrated Circuits Emphasis
TEM	Transverse Electric and Magnetic
VLSI	Very Large Scale Integration

List of Symbols

\mathbf{A}	Matrix A
\mathbf{A}^T	Matrix A transposed
$\mathbf{0}$	Matrix of zeros
\mathfrak{R}^N	The set of all real vectors of size N
$\mathfrak{R}^{N \times N}$	The set of all real matrices of size $N \times N$
$\tilde{\mathbf{R}}$	p.u.l resistance matrix
$\tilde{\mathbf{L}}$	p.u.l inductance matrix
$\tilde{\mathbf{G}}$	p.u.l conductance matrix
$\tilde{\mathbf{C}}$	p.u.l capacitance matrix
\mathbf{x}_ϕ	Vector of unknowns in MNA formulation for circuit ϕ
\mathbf{C}_ϕ	MNA matrix representing memory elements for circuit ϕ
\mathbf{G}_ϕ	MNA matrix representing memoryless elements for circuit ϕ
$\mathbf{b}_\phi(t)$	Vector of independent sources for circuit ϕ
$\mathbf{F}_\phi(\mathbf{x}_\phi)$	Vector of non-linear functions for circuit ϕ
n_ϕ	Number of variables in circuit ϕ
n_π	Number of ports in the interconnect subnetwork π
N_t	Total number of distributed coupled interconnect sets

K	Number of lumped segments
\mathbf{x}	Vector of unknowns in the discretized network
\mathbf{C}	MNA matrix for memory elements in the discretized network
\mathbf{G}	MNA matrix for memoryless elements in the discretized network
$\mathbf{b}(t)$	Vector of independent sources for the discretized network
$\mathbf{F}(\mathbf{x})$	Vector of non-linear functions for the discretized network
n_l	Total number of variables in the discretized network
λ	Constant depending on chosen numerical integration formula
h	Time step size used for numerical integration
\mathbf{M}	Jacobian matrix for the discretized interconnect network
\mathbf{M}_i	Matrix representing i -th domain interior variables
\mathbf{P}_i	Matrix representing coupling from interface variables to $i - th$ domain
\mathbf{Q}_i	Matrix representing coupling from $i - th$ domain variables to the interface variables
\mathbf{M}_z	Matrix representing coupling between interface variables
$\Delta \mathbf{x}_i$	Vector of interior variables for i -th domain
$\Delta \mathbf{x}_{n+1}$	Vector of interface variables
ϕ_i	RHS vector for set of equations representing i -th domain
n_σ	Number of interface variables
\mathbf{S}	Schur complement matrix

N	Number of coupled lines in a MTL
I	Average number of Newton iterations per time point
ν	Number of intermediate domains
ζ, α	CPU coefficients for LU decomposition of a sparse matrix
ξ, β	CPU coefficients for FB substitution of a sparse matrix
$\zeta_\sigma, \alpha_\sigma$	CPU coefficients for LU decomposition of the Schur complement matrix
ξ_σ, β_σ	CPU coefficients for FB substitution of the Schur complement matrix
n_s	Matrix size of any intermediate sub-domain
$g(\nu)$	Estimated speed-up with proposed method when compared to traditional method
ν_o	Feasible optimal number of domains

Chapter 1

Introduction

1.1 Background and Motivation

Advances in VLSI technology toward more complex designs, higher operating frequencies, sharper rise times, smaller device sizes, low power consumption, and increased integration of analog circuits with digital blocks has made signal integrity analysis a challenging task. The ever increasing demand for high-speed applications has made high frequency effects in transmission line (See Fig. 1.1) such as delay, reflections, crosstalk, and attenuation previously neglected during circuit analysis, to become prominent [1]. Therefore, if these effects are not predicted at early design stages, they can severely degrade system performance. Generally, simulation of interconnect networks involves solving large systems of equations and therefore, the cost of simulation becomes prohibitively CPU expensive.

In high frequency applications, length of interconnects becomes a significant portion of the minimum wavelength of interest. This gives rise to signal distorting effects that

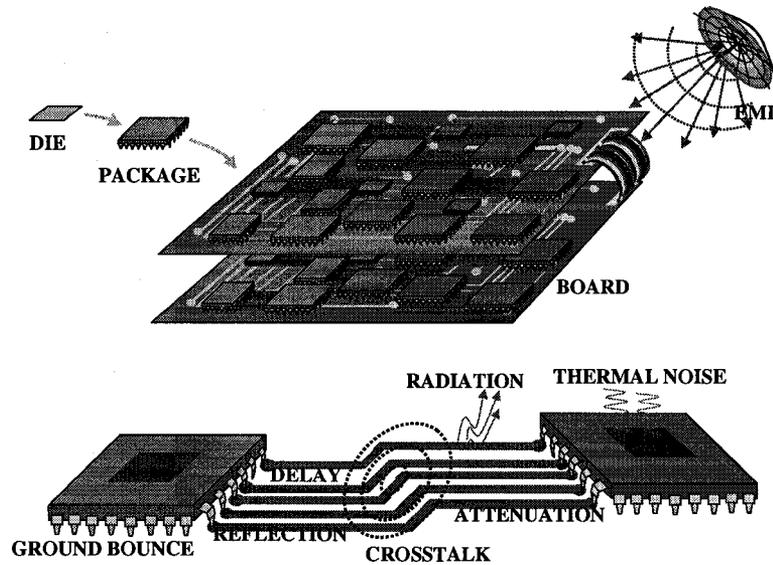


Figure 1.1: Interconnect problem

do not exist at lower frequencies. Therefore, an interconnect can no longer be treated as a short-circuit at higher frequencies; rather, it exhibits distributed impedance effects. As a result, the interconnect electrically isolates the driver from the receivers and therefore, the receivers can no longer be considered as direct loads of the driver. The impedance of the interconnect becomes the load for the driver and also the input impedance to the receivers. This causes the transmission line effects such as reflections, overshoot, undershoot, and crosstalk.

If high speed interconnect effects are neglected at early design stages, they can cause logic glitches that render a fabricated digital circuit inoperable. Extra iterations in design cycle are costly and delays the time-to-market. Therefore, accurate simulation of high-speed interconnect effects becomes extremely important.

Interconnect behavior is accurately captured in frequency domain using Telegra-

pher's equations. These equations however do not have a direct representation in the time domain. On the other hand, nonlinear components such as drivers and receivers present in the network can only be represented in the time domain. As a result, accurate interconnect simulation leads to a mixed frequency-time problem [1]. In order to circumvent this issue, interconnects are often discretized using segmentation techniques [2] such that they can be represented in time-domain and simulated along with other nonlinear components. However, this leads to large systems of equations the simulation of which is CPU expensive [2–5].

In literature, several interconnect simulation algorithms [1–11] have been proposed in order to address the above-mentioned issue. Some of these methods [6–8] have proposed alternate ways of discretizing the interconnect network to bring down the computation cost by trading-off on the accuracy of the interconnect network. Other methods involve model-order-reduction methods [9–11], that mathematically project the interconnect network into a reduced subspace to obtain a reduced-macromodel. The simulation of the macromodel so obtained is usually much faster than the original network. However, it has to be noted that the response obtained from the macromodel is an approximation of the original network. In addition, model-order-reduction methods suffer from a degradation in efficiency with an increase in the number of ports [12].

In this thesis, a technique is proposed to achieve considerable speed-up in the simulation of interconnect networks without sacrificing the accuracy of the system of equations obtained through discretization techniques. This is achieved by exploiting the repetitive nature of segments present in the discretized interconnect equations through application of longitudinal partitioning in conjunction with domain decom-

position techniques [13–17]. This results in a considerable speed up in the simulation of coupled interconnect networks with no loss in numerical accuracy.

1.2 Contributions

The work performed in this Thesis contributes to the knowledge of high-speed interconnect simulation by providing an efficient technique to achieve the exact solution of conventional lumped segmentation method [18]. Specifically, the proposed method addresses the main problem with the conventional method, which is high CPU cost due to solving large linear system of equations. The main contributions of this Thesis are as follows.

1. A new partitioning approach for multiconductor transmission lines is provided such that it forms identical partitions along the direction of signal propagation, while minimizing coupling between formed partitions.
2. An efficient technique for the simulation of high-speed interconnects is proposed which eliminates redundant partitions in the simulation process with the aid of domain decomposition methods and gives considerable speed-up compared to the conventional method.
3. Detailed implementation techniques are provided in order to perform efficient nonlinear transient simulation algorithm using proposed method.
4. An approach to estimate optimal number of identical partitions (domains) that need to be formed in order to achieve maximum possible speed-up for the proposed method is presented.

1.3 Organization of the Thesis

This thesis is organized as follows. Chapter 2 presents a background about simulation of high-speed interconnects. In this chapter, a review of high-speed interconnect issues, formulation of distributed transmission line equations, macromodeling approaches, and formulation of network equations is provided. In Chapter 3, a new efficient algorithm to remove redundancy in the simulation of interconnect networks is presented with numerical examples. Chapter 4 presents detailed implementation techniques for the proposed algorithm. In Chapter 5, an approach to estimate the optimal number of domains in order to achieve maximum possible speed-up is discussed. Finally, the conclusions and future work are presented in Chapter 6.

Chapter 2

Simulation of High-Speed Interconnect Networks

As discussed in the introduction, interconnect effects become critical in high-speed electrical systems. Predicting these effects accurately is essential for avoiding additional cost in manufacturing and delays in the design cycle. This chapter reviews some of the important aspects of high-speed interconnects along with their simulation issues.

2.1 Overview of High Speed Interconnect Issues

One of the main aspects of the high-speed interconnects is that the time taken by the propagating signal between its end points cannot be neglected. The length of the interconnect is the primary factor influencing the delay of the interconnect. Signal propagation time is directly proportional to the length of the interconnect. Signal integrity problems start to arise when the propagation time becomes close enough

to the rise/fall time of the propagating signal. As the signal makes rapid transitions between its high and low voltage levels, impedance of the interconnect becomes load for the driver and input impedance to the receivers. This leads to various transmission line effects such as delay, reflections, attenuation, and crosstalk. In high-speed

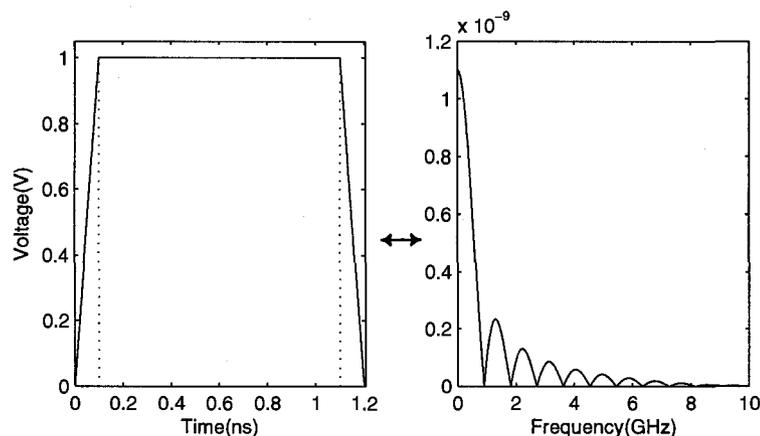


Figure 2.1: Frequency response of a trapezoidal pulse

circuits, sharper rise/fall times add high frequency content to the propagating signal. Fig. 2.1 illustrates the frequency spectrum of a trapezoidal pulse with 0.1 ns rise/fall time. It can be observed from the graph that the signal energy is spread up to several GHz.

At low frequencies, an interconnect behaves effectively as a short circuit. However, at high frequencies, the same interconnect exhibits inductive/capacitive effects. An interconnect is called *electrically short* [1] if, at the highest operating frequency of interest, the interconnect length is physically shorter than approximately one-tenth of the wavelength. Otherwise, the interconnect is referred to as *electrically long* [2,3]. Electrically short interconnects can be represented by lumped models whereas electrically long interconnects need distributed or full-wave models.

As it can be seen from Fig. 2.1, the energy spectrum of a trapezoidal pulse is spread over an infinite frequency range. However, most of the signal energy is concentrated near the low frequency region and decreases rapidly with the increase in frequency. Therefore, a maximum frequency f_{max} can be set such that neglecting the spectrum above f_{max} will not significantly alter the overall signal shape. For practical purpose, relationship between the maximum frequency of interest (f_{max}) and the rise/fall time of the signal (t_r) can be expressed as [19–22],

$$f_{max} \approx \frac{0.35}{t_r}. \quad (2.1)$$

In some cases, the limit can be more conservatively set as $f_{max} \approx 1/t_r$ [22].

2.2 High-speed Interconnect Effects

2.2.1 Propagation Delay

Propagation delay (T_d) is the time taken by the signal to travel from one end of a transmission line to the other. It depends on the physical properties of the transmission line: length, cross section, and relative permittivity and permeability of the dielectric material used. For a uniform transmission line immersed in a homogeneous medium with pure TEM mode propagation, the pure delay can be given by,

$$T_d = d\sqrt{\mu\epsilon} = d\sqrt{LC} \quad (2.2)$$

where,

- d : length of transmission line
- μ, ϵ : permeability and permittivity of the surrounding medium
- L, C : per-unit-length inductance and capacitance of the transmission line.

In addition to the pure delay, there will be an additional delay in rise time at the receiver end called rise-time degradation where the rise time at the receiver end (t_R) is larger than the rise time at the source end (t_r) [19, 23].

2.2.2 Reflection

Reflection in interconnects occurs due to discontinuity in characteristic impedance in the signal path. Characteristic impedance changes when the signal traverses through interconnects existing at various levels of design hierarchy: on-chip, packaging structures, multi-chip modules, printed circuit boards, back-planes and cables. Impedance mismatch between characteristic impedance of the transmission line and source/terminating impedances also causes reflection. Signal reflection affects the smoothness of the signal by causing ringing, overshoots, and undershoots. These effects can severely distort signal propagation at higher frequencies.

2.2.3 Crosstalk

Crosstalk refers to the coupling of electromagnetic energy into adjacent conductors through mutual inductance and mutual capacitance. The coupled energy causes noise voltage-currents in the victim line. This may lead to unnecessary switching and system malfunctioning. Crosstalk becomes a major problem in the current technology due to the high-density and closely laid interconnects.

2.2.4 Attenuation

Attenuation in interconnects occurs due to the resistance of the interconnect (ohmic losses) and dielectric losses (conductance losses). Attenuation causes reduction in signal strength and therefore, if the attenuation is very high, the propagating signal may not retain the specified logic levels at the receiver end.

The resistance of interconnects is increasing due to increased scaling of wire cross-sections in today's VLSI industry. At higher frequencies, uneven current distributions in the interconnects characterized as skin, edge, and proximity effects add more ohmic losses [2, 24–26]. The skin effect refers to the tendency of the current to concentrate near the surface of the conductor; edge effect refers to the tendency of the current to concentrate near the sharp edges of the conductor; and proximity effect refers to the tendency of the returning current through ground plane to concentrate close to the signal conductors.

Dielectric losses occur due to the leakage current through imperfect dielectric material. The dielectric loss is proportional to the loss tangent of the given dielectric material.

2.3 High-Speed Interconnect Models

In order to accurately predict the aforementioned high-speed interconnect effects, a complete description of the dynamics of the electromagnetic field guided through the interconnects becomes necessary. If the interconnect's largest dimension is electrically small when compared to the minimum wavelength of interest, the electromagnetic effects can be lumped into circuit elements and the resulting network can be simulated

using lumped circuit theory [2].

When the length of the interconnect becomes electrically long, so called *transmission-line formulation* is used to represent the line as a distributed parameter structure along the structure axis [2]. In order to use the transmission-line formulation, the cross-sectional dimensions must be electrically small. The fundamental assumption of the transmission-line formulation is that the field structure surrounding the conductors obeys a Transverse Electromagnetic (TEM) mode [2]. In TEM mode, the electric and magnetic fields are perpendicular to the direction of propagation, but there is no field component in the direction of propagation. Electrically large cross-sectional dimensions and inhomogeneities in practical wiring configuration give rise to other higher-order modes of propagation [1, 2, 27]. In reality, the electromagnetic fields in multiconductor guiding structures are never exactly of the TEM type [28]. However, for parallel lines wherein the cross-sectional dimensions are much less than the wavelength, solution of the transmission-line formulation gives significant contribution to the overall fields and resulting terminal voltages and currents [2, 28]. This is referred to as the *quasi-TEM approximation* [2].

Another important and inherent assumption in the transmission-line formulation is that the sum of the line currents at any cross section (transverse section) of the line is zero [2]. The only currents considered in the transmission-line formulation are called *differential mode* currents [28]. Differential mode currents are equal in magnitude at the same cross-section and are oppositely directed. The other currents, which are equal in magnitude at the same cross-section but have same directions, are called *common mode* currents [2, 28, 29]. Common mode currents cannot be predicted by transmission-line formulation.

The transmission line equations based on quasi-TEM approximation are characterized by distributed R , L , C , G per unit length (p.u.l) parameters [2] and discussed in detail in Sec. 2.4.

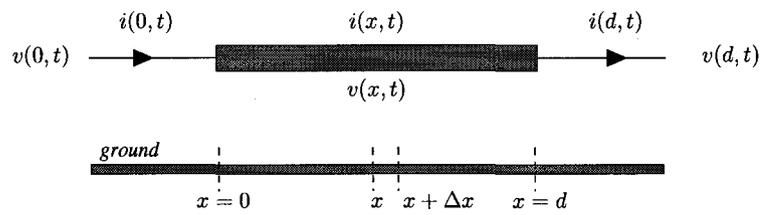
A complete solution for the multiconductor structures which have non-TEM modes can be obtained with so-called full-wave solution of Maxwell equations. However performing full-wave solutions is extremely CPU expensive. Models such as partial element equivalent circuit (PEEC) models [30–33] have been developed for these cases. However, since they result in very large lumped equivalent circuits, they are also CPU intensive.

2.4 Distributed Transmission Line Equations

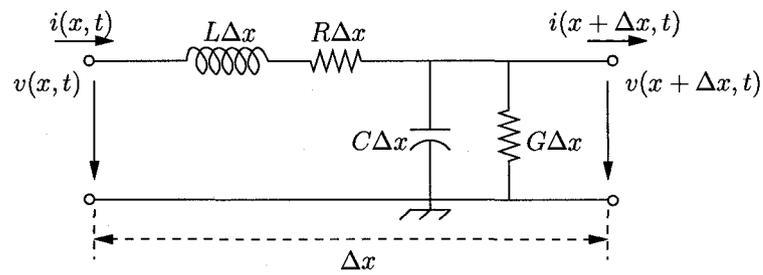
Transmission line characteristics are in general described by Telegrapher's equations. Consider the transmission line system shown in Fig. 2.2(a). Telegrapher's equations for such a structure can be derived by discretizing the line into infinitesimal sections (see Fig. 2.2(b)) of length Δx , assuming uniform p.u.l. parameters of resistance (R), inductance (L), conductance (G), and capacitance (C). Each section then includes a resistance $R\Delta x$, inductance $L\Delta x$, conductance $G\Delta x$, and capacitance $C\Delta x$. Using Kirchhoff's current and voltage laws for the infinitesimal section described above, and taking the limit $\Delta x \rightarrow 0$ one can derive the Telegrapher's equations to be

$$\frac{\partial}{\partial x}v(x, t) = -Ri(x, t) - L\frac{\partial}{\partial t}i(x, t) \quad (2.3a)$$

$$\frac{\partial}{\partial x}i(x, t) = -Gv(x, t) - C\frac{\partial}{\partial t}v(x, t). \quad (2.3b)$$



(a) A transmission line system



(b) An infinitesimal section of the transmission line

Figure 2.2: Single transmission line system

Taking the Laplace transform of (2.3) gives

$$\frac{\partial}{\partial x}V(x, s) = -(R + sL)I(x, s) = -ZI(x, s) \quad (2.4a)$$

$$\frac{\partial}{\partial x}I(x, s) = -(G + sC)V(x, s) = -YV(x, s) \quad (2.4b)$$

where, Z and Y represent the p.u.l. impedance and admittance of the transmission line, given by

$$Z = R + sL \quad (2.5a)$$

$$Y = G + sC. \quad (2.5b)$$

In the case of a multi-conductor transmission line (MTL) system with N coupled conductors (see Fig. 2.3), per-unit-length parameters ($\tilde{\mathbf{R}}, \tilde{\mathbf{L}}, \tilde{\mathbf{G}}, \tilde{\mathbf{C}}$) become matrices and voltage-current variables become vectors represented by \mathbf{v} and \mathbf{i} , respectively. Noting these changes, (2.3) can be written as,

$$\frac{\partial}{\partial x}\mathbf{v}(x, t) = -\tilde{\mathbf{R}}\mathbf{i}(x, t) - \tilde{\mathbf{L}}\frac{\partial}{\partial t}\mathbf{i}(x, t) \quad (2.6a)$$

$$\frac{\partial}{\partial x}\mathbf{i}(x, t) = -\tilde{\mathbf{G}}\mathbf{v}(x, t) - \tilde{\mathbf{C}}\frac{\partial}{\partial t}\mathbf{v}(x, t). \quad (2.6b)$$

Taking the Laplace transform of (2.6) gives

$$\frac{\partial}{\partial x}\mathbf{V}(x, s) = -(\tilde{\mathbf{R}} + s\tilde{\mathbf{L}})\mathbf{I}(x, s) = -\mathbf{Z}\mathbf{I}(x, s) \quad (2.7a)$$

$$\frac{\partial}{\partial x}\mathbf{I}(x, s) = -(\tilde{\mathbf{G}} + s\tilde{\mathbf{C}})\mathbf{V}(x, s) = -\mathbf{Y}\mathbf{V}(x, s) \quad (2.7b)$$

where \mathbf{Z} and \mathbf{Y} represent the p.u.l. impedance and admittance of the transmission

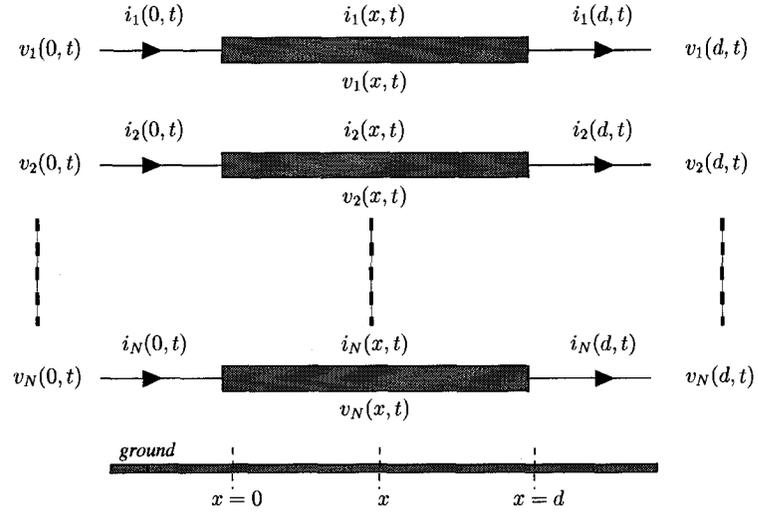


Figure 2.3: A multi-conductor transmission line system

line, given by

$$\mathbf{Z} = \tilde{\mathbf{R}} + s\tilde{\mathbf{L}} \quad (2.8a)$$

$$\mathbf{Y} = \tilde{\mathbf{G}} + s\tilde{\mathbf{C}} \quad (2.8b)$$

Equations in (2.7) can be formulated [1, 10] as,

$$\frac{\partial}{\partial x} \begin{bmatrix} \mathbf{V}(x, s) \\ \mathbf{I}(x, s) \end{bmatrix} = (\mathbf{D} + s\mathbf{E}) \begin{bmatrix} \mathbf{V}(x, s) \\ \mathbf{I}(x, s) \end{bmatrix} \quad (2.9a)$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{0} & -\tilde{\mathbf{R}} \\ -\tilde{\mathbf{G}} & \mathbf{0} \end{bmatrix} \quad (2.9b)$$

$$\mathbf{E} = \begin{bmatrix} \mathbf{0} & -\tilde{\mathbf{L}} \\ -\tilde{\mathbf{C}} & \mathbf{0} \end{bmatrix}. \quad (2.9c)$$

Using the terminal conditions, the solution of (2.9) can be written as,

$$\begin{bmatrix} \mathbf{V}(d, s) \\ \mathbf{I}(d, s) \end{bmatrix} = e^{(\mathbf{D}+s\mathbf{E})d} \begin{bmatrix} \mathbf{V}(0, s) \\ \mathbf{I}(0, s) \end{bmatrix}. \quad (2.10)$$

2.5 General Formulation of Network Equations

The modified nodal analysis (MNA) formulation for a general network ϕ containing an arbitrary number of lumped components, interconnects, and non-linear components can be expressed as [34, 35]

$$\mathbf{C}_\phi \dot{\mathbf{x}}_\phi + \mathbf{G}_\phi \mathbf{x}_\phi + \mathbf{F}_\phi(\mathbf{x}_\phi) + \sum_{\pi=1}^{N_t} \mathbf{D}_\pi \mathbf{i}_\pi(t) = \mathbf{b}_\phi(t) \quad (2.11)$$

where,

- $\mathbf{x}_\phi \in \mathfrak{R}^{n_\phi}$ is the vector of node voltage waveforms appended by independent voltage source currents, linear inductor currents, nonlinear capacitor charges, nonlinear inductor flux waveforms and currents and voltages due to nonlinear components;
- $\mathbf{C}_\phi \in \mathfrak{R}^{n_\phi \times n_\phi}$ and $\mathbf{G}_\phi \in \mathfrak{R}^{n_\phi \times n_\phi}$ are constant matrices describing the lumped memory and memoryless elements of the network, respectively;
- $\mathbf{b}_\phi(t) \in \mathfrak{R}^{n_\phi}$ is a vector with entries determined by independent voltage/current sources;
- $\mathbf{F}_\phi(\mathbf{x}_\phi) \in \mathfrak{R}^{n_\phi}$ is a function describing nonlinear elements of the circuit;
- n_ϕ is the total number of variables in the MNA formulation, and n_π is the total number of ports in the interconnect subnetwork π ;

- $\mathbf{D}_\pi = [d_{i,j}]$ with elements $d_{i,j} \in \{0, 1\}$ where $i \in \{1, 2, \dots, n_\phi\}$, $j \in \{1, 2, \dots, n_\pi\}$ with a maximum of one non-zero in each row or column, is a selector matrix that maps $\mathbf{i}_\pi(t) \in \mathfrak{R}^{n_\pi}$, the vector of currents entering the interconnect subnetwork π , into node space \mathfrak{R}^{n_ϕ} of the network ϕ ;
- N_t is the total number of distributed coupled interconnect sets.

The terminal behavior of the interconnect subnetwork π containing distributed interconnects can be written in the frequency-domain as

$$\mathbf{I}_\pi(s) = \mathbf{Y}_\pi(s)\mathbf{V}_\pi(s) = \mathbf{Y}_\pi(s)\mathbf{D}_\pi^T\mathbf{X}_\phi(s) \quad (2.12)$$

where $\mathbf{Y}_\pi(s)$ is the y -parameter matrix of subnetwork π , $\mathbf{V}_\pi(s)$ is the vector of terminal voltage nodes that connect the subnetwork π to the network ϕ . $\mathbf{I}_\pi(s)$ and $\mathbf{X}_\phi(s)$ are the Laplace transform of $\mathbf{i}_\pi(t)$ and $\mathbf{x}_\phi(t)$ respectively. The y -parameters of an interconnect can be obtained from the hybrid parameters given by (2.10).

The major difficulty in simulating high speed interconnects is that distributed transmission line equations are best represented in frequency domain; the exponential matrix $e^{(\mathbf{D}+s\mathbf{E})d}$ in (2.10) cannot be represented directly in time-domain. On the other hand, nonlinear components can only be represented in time-domain as in (2.11). Therefore, interconnect simulation with nonlinear components is referred to in literature as the *mixed frequency-time problem*. In order to address this problem, distributed transmission lines equations are approximated and transformed into a set of ordinary differential equations in time domain. These equations can then be simulated using nonlinear time-domain simulators such as SPICE [1]. In literature, several time domain models or so-called *macromodels* [1–11] have been proposed in

order to address the above-mentioned issue.

Time domain interconnect models are obtained through techniques such as discretization techniques and method of characteristics. Discretization techniques provide some form of discretization of the Telegrapher's equations in order to obtain a time domain interconnect stamp in the form of ordinary differential equations. The method of characteristics [36–38] on the other hand transforms the Telegrapher's equations into ordinary differential equations using time-delayed controlled sources. Method of characteristics remains most practical for lossless lines, while discretization techniques are acceptable for general case.

Methods such as conventional lumped segmentation [2], Chebyshev polynomials [39], compact finite-difference based approximation [8, 22], transfer function approximation [40], and matrix rational approximation [41–43] fall under the category of discretization methods. The discretization methods typically lead to large circuit matrices, rendering the methods CPU inefficient. In order to address this issue, model-reduction techniques such as Asymptotic Waveform Evaluation (AWE) [9], Complex Frequency Hopping (CFH) [10], Passive Reduced-order Interconnect Macromodeling Algorithm (PRIMA) [11], Padé Via Lanczos (PVL) [44, 45] have been proposed in literature. The model-reduction techniques bring down the computational cost by approximating the original discretized system with reduced order model that captures only fewer poles closer to the imaginary axis.

Combining both method of characteristics and matrix rational approximation, another macromodeling approach called Delay Extraction-Based Passive Compact Transmission-Line Macromodeling Algorithm (DEPACT) has been proposed in [46]. DEPACT provides a passive delay extraction macromodel based on the modified Lie

product to efficiently model long low-loss lines. However, as the losses of the line increase, a higher order Lie product approximation is required in order to provide accurate results [47].

Recently, for efficient simulation of large number of coupled lines, an approach based on transverse partitioning and waveform relaxation has been proposed in [48]. In this approach, a relaxation-based algorithm is used to iterate between individual transmission lines until convergence is obtained. It has to be noted that this is an iterative approach and convergence might not be guaranteed when there is strong coupling between lines.

2.6 Conventional Lumped Segmentation

The conventional approach for discrete modeling of distributed interconnects is to divide the transmission line into several segments of Δx , chosen to be small fraction of the minimum wavelength of interest [1, 2, 20]. Then each segment is replaced by lumped circuit elements.

In this method, partial differential equations in (2.6) are approximated by the following difference equations [2].

$$\mathbf{v}_{k+1}(t) - \mathbf{v}_k(t) \approx -\tilde{\mathbf{R}}\Delta x \mathbf{i}_k(t) - \tilde{\mathbf{L}}\Delta x \frac{\partial}{\partial t} \mathbf{i}_k(t) \quad (2.13a)$$

$$\mathbf{i}_{k+1}(t) - \mathbf{i}_k(t) \approx -\tilde{\mathbf{G}}\Delta x \mathbf{v}_k(t) - \tilde{\mathbf{C}}\Delta x \frac{\partial}{\partial t} \mathbf{v}_k(t) \quad (2.13b)$$

where

- $k = 1, 2, \dots, K$

- $\Delta x = d/K$
- K - number of segments
- d - length of interconnect
- $\Delta x \ll \lambda$.

Equation (2.13) can be implemented by lumped equivalent circuits composed of resistors, inductors, and capacitors as shown in Fig. 2.4.

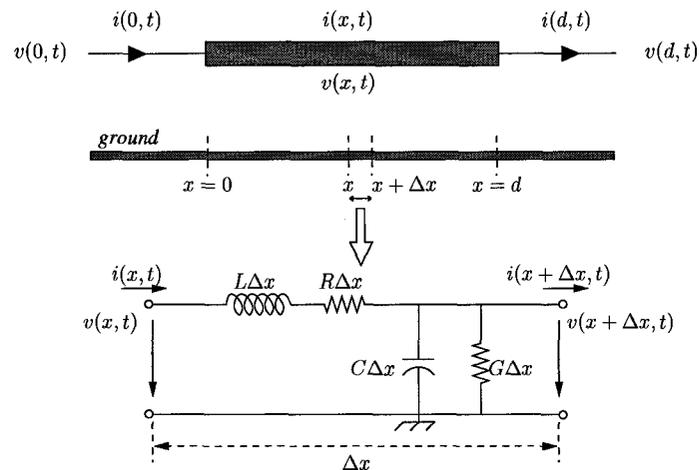


Figure 2.4: Lumped Segmentation

Ideally, if $\Delta x \rightarrow 0$, then the solution will be identical to the solution of Telegrapher's equations. However, because of limited computational resources, only a finite number of lumped segments can be used. A general rule of thumb is given by [1] to predict number of lumped segments (K):

$$K \geq \frac{10\tau d}{t_r} \quad (2.14)$$

where $\tau = \sqrt{LC}$ is the p.u.l delay of the line, and t_r is the rise time of the propagating signal.

The main advantage of this method is that the lumped components can be readily incorporated into any of the various circuit simulators such as SPICE [2]. Also passivity of the macromodel is always preserved since all components of the model are passive elements. As a fundamental approach, this method has been extensively referenced in the literature [20, 43, 49–53] for comparison with other macromodeling techniques. However, if the frequency of interest is very high or the interconnect is electrically long, then many lumped segments are required. This will lead to large circuit matrices, rendering the method inefficient.

2.6.1 Simulation of Interconnects Discretized with Lumped Segments

Once all the interconnect subnetworks are discretized using lumped elements as given by Fig. 2.4, the resulting network will only contain lumped passive elements, source elements, and non-linear elements. Then by incorporating all these elements using lumped circuit theory, time domain ordinary differential equation can be formulated as follows:

$$\mathbf{C}\dot{\mathbf{x}} + \mathbf{G}\mathbf{x} + \mathbf{F}(\mathbf{x}) = \mathbf{b}(t) \quad (2.15)$$

where,

- $\mathbf{x} \in \mathbb{R}^{n_l}$, $\mathbf{b}(t) \in \mathbb{R}^{n_l}$, $\mathbf{C} \in \mathbb{R}^{n_l \times n_l}$, $\mathbf{G} \in \mathbb{R}^{n_l \times n_l}$, and $\mathbf{F}(\mathbf{x}) \in \mathbb{R}^{n_l}$ are expanded by incorporating newly generated lumped components (due to the discretization of interconnect subnetworks $\pi = 1 \dots N_t$) with the existing \mathbf{x}_ϕ , \mathbf{b}_ϕ , \mathbf{C}_ϕ , \mathbf{G}_ϕ , and $\mathbf{F}_\phi(\mathbf{x}_\phi)$ in (2.11) respectively;
- $n_l \gg n_\phi$.

In order to compute the solution of the above nonlinear algebraic differential equation numerically, numerical integration techniques need to be applied and therefore, the equation can be transformed into a set of nonlinear algebraic difference equations in the form:

$$\left(\mathbf{G} + \frac{\lambda}{h}\mathbf{C}\right)\mathbf{x}_{k+1} + \mathbf{F}(\mathbf{x}_{k+1}) - \mathbf{R}(\mathbf{x}_k) = 0 \quad (2.16)$$

where h is the step size used in the simulation, and \mathbf{x}_k is the vector of unknowns at the k -th time point. The constant λ and the function $\mathbf{R}(\mathbf{x}_k)$ are dependent on the chosen numerical integration formula. The solution to the original system of equations is obtained by performing Newton iterations on (2.16) and it can be formulated as follows

$$\mathbf{M}\Delta\mathbf{x} = \phi \quad (2.17)$$

where,

$$\mathbf{M} = \mathbf{G} + \frac{\lambda\mathbf{C}}{h} + \frac{\partial\mathbf{F}}{\partial\mathbf{x}_{k+1}} \quad (2.18)$$

$$\phi = \left(\mathbf{G} + \frac{\lambda\mathbf{C}}{h}\right)\mathbf{x}_{k+1} + \mathbf{F}(\mathbf{x}_{k+1}) - \mathbf{R}(\mathbf{x}_k). \quad (2.19)$$

The size of the Jacobian matrix \mathbf{M} , is extremely large as it arises from discretization of interconnects in the network. In order to obtain the solution, LU decompositions on \mathbf{M} need to be performed for each Newton-Raphson iteration. Performing LU decompositions on such a large matrix several times is computationally intensive. Especially in the case where operating frequencies are very high, the simulation time becomes worse since the transmission line must be discretized very finely in order to obtain accurate results.

The proposed method avoids the necessity to solve large linear system by eliminating redundancies present in simulation of interconnect networks by exploiting the repetitive nature of lumped segments with the aid of domain decomposition techniques. Hence, it gives considerable speed-up without approximating the solution of (2.17). Details of this method are given in Chapter 3.

There are efficient macromodeling techniques proposed in literature [9, 11, 36–38, 44, 45] that do not require solving large linear system of equations. However, in these methods, accuracy or passivity is compromised in order to obtain fast results. Brief overview of some of those methods (method of characteristics and model-reduction techniques) and issues relating to them are given in the next section.

2.7 Efficient Macromodeling Approaches

Efficient methods were proposed in the literature to address simulation issues of interconnect networks using a brute-force lumped segmentation technique. While bringing down the CPU cost of simulating interconnect networks these methods typically *compromise on the accuracy* of the response and on some cases even end up *generating non-passive networks*. Interconnect networks are passive in nature and any macromodel representing them should also be passive. Passivity [1, 7, 11, 43, 54–57] is an important property, because stable but not passive macromodels can lead to unstable systems when connected to other passive systems. The loss of passivity can be a serious problem because transient simulation of non-passive networks may result in artificial oscillations. Issues relating with two broad macromodelling approaches are discussed in detail in the following subsections.

2.7.1 Method of Characteristics

Method of characteristics [36–38] transforms partial differential equations of transmission lines into ordinary differential equations with the aid of time-delayed controlled sources.

The frequency domain solution of (2.3) for a single transmission line can be expressed [37] as,

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \frac{1}{Z_0(1 - e^{-2\gamma d})} \begin{bmatrix} 1 + e^{-2\gamma d} & -2e^{-\gamma d} \\ -2e^{-\gamma d} & 1 + e^{-2\gamma d} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (2.20a)$$

$$\gamma = \sqrt{(R + sL)(G + sC)} \quad (2.20b)$$

$$Z_0 = \sqrt{\frac{R + sL}{G + sC}} \quad (2.20c)$$

where γ is the propagation constant, Z_0 is the characteristic impedance, V_1 and I_1 are the terminal voltage and current at the near end of the line, V_2 and I_2 are the terminal voltage and current at the far end of the line. By re-arranging the terms in equation (2.20) the following expressions are obtained

$$V_1 = Z_0 I_1 + e^{-\gamma d} [2V_2 - e^{-\gamma d} (Z_0 I_1 + V_1)] \quad (2.21a)$$

$$V_2 = Z_0 I_2 + e^{-\gamma d} [2V_1 - e^{-\gamma d} (Z_0 I_2 + V_2)]. \quad (2.21b)$$

Expression (2.21) can be rewritten as

$$V_1 - Z_0 I_1 = W_1 \quad (2.22a)$$

$$V_2 - Z_0 I_2 = W_2 \quad (2.22b)$$

where

$$W_1 = e^{-\gamma d}[2V_2 - e^{-\gamma d}(Z_0 I_1 + V_1)] \quad (2.23a)$$

$$W_2 = e^{-\gamma d}[2V_1 - e^{-\gamma d}(Z_0 I_2 + V_2)]. \quad (2.23b)$$

Using (2.21) and (2.23), a recursive relation for W_1 and W_2 can be obtained as

$$W_1 = e^{-\gamma d}[2V_2 - W_2] \quad (2.24a)$$

$$W_2 = e^{-\gamma d}[2V_1 - W_1]. \quad (2.24b)$$

If the transmission line is lossless, γ and Z_0 are reduced to

$$\gamma = s\sqrt{LC} \quad (2.25a)$$

$$Z_0 = \sqrt{\frac{L}{C}}. \quad (2.25b)$$

Here γ is purely imaginary and Z_0 is a real constant. Therefore, using inverse Laplace transform, the frequency domain expression (2.24) can be analytically converted into time domain as,

$$w_1(t + \tau) = 2v_2(t) - w_2(t) \quad (2.26a)$$

$$w_2(t + \tau) = 2v_1(t) - w_1(t). \quad (2.26b)$$

where

$$\tau = d\sqrt{LC} \quad (2.27)$$

Equation (2.26) can be easily be incorporated into nonlinear circuit simulators with the aid of time-delayed controlled sources.

For lossy lines, the propagation constant is not purely imaginary and, hence, cannot be replaced by a pure delay. In that case, analytical expressions for $w_1(t)$ and $w_2(t)$ cannot be directly obtained in time domain. Therefore, this method is being more practical for lossless lines. In order to handle lossy lines, classical method of characteristics can be extended through Padé synthesis [37]. In case of coupled lines, this method can be applied through decoupling of multi-conductor transmission line equations [2].

The main disadvantage of method of characteristics is that the resulting macromodels are not guaranteed to be passive [1].

2.7.2 Model-Reduction Techniques

Interconnect networks generally tend to have a large number of poles spread over a wide frequency range. Model-reduction techniques approximate the original system with a reduced order model that captures few poles that are closer to the imaginary axis, called dominant poles, which preserve the dominant characteristics of the system. Model-reduction methods can be classified into two categories: 1) explicit moment-matching methods (such as AWE [9], CFH [10]) 2) implicit moment-matching methods (such as PVL [44, 45], PRIMA [11]). Explicit methods, which are based on Padé approximation, can generally find fewer than ten poles [1] due to ill-conditioning associated with the moment-matrix. Also these methods do not guarantee passivity of the reduced macromodel. Implicit methods obtain reduced macromodel based on Krylov-subspace techniques and congruent transformation. They construct reduced-

model based on the extraction of leading eigenvalues (those with largest magnitude) of a given system.

Despite implicit moment-matching methods overcoming issues with explicit methods, it has to be noted that the response obtained from the macromodel is still an approximation of the original network. In addition, model-reduction methods suffer from a degradation in efficiency with an increase in the number of ports [12]. As the operating frequency increases, number of poles need to be captured must be increased too. However, increasing number of poles need to be captured in turn increases the computational cost significantly in case of coupled lines. Also, number of poles required to obtain an accurate response within the frequency of interest cannot be determined *as a priori*. Therefore, if the efficiency is of concern, then the accuracy cannot be guaranteed, and vice-versa.

Typically, techniques such as those mentioned above eliminate an implicit redundancy present in interconnect networks while trading-off on accuracy and/or passivity of the system. The next chapter presents a technique that takes advantage of the redundancy present in interconnect networks without affecting either the accuracy of passivity of the system.

Chapter 3

Proposed Interconnect Simulation

Method

The proposed technique partitions the interconnect longitudinally into several domains and solves the system of equations using domain decomposition techniques. By exploiting the repetitive nature of lumped segments, redundancy in simulation of interconnect networks is removed and hence, computational cost of direct solution of (2.17) is reduced. Sec. 3.1 explains the partitioning approach for interconnect networks that minimizes coupling between these domains. Background information of domain decomposition technique that used to solve the system of equations is presented in Sec. 3.2. In Sec. 3.4, an approach to remove the redundancy in simulation of interconnect networks is presented.

3.1 Circuit Partitioning

In order to reduce the computational cost of a direct solution of (2.17), the coupled interconnect network discretized with lumped segments is initially partitioned longitudinally into n domains as shown in Fig. 3.1. The length of partitions $2 \dots n-1$ are kept to be equal.

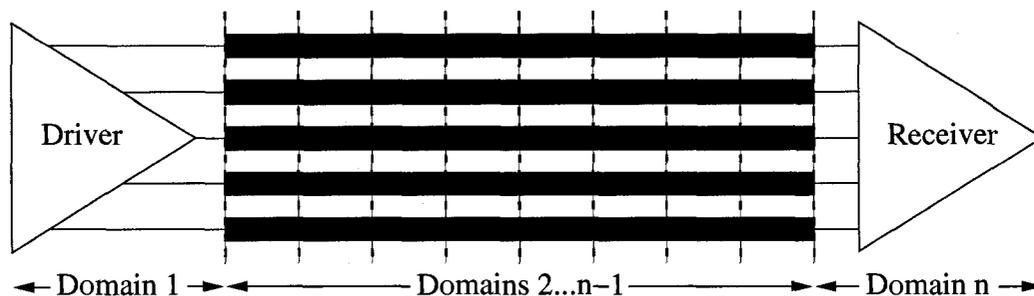


Figure 3.1: Longitudinal partitioning of an interconnect

From Fig. 3.1 it can be observed that domains $2 \dots n-1$ are identical for uniform coupled interconnect networks. The proposed method exploits this observation by selecting any one domain out of identical domains $2 \dots n-1$ along with the first and last partitions (domains 1 and n) to simulate the interconnect network leading to a considerable speedup. Such a strategy can be mathematically achieved through domain decomposition methods as illustrated in the rest of this section.

When solving through a partitioning approach such as domain decomposition method, the important aspect that affects the efficiency of the applied method is that the number of coupling variables or interface variables between partitions. The larger the number of interface variables, the lesser the efficiency that can be achieved through approaches like domain decomposition method. Arbitrary partitioning could lead to a large number of interface variables. Therefore partitions have to be formed

such that interface variables are minimized.

Since the transmission line formulation (Chapter 2) is always under the assumption of quasi-TEM mode propagation [2], there is negligible coupling in the direction of propagation due to the assumption of negligible E and H field components in the direction of propagation. Therefore, longitudinal partitioning approach, utilized by the proposed method, results in forming partitions with minimal interface variables.

For example, segment 1 of the first line in Fig. 3.2, has coupling with the first segment of the second line. However, there is relatively less coupling between segment 1 and segment 2 or segment 3. In general, there is negligible coupling between i -th and j -th segments of the interconnect network, where $i \neq j$. Therefore, in this partitioning approach the coupling effects are encapsulated within each domain.

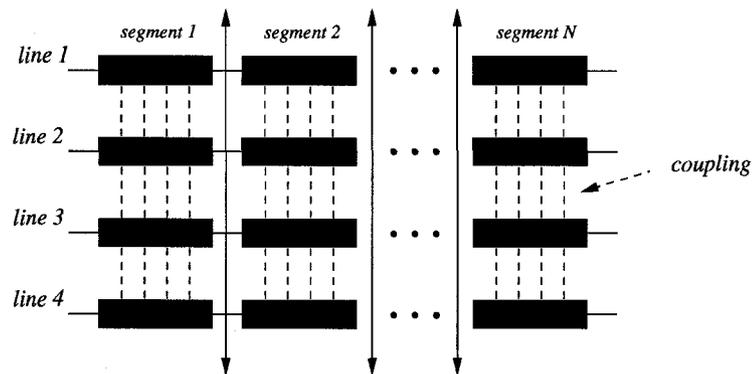


Figure 3.2: Longitudinal partitioning and coupling between each discretized segment

3.2 Simulation of Discretized Interconnects using Domain Decomposition

Domain decomposition in the context of linear algebra refers to the process of splitting a large linear system into smaller problems on sub-domains in order to obtain the solution of the linear system by solving the sub-domains individually [13, 58, 59]. Domain decomposition is typically used for parallel processing applications in literature [13–16, 58, 60, 61].

In domain decomposition theory, variables can be classified into two categories: (1) interior variables coupled only with variables belonging to a specific sub-domain; (2) interface variables located on the boundaries between neighboring sub-domains and coupled with interior variables of neighboring sub-domains as well as other interface variables [15, 60]. Fig. 3.3 illustrates the classification of MNA variables (interface variables circled) according to domain decomposition theory for a small circuit.

By separating variables in $\Delta \mathbf{x}$ into interior and interface variables, (2.17) can be rewritten as [15],

$$\begin{bmatrix} \mathbf{M}_1 & & & \mathbf{P}_1 \\ & \mathbf{M}_2 & & \mathbf{P}_2 \\ & & \ddots & \vdots \\ \mathbf{Q}_1 & \mathbf{Q}_2 & \dots & \mathbf{M}_z \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_1 \\ \Delta \mathbf{x}_2 \\ \vdots \\ \Delta \mathbf{x}_{n+1} \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n+1} \end{bmatrix} \quad (3.1)$$

where

- $\mathbf{M}_i \in \mathfrak{R}^{n_i \times n_i}$ is a sparse matrix representing equations of the i -th domain,
- $\mathbf{P}_i \in \mathfrak{R}^{n_i \times n_\sigma}$, $\mathbf{Q}_i \in \mathfrak{R}^{n_\sigma \times n_i}$ are sparse matrices representing interactions between

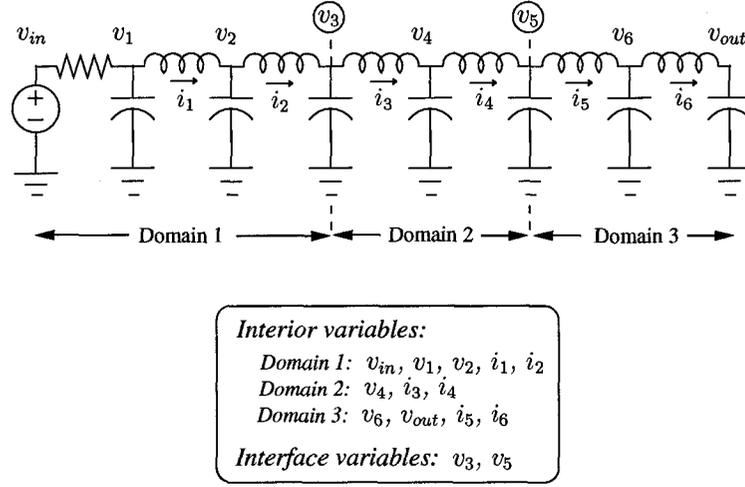


Figure 3.3: Example of classification of MNA variables according to domain decomposition theory

interior variables of domain i and interface variables,

- $\mathbf{M}_z \in \mathfrak{R}^{n_\sigma \times n_\sigma}$ is a sparse matrix representing coupling between interface variables,
- $\Delta \mathbf{x}_i \in \mathfrak{R}^{n_i}$ is the vector of variables of the i -th domain,
- $\Delta \mathbf{x}_{n+1} \in \mathfrak{R}^{n_\sigma}$ is the vector containing interface variables,
- ϕ_i is the right hand side of (2.17) corresponding to the i -th domain,
- ϕ_{n+1} is the right hand side of (2.17) corresponding to interface variables,
- n_i is the size of each domain,
- n_σ is the size of the interface matrix.

From (3.1), the interior variables can be expressed as

$$\Delta \mathbf{x}_i = \mathbf{M}_i^{-1}(\phi_i - \mathbf{P}_i \Delta \mathbf{x}_{n+1}); \quad i = 1 \dots n \quad (3.2)$$

Using (3.2) and (3.1),

$$\mathbf{S}\Delta\mathbf{x}_{n+1} = \phi_{n+1} - \sum_{i=1}^n (\mathbf{Q}_i \mathbf{M}_i^{-1} \phi_i) \quad (3.3)$$

where

$$\mathbf{S} = \mathbf{M}_z - \sum_{i=1}^n (\mathbf{Q}_i \mathbf{M}_i^{-1} \mathbf{P}_i). \quad (3.4)$$

\mathbf{S} is called the *Schur complement matrix*. Solving (3.3) for $\Delta\mathbf{x}_{n+1}$ evaluates the interface variables. These interface variables are substituted in (3.2) to evaluate $\Delta\mathbf{x}_i$ for each domain which is also the solution of the original interconnect network.

The Fig. 3.4 - 3.5 illustrate an example of sparsity patterns for \mathbf{M} , which was obtained with a two transmission line network discretized with 10 lumped segments/line, with natural and domain decomposed node ordering respectively.

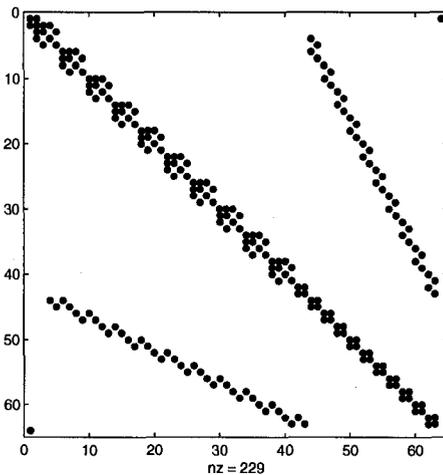


Figure 3.4: Sparsity pattern of \mathbf{M} with natural node ordering.

In Sec. 3.4, a way of taking advantage of the fact that domains $2 \dots n - 1$ are

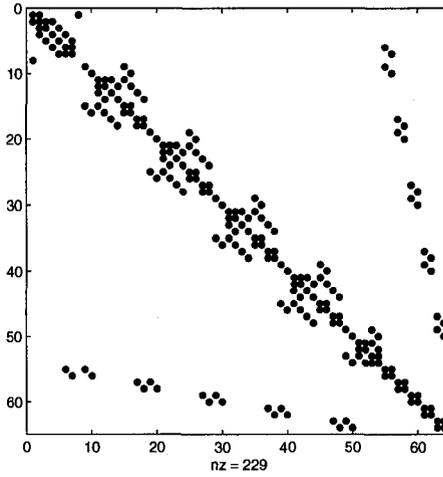


Figure 3.5: Sparsity pattern of M with domain decomposed ordering (6 domains).

identical to obtain a speed-up is presented.

3.3 Analysis of Transient Simulation with Domain Decomposition

When domain decomposition is applied for an interconnect network, the matrices involved in MNA equations can be written in the same form of (3.1) as,

$$G = \begin{bmatrix} G_1 & & & G_{p1} \\ & G_2 & & G_{p2} \\ & & \ddots & \vdots \\ G_{q1} & G_{q2} & \dots & G_z \end{bmatrix} \quad (3.5a)$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & & & \mathbf{C}_{\mathbf{p}1} \\ & \mathbf{C}_2 & & \mathbf{C}_{\mathbf{p}2} \\ & & \ddots & \vdots \\ \mathbf{C}_{\mathbf{q}1} & \mathbf{C}_{\mathbf{q}2} & \dots & \mathbf{C}_z \end{bmatrix} \quad (3.5b)$$

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} \mathbf{F}_1(\mathbf{x}) \\ \mathbf{F}_2(\mathbf{x}) \\ \vdots \\ \mathbf{F}_{n+1}(\mathbf{x}) \end{bmatrix} \quad (3.5c)$$

$$\frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{F}_1}{\partial \mathbf{x}_1} & & & \frac{\partial \mathbf{F}_1}{\partial \mathbf{x}_{n+1}} \\ & \frac{\partial \mathbf{F}_2}{\partial \mathbf{x}_2} & & \frac{\partial \mathbf{F}_2}{\partial \mathbf{x}_{n+1}} \\ & & \ddots & \vdots \\ \frac{\partial \mathbf{F}_{n+1}}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{F}_{n+1}}{\partial \mathbf{x}_2} & \dots & \frac{\partial \mathbf{F}_{n+1}}{\partial \mathbf{x}_{n+1}} \end{bmatrix} \quad (3.5d)$$

Then the domain decomposition equation in (3.1), can be expanded in the form of (2.18) as,

$$\mathbf{M}_i = \mathbf{G}_i + \frac{\lambda \mathbf{C}_i}{h} + \frac{\partial \mathbf{F}_i}{\partial \mathbf{x}_i} \quad (3.6a)$$

$$\mathbf{P}_i = \mathbf{G}_{\mathbf{p}i} + \frac{\lambda \mathbf{C}_{\mathbf{p}i}}{h} + \frac{\partial \mathbf{F}_i}{\partial \mathbf{x}_{n+1}} \quad (3.6b)$$

$$\mathbf{Q}_i = \mathbf{G}_{\mathbf{q}i} + \frac{\lambda \mathbf{C}_{\mathbf{q}i}}{h} + \frac{\partial \mathbf{F}_{n+1}}{\partial \mathbf{x}_i} \quad (3.6c)$$

$$\mathbf{M}_z = \mathbf{G}_z + \frac{\lambda \mathbf{C}_z}{h} + \frac{\partial \mathbf{F}_{n+1}}{\partial \mathbf{x}_{n+1}} \quad (3.6d)$$

In a typical driver-interconnect-receiver signal integrity analysis, the nonlinear elements are found in the driver and receiver components only. When applying the proposed partitioning technique, the driver components are contained within first

domain and the receiver components are contained within last domain. As a result, the nonlinear functions can be found only in the first and last domains. Therefore, the nonlinear functions for other domains as well as interface domain become zero as given by,

$$\mathbf{F}_i(\mathbf{x}) = \mathbf{0}; \quad i = 2 \dots (n-1), (n+1). \quad (3.7)$$

Also, the nonlinear functions representing first and last domains have dependency only with their local domain interior variables and interface variables. This can be expressed as,

$$\mathbf{F}_1(\mathbf{x}) = \mathbf{F}_1(\mathbf{x}_1, \mathbf{x}_{n+1}) \quad (3.8a)$$

$$\mathbf{F}_n(\mathbf{x}) = \mathbf{F}_n(\mathbf{x}_n, \mathbf{x}_{n+1}). \quad (3.8b)$$

Using (3.7) and (3.8), the expressions (3.5c) and (3.5d) can be written as,

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} \mathbf{F}_1(\mathbf{x}_1, \mathbf{x}_{n+1}) \\ \mathbf{0} \\ \vdots \\ \mathbf{F}_n(\mathbf{x}_n, \mathbf{x}_{n+1}) \\ \mathbf{0} \end{bmatrix} \quad (3.9a)$$

$$\frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{F}_1(\mathbf{x}_1)}{\partial \mathbf{x}_1} & & & & \frac{\partial \mathbf{F}_1}{\partial \mathbf{x}_{n+1}} \\ & \mathbf{0} & & & \mathbf{0} \\ & & \ddots & & \vdots \\ & & & \frac{\partial \mathbf{F}_n}{\partial \mathbf{x}_n} & \frac{\partial \mathbf{F}_n}{\partial \mathbf{x}_{n+1}} \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{0} \end{bmatrix}. \quad (3.9b)$$

Using (3.9), (3.6) can be simplified for all domains except first and last as given by

$$\mathbf{M}_i = \mathbf{G}_i + \frac{\lambda \mathbf{C}_i}{h} \quad (3.10a)$$

$$\mathbf{P}_i = \mathbf{G}_{\mathbf{p}_i} + \frac{\lambda \mathbf{C}_{\mathbf{p}_i}}{h} \quad (3.10b)$$

$$\mathbf{Q}_i = \mathbf{G}_{\mathbf{q}_i} + \frac{\lambda \mathbf{C}_{\mathbf{q}_i}}{h} \quad (3.10c)$$

$$\mathbf{M}_z = \mathbf{G}_z + \frac{\lambda \mathbf{C}_z}{h} \quad (3.10d)$$

where

$$i = 2, \dots, n - 1.$$

The formulation (3.10a) yields certain advantages over traditional simulation strategies. In transient simulation, the step size, h , between any two time points, $t = t_k$ and $t = t_{k+1}$, will remain constant throughout all Newton iterations until the solution, $x(t_{k+1})$, converges. Also, the matrices \mathbf{G}_i and \mathbf{C}_i will remain constant during simulation. Therefore, the LU decompositions (for \mathbf{M}_i^{-1}) for the domains $2, \dots, n - 1$ need to be performed only during the first Newton iteration; for the subsequent iterations \mathbf{M}_i^{-1} will not change for $i = 2, \dots, n - 1$. Only \mathbf{M}_1 and \mathbf{M}_n will change with subsequent iterations due to the presence of nonlinearities. However, since the size of \mathbf{M}_1 and \mathbf{M}_n is small, performing LU decompositions for them is not very CPU expensive. On the other hand, traditional simulation requires performing LU decompositions for the large matrix, \mathbf{M} , for each iteration in a given time point. In some cases, where several Newton iterations required for a specific time point, the proposed method could save a considerable amount of CPU time compared to the traditional simulation.

The proposed method can save a significant amount of CPU time in the case of having a constant step size throughout the transient simulation. From (3.10a), when h remains constant, \mathbf{M}_i will also remain constant for $i = 2, \dots, n - 1$ since \mathbf{G}_i and \mathbf{C}_i are constant. Therefore, it is sufficient to perform only *one LU decomposition* for $\mathbf{M}_2, \dots, \mathbf{M}_{n-1}$ at the beginning of the transient simulation program. Then during the time marching and Newton iteration process, we need to perform LU decompositions for \mathbf{M}_1 and \mathbf{M}_n alone. Therefore, if an appropriate constant step size can be chosen for a given interconnect simulation problem, then the proposed method will be considerably efficient. On the other hand, in traditional simulation, keeping h constant will not lead to any improvements since variation of the Jacobian term, $\frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{x}}$, will change \mathbf{M} during each Newton iteration and hence, LU decomposition for the large matrix must be performed for each iteration.

In addition, further computational speed-up can be obtained as detailed in the following subsection.

3.4 Removing Redundancy in Simulation of Interconnect Networks

When the proposed partitioning technique is applied, partitions excluding first and last will not contain any additional active or passive components. Therefore, given that the physical dimensions of domains $2 \dots n - 1$ are kept to be equal, and their p.u.l parameters are uniform, it can be proved that the domains $2 \dots n - 1$ are identical to

each other. Therefore, it can be written as,

$$\mathbf{G}_2 = \mathbf{G}_3 = \dots = \mathbf{G}_{n-1} \quad (3.11a)$$

$$\mathbf{C}_2 = \mathbf{C}_3 = \dots = \mathbf{C}_{n-1}. \quad (3.11b)$$

Using (3.11) and (3.10a), we have,

$$\mathbf{M}_2 = \mathbf{M}_3 = \dots = \mathbf{M}_{n-1}. \quad (3.12)$$

Therefore, without loss of generality, substituting \mathbf{M}_2 for $\mathbf{M}_i, i = 3 \dots n - 1$, (3.2)–(3.4) become

$$\Delta \mathbf{x}_i = \begin{cases} \mathbf{M}_i^{-1}(\phi_i - \mathbf{P}_i \Delta \mathbf{x}_{n+1}) & i = 1, n \\ \mathbf{M}_2^{-1}(\phi_i - \mathbf{P}_i \Delta \mathbf{x}_{n+1}) & i = 2 \dots n - 1 \end{cases} \quad (3.13)$$

$$\mathbf{S} \Delta \mathbf{x}_{n+1} = \phi_{n+1} - \mathbf{Q}_1 \mathbf{M}_1^{-1} \phi_1 - \sum_{i=2}^{n-1} (\mathbf{Q}_i \mathbf{M}_2^{-1} \phi_i) - \mathbf{Q}_n \mathbf{M}_n^{-1} \phi_n \quad (3.14)$$

$$\mathbf{S} = \mathbf{M}_z - \mathbf{Q}_1 \mathbf{M}_1^{-1} \mathbf{P}_1 - \sum_{i=2}^{n-1} (\mathbf{Q}_i \mathbf{M}_2^{-1} \mathbf{P}_i) - \mathbf{Q}_n \mathbf{M}_n^{-1} \mathbf{P}_n. \quad (3.15)$$

The above formulation shows that only three small LU decompositions of the domain matrices (\mathbf{M}_1^{-1} , \mathbf{M}_2^{-1} , and \mathbf{M}_n^{-1}) need to be performed to obtain the solution as opposed to large LU decomposition on \mathbf{M} for traditional method. LU decomposition of domains, $3 \dots n - 1$, becomes redundant and is removed from the computational process. By removing this redundancy the proposed algorithm achieves a considerable speed-up over traditional methods. The solution obtained using the proposed

method has the same numerical accuracy as (2.17).

This method can be parallelized since it is based on domain decomposition method; it has to be noted that domain decomposition methods were originally used for parallel processing. Parallelization techniques can be similar to the techniques provided in literature [14–16, 61] for parallel circuit simulation.

Numerical results are presented in the next section that demonstrate the accuracy and efficiency of the proposed method.

3.5 Numerical Results

A nonlinear simulator with the proposed algorithm has been implemented in C using UMFPACK 4.6 (matrix solving package) [62, 63]. Pseudocode of the implementation and further implementation techniques are given in Chapter 4. The numerical examples are simulated in the following environment:

- CPU: AMD Athlon 64-bit (2.2 GHz)
- Memory: 1.0 GB
- Operating System: Linux

3.5.1 Example 1

A network consisting of two coupled lines connected to an active load as shown in Fig. 3.6 is analyzed using the proposed technique. The simulation is performed for two different cases of number of lumped segments: (1) 120 (2) 1200. The time domain response of a trapezoidal input pulse with 0.05 ns rise/fall time, 0.9 ns pulsewidth, and a 2 ns period are shown in Figs. 3.7 - 3.8. The results obtained from this

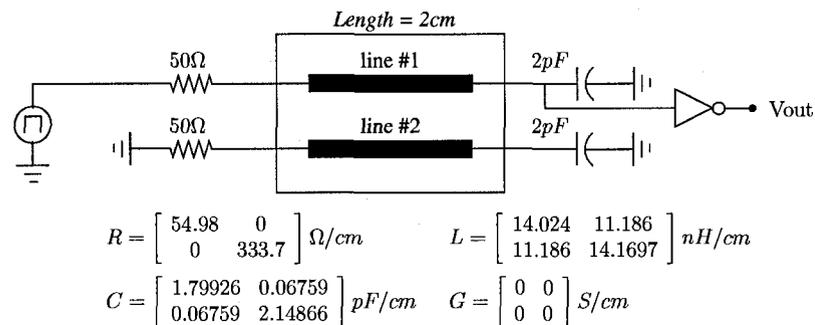


Figure 3.6: Two coupled interconnect network for Example 1

Lumped segments	Size	CPU Time		Speed-up
		Traditional	Proposed	
120	728	3.11	0.62	5.01
1200	7208	35.23	4.51	7.81

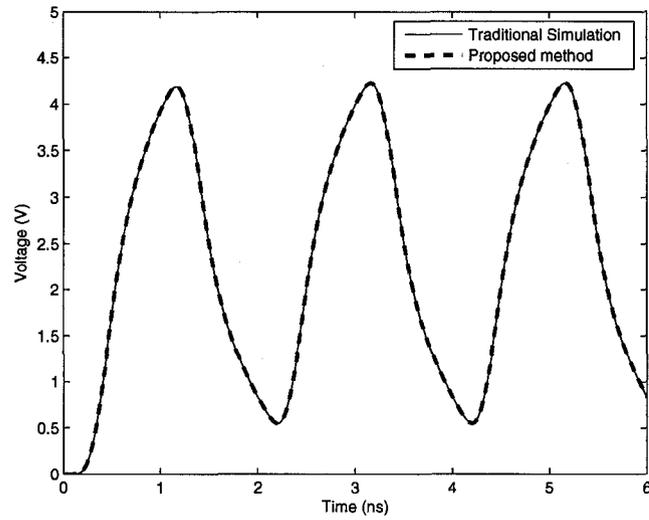
Table 3.1: Computational results for Example 1 using variable step-size

method are exactly equal to the results obtained from the traditional simulation of the interconnect network.

The CPU time results and the speed-up obtained are shown in Table 3.1 - 3.2 for variable and constant step size cases respectively. For variable step size, an optimal speed-up of 5.0 and 7.8 were obtained with 15 and 40 intermediate domains respectively for the two cases (120 and 1200) considered. For constant step size, an optimal speed-up of 7.9 and 9.8 were obtained with 4 and 20 intermediate domains respectively for the two cases.

3.5.2 Example 2

In this example, a lossless nine transmission line network from [64] is connected to an active load as shown in Fig. 3.9. Two cases of interconnect length (d) were considered:



(a) Far end - signal line

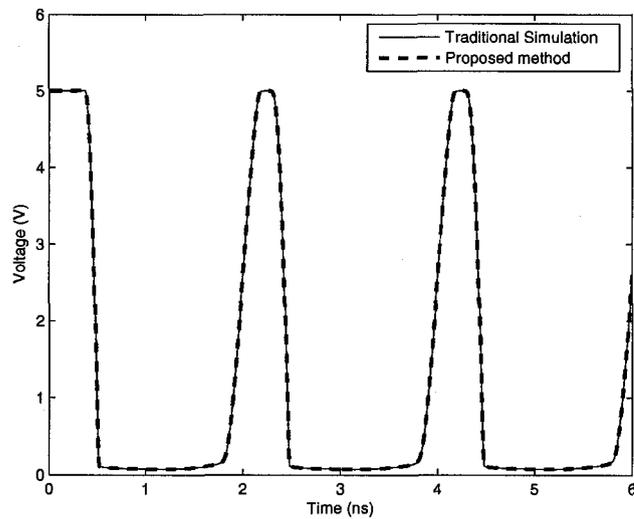
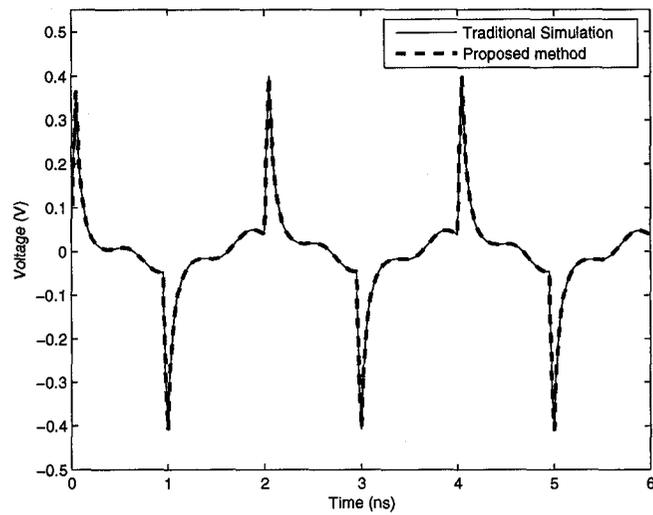
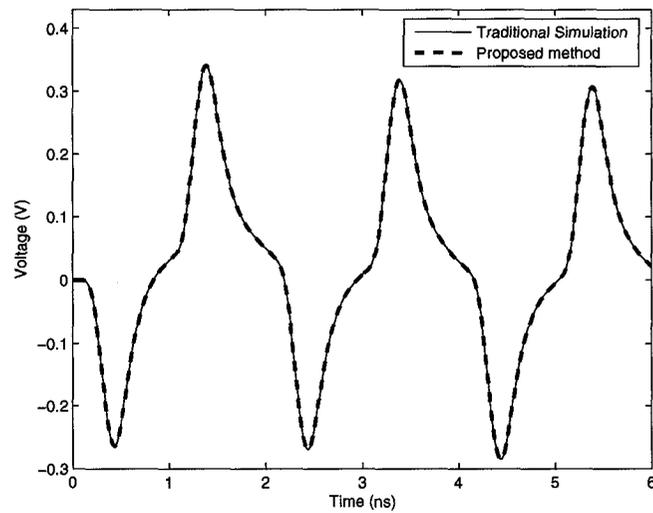
(b) V_{out}

Figure 3.7: Time-domain response of interconnect network in Example 1 with 120 lumped segments



(a) Near end - victim line



(b) Far end - victim line

Figure 3.8: Time-domain response of interconnect network in Example 1 with 1200 lumped segments

Lumped segments	Size	CPU Time		Speed-up
		Traditional	Proposed	
120	728	3.10	0.39	7.95
1200	7208	35.02	3.56	9.84

Table 3.2: Computational results for Example 1 using constant step-size

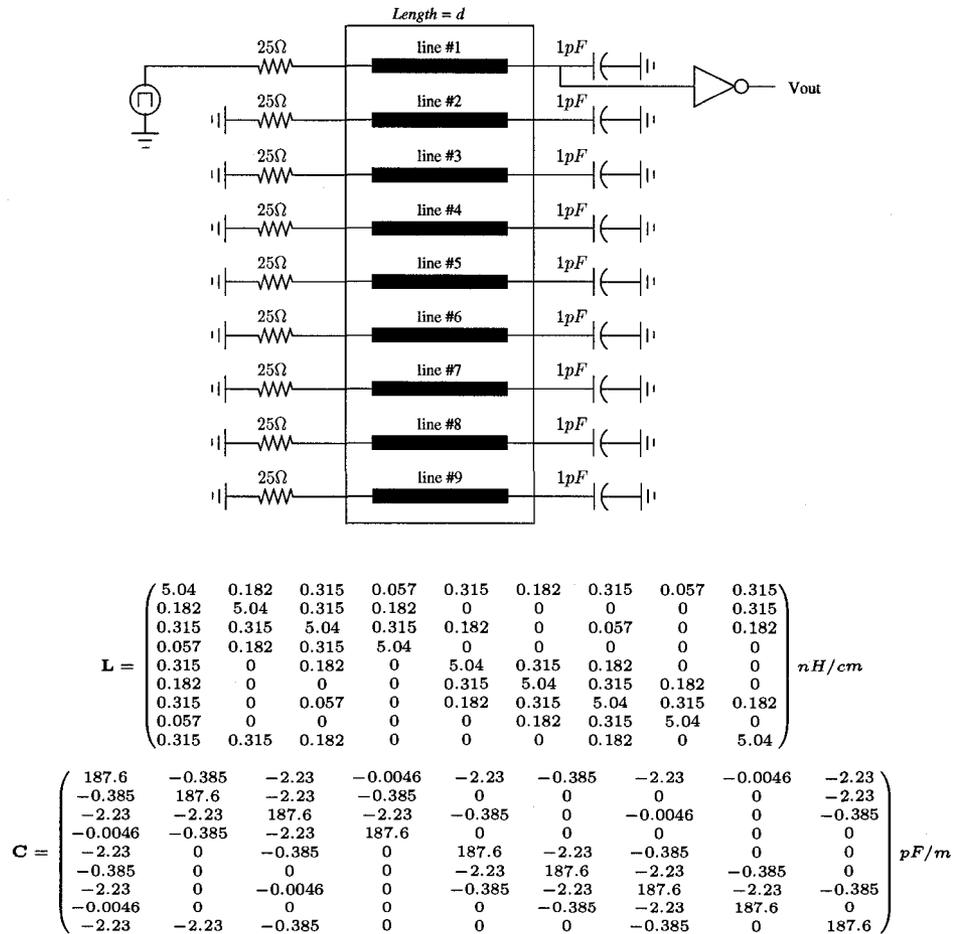


Figure 3.9: Nine coupled interconnect network from [64] for Example 2

$d(cm)$	Size	CPU Time		Speed-up
		Traditional	Proposed	
5	2171	14.96	4.77	3.14
15	6491	47.02	11.13	4.22

Table 3.3: Computational results for Example 2 using variable step-size

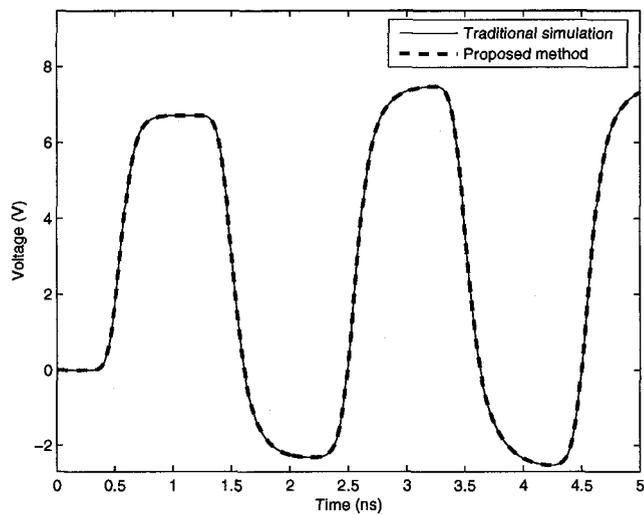
$d(cm)$	Size	CPU Time		Speed-up
		Traditional	Proposed	
5	2171	14.81	2.50	5.92
15	6491	46.53	6.32	7.36

Table 3.4: Computational results for Example 2 using constant step-size

(1) $d = 5cm$ (120 lumped segments)(2) $d = 15cm$ (360 lumped segments). The time domain response of a trapezoidal input pulse with 0.05 ns rise/fall time, 0.9 ns pulsewidth, and a 2 ns period are shown in Figs. 3.10 - 3.13. The CPU time results and the speed-up obtained are shown in Table 3.3 - 3.4 for variable and constant step size cases respectively. For variable step size, an optimal speed-up of 3.1 and 4.2 were obtained with 20 and 30 intermediate domains respectively for the two cases considered. For constant step size, an optimal speed-up of 5.9 and 7.4 were obtained with three and eight intermediate domains respectively for the two cases.

3.5.3 Example 3

In this example, a network comprised of two and nine transmission lines, with p.u.l parameters given in example 1 and 2 respectively, are connected as in Fig. 3.14. All the nine transmission line portions have an equal length of 2.5 cm and each portion was discretized with 60 lumped segments, while all two transmission line portions have



(a) Far end - signal line

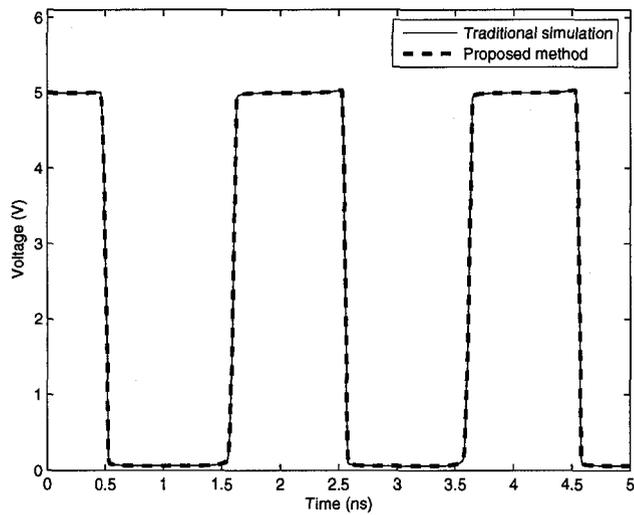
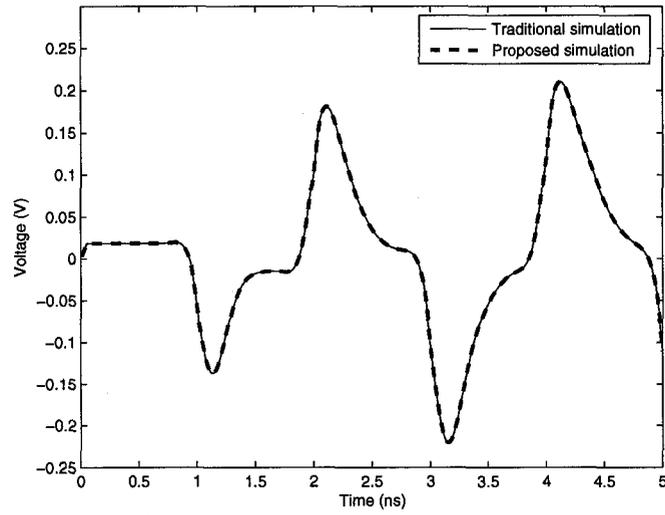
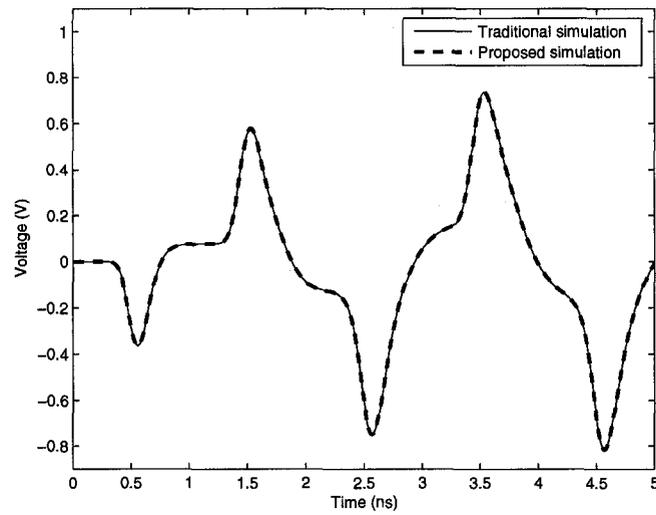
(b) V_{out}

Figure 3.10: Time-domain response of Example 2 ($d = 5\text{cm}$) for far-end of signal line and V_{out}

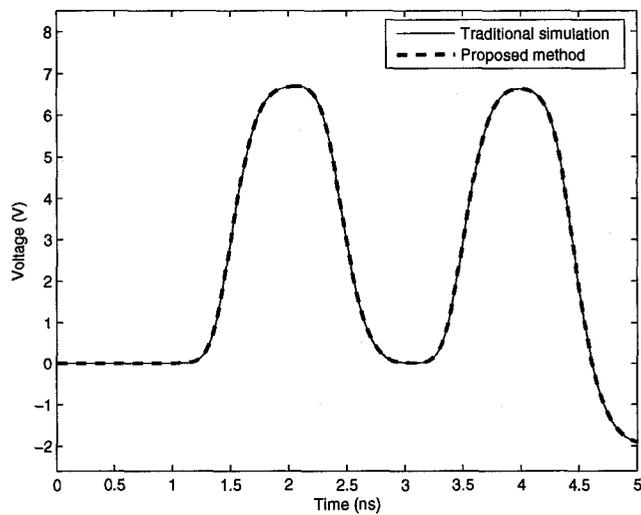


(a) Near end - first victim line



(b) Far-end - last line

Figure 3.11: Time-domain response of Example 2 ($d = 5\text{cm}$) for near-end of first victim line and far-end of last line



(a) Far end - signal line

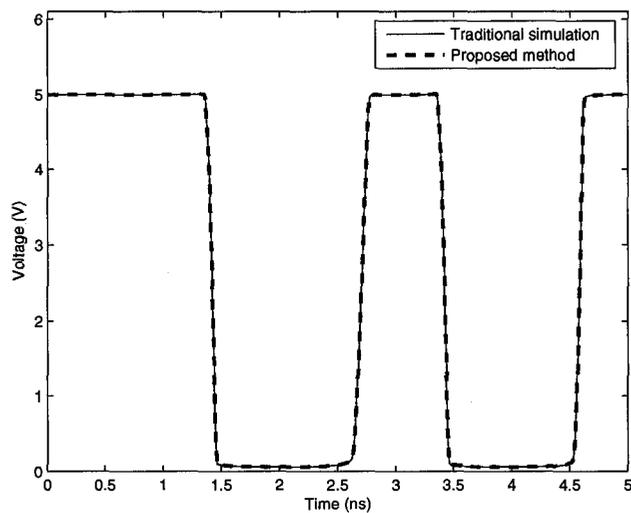
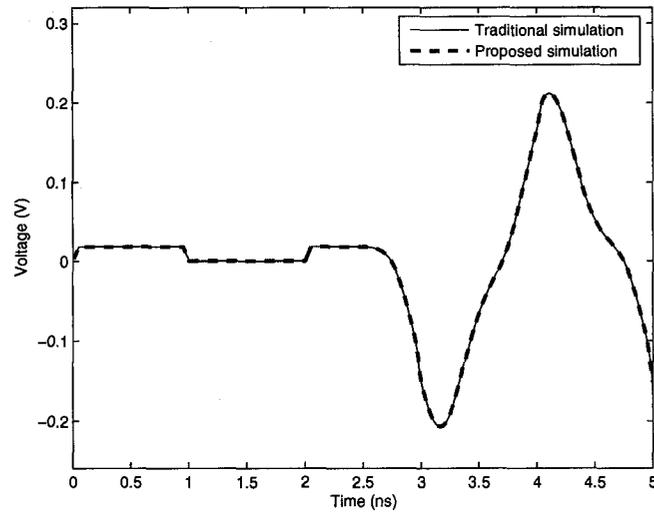
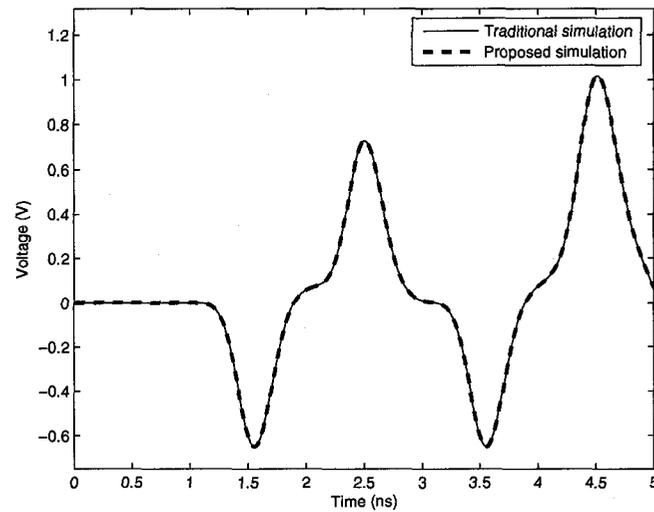
(b) V_{out}

Figure 3.12: Time-domain response of Example 2 ($d = 15\text{cm}$) for far-end of signal line and V_{out}



(a) Near end - first victim line



(b) Far end - last line

Figure 3.13: Time-domain response of Example 2 ($d = 15\text{cm}$) for near-end of first victim line and far-end of last line

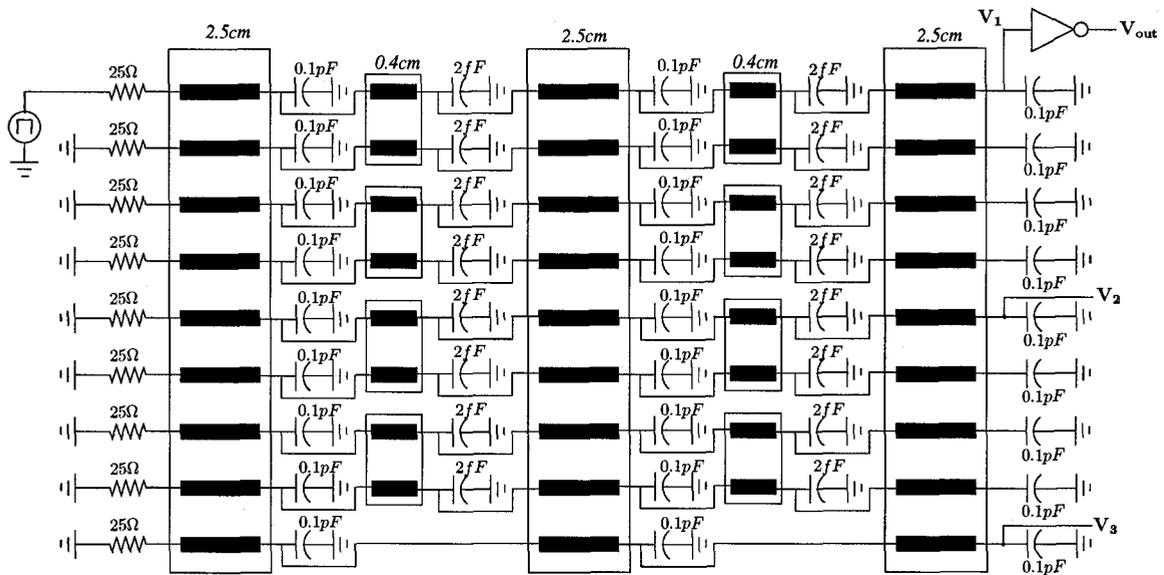
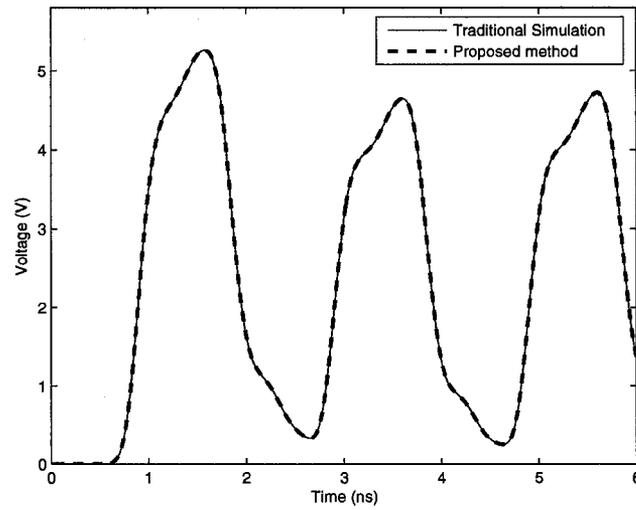
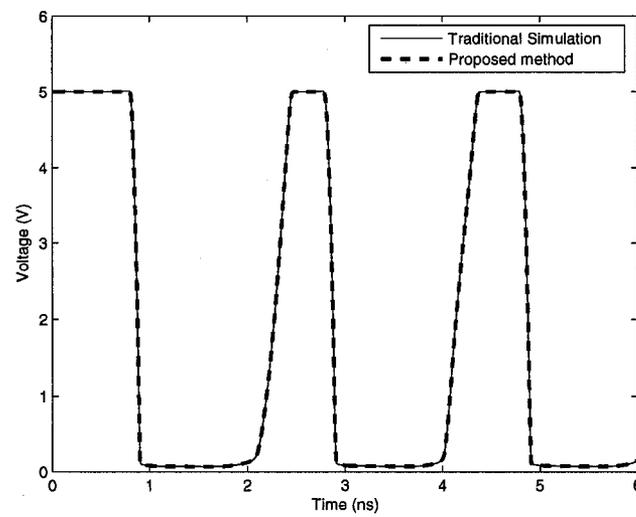
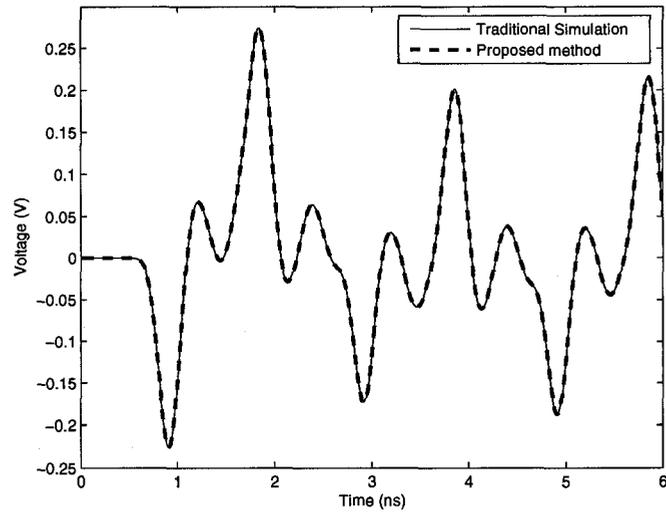
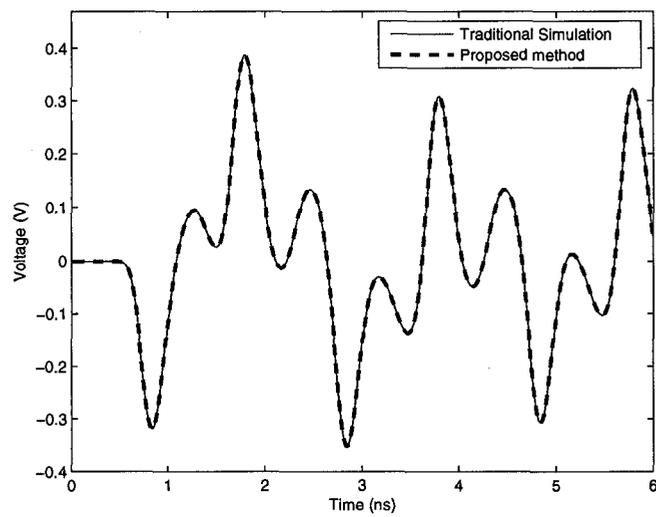


Figure 3.14: Interconnect network comprised two and nine transmission lines for Example 3

an equal length of 0.4 cm and each portion was discretized with 20 lumped segments. The time domain response of a trapezoidal input pulse with 0.05 ns rise/fall time, 0.9 ns pulsewidth, and a 2 ns period are shown in Figs. 3.15–3.16. The computational results are given in Table 3.5. For variable step size, an optimal speed-up of 3.7 was obtained with six domains for a nine transmission line portion and two domains for a two transmission line portion (total of 34 intermediate domains). For constant step size, an optimal speed-up of 6.9 was obtained with a single domain for a nine transmission line portion and a single domain for a two transmission line portion (total of 11 intermediate domains).

(a) V_1 (b) V_{out} Figure 3.15: Time-domain response for Example 3 (V_1 and V_{out})

(a) V_2 (b) V_3 Figure 3.16: Time-domain response for Example 3 (V_2 and V_3)

Step-size	Size	CPU Time		Speed-up
		Traditional	Proposed	
variable	4251	35.71	9.60	3.72
constant	4251	35.26	5.07	6.95

Table 3.5: Computational results for Example 3

Chapter 4

Implementation Techniques for the Proposed Method

The main contribution of the proposed method from Chapter 3 is the reduction in the cost of LU decomposition for the large Jacobian matrix. However, there are inherent computational overheads such as additions, multiplications, multi-column forward/backward substitutions, and solving interface variables present in the domain decomposition method. Particularly, many of these overheads are involved in the computation of Schur complement matrix \mathbf{S} , as given by (3.15). If these operations are performed in a general way (i.e., without exploiting any special properties of the matrices involved), a considerable speed-up cannot be achieved. In this chapter, such efficient techniques for these operations are presented with details in Sec. 4.1. In Sec. 4.2, detail implementation procedure of a non-linear transient simulation algorithm using proposed method is presented using a pseudocode.

4.1 Efficient Computation of Schur Complement Matrix

The Schur complement expression (3.15) is re-expressed as,

$$\mathbf{S} = \mathbf{M}_z - \mathbf{Q}_1 \mathbf{M}_1^{-1} \mathbf{P}_1 - \sum_{i=2}^{n-1} (\mathbf{Q}_i \mathbf{M}_2^{-1} \mathbf{P}_i) - \mathbf{Q}_n \mathbf{M}_n^{-1} \mathbf{P}_n. \quad (4.1)$$

Exploiting the uniformity of transmission line structures allows us to get rid of computations of matrices $\mathbf{Q}_i \mathbf{M}_2^{-1} \mathbf{P}_i$ for $i = 3 \dots n-1$. It only requires one to calculate $\mathbf{Q}_j \mathbf{M}_2^{-1} \mathbf{P}_j$ for $j = 1, 2, n$. This saves many forward/backward substitution and thus, it helps to improve the speed-up. The supporting mathematical details are presented in Sec. 4.1.1.

The non-zero entries of the matrix \mathbf{S} can be divided into block matrices using a specific pattern such that many of these block matrices are always identical to each other. Then \mathbf{S} can be obtained by computing only the fewer unique block matrices. The details are given in Sec. 4.1.2.

4.1.1 Avoiding Computation of Matrices $\mathbf{Q}_i \mathbf{M}_2^{-1} \mathbf{P}_i$

The matrices \mathbf{P}_i represent interaction from the interface variables to the system of equations of domain i . The matrices \mathbf{Q}_i represent interaction from i -th domain interior variables to the system of equations of interface variables. First and last domains couple with only N (number of coupled lines) interface variables, while the intermediate domains couple with $2N$ interface variables. When the node numbering is

Matrices $\mathbf{M}_2^{-1}\mathbf{P}_i$ are obtained by forward/backward substitution of the LU factors of \mathbf{M}_2 with \mathbf{P}_i . The nonzero portions of $\mathbf{M}_2^{-1}\mathbf{P}_i$, which are also banded within $2N$ contiguous columns, can be written in terms of $\mathbf{M}_2^{-1}\mathbf{P}_2$ as,

$$(\mathbf{M}_2^{-1}\mathbf{P}_i)[1, \dots, r_i; a, \dots, b] = (\mathbf{M}_2^{-1}\mathbf{P}_2)[1, \dots, r_i; 1, \dots, 2N]. \quad (4.3)$$

It can be noted that the forward/backward substitutions are required only for the first $2N$ columns of \mathbf{P}_2 matrix.

Now the products $\mathbf{Q}_i\mathbf{M}_2^{-1}\mathbf{P}_i$ can be obtained by calculating only $\mathbf{Q}_2\mathbf{M}_2^{-1}\mathbf{P}_2$ since the non-zero portion (which is a block matrix with a size of $2N \times 2N$) of these matrices are identical. The non-zero block matrix is defined as $\mathbf{\Gamma}$ and can be expressed as,

$$\mathbf{\Gamma} = (\mathbf{Q}_2\mathbf{M}_2^{-1}\mathbf{P}_2)[1, \dots, 2N; 1, \dots, 2N] = (\mathbf{Q}_i\mathbf{M}_i^{-1}\mathbf{P}_i)[a, \dots, b; a, \dots, b] \quad (4.4)$$

where $\mathbf{\Gamma}$ is the only non-zero block found in the matrices $\mathbf{Q}_i\mathbf{M}_i^{-1}\mathbf{P}_i$ for $i = 2 \dots n - 1$.

Shifting both rows and columns of non-zeros of $\mathbf{Q}_2\mathbf{M}_2^{-1}\mathbf{P}_2$ by $N(i - 2)$ gives the product $\mathbf{Q}_i\mathbf{M}_i^{-1}\mathbf{P}_i$ for $i = 3 \dots n - 1$. Fig. 4.2 illustrates the sparsity pattern of the sample matrices $\mathbf{Q}_2\mathbf{M}_2^{-1}\mathbf{P}_2$, $\mathbf{Q}_5\mathbf{M}_5^{-1}\mathbf{P}_5$, and $\mathbf{Q}_9\mathbf{M}_9^{-1}\mathbf{P}_9$ for a five transmission line circuit with 10 domains.

It should be noted that if the uniformity of coupled transmission lines is not exploited, then one has to perform additional $2N \times (n - 3)$ column forward/backward substitutions in order to obtain $\mathbf{M}_i^{-1}\mathbf{P}_i$ for $i = 3 \dots n - 1$. This will consume a significant amount of CPU time.

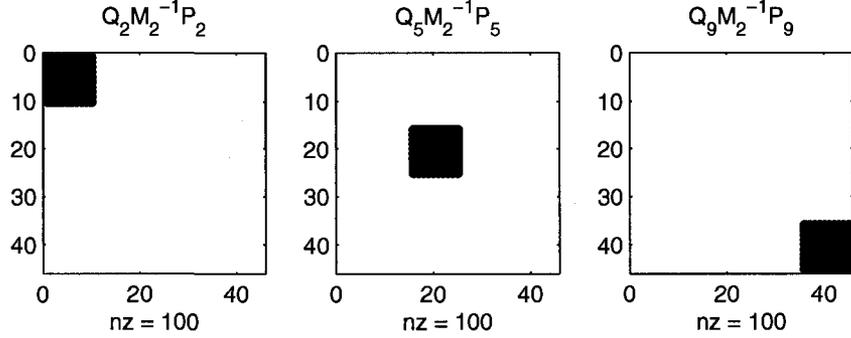


Figure 4.2: Sparsity pattern of $\mathbf{Q}_i \mathbf{M}_2^{-1} \mathbf{P}_i$ matrices ($i = 2, 5, 9$)

4.1.2 Computation of Non-Zero Block Matrices Found in \mathbf{S}

In Schur complement matrix calculation in (4.1), the \mathbf{M}_z represents the coupling between interface variables themselves. Interface variables are found on the boundary of neighboring domains. Let \mathbf{M}_σ be the matrix that represents the coupling relationship between a set of interface nodes on a given boundary for a uniform multi-conductor transmission line. Due to the uniformity of transmission lines, relationship \mathbf{M}_σ is found on all boundaries as given by Fig. 4.3. Therefore, the \mathbf{M}_z matrix, which represents the coupling relationship between all interface variables themselves can be expressed in terms of \mathbf{M}_σ (\mathbf{M}_σ repeats along the diagonal of \mathbf{M}_z) and it is given by

$$\mathbf{M}_z = \begin{bmatrix} \mathbf{M}_\sigma & & \\ & \ddots & \\ & & \mathbf{M}_\sigma \end{bmatrix} \quad (4.5)$$

where,

$$\mathbf{M}_\sigma = \mathbf{M}_z[1 + (i - 1)N, \dots, iN; 1 + (i - 1)N, \dots, iN] \quad i = 1 \dots n - 1.$$

Fig. 4.4 shows the sparsity pattern of \mathbf{M}_z obtained for nine coupled transmission lines with 10 domains.

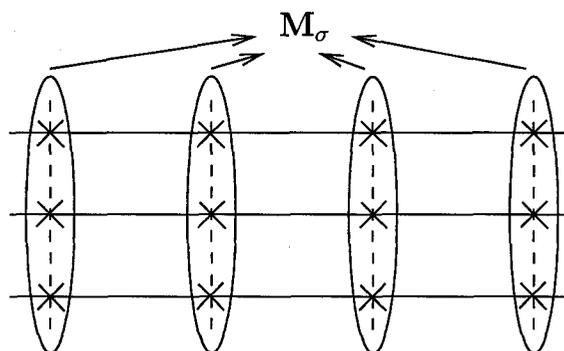


Figure 4.3: Interaction relationship for set of interface nodes lying on any given boundary

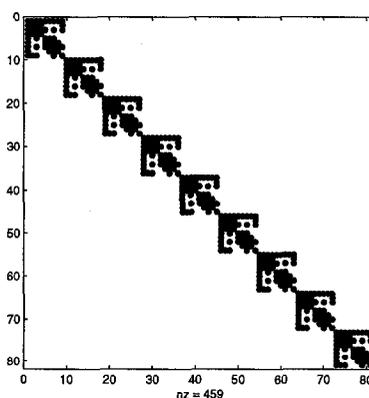


Figure 4.4: Sparsity pattern of \mathbf{M}_z matrix

Using (4.4) and (4.5), (4.1) can be written as,

$$\mathbf{S} = \begin{bmatrix} \boldsymbol{\Psi}_1 & -\boldsymbol{\Gamma}_{12} & & & \\ -\boldsymbol{\Gamma}_{21} & \boldsymbol{\Psi}_2 & -\boldsymbol{\Gamma}_{12} & & \\ & \ddots & \ddots & \ddots & \\ & & -\boldsymbol{\Gamma}_{21} & \boldsymbol{\Psi}_2 & -\boldsymbol{\Gamma}_{12} \\ & & & -\boldsymbol{\Gamma}_{21} & \boldsymbol{\Psi}_3 \end{bmatrix} \quad (4.6)$$

where

$$\begin{bmatrix} \boldsymbol{\Gamma}_{11} & \boldsymbol{\Gamma}_{12} \\ \boldsymbol{\Gamma}_{21} & \boldsymbol{\Gamma}_{22} \end{bmatrix} = \boldsymbol{\Gamma}$$

$$\boldsymbol{\Psi}_1 = \mathbf{M}_\sigma - (\mathbf{Q}_1 \mathbf{M}_1^{-1} \mathbf{P}_1)[1, \dots, N; 1, \dots, N] - \boldsymbol{\Gamma}_{11}$$

$$\boldsymbol{\Psi}_2 = \mathbf{M}_\sigma - \boldsymbol{\Gamma}_{11} - \boldsymbol{\Gamma}_{22}$$

$$\boldsymbol{\Psi}_3 = \mathbf{M}_\sigma - (\mathbf{Q}_n \mathbf{M}_n^{-1} \mathbf{P}_n)[n_\sigma - N + 1, \dots, n_\sigma; n_\sigma - N + 1, \dots, n_\sigma] - \boldsymbol{\Gamma}_{22}$$

$$n_\sigma = \text{number of interface variables} = N(n - 1).$$

This formulation avoids performing additions for all non-zero entries of the matrices involved in (4.1). Only the non-zero entries of $N \times N$ block matrices, $\boldsymbol{\Psi}_1$, $\boldsymbol{\Psi}_2$, $\boldsymbol{\Psi}_3$, $-\boldsymbol{\Gamma}_{12}$, and $-\boldsymbol{\Gamma}_{21}$ need to be found by performing addition; rest of the non-zeros are repeated as in (4.6). This slightly reduces the total number of floating point operations involved in the simulation. Fig. 4.5 illustrates the sparsity pattern of \mathbf{S} such that portions of non-zeros obtained by performing addition and portions of non-zeros repeated according to (4.6) can be identified visually.

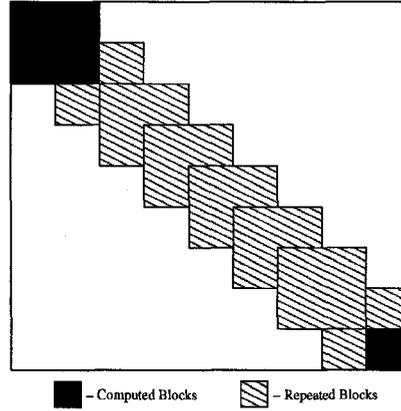


Figure 4.5: Computed and shifted non-zeros of \mathbf{S}

4.2 Implementation Procedure of the Proposed Method

The serial implementation of the proposed method involves the following major steps:

1. Ordering for \mathbf{M}_1 , \mathbf{M}_2 , \mathbf{M}_n and \mathbf{S} .
2. LU decomposition for \mathbf{M}_1 , \mathbf{M}_2 , \mathbf{M}_n
3. Forward/backward substitutions for $\mathbf{M}_j^{-1}\mathbf{P}_j$ for $j = 1, 2, n$
4. Matrix multiplications to obtain $\mathbf{Q}_j\mathbf{M}_j^{-1}\mathbf{P}_j$ for $j = 1, 2, n$
5. Computation of Schur complement matrix \mathbf{S}
6. Computation of $\mathbf{v} = \phi_{n+1} - \mathbf{Q}_1\mathbf{M}_1^{-1}\phi_1 - \sum_{i=2}^{n-1}(\mathbf{Q}_i\mathbf{M}_2^{-1}\phi_i) - \mathbf{Q}_n\mathbf{M}_n^{-1}\phi_n$ (\mathbf{v} - RHS of (3.14))
7. LU decomposition of \mathbf{S}
8. Forward/backward substitution to obtain $\Delta\mathbf{x}_{n+1} = \mathbf{S}^{-1}\mathbf{v}$ (solving interface variables)
9. Solving interior variables, $\Delta\mathbf{x}_i$, using (3.13). However, for solving interior

variables for the domains $3 \dots n-1$, $\mathbf{M}_2^{-1}\mathbf{P}_2$ can be used by shifting its columns appropriately as given by Sec. 4.1.1 to obtain $\mathbf{M}_2^{-1}\mathbf{P}_i$ for $i = 3 \dots n - 1$.

The matrix multiplications or additions in general are performed selectively on corresponding non-zero entries of the matrices. Since the matrices are sparse, these operations consume relatively less amount of CPU time compared to the time spent for LU and FB.

The serial implementation of the non-linear transient simulation using proposed method is given below as a pseudocode.

```

Initialize:  $\mathbf{G}_i, \mathbf{G}_{p_i}, \mathbf{G}_{q_i}, \mathbf{G}_z, \mathbf{C}_i, \mathbf{C}_{p_i}, \mathbf{C}_{q_i}, \mathbf{C}_z$  where  $i = 1, 2, n$ ;
Define global:  $\mathbf{M}_i, \mathbf{P}_i, \mathbf{Q}_i, \mathbf{M}_z, \mathbf{S}, \mathbf{S}_1, \mathbf{S}_n, \mathbf{L}_i, \mathbf{U}_i, \mathbf{R}_i$ ;
Initialize sparsity pattern of  $\mathbf{M}_i, \mathbf{S}$ ;

sparse_ordering( $\mathbf{M}_i$ );
sparse_ordering( $\mathbf{S}$ );

/*..... DC Solution ..... */
 $\mathbf{M}_2 = \mathbf{G}_2$ ;
 $\mathbf{P}_2 = \mathbf{G}_{p_2}$ ;
 $\mathbf{Q}_i = \mathbf{G}_{q_i}$ ;
 $\mathbf{M}_z = \mathbf{G}_z$ ;

prepare_Schur();

Initial guess:  $\mathbf{x}_k = \mathbf{x}_{0k}, k = 1 \dots n + 1$ ;

do /* Newton iteration */
{
   $\mathbf{M}_1 = \mathbf{G}_1 + \frac{\partial \mathbf{F}_1}{\partial \mathbf{x}_1}$ ;
   $\mathbf{M}_n = \mathbf{G}_n + \frac{\partial \mathbf{F}_n}{\partial \mathbf{x}_n}$ ;
   $\mathbf{P}_1 = \mathbf{G}_{p_1} + \frac{\partial \mathbf{F}_1}{\partial \mathbf{x}_{n+1}}$ ;
   $\mathbf{P}_n = \mathbf{G}_{p_n} + \frac{\partial \mathbf{F}_n}{\partial \mathbf{x}_{n+1}}$ ;

  Calculate:  $\phi_k$ ; /* right hand side vectors */

  update_Schur();
   $\Delta \mathbf{x}_k = \text{Solve}()$ ;
   $\mathbf{x}_k = \mathbf{x}_k + \Delta \mathbf{x}_k$ ;
} while(isConverged( $\Delta \mathbf{x}_k, \phi_k(\mathbf{x}_k)$ ) = false)

/*..... Transient Solution ..... */

```

```

h = h0;
t = h;

if (constant_step_size)
{
    /*  $\lambda$  depends on numerical integration */
    M2 = G2 +  $\frac{\lambda}{h}$ C2;
    P2 = GP2 +  $\frac{\lambda}{h}$ CP2;
    Qi = Gqi +  $\frac{\lambda}{h}$ Cqi;
    Mz = Gz +  $\frac{\lambda}{h}$ Cz;

    prepare_Schur();
}

do /* for each time point */
{
    if(variable_step_size) << same as constant step size if loop >>

    do /* Newton iteration */
    {
        M1 = G1 +  $\frac{\lambda}{h}$ C1 +  $\frac{\partial F_1}{\partial x_1}$ ;
        Mn = Gn +  $\frac{\lambda}{h}$ Cn +  $\frac{\partial F_n}{\partial x_n}$ ;
        P1 = GP1 +  $\frac{\lambda}{h}$ CP1 +  $\frac{\partial F_1}{\partial x_{n+1}}$ ;
        Pn = GPn +  $\frac{\lambda}{h}$ CPn +  $\frac{\partial F_n}{\partial x_{n+1}}$ ;

        << same as last four steps of DC Newton iteration code >>

    } while(isConverged( $\Delta x_k$ ,  $\phi_k(x_k)$ ) = false)

    if(variable_step_size) update: h;
    t = t + h;
} while(t  $\leq$  tend)

return;

/* Fill non-zero blocks due to intermediate domains */
function prepare_Schur()
{
    [L2,U2] = LU(M2);

    for i = 1:2N /* 2N column FB substitution */
    {
        R2[:,i] = FB_Substitution(L2, U2, P2[:,i]);
    }

    T = -(Q2[1:2N,:])R2; /* Matrix multiplication */

    /* Break T into sub-matrices as follows */
    TA = T[1:N,1:N];
    TB = T[1:N,N+1:2N];
    TC = T[N+1:2N,1:N];
    TD = T[N+1:2N,N+1:2N];
}

```

```

/* Define */
Mσ = Mz[1:N, 1:N];
S1 = Mσ + TA;
Sn = Mσ + TD;
Tσ = Mσ + TA + TD;

/* Prepare S such that: */
S = 
$$\begin{bmatrix} \mathbf{S}_1 & \mathbf{T}_B & & & \\ & \mathbf{T}_C & \mathbf{T}_\sigma & \ddots & \\ & & \ddots & \ddots & \\ & & & \ddots & \mathbf{S}_n \end{bmatrix}$$

}

/* Fill non-zero blocks due to non-linear domains (first and last) */
function update_S()
{
  [L1, U1] = LU(M1);
  [Ln, Un] = LU(Mn);

  nσ = number_of_interface_variables;

  for i = 1:N /* N column FB substitution */
  {
    R1[:, i] = FB_Substitution(L1, U1, P1[:, i]);
    k = nσ - N + i;
    Rn[:, i] = FB_Substitution(Ln, Un, Pn[:, k]);
  }

  T1 = -(Q1[1:N, :])R1;
  Tn = -(Qn[nσ-N+1:nσ, :])Rn;
  S[1:N, 1:N] = S1 + T1;
  S[nσ-N+1:nσ, nσ-N+1:nσ] = Sn + Tn;
}

function Solve(φk) where k = 1...n+1
{
  nσ = number_of_interface_variables;
  v = φn+1;

  for i = 1:n /* RHS of (3.14) */
  {
    k = 
$$\begin{cases} i & i = 1, n \\ 2 & i = 2 \dots n - 1 \end{cases}$$

    p1 = 
$$\begin{cases} 1 & k = 1, 2 \\ n_\sigma - N + 1 & k = n \end{cases}$$

    p2 = 
$$\begin{cases} p_1 + N - 1 & k = 1, n \\ p_1 + 2N - 1 & k = 2 \end{cases}$$

    vi = FB_substitution(Lk, Uk, φi);
    v = v - (Qk[p1:p2, :])vi;
  }
}

```

```

/* Solve interface variables */
[Ls, Us] = LU(S);
Δxn+1 = FB-substitution(Ls, Us, v);

/* Solve interior variables */
for i = 1:n
{
  << define k and p2 same as in above for loop >>
  p1 = { 1          k = 1
         nσ - N + 1 k = n
         1 + (i - 2)N k = 2

  yi = Rk(Δxn+1[p1:p2]);
  Δxi = vi - yi;
}
return Δxi, for i = 1...n + 1;
}

```

This algorithm can be easily extended to handle interconnect trees comprised of several uniform transmission lines as in Fig. 3.14.

Chapter 5

Estimation of Optimal Number of Domains

The efficiency of the proposed method heavily depends on the number of domains selected. If very few number of domains are selected, it will lead to large sub-domain matrices; hence, the full potential of removing redundancy cannot be realized. On the other hand, having higher number of domains will lead to higher number of interface variables; as number of interface variables are increased, CPU time to solve them (as in (3.14)) will also increase. Fig. 5.1 shows the CPU time for the main operations involved in the proposed method as number of domains are varied. From this plot, it can be easily understood that choosing an appropriate number of domains is essential to obtain optimal results. In this chapter, a practical way to estimate the optimal number of domains for a given number of coupled lines is formulated.

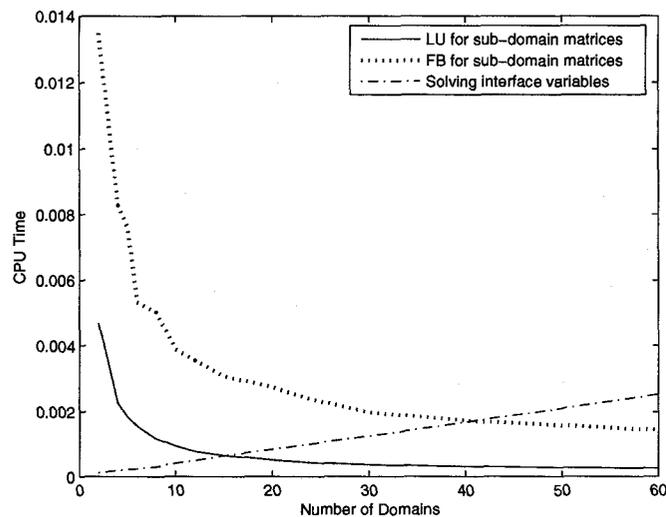


Figure 5.1: Effects of number of domains in the CPU time for DD operations

5.1 Estimation of Optimal Domains for Variable Step Size

For the traditional method, at a specific time point in the transient simulation, at least two Newton iterations need to be performed per time point. Therefore, at least two LU decompositions and two FB substitutions must be performed. For the proposed algorithm, apart from the one time ordering of sub-domain matrices, and matrix additions and multiplications (since these are performed selectively on corresponding non-zero entries), the following main operations need to be performed per time point:

- 3 LU decompositions (for first, second, and last domains)
- N column FB substitutions for the first and last domains, and $2N$ column FB substitutions for the second domain (for (3.15)), where N is the number of

coupled lines.

- FB substitutions for n sub-domain matrices (as in (3.14))
- LU decomposition for \mathbf{S}
- FB substitution to solve the interface variables.

For additional iterations due to the change in non-linear derivatives in the Jacobian matrix, following operations need to be performed:

- 2 LU decompositions for the first and last domain
- N column FB substitutions for the first and last domain (for (3.15))
- FB substitutions for n sub-domain matrices (as in (3.14))
- LU decomposition for \mathbf{S}
- FB substitution to solve the interface variables.

Here the CPU time for LU and FB for the first and last domain can be neglected since sizes of these matrices are very small compared to the size of the second domain. Therefore an expression for approximate speed-up of the proposed method in solving linear system can be given as follows:

$$speed-up \approx \frac{T_{CPU}\{I(LU_{large} + FB_{large})\}}{T_{CPU}\{LU_{small} + 2 \cdot N \cdot FB_{small} + I \cdot \nu \cdot FB_{small} + I(LU_{schur} + FB_{schur})\}} \quad (5.1)$$

where

- I : average number of Newton iterations per time point
- $\nu = n - 2$: number of intermediate domains.

The CPU time of LU and FB can be estimated by formulating a mathematical expression such that the coefficients of the expression can be obtained by performing

a least square fitting on the obtained CPU time data. Here the CPU time for LU decomposition and FB substitution are fitted to the form ζn_l^α and ξn_l^β respectively, where n_l is size of the original matrix. The CPU time of matrix solution was experimented with sparse matrix package UMFPACK 4.6 [62, 63] in a system with AMD 64-bit 2.2 GHz processor, 1 GB RAM, and Linux operating system. The coefficients for the CPU time of LU and FB of the large matrix \mathbf{M} are given in Table 5.1.

Coupled lines (N)	LU		FB	
	ζ	α	ξ	β
1	1.0126×10^{-6}	1.0574	1.0821×10^{-7}	1.0273
2	1.1833×10^{-6}	1.0962	1.9235×10^{-7}	1.0786
3	1.2201×10^{-6}	1.0971	2.0532×10^{-7}	1.0803
4	1.4684×10^{-6}	1.0867	2.1277×10^{-7}	1.0885
5	1.6045×10^{-6}	1.0825	2.2639×10^{-7}	1.0915
6	1.8144×10^{-6}	1.0744	2.4385×10^{-7}	1.0896
7	2.0553×10^{-6}	1.0642	2.7673×10^{-7}	1.0927
8	2.5013×10^{-6}	1.0505	3.0097×10^{-7}	1.0933
9	3.1846×10^{-6}	1.0307	3.3830×10^{-7}	1.0968
10	3.4623×10^{-6}	1.0256	3.6965×10^{-7}	1.1062
12	4.2327×10^{-6}	1.0117	3.9408×10^{-7}	1.1106
15	4.5878×10^{-6}	1.0155	4.8460×10^{-7}	1.1093
20	6.5168×10^{-6}	1.0038	6.2796×10^{-7}	1.1159
25	6.7327×10^{-6}	1.0090	7.1693×10^{-7}	1.1207
30	7.0609×10^{-6}	1.0161	9.9345×10^{-7}	1.1213

Table 5.1: CPU time coefficients of large sparse matrix

Since the sub-domain matrices, \mathbf{M}_i have similar sparsity pattern of the large \mathbf{M} matrix, their CPU coefficients ζ_s and ξ_s are approximately equal to the coefficients of the large matrix ζ and ξ respectively; this approximate equivalence has also been verified experimentally. Therefore, for the sake of simplicity, the CPU time involved in sub-domain matrices can also be estimated using ζn_s^α and ξn_s^β for LU and FB

respectively, where n_s is the number of variables of domain 2.

However Schur complement matrix, \mathbf{S} , is not sparse like the original or sub-domain matrices. As seen in Fig. 4.5, a block of non-zeros are found along the diagonal of \mathbf{S} . The Table 5.2 shows the coefficients of LU and FB time involved in solving interface variable (as in (3.14)) such that the CPU time for LU and FB can be estimated with $\zeta_\sigma n_\sigma^{\alpha_\sigma}$ and $\xi_\sigma n_\sigma^{\beta_\sigma}$ respectively, where n_σ is number of interface variables.

Coupled lines (N)	LU		FB	
	ζ_σ	α_σ	ξ_σ	β_σ
1	1.9697×10^{-6}	1.0010	1.1952×10^{-7}	1.0189
2	2.4973×10^{-6}	1.0132	1.5672×10^{-7}	1.1027
3	2.7937×10^{-6}	1.0184	1.8390×10^{-7}	1.1039
4	2.8836×10^{-6}	1.0201	2.0257×10^{-7}	1.1056
5	2.9265×10^{-6}	1.0254	2.3728×10^{-7}	1.1107
6	2.9939×10^{-6}	1.0369	2.5985×10^{-7}	1.1153
7	3.1473×10^{-6}	1.0451	2.8863×10^{-7}	1.1182
8	3.3990×10^{-6}	1.0508	3.2908×10^{-7}	1.1238
9	3.5528×10^{-6}	1.0617	3.5280×10^{-7}	1.1245
10	3.7509×10^{-6}	1.0710	3.9322×10^{-7}	1.1253
12	3.8708×10^{-6}	1.0839	4.3093×10^{-7}	1.1260
15	4.1942×10^{-6}	1.1027	5.7847×10^{-7}	1.1285
20	4.6245×10^{-6}	1.1170	6.9368×10^{-7}	1.1337
25	5.1096×10^{-6}	1.1262	8.2262×10^{-7}	1.1370
30	5.8714×10^{-6}	1.1287	9.8016×10^{-7}	1.1381

Table 5.2: CPU time coefficients of Schur Complement matrix

Using the coefficients, the approximate speed-up expression given in (5.1), a speed-up function, g , can be formulated as follows

$$g = \frac{I(\zeta n_i^\alpha + \xi n_i^\beta)}{\zeta n_s^\alpha + (2N + I\nu)\xi n_s^\beta + I(\zeta_\sigma n_\sigma^{\alpha_\sigma} + \xi_\sigma n_\sigma^{\beta_\sigma})}. \quad (5.2)$$

For an interconnect discretized with RLGC lumped components, n_l , n_s and n_σ can be expressed (by neglecting the size of MNA variables from driver and receiver for n_l) as follows:

$$n_l \approx 3NK + N \quad (5.3a)$$

$$n_s = \frac{3NK}{\nu} - N \quad (5.3b)$$

$$n_\sigma = N(\nu + 1) \quad (5.3c)$$

where K is the number of lumped segments.

For a specific interconnect problem, N and K are always predetermined values. Therefore, the function g can be considered as a function of number of intermediate domains, ν , if the average number of Newton iterations per time point, I , can be predicted. Then the optimal number of domains can be obtained by solving $g'(\nu) = 0$. However any value of ν is not acceptable since the number of lumped segments, K , must be divisible by ν in order to form identical partitions. Therefore, the feasible optimal number of domains, ν_o , need to satisfy the following conditions.

$$g'(\nu_o) \leq g'(\nu_k), \quad \forall \nu_k \quad (\nu_k \in \mathbb{N}; K | \nu_k; \nu_o \in \nu_k) \quad (5.4)$$

It should be noted that the speed-up discussed in this section is the speed-up in CPU time solely dedicated to the matrix solving time in transient simulation. It does not take into account other CPU consuming factors in the transient simulation such as convergence check, step size determination, etc.

Also, it should be noted that the CPU coefficients listed in the tables will vary from system to system. For example, the CPU proportional constant value, ζ , will decrease if the CPU speed is higher. However the overall estimated speed-up graph for the function $g(\nu)$ will not change drastically from system to system. Therefore, the optimal domains obtained with these coefficients will probably be suitable for any system running the interconnect problem according to the proposed method given that the system will be using UMFPACK 4.6 for matrix solving.

5.1.1 Results

Optimal domains estimated using (5.2) are compared with actual results for 2, 10, and 25 transmission line circuits. Table 5.3 shows the comparison. Here I is set to 2 and $K = 120$. Figs. 5.2 - 5.4 show estimated speed-up and the actual speed-up obtained for these transmission line examples.

N	Exact solution of $g'(\nu) = 0$	Feasible solution (ν_o)	Actual optimal domains
2	15.6833	15	15
10	24.6910	24	24
25	34.5382	30	30

Table 5.3: Comparison of optimal domains

Fig. 5.5 shows estimated speed-up at six different K (number of lumped segments) values for the two transmission line circuit, where I is set to 2. Since $n_l/K > n_s/K$, the speed-up function, $g(\nu)$, increases for higher K values as it can be observed from the plot.

Fig. 5.6 shows the estimated speed-up for the two transmission line circuit at four different I values ranging from 2 to 4.

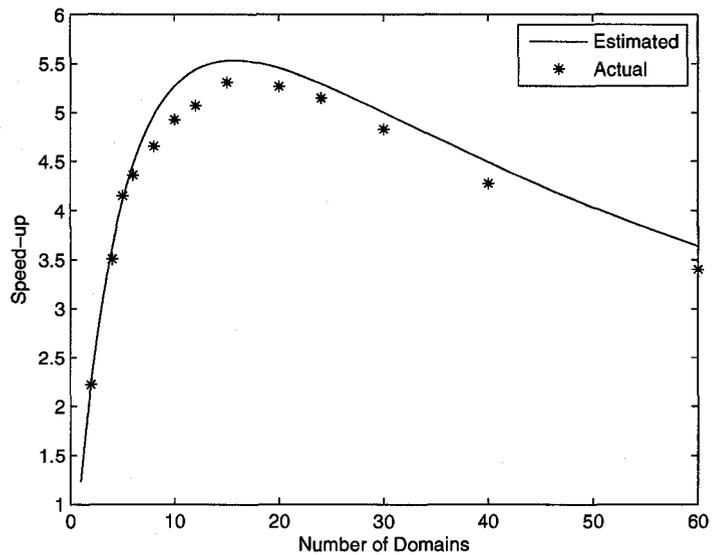


Figure 5.2: Estimated and actual speed-up for two transmission line for the variable step size case

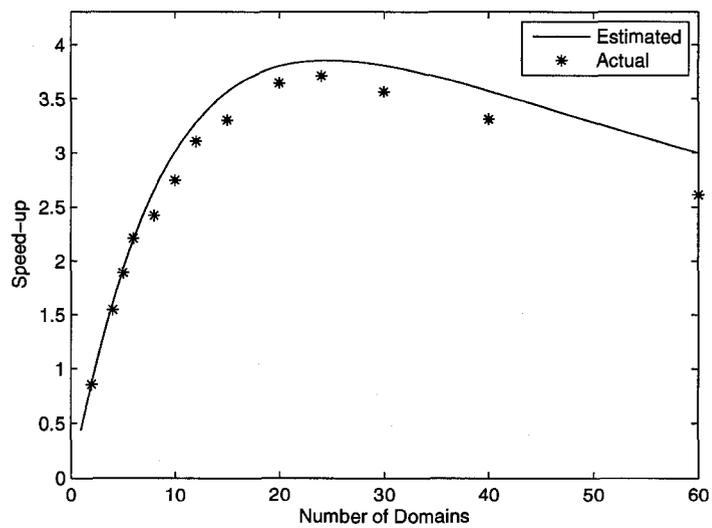


Figure 5.3: Estimated and actual speed-up for 10 transmission line for the variable step size case

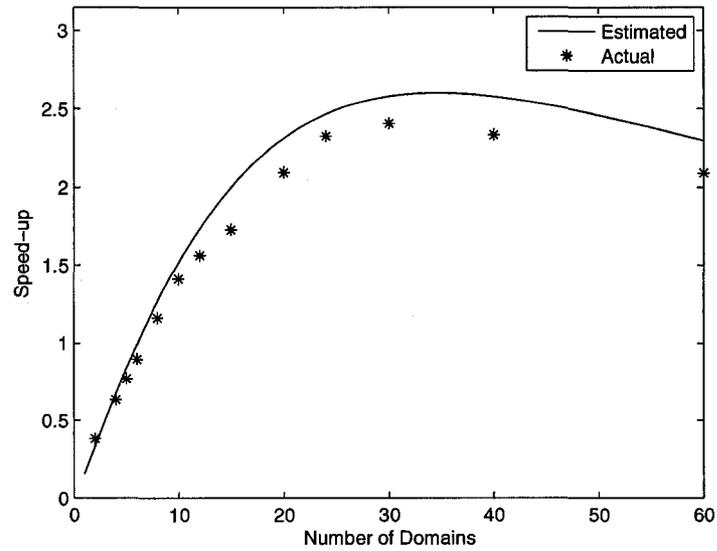


Figure 5.4: Estimated and actual speed-up for 25 transmission line for the variable step size case

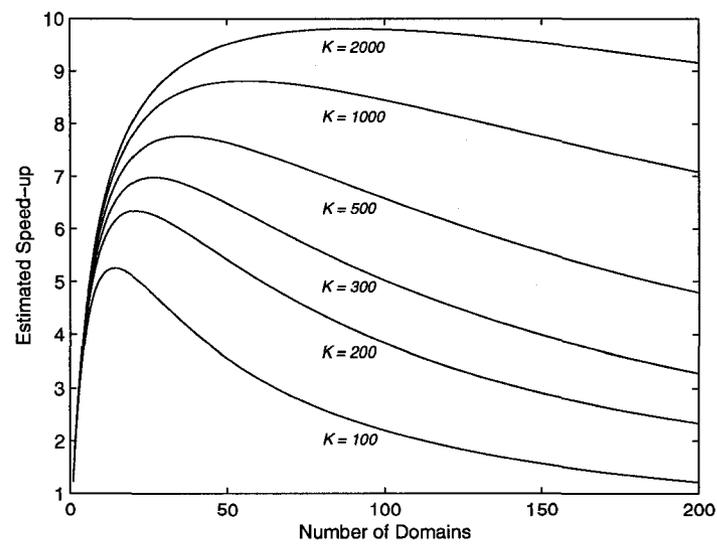


Figure 5.5: Estimated speed-up for two transmission line at six different number of lumped segments (K) for the variable step size case

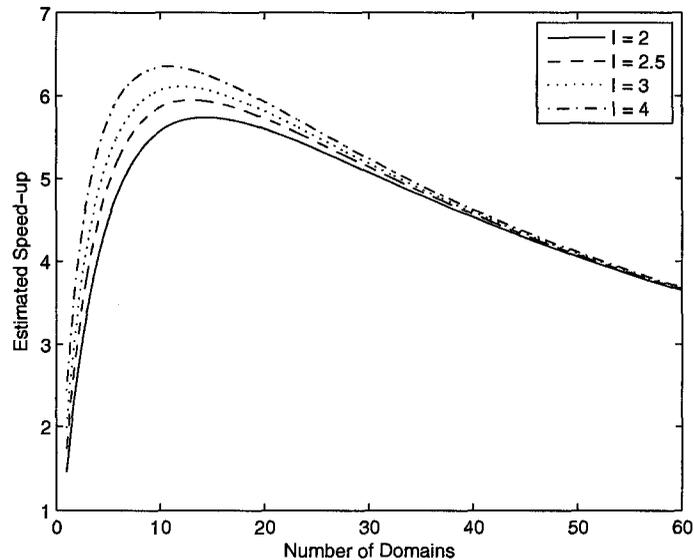


Figure 5.6: Estimated speed-up for two transmission line at four different I values for the variable step size case

5.2 Estimation of Optimal Domains for Constant Step Size

In contrast to the variable step size algorithm, in the constant step size algorithm, LU decomposition and FB substitution involved with M_2 are performed as one time operations. Neglecting the CPU time for these one-time operations, the approximate speed-up expression for the constant step size can be written as,

$$speed - up \approx \frac{T_{CPU}\{LU_{large} + FB_{large}\}}{T_{CPU}\{\nu \cdot FB_{small} + LU_{schur} + FB_{schur}\}}. \quad (5.5)$$

Then the speed-up function, g_c , using the same CPU coefficients listed in Sec. 5.1

can be formulated as

$$g_c = \frac{\zeta n_l^\alpha + \xi n_l^\beta}{\nu \xi n_s^\beta + \zeta_\sigma n_\sigma^{\alpha\sigma} + \xi_\sigma n_\sigma^{\beta\sigma}}. \quad (5.6)$$

The procedure for finding optimal domains is same as that for variable step size case.

5.2.1 Results

Table 5.4 shows the comparison optimal domains for 2, 10, and 25 transmission line circuits.

N	Exact solution of $g'_c(\nu) = 0$	Feasible solution (ν_o)	Actual optimal domains
2	3.2899	3	3
10	5.2378	5	5
25	6.5482	6	6

Table 5.4: Comparison of optimal domains obtained for constant step size

Figs. 5.7 - 5.9 show estimated and the actual speed-up for these transmission line examples with $K = 120$ and $I = 2$. It can be noted from the graphs that unlike the variable step size case, the optimal speed-up is achieved with fewer number of domains. Since the one-time LU decomposition cost of \mathbf{M}_2 is neglected, the interface variable solving becomes an important factor to affect the speed-up. As a result, fewer domains give optimal results.

Fig. 5.10 shows estimated speed-up at six different K values for the two transmission line circuit.

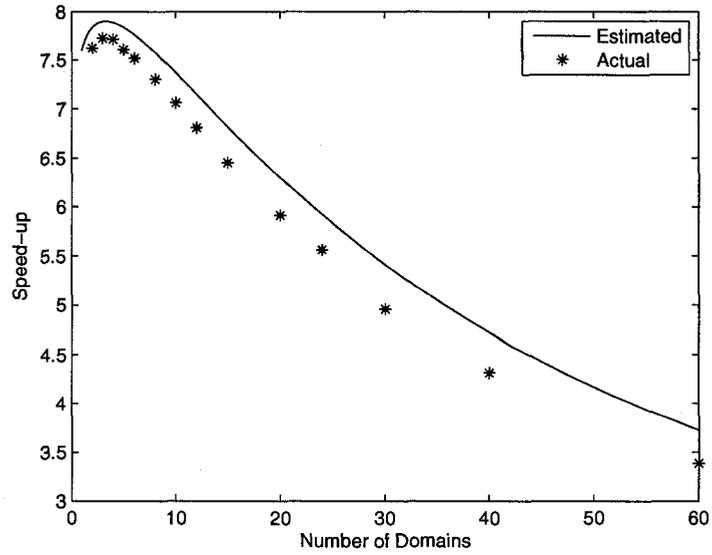


Figure 5.7: Estimated and actual speed-up for two transmission line for the constant step size case

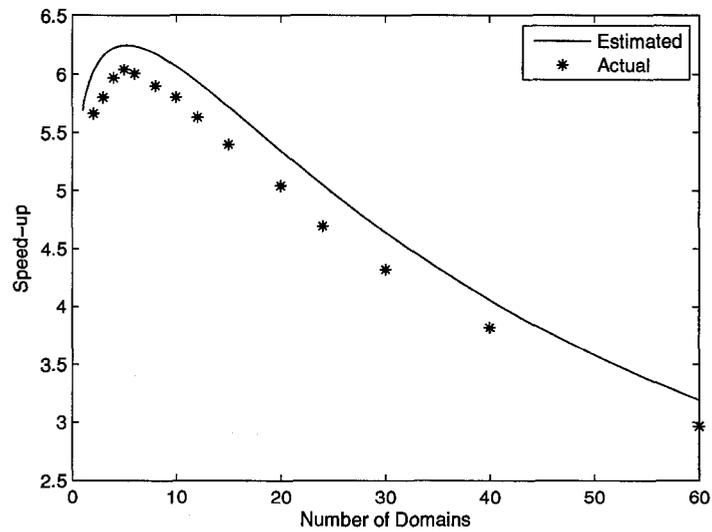


Figure 5.8: Estimated and actual speed-up for 10 transmission line for the constant step size case

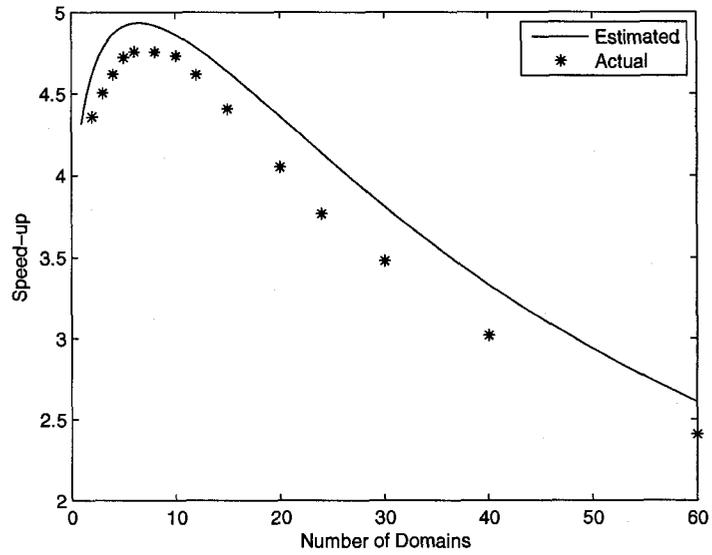


Figure 5.9: Estimated and actual speed-up for 25 transmission line for the constant step size case

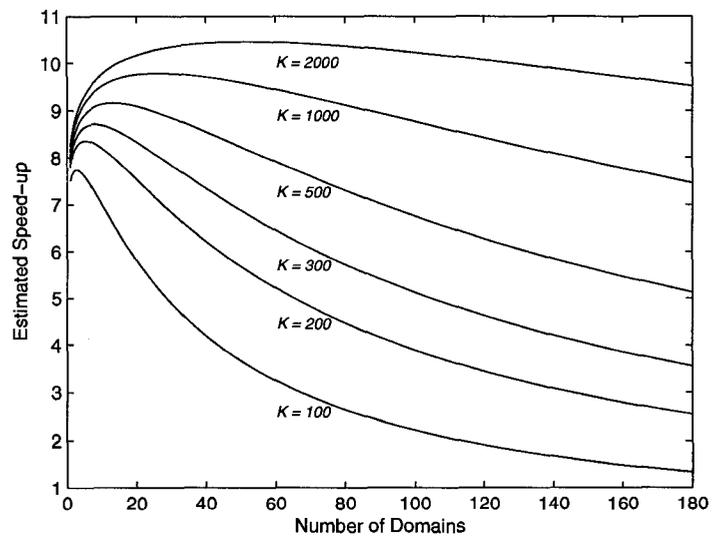


Figure 5.10: Estimated speed-up for two transmission line at six different number of lumped segments (K) for the constant step size case

Chapter 6

Conclusions and Future Work

6.1 Summary

In this Thesis, a novel efficient high-speed interconnect simulation algorithm is presented. Efficiency in simulation was obtained by removing inherent redundancies in interconnect networks through the application of longitudinal partitioning coupled with domain decomposition techniques. Considerable speedup was observed using the proposed method without any loss in numerical accuracy. It was also proved that further speed-up can be obtained by using constant step size throughout the transient simulation. In addition, several implementation techniques were provided to enhance efficiency of the proposed algorithm. Finally, the effects of number of domains on CPU time for the proposed method was discussed and an approach to determine optimal number of domains was presented.

The main advantages of the proposed method can be summarized as follows:

1. Provides all the advantages that of the conventional lumped segmentation method: conservation of passivity, better accuracy, versatility in performing nonlinear transient simulation by incorporating any type of active or passive components to the interconnect analysis, etc.
2. Compared to the conventional method, it provides better efficiency in simulation by removing redundancies in interconnect networks.
3. Further speed-up can be achieved by performing non-linear transient simulation with a constant step size.
4. The algorithm can be easily parallelized since it is based on domain decomposition method.

6.2 Future Work

The following future research directions are proposed:

1. Implementation of parallel version of the algorithm: Since the algorithm is based on domain decomposition method, parallel processing algorithm can be readily implemented. The partitioning approach, which gives lower number of interface variables, will lead to efficient inter-process communication between multiple processors. The parallel processing can handle non-uniform transmission lines as well since each domain can be processed by a separate processor.
2. Simulation of frequency dependent transmission lines: Frequency dependent transmission lines can be discretized using lumped circuit elements by performing circuit synthesis of frequency dependent p.u.l impedance and admittance transfer functions as proposed in literature [65–69]. After it is discretized us-

ing lumped components, the simulation using proposed method is no different from the case of frequency independent p.u.l parameters. In fact, a better speed-up can be expected when compared to the conventional method since the resulting complex network will lead to larger matrices.

3. Other areas of application: In addition to interconnect simulation, the strategy, forming identical partitions and removing redundancies using domain decomposition method, can be applied to several possible areas such as simulation of large power grids discretized with lumped components and thermal networks.

References

- [1] R. Achar and M. Nakhla, "Simulation of high-speed interconnects," in *Proceedings of IEEE*, vol. 89, May 2001, pp. 693–728.
- [2] C. Paul, *Analysis of Multiconductor Transmission Lines*. New York, NY: John Wiley and Sons Inc., 1994.
- [3] H. B. Bakoglu, *Circuits, Interconnects, and Packaging for VLSI*. Reading, MA: Addison-Wesley, 1990.
- [4] A. Deutsch, "Electrical characteristics of interconnects for high-performance systems," in *Proceedings of IEEE*, vol. 86, Feb. 1998, pp. 315–355.
- [5] M. Nakhla and R. Achar, *Handbook on VLSI*. Boca Raton, FL: CRC Press, 2000, ch. Modeling and Simulation of High-Speed Interconnects.
- [6] A. Dounavis, P. Gunupudi, E. Gad, R. Khazaka, R. Griffith, R. Achar, and M. Nakhla, "Addressing transient analysis of multiport distributed interconnects," in *Proceedings of IEEE Topical Meeting on Electrical Performance on Electronic Packaging*, Oct. 2000, pp. 243–245.

- [7] Q. Yu, J. M. L. Wang, and E. S. Kuh, "Passive multipoint moment matching model order reduction algorithm on multiport distributed interconnect networks," *IEEE Transactions on Circuits and Systems*, vol. 46, no. 1, pp. 140–160, Jan. 1999.
- [8] S. Lele, "Compact finite difference schemes with spectral-like resolution," in *Journal of Computational Physics*, vol. 103, 1992, pp. 16–42.
- [9] L. Pileggi and R. Roher, "Asymptotic waveform evaluation for timing analysis," *IEEE Transactions on Computer-Aided Design*, vol. 9, pp. 352–366, Apr. 1990.
- [10] E. Chiprout and M. Nakhla, "Analysis of interconnect networks using complex frequency hopping (CFH)," *IEEE Transactions on Computer-Aided Design*, vol. 14, no. 2, pp. 186–200, Feb. 1995.
- [11] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodelling algorithm," *IEEE Transactions on Computer-Aided Design*, vol. 17, no. 8, pp. 645–653, Aug. 1998.
- [12] P. Li and W. Shi, "Model order reduction of linear networks with massive ports via frequency-dependent port packing," in *Proceedings of IEEE/ACM Design Automation Conference*, July 2006, pp. 267–272.
- [13] B. Smith, P. Bjorstad, and W. Gropp, *Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.
- [14] N. Frohlich, B. Riess, U. Wever, and Q. Zheng, "A new approach for parallel simulation of VLSI circuits on a transistor level," *Circuits and Systems I: Fun-*

- damental Theory and Applications, IEEE Transactions on*, vol. 45, no. 6, pp. 601–613, Jun 1998.
- [15] Q. Zhou, K. Sun, K. Mohanram, and D. C. Sorensen, “Large power grid analysis using domain decomposition,” in *Proceedings of Design, Automation and Test in Europe*, vol. 1, March 2006, pp. 1–6.
- [16] U. Kleis, O. Wallat, U. Wever, and Q. Zheng, “Domain decomposition methods for circuit simulation,” in *ACM SIGSIM Simulation Digest*, vol. 24, July 1994, pp. 183 – 186.
- [17] A. Quarteroni, “Domain decomposition and parallel processing for the numerical solution of partial differential equations,” in *Survey on Mathematics for Industry I*, 1991, pp. 75–118.
- [18] V. Ambalavanar, A. Jerome, and P. Gunupudi, “Removing redundancy in interconnect simulation using domain decomposition techniques,” in *Proceedings of 11th IEEE Workshop on Signal Propagation on Interconnects*, May 2007, pp. 85–88.
- [19] H. W. Jhonson and M. Grahaml, *High-Speed Digital Design*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [20] T. Dhaene and D. Zutter, “Selection of lumped elements models for coupled lossy transmission lines,” *IEEE Transactions on Computer-Aided Design*, vol. 11, pp. 959–967, Jul. 1992.

- [21] S. Y. Kim, N. Gopal, and L. T. Pillage, "Time-domain macromodels for VLSI interconnect analysis," *IEEE Transactions on Computer-Aided Design*, vol. 13, pp. 1257–1270, Oct. 1994.
- [22] A. Cangellaris, A. Pasha, J. Prince, and M. Celik, "A new discrete transmission line model for passive model order reduction and macromodelling of high-speed interconnections," *IEEE Transactions on Advanced Packaging*, vol. 22, pp. 356–364, Aug. 1999.
- [23] R. K. Poon, *Computer Circuits Electrical Design*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [24] A. R. Djordjević and T. K. Sarkar, "Closed-form formulas for frequency dependent resistance and inductance per unit length of microstrip and strip transmission lines," *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, pp. 241–248, Feb. 1994.
- [25] C. Yen, Z. Fazarinc, and R. L. Wheeler, "Time-domain skin-effect model for transient analysis of lossy transmission lines," *Proceedings of IEEE*, pp. 750–757, 1982.
- [26] R. Khazaka, E. Chiprout, M. Nakhla, and Q. J. Zhang, "Analysis of high-speed interconnects with frequency dependent parameters," in *Proc. Intl. Symp. on EMC*, Zurich, Switzerland, Mar. 1995, pp. 203–208.
- [27] I. Lindell, "On the quasi-TEM modes in inhomogeneous multiconductor transmission lines," *IEEE Transactions on Microwave Theory and Techniques*, vol. MTT-29, pp. 812–817, Aug. 1984.

- [28] G. Miano and A. Maffucci, *Transmission Lines and Lumped Circuits*. San Diego, CA: Academic Press, 2001.
- [29] K. Lee, "Two parallel terminated conductors in external fields," *Electromagnetic Compatibility, IEEE Transactions on*, vol. EMC-20, no. 2, pp. 288–296, May 1978.
- [30] A. E. Ruehli, "Equivalent circuit models for three dimensional multiconductor systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 22, pp. 216–224, Mar. 1974.
- [31] H. Heeb and A. Ruehli, "Retarded models for PC board interconnects - or How the speed of light affects your SPICE circuit simulation," in *Proceedings of IEEE International Conference on Computer-Aided Design*, Nov. 1991, pp. 70–73.
- [32] —, "Three-dimensional interconnect analysis using partial element equivalent circuits," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 39, no. 11, pp. 974–982, Nov 1992.
- [33] A. Ruehli and H. Heeb, "Circuit models for three-dimensional geometries including dielectrics," *IEEE Transactions on Microwave Theory and Techniques*, vol. 40, no. 7, pp. 1507–1516, Jul 1992.
- [34] C. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on Circuits and Systems*, vol. CAS-22, pp. 504–509, Jun. 1975.
- [35] A. E. Ruehli, *Circuit Analysis, Simulation and Design*. New York: North-Holland, 1988.

- [36] F.-Y. Chang, "Transient analysis of lossless coupled transmission lines in a non-homogeneous dielectric medium," *IEEE Transactions on Microwave Theory and Techniques*, vol. 18, no. 9, pp. 616–626, Sep 1970.
- [37] ———, "The generalized method of characteristics for waveform relaxation analysis of lossy coupled transmission lines," *IEEE Transactions on Circuits and Systems*, vol. 37, pp. 2028–2038, Dec. 1989.
- [38] ———, "Waveform relaxation analysis of nonuniform lossy transmission lines characterized with frequency-dependent parameters," *IEEE Transactions on Circuits and Systems*, vol. 38, no. 12, pp. 1484–1500, Dec 1991.
- [39] M. Celik, A. Cangellaris, and A. Yaghnour, "An all-purpose transmission-line model for interconnect simulation in SPICE," *IEEE Transactions on Microwave Theory and Techniques*, vol. 45, no. 10, pp. 1857–1867, Oct 1997.
- [40] W. Beyene and J. Schutt-Aine, "Accurate frequency-domain modeling and efficient circuit simulation of high-speed packaging interconnects," *IEEE Transactions on Microwave Theory and Techniques*, vol. 45, no. 10, pp. 1941–1947, Oct 1997.
- [41] A. Dounavis, A. Li, M. Nakhla, and R. Achar, "Passive closed-loop transmission line model for general purpose circuit simulators," *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, pp. 2450–2459, Dec. 1999.
- [42] A. Dounavis, R. Achar, and M. Nakhla, "Efficient passive circuit models for distributed networks with frequency-dependent parameters," *IEEE Transactions on Advanced Packaging*, vol. 23, no. 3, pp. 382–392, Aug 2000.

- [43] A. Dounavis, E. Gad, R. Achar, and M. Nakhla, "Passive model reduction of multiport distributed interconnects," *IEEE Transactions on Microwave Theory and Techniques*, vol. 48, no. 12, pp. 2325–2334, Dec 2000.
- [44] P. Feldmann and R. Freund, "Efficient linear cut analysis by Padé via Lanczos process," *IEEE Transactions on Computer-Aided Design*, vol. 14, pp. 639–649, May 1995.
- [45] M. Celik and A. C. Cangellaris, "Simulation of dispersive multiconductor transmission lines by Padé approximation via Lanczos process," *IEEE Transactions on Microwave Theory and Techniques*, vol. 44, pp. 2525–2535, Dec. 1996.
- [46] N. Nakhla, A. Dounavis, R. Achar, and M. Nakhla, "DEPACT: Delay extraction-based passive compact transmission-line macromodeling algorithm," *IEEE Transactions on Advanced Packaging*, vol. 28, no. 1, pp. 13–23, Feb. 2005.
- [47] A. Dounavis and V. Pothiwala, "Passive closed-form transmission line macromodel using method of characteristics," *IEEE Transactions on Advanced Packaging*, vol. 31, no. 1, pp. 190–202, Feb. 2008.
- [48] N. Nakhla, A. Ruehli, M. Nakhla, and R. Achar, "Simulation of coupled interconnects using waveform relaxation and transverse partitioning," *IEEE Transactions on Advanced Packaging*, vol. 29, no. 1, pp. 78–87, Feb. 2006.
- [49] M. Celik, A. Cangellaris, and A. Yaghnour, "An all-purpose transmission-line model for interconnect simulation in SPICE," *IEEE Transactions on Microwave Theory and Techniques*, vol. 45, no. 10, pp. 1857–1867, Oct 1997.

- [50] A. Dounavis, I. Elfadel, R. Achar, M. Nakhla, A. Ruehli, and H.-M. Huang, "Lossy transmission line passive macromodeling algorithm - three case studies," in *Proceedings of IEEE Topical Meeting on Electrical Performance on Electronic Packaging*, Oct. 2002, pp. 355–358.
- [51] Y.-S. Guo, "Transient simulation of high-speed interconnects based on the semidiscretization of Telegrapher's equations," *IEEE Transactions on Computer-Aided Design*, vol. 21, no. 7, pp. 799–809, Jul 2002.
- [52] M. Celik and A. Cangellaris, "Simulation of multiconductor transmission lines using Krylov subspace order-reduction techniques," *IEEE Transactions on Computer-Aided Design*, vol. 16, no. 5, pp. 485–496, May 1997.
- [53] A. Dounavis, R. Achar, and M. Nakhla, "On passive time-domain macromodels of distributed transmission line networks," *IEEE MTT-S International Microwave Symposium Digest*, vol. 2, pp. 975–978, 2002.
- [54] K. J. Kerns and A. T. Yang, "Preservation of passivity during RLC network reduction via split congruence transformations," *IEEE Transactions on Computer-Aided Design*, vol. 17, pp. 582–591, Jul. 1998.
- [55] R. Achar, P. Gunupudi, and M. Nakhla, "Passive interconnect reduction algorithm for distributed/measured networks," *IEEE Transactions on Circuits and Systems*, vol. 47, no. 4, pp. 287–301, Apr. 2000.
- [56] L. Weinberg, *Network Analysis and Synthesis*. New York: McGraw-Hill, 1962.
- [57] E. Guillemin, *Synthesis of Passive Networks*. New York: Wiley, 1957.

- [58] A. Toselli and O. Widlund, *Domain Decomposition Methods - Algorithms and Theory*. Berlin: Springer, 2005.
- [59] Y. Saad, *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.
- [60] E. Kamgnia, G.-A. Atenekeng-Kahou, B. Philippe, and M. Sosonkina, "A parallel implementation with a comparative study of GMRES(m) preconditioned by multiplicative Schwarz methods," in *Parallel Matrix Algorithms and Applications (PMAA06)*, 2006.
- [61] P. F. Cox, R. G. Burch, D. E. Hocevar, P. Yang, and B. D. Epler, "Direct circuit simulation algorithms for parallel processing," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, June 1991, pp. 714–725.
- [62] T. A. Davis and I. S. Duff, "An unsymmetric-pattern multifrontal method for sparse LU factorization," *SIAM J. Matrix Anal. Appl.*, vol. 18, no. 1, pp. 140–158, 1997.
- [63] T. Davis. UMFPACK version 4.6 user guide. [Online]. Available: <http://www.cise.ufl.edu/research/sparse/umfpack/>
- [64] A. Cangellaris and A. Reuhli, "Model order reduction techniques applied to electromagnetic problems," in *Proceedings of IEEE Topical Meeting on Electrical Performance on Electronic Packaging*, Oct. 2000, pp. 239–242.
- [65] K. Coperich, J. Morsey, V. Okhmatovski, A. Cangellaris, and A. Ruehli, "Systematic development of transmission-line models for interconnects with

- frequency-dependent losses,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 49, no. 10, pp. 1677–1685, Oct 2001.
- [66] S. Pasha, M. Celik, A. Cangellaris, and J. Prince, “Passive SPICE-compatible models of dispersive interconnects,” *Proceedings. 49th Electronic Components and Technology Conference*, pp. 493–499, June 1999.
- [67] S. Mei and Y. Ismail, “Modeling skin and proximity effects with reduced realizable RL circuits,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 4, pp. 437–447, April 2004.
- [68] J. Morsey and A. Cangellaris, “PRIME: Passive realization of interconnect models from measured data,” in *Proceedings of IEEE Topical Meeting on Electrical Performance on Electronic Packaging*, Oct. 2001, pp. 47–50.
- [69] R. Neumayer, F. Haslinger, A. Stelzer, and R. Weigel, “On the synthesis of equivalent circuit models for multiports characterized by frequency-dependent parameters,” *IEEE MTT-S International Microwave Symposium Digest*, vol. 2, pp. 721–724, 2002.