1

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

PRECISIONSM RESOLUTION TARGETS

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ont.)
K1A 0N4

Vous le / votre référence

Quelle / Notre référence

NOTICE

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

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

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

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

AVIS

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

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

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

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

Moment-matching analysis of high-speed VLSI interconnect models

by

Eli Chiprout

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

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

September 16, 1994

© Eli Chiprout 1994

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

THE AUTHOR HAS GRANTED AN
IRREVOCABLE NON-EXCLUSIVE
LICENCE ALLOWING THE NATIONAL
LIBRARY OF CANADA TO
REPRODUCE, LOAN, DISTRIBUTE OR
SELL COPIES OF HIS/HER THESIS BY
ANY MEANS AND IN ANY FORM OR
FORMAT, MAKING THIS THESIS
AVAILABLE TO INTERESTED
PERSONS.

L'AUTEUR A ACCORDE UNE LICENCE
IRREVOCABLE ET NON EXCLUSIVE
PERMETTANT A LA BIBLIOTHEQUE
NATIONALE DU CANADA DE
REPRODUIRE, PRETER, DISTRIBUER
OU VENDRE DES COPIES DE SA
THESE DE QUELQUE MANIERE ET
SOUS QUELQUE FORME QUE CE SOIT
POUR METTRE DES EXEMPLAIRES DE
CETTE THESE A LA DISPOSITION DES
PERSONNE INTERESSEES.

THE AUTHOR RETAINS OWNERSHIP
OF THE COPYRIGHT IN HIS/HER
THESIS. NEITHER THE THESIS NOR
SUBSTANTIAL EXTRACTS FROM IT
MAY BE PRINTED OR OTHERWISE
REPRODUCED WITHOUT HIS/HER
PERMISSION.

L'AUTEUR CONSERVE LA PROPRIETE
DU DROIT D'AUTEUR QUI PROTEGE
SA THESE. NI LA THESE NI DES
EXTRAITS SUBSTANTIELS DE CELLE-
CI NE DOIVENT ETRE IMPRIMES OU
AUTREMENT REPRODUITS SANS SON
AUTORISATION.

ISBN 0-315-98446-5

Canada

Name Eli Chiprout

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

Electrical Engineering (Electronics)
SUBJECT TERM

0544 U.M.I.
SUBJECT CODE

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS

Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION

General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523

Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS

Language
General 0679
Ancient 0289
Linguistics 0290
Modern 0291
Literature
General 0401
Classical 0294
Comparative 0295
Medieval 0297
Modern 0298
African 0316
American 0591
Asian 0305
Canadian (English) 0352
Canadian (French) 0355
English 0593
Germanic 0311
Latin American 0312
Middle Eastern 0315
Romance 0313
Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY

Philosophy 0422
Religion
General 0318
Biblical Studies 0321
Clergy 0319
History of 0320
Philosophy of 0322
Theology 0469

SOCIAL SCIENCES

American Studies 0323
Anthropology
Archaeology 0324
Cultural 0326
Physical 0327
Business Administration
General 0310
Accounting 0272
Banking 0770
Management 0454
Marketing 0338
Canadian Studies 0385
Economics
General 0501
Agricultural 0503
Commerce-Business 0505
Finance 0508
History 0509
Labor 0510
Theory 0511
Folklore 0358
Geography 0366
Gerontology 0351
History
General 0578

Ancient 0579
Medieval 0581
Modern 0582
Black 0328
African 0331
Asia, Australia and Oceania 0332
Canadian 0334
European 0335
Latin American 0336
Middle Eastern 0333
United States 0337
History of Science 0585
Law 0398
Political Science
General 0615
International Law and Relations 0616
Public Administration 0617
Recreation 0814
Social Work 0452
Sociology
General 0626
Criminology and Penology 0627
Demography 0938
Ethnic and Racial Studies 0631
Individual and Family Studies 0628
Industrial and Labor Relations 0629
Public and Social Welfare 0630
Social Structure and Development 0700
Theory and Methods 0344
Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES

Agriculture
General 0473
Agronomy 0285
Animal Culture and Nutrition 0475
Animal Pathology 0476
Food Science and Technology 0359
Forestry and Wildlife 0478
Plant Culture 0479
Plant Pathology 0480
Plant Physiology 0817
Range Management 0777
Wood Technology 0746
Biology
General 0306
Anatomy 0287
Biostatistics 0308
Botany 0309
Cell 0379
Ecology 0329
Entomology 0353
Genetics 0369
Limnology 0793
Microbiology 0410
Molecular 0307
Neuroscience 0317
Oceanography 0416
Physiology 0433
Radiation 0821
Veterinary Science 0778
Zoology 0472
Biophysics
General 0786
Medical 0760

EARTH SCIENCES

Biogeochemistry 0425
Geochemistry 0996

Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES

Environmental Sciences 0768
Health Sciences
General 0566
Audiology 0300
Chemotherapy 0992
Dentistry 0567
Education 0350
Hospital Management 0769
Human Development 0758
Immunology 0982
Medicine and Surgery 0564
Mental Health 0347
Nursing 0569
Nutrition 0570
Obstetrics and Gynecology 0380
Occupational Health and Therapy 0354
Ophthalmology 0381
Pathology 0571
Pharmacology 0419
Pharmacy 0572
Physical Therapy 0382
Public Health 0573
Radiology 0574
Recreation 0575

Speech Pathology 0460
Toxicology 0383
Home Economics 0386

PHYSICAL SCIENCES

Pure Sciences

Chemistry
General 0485
Agricultural 0749
Analytical 0486
Biochemistry 0487
Inorganic 0488
Nuclear 0738
Organic 0490
Pharmaceutical 0491
Physical 0494
Polymer 0495
Radiation 0754
Mathematics 0405
Physics
General 0605
Acoustics 0986
Astronomy and Astrophysics 0606
Atmospheric Science 0608
Atomic 0748
Electronics and Electricity 0607
Elementary Particles and High Energy 0798
Fluid and Plasma 0759
Molecular 0609
Nuclear 0610
Optics 0752
Radiation 0756
Solid State 0611
Statistics 0463

Applied Sciences

Applied Mechanics 0346
Computer Science 0984

Engineering

General 0537
Aerospace 0538
Agricultural 0539
Automotive 0540
Biomedical 0541
Chemical 0542
Civil 0543
Electronics and Electrical 0544
Heat and Thermodynamics 0348
Hydraulic 0545
Industrial 0546
Marine 0547
Materials Science 0794
Mechanical 0548
Metallurgy 0743
Mining 0551
Nuclear 0552
Packaging 0549
Petroleum 0765
Sanitary and Municipal 0554
System Science 0790
Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY

General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

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

Moment-matching analysis of high-speed VLSI interconnect models

submitted by Eli Chiprout
in partial fulfilment of the requirements for
the degree of Doctor of Philosophy.

Thesis Supervisor

Thesis Co-Supervisor

Chairman,
Department of Electronics

Ottawa-Carleton Institute for Electrical Engineering,
Department of Electronics
Faculty of Engineering
Carleton University
September 16, 1994

Abstract

Increasing operating speeds and decreasing feature sizes of present-day VLSI systems have highlighted previously negligible effects of the physical interconnections between active devices, both on and off-chip. As signal frequencies get into the hundreds of MHz range and as VLSI device fabrication sizes enter the deep submicron, the effects of delay, distortion, ringing and crosstalk become the dominant factors limiting the overall performance of VLSI systems. Moment-matching techniques which rely on Padé rational approximations, such as asymptotic waveform evaluation (AWE), have been introduced to enable the rapid simulation of large RLC and transmission line structures within the context of general circuit simulation. However, while these methods may be fast, there remain critical questions and problems relating to their accuracy.

With a novel multipoint technique known as complex frequency hopping (CFH) which is presented here, many of the accuracy concerns of moment-matching are addressed while its CPU advantage is maintained. CFH enables the generation of the accurate dominant eigenvalues of lumped and even distributed networks. This method, combined with a new way of generating the moments for a transmission line network, provides for quick waveform estimation needed early in the design cycle, as well as a more accurate but somewhat slower simulation used for final design verification.

Various techniques using moment-matching analysis are discussed and it is shown how the new methods are merged with them. An overview of the accuracy limitations and the applicability of using Padé approximations for linear analysis is also presented.

Acknowledgements

This thesis would not have been possible without the support, encouragement and contributions of many people. I would like to especially thank my thesis supervisor Prof. Michel S. Nakhla for many long hours of discussion and debate about algorithms and many other interesting subjects. His friendly manner, helpful suggestions and general encouragement was always refreshing and welcome. I am very much in debt to him.

Prof. Q. J. Zhang at Carleton University contributed continuously with helpful comments, ideas and suggestions. His critical assessment of the strengths and weaknesses of the methods I proposed were useful, necessary and always appreciated. I wish to also thank Prof. Ron Rohrer of Carnegie Mellon University, Pittsburgh PA, for his interest in the work and for providing pre-prints to his publications.

I appreciated the contributions of my colleagues Richard Griffith, Sanjay Manney, Nelson Liu, Dong Xie which enabled me to give a better background overview of AWE and its methods. The office staff of Guy Asselin, Barbara Lynn and Dennis Piamonte were always there to assist any time I needed some rush job to be done. Their friendly cheer was welcome, as were the occasional parties, chocolates, and balloon rides.

I leave a special note of appreciation for my wife Carol. Without her continued loving support, encouragement and good humor, working on this thesis would have been much more difficult if not impossible. As it was, I had a lot of fun. I am also especially grateful to my parents for years of patient backing in all things. I hope that their patience has finally paid off.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Contributions	6
1.3	Overview	8
1.3.1	Moment matching	8
1.3.2	RC trees	11
1.3.3	Transmission lines	14
1.3.4	3-D analysis	15
1.3.5	Nonlinear terminations	16
1.4	Organization of the Thesis	16
2	Asymptotic Waveform Evaluation	18
2.1	State space formulation	18
2.1.1	System moments	21
2.1.2	Multiple inputs/outputs and initial conditions	22
2.2	Generating an approximate response	23
2.2.1	Poles and residues	24
2.3	MNA formulation	27
2.4	Examples	31
3	Transmission Lines	44

3.1	Linear subnetwork formulation	45
3.2	Moments of distributed networks	46
3.3	Uniform lossy coupled transmission lines	47
3.3.1	Eigenvalue moment method	49
3.3.2	Matrix exponential moment method	56
3.4	Nonuniform transmission lines	63
3.5	Miscellaneous combinations	69
4	Padé Approximations	70
4.1	The rational approximation	71
4.2	Accuracy	74
4.2.1	Properties	76
4.2.2	Error analysis	81
4.3	Laurent expansion about ∞	85
5	Accuracy Improvements	88
5.1	Higher precision programming	88
5.2	Moment scaling	89
5.3	Padé table analysis	90
5.4	Optimal pole selection	94
5.5	Partitioning	95
5.6	Method of characteristics	99
5.7	Frequency shifting	104
6	Complex Frequency Hopping	107
6.1	Complex expansions	109
6.2	Pole convergence	110
6.3	Pole selection	111
6.3.1	<u>Pole selection algorithm</u>	113

6.4	Search strategy	114
6.4.1	<u>Binary search algorithm</u>	116
6.5	Extra CPU requirements	117
7	Nonlinear Analysis	133
7.1	Notation	134
7.2	Differential equations (SPICE) approach	138
7.2.1	Polynomial expression	139
7.2.2	Pole/residue expression	142
7.3	Recursive convolution approach	145
8	Other Applications	157
8.1	Eigenvalue calculation	157
8.2	PEEC simulation	158
8.2.1	PEEC and rPEEC	160
8.3	Resonant frequencies	164
9	Conclusion	167
A	Transmission Line Equations	171
A.1	Partial differential equation description	171

List of Figures

1.1	PQFP copper ground plate with transmission line leads.	12
1.2	RL mesh with mutual inductive couplings and transmission lines extracted from a 3-D field solver applied to the ground plate.	12
1.3	PQFP ground bounce simulation.	13
2.1	Simple state space network.	19
2.2	RC ladder network.	31
2.3	RC network transient response: 1st order approximation.	32
2.4	RC network transient response: 2nd order approximation.	32
2.5	RC network transient response: 3rd order approximation.	33
2.6	RC network frequency response: 1st order approximation.	33
2.7	RC network frequency response: 2nd order approximation.	34
2.8	RC network frequency response: 3rd order approximation.	34
2.9	RLC network.	36
2.10	RLC network transient response: 2nd order approximation.	39
2.11	RLC network transient response: 3rd order approximation.	39
2.12	RLC network transient response: 4th order approximation.	40
2.13	RLC network transient response: 5th order approximation.	40
2.14	RLC network frequency response: 2nd order approximation.	41
2.15	RLC network frequency response: 3rd order approximation.	41
2.16	RLC network frequency response: 4th order approximation.	42

2.17	RLC' network frequency response: 5th order approximation.	42
2.18	RLC' network frequency response: 6th order approximation.	43
3.1	Coupled transmission line system.	48
3.2	Small interconnect network with single lossless lines.	54
3.3	Transient response of Fig. 3.2.	54
3.4	Large transmission line interconnect network.	55
3.5	Transient response of Fig. 3.4.	55
3.6	Lossy coupled transmission line network.	57
3.7	Transient response of Fig. 3.6.	57
3.8	The matrix exponential recursion algorithm.	61
3.9	Moment comparison of a simple transmission line network.	62
3.10	Prototype chip interconnect: (a) Top view of metallization (b) Cross section (c) Circuit configuration (d) Driving voltage.	65
3.11	Solution of Fig. 3.10 at active line.	67
3.12	Solution of Fig. 3.10 at quiescent line.	67
3.13	Interconnect model with 7 nonuniform transmission lines.	68
3.14	Transient response of Fig. 3.13.	68
5.1	General partitioned network.	96
6.1	Pole selection algorithm using two hops.	113
6.2	Binary search and CFH search area in the complex s plane.	115
6.3	Example circuit proposed by Chang.	120
6.4	Pole distribution (x) of Fig. 6.3. Approximate poles obtained at the origin indicated by (o).	120
6.5	Transient response of Fig. 6.3.	121
6.6	Frequency response of Fig. 6.3.	121
6.7	A lossy coupled transmission line network.	122
6.8	Poles of Fig. 6.7.	123

6.9	Transient response of Fig. 6.7 at V_1 (7 poles).	124
6.10	Transient response of Fig. 6.7 at V_1 (10 poles).	124
6.11	Transient response of Fig. 6.7 at V_1 (22 poles).	124
6.12	Transient response of Fig. 6.7 at V_2 (7 poles).	125
6.13	Transient response of Fig. 6.7 at V_2 (22 poles).	125
6.14	Transient response of Fig. 6.7 at V_2 (30 poles).	125
6.15	Transient response of CFH (Example 3).	127
6.16	Frequency response of CFH (Example 3).	127
6.17	PEEC Frequency domain CFH analysis using $f_{max}=1$ GHz.	130
6.18	PEEC Frequency domain CFH analysis using $f_{max}=4.8$ GHz.	130
6.19	Cooke/Prince/Cangelaris coupled line example.	131
6.20	Transient simulation of Fig. 6.19 with $f_{max}=1$ GHz.	132
6.21	Transient simulation of Fig. 6.19 with $f_{max}=5$ GHz.	132
7.1	Nonlinear/linear partition.	134
7.2	Example nonlinear network and formulation.	136
7.3	State space impedance model of a single pole.	142
7.4	Circuit representation of Fig. 7.3.	143
7.5	State space impedance model of a complex pole using real valued gains.	143
7.6	State space model of the first row of the Z-parameter macro model of (7.11).	144
7.7	Lossless transmission line network with nonlinear drivers/receivers.	148
7.8	Transient response of Fig. 7.7 at V_1 .	148
7.9	Transient response of Fig. 7.7 at V_2 .	149
7.10	Transient response of Fig. 7.7 at V_4 .	149
7.11	Large multi-chip interconnect network with nonlinear termination.	150
7.12	Coupled lossy line network with nonlinear terminations.	152
7.13	Transient response of Fig. 7.12 at V_1 .	153
7.14	Transient response of Fig. 7.12 at V_2 .	153

7.15	Transient response of Fig. 7.12 at V_3	154
7.16	Transient response of Fig. 7.12 at V_4	154
7.17	The dominant poles of the network at two clock pins as generated by CFH, 'x' for one output, 'o' for the other	155
7.18	A clock tree network driven by 2 CMOS inverters.	155
7.19	Transient response of Fig. 7.18.	156
8.1	The PEEC voltage source model of two potential nodes.	161
8.2	CFH frequency analysis of a large rPEEC network.	166
8.3	CFH transient analysis of a large rPEEC network.	166
A.1	Single transmission line.	171
A.2	Lumped equivalent of single transmission line.	172

Notation

$\mathbf{A}$	Matrix $\mathbf{A}$
$\mathbf{A}^T$	Matrix $\mathbf{A}$ transposed
$\mathbf{A}^{(r)}$	r -th derivative of $\mathbf{A}$
$ \mathbf{A} $	Determinant value of matrix $\mathbf{A}$
$\mathbf{0}$	Matrix of zeros
$\mathbf{1}$	Unity Matrix
$\mathbb{R}^N$	The set of all real vectors of size N
$\mathbb{R}^{N \times N}$	The set of all real matrices of size $N \times N$
$\mathbb{R}^N \rightarrow \mathbb{R}^N$	The set of functions mapping $\mathbb{R}^N$ to $\mathbb{R}^N$
$[d_{i,j}]$	Matrix composed of elements $d_{i,j}$ in i -th row and j -th column
$\delta(t)$	Unit dirac delta function
$u(t)$	Unit step function
$\hat{a}$	Approximation of a
s	Laplace operator ($= j\omega$)
$\mathbf{M}_n$	Vector of n -th moments of a network
m_n	n -th moment of an output
$\mathbf{m}_h$	Hankel moment matrix
$\mathbf{m}_v$	Pade moment vector
$\mathbf{b}_v$	Vector of unknown denominator coefficients
$H(s)$	Transfer function of an output
$h(t)$	Impulse response of an output
p_i	i -th network pole
k_i	i -th network residue
$\mathbf{G}_\phi$	MNA matrix $\mathbf{G}$ matrix for linear sub-network ϕ
$\mathbf{C}_\phi$	MNA matrix $\mathbf{C}$ matrix for linear sub-network ϕ
$\mathbf{Y}_\phi(s)$	MNA matrix for linear sub-network ϕ
$\mathbf{V}_\phi(s)$	Vector of unknown variables in MNA formulation of linear sub-network ϕ

$\mathbf{E}_\phi$	Source vector for MNA formulation of linear sub-network ϕ
$\mathbf{R}$	Transmission line resistance per-unit-length matrix
$\mathbf{G}$	Transmission line conductance per-unit-length matrix
$\mathbf{L}$	Transmission line inductance per-unit-length matrix
$\mathbf{C}$	Transmission line capacitance per-unit-length matrix
$\mathbf{Z}_p$	Transmission line impedance per-unit-length matrix
$\mathbf{Y}_p$	Transmission line admittance per-unit-length matrix
γ_n	n -th transmission line eigenvalue
$\underline{\mathbf{S}}_n$	n -th transmission line eigenvector
$\mathbf{S}_n$	Matrix of all transmission line eigenvectors = $[\underline{\mathbf{S}}_n]$
ϵ	Moment skew

Abbreviations

AWE	Asymptotic waveform evaluation
CFH	Complex frequency hopping
EM	Electromagnetic
FFT	Fast fourier transform
IFFT	Inverse fast fourier transform
MCM	Multi-chip module
MNA	Modified nodal admittance
PCB	Printed circuit board
(r)PEEC	(retarded) Partial element equivalent circuit
RC	Resistor-capacitor (networks)
RLC	Resistor-inductor-capacitor (networks)
STM	State transition matrix
ps	pico-seconds
ns	nano-seconds

Chapter 1

Introduction

1.1 Motivation

Analysis and design of interconnections in high-speed VLSI chips, Multi-chip modules (MCM's), printed circuit boards (PCB's), backplanes and in system level components is gaining importance due to the rapid increase in operating frequencies and decrease in feature size. Physical interconnects represent a major performance-limiting factor of these components as the previously negligible effects of signal retardation, distortion, and crosstalk have become highlighted with faster switching speeds increasingly used for large data flow. The interconnected blocks could be analog, digital, or mixed. The physical interconnect structures can be diverse (pins, connectors, leads, microstrips, striplines, etc.) and present at any of the hierarchical packaging levels: integrated circuits, printed circuit boards, multi-chip modules or system backplanes.

With these higher signal speeds, the electrical length of interconnects becomes a significant portion of the wavelength. This gives rise to signal distorting effects that do not exist at lower frequencies. The immediate effects in digital circuits can be undetected pulses on the active lines or false switching at quiescent lines due to coupling effects between adjacent lines. In both analog and digital circuits, a loss of signal integrity may result. While at low frequencies interconnect circuits can be

treated as simple shorts, at high enough frequencies, simulation of interconnect effects in conjunction with other analog simulation has become a necessity.

The intense drive for clean signals (signal integrity) has been at the forefront of new developments in CAD algorithms. Thousands of engineers, intent on achieving the best design possible, use the SPICE[Nag75] circuit simulator on a daily basis for analog simulation and general circuit analysis. But the effects of interconnects are not always handled appropriately by the present levels of SPICE. Interconnect networks are generally modeled as linear networks. These networks can sometimes consist of hundreds or thousands of components such as resistors, capacitors, inductors and/or lossy, coupled transmission lines. At the terminations there generally exist some nonlinear elements such as drivers or receivers. If only lumped RLC models are used for the linear elements, ordinary differential equation solvers such as SPICE or ASTAP[Wee73] may be used for simulation purposes. The CPU cost may be large however, because SPICE or ASTAP are mainly nonlinear differential equation solvers. Large linear RLC networks are not handled efficiently in these simulators. There is also the fact that numerical integration methods used depend on the complicating factor of time step size.

Additionally, ordinary differential equation solvers do not handle more complicated distributed transmission line interconnect models. Distributed models are described by partial differential equations rather than by ordinary differential equations. The equations are made more complex by the introduction of nonlinear terminations to the linear interconnect networks. Another method is therefore needed for the simulation of large interconnect structures in conjunction with all of the other components that normally make up an analog or digital network.

Padé approximation moment-matching techniques such as asymptotic waveform evaluation (AWE) have recently proven useful in the analysis of interconnect structures and various networks containing large linear structures with nonlinear terminations. Previously, all that was available to the designer was a full SPICE simulation

or a quick but uncertain timing estimation. Moment matching, used in linear systems analysis as a method of model reduction, relies on a rational approximation to extract a small set of dominant poles from a large network. The information is obtained from the Taylor series coefficients (which, because of their analogy to the time-domain moments are known simply as the moments) of that system. At the CPU cost of a little more than one DC analysis, the method has proven to be up to 2-3 orders of magnitude faster than full simulation techniques requiring a nonlinear simulator such as SPICE. It has also shown the same speedup over FFT analysis needed for transmission line problems.

McCormick and Allen of MIT first showed that the moments of an electrical network could be used to form a low-order approximate response[McC'89]. Pillage and Rohrer of Carnegie Mellon University (CMU) used the form of the RLC network equations to generate the network moments in an efficient manner and applied a rational approximation to extract an approximate set of network poles[PR90],[Pil89]. They demonstrated that the resultant low-order model was a generalization of simpler one or two-pole RC-tree timing estimation methods proposed by Penfield, Rubinstein[RP81]. The method was called "asymptotic waveform evaluation" because of its asymptotic behavior to the actual waveform with increasing time. AWE was also applied by CMU to symbolic analysis and sensitivity analysis of RLC networks.

The need for more robust interconnect models led Tang, Xie, Manney and Nakhla at Carleton University, to extrapolate AWE to interconnect networks that contained not only lumped RLC elements but also lossy, coupled uniform[TN90],[TN92] and nonuniform[MNZ92] transmission lines with nonlinear terminations. In order to include nonlinear elements, a macromodel notation was introduced, so that AWE could pre-process the linear subnetwork before combining it with the nonlinear terminations. It turned out that the CPU advantage of AWE was equally impressive on these models. Liu, Zhang and Nakhla further extended sensitivity analysis to transmission line networks[LZN92].

Alternative methods of moment matching for lossy, coupled uniform transmission lines with nonlinear terminations were proposed by Bracken, Raghavan and Rohrer of CMU[BRR92a] and by Lin and Kuh of U.C. Berkeley[LK92]. The former group, in particular, extended AWE with a method to extract long delays of transmission lines. This increased the accuracy of the approximation.

The importance of moment matching to this class of simulation problems has become increasingly evident and research efforts have intensified. However, some major problems remain to be addressed of the methods used:

1. The methods mentioned rely on a single Padé rational approximation. The number of poles generated by a single Padé approximation are limited, due to truncation errors, to generally 8-10 poles. This is a problem when a response requires more poles for an accurate waveform estimation. For examples some waveforms may require 50 or more poles because of the fine details of the ringing. This is also true if the waveform contains long delays.
2. There is no error criterion to judge the accuracy generated by a Padé approximation. An approximation will be accurate for a certain frequency range starting at DC and may be inaccurate at other ranges of frequency. There is no way to predict where the loss of accuracy begins and how it will fall off. This poses a problem for determining the accuracy range of the response.
3. The moments generated in AWE for transmission lines cause an error in the formation of Padé approximations. This is due to the methods used to calculate the moments which relies essentially on the eigenvalues of the line propagation matrix. These eigenvalues cannot be calculated with complete precision and result in an increasing truncation error that is propagated through the higher order moments. As one attempts to extract approximations of even medium or small order, unstable inaccurate poles can result. This may yield a mediocre quality simulation that gives only the major part of the waveform at best. An-

other method is needed to extract the moments of a transmission line network.

4. In nonlinear networks, there is a problem of finding the same poles for all the nodes of a macromodel. This is needed in order to represent the macromodel as a much smaller network containing only the approximate poles together with different residues for each node. In order to solve this problem various techniques have been proposed, none of which is completely adequate for an accurate simulation. Another method is needed in order to choose the same poles for every macromodel node.

Up to about 1-10 GHz, transmission line models, which are based on a quasi-TEM¹ assumption are useful. Beginning at these frequencies and beyond, the quasi-TEM transmission line models become inadequate. These models also become inadequate when there are nonuniform 3-D structures such as vias or gridded ground planes. In their place full-wave 3-dimensional analysis based on Maxwell's equations becomes necessary. These 3-D structures can also be analyzed using AWE. However, once again, the accuracy issue crops up in that for high frequency models, the accuracy must be maintained not only for low frequencies but for a range of frequencies. Further, it is important to be able to detect the resonant frequencies, which are sometimes sharp thin bands in the frequency domain, accurately and effectively without having to do a fine frequency scan requiring hundreds of frequency points. The practical interest is that one does not want to run the clock at the resonant frequencies. Present moment-matching techniques do not allow for this kind of accuracy. Full-wave 3D models such as retarded partial element equivalent circuits (rPEEC)[Rue74][Rue92] need to be also simulated in the time domain. Since these models contain retardation, present simulation techniques call for the use of a delay buffer mechanism for each delay element. Large rPEEC circuits may contain thousands of delays and as such, the memory and CPU requirement will be too large for more machines to simulate.

¹quasi Transverse Electromagnetic Mode, where it is assumed that the electromagnetic waves propagate essentially perpendicular to the line axis of the current flow.

It is also known that delay buffer simulation can result in numerical errors that grow without bound. These numerical, CPU and memory problems both in the time and frequency domains call for another method to simulate these networks.

1.2 Contributions

The subject of this thesis is the simulation of modelled high-speed interconnects using moment-matching techniques. Simulated interconnect models can include diverse elements such as lumped RLC elements, transmission lines, or discretized 3-dimensional components.

Specifically, in view of the above-mentioned problems, the main contributions of this thesis are:

1. A new method of generating the Taylor series moments of a lossy coupled transmission line network (chapter 3): It enables a higher level of accuracy in a moment-matching simulation by providing for the accurate generation of poles/residues using Padé approximations. This means that accuracy is enhanced both in the time and frequency domains. The method, called the Matrix Exponential moment generation method, relies only on the multiplication of the matrices that define the per-unit-length line parameters. As such it bypasses the numerical truncation and propagation errors that plague present techniques which rely on the eigenvalues of the propagation matrix. Besides the accuracy enhancement, the new method is far easier to understand and to program than the previous two published methods.
2. A novel *multipoint* moment-matching algorithm, namely complex frequency hopping (CFH) (chapter 6). Previous methods have relied on one expansion point or one Padé approximation. These methods have limited frequency range and as result the accuracy is compromised. CFH leads to accuracy that is predefined for an entire frequency range. Specific to the CFH method are a

well-defined search area in the complex plane, a binary search algorithm that minimizes the necessary expansion points and a pole selection technique that relies on the properties of Padé approximations. A further minimization of CPU is achieved because of the properties of sparse matrix decomposition. As a result each additional expansion point is generally far less expensive than the first, or even several additional expansion points are as expensive as the first. CFH is shown to provide some other unexpected benefits (chapter 8).

1. It provides a new method to extract the eigenvalues of a system described by lumped or distributed elements. Thus far there is no known algorithm for extracting the eigenvalues of the most generalized eigenvalue problem, or of distributed networks. The CPU cost of CFH is less than conventional methods of extracting eigenvalues.
2. It is the only method available to extract the exact resonant frequencies of a linear circuit without requiring a CPU intensive line frequency scan.
3. It allows the simulation of specialized 3D full-wave models needed for high frequency analysis. These models are not easily simulated in the time and frequency domains, and sometimes, due to the large size of the 3D networks, it is impossible to simulate them in any other way.

3. CFH is applied to macromodeling linear subnetworks in order to enable simulation with nonlinear terminations (chapter 7). When using either recursive convolution, differential equations or state space formulation, contrary to previous approaches, poles generated are the same for every node of the network enabling the accurate extraction of residues. This facilitates the inclusion of nonlinear terminations in a simulation and to ensure accuracy up to a pre-specified upper frequency.
4. An overview of published moment-matching methods that have been developed thus far within the context of asymptotic waveform evaluation (chapters 2 and 3). This information, previously contained in scattered conference and journal publications, is unified and clarified and presented with the new research

throughout the thesis. Included in this is an empirical and qualitative study and interpretation of Padé rational approximations as they apply to circuit and interconnect analysis (chapter 4). Properties of Padé approximations are specifically categorized and the applicability, strength and weaknesses of these approximations are discussed in detail. Methods used to improve Padé approximations, including new ones introduced in this thesis, are covered in chapter 5. It is shown that none of these methods, by themselves, are truly sufficient to address the general accuracy problem of moment matching, and thereby sets the stage for the necessity of CFH which relies directly on the properties of Padé approximations and moment-matching.

It is demonstrated that these techniques provide a marked improvement in the simulation of various kinds of interconnect networks.

1.3 Overview

1.3.1 Moment matching

Moment methods consist of two main steps:

- moment generation
- moment matching

Moment generation is the process by which a set of linear network equations is expanded in a Taylor series. The coefficients of the expansion are known as moments because they are related to the time moments of the transfer function. While there may be different approaches to generating the system moments, AWE uses a recursive relationship requiring only one decomposition of the system equations. This is approximately equal in CPU usage to one DC solution of the network, a minimal expense. The side benefit of doing the analysis, is that the moments for every output

variable, current or voltage, in the original equations is calculated simultaneously. The moments for a selected output can be extracted simply, while information at other outputs is readily available. This fact can be used to advantage in improving the accuracy of an approximation.

Moment matching is a technique where a set of coefficients of a low-order approximate rational transfer function are found such that the moments of the approximate function matches a given number of initial moments of the original function. This is achieved through a set of linear equations that determines the coefficients of the denominator polynomial. Given those coefficients there is an option of either determining the coefficients of the numerator polynomial or factoring the denominator and calculating a set of residues to match the resultant q poles. Each of these forms has its usefulness. Together, moment generation and moment matching, are generally referred to as moment matching techniques.

Moment matching was introduced in linear systems analysis as a form of model reduction[BL72]. When applied to electrical networks in the form of asymptotic waveform evaluation (AWE)[PR90],[Pil89] it provides a generalized approach to waveform estimation for RLC circuits with initial conditions and nonzero input signal rise times. The RLC circuits may contain floating capacitors, grounded resistors, inductors and linear controlled sources. The efficiency of this type of approach can be 2-3 orders of magnitude faster than a nonlinear solver such as SPICE. AWE extracts a small approximate set of poles and residues, or a small approximate transfer function, for a large network that may contain hundreds of actual poles. The CPU cost is approximately equal to a DC analysis of the network. The complicating issue of step size in numerical integration is not a problem for moment-matching in linear networks, because the entire waveform is generated simultaneously.

In AWE moment-matching, the linear network equations are expanded in terms of a Taylor series in the frequency (s) domain and the coefficients (moments) of the expansion are matched to a low order set of poles and residues, or a low order transfer

function. Moments are usually generated about $s = 0$ (DC) and represent the value of the function and its derivatives at the point of expansion. This means that the approximate transfer function is matched to the bulk of the signal energy at lower frequencies, while the approximate transient response is asymptotic to the actual response as the time $t \rightarrow \infty$ in low-pass structures such as interconnects. AWE is also formulated to include moments from an expansion about $s = \infty$. This allows the matching of the original function at both $s = 0$ and $s = \infty$ and regions near those points, improving the transient simulation near $t = 0$. In low-pass interconnect transfer functions, AWE is remarkably accurate while being efficient.

Moment-matching can be readily incorporated into a circuit simulator that includes numerous features such as simulation, optimization and parallel processing [GCZN92].

AWE relies primarily on Padé rational approximations and, as such, it is important to understand the behavior of Padé approximations. However, there are several deviations from the standard Padé approximation, which when applied to network analysis have been shown to improve the accuracy considerably. A seemingly natural extension is to use more than one expansion point. However, the implementation questions that arise are not immediately answered. How is information to be combined from two or more expansion points? How are the poles to be matched to the moments? What does one do about approximate poles? The purpose of this thesis is to give answers to those questions and give, as a result, an algorithm that extends moment-matching to multipoint analysis.

Moment matching can also be extended to forming an n -port q -pole macromodel for a linear subnetwork. The moments of each input/output transfer function are matched to each of the approximating port parameters. The linear macromodel can then be efficiently used to simulate the linear network with nonlinear terminations such as drivers or receivers.

As an example of the strength of these methods, consider a large RL mesh (including multiple mutual couplings between the inductances) with transmission line leads (Fig. 1.2). The network was extracted from a Plastic Quad Flat Pac (PQFP) copper-plate ground-plane layer using a 3-D field equation analyzer. The resultant network contained 768 resistances, 747 inductances, 7717 mutual inductances and 20 transmission line leads. The copper plate had 10 sources and 10 sinks which were modelled as lossless transmission lines (see Fig. 1.1). The plate network was simulated for ground bounce effects using AWE and a source input step of 0.1 ns rise time. The transient waveform was analyzed at the sinks. The cost of AWE analysis was just above the cost of a DC analysis of the circuit, approximately 30 CPU seconds (SUN SPARC 10), while in contrast, SPICE took over an hour. The waveform comparison is given in Fig. 1.3.

1.3.2 RC trees

In the past, interconnect models have been generally restricted to RC tree models. RC trees are RC circuits with capacitors from all nodes to ground, no floating capacitors, no resistor loops, and no resistors to ground. Due to the complexity and CPU time required to analyze large linear circuits, efficient algorithms were proposed for the analysis of RC interconnect models. Timing analyzers[Ous83]-[Jou87] and timing simulators[LM81] have been used to capture the effect of this interconnect delay for a timing verification. For many MOS circuits, timing analyzers have often been able to predict the interconnect delay with a simplified RC model to within 10 percent of a SPICE simulation prediction[RP84]. The signal delays through RC tree were often estimated using a form of the Elmore delay [Elm47], which provided a dominant time constant approximation for monotonic step responses.

However, at increasing speeds, and in divergent technologies such as Bipolar, BiCMOS, high-speed MOS or MCM's, RC tree models are no longer adequate[OW86], [Wya87], [ABL⁺92a]. In Bipolar circuits, lumped-element interconnect models may

Figure 1.1: PQFP copper ground plate with transmission line leads.

Figure 1.2: RL mesh with mutual inductive couplings and transmission lines extracted from a 3-D field solver applied to the ground plate.

Figure 1.3: PQFP ground bounce simulation.

require the use of inductors or grounded resistors, models which are not compatible with RC trees. Even for MOS circuits at high speeds, the effects of coupling capacitances may need to be included in the delay estimate. At the PCB and MCM level, input voltage rise time can dominate the timing of an interconnect net. Therefore, a step response approximation is not always useful. Moreover, for generality, a solution is required when there are non-equilibrium initial conditions, so that the delays due to charge sharing effects can be predicted.

RLC circuits with non-equilibrium initial conditions may have response waveforms which are non-monotonic. This typically results in visible signal ringing in the waveform. A single time constant approximation with the Elmore delay is not generally applicable for such circuits. Two time constant models have been shown to improve the accuracy[CH87], but only as applied to RC tree monotone response approximations. AWE provides a general method to obtain a waveform estimation of not only RC trees but also general RLC networks as well as other linear devices. However, while the response for RC or RL networks is in excellent agreement with a full simulation, results can vary for a general RLC network. This is because RLC networks contain poles off the real axis and may cause a problem. CFH address this concern by multipoint expansions which can obtain poles anywhere in the complex plane.

1.3.3 Transmission lines

At higher switching speeds, lumped-element interconnect models become inadequate and fully distributed transmission line models with per-unit-length parameters are needed[Bak90]. The basic model is the simple "delay" line, or lossless transmission line. More complicated models include per-unit-length loss (either in the direction of the traveling wave or due to dielectric substrate loss), or coupling between adjacent transmission lines, where the coupling may be resistive, inductive, capacitive or a combination. The models may even require frequency-dependent parameters for effects such as skin loss, or nonuniform per-unit-length parameters to accurately model different geometries.

The main problem in simulating such models lies in the fact that distributed elements, while formulated in terms of partial differential equations, are best described in the frequency domain. Nonlinear terminations of the network, on the other hand, can only be given in the time domain. These simultaneous formulations cannot be handled by a traditional ordinary differential equation solver. Several methods have been proposed recently for the simulation of interconnect models which incorporate distributed elements[DSH86]-[GN92]. While many of these simulation methods are accurate, the CPU speed may be disproportionately high. Moment matching provides an alternative and fresh approach to the problem by handling, in an efficient manner, large interconnect networks consisting of lumped elements and distributed elements such as a lossy coupled uniform or nonuniform transmission lines. The results are as efficient as is moment matching applied to lumped element networks. For lines not containing very long delays, and for networks with relatively low inductive properties, AWE in its fundamental form yields good accuracy of waveform estimation.

In order to increase accuracy it is possible to model each set of transmission lines separately, and to extract a time domain delay out of each set, thereby improving a moment-matching approximation on the remaining structure. This "method of characteristics" formulation has been used in conjunction with AWE to improve the

accuracy for long delay lines that cannot be approximated well by a small number of poles typically generated with one moment-matching approximation[BRR92b]. However this method's drawback is in the fact that CPU time has to be expended in analyzing and forming a macromodel for each transmission line separately, as well as to simulate the delays via a memory delay buffer mechanism. CFH addresses the problem by extracting more accuracy out of a network than is possible by a single expansion.

1.3.4 3-D analysis

At higher speeds, electromagnetic (EM) field effects become prevalent, and even the two-dimensional models[Mat69] become inadequate. Current may flow in complicated patterns through flat interconnect strips, down vias, and out to planar structures, interacting with special EM fields. Fully three-dimensional models based on Maxwell's equations become a necessity. Partial element equivalent circuit (PEEC)[Rue74] models have been shown to be a useful method to model these networks. PEEC models are RLC circuits where the individual resistances and capacitances are extracted from the geometry using a quasi-static (non retarded) solution of the Maxwell equations. These models require solutions both in the frequency and time domains. Although the large resultant networks are solvable with general purpose circuit simulators like SPICE or ASTAP, the CPU time required is generally large. The simulation becomes even more expensive if the PEEC models include the effect of the finite speed of light (retarded PEEC or rPEEC)[H92]. Simulating rPEEC models requires large memory in the form of delay buffers and extensive CPU time for both time and frequency domain analysis, and can incur some numerical errors. CFH, however, can provide a quick method to extract both the time and frequency responses simultaneously without the necessity of delay buffers.

Also in general use are large RC or RL models extracted from a 3-D field solver. Simulating these models can be very expensive in terms of CPU. Moment-matching

can provide a method to perform these large simulations at 2-3 orders of magnitude speedup.

1.3.5 Nonlinear terminations

Drivers and receivers of interconnect nets, as well as other terminations, must generally be modelled with nonlinear elements. If a large linear network contains some of these nonlinear terminations, then they also can be included in a moment-matching simulation. This is accomplished by reducing the linear subnetwork to a smaller macromodel based on its dominant modes. The macromodel will consist of lumped elements and generally contain less circuitry than the full linear subnetwork. The macromodel, together with the nonlinear terminations, can then be inserted into nonlinear simulator such as SPICE. Another option is to simulate the network directly outside of the context of SPICE using a recursive convolution formulation[CN92b], [LK92]. However, in both cases, it is expedient to obtain the same poles for each of the port transfer functions because it reduces the size of the macromodel and the CPU time to simulate it. Present techniques artificially constrain the poles to be the same. CFH provides for this capability because it generates the actual dominant poles of a network rather than its approximate poles.

1.4 Organization of the Thesis

AWE as applied to RLC circuits is a basic starting point and will be outlined in Chapter 2. The extension of moment generation to transmission lines will be discussed in Chapter 3. In this chapter it is shown how present methods generate numerical accuracy errors. An alternative method, matrix exponential, of generating the moments will be given and will be shown more accurate and easier to capture. Chapter 4 will deal with Padé approximations and their properties which are necessary to understanding the techniques introduced in this thesis. It will be shown why improving the

accuracy of the moments even slightly by the novel matrix exponential method, can improve the accuracy greatly. Some methods currently used to improve the accuracy of AWE, and some new ones, will be presented in Chapter 5. Their strengths and inadequacies will be discussed. A new multi-point moment matching method (CFH) will be introduced in Chapter 6. It will be shown how CFH can deal with the accuracy concerns without greatly multiplying CPU cost. Chapter 7 will describe the application of macromodel formation to nonlinear simulation. The new method addresses the problem of finding the same poles for each output node. Finally, Chapter 8 will explore the application of moment matching to three-dimensional electromagnetic models (rPEEC). CFH is also efficient and even necessary for simulating these models.

Chapter 2

Asymptotic Waveform Evaluation

I begin with a description of AWE as applied to RLC lumped-element circuits. It will be shown in the next chapters how this approach can be efficiently extended to networks with transmission lines, while at the same time allowing the inclusion of nonlinear terminations in the simulation.

AWE, as moment matching in general, consists of two main parts: 1. Moment generation, and 2. Moment matching. AWE approximates the transient response of an RLC circuit by first expanding the system equations in moments (coefficients) of a Taylor expansion about $s = 0$ (Maclaurin series) in the frequency domain, and then matching the first $2q$ moments of the series, as well as the initial time conditions, to a low-order q pole model. An efficient set of algorithms for this purpose is developed. In the case of an RC tree driven by a step input, a first order AWE approximation is equivalent to the methods which employ the Elmore delay expression.

2.1 State space formulation

Asymptotic waveform evaluation can best be described in terms of the differential state equations for a lumped, linear, time-invariant circuit. While this is not the only possible description, (another is covered in Section 2.3), it is the simplest to

understand and formulate. Such a network description usually exists [CL75], and is given by,

$$\begin{aligned}\frac{d}{dt}\mathbf{x}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{c}^T\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)\end{aligned}\quad (2.1)$$

where $\mathbf{x}(t)$ is the n -dimensional state vector, $\mathbf{u}(t)$ the m -dimensional excitation vector, and $\mathbf{y}(t)$ the vector of required outputs.

Figure 2.1: Simple state space network.

As a simple example consider the network in Fig. 2.1. The state space equations for the network are given as,

$$\begin{bmatrix} \frac{d}{dt}v_{C1} \\ \frac{d}{dt}v_{C2} \end{bmatrix} = \begin{bmatrix} -\left(\frac{1}{R_1C_1} + \frac{1}{R_2C_1}\right) & \frac{1}{R_2C_1} \\ \frac{1}{R_2C_2} & -\frac{1}{R_2C_2} \end{bmatrix} \begin{bmatrix} v_{C1} \\ v_{C2} \end{bmatrix} + \begin{bmatrix} \frac{1}{R_1C_1} \\ 0 \end{bmatrix} v_{in}$$

$$v_{out} = \begin{bmatrix} -1 & 0 \end{bmatrix} \begin{bmatrix} v_{C1} \\ v_{C2} \end{bmatrix} + [1] v_{in}$$

Taking the Laplace transform of (2.1) we obtain,

$$\begin{aligned}s\mathbf{X}(s) - \mathbf{x}(0^+) &= \mathbf{A}\mathbf{X}(s) + \mathbf{b}\mathbf{U}(s) \\ \mathbf{Y}(s) &= \mathbf{c}^T\mathbf{X}(s) + \mathbf{D}\mathbf{U}(s)\end{aligned}\quad (2.2)$$

Solving for $\mathbf{Y}(s)$,

$$\mathbf{Y}(s) = \mathbf{c}^T(s\mathbf{1} - \mathbf{A})^{-1}(\mathbf{b}\mathbf{U}(s) + \mathbf{x}(0^+)) + \mathbf{D}\mathbf{U}(s) \quad (2.3)$$

Because this is a linear system, superposition applies. $\mathbf{Y}(s)$ may be decomposed into $\mathbf{D}\mathbf{U}(s)$, which is a direct coupling between the input and the output and needs no special calculation, $\mathbf{c}^T(s\mathbf{1} - \mathbf{A})^{-1}\mathbf{b}\mathbf{U}(s)$, which is the response of the system to input $\mathbf{U}(s)$, and $\mathbf{c}^T(s\mathbf{1} - \mathbf{A})^{-1}\mathbf{x}(0^+)$ which is the response of the system to an impulse source equal to the initial conditions of the network, $\mathbf{x}(0^+)$. The AWE calculation can be simplified by calculating an impulse response once, and then using the impulse response to calculate the network response of any other input waveform. However, the number of calculations required will depend on the vectors $\mathbf{U}(s)$ and $\mathbf{x}(0^+)$.

For a single input waveform $\mathbf{U}(s) = U(s)$ (as in the example above, $U(s) = v_{in}$), and a set of zero initial conditions, the solution simplifies to,

$$\mathbf{Y}(s) = \mathbf{c}^T(s\mathbf{1} - \mathbf{A})^{-1}\mathbf{b}U(s) + \mathbf{D}U(s) \quad (2.4)$$

The goal of AWE is to reduce the system transfer function, $\mathbf{c}^T(s\mathbf{1} - \mathbf{A})^{-1}\mathbf{b}$, to an approximate transfer function, using an efficient model reduction method. Taking the single source $U(s) = 1.0$, a scalar representing an impulse $\delta(t)$ in the time domain¹, equation (2.4) then becomes,

$$\mathbf{Y}(s) = \mathbf{c}^T(s\mathbf{1} - \mathbf{A})^{-1}\mathbf{b} + \mathbf{D} \quad (2.5)$$

Applying AWE, we solve (2.5) once to obtain an approximate transfer function. The response to any other input waveform, $u(t)$ may then be calculated by multiplying (in the frequency domain) or by convoluting (in the time domain) the new input waveform with the approximate impulse response generated by AWE. Any nonzero direct-coupling term may be superimposed on this solution.

¹ $\mathbf{b}$ is then a vector of values, b_j ($j = 1, \dots, n$), in the j -th position indicating an impulse source (voltage or current) of strength $b_j > 0$ connected to the j -th state variable, or no external source at the j -th position if $b_j = 0$. This implies that, in the case of multiple sources, which are mapped from the single source $\mathbf{u}(t) = \delta(t)$ by $\mathbf{b}$ onto $\mathbf{x}(t)$, all are identical waveforms with the exception of magnitude, as though they were connected to one central source.

2.1.1 System moments

Expanding $\mathbf{Y}(s)$ in terms of a Maclaurin series we obtain,

$$\mathbf{Y}(s) = \mathbf{D} - \mathbf{c}^T \mathbf{A}^{-1} \mathbf{b} - s \mathbf{c}^T \mathbf{A}^{-2} \mathbf{b} - s^2 \mathbf{c}^T \mathbf{A}^{-3} \mathbf{b} + \dots \quad (2.6)$$

or,

$$\mathbf{Y}(s) = \mathbf{M}_0 + s \mathbf{M}_1 + s^2 \mathbf{M}_2 + \dots \quad (2.7)$$

The moments, $\mathbf{M}_i$ of the expansion of $\mathbf{Y}(s)$ are given as,

$$\mathbf{M}_0 = \mathbf{D} - \mathbf{c}^T \mathbf{A}^{-1} \mathbf{b} \quad (2.8)$$

$$\mathbf{M}_i = -\mathbf{c}^T \mathbf{A}^{-(i+1)} \mathbf{b} \quad (i > 0) \quad (2.9)$$

The $\mathbf{M}_i$ represent the i -th moment vectors containing the moments of all outputs in $\mathbf{Y}(s)$. Selecting a particular output (row) of the moment vectors, the selected scalar moments are labeled as m_i . These moments are the coefficients of the Maclaurin expansion of the the selected output node, given an impulse at the input. In other words, the transfer function $H(s)$ of the selected output is given by,

$$H(s) = m_0 + s m_1 + s^2 m_2 + \dots \quad (2.10)$$

The m_i are identical to the time domain moments of the impulse response of the particular output, $h(t) = \mathcal{L}^{-1}(H(s))$,

$$m_i = \frac{(-1)^i}{i!} \int_0^\infty t^i h(t) dt \quad (i \geq 0) \quad (2.11)$$

This can be seen by expanding the definition of the Laplace transform in a Maclaurin series about s ,

$$H(s) = \int_0^\infty e^{-st} h(t) dt = \sum_{i=0}^\infty \frac{(-1)^i}{i!} s^i \int_0^\infty t^i h(t) dt = \sum_{i=0}^\infty s^i m_i \quad (2.12)$$

It has been established that the time moments provide a good measure of delays and rise times[Elm48],[VS83]. AWE uses this fact to extract an approximate transfer function from the moments.

2.1.2 Multiple inputs/outputs and initial conditions

Multiple outputs are accommodated quite easily, because the moments for each output are generated simultaneously with all of the other outputs. A separate approximation may be formed for each, or using the method of optimal pole selection (Chapter 5), the poles are generated at only one node and then the residues are calculated for each node separately.

For multiple inputs with different waveform sources in $U(s)$, the algorithm is carried out once for each separate input source and superimposed. An impulse response is calculated for every linearly independent waveform input by zeroing the other sources².

If a non-zero initial condition vector $x(0^+)$ exists, the procedure must also be carried out for the unique initial condition vector, and then again for each new initial condition vector simulated which is not linearly dependent on the previous one. If an initial condition vector is linearly dependent on previous ones simulated, then its response can be obtained by a linear combination of the other responses³.

If a single ramp response with non-zero initial conditions is sought, it may be calculated in one step without requiring both the impulse response and separate initial condition response[PR90].

²This is equivalent to a macromodel formation which will be dealt with in more detail in Chapter 7.

³ b may also be included in the set of previous vectors simulated. If, for instance, an initial condition vector is a multiplicative constant times matrix b , then its response is easily obtainable from (2.5).

2.2 Generating an approximate response

The form of an actual one input, one output transfer function may be described as,

$$H(s) = \frac{P(s)}{Q(s)} \quad (2.13)$$

where $P(s)$ and $Q(s)$ are polynomials in s where the degree of the numerator is less than or equal to the degree of the denominator. Applying a partial fraction decomposition on (2.13) yields,

$$H(s) = c + \sum_{j=1}^{N_p} \frac{k_j}{s - p_j} \quad (2.14)$$

N_p is the number of actual poles, p_j , of the network and the number of residues, k_j , of the selected output. While the poles of the network are the same at every output node, the residues will vary from node to node. The constant c also varies with the output and indicates direct coupling between the input and the output. $H(s)$ may also be decomposed into N_z zeros as well as N_p poles,

$$H(s) = \frac{(s - z_1)(s - z_2) \cdots (s - z_{N_z})}{(s - p_1)(s - p_2) \cdots (s - p_{N_p})} \quad (2.15)$$

The form in (2.14) is more convenient for an inversion of Laplace transform into the time domain. The time domain impulse response of the output can be given in closed form in terms of the poles and residues as,

$$h(t) = cd(t) + \sum_{j=1}^{N_p} k_j e^{p_j t} \quad (2.16)$$

In large networks, N_p , and therefore the order of $Q(s)$, can be of the order of hundreds or thousands. Generating the N_p poles of even a small network is very CPU intensive. With large networks, it is completely impractical to attempt a direct extraction of $H(s)$ or the poles[Hua90].

Using the moments generated, a low-order q -pole approximation to (2.14) can be constructed in the form,

$$H(s) \approx \hat{H}(s) = \frac{\hat{P}(s)}{\hat{Q}(s)} = \frac{(s - \hat{z}_1)(s - \hat{z}_2) \cdots (s - \hat{z}_q)}{(s - \hat{p}_1)(s - \hat{p}_2) \cdots (s - \hat{p}_q)} \quad q \ll N_p \quad (2.17)$$

where the $\hat{z}_j$ are the approximate zeros and $\hat{p}_j$ are the approximate poles. Equation (2.17) can also be reformulated by partial fraction decomposition as,

$$\hat{H}(s) = \hat{c} + \sum_{j=1}^q \frac{\hat{k}_j}{s - \hat{p}_j} \quad (2.18)$$

In the time domain, an approximate impulse response can then be given by,

$$h(t) \approx \hat{h}(t) = \hat{c}\delta(t) + \sum_{j=1}^q \hat{k}_j e^{\hat{p}_j t} \quad (2.19)$$

Similar closed forms are also possible for an approximate step response, ramp response, or exponential response, by multiplying the input function by the impulse response in the frequency domain and decomposing the result into partial fractions. Each of these is outlined in Table 2.1. All other complex input waveforms may be decomposed into a linear combination of these simpler forms.

2.2.1 Poles and residues

In order to obtain the $\hat{p}_j$ and $\hat{k}_j$, we expand (2.18) in a Maclaurin series,

$$\hat{H}(s) = \hat{c} - \sum_{n=0}^{\infty} s^n \left(\sum_{j=1}^q \frac{\hat{k}_j}{\hat{p}_j^{n+1}} \right) \quad (2.20)$$

The $\hat{c}$, in the case of a state space formulation, may be obtained directly from the **D**. It is in the same position of the selected output of **Y(s)**. It will therefore be made equal to c . When (2.20) is equated with (2.10) it yields the $2q$ equations needed to solve for the q approximate poles, $\hat{p}_j$, and the q approximate residues $\hat{k}_j$,

$$m_0 = c - \sum_{j=1}^q \frac{\hat{k}_j}{\hat{p}_j} \quad (2.21)$$

and,

$$m_i = - \sum_{j=1}^q \frac{\hat{k}_j}{\hat{p}_j^{i+1}} \quad (0 < i < 2q) \quad (2.22)$$

Input function ($t \rightarrow s$)	Response:
	$H(s)$
	$h(t)$
Impulse $\delta t \rightarrow 1$	$c + \sum_{j=1}^q \frac{k_j}{s - p_j}$ $c\delta(t) + \sum_{j=1}^q k_j e^{p_j t}$
Step $u(t) \rightarrow \frac{1}{s}$	$\frac{c}{s} + \sum_{j=1}^q \left(\frac{k_j/p_j}{s - p_j} - \frac{k_j/p_j}{s} \right)$ $c u(t) + \sum_{j=1}^q \left(\frac{k_j}{p_j} e^{p_j t} - \frac{k_j}{p_j} u(t) \right)$
Ramp $tu(t) \rightarrow \frac{1}{s^2}$	$\frac{c}{s^2} + \sum_{j=1}^q \left(\frac{k_j/t_r}{p_j^2} \frac{1}{s - p_j} - \frac{k_j}{sp_j^2} - \frac{k_j}{s^2 p_j} \right)$ $ctu(t) + \sum_{j=1}^q \left(\frac{k_j}{p_j^2} e^{p_j t} - \frac{k_j}{p_j^2} u(t) - \frac{k_j}{p_j} tu(t) \right)$
Exponential $e^{at} \rightarrow \frac{1}{s-a}$ ($a \neq p_j$)	$\frac{c}{s-a} + \sum_{j=1}^q \left(\frac{k_j/(a-p_j)}{s-a} - \frac{k_j/(a-p_j)}{s-p_j} \right)$ $c e^{at} + \sum_{j=1}^q \left(\frac{k_j}{a-p_j} e^{at} - \frac{k_j}{a-p_j} e^{p_j t} \right)$

Table 2.1: Response of a network to various inputs

This set of nonlinear equations may be solved directly. However, it is more convenient to transform it into a set of linear equations[PR90]. This can be accomplished by solving for the q coefficients⁴ of,

$$\hat{Q}(s) = b_q s^q + b_{q-1} s^{q-1} + \cdots + b_0 + 1 \quad (2.23)$$

⁴ b_0 is taken to be equal to 1, which eliminates an arbitrary multiplicative factor in $\frac{P(s)}{Q(s)}$

The result is a set of equations in the form,

$$\begin{bmatrix} m_0 & m_1 & \dots & m_{q-1} \\ m_1 & m_2 & \dots & m_q \\ \vdots & \vdots & \ddots & \vdots \\ m_{q-1} & m_q & \dots & m_{2q-2} \end{bmatrix} \begin{bmatrix} b_q \\ b_{q-1} \\ \vdots \\ b_1 \end{bmatrix} = - \begin{bmatrix} m_q \\ m_{q+1} \\ \vdots \\ m_{2q-1} \end{bmatrix} \quad (2.24)$$

or,

$$\mathbf{m}_h \mathbf{b}_v = \mathbf{m}_v \quad (2.25)$$

The $\hat{p}_j$ can be obtained from (2.24) by applying a root solving algorithm on the resulting $\hat{Q}(s)$. In order to solve for the residues, the now calculated poles are substituted into (2.21) and (2.22) to give a set of linear equations for the $\hat{k}_j$. In order to improve the accuracy at $t = 0^+$, the equation for m_{q-1} is substituted for a time domain condition of,

$$\sum_{j=1}^q \hat{k}_j = h(0^+) \quad (2.26)$$

This condition is obtained from (2.16) at $t = 0^+$. The solution to all outputs of $\mathbf{y}(t)$ at $t = 0^+$ is given by,

$$\mathbf{y}(0^+) = \mathbf{c}^T \mathbf{b} = \mathbf{M}_{-1} \quad (2.27)$$

For reasons explained in Chapter 5, it is referred to as the negative first moment vector. Again, only the single output node is extracted from $\mathbf{M}_{-1}$ to give m_{-1} , yielding,

$$\sum_{j=1}^q \hat{k}_j = m_{-1} \quad (2.28)$$

The equations are then given by,

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ \hat{p}_1^{-1} & \hat{p}_2^{-1} & \dots & \hat{p}_q^{-1} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{p}_1^{-q+1} & \hat{p}_2^{-q+1} & \dots & \hat{p}_q^{-q+1} \end{bmatrix} \begin{bmatrix} \hat{k}_1 \\ \hat{k}_2 \\ \vdots \\ \hat{k}_q \end{bmatrix} = - \begin{bmatrix} m_{-1} \\ m_0 - c \\ \vdots \\ m_{q-2} \end{bmatrix} \quad (2.29)$$

Generating an approximate set of poles and residues from a set of moments is not costly in terms of CPU time. The matrices of (2.24) and (2.29), are relatively small for a small q and the only other expense is the root finding of $\hat{Q}(s)$, which is also relatively small for a small q . All these matrix inversions are of $O(q^3)$. The accuracy of the approximation will be dependent on the number of poles extracted. The question of how to choose q is an important one and will be dealt with later.

2.3 MNA formulation

While the formulation of the above equations was in the form of the system matrix $\mathbf{A}$, it is not always necessary to reduce the network to this formulation. Extracting the $\mathbf{A}$ requires the elimination of all linearly dependent variables such as dependent capacitor voltages in capacitor loops or dependent inductor currents in inductor cut-sets.

If we assume that a linear subnetwork ϕ contains only lumped elements, then the modified nodal admittance (MNA)[HRB75] formulation of the subnetwork may be used. In practice it is also possible to use any other formulation method such as the sparse tableau formulation[Wee73]. I choose the MNA formulation here for simplicity and generality. Describing a subnetwork ϕ in terms of the MNA formulation yields,

$$\mathbf{C}_\phi \frac{d}{dt} \mathbf{v}_\phi(t) + \mathbf{G}_\phi \mathbf{v}_\phi(t) - \mathbf{e}_\phi(t) = 0 \quad (2.30)$$

where:

$\mathbf{v}_\phi(t) \in \mathbb{R}^{N_\phi}$ is the vector of the N_ϕ variables describing the subnetwork ϕ which includes node voltage waveforms appended by independent voltage source current, linear inductor current, nonlinear capacitor charge and nonlinear inductor flux waveforms. $\mathbf{C}_\phi \in \mathbb{R}^{N_\phi \times N_\phi}$, $\mathbf{G}_\phi \in \mathbb{R}^{N_\phi \times N_\phi}$ are constant matrices describing the lumped memory and memoryless elements of network ϕ respectively, and $\mathbf{e}_\phi(t) \in \mathbb{R}^{N_\phi}$ is the vector of source waveforms.

Taking the Laplace transform of (2.30) and assuming only an impulse source, $\mathbf{e}_\phi(t) = \mathbf{E}_\phi \delta(t)$, we get,

$$(\mathbf{G}_\phi + s\mathbf{C}_\phi)\mathbf{V}_\phi(s) = \mathbf{E}_\phi \quad (2.31)$$

Expression (2.31) can be used to derive the moments $\mathbf{V}_{\phi[i]}$ of $\mathbf{V}_\phi(s)$ in an efficient manner. Expanding $\mathbf{V}_\phi(s)$ in a Maclaurin series we get,

$$(\mathbf{G}_\phi + s\mathbf{C}_\phi)(\mathbf{V}_{\phi[0]} + s\mathbf{V}_{\phi[1]} + s^2\mathbf{V}_{\phi[2]} + \dots) = \mathbf{E}_\phi \quad (2.32)$$

Equating like powers of s gives a recursive relationship for the moments:

$$\mathbf{G}_\phi \mathbf{V}_{\phi[0]} = \mathbf{E}_\phi \quad (2.33)$$

and,

$$\mathbf{G}_\phi \mathbf{V}_{\phi[i]} = -\mathbf{C}_\phi \mathbf{V}_{\phi[i-1]} \quad (i > 0) \quad (2.34)$$

Because matrices $\mathbf{G}_\phi$ and $\mathbf{C}_\phi$ represent sparse matrices, the solution of (2.33) and (2.34) can be accomplished efficiently. Further, any invertible matrix $\mathbf{G}_\phi$ can be decomposed (factorized) into two triangular matrices. The result is a lower triangular matrix $\mathbf{L}$ and an upper triangular matrix $\mathbf{U}$. where,

$$\mathbf{LU} = \mathbf{G}_\phi \quad (2.35)$$

Once this is done, then any system of equations,

$$\mathbf{G}_\phi \mathbf{X} = \mathbf{B} \quad (2.36)$$

may be solved efficiently by first solving sequentially the two simpler equations,

$$\mathbf{LY} = \mathbf{B} \quad (2.37)$$

and,

$$\mathbf{UX} = \mathbf{Y} \quad (2.38)$$

The last two steps are solved by a forward and then backward substitution of the unknown variables. The greatest use of CPU time is in the LU factorization of the

matrix. Forward/backward substitution are less costly for each additional solution. Equations (2.33) and (2.34) can be solved recursively with one LU factorization of matrix $\mathbf{G}_\phi$. This is approximately equal in CPU time to a DC solution of the circuit equations. In fact (2.33) is equivalent to the DC solution of the network with constant DC sources $\mathbf{E}_\phi$. Each moment can then be obtained from the previous one by pre-multiplying by $\mathbf{C}_\phi$ (a sparse matrix) and one forward/backward substitution using the LU decomposed matrix $\mathbf{G}_\phi$.

The matrix $\mathbf{G}_\phi$ may be alternatively decomposed using a graph search routine which exploits the tree-like structure of some interconnect networks[RGP91]. For these structures there is a good speed-up gain. If the $\mathbf{G}_\phi$ is not invertible it implies that there exist floating nodes connected only through capacitors or inductors which at DC are not connected to the network. This problem can be easily rectified by invoking charge conservation on the floating node. The charge conservation equations are then incorporated into the matrix[PHR91] making it invertible.

Note that the moments for every variable in $\mathbf{V}_\phi$ are calculated simultaneously. This property will come in useful when we are interested in improving the accuracy of a moment-matching approximation or forming macromodels. No extra CPU time is necessary for other variables of interest. However, for a particular variable in position j in $\mathbf{V}_\phi$, moments for that variable will be available in the same position j in the system moments, $\mathbf{V}_\phi[i]$.

The number of moments necessary depends on the order of the approximation. In general, when constructing a q -pole approximate transfer function, $2q$ moments are needed.

It is not necessary to calculate c directly and the extraction of an approximation,

$\hat{c}$, may be done assuming it also as an unknown variable,

$$\begin{bmatrix} 1 & 1 & \dots & 1 & 0 \\ \hat{p}_1^{-1} & \hat{p}_2^{-1} & \dots & \hat{p}_q^{-1} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \hat{p}_1^{-q} & \hat{p}_2^{-q} & \dots & \hat{p}_q^{-q} & 0 \end{bmatrix} \begin{bmatrix} \hat{k}_1 \\ \vdots \\ \hat{k}_q \\ \hat{c} \end{bmatrix} = - \begin{bmatrix} m_{-1} \\ \vdots \\ m_{q-2} \\ m_{q-1} \end{bmatrix} \quad (2.39)$$

However, another approach can be used to solve for c exactly, but takes slightly more processing. I note first that,

$$c = \lim_{s \rightarrow \infty} \left(c + \sum_{j=1}^{N_p} \frac{k_j}{s - p_j} \right) = \lim_{s \rightarrow \infty} H(s) \quad (2.40)$$

By setting the frequency as infinite, the solution of the network at the required node is c . This is achieved by solving the circuit when the capacitors are short-circuited and the inductors open-circuited. Generally, only nodes close to a source will be directly coupled with the source and therefore have a non-zero c . Once c is solved for, (2.29) may be used to solve for the residues.

Solving for m_{-1} is slightly more complicated in the MNA formulation because the state space matrices $\mathbf{b}$ and $\mathbf{c}$ are not defined. By property of Laplace transform,

$$\lim_{s \rightarrow \infty} (sH(s)) = h(0^+) = m_{-1} \quad (2.41)$$

or, if there is there no direct coupling at the node,

$$\lim_{s \rightarrow \infty} \left(s \sum_{j=1}^{N_p} \frac{k_j}{s - p_j} \right) = \sum_{j=1}^{N_p} k_j = h(0^+) = m_{-1} \quad (2.42)$$

By placing a source of value s at the input and solving the circuit the solution will be $\mathbf{M}_{-1}$. The current must be solved first by comparing orders of s and eliminating diminishing impedances. Most nodes away from the source will receive no current. This is because an impulse is absorbed by the first energy elements it comes in contact with giving a finite step. All other nodes will change in a continuous fashion.

Once the elements with no current are eliminated, the much smaller circuit may be solved for voltages. These nonzero voltages will be the conditions at $t = 0^+$.

2.4 Examples

I use two small examples to illustrate the method. The order of the approximation refers to the number q or the number of poles generated by an approximation.

RC network

Figure 2.2: RC ladder network.

A small RC network is given in Fig. 2.2. AWE was applied to V_{out} in both frequency and time domain analysis. The first moments, scaled by 10^9 , are given in Table 2.2. In the first order approximation only a single pole/residue pair was generated. An input pulse waveform of 0.1 ns rise/fall time and 0.3 ns duration was simulated. The transient first order response approximation is given in Fig. 2.3. Subsequently an order 2 and order 3 approximation was performed with the results given in Fig. 2.4 and Fig. 2.5 respectively. A 2nd order approximation is seen to be enough for the waveform to converge. The 6 actual poles of the network using a pole/zero analysis in SPICE and the approximate poles generated by AWE are compared in Table 2.3. The dominant pole is closest to the origin and is approximated well by second order.

The frequency response approximation corresponding to the first order impulse response approximation is given in Fig. 2.6. The approximations corresponding to 2nd and 3rd order ones are seen in Fig. 2.7 and Fig. 2.8 respectively.

Figure 2.3: RC network transient response: 1st order approximation.

Figure 2.4: RC network transient response: 2nd order approximation.

Figure 2.5: RC network transient response: 3rd order approximation.

Figure 2.6: RC network frequency response: 1st order approximation.

Figure 2.7: RC network frequency response: 2nd order approximation.

Figure 2.8: RC network frequency response: 3rd order approximation.

Moment	value
m_{-1}	0
m_0	3.3333e-1
m_1	-3.6296e-2
m_2	2.9671e-3
m_3	-2.2647e-4
m_4	1.6999e-5
m_5	-1.2709e-6
m_6	9.4923e-8
m_7	-7.0886e-9
m_8	5.2933e-10
m_9	-3.9527e-11

Table 2.2: Moments of Fig. 2.2.

RLC network

Consider the small RLC network in Fig. 2.9. Formulating the equations in terms of G_ϕ and C_ϕ we get for each respectively,

q	poles $\times 1e^{-10}$					
1	-.91837					
2	-1.3499 -4.1697					
3	-1.3392 -6.2617 -9.1976					
Act.	-1.3392	-5.9729	-1.3692	-22.932	-31.620	-37.778

Table 2.3: Approximate/actual poles of Fig. 2.2.

Figure 2.9: RLC network.

 $\mathbf{G}_\phi =$

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0.1 & -0.1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -0.1 & 0.1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & -0.01 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -0.01 & 0.01 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0333 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$C_\phi =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10^{-12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10^{-12} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-12} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10^{-9} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10^{-9} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -10^{-9} \end{bmatrix}$$

I solve for the 7th voltage node as output (indicated in Fig. 2.9). Applying (2.33) and (2.34) to generate the moment vectors and taking the 7th nodal voltage variable yields the moments: $m_{-1} = 0$, $m_0 = 2.1277e-1$, $m_1 = -1.921e-11$, $m_3 = -1.4784e-22$, etc. which can be seen by inspection.

Solving for the coefficients of $\hat{P}(s)$ and extracting the poles for different values of q , gives Table 2.4. The poles converge to the actual poles of the network, which, at $q = 6$, are identical.

An input pulse waveform of 0.1 ns rise/fall time and 0.3 ns duration was simulated. The convergence for different orders of approximation in the time domain can be seen in Fig. 2.10-Fig. 2.13. The waveform converges by the 5th order. The various frequency response approximations are given in Fig. 2.14-Fig. 2.18.

While the frequency response does not converge by the 5th order approximation, the transient response does, due to the particular input waveform. The higher frequency harmonics are not significant enough to affect the output. However, there may be some signals for which the response of this network will contain significant high frequency harmonics. For higher frequency signals, it may be necessary to in-

q	poles $\times 1e^{-10}$		
1	-1.7848		
2	-.61837 $\pm$ j2.0092		
3	-.46874 $\pm$ j2.0023	-25.804	
4	-.47185 $\pm$ j2.0051	-16.124	-43.373
5	-.47809 $\pm$ j2.0083	-.44396 $\pm$ j 4.8807	-6.7701
6/Act.	-.47801 $\pm$ j2.0084	-.51090 $\pm$ j5.1035	-6.2278 $\pm$ j1.7704

Table 2.4: Approximate/actual poles of Fig. 2.9.

crease the order of the approximation. Vice versa, some orders of approximation may converge more quickly for low frequency signals.

There are some networks for which the frequency response will not converge to the full range at any order of approximation. Additional methods to be discussed are used in such cases.

Figure 2.10: RLC network transient response: 2nd order approximation.

Figure 2.11: RLC network transient response: 3rd order approximation.

Figure 2.12: RLC network transient response: 4th order approximation.

Figure 2.13: RLC network transient response: 5th order approximation.

Figure 2.14: RLC network frequency response: 2nd order approximation.

Figure 2.15: RLC network frequency response: 3rd order approximation.

Figure 2.16: RLC network frequency response: 4th order approximation.

Figure 2.17: RLC network frequency response: 5th order approximation.

Figure 2.18: RLC network frequency response: 6th order approximation.

Chapter 3

Transmission Lines

At high-enough frequencies, lumped elements are no longer enough to adequately describe the delay and distortion effects on interconnect lines. Distributed elements such as lossy, coupled transmission lines become more accurate. It is possible to model distributed interconnects as lumped elements and then incorporate them into an AWE simulation[GNP91]. However, distributed models, best formulated in the frequency domain, may be incorporated directly in AWE[TN92]. Given the formulation for a linear subnetwork ϕ , I next describe how to generate the moments for a network containing these elements. Once the moments are generated, the same moment-matching technique described can be used to generate an approximate set of q poles/residues.

The inclusion of distributed elements such as lossy, coupled transmission lines, cannot be entered into the state space formulation given in the previous chapter. A more general formulation containing these elements can be given using the MNA formulation.

3.1 Linear subnetwork formulation

Consider a linear subnetwork ϕ which contains linear lumped components and N_t lossy coupled transmission line sets, with n_k coupled conductors in transmission line set k . Assuming N_ϕ nodal variables internal to the subnetwork ϕ , the MNA matrix equations for the subnetwork can be formulated as,

$$\mathbf{C}_\phi \frac{d}{dt} \mathbf{v}_\phi(t) + \mathbf{G}_\phi \mathbf{v}_\phi(t) + \sum_{k=1}^{N_t} \mathbf{D}_k \mathbf{i}_k(t) - \mathbf{e}_\phi(t) = \mathbf{0} \quad (3.1)$$

where,

$\mathbf{v}_\phi(t)$, $\mathbf{e}_\phi(t)$, $\mathbf{C}_\phi$ and $\mathbf{G}_\phi$ are as defined for (2.30),

$\mathbf{D}_k = [d_{i,j}]$ with elements $d_{i,j} \in \{0, 1\}$ and $i \in \{1, 2, \dots, N_\phi\}$, $j \in \{1, 2, \dots, 2n_k\}$ with a maximum of one non-zero in each row or column is a selector matrix that maps $\mathbf{i}_k(t) \in \mathbb{R}^{2n_k}$ the vector of currents entering the transmission line subnetwork k , into the node space $\mathbb{R}^{N_\phi}$ of the network ϕ .

If there are no transmission lines in the network then (3.1) collapses to (2.30).

A general way to describe the frequency domain equation of the transmission line subnetwork k is in the form,

$$\tilde{\mathbf{A}}_k \mathbf{V}_k(s) + \tilde{\mathbf{B}}_k \mathbf{I}_k(s) = \mathbf{0} \quad (1 \leq k \leq N_t) \quad (3.2)$$

where $\mathbf{V}_k$ and $\mathbf{I}_k$ represent the Laplace domain terminal voltages and currents of the subnetwork k respectively. $\tilde{\mathbf{A}}_k$ and $\tilde{\mathbf{B}}_k$ can be described in terms of the line parameters. I note that other formulations are possible. In the case of the $\mathbf{Y}$ or $\mathbf{Z}$ parameters we get,

$$\mathbf{V}_k(s) - \mathbf{Z}_k \mathbf{I}_k(s) = \mathbf{0} \quad (3.3)$$

or,

$$\mathbf{I}_k(s) - \mathbf{Y}_k \mathbf{V}_k(s) = \mathbf{0} \quad (3.4)$$

where $\mathbf{Y}_k = -\tilde{\mathbf{B}}_k^{-1} \tilde{\mathbf{A}}_k$ and $\mathbf{Z}_k = -\tilde{\mathbf{A}}_k^{-1} \tilde{\mathbf{B}}_k$.

Another formulation is in terms of the Transmission (ABCD) parameters $\mathbf{T}$, and differentiates between the voltages/currents at the far end of the transmission line, $(\mathbf{V}_k(d, s), \mathbf{I}_k(d, s))$, and those at the near end, $(\mathbf{V}_k(0, s), \mathbf{I}_k(0, s))$. The relationship is described by,

$$\begin{bmatrix} \mathbf{V}_k(d, s) \\ \mathbf{I}_k(d, s) \end{bmatrix} = \mathbf{T}(s) \begin{bmatrix} \mathbf{V}_k(0, s) \\ \mathbf{I}_k(0, s) \end{bmatrix} \quad (3.5)$$

Given Laplace transform of (3.1) and the formulation of (3.2), both can be compactly summarized as,

$$\begin{bmatrix} s\mathbf{C}_\phi + \mathbf{G}_\phi & \mathbf{D}_1 & \mathbf{D}_1 & \cdots & \mathbf{D}_{N_t} \\ \hat{\mathbf{A}}_1 \mathbf{D}_1^T & \hat{\mathbf{B}}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \hat{\mathbf{A}}_2 \mathbf{D}_2^T & \mathbf{0} & \hat{\mathbf{B}}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & & \mathbf{0} \\ \vdots & \vdots & \vdots & & \vdots \\ \hat{\mathbf{A}}_{N_t} \mathbf{D}_{N_t}^T & \mathbf{0} & \mathbf{0} & \cdots & \hat{\mathbf{B}}_{N_t} \end{bmatrix} \begin{bmatrix} \mathbf{V}_\phi(s) \\ \mathbf{I}_1(s) \\ \mathbf{I}_2(s) \\ \mathbf{I}_3(s) \\ \vdots \\ \mathbf{I}_{N_t}(s) \end{bmatrix} = \begin{bmatrix} \mathbf{E}_\phi \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (3.6)$$

or,

$$\tilde{\mathbf{Y}}_\phi(s) \tilde{\mathbf{V}}_\phi(s) = \tilde{\mathbf{E}}_\phi \quad (3.7)$$

3.2 Moments of distributed networks

AWE can also provide a method for generating an approximate set of poles for a distributed network, and, as will be shown, it is possible to use CFH to extract an exact set of dominant poles.

Referring to (3.7), all terms are expanded in a Maclaurin series,

$$(\tilde{\mathbf{Y}}_{\phi[0]} + s\tilde{\mathbf{Y}}_{\phi[1]} + s^2\tilde{\mathbf{Y}}_{\phi[2]} + \cdots)(\tilde{\mathbf{V}}_{\phi[0]} + s\tilde{\mathbf{V}}_{\phi[1]} + s^2\tilde{\mathbf{V}}_{\phi[2]} + \cdots) = \mathbf{E}_\phi \quad (3.8)$$

where,

$$\tilde{\mathbf{Y}}_{\phi[n]} = \frac{[\frac{\partial^n}{\partial s^n} \tilde{\mathbf{Y}}_\phi(s)]_{s=0}}{n!} \quad (3.9)$$

Equating equal powers of s gives the following recursive relationship,

$$\bar{\mathbf{Y}}_{\phi[0]} \bar{\mathbf{V}}_{\phi[0]} = \mathbf{E}_{\phi} \quad (3.10)$$

and,

$$\bar{\mathbf{Y}}_{\phi[0]} \bar{\mathbf{V}}_{\phi[n]} = - \sum_{r=1}^n \bar{\mathbf{Y}}_{\phi[r]} \bar{\mathbf{V}}_{\phi[n-r]} \quad (3.11)$$

Once again, (3.10), is the solution of the network at DC, with DC input sources given by $\mathbf{E}_{\phi}$. In order to complete the calculations however, it is necessary to generate the derivatives of $\bar{\mathbf{Y}}_{\phi}(s)$ as in (3.9). Since $\bar{\mathbf{Y}}_{\phi}(s)$ consists of the frequency domain stamps of each of the elements, the derivatives of each of the frequency port descriptions (3.2) of the elements are required.

3.3 Uniform lossy coupled transmission lines

A uniform N -conductor transmission line (Fig. 3.1) is described fully by its length and by four $N \times N$ per-unit-length line parameter matrices: $\mathbf{R}$, the resistances per unit length, $\mathbf{L}$, the inductances per unit length, $\mathbf{C}$, the capacitances per unit length, and $\mathbf{G}$, the conductances per unit length. The diagonal elements represent the self values, and the off-diagonal, the mutual coupling between the lines. The Telegrapher's partial differential equations (see Appendix A) relating the distributed voltages and currents on the transmission lines is then given by,

$$\frac{\partial}{\partial x} \mathbf{v}(x, t) = -\mathbf{R} \mathbf{i}(x, t) - \mathbf{L} \frac{\partial}{\partial t} \mathbf{i}(x, t) \quad (3.12)$$

$$\frac{\partial}{\partial x} \mathbf{i}(x, t) = -\mathbf{G} \mathbf{v}(x, t) - \mathbf{C} \frac{\partial}{\partial t} \mathbf{v}(x, t) \quad (3.13)$$

where $\mathbf{v}(x, t)$ and $\mathbf{i}(x, t)$ are the vectors of voltages and currents respectively, at

Figure 3.1: Coupled transmission line system.

different points along the lines, i.e.,

$$\mathbf{v}(x, t) = \begin{bmatrix} v_1(x, t) \\ v_2(x, t) \\ \vdots \\ v_N(x, t) \end{bmatrix} \quad \mathbf{i}(x, t) = \begin{bmatrix} i_1(x, t) \\ i_2(x, t) \\ \vdots \\ i_N(x, t) \end{bmatrix} \quad (3.14)$$

Taking the Laplace transform of (3.12) and (3.13) we obtain,

$$\frac{\partial}{\partial x} \mathbf{V}(x, s) = -(\mathbf{R} + s\mathbf{L})\mathbf{I}(x, s) \quad (3.15)$$

$$\frac{\partial}{\partial x} \mathbf{I}(x, s) = -(\mathbf{G} + s\mathbf{C})\mathbf{V}(x, s) \quad (3.16)$$

or,

$$\frac{\partial}{\partial x} \mathbf{V}(s) = -\mathbf{Z}_p \mathbf{I}(s) \quad (3.17)$$

$$\frac{\partial}{\partial x} \mathbf{I}(s) = -\mathbf{Y}_p \mathbf{V}(s) \quad (3.18)$$

where, $\mathbf{Z}_p = \mathbf{R} + s\mathbf{L}$ and $\mathbf{Y}_p = \mathbf{G} + s\mathbf{C}$.

In order to obtain the derivatives (moments)¹ with respect to s of the line equations one can either solve the differential equations (3.15) and (3.16) in x first and then obtain the derivatives of the s -domain stamp, or alternatively, expand these equations in terms of the moments with respect to s giving a spatial derivative relationship for each of the moments, which in turn can be solved recursively in x . Either of these approaches is equivalent mathematically. However, numerically and CPU-wise they will differ, depending on how the solutions are generated. They can be implemented either using the eigenvalues of the resistive line parameters or without. I will cover one of the eigenvalue approaches here and refer the reader to [BRR92b] for reference to the other equivalent method. Following that I will describe a novel and more accurate method that does not rely on eigenvalues but consumes slightly more CPU time.

3.3.1 Eigenvalue moment method

If we write (3.2) more explicitly in terms of the port voltages and currents, we get,

$$\tilde{\mathbf{A}}(s) \begin{bmatrix} \mathbf{V}(0, s) \\ \mathbf{V}(d, s) \end{bmatrix} + \tilde{\mathbf{B}}(s) \begin{bmatrix} \mathbf{I}(0, s) \\ \mathbf{I}(d, s) \end{bmatrix} = \mathbf{0} \quad (3.19)$$

the subscript k being dropped for simplicity. Let γ_m^2 be the eigenvalues of $(\mathbf{Z}_p \cdot \mathbf{Y}_p)$ with associated eigenvectors $\underline{\mathbf{S}}_m$. Let $\mathbf{S}_v$ be a matrix of eigenvectors $\underline{\mathbf{S}}_m$ (see Appendix A). $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ can be then formulated [TN92] in terms of γ_m^2 and $\mathbf{S}_v$ as,

$$\tilde{\mathbf{A}}(s) = \begin{bmatrix} \mathbf{S}_v \mathbf{E}_1 \mathbf{S}_v^{-1} & -\mathbf{1} \\ \mathbf{S}_v \mathbf{E}_2 \mathbf{S}_v^{-1} & \mathbf{0} \end{bmatrix} \quad (3.20)$$

¹The derivatives of a function are related to the moments by a factorial constant. That is $[\frac{d^n}{ds^n} F(s)]_{s=0} = n!m_n$.

$$\hat{\mathbf{B}}(s) = \begin{bmatrix} \mathbf{S}_v \mathbf{E}_2 \mathbf{S}_i^{-1} & \mathbf{0} \\ \mathbf{S}_i \mathbf{E}_1 \mathbf{S}_i^{-1} & -\mathbf{I} \end{bmatrix} \quad (3.21)$$

where,

$$\mathbf{E}_1 = \begin{bmatrix} \frac{1}{2}(e^{-\gamma_1 d} + e^{\gamma_1 d}) & 0 & \cdots & 0 \\ 0 & \frac{1}{2}(e^{-\gamma_2 d} + e^{\gamma_2 d}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{2}(e^{-\gamma_N d} + e^{\gamma_N d}) \end{bmatrix} \quad (3.22)$$

$$\mathbf{E}_2 = \begin{bmatrix} \frac{1}{2}(e^{-\gamma_1 d} - e^{\gamma_1 d}) & 0 & \cdots & 0 \\ 0 & \frac{1}{2}(e^{-\gamma_2 d} - e^{\gamma_2 d}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{2}(e^{-\gamma_N d} - e^{\gamma_N d}) \end{bmatrix} \quad (3.23)$$

and,

$$\mathbf{S}_i = \mathbf{Z}_p^{-1} \mathbf{S}_v \mathbf{\Gamma} \quad (3.24)$$

where,

$$\mathbf{\Gamma} = \begin{bmatrix} \gamma_1 & 0 & \cdots & 0 \\ 0 & \gamma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_N \end{bmatrix} \quad (3.25)$$

The length of the line is denoted by d .

In order to find a closed-form expression for the derivatives $\dot{\mathbf{A}}^{(r)}$ and $\dot{\mathbf{B}}^{(r)}$ needed for (3.9), the derivatives are expanded in terms of the eigenvectors $\underline{\mathbf{S}}_m$ and the eigenvalues γ_m^2 . Let,

$$\Phi_a = \mathbf{S}_v \mathbf{E}_1 \mathbf{S}_v^{-1} \quad (3.26)$$

Then,

$$\Phi_a \mathbf{S}_v = \mathbf{S}_v \mathbf{E}_1 \quad (3.27)$$

Differentiating (3.27) we get,

$$\Phi_a^{(n)} \mathbf{S}_v = \mathbf{S}_v^n \mathbf{E}_1 + \sum_{r=1}^n \binom{n}{r} [\mathbf{S}_v^{(n-r)} \mathbf{E}_1^{(r)} - \Phi_a^{(n-r)} \mathbf{S}_v^{(r)}] \quad (3.28)$$

Similar equations may be obtained for,

$$\Phi_b = \mathbf{S}_v \mathbf{E}_2 \mathbf{S}_i^{-1} \quad (3.29)$$

$$\Psi_a = \mathbf{S}_i \mathbf{E}_2 \mathbf{S}_v^{-1} \quad (3.30)$$

$$\Psi_b = \mathbf{S}_i \mathbf{E}_1 \mathbf{S}_i^{-1} \quad (3.31)$$

These equations give derivatives of Φ_a , Φ_b , Ψ_a and Ψ_b in terms of lower order derivatives of $\mathbf{E}_1$, $\mathbf{E}_2$, $\mathbf{S}_i$, and $\mathbf{S}_v$. Next the derivatives of $\mathbf{E}_1$, $\mathbf{E}_2$, $\mathbf{S}_i$, and $\mathbf{S}_v$ are evaluated recursively in terms of the derivatives of the eigenvalues γ_m^2 and the eigenvectors $\underline{\mathbf{S}}_m$ as follows.

$$\mathbf{E}_1^{(n+1)} = \text{diagonal} \left[-\frac{d}{2} \sum_{r=0}^n ([e^{-\gamma_m d} - e^{\gamma_m d}]^{(n-r)} \gamma_m^{(r+1)}) \right] \quad (m = 1, 2, \dots, N) \quad (3.32)$$

$$\mathbf{E}_2^{(n+1)} = \text{diagonal} \left[-\frac{d}{2} \sum_{r=0}^n ([e^{-\gamma_m d} + e^{\gamma_m d}]^{(n-r)} \gamma_m^{(r+1)}) \right] \quad (m = 1, 2, \dots, N) \quad (3.33)$$

$$\mathbf{S}_v^{(n)} = [\underline{\mathbf{S}}_m^{(n)}], \quad (m = 1, 2, \dots, N) \quad (3.34)$$

$$\mathbf{S}_i^{(n)} = (\mathbf{Z}_p^{-1})^{(n)} \mathbf{S}_v \mathbf{\Gamma} + \sum_{r=1}^n \binom{n}{r} (\mathbf{Z}_p^{-1})^{(n-r)} (\mathbf{S}_v \mathbf{\Gamma})^{(r)} \quad (3.35)$$

where,

$$(\mathbf{Z}_p^{-1})^{(n)} \mathbf{Z}_p = \sum_{r=1}^n \binom{n}{r} (\mathbf{Z}_p^{-1})^{(n-r)} \mathbf{Z}_p^{(r)} \quad (3.36)$$

$$(\mathbf{S}_v \Gamma)^{(n)} = \mathbf{S}_v^{(n)} \Gamma + \sum_{r=1}^n \binom{n}{r} (\mathbf{S}_v^{(n-r)} \Gamma^{(r)}) \quad (3.37)$$

$$\Gamma^{(n)} = \begin{bmatrix} \gamma_1^{(n)} & 0 & \cdots & 0 \\ 0 & \gamma_2^{(n)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \gamma_N^{(n)} \end{bmatrix} \quad (m = 1, 2, \dots, N) \quad (3.38)$$

Next, in order to evaluate the derivatives of γ_m and $\underline{\mathbf{S}}_m$, we use,

$$[\gamma_m^2 \mathbf{1} - \mathbf{Z}_p \mathbf{Y}_p] \underline{\mathbf{S}}_m = \mathbf{0} \quad (3.39)$$

The first derivative is obtained as,

$$\left[2\gamma_m \frac{\partial}{\partial s} \gamma_m \mathbf{1} - \frac{\partial}{\partial s} (\mathbf{Z}_p \mathbf{Y}_p) \right] \underline{\mathbf{S}}_m + [\gamma_m^2 \mathbf{1} - \mathbf{Z}_p \mathbf{Y}_p] \frac{\partial \underline{\mathbf{S}}_m}{\partial s} = \mathbf{0} \quad (3.40)$$

Equation (3.40) can be rewritten in matrix form.

$$\begin{bmatrix} \gamma_m^2 \mathbf{U} - \mathbf{Z}_p \mathbf{Y}_p & 2\gamma_m \underline{\mathbf{S}}_m \end{bmatrix} \begin{bmatrix} \frac{\partial \underline{\mathbf{S}}_m}{\partial s} \\ \frac{\partial \gamma_m}{\partial s} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial}{\partial s} (\mathbf{Z}_p \mathbf{Y}_p) \right) \underline{\mathbf{S}}_m \end{bmatrix} \quad (3.41)$$

This equation represents a system of n equations with $n + 1$ unknowns. Let the eigenvector $\underline{\mathbf{S}}_m$ be normalized such that,

$$\underline{\mathbf{S}}_m^T \underline{\mathbf{S}}_m = 1 \quad (3.42)$$

or,

$$\underline{\mathbf{S}}_m^T \frac{\partial \underline{\mathbf{S}}_m}{\partial s} = 0 \quad (3.43)$$

Using (3.43) and (3.41),

$$\begin{bmatrix} \gamma_m^2 \mathbf{1} - \mathbf{Z}_p \mathbf{Y}_p & 2\gamma_m \underline{\mathbf{S}}_m \\ \underline{\mathbf{S}}_m^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \frac{\partial \underline{\mathbf{S}}_m}{\partial s} \\ \frac{\partial \gamma_m}{\partial s} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial}{\partial s} (\mathbf{Z}_p \mathbf{Y}_p) \right) \underline{\mathbf{S}}_m \\ \mathbf{0} \end{bmatrix} \quad (3.44)$$

Higher-order derivatives are obtained recursively as,

$$\mathbf{T} \begin{bmatrix} \underline{\mathbf{S}}_m^{(n+1)} \\ \gamma_m^{(n+1)} \end{bmatrix} + \sum_{r=1}^n \binom{n}{r} \mathbf{T}^{(r)} \begin{bmatrix} \underline{\mathbf{S}}_m^{(n-r+1)} \\ \gamma_m^{(n-r+1)} \end{bmatrix} = \begin{bmatrix} \sum_{r=0}^n \binom{n}{r} [\mathbf{Z}_p \mathbf{Y}_p]^{(n-r+1)} \underline{\mathbf{S}}_m^{(r)} \\ \mathbf{0} \end{bmatrix} \quad (3.45)$$

where,

$$\mathbf{T} = \begin{bmatrix} \gamma_m^2 \mathbf{1} - \mathbf{Z}_p \mathbf{Y}_p & 2\gamma_m \underline{\mathbf{S}}_m \\ \underline{\mathbf{S}}_m^T & \mathbf{0} \end{bmatrix} \quad (3.46)$$

It is to be noted that $\mathbf{Z}_p$ and $\mathbf{Y}_p$ in (3.39) and (3.40) are assumed to be general differentiable functions of frequency. Thus their derivatives are obtained as,

$$\mathbf{Z}_p^{(r)} = \mathbf{R}^{(r)} + (s\mathbf{L})^{(r)} \quad (3.47)$$

$$\mathbf{Y}_p^{(r)} = \mathbf{G}^{(r)} + (s\mathbf{C})^{(r)} \quad (3.48)$$

In the special case where $\mathbf{R}$, $\mathbf{L}$, $\mathbf{C}$, and $\mathbf{G}$ are frequency independent,

$$\mathbf{Z}_p^{(1)} = \mathbf{L} \quad \mathbf{Z}_p^{(r)} = \mathbf{0} \quad (r > 1) \quad (3.49)$$

$$\mathbf{Y}_p^{(1)} = \mathbf{C} \quad \mathbf{Y}_p^{(r)} = \mathbf{0} \quad (r > 1)$$

Example 1

An interconnect circuit consisting of terminal networks and seven single lossless interconnects is shown in Fig. 3.2. An applied voltage consists of a pulse with 10 ps rise/fall time, 5 ns pulse width and 5V amplitude. The transient response at V_{out} is obtained by applying AWE, and is compared with HSPICE response in Fig. 3.3.

Figure 3.2: Small interconnect network with single lossless lines.

Figure 3.3: Transient response of Fig. 3.2.

Figure 3.4: Large transmission line interconnect network.

Figure 3.5: Transient response of Fig. 3.4.

Example 2

An input waveform for a 105 single lossless transmission line interconnect circuit (Fig. 3.4), consists of a pulse with 10 ps rise/fall time and a 20ns pulse width. The output waveform is calculated at V_{out} and is displayed in Fig. 3.5.

Example 3

The circuit in Fig. 3.6 consists of 3 lossy coupled interconnects. The length of the interconnects 1, 2 and 3 are 0.05m, 0.04m and 0.03m respectively. The electrical parameter matrices for the three interconnects are identical. The applied voltage is a pulse with 50 ps rise/fall time, 4ns pulse width and 1V amplitude. The transient response at V_{out} is calculated using AWE and is shown in Fig. 3.7.

3.3.2 Matrix exponential moment method

If the parameters of the line are not frequency dependent, an alternative technique may be used to extract the moments of lossy, coupled lines. This method is described in [BRR92b]. Both methods rely on a recursive propagation of the eigenvalue solution of the per-unit-length resistive line parameters and both give identical results. I refer to them as eigenvalue moment methods.

Eigenvalue moment methods can result in unwanted truncation error which increases with each new moment generated. This is due to the truncation error in the eigenmode solutions. The eigenvalues and eigenvectors are propagated recursively in the equations and any truncation error that exists in the eigenmodes is also recursively propagated. The existence of this error can be verified in two ways. First, moments generated using eigenvalue methods do not display a property to be discussed in Chapter 4, that of convergence. This is a direct result of increasing truncation error in the higher-order moments. Second, highly accurate moments of a transmission line may be generated using extended precision arithmetic[Wol91]. Extended precision

Figure 3.6: Lossy coupled transmission line network.

Figure 3.7: Transient response of Fig. 3.6.

arithmetic is a software simulation of higher precision computer architectures. While extremely inefficient, it allows the elimination of truncation errors up to any desired degree of accuracy. Extended precision programming using Mathematica[Wol91] has also experimentally demonstrated that eigenvalue methods yield increasing truncation error in the moments.

Moment-matching relies on high accuracy moments, including moments of high order (see Chapter 4). Any truncation error present will adversely affect the result of the moment-matching calculation. A third method of generating the moments of the transmission lines is described here which does not rely on the eigenmodes of the lines[CN92a]. This method, while requiring slightly more CPU time, displays improved accuracy, especially for the higher-order moments. I refer to it as the matrix exponential method of generating the moments.

Equations (3.15) and (3.16) may be written as,

$$\frac{\partial}{\partial x} \begin{bmatrix} \mathbf{V}(x, s) \\ \mathbf{I}(x, s) \end{bmatrix} = (\mathbf{D} + s\mathbf{E}) \begin{bmatrix} \mathbf{V}(x, s) \\ \mathbf{I}(x, s) \end{bmatrix} \quad (3.50)$$

where,

$$\mathbf{D} = \begin{bmatrix} \mathbf{0} & -\mathbf{R} \\ -\mathbf{G} & \mathbf{0} \end{bmatrix} \quad (3.51)$$

and,

$$\mathbf{E} = \begin{bmatrix} \mathbf{0} & -\mathbf{L} \\ -\mathbf{C} & \mathbf{0} \end{bmatrix} \quad (3.52)$$

with $\mathbf{0}$ an $n \times n$ matrix of zeros. The solution for (3.50) in terms of x is a matrix exponential[ML78],

$$\begin{bmatrix} \mathbf{V}(x, s) \\ \mathbf{I}(x, s) \end{bmatrix} = e^{(\mathbf{D} + s\mathbf{E})x} \begin{bmatrix} \mathbf{V}(0, s) \\ \mathbf{I}(0, s) \end{bmatrix} \quad (3.53)$$

The far end voltages, $\mathbf{V}(d, s)$, and currents, $\mathbf{I}(d, s)$, (where d is the length of the line)

are described in terms of the near end voltages $\mathbf{V}(0, s)$ and currents $\mathbf{I}(0, s)$ as,

$$\begin{bmatrix} \mathbf{V}(d, s) \\ \mathbf{I}(d, s) \end{bmatrix} = e^{(\mathbf{D}+s\mathbf{E})d} \begin{bmatrix} \mathbf{V}(0, s) \\ \mathbf{I}(0, s) \end{bmatrix} \quad (3.51)$$

Comparing to (3.5), we see that the transmission parameters of the line is given by,

$$\mathbf{T}(s) = e^{(\mathbf{D}+s\mathbf{E})d} \quad (3.55)$$

An important property of matrix exponentiation is that while it parallels scalar exponentiation in many ways, it does not necessarily follow the exponential addition rule. That is,

$$(\mathbf{A} \cdot \mathbf{B} \neq \mathbf{B} \cdot \mathbf{A}) \leftrightarrow (e^{\mathbf{A}}e^{\mathbf{B}} \neq e^{\mathbf{A}+\mathbf{B}}) \quad (3.56)$$

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

$$e^{(\mathbf{D}+s\mathbf{E})d} \neq e^{\mathbf{D}d}e^{s\mathbf{E}d} \quad (3.57)$$

$e^{(\mathbf{D}+s\mathbf{E})d}$ represents a complex function of s even though it appears simple on the surface. It can be the frequency domain stamp that is entered into the $\mathbf{Y}(s)$ matrix, equivalent to (3.20) and (3.21).

We wish to obtain a Maclaurin expansion of the matrix exponential,

$$e^{(\mathbf{D}+s\mathbf{E})d} = \mathbf{C}_0 + s\mathbf{C}_1 + s^2\mathbf{C}_2 + \dots \quad (3.58)$$

By the property of matrix exponential applied to an arbitrary matrix $\mathbf{A}$,

$$e^{\mathbf{A}} = \mathbf{1} + \mathbf{A} + \frac{\mathbf{A}^2}{2!} + \dots + \frac{\mathbf{A}^n}{n!} + \dots \quad (3.59)$$

Applying (3.59) on the transmission line matrix gives,

$$e^{(\mathbf{D}+s\mathbf{E})d} = \mathbf{1} + (\mathbf{D} + s\mathbf{E})d + \frac{(\mathbf{D} + s\mathbf{E})^2 d^2}{2!} + \dots + \frac{(\mathbf{D} + s\mathbf{E})^n d^n}{n!} + \dots \quad (3.60)$$

Further expanding (3.60) and collecting terms in powers of s we obtain,

$$\begin{aligned} e^{(\mathbf{D}+s\mathbf{E})d} &= [1 + \mathbf{D}d + \frac{1}{2!}\mathbf{D}^2d^2 + \dots] + \\ &s \left[\mathbf{E}d + \frac{1}{2!}(\mathbf{D}\mathbf{E} + \mathbf{E}\mathbf{D})d^2 + \frac{1}{3!}(\mathbf{D}^2\mathbf{E} + \mathbf{D}\mathbf{E}\mathbf{D} + \mathbf{E}\mathbf{D}^2)d^3 + \dots \right] + \\ &s^2 \left[\frac{1}{2!}(\mathbf{E}^2)d^2 + \frac{1}{3!}(\mathbf{D}\mathbf{E}^3 + \mathbf{E}\mathbf{D}\mathbf{E}^2 + \mathbf{E}^2\mathbf{D}\mathbf{E} + \mathbf{E}^3\mathbf{D})d^3 + \dots \right] + \dots \end{aligned} \quad (3.61)$$

Equating coefficients with (3.58) yields,

$$\begin{aligned} \mathbf{C}_0 &= \mathbf{1} + \mathbf{D}d + \frac{1}{2!}\mathbf{D}^2d^2 + \dots \\ &= \mathbf{C}_{0,0} + \mathbf{C}_{0,1} + \mathbf{C}_{0,2} + \dots \end{aligned} \quad (3.62)$$

$$\begin{aligned} \mathbf{C}_1 &= \mathbf{E}d + \frac{1}{2!}(\mathbf{D}\mathbf{E} + \mathbf{E}\mathbf{D})d^2 + \frac{1}{3!}(\mathbf{D}^2\mathbf{E} + \mathbf{D}\mathbf{E}\mathbf{D} + \mathbf{E}\mathbf{D}^2)d^3 + \dots \\ &= \mathbf{C}_{1,0} + \mathbf{C}_{1,1} + \mathbf{C}_{1,2} + \dots \end{aligned} \quad (3.63)$$

$$\begin{aligned} \mathbf{C}_2 &= \frac{1}{2!}(\mathbf{E}^2)d^2 + \frac{1}{3!}(\mathbf{D}\mathbf{E}^2 + \mathbf{E}\mathbf{D}\mathbf{E} + \mathbf{E}^2\mathbf{D}) + \\ &\frac{1}{4!}(\mathbf{D}\mathbf{E}^3 + \mathbf{E}\mathbf{D}\mathbf{E}^2 + \mathbf{E}^2\mathbf{D}\mathbf{E} + \mathbf{E}^3\mathbf{D})d^3 + \dots \\ &= \mathbf{C}_{2,0} + \mathbf{C}_{2,1} + \mathbf{C}_{2,2} + \dots \end{aligned} \quad (3.64)$$

and so on. From these results we can formulate a concise recursive description of the algorithm to generate the moments:

$$\mathbf{C}_i = \sum_{j=0}^{\infty} \mathbf{C}_{i,j} \quad (3.65)$$

where,

$$\mathbf{C}_{i,j} = \frac{(\mathbf{D}\mathbf{C}_{i,j-1} + \mathbf{E}\mathbf{C}_{i-1,j})d}{i+j} \quad (i \geq 0, j \geq 0, (i+j) \neq 0) \quad (3.66)$$

with,

$$\mathbf{C}_{0,0} = \mathbf{1} \quad (3.67)$$

and,

$$\mathbf{C}_{i,j} = \mathbf{0} \quad (i < 0 \text{ or } j < 0) \quad (3.68)$$

Figure 3.8: The matrix exponential recursion algorithm.

The recursion is illustrated in Fig. 3.8. In practice only 20-30 terms are needed in (3.65) for convergence to be attained and ∞ is just a formality of description.

Using a series expansion for a matrix exponential can cause truncation error because the $\mathbf{A}^n$ expression can, for the first few terms in (3.59), grow faster than the $n!$ expression [ML78]. This problem can also result in slow convergence because n must be large for $\mathbf{A}^n/n!$ to become insignificant.

In order to control this problem, I note that the growth of $\mathbf{A}^n$ depends on the eigenvalues of $\mathbf{A}$. If all the eigenvalues of $\mathbf{A}$ are inside the unit circle in the complex plane, then $\mathbf{A}^n$ will decay with increasing n and (3.59) will converge quickly to the desired sum. The same is true of the series in (3.61). Quick convergence is therefore achieved by controlling d in (3.60) and (3.61). With a small enough d , the eigenvalues of $(\mathbf{D} + s\mathbf{E})d$ (for a given frequency range of s) will also be small enough so as not to cause truncation errors or slow convergence. Reducing d implies truncating the transmission line. This can be achieved in one of two ways. Either by sub-dividing the line into m sections and entering each section as a separate stamp in the $\mathbf{Y}(s)$ matrix, or, more efficiently, by noting that,

$$e^{\mathbf{A}d} = e^{\mathbf{A}\frac{d}{2}} e^{\mathbf{A}\frac{d}{2}} \quad (3.69)$$

Figure 3.9: Moment comparison of a simple transmission line network.

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

$$C_0 + sC_1 + s^2C_2 + \dots = (C_0^{[\frac{1}{2}] + sC_1^{[\frac{1}{2}]} + s^2C_2^{[\frac{1}{2}]} + \dots)(C_0^{[\frac{1}{2}]} + sC_1^{[\frac{1}{2}]} + s^2C_2^{[\frac{1}{2}]} + \dots) \quad (3.70)$$

giving,

$$C_n = \sum_{i=0}^n C_i^{[\frac{1}{2}]} C_{n-i}^{[\frac{1}{2}]} \quad (3.71)$$

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

An example of the accuracy of the matrix exponential method is given in Fig. 3.9. Moments were generated via eigenvalue methods and compared to moments generated via extended precision arithmetic. Clearly, the eigenvalue moments increasingly deviate from the accurate moments at higher moment numbers. When the matrix ex-

ponential method is applied on the same network, the moments show marked increase in accuracy.

The CPU time for matrix exponential moment generation is slightly higher than for eigenvalue moment methods. However, I have found that the resultant increase in accuracy, especially to the Padé approximation, was worth it.

3.4 Nonuniform transmission lines

Owing to the presence of complex interconnect geometries, varying cross-sectional areas, interconnects crossing at different heights or due to inhomogeneous dielectric materials, interconnects sometimes need to be modelled as nonuniform multi-conductor lossy transmission lines. In this case the per-unit-length line parameters are functions of x , the distance along the length of the lines.

In this general case, the transmission line equations in the frequency domain become,

$$\frac{\partial}{\partial x} \mathbf{V}(x, s) = -(\mathbf{R}(x) + s\mathbf{L}(x))\mathbf{I}(x, s) \quad (3.72)$$

$$\frac{\partial}{\partial x} \mathbf{I}(x, s) = -(\mathbf{G}(x) + s\mathbf{C}(x))\mathbf{V}(x, s) \quad (3.73)$$

or,

$$\frac{\partial}{\partial x} \begin{bmatrix} \mathbf{V}(x, s) \\ \mathbf{I}(x, s) \end{bmatrix} = (\mathbf{D}(x) + s\mathbf{E}(x)) \begin{bmatrix} \mathbf{V}(x, s) \\ \mathbf{I}(x, s) \end{bmatrix} \quad (3.74)$$

The goal is to generate moments for these sets of lines so that they may be included in a general network $\tilde{\mathbf{Y}}(s)$ consisting of other elements[MNZ92]. In the general case, there is no closed form solution for (3.74). However, a solution generally exists and may be described in terms of a state transition matrix (STM), $\mathbf{H}(x, s)$,

$$\begin{bmatrix} \mathbf{V}(x, s) \\ \mathbf{I}(x, s) \end{bmatrix} = \mathbf{H}(x, s) \begin{bmatrix} \mathbf{V}(0, s) \\ \mathbf{I}(0, s) \end{bmatrix} \quad (3.75)$$

Equation (3.75) is a more general form of (3.53), and in the case where the parameters do not depend on x , $\mathbf{H}(x, s)$ collapses to $e^{(\mathbf{A}+s\mathbf{B})x}$.

By inspection of (3.75), STM has the following properties,

$$\mathbf{H}(0, s) = \mathbf{1} \quad (3.76)$$

$$\frac{\partial}{\partial x} \mathbf{H}(x, s) = (\mathbf{D}(x) + s\mathbf{E}(x))\mathbf{H}(x, s) \quad (3.77)$$

The STM matrix is the ABCD stamp of the transmission lines. In order to apply AWE, $\mathbf{H}(x, s)$ is expanded in term of a Maclaurin series,

$$\mathbf{H}(x, s) = \mathbf{H}_0(x) + s\mathbf{H}_1(x) + s^2\mathbf{H}_2(x) + \cdots \quad (3.78)$$

Substitution of (3.78) in (3.76) and (3.77) results in the recursive differential equations (with initial conditions),

$$\begin{aligned} \frac{d}{dx} \mathbf{H}_0(x) &= \mathbf{D}(x)\mathbf{H}_0(x), & \mathbf{H}_0(0) &= \mathbf{1} \\ \frac{d}{dx} \mathbf{H}_i(x) &= \mathbf{D}(x)\mathbf{H}_i(x) + \mathbf{E}(x)\mathbf{H}_{i-1}(x), & \mathbf{H}_i(0) &= \mathbf{0} \quad (i > 1) \end{aligned} \quad (3.79)$$

Since these equations have no closed form solution, the solutions are obtained recursively by performing a numerical integration from $x = 0$ to $x = d$ on each of the $\mathbf{H}_i$ starting with $\mathbf{H}_0$. A Runge-Kutta integration method has been used, but any other integration scheme may also be employed.

Example 1

A double line prototype chip interconnect is shown in Fig. 3.10(a)–(b). The line parameters are as listed in Table 3.1. The circuit model configuration is shown in Fig. 3.10(c). The transient response due to the input voltage of Fig. 3.10(d) was computed using AWE and is shown in Fig. 3.11 and Fig. 3.12. The results compare favorably with the spectral technique[PL89], the CPU time of the AWE solution consumed 2.9 seconds on 4/75 SUN SPARC 2 while the CPU time of the solution

Figure 3.10: Prototype chip interconnect: (a) Top view of metallization (b) Cross section (c) Circuit configuration (d) Driving voltage.

x (cm)	$c_{11} = c_{22}$ (pF/cm)	$c_{12} = c_{21}$ (pF/cm)	$l_{11} = l_{22}$ (pF/cm)	$l_{12} = l_{21}$ (pF/cm)
0.0 - 1.0	1.84	-0.090	1.96	0.23
1.14	1.76	-0.073	2.04	0.22
1.29	1.60	-0.050	2.22	0.20
1.43	1.44	-0.035	2.43	0.19
1.57	1.28	-0.024	2.69	0.17
1.71	1.12	-0.017	3.01	0.17
1.86	0.96	-0.012	3.44	0.16
2.0 - 3.0	0.88	-0.009	3.71	0.16

Table 3.1: Parameters of a prototype chip interconnect.

from the spectral method was 199 seconds on a VAX-11/ 750 VMS. The spectral method uses a convolution of a system of equations the size of which grows rapidly as the number of coupled lines is increased. Therefore the CPU gain of the AWE method over the spectral method will be higher as the number of coupled lines is increased.

Example 2

A circuit containing seven nonuniform transmission lines is shown in Fig. 3.13. The line parameters are tabulated and represent tapered interconnection geometries. The response of AWE is compared with the published numerical inversion of Laplace transform (NILT) based method[GN92] in Fig. 3.14.

Figure 3.11: Solution of Fig. 3.10 at active line.

Figure 3.12: Solution of Fig. 3.10 at quiescent line.

Figure 3.13: Interconnect model with 7 nonuniform transmission lines.

Figure 3.14: Transient response of Fig. 3.13.

3.5 Miscellaneous combinations

Models of interconnect networks may be generated by various sources. Generally three dimensional field analyzers are used to generate lumped components, or at times uniform and even nonuniform transmission line models with their respective parameters. Other models may be based directly on known process parameters and lengths of interconnections, or on estimates. It is important to note that the formulations given above can include all of these models *simultaneously*. Networks may contain lumped elements, lossless delay lines, lossy transmission lines, coupled transmission lines and even nonuniform transmission lines in any combination. Any other element or n -port subnetwork for which a Maclaurin series can be generated can also be included in the network $\tilde{\mathbf{Y}}(s)$. Once the moments of the element or subnetwork are generated, they are used in (3.10) and (3.11). These equations are isolated from the process by which the moments are generated. This general method is called partitioning and will be covered in Chapter 5.

Chapter 4

Padé Approximations

The approximation that was given in (2.24) conforms to a particular type of approximation known as a Padé rational approximation[Bak75]. Padé approximations have long been a subject of interest in numerical analysis¹. Although several forms of approximating functions exist, including polynomial and exponential approximations, Padé approximation employs a rational approximant. Padé approximations may not always be well-suited to general mathematical functions, but their applicability to network transfer functions is particularly appealing since these functions are in a rational form (as is a network transfer function), and the poles of the Padé approximant approach the poles of the system rather well.

¹The foundation of Padé approximants was laid by Cauchy in 1821. Based on his work, Jacobi (1846) is credited with first discovering and describing them. Later Frobenius (1881) gave a detailed investigation of the properties of Padé approximants. Padé (1892) arranged the approximants in a Padé table and investigated the properties of this table and specifically the table for e^x . While finding their place in several scientific and engineering endeavors such as Physics, Chemistry and Mathematics, they were first applied to extract a small set of dominant poles from a set of linear system equations in systems research. The method was called model reduction[BL72]. Baker[Bak75] describes their first application to circuits and inverse Laplace. In circuit oriented analysis, several researchers have found Padé approximations useful in their work, including J. Vlach[VS83], F. Y. Chang[Cha92a, Cha92b], E. C. Chang and S. Kang[CK92], and S. Lin and E.S. Kuh[LK92].

The residue strength of the typical electrical network poles is concentrated in a small region of the complex plane, making the approximation practical and accurate for these types of networks. Added to this property is the large CPU advantage of using this type of approximation compared to full network analysis. Padé approximations are useful when it is more efficient to obtain the moments of a function rather than to evaluate the function in closed form. This is the case with linear electrical circuit network transfer functions.

For a given set of moments, there are an infinite number of these approximations possible. Selecting the right approximation is an important question that needs to be addressed. Further, we need to know how to improve on an approximation while avoiding truncation errors as much as possible. Therefore, in order to work with and understand the properties of moment-matching, it is necessary to have an overview of the theory and numerical properties of Padé approximations. This material is covered in more detail in [Bak75] and [BGM81]. However, while the general theory may be of interest, this chapter will cover points that are useful to forming and understanding moment-matching approximations for electrical and interconnect network analysis.

4.1 The rational approximation

Padé approximations begin with a Maclaurin series expansion of a function,

$$F(s) = m_0 + sm_1 + s^2m_2 + s^3m_3 + \cdots = \sum_{n=0}^{\infty} s^n m_n \quad (4.1)$$

where the coefficients of the expansion are also known as moments. The series is then matched to a rational polynomial approximation where the Maclaurin expansion of the approximant matches the original Maclaurin series up to a set number of moments. The Maclaurin moments contain information of the function about the origin. In the frequency domain they represent the low frequency behavior of a network transfer function.

Definition 1 An L, M Padé approximant to $F(s)$ is denoted by,

$$[L/M] = \frac{\hat{P}_L(s)}{\hat{Q}_M(s)} \quad (4.2)$$

where,

$$\hat{P}_L(s) = a_0 + sa_1 + s^2a_2 + \cdots + s^La_L \quad (4.3)$$

and,

$$\hat{Q}_M(s) = b_0 + sb_1 + s^2b_2 + \cdots + s^Mb_M \quad (4.4)$$

The coefficients of $\hat{P}_L$ and $\hat{Q}_M$ are obtained uniquely from the coefficients of the Maclaurin series expansion when the normalizing² condition of

$$\hat{Q}_M(0) = 1 \rightarrow b_0 = 1 \quad (4.5)$$

is imposed. Since there are $L + M$ unknowns, it is required that $L + M$ moments be matched. If we cross multiply (4.1) with (4.2) we get,

$$(m_0 + sm_1 + s^2m_2 + s^3m_3 + \cdots + s^{L+M}m_{L+M}) \times (b_0 + sb_1 + s^2b_2 + \cdots + s^Mb_M) = a_0 + sa_1 + s^2a_2 + \cdots + s^La_L \quad (4.6)$$

Equating the coefficients for powers $s^{L+1}, \dots, s^{L+M}$ and remembering that $b_0 = 1$ yields,

$$\begin{bmatrix} m_{L-M+1} & m_{L-M+2} & \cdots & m_L \\ m_{L-M+2} & m_{L-M+3} & \cdots & m_{L+1} \\ \vdots & \vdots & \ddots & \vdots \\ m_L & m_{L+1} & \cdots & m_{L+M-1} \end{bmatrix} \begin{bmatrix} b_M \\ b_{M-1} \\ \vdots \\ b_1 \end{bmatrix} = - \begin{bmatrix} m_{L+1} \\ m_{L+2} \\ \vdots \\ m_{L+M} \end{bmatrix} \quad (4.7)$$

If $j < 0$, $c_j = 0$. Equation (2.21) is obtained when setting $\omega = M - 1$, i.e. when it is assumed that the degree of the numerator is one less than the degree of the

²Because $[L/M]$ is a rational function, it can be divided by an arbitrary constant.

denominator. The numerator coefficients can be found by equating powers $s^0, \dots, s^L$,

$$\begin{aligned} a_0 &= m_0, \\ a_1 &= m_1 + b_1 m_0, \\ a_2 &= m_2 + b_1 m_1 + b_2 m_0, \\ &\vdots \\ a_L &= m_L + \sum_{i=1}^{\min(L,M)} b_i m_{L-i} \end{aligned} \quad (4.8)$$

With many possible choices of L and M for an $[L/M]$ approximation, it is possible to build an infinite array of all the approximants.

Definition 2 *The Padé table is a two-dimensional infinite array of all approximants to a given Maclaurin series for different selections of L and M . The horizontal sequence is given by increasing L and the vertical sequence is given by increasing M . The diagonal sequence is given when $L = M$ and e.g. off-diagonal sequence is given when $L = M - J$ for some J .*

M ↓	L →	0	1	2	...
0		[0/0]	[1/0]	[2/0]	...
1		[0/1]	[1/1]	[2/1]	...
2		[0/2]	[1/2]	[2/2]	...
⋮		⋮	⋮	⋮	⋮

Table 4.1: The Padé table.

A Padé table is illustrated in Table 4.1. For example, if we approximate the exponential function, the Maclaurin expansion is readily available as,

$$e^s = 1 + s + \frac{s^2}{2!} + \dots + \frac{s^n}{n!} + \dots \quad (4.9)$$

Forming $[L/M]$ approximations for different possibilities of L and M results in Table 4.2.

$M \downarrow$	$L \rightarrow$	0	1	2	...
0		$\frac{1}{1}$	$\frac{1+s}{1}$	$\frac{2+2s+s^2}{2}$	...
1		$\frac{1}{1-s}$	$\frac{2+s}{2-s}$	$\frac{6+4s+s^2}{6-2s}$	...
2		$\frac{2}{2-2s+s^2}$	$\frac{6+2s}{6-4s-s^2}$	$\frac{12+6s+s^2}{12-6s+s^2}$	...
$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\ddots$

Table 4.2: Part of the Padé table of e^s . [Padé, 1892]

Once the numerator and denominator coefficients are obtained, they may be decomposed into the form,

$$\frac{\hat{P}_L(s)}{\hat{Q}_M(s)} = \hat{c} + \sum_{i=1}^M \frac{\hat{k}_i}{s - \hat{p}_i} \quad (L \leq M) \quad (4.10)$$

An alternate approach to (4.8) is to factor the denominator in terms of its poles and then obtain the residues of the poles as in (2.29) and (2.39). But in this case only the first q or $q + 1$ moments will be guaranteed to be matched to the original function.

When forming an approximation it is important to select L and M appropriately, which implies selecting the order of the approximation. Each of the approximations will produce a slightly different result.

4.2 Accuracy

No simple closed-form formula is known for the error of an approximation as a function of the value s . This is due to the complicated functional dependence of the error on the truncation error of the moments, which in turn depends on the actual location of

the poles and the respective value of their residues. This property can be seen when expanding an actual system function (2.14) in terms of its moments (assuming no direct coupling c),

$$\begin{aligned} \sum_{i=1}^{N_p} \frac{k_i}{s - p_i} &= \left(\sum_{i=1}^{N_p} \frac{-k_i}{p_i} \right) + s \left(\sum_{i=1}^{N_p} \frac{-k_i}{p_i^2} \right) + s^2 \left(\sum_{i=1}^{N_p} \frac{-k_i}{p_i^3} \right) + \dots \\ &= m_0 + sm_1 + s^2 m_2 + \dots \end{aligned} \quad (4.11)$$

If we examine the first moment, we can observe that truncation error in a finite precision machine may start to occur when $|\frac{k_i}{p_i}| > |\frac{k_j}{p_j}|$ for some i, j . For the second moment, it may occur when $|\frac{k_i}{p_i^2}| > |\frac{k_j}{p_j^2}|$ for some i, j . This truncation error will reduce the contribution of a particular pole to the moments. If numerical truncation occurs on a pole in the first moment, then the truncation becomes greater for the second moment and greater still for successive moments. Given two poles with residues of equal moduli³, the pole with the larger self modulus (farther from the origin) will suffer more truncation error. This becomes increasingly pronounced with increasing moments. Given one pole that is closest to the origin, $|p_s| = \min(p_1, p_2, \dots, p_{N_p})$, and its residue, k_s ,

$$\lim_{n \rightarrow \infty} \sum_{i=1}^{N_p} \frac{k_i}{p_i^n} = \frac{k_s}{p_s^n} \quad (4.12)$$

The single pole closest to the origin is said to "dominate" the moments. A similar relation holds true for two or more poles equidistant but closest to the origin, as in a complex conjugate pair. The contribution of the farther poles becomes increasingly diminished in the higher moments, until it is nonexistent numerically. The truncation error on a pole, is highly dependent on the position of the other poles with respect to the nearest poles as well as on the size of the residues[BGM81].

However, there is no knowledge of the actual poles and residues in large networks as this is the kind of calculation that we are attempting to bypass. Therefore there is no way to ascertain the truncation error by simply inspecting or manipulating a single

³The modulus of a complex number is its absolute scalar value and is equal to its distance from the origin in the complex plane.

set of moments. It will neither be possible to precisely determine how many poles make up that set of moments, nor to determine their exact location in the complex plane or the value of their residue. A Padé approximation however, will have the property of being able to give information about *some* of the poles of the network, as well as matching the value of the transfer function and its derivatives at the point of expansion. It will be shown in Chapter 6, how the limitations of a single Padé approximation may potentially be bypassed to build an error criterion for the derived poles and residues using multi-point expansions.

Using a single Padé approximation, we must therefore satisfy ourselves with empirical results gained through many applications of these approximants to electrical networks, with the careful weighing of results that must go along with any empirical conclusions. Since the approximation is performed on a frequency-domain function, it is also important to get an idea of the time-domain error of an approximation.

4.2.1 Properties

Padé approximations attempt to extract pole/residue information from the moments. The detection of a particular pole will depend critically on the truncation error that exists in that pole. The larger the truncation error, the more approximate will be the generated pole. The poles closest to the point of expansion are detected best. The error will depend on the order of the approximation. If few poles are generated for a system containing many poles, most will not be seen or detected. However, it is the property of electrical networks that they are generally characterized by a few dominant poles. Generally, a Padé approximation is used to extract far fewer approximate poles than are actually in the network.

Sometimes, when a pole is far from the point of expansion, it will result in such a large truncation error, that an approximate pole will appear on the other side of the complex plane in a Padé approximation. That is, an approximate pole will be detected in the right-half plane for a real pole that exists in the left-half plane.

Known stable systems have all poles in the left-half plane. A pole in the right-half plane implies an unstable solution because (2.19) will grow without bounds. This problem of instability needs to be addressed when performing Padé approximations.

The matrix, $\mathbf{m}_h$ in (4.7) is known as the Hankel moment matrix. Its condition and invertibility is also a property of the distribution of actual poles in the network. The matrix can be shown to be singular when attempting to extract more poles than actually exist in the network. Since this is not the purpose of performing such an approximation, we can discount this type of problem. However, the same effect may also happen numerically when trying to extract more poles than those that are numerically prominent in the moments. Those poles that are far away or with comparatively small residues, will not have much effect on the moments and therefore can cause the Hankel moment matrix to be ill-conditioned or non-invertible.

Further, using standard precision computing (32-bit and 64-bit double words) the Hankel moment matrix, $\mathbf{m}_h$ is generally invertible for, at most 8 to 12 poles. This is again due to truncation error due to poles being a different distance from the origin. So, practically, extracting many more poles than this results in very inaccurate poles and right-half plane poles for stable systems.

Some of the properties of Padé approximations are derived from the properties of a Maclaurin series. A Maclaurin series in s has a radius of convergence $|s| = R_{con}$ beyond which the series is not finite for any substituted value of s . In the case of a function that has singularities, such as a network transfer function with poles, the radius of convergence is equal to the distance of the nearest pole. We can see this from (4.11). Here,

$$R_{con} = \min(|p_1|, |p_2|, \dots, |p_q|) \quad (4.13)$$

because for the n -th term,

$$\lim_{n \rightarrow \infty} s^n \frac{k}{p^{n+1}} \rightarrow \infty \quad (|s| > R_{con}) \quad (4.14)$$

However, Padé approximations have a powerful property of extending information

about the function beyond the radius of convergence of the original Maclaurin series. They appear to give more information than the original Maclaurin series.

I now summarize several empirical and mathematical properties of Padé approximations. These properties are essential in understanding the kind of error generated by the approximation and enable us to improve on an approximation.

Property 1 *Padé approximants are completely accurate at the point of expansion and lose accuracy with increasing distance from the point of expansion.*

This property stems from the moment matching done by a Padé approximation. The value of the function and several of its low-order derivatives are matched by the approximating function. However, there is a numerical limit to how far afield an approximation can predict the actual function from its derivatives at one point. That predictive power of the derivatives numerically decays when the distance from the point of expansion is increased.

Property 2 *Padé approximants' accuracy depends directly on the nearest singularity (pole) or the nearest equidistant singularities. Accuracy deteriorates more quickly with distance the closer that singularity (or singularities) is to the origin.*

The nearest singularity or singularities (as in complex conjugate pairs) will dominate the moments causing truncation error in the poles farther away. The relatively closer a pole is to the origin the more it will dominate the moments and the more it will cause truncation error in the other poles. This will cause accuracy to drop off more quickly with distance. One says that the close pole "blocks" the view of the other poles. This property is important in the case of stiff systems, where a pole exists very close to the origin, while other poles are relatively farther off. This large variation between poles will cause problems for a Padé approximation. Frequency shifting (Chapter 5) is one way of dealing with such effects.

For some of the properties that follow I will need the following definition:

Definition 3 *A pole p_j of strength k is one whose residue modulus is $k = |k_j|$.*

Property 3 *Given a system pole of strength k , if a Padé approximation detects it with a particular relative error, then any system poles closer to the expansion point and with the same or larger strength will be detected with less relative error.*

This property is a direct result of the impact of a pole on the moments. If a pole is detected, then certainly a smaller pole with the same or greater residue modulus will make a greater contribution to the moments and will also be detected with less error.

Property 4 *In a horizontal sequence of increasing L ,*

$$\lim_{L \rightarrow \infty} [Q_{[L/M]}(s)] \rightarrow \prod_{j=1}^M \left(1 - \frac{s}{p_j}\right) \quad (1.15)$$

where p_j are the actual poles of the network arranged in order of increasing modulus: $|p_1| \leq |p_2| \leq \dots \leq |p_M| \leq \dots$ and $|p_M| \neq |p_{M+1}|$. A horizontal sequence in the Padé Table (increasing L , fixed M) will converge to a polynomial with some of the poles of the system as its roots. The poles will be the ones closest to the expansion point.

This is one of the most useful and important provable properties of Padé approximation, and is referred to as the horizontal sequence convergence property. By increasing L while leaving M constant, the poles of the resulting denominator converge to the actual system poles. This is a useful property when we need to extract the actual system poles rather than approximating poles.

Property 5 *Some poles will be so far away from the point of expansion that they will appear as right-half plane poles in the approximation of stable systems.*

This property, already mentioned briefly, is known as instability. There has been much discussion about the instability of Padé approximations. The problem can be alleviated by using partial Padé approximations or completely bypassed by using multi-point expansions.

Property 6 *Padé approximations are very sensitive to the moments. A small truncation error in the moments, can cause large deviation in the approximate poles or residues generated.*

Numerical accuracy is critical to a proper Padé approximation. The accuracy of the approximation will be no greater than the accuracy of the moments, and generally much less. Often, this is the primary problem of carrying out a Padé approximation. For example, eigenvalue moment methods generate increasing truncation error in the moments (Chapter 3). As a result these moments do not converge using the horizontal sequence convergence property. Further, poles are much more perturbed from their original value than is indicated by the small error contained in the moments.

Property 7 *Padé approximations may generate extremely small “parasitic” poles (close to the origin) with very small residues. The moduli of these poles are smaller than the moduli of the p_s , the actual poles nearest the origin.*

This is a result of numerical error in the moments and is not a large problem as the smallest poles may be known with certainty.

Property 8 *Padé approximations do not give good results when there exist poles farther away and of larger strength than those closer to the origin.*

The contribution to the moments of the farther poles is mixed. These poles may contribute more to the overall first moment, but in the higher order moments, their contribution is diminished exponentially. A Padé approximation has difficulty distinguishing far poles with large residues from closer poles with smaller residues. This type of system is typically in band-pass or high-pass filters.

Property 9 *Generally a diagonal or slightly off-diagonal Padé approximation is sufficient for realizing the maximum accuracy of a Padé approximation. However, on some occasions, such as in stiff systems, this may not be the case.*

If no extra CPU time is to be expended in searching the Padé table, then it is safest to stick with a diagonal sequence of the Padé table[Hua90]. Starting with a small approximation (2-3 poles) and working up to a larger one (8-10 poles) will show convergence of the approximation to the actual transfer function and the approximate

poles to the nearby system poles. When extra accuracy is required, it is often useful to perform a search of the Padé table (Chapter 5) in order to choose the best approximation. This will entail more CPU cost.

4.2.2 Error analysis

Given the previous properties, I can now describe the kind of error generated by a Padé approximation of a system function. It is important though to first define the context of applicability: electrical network transfer functions.

Definition 4 *Dominant poles in a set of poles and residues forming a system response in the frequency domain.*

$$H(s) = \sum_{j=1}^{N_p} \frac{k_j}{s - p_j}$$

or in the time domain,

$$h(t) = \sum_{j=1}^{N_p} k_j e^{p_j t}$$

are those poles in the set that are closest to imaginary axis and that have large residue moduli.

I note that this definition is a qualitative one. Of interest are the poles and respective residues that are the major contributors to the system response, such that if we remove the non-dominant poles and their respective residues from the response, it would remain essentially the same. Poles with small residues, or poles distant from the imaginary axis will not significantly affect the system response.

Interconnect transfer functions are generally low-pass filters in the frequency domain. That is, in terms of poles and residues, the dominant poles are of low frequency (lower down the $j\omega$ axis) with large residues, giving a band pass at low frequencies and a drop off at higher frequencies.

2

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

PRECISIONSM RESOLUTION TARGETS

Specifically, RC and RL networks have all poles directly on the negative real axis. Using one Padé approximation is very appropriate in this case, as the approximation will directly detect poles that are closest to the origin but on the negative real axis. These are the dominant poles of the network. The accuracy of a Padé approximation will deteriorate with distance from the origin, but in these networks it is inconsequential as the more distant poles along the negative real axis are unimportant. The order of the approximation needed will be low, and increasing the order of the approximation will not add to the accuracy (see Example 1 in Chapter 2). With some circuits, more than two poles may dominate and it may be necessary to increase the order of the approximation for convergence of the waveform. In some untypical RC or RL circuits, poles close to the origin may have small residues while those farther away on the negative real axis have larger residues, making for a difficult Padé approximation (property 8). In this case, frequency shifting (Chapter 5) often enhances an approximation. This analysis also applies to RC networks with transmission lines that contain only R and C per unit length, or RL networks with transmission lines that contain only R and L per unit length.

Complications arise in general RLC circuits, or especially, LC circuits. These circuits contain complex conjugate poles, which in some circuits may be far from the origin. If the network is strongly resistive in blocking out the effects of the L, i.e. small resistances in parallel with L and large resistances in series with L, then the circuit will operate essentially as an RC circuit with some weak inductance effect. This is an accurate model of many interconnect networks with inductive effects. The network may have some complex conjugate poles but they will be close to the negative real axis. Essentially, they will be detectable by a Padé approximation near the origin. The same principle applies to an RLC network with resistances which are blocking the capacitive effect. The network will essentially operate as an RL network with some capacitive effects. Again, the poles will be on or close to the negative real axis.

If the network is weakly resistive, i.e. the resistances in parallel with capacitors or inductors are large and/or resistances in series with those elements are small, the circuit will act essentially as the same network without resistance, i.e. an LC network. These will have high Q poles close to the imaginary axis and at times, far from the origin. When performing a Padé approximation near the origin on these types of networks, the approximation will capture the poles close to the origin but may miss poles higher up the axis. The error in linear simulation, however, is mitigated for two reasons.

Firstly, the residues of the approximated poles will be slightly different than the actual residues. This is because once poles are extracted, the matching is done in (2.29) to the moments. If there are significant poles missing from the approximation, the residues of the captured poles will be compensated. The waveform in such a network will essentially match the larger low frequency oscillations but may be missing some smaller high frequency components, i.e. if the essential response energy is at low frequencies then an approximation will capture most of the signal energy. However, as most of the energy shifts to high frequencies, an approximation may become less accurate.

Secondly, in simulation applications, if the general consideration is either a ramp response or a step response, then the $H(s)$ is multiplied by a factor $1/s^2$ or by $1/s$ respectively. This means that the frequency response is attenuated at high frequencies where these multipliers become small as in Table 2.1. The resulting residues of high frequency poles also become small. The attenuation of the residues is greater the higher up in frequency we find the pole. This means that the dominant poles will be increasingly found at lower frequencies. In many of these cases therefore a Padé approximation will give an accurate response. However, even in the case of attenuation of frequency spectrum at high frequencies, poles near the origin may sufficiently obscure dominant poles further up the frequency axis, so as to cause moderate error in the response. This is especially true of some distributed interconnect networks that

may have many poles along the imaginary axis. In many of these cases though, the approximation will be seen to follow the major flow of the actual waveform but miss some of the high-frequency (but smaller) oscillations. Signals with relatively long rise times will cause the unit step and unit ramp in Table 2.1 to be more prominent than the exponential response of the circuit. In this case a Padé approximation will also be accurate.

High pass or band pass networks will be a problem for a Padé approximation centered near the origin. In these networks, the residues of high frequency poles (far from the origin) are of larger moduli than those of poles near the origin at low frequencies. A Padé approximation will not be able to capture these poles well (property 8). Therefore, a single Padé approximation near the origin is not recommended for high pass or band pass structures where most of the response is concentrated far above DC. These networks however, are not typical of interconnect networks.

There are several methods to increase the accuracy of an approximation in specific situations (Chapter 5). Poles may be selected from nodes other than the one simulated; the network may be partitioned and an approximation generated for each partition separately; long delays may be extracted out of transmission lines; the network poles may be mapped to bring them closer to the origin; a selection of the best approximation in the Padé table may be made. In general, however, even with accuracy enhancements, using one Padé approximation does not give a guarantee of the error of the approximation, although accurate results may be extracted most of the time. This can be seen by comparing the frequency response of one Padé approximation with the actual frequency response of an arbitrary network. In the general case, the frequency response will be matched only in certain regions on the frequency axis. It will not be possible to determine *a-priori* the highest frequency of match. In order to guarantee accuracy up to a pre-defined highest frequency, f_{max} , multi-point moment matching (Chapter 6) is needed. If there is a reasonable certainty about the response and/or type of network, or the usefulness of accuracy-enhancing methods,

then a single expansion is sufficient.

A single Padé approximation is therefore recommended for RC, RL circuits, for RLC networks that operate essentially as RC or RL circuits, and for outputs acting as low pass filters. Transmission line networks are also accommodated if the electrical length is not too long or the lines are weakly capacitive or weakly inductive. If the delay of the lines is long, then the method of characteristics (Chapter 5) can be applied together with a partitioning scheme.

In nonlinear simulation (Chapter 7), any small error in high frequencies may cause large error in low frequencies. This is known as the spread spectrum effect. For this reason, in nonlinear networks, it is important to obtain a moment-matching approximation that is as accurate as possible. Generally, in order to guarantee the accuracy of a frequency band, multi-point moment matching must be used.

4.3 Laurent expansion about ∞

AWE can extend the application of a single Padé approximation to an approximation containing moments from two points. The second point is conveniently chosen to be at $s = \infty$. This expansion is known as a Laurent series[HRR90].

Expanding (2.5) at $s = \infty$ gives,

$$\mathbf{Y}(s) = s^{-1}\mathbf{c}^T\mathbf{b} + s^{-2}\mathbf{c}^T\mathbf{A}\mathbf{b} + \cdots + s^{-n-1}\mathbf{c}^T\mathbf{A}^n\mathbf{b} + \cdots \quad (4.16)$$

or,

$$\mathbf{Y}(s) = s^{-1}\mathbf{M}_{-1} + s^{-2}\mathbf{M}_{-2} + \cdots + s^{-n}\mathbf{M}_{-n} + \cdots \quad (4.17)$$

with,

$$\mathbf{M}_{-i} = \mathbf{c}^T\mathbf{A}^{i-1}\mathbf{b} \quad (4.18)$$

This corresponds to expanding an output transfer function in (2.14) about $s = \infty$,

$$\begin{aligned} H(s) &= s^{-1}\left(\sum_{j=1}^{N_p} k_j\right) + s^{-2}\left(\sum_{j=1}^{N_p} k_j p_j\right) + \cdots + s^{-n}\left(\sum_{j=1}^{N_p} k_j p_j^{n-1}\right) + \cdots \\ &= s^{-1}m_{-1} + s^{-2}m_{-2} + \cdots + s^{-n}m_{-n} + \cdots \end{aligned} \quad (4.19)$$

These “negative” moments can also be incorporated in a Padé approximation. A Padé approximation using combined moments from the origin and from infinity are known as a two-point Padé approximation. Normally Padé approximations use the moments from an expansion about the origin. However, since the moment matrix in (4.7) contains a consecutive set of moments, it is only natural to extend the set to the moments of negative powers of s [McC75], [JM80]. When the Hankel moment matrix, $\mathbf{m}_h$, was defined, it was stipulated that $m_j = 0$ if $j < 0$. Removing that condition allows the inclusion of “negative moments”.

Numerically, the resulting function will also be matched at $s = \infty$ (i.e. the circle at infinity surrounding the origin in all directions, not a point but a set of points), as well as $s = 0$. By implication, the more the proportion of moments is derived from $s = 0$, the more the approximate function’s accuracy will conform to the function at that point, and vice versa for $s = \infty$. Large poles (farthest from the origin) can be detected more accurately by using more moments about $s = \infty$, but it might not always be useful to do so. Poles that are far into the left-half plane are not dominant and therefore not of interest. Poles far from the origin but near the imaginary axis are of interest. However, they will only be detected if no other poles are larger (farther from the origin).

The method of including “negative” moments has been tried with mixed success. When applied to small networks, especially as partitions (Chapter 5) or to certain types of other small networks, the result can be an improvement over a one point Padé approximation. Otherwise, the results are not always useful. The method does not prevent the appearance of right-half plane poles for stable systems. Further, it is not clear how many moments to take from each expansion.

The most useful technique is to include just the first negative moment in (2.29) which is equivalent to the extra condition that was stipulated at $t = 0^+$. In this way most of the moments are matched at $s = 0$ (bulk of the signal energy for low pass networks) but one matching is done at $s = \infty$ which is equivalent to increasing the

accuracy of the transient response at $t = 0$.

A single Padé approximation, with its limitations, can be very useful for analysis of a lumped interconnect circuit. Large RC and RL networks can be analyzed quickly and efficiently. Other RLC networks and general linear networks with transmission lines can, under certain conditions also be handled well. Problems may appear when analyzing networks with high Q effects from signal coupling and reflections. When these networks are stimulated with short rise time pulses, these effects show up in the waveform in the form of ringing overshoots or undershoots. In this case a single expansion Padé approximation is generally not enough.

I describe in the next chapters several techniques to improve a Padé approximation and a method to gauge the accuracy of an approximation using multi-point moment matching.

Chapter 5

Accuracy Improvements

While AWE can be remarkably accurate in generating approximations for several types of linear circuits, unwanted errors can sometimes result. These errors are a direct result of the properties of Padé approximations. I present below some methods to address the dual issues of instability and loss of accuracy.

5.1 Higher precision programming

Padé approximations are notoriously sensitive to the moments. Any small deviation in the moments will cause large deviations in the poles, increasingly so farther from the point of expansion. This is due to the already existing truncation error in poles far from the point of expansion which is aggravated by any deviation in the moment values.

One way to address the truncation error of the poles and residues is to increase the operating precision. Padé approximations must be done on as high precision machines as possible. Today, machine architectures for engineering workstations are designed to operate in 32-bit floating point modes and 64-bit double modes. Padé approximations should be performed in full 64-bit floating point. Lower precision, such as 16-bit machines or 32-bit words are not sufficient. The accuracy of the

moments will not be enough for a reasonable application.

New architectures that are expected on the market will use full 64-bit and 128-bit double double machine words. Using this type of workstation architecture will significantly increase the accuracy distance about an expansion point. This can be verified by programming AWE using multiple precision software. That is, routines exist which will simulate multiple precision machines on software. While the CPU times increase dramatically for such simulations, it is possible to verify the accuracy levels of the approach. I have programmed multiple precision programming using Mathematica[Wol91]. The results showed that AWE accuracy was increased for a broader circle about the expansion point and that poles were detected more accurately.

5.2 Moment scaling

The moments of the system can be calculated using the given values of inductances and capacitances. However these values may be very small compared to the order of the values of resistances. The moments, as a result will display a large shift of order between one moment and the next. In the case of typical networks, one moment could be 10^9 larger than the next. When generating the moments, the limit of machine exponent size may be reached by generating moments that are either too small or too large. As well, when attempting to solve the Hankel moment matrix, $\mathbf{m}_h$, in (4.7), numerical error may result due to the widely varying values of the moments.

To rectify this problem it is useful to scale[Pil89] all the inductive/capacitive values by γ . Typically $\gamma = 10^9$ is a good scaling value. A nH would become H and pF would become mF. This results in a scaling of the frequency as well as time: 1 GHz becomes 1 Hz and 1 ns becomes 1 second.

Additional scaling to this first kind can be performed on the moments themselves once generated. This is equivalent to scaling the elements before generating the moments, except that here, the value of the scale is selected to minimize numerical

problems with the Hankel moment matrix. A convenient scale factor can be chosen by selecting the highest-order moment generated and its growth with respect to m_0 . So,

$$\gamma = \left(\frac{m_0}{m_n} \right)^{\frac{1}{n}} \quad (5.1)$$

The result is a frequency scaling of γ .

$$H(\gamma s) = m_0 + (\gamma m_1)s + (\gamma^2 m_2)s^2 + \cdots + (\gamma^n m_n)s^n + \cdots \quad (5.2)$$

5.3 Padé table analysis

As was given in Definition 2, the set of all possible Padé approximations using different L 's and M 's is known as the Padé table. Generally, a diagonal search of the Padé table is used, by increasing L and M simultaneously. In some low-pass circuits, as little as 2 or 3 pole approximations are sufficient for an accurate response. The waveform can be seen to converge by this low order. Other approximations may take a higher order to converge. At some high order however, all approximations begin to generate right-half plane poles for known stable systems.

A more general approach attempts to analyze different approximations from the Padé table and to compare them[Hua90],[CN92c]. While this may appear to be CPU intensive, there is a method of decreasing the expense to the point of being useful for large systems. The extra expense is generally less than an LU factorization of the network. The goal of the generalized method is to choose one entry, one selection of L and M , of the Padé table that yields the best set of approximate poles. Two selection criteria are needed for this purpose. The final approximation is guaranteed stable for a stable system and displays generally increased accuracy compared to an arbitrary Padé approximation. The approach relies on Padé approximations only as a starting point for generating the single most accurate approximation. No assumption is made as to the accuracy of a particular choice of approximation.

Definition 5 Given the network poles $\{p_j\}$, ($j = 1, 2, \dots, N$), the pole(s) of smallest modulus, $\{p_s\}$, are defined as,

$$\{p_s \in \{p_j\} \mid |p_s| = \min(|p_1|, |p_2|, \dots, |p_N|)\}$$

The $\{p_s\}$ may be calculated accurately and independently of any of the other poles using property 4 of the Padé table.

Once the $\{p_s\}$ are calculated, the Padé table is determined up to a maximum L and M (usually 10-12) and the denominator polynomial calculated for each. Efficient algorithms are available to carry this out without solving (4.7) for each entry[BGM81]. The two selection criteria may now be applied.

Criterion 1 A Padé table entry is eliminated as an approximation if the roots of its denominator do not include the $\{p_s\}$.

Padé approximations decrease in accuracy with increased distance from the origin. If an approximation does not contain the $\{p_s\}$ as its roots, it is safe to assume that it is not an accurate approximation. Several table entries may be eliminated in this way without further processing.

This criterion is checked, *without generating poles at each approximation*, by simply substituting the $\{p_s\}$ into the $\hat{Q}_{[L/M]}$ and verifying that the residual error is almost zero¹. Criterion 1, however, is not a sufficient condition for accuracy.

Expanding (2.18) in a Maclaurin series gives,

$$\hat{H}(s) = \left(\hat{c} + \sum_{j=1}^q \frac{-\hat{k}_j}{\hat{p}_j} \right) + \sum_{n=1}^{\infty} s^n \left(\sum_{j=1}^q \frac{-\hat{k}_j}{\hat{p}_j^{n+1}} \right) \quad (5.3)$$

in which the approximate moments of the response are,

$$\hat{m}_0 = \hat{c} + \sum_{i=1}^q \frac{-\hat{k}_i}{\hat{p}_i} \quad \hat{m}_j = \sum_{i=1}^q \frac{-\hat{k}_i}{\hat{p}_i^{j+1}} \quad (5.4)$$

¹If the $\{p_s\}$ are indeed the roots of the denominator then the residual error will be exactly zero.

The equations used to calculate the approximate residues and the coupling constant for a set of q approximate poles is given by (2.29)².

By implication therefore,

$$m_n = \hat{m}_n \quad (n = 0, 1, \dots, q-1) \quad (5.5)$$

A second error criterion for a set of poles and residues is provided by comparing the "skew" of the next moments unused in (2.29) from the corresponding moments of the approximate response, i.e. comparing m_n to $\hat{m}_n$ for $n \geq q$.

For a given moment m_n , and respective approximate moment $\hat{m}_n$, the n th moment skew, ϵ_n , is defined as the scaled difference between the two values,

$$\epsilon_n = \left| \frac{m_n - \hat{m}_n}{m_n} \right| \quad (5.6)$$

This skew takes into account any scaling in the moments and therefore gives the relative error of the moments. There are an infinite number of moment skews but the sequence of them converges to zero for any set of poles that passes Criterion 1 because the $\{p_s\}$ dominate both the system moments and the approximate moments. The total moment skew, ϵ , for a particular set of poles and residues is given by,

$$\epsilon = \left(\sum_{n=q}^{\infty} \epsilon_n^a \right)^{1/a} \quad (a \geq 1) \quad (5.7)$$

($\alpha = 1$ is generally chosen). A large error in approximation implies a large total moment skew.

Criterion 2 *Of the entries remaining from Criterion 1, the set of filtered (non-parasitic, left-half plane) poles and subsequent residues of each entry that minimizes the total moment skew, ϵ , is the one selected.*

A set of approximate poles is extracted from the each $\hat{Q}_{[L/M]}$ of the Padé table entries not rejected by Criterion 1. Poles known to be inaccurate may be removed or "filtered"

²Or by (2.39), in which case one more moment is matched.

out from a set without reference to the other possibly accurate poles in the set. The nature of Padé approximations is that they may produce both accurate and inaccurate poles in the same approximation. This procedure may be carried out because Criterion 2 based on the unused moments will determine if a set of poles is collectively accurate.

Inaccurate poles may be in the form of left-half plane poles or parasitic ones. Left-half plane poles are known to be inaccurate when the approximation is that of a stable network. Their removal implies that all generated approximations will be guaranteed stable. Parasitic poles (Property 7) are spurious poles generated by a Padé approximation which are relatively close to the origin and are associated with relatively small residues. These poles are caused by numerical error and do not reflect actual network poles. Any poles clearly smaller than the known $\{p_s\}$ are deemed to be spurious.

Once q poles are known for an approximation, q residues are calculated. Criterion 2 implies that approximate q poles and q residues of the selected table entry not only match the first moments, but minimize the error of higher order moments.

The cost of Padé analysis has been timed at an average of 0.2-0.6 sec. using a Sun SPARC II, depending on the range of the Padé table examined. The cost of moment computation for a large sample network (105 transmission lines, 435 lumped elements) was approximately 2.0 seconds, making the extra CPU cost a reasonable tradeoff for the increased accuracy and stability for large networks.

The method requires more moments than is generally used by a Padé approximation in order to check for the moment skew. As in a regular Padé approximation, moment-matching error minimization is done in the frequency domain, which implies increasing the accuracy about $s = 0$ for low-pass networks.

5.4 Optimal pole selection

Analysis of a network generates moments for every node in that network by necessity. This is because any moment depends recursively on the lower order moments at every node. This fact can be used to improve an approximation. Moment skew, if large, indicates that the approximation is inaccurate. Sometimes it is not possible to obtain a small moment skew out of any entry in the Padé table for a calculated set of moments. In order to further increase accuracy, approximate poles calculated using another node's moments can be applied to the output node of interest[C'N92d]. Using the moments from the desired output node and poles generated at another node, residues for the output node are calculated using (2.29). This is known as a partial Padé approximation[Ale84]. The moments at the required output nodes are still matched. A resultant set of poles and residues can be checked for accuracy using the moment skew criterion.

The same system poles appear at every node, because the poles are a unique property of the system. Only the residues of these poles are different for each node. A Padé approximation may be performed at a node in order to extract only the poles. Using one node over another may extract more accurate poles in one node over the other. The poles of several other nodes may be generated at small additional expense. Then, at the node of interest, several sets of poles may be generated. The moment skew error criterion may be used to choose the node with most appropriate poles.

One method of choosing the pole generation node of a single input network was proposed in [AGKP92]. In this method, the node is the single driving point of the network. It is known that interconnect networks act as low-pass networks that attenuate the high frequency components of incoming signals. However, at the driving point of a circuit, the high frequency components of a signal are still to be found. Performing a Padé approximation at an output node may produce unstable right half plane poles due to the existence of some high frequency poles with small residue values. These poles will be truncated in the moments and "blocked" by the low frequency poles

close to the origin. By taking the driving point however, the high frequency poles are more easily detected. In order to remove the effect of any truncated poles at the driving point, an approximation is generated by a horizontally converged sequence in the Padé table (4.15). In this way any poles that are significant to the moments are captured accurately. These poles can then be used in conjunction with the moments at the required output node to extract the residues at that node. Since the high frequency poles will have minor residues at the output node it would not have been possible to capture them accurately at that node. Using the driving point enables the accurate extraction of all poles significant to the output waveform.

5.5 Partitioning

Circuit partitioning[Roh88] is a technique which divides a circuit into subnetworks, obtains solutions for the subnetworks and combines these partial solutions to arrive at a global solution. This "divide and conquer" strategy therefore converts a large problem into a number of smaller ones which can be solved separately. Special techniques can be used on specific partitions to find partial solutions quickly. Solving partitioned circuits can lead to significant savings in both CPU time and memory utilization. More importantly, partitioning often increases the accuracy of a moment-matching simulation by allowing an increase in the number of poles generated per network. Instead of generating one set of moments for the entire network, each partition may be used to generate its own moments and consequently its separate Padé approximations. The final network response will be represented by several Padé approximations rather than one. Alternatively, the moments of a partition can be combined with moments from other partitions to generate moments for the global circuit. The moments generated by using partitioning are identical to those generated when analyzing the global circuit as a whole.

Partitioning can be applied to AWE, without any loss of generality[ABL⁺92b].

Consider the partitioned circuit shown in Fig. 5.1. Each partition (subnetwork) $P_1, P_2, \dots$ is represented as a multi-port element in the global circuit, where the connections among the partitions are represented in the *global circuit* block. The moments of general hybrid representation for each partition can be computed independently. These can then be used to determine the moments of the global circuit.

Figure 5.1. General partitioned network.

For the j 'th partition, the hybrid representation³ is given by

$$\mathbf{H}^{(j)}(s)\mathbf{u}_p^{(j)}(s) + \mathbf{e}^{(j)}(s) = \mathbf{y}_p^{(j)}(s) \quad (5.8)$$

where, $\mathbf{u}_p^{(j)}$ is a vector of voltages and currents across external voltage and current ports of the partition respectively,

$\mathbf{y}_p^{(j)}$ is a vector of currents and voltages across the same ports respectively,

$\mathbf{e}^{(j)}$ is a vector representing the effect of all voltage/current sources internal to the partition j on the output ports, and

$\mathbf{H}^{(j)}(s)$ are the Hybrid parameters which represent the effect of $\mathbf{u}_p^{(j)}$ on $\mathbf{y}_p^{(j)}$.

³Any representation, such as the Z or Y parameters could be used. The hybrid parameters are used to denote *all* of these possibilities.

Deciding which nodes become voltage/current ports, i.e. how the circuit is partitioned, is based upon a topological check at the global circuit level and on heuristics developed from the type of partitions required.

Initial conditions and internal independent source inputs are incorporated in partitioned circuits. To explain this, the state variable form in (2.1) is referred to. The inputs to a partition are voltages/currents at the voltage/current ports $\mathbf{u}_p$, and independent sources within a partition $\mathbf{u}_s$. The outputs of a partition are the appropriate complementary port variables $\mathbf{y}_p$ and the desired internal variables $\mathbf{y}_0$. Using this notation, (2.1) becomes,

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{B}_p\mathbf{u}_p + \mathbf{B}_s\mathbf{u}_s \quad (5.9)$$

$$\mathbf{y}_p = \mathbf{C}_p\mathbf{x} + \mathbf{D}_{pp}\mathbf{u}_p + \mathbf{D}_{ps}\mathbf{u}_s \quad (5.10)$$

$$\mathbf{y}_0 = \mathbf{C}_0\mathbf{x} + \mathbf{D}_{op}\mathbf{u}_p + \mathbf{D}_{os}\mathbf{u}_s \quad (5.11)$$

Combining (5.9) and (5.10) and taking the Laplace transform gives,

$$\begin{aligned} \mathbf{y}_p(s) = & (\mathbf{C}_p(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}_p + \mathbf{D}_{pp})\mathbf{u}_p(s) \\ & + (\mathbf{C}_p(s\mathbf{I} - \mathbf{A})^{-1}(\mathbf{x}_0 + \mathbf{B}_s\mathbf{u}_s(s)) + \mathbf{D}_{ps}\mathbf{u}_s(s) \end{aligned} \quad (5.12)$$

where $\mathbf{x}_0$ is the vector of initial condition values. For simplicity, it is assumed that all the internal independent sources are connected to a single unit step source. More complicated sources can be decomposed into a summation of multiple steps of different amplitudes. The assumption does not change the generality of the partitioning scheme. This can be formulated as,

$$\mathbf{B}_s\mathbf{u}_s(s) = \mathbf{f}s^{-1} \quad (5.13)$$

$$\mathbf{D}_{ps}\mathbf{u}_s(s) = \mathbf{d}s^{-1} \quad (5.14)$$

where $\mathbf{f}$ and $\mathbf{d}$ are constant vectors. Expanding $(s\mathbf{I} - \mathbf{A})^{-1}$ in a Maclaurin series and combining (5.12), (5.13), and (5.14) yields,

$$\begin{aligned} \mathbf{y}_p(s) = & (\mathbf{D}_{pp} - \mathbf{C}_p\mathbf{A}^{-1}\mathbf{B}_p - s\mathbf{C}_p\mathbf{A}^{-2}\mathbf{B}_p - \cdots)\mathbf{u}_p(s) \\ & + (-s^{-1}\mathbf{C}_p\mathbf{A}^{-1}\mathbf{f} - \mathbf{C}_p\mathbf{A}^{-1}(\mathbf{x}_0 + \mathbf{A}^{-1}\mathbf{f}) - \cdots) - s^{-1}\mathbf{d} \end{aligned} \quad (5.15)$$

Combining like terms gives an expression in the form,

$$\mathbf{y}_p(s) = (\mathbf{H}_0 + s\mathbf{H}_1 + \cdots)\mathbf{u}_p(s) + (s^{-1}\mathbf{e}_{-1} + \mathbf{e}_0 + s\mathbf{e}_1 + \cdots) \quad (5.16)$$

where the superscript i has been dropped for simplicity. Equation (5.16) represents the moment expansion for the branch relations of a given partition i . The $\mathbf{e}_i$ terms are calculated by zeroing all port sources, and exciting the internal independent sources and initial conditions. To obtain the $\mathbf{H}_i$ terms, the internal sources and initial conditions are zeroed, and the port sources are excited. The j 'th column of the $\mathbf{H}_i$ terms is found by exciting only the j 'th port source. The moments are found efficiently by using the same recursion as in (3.10) and (3.11), independently of the rest of the circuit. As in a global solution, finding these moments requires only one LU decomposition of the circuit matrix of this partition. If the partition has a special structure more efficient special case solvers can be used.

Once the moments of a partition have been determined, the internal details of the partition become unnecessary. The moments can be used to calculate the moments of the global circuit to determine a global solution, or the partition moments can be used to form a macromodel of poles and residues (Chapter 7) to be combined with the rest of the circuit in simulation. This can lead to significant memory savings, and thus larger problems may be simulated.

In the global circuit, partitions act just like multi-terminal devices and the parameters found correspond to their branch constitutive relations. The global circuit can then be formulated using these relations between input and output ports and the Kirchhoff constraints. Any general formulation method such as sparse tableau analysis or modified nodal analysis can be applied.

The global matrix to be solved is smaller than the original global matrix because the internals of the partitions are replaced by a smaller set of elements.

In order to calculate a set of moments for the global matrix, each partition contributes a multi-port stamp to the global circuit matrix. Further expanding (5.8) in

terms of a series for each constituent yields,

$$\begin{aligned} & (\mathbf{H}_0^{(j)} + s\mathbf{H}_1^{(j)} + \cdots)(s^{-1}\mathbf{u}_{p,-1}^{(j)} + \mathbf{u}_{p,0}^{(j)} + \cdots) \\ & + (s^{-1}\mathbf{e}_{-1}^{(j)} + \mathbf{e}_0^{(j)} + s\mathbf{e}_1^{(j)} + \cdots) = (s^{-1}\mathbf{y}_{p,-1}^{(j)} + \mathbf{y}_{p,0}^{(j)} + \cdots) \end{aligned} \quad (5.17)$$

Matching like coefficients of s gives,

$$\begin{aligned} \mathbf{H}_0^{(j)}\mathbf{u}_{p,-1}^{(j)} &= \mathbf{y}_{p,-1}^{(j)} - \mathbf{e}_{-1}^{(j)} \\ \mathbf{H}_0^{(j)}\mathbf{u}_{p,0}^{(j)} &= \mathbf{y}_{p,0}^{(j)} - \mathbf{e}_0^{(j)} - \mathbf{H}_1^{(j)}\mathbf{u}_{p,-1}^{(j)} \\ \mathbf{H}_0^{(j)}\mathbf{u}_{p,k}^{(j)} &= \mathbf{y}_{p,k}^{(j)} - \mathbf{e}_k^{(j)} - \sum_{i=1}^{k+1} \mathbf{H}_i^{(j)}\mathbf{u}_{p,-1}^{(j)} \end{aligned} \quad (5.18)$$

The $\mathbf{H}_i^{(j)}$'s and $\mathbf{e}_i^{(j)}$'s were calculated from elements internal to the partition. The equations therefore give a relationship between the moments of the output variable as a function of the moments of the input variable. The relationships for each partition are stamped into a global set of equations which, in a manner paralleling the partition, can be solved recursively.

5.6 Method of characteristics

If a linear partition includes only a single transmission line or a single system of coupled lines, then it is possible to apply AWE and the (generalized) method of characteristics[Cha89] to produce a macromodel for that partition[BRR92b]. A "pure delay" term can be pulled out of the approximation. The resulting macromodel encapsulates less linear circuitry than when applying AWE to the entire linear network, but the tradeoff will be greater accuracy when the lines contain long delays compared to the rise times of the signals.

Properly terminated lossless lines, for example, can be replaced by a pure delay because a signal is transmitted without any distortion or loss and is essentially received at the other end identically except for a time delay. A Padé approximation has difficulty with long delays because by implication, the signal must stay unchanged

(flat) for the period of the time delay. It is not easy to characterize this kind of flat delay with exponentials alone. Any small set of exponentials will display a non-physical ringing effect⁴.

The Laplace transform of the Telegrapher's equations are given in (3.15) and (3.16), and as one set in (3.50). These equations can be solved by diagonalizing,

$$\mathbf{D} + s\mathbf{E} = - \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{G} & \mathbf{0} \end{bmatrix} - s \begin{bmatrix} \mathbf{0} & \mathbf{L} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \quad (5.19)$$

Let the matrix $\bar{\mathbf{T}}(s)$ of left eigenvectors be expressed in the block form,

$$\mathbf{T}(s) = \begin{bmatrix} \mathbf{A}(s) & \mathbf{B}(s) \\ \mathbf{A}(s) & -\mathbf{B}(s) \end{bmatrix} \quad (5.20)$$

which satisfies the relation,

$$\begin{aligned} & \begin{bmatrix} \mathbf{A}(s) & \mathbf{B}(s) \\ \mathbf{A}(s) & -\mathbf{B}(s) \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{R} + s\mathbf{L} \\ \mathbf{G} + s\mathbf{C} & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} -\mathbf{\Gamma}(s) & \mathbf{0} \\ \mathbf{0} & \mathbf{\Gamma}(s) \end{bmatrix} \begin{bmatrix} \mathbf{A}(s) & \mathbf{B}(s) \\ \mathbf{A}(s) & -\mathbf{B}(s) \end{bmatrix} \end{aligned} \quad (5.21)$$

where the matrix $\mathbf{\Gamma}(s)$ is diagonal, consisting of eigenvalues of the block matrix $\mathbf{T}(s)$. The solution to (3.50) can then be expressed as:

$$\begin{aligned} & \begin{bmatrix} \mathbf{A}(s) & \mathbf{B}(s) \\ \mathbf{A}(s) & -\mathbf{B}(s) \end{bmatrix} \begin{bmatrix} \mathbf{V}(x, s) \\ \mathbf{I}(x, s) \end{bmatrix} \\ &= \begin{bmatrix} e^{-x\mathbf{\Gamma}(s)} & \mathbf{0} \\ \mathbf{0} & e^{x\mathbf{\Gamma}(s)} \end{bmatrix} \begin{bmatrix} \mathbf{A}(s) & \mathbf{B}(s) \\ \mathbf{A}(s) & -\mathbf{B}(s) \end{bmatrix} \begin{bmatrix} \mathbf{V}(0, s) \\ \mathbf{I}(0, s) \end{bmatrix} \end{aligned} \quad (5.22)$$

⁴The non-physical ringing can be significantly reduced by increasing the number of exponentials used. However this requires more poles than can be achieved by one expansion point and Padé approximation. Instead a general multi-point algorithm is needed (Chapter 6).

Or equivalently,

$$\begin{aligned}
 & \begin{bmatrix} \mathbf{B}(s) & \mathbf{0} \\ \mathbf{0} & \mathbf{B}(s) \end{bmatrix} \begin{bmatrix} \mathbf{Y}_0(s) & \mathbf{1} \\ \mathbf{Y}_0(s) & -\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{V}(x, s) \\ \mathbf{I}(x, s) \end{bmatrix} \\
 &= \begin{bmatrix} e^{-x\Gamma(s)} & \mathbf{0} \\ \mathbf{0} & e^{x\Gamma(s)} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{B}}(s) & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{B}}(s) \end{bmatrix} \\
 & \quad \begin{bmatrix} \mathbf{Y}_0(s) & \mathbf{U} \\ \mathbf{Y}_0(s) & -\mathbf{U} \end{bmatrix} \begin{bmatrix} \mathbf{V}(0, s) \\ \mathbf{I}(0, s) \end{bmatrix}
 \end{aligned} \tag{5.23}$$

where,

$$\mathbf{Y}_0(s) = \mathbf{B}^{-1}(s)\bar{\mathbf{A}}(s) \tag{5.24}$$

is the *characteristic admittance* matrix for the system of lines. For simplicity, using (3.5) and dropping the subscript k , the following port-variable vectors are defined,

$$\begin{aligned}
 \mathbf{V}_1(s) &= \mathbf{V}(0, s) & \mathbf{I}_1(s) &= \mathbf{I}(0, s) \\
 \mathbf{V}_2(s) &= \mathbf{V}(d, s) & \mathbf{I}_2(s) &= -\mathbf{I}(d, s)
 \end{aligned} \tag{5.25}$$

Equation (5.23) is evaluated at $x = d$, and simplified using (5.25),

$$\begin{aligned}
 \mathbf{Y}_0(s)\mathbf{V}_1(s) - \mathbf{I}_1(s) &= \bar{\mathbf{B}}^{-1}(s)e^{-d\Gamma(s)}\bar{\mathbf{B}}(s)[\mathbf{Y}_0(s)\mathbf{V}_2(s) + \mathbf{I}_2(s)] \\
 \mathbf{Y}_0(s)\mathbf{V}_2(s) - \mathbf{I}_2(s) &= \bar{\mathbf{B}}^{-1}(s)e^{-d\Gamma(s)}\bar{\mathbf{B}}(s)[\mathbf{Y}_0(s)\mathbf{V}_1(s) + \mathbf{I}_1(s)]
 \end{aligned} \tag{5.26}$$

AWE requires that the moments of the network be expanded about $s = 0$ and that the first moment about $s = \infty$ be matched. This means that the matrices $\mathbf{Y}_0(s)$, $\mathbf{B}(s)$, and $e^{-d\Gamma(s)}$ must each, in turn, be expanded about those points. A Maclaurin series expansion is achieved by substituting,

$$\begin{aligned}
 \mathbf{A}(s) &= \mathbf{A}_0 + s\dot{\mathbf{A}}_1 + s^2\ddot{\mathbf{A}}_2 + \dots \\
 \mathbf{B}(s) &= \mathbf{B}_0 + s\dot{\mathbf{B}}_1 + s^2\ddot{\mathbf{B}}_2 + \dots \\
 \Gamma(s) &= \Gamma_0 + s\dot{\Gamma}_1 + s^2\ddot{\Gamma}_2 + \dots
 \end{aligned} \tag{5.27}$$

in (5.22). Matching equal powers of s results in,

$$\begin{aligned} & \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{A}_k & -\mathbf{B}_k \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \\ & + \begin{bmatrix} \mathbf{A}_{k-1} & \mathbf{B}_{k-1} \\ \mathbf{A}_{k-1} & -\mathbf{B}_{k-1} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{L} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \\ & = \sum_{i=0}^k \begin{bmatrix} -\Gamma_i & \mathbf{0} \\ \mathbf{0} & \Gamma_i \end{bmatrix} \begin{bmatrix} \mathbf{A}_{k-i} & \mathbf{B}_{k-i} \\ \mathbf{A}_{k-i} & -\mathbf{B}_{k-i} \end{bmatrix} \end{aligned} \quad (5.28)$$

This equation can be solved starting at $k = 0$ and working up to higher orders. The $k = 0$ solution corresponds to a frequency independent eigenproblem. Higher order solutions ($k > 0$) can be obtained by appending a normalization criterion in which it is asserted that the rows of $\mathbf{T}(s)$ are vectors of unit length. This gives a set of linear equations and result in the moments $\mathbf{A}_k$ and $\mathbf{B}_k$. Once these have been generated the moments of the characteristic admittance matrix $\mathbf{Y}_0(s)$ can be computed using,

$$\mathbf{B}(s)\mathbf{Y}_0(s) = \mathbf{A}(s) \quad (5.29)$$

Expanding each term in a Taylor series and matching corresponding powers of s gives,

$$\sum_{i=0}^k \mathbf{B}_i \mathbf{Y}_{0k-i} = \mathbf{A}_k \quad (5.30)$$

which can be solved as well starting with $k = 0$.

There are two options for expanding $e^{-d\Gamma(s)}$. Either expand it directly in a Maclaurin series, or pull out a "delay" term first, before performing an expansion. A pure delay is given in form of e^{-as} in the frequency domain, i.e.,

$$\mathcal{L}^{-1}(e^{-as}F(s)) = f(t-a) \quad (5.31)$$

where $\mathcal{L}(f(t)) = F(s)$.

A pure delay can be pulled out of $e^{-d\Gamma(s)}$ by expanding about $s = \infty$. The only stipulation is that the series for $\Gamma(s)$ begin with an s term, so that its coefficient will

be the delay term required. We then have,

$$\begin{aligned}\Gamma(s) &= s\Gamma_{-0} + \Gamma_{-1} + s^{-1}\Gamma_{-2} + \dots \\ \mathbf{A}(s) &= \mathbf{A}_{-0} + s^{-1}\bar{\mathbf{A}}_{-1} + s^{-2}\bar{\mathbf{A}}_{-2} + \dots \\ \mathbf{B}(s) &= \mathbf{B}_{-0} + s^{-1}\bar{\mathbf{B}}_{-1} + s^{-2}\bar{\mathbf{B}}_{-2} + \dots\end{aligned}\quad (5.32)$$

Inserting these series into (5.21) and matching corresponding powers of s leads to the first term,

$$\begin{aligned}& \begin{bmatrix} \mathbf{A}_{-0} & \bar{\mathbf{B}}_{-0} \\ \bar{\mathbf{A}}_{-0} & -\bar{\mathbf{B}}_{-0} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{L} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} -\Gamma_{-0} & \mathbf{0} \\ \mathbf{0} & \Gamma_{-0} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{A}}_{-0} & \bar{\mathbf{B}}_{-0} \\ \bar{\mathbf{A}}_{-0} & -\bar{\mathbf{B}}_{-0} \end{bmatrix}\end{aligned}\quad (5.33)$$

Equation (5.33) is solved as an eigenproblem for $\bar{\mathbf{A}}_{-0}, \bar{\mathbf{B}}_{-0}$, and Γ_{-0} . The entries of the diagonal matrix Γ_{-0} contain the per-unit-length delays for the line, while $\bar{\mathbf{A}}_{-0}$ and $\bar{\mathbf{B}}_{-0}$ are the direct coupling terms for the approximation of $\mathbf{Y}_0(s)$.

When the expansion about $s = \infty$ is taken to the next recursion, the result is,

$$\begin{aligned}& \begin{bmatrix} \mathbf{A}_{-1} & \bar{\mathbf{B}}_{-1} \\ \bar{\mathbf{A}}_{-1} & -\bar{\mathbf{B}}_{-1} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{L} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{A}}_{-0} & \bar{\mathbf{B}}_{-0} \\ \bar{\mathbf{A}}_{-0} & -\bar{\mathbf{B}}_{-0} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} -\Gamma_{-1} & \mathbf{0} \\ \mathbf{0} & \Gamma_{-1} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{A}}_{-0} & \bar{\mathbf{B}}_{-0} \\ \bar{\mathbf{A}}_{-0} & -\bar{\mathbf{B}}_{-0} \end{bmatrix} \\ &+ \begin{bmatrix} -\Gamma_{-0} & \mathbf{0} \\ \mathbf{0} & \Gamma_{-0} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{A}}_{-1} & \bar{\mathbf{B}}_{-1} \\ \bar{\mathbf{A}}_{-1} & -\bar{\mathbf{B}}_{-1} \end{bmatrix}\end{aligned}\quad (5.34)$$

When this equation is solved, the resulting diagonal matrix Γ_{-1} provides the direct coupling information for the approximation of $\Gamma(s)$.

The moments of the exponential with the delay removed, $e^{sd\Gamma_{-0}-d\Gamma(s)}$, are computed by calculating the moments of,

$$e^{-d[\Gamma_0 + s(\Gamma_1 - \Gamma_{-0}) + s^2\Gamma_2 + \dots]}\quad (5.35)$$

The first few terms are given by.

$$e^{-d\Gamma_0} \left[\mathbf{U} - sd(\Gamma_1 - \Gamma_{-0}) + \frac{1}{2}s^2 (d^2(\Gamma_1 - \Gamma_{-0})^2 - d\Gamma_2) - \dots \right] \quad (5.36)$$

Simulation of the pure delay term is done via a delay buffer which is kept in memory for each delay calculated, as in [Bra67]. The rest of the model can be transformed into a lumped element model that corresponds to the calculated poles and residues.

5.7 Frequency shifting

Stiff systems can cause problems for a Padé approximation. The stiffness is due to a few poles relatively close to the origin while other poles are relatively farther away. A Padé approximation will only detect the few poles (sometimes only one) that are closest to the origin (Property 2). In order to address this problem a frequency shift is performed. In this process, a Taylor series expansion about a positive point α_0 away from the origin replaces the original Maclaurin expansion. The object of the shift is to move away from the closest poles blocking the view of the other poles farther away. The shift is chosen judiciously so as not to be too far into the right-half plane and therefore too far from the poles to detect them accurately. Generally, a fraction of the highest frequency f_{max} of interest is chosen as the shifting point.

Frequency shifting is equivalent to adding resistors of value $\alpha_0 L$ in series with inductors of value L and resistors of value $1/\alpha_0 C'$ in parallel with capacitors of value C' . The same applies to the per-unit-length parameters of transmission lines.

When expanding about another point, it is sufficient to shift the system equations by that point and compute an expansion about the new shifted origin. Any poles obtained from the new moments can then be shifted back to the old origin. So for $s' = s - \alpha_0$ (2.5) becomes,

$$\mathbf{Y}(s') = \mathbf{c}^T (s' \mathbf{1} + \alpha_0 \mathbf{1} - \mathbf{A})^{-1} \mathbf{b} + \mathbf{D} \quad (5.37)$$

giving positive and negative moments about $s' = 0$ in the form,

$$\mathbf{M}'_0 = \mathbf{D} - \mathbf{c}^T(\mathbf{A} - \alpha_0 \mathbf{1})^{-1} \mathbf{b} \quad (5.38)$$

$$\mathbf{M}'_i = -\mathbf{c}^T(\mathbf{A} - \alpha_0 \mathbf{1})^{-(i+1)} \mathbf{b} \quad (i > 0) \quad (5.39)$$

If the network is given in the MNA formulation, then substituting $s' + \alpha$ for s gives,

$$(\mathbf{G}_\phi + (s' + \alpha)\mathbf{C}_\phi)\mathbf{V}_\phi(s) = \mathbf{E}_\phi \quad (5.40)$$

which in turn gives moments $\mathbf{M}'_i$ about $s = \alpha$ in the form,

$$(\mathbf{G}_\phi + \alpha\mathbf{C}_\phi)\mathbf{V}_{\phi[0]} = \mathbf{E}_\phi \quad (5.41)$$

and,

$$(\mathbf{G}_\phi + \alpha\mathbf{C}_\phi)\mathbf{V}_{\phi[i]} = -\mathbf{C}_\phi\mathbf{V}_{\phi[i-1]} \quad (i > 0) \quad (5.42)$$

In the case of transmission lines, expansion about α is done by shifting the per-unit-length matrices. The characteristic per-unit-length line impedance matrix, $\mathbf{Z}_p = \mathbf{R} + s\mathbf{L}$ would be replaced by $\mathbf{Z}'_p = (\mathbf{R} + \alpha\mathbf{L}) + s'\mathbf{L}$ and $\mathbf{Y}_p = \mathbf{G} + s'\mathbf{C}$ by $\mathbf{Y}'_p = (\mathbf{G} + \alpha\mathbf{C}) + s\mathbf{C}$. With the exponential method, we therefore set $\mathbf{D}' = \mathbf{D} + \alpha\mathbf{E}$ and $\mathbf{E}' = \mathbf{E}$ in (3.66) and (3.67) and apply the recursion algorithm.

Another method, similar to shifting, is conformal mapping (transformation) of the s variable to another variable in a similar complex plane. The point of the mapping is to bring the poles of the mapped function closer to the origin in order to improve, at times, a Padé approximation. A mapping may improve the detection of some poles and make worse the detection of others. I refer the reader to [ALR92] for more details.

In the general case, it is possible to "hop" to any point in the complex s -plane and expand in a Taylor series there. However, expansions outside of the real axis or $s = \infty$ result in complex moments and the calculations must be carried out with complex numbers which can be CPU intensive. Further, it is not immediately clear how to choose other expansion points nor how to combine information from two or more expansion points into a unified whole rather than replacing one expansion

point for another. The next chapter proposes how to solve these problems to include information from multiple expansion points in a data-set.

Chapter 6

Complex Frequency Hopping

One of the major problems with the moment-matching schemes described so far is that they offer no means of pre-determining the accuracy range of the approximation. Padé approximations are accurate in the neighborhood of an approximation. If that neighborhood contains most of the signal energy (most of the large residues and poles close to the origin) then the time domain approximation will be accurate. If some of the signal energy exists at frequencies not attained by the approximation then inaccuracies will appear in the transient simulation. There will be no reliable way to determine if this has happened unless carrying out a full accurate simulation and comparing.

In the frequency domain, any approximation will be accurate near points used to generate the moments. If moments are generated and matched at $s = 0$, for example, the frequency response will be accurate at DC and loss of accuracy can occur at high frequencies. If moments from both $s = 0$ and $s = \infty$ are matched, then the approximation will be accurate in those regions and may be inaccurate in unknown regions in the frequency plane between those two points. These problems are highlighted when one inspects the approximate frequency response generated by a Padé approximation. The approximate response, in general, will be true to the actual frequency response only at certain frequency ranges and it is not possible to

determine *a-priori* what the ranges of accuracy will be. The accuracy ranges will be different for different circuits. If the circuit is operating in frequencies that are approximated accurately then no error will result in the transient analysis. However, in general, this result cannot be guaranteed for all frequencies and any network.

For example, the analysis of linear networks may require hundreds of closely-spaced frequency point analyses in order to determine the resonant frequencies accurately[CHNR93]. These frequencies show up as sharp, narrow bands in the frequency response, making them difficult to detect with a widely spaced sweep. A single Padé approximation cannot guarantee that all resonant frequencies within a given band will be detected, and will consequently not be able to determine all the resonant frequencies accurately.

Partitioning, especially when coupled with the method of characteristics, can significantly improve an approximation, but the tradeoff is that each set of coupled transmission lines must be analyzed separately and delay buffers generated and simulated for each. This may give rise to a large cost for large systems containing hundreds of transmission lines. Even with the improved approximation there is no guarantee that an entire frequency range is approximated accurately.

I address these problems using a multi-point moment-matching algorithm which I entitle complex frequency hopping (CFH). Multiple expansion points can cover an entire desired range of the frequency domain which may then be approximated accurately, making the time domain response also more accurate. The complete unpartitioned network can be approximated once without the need for partitions, preserving the efficiency of global moment-matching. Delays and sharp ringing is approximated by increasing the order of the approximation and therefore the number of exponentials. CFH has been applied to circuits to successfully extract 40 50 or more poles[CN93],[CN92a], [CHNR93].

The method is so called because one thinks of "hopping" to several expansion points in the complex s plane in order to better detect poles in a different region of

the plane. CFH relies directly on the properties of Padé approximations. An efficient search strategy is incorporated for the selection and minimization of expansion points. The method generates the exact dominant poles of a linear circuit to within a specified error tolerance rather than approximate poles. This property results from an accuracy check mechanism that is not possible with a single approximation.

6.1 Complex expansions

Choosing any arbitrary complex frequency point $p = \alpha + j\beta$, one can perform a Taylor series approximation at the point in the same manner as a real frequency shift by the substitution $s' = s - p$. If the system consists of lumped element components given in the MNA formulation, analogous to (5.40) and (5.41), the shifted system moments can be derived by,

$$(\mathbf{G}_\phi + (\alpha + j\beta)\mathbf{C}_\phi)\mathbf{V}_{\phi[0]} = \mathbf{E}_\phi \quad (6.1)$$

and,

$$(\mathbf{G}_\phi + (\alpha + j\beta)\mathbf{C}_\phi)\mathbf{V}_{\phi[i]} = -\mathbf{C}_\phi\mathbf{V}_{\phi[i-1]} \quad (i > 0) \quad (6.2)$$

These moments, in contrast to moments about $s = 0$, $s = \alpha$ and $s = \infty$, will be in complex numbers. Care must be taken to use the proper programming techniques because the calculations generating these moments will require complex number operations including matrix solving and root finding. It is recommended that pre-written mathematical routines generally available in commercial packages be used because they will perform the routines specific to complex numbers.

In the case of lossy, coupled transmission lines, the generation of moments could be done by using eigenvalue moment methods. However, truncation error in these methods make them difficult to use in Padé table convergence of (4.15) which is needed in CFH. Matrix exponential is the preferred method for generating the moments in CFH. We therefore set $\mathbf{D}' = \mathbf{D} + (\alpha + j\beta)\mathbf{E}$ and $\mathbf{E}' = \mathbf{E}$ in (3.66) and apply the recursion algorithm to generate moments for the expansion about $\alpha + j\beta$.

At each new expansion point (hop), a Padé approximation will be performed on the moments to determine the relevant poles. A complex frequency hop is used to detect poles in a region of the complex plane not visible from another expansion point. If one expansion point is near enough to another, many or some of the same poles will be detected thereby giving confirmation of the accuracy of pole/residue pairs. This is an error check not available by moment-matching methods using one Padé approximation, which may contain both accurate and inaccurate poles.

Once the complex poles are calculated about an expansion point used as the new origin, these poles can then be shifted back to the true origin. So, for instance, if a pole exists at $-1 + j$ and an expansion is performed about j , the pole will appear in the position of -1 on the new shifted axis.

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

6.2 Pole convergence

The strategy of CFH is to target and generate the actual dominant poles of the circuit rather than approximate poles. When trying to generate an approximating *function* to the original transfer function as in (2.18), a diagonal or near diagonal Padé approximation is generally used. However, the approximate *poles* generated from such an approximation will not necessarily reflect the actual system poles. To generate the real system poles accurately, one needs to use (4.15) which allows convergence to the

actual system poles. M is chosen large enough (usually 8-9) so that the sequence converges quickly and the number of moments needed is not excessive.

Using this method, the poles generated about any hop are accurate with respect to the actual poles in the neighborhood of the expansion. The closer the point of expansion to a pole the more accurate the extracted pole. This is significant considering that the poles of distributed systems are transcendental and infinite in number and there is no general algorithm for generating them. Nevertheless, in practice, numerical truncation error affects the convergence of (4.15). As a result some poles far from the point of expansion do not converge well and a further series of steps are needed to ensure the accuracy of the poles. I describe this first process in Section 6.3. As well, a search strategy necessary to select the location of the hops and minimize their number is covered in Section 6.4.

6.3 Pole selection

With two or more complex frequency hops, pole information from each is combined into an accurate set of poles and residues once convergence of each hop is achieved. The algorithm is based on Properties 1, 2 and 3 of Padé approximations. The accuracy of an approximation is good near the point of expansion but decreases with increasing distance from the point.

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

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

Step 1 Poles from each hop are converged using (4.15).

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

$$\begin{bmatrix} \hat{p}_1^{-L-2} & \hat{p}_2^{-L-2} & \dots & \hat{p}_q^{-L-2} \\ \hat{p}_1^{-L-3} & \hat{p}_2^{-L-3} & \dots & \hat{p}_q^{-L-3} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{p}_1^{-L-M-1} & \hat{p}_2^{-L-M-1} & \dots & \hat{p}_q^{-L-M-1} \end{bmatrix} \begin{bmatrix} \hat{k}_1 \\ \hat{k}_2 \\ \vdots \\ \hat{k}_q \end{bmatrix} = - \begin{bmatrix} m_{L+1} \\ m_{L+2} \\ \vdots \\ m_{L+M} \end{bmatrix} \quad (6.3)$$

Equation (6.3) differs from the one used in (2.29) and (2.39). It derives the actual residues of the (converged) poles rather than approximate residues.

Step 3 If the same poles (with the error tolerance ε_{err}) are detected in two different expansions, they are marked as accurate.

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

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

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

Assuming that the moments about two expansion points a and b have been generated, and that (4.15) and (6.3) has been used to collect a set of q_a poles $\{p_i\}^{(a)}$ from hop a and q_b poles $\{p_j\}^{(b)}$ from hop b , as well as their respective residues $\{k_i\}^{(a)}$ and $\{k_j\}^{(b)}$ the algorithm for collecting poles/residues at hop a is given in Section 6.3.1. The pole selection algorithm is illustrated in Fig. 6.1.

Figure 6.1: Pole selection algorithm using two hops.

6.3.1 Pole selection algorithm

Un-Mark all $p_i^{(a)}$ and $k_i^{(a)}$;

if ($\exists i, j$ such that $|p_i^{(a)} - p_j^{(b)}| < \varepsilon_{err}$)

$R_{acc}^{(a)} \leftarrow \max(|p_i^{(a)}|)$ for which $|p_i^{(a)} - p_j^{(b)}| < \varepsilon_{err}$ for some j ;

end if for ($i \leq q_a$)

if ($|p_i^{(a)}| \leq R_{acc}^{(a)}$)

$p_i^{(a)} \leftarrow$ **accurate**

$k_i^{(a)} \leftarrow$ **accurate**

end if

end for

6.4 Search strategy

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

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

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

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

Constraint 3 *The search should be confined to points below the highest frequency f_{max} .* The expected operating frequency of the circuit, which depends on the input waveform and the type of circuit, limits the search up the frequency axis. Only information up to a limited operating frequency is needed. Poles higher up the imaginary axis will not be important. The f_{max} of a circuit will depend on the input ramp in the case of linear circuits [GNP91], or on the type of circuit in the case of nonlinear circuits.

The above three constraints describe a CFH search area and drastically reduce the search space as illustrated in Fig. 6.2. Using the above search area an algorithm is developed which uses a binary search strategy.

Figure 6.2: Binary search and CFH search area in the complex s plane.

Step 1 Expand at the origin ($s = 0$). Collect poles by convergence of (4.15).

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

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

Step N Continue the search in similar binary fashion between $s = 0$ and $s = j\omega_{max}/2$ and between $s = j\omega_{max}/2$ and $s = j\omega_{max}$ defining and expanding at midpoints if no common poles are found.

The algorithm is summarized in Section 6.4.1.

6.4.1 Binary search algorithm

```

OrigHop  $\leftarrow 0$ 
Expand at OrigHop;
Converge to poles  $\{p\}^{(OrigHop)}$  in the neighborhood of OrigHop;
 $\omega_{max} \leftarrow$  highest frequency of interest;
EndHop  $\leftarrow j\omega_{max}$ 
PreviousHop(EndHop)  $\leftarrow$  OrigHop;
Expand at EndHop;
Converge to poles  $\{p\}^{(EndHop)}$ ;
CommonPoleWithPreviousHop(EndHop)  $\leftarrow$  NO
SearchFinished  $\leftarrow$  NO
while (Not SearchFinished)
    SearchFinished  $\leftarrow$  YES
    SearchHop  $\leftarrow$  EndHop
    while (SearchHop  $\neq$  OrigHop)
        if (CommonPoleWithPreviousHop(SearchHop) = YES)
            SearchHop  $\leftarrow$  PreviousHop(SearchHop)
        else if (  $\exists i, j$  such that  $p_i \in \{p\}^{(SearchHop)}$   $p_j \in \{p\}^{(PreviousHop(SearchHop))}$ 
            such that  $|p_i - p_j| < \varepsilon_{err}$  )
            CommonPoleWithPreviousHop(SearchHop)  $\leftarrow$  YES ;
            SearchHop  $\leftarrow$  PreviousHop(SearchHop)
        else
            MidHop  $\leftarrow$  Midpoint(SearchHop, PreviousHop(SearchHop))
            PreviousHop(MidHop) = PreviousHop(SearchHop)
            PreviousHop(SearchHop) = MidHop
            Expand at MidHop;
            Converge to poles  $\{p\}^{(MidHop)}$ ;
            SearchFinished  $\leftarrow$  NO
            SearchHop  $\leftarrow$  EndHop
        end if
    end while
end while

```

Each hop must be near enough to the neighboring hop in order to confirm the same outer poles between the two expansions. Generally, in most low frequency circuits, the search bottoms out after two expansions. In some difficult examples of electrically long transmission lines with associated long delays, crosstalk and signal reflections, the search will need more hops. The CPU cost will be proportional to the ringing (number of dominant poles) in the circuit but the CFH method guarantees an accurate waveform within the specified frequency range. The moments generated for any circuit must be accurate for this type of search. Any small truncation error in the moments will mean that poles far from a point of expansion will not be accurate, necessitating more hops and increasing the CPU cost.

6.5 Extra CPU requirements

Using the CFH method requires calculations using complex numbers in contrast to expansions about the real axis which requires only real number processing. Consequently, the cost of an arbitrarily complex frequency expansion may, at first glance, seem more expensive than an expansion in real numbers about a point on the real axis. However, a dominant cost of a real expansion is the LU factorization of the MNA matrix. This is approximately equivalent to the CPU cost of one frequency point analysis. Sparse matrix analysis, however, requires the cost of matrix reordering before performing the LU factorization. This reordering need not be done again for any additional frequency points[Kun86]. The CPU cost at the origin is partly composed of a reordering and a (real) LU factorization. Every other expansion point consists of the following CPU cost: 1. Complex LU factorization (without reordering), 2. Padé table convergence, and 3. Complex moment generation (repeated forward-backward substitution). In general a sparse matrix reordering consumes more CPU time than one LU factorization. For example, CFH was implemented using Berkeley sparse matrix routines[Kun87]. A real LU factorization of a 240 node circuit about

the origin took 0.22 CPU seconds on a SUN SPARC II, while a subsequent complex factorization of the same circuit took 0.02 seconds. The ratio was 13.83 seconds to 0.77 seconds for a 1143 node circuit. It is to be noted that the matrix to be factored at a shifted point $\mathbf{G}_\phi + p\mathbf{C}_\phi$ is generally more dense and therefore somewhat more expensive than the matrix necessary to expand at the origin $\mathbf{G}_\phi$. However, even if only a single expansion about the origin is needed, it may be necessary in some stiff systems to perform a shifted real LU factorization in addition to an expansion about the origin[Hua90]. Assuming that possibility therefore might make it necessary to factorize a shifted matrix at the start.

As an example, a large network containing 306 resistors, 2100 capacitors, 172 inductors and 6990 mutual inductances was analyzed to 1 GHz (Example 4). The original hop about the origin consumed 96.2 CPU seconds which showed it to be a stiff system. Another real hop was performed which took 17.7 seconds. A final highest frequency complex hop took 26.3 seconds.

The additional CPU expense of Padé table convergence and moment generation is more expensive using complex numbers as compared to real numbers. The former is independent of circuit size and diminishes with the size of the circuit, while the latter is generally 3-4 times more expensive than real moment generation. However, since moment generation and Padé analysis use a small fraction of the cost of LU factorization in a large circuit [PR90],[TN92], the increase is minimal. Table 6.1 gives a sample of some CPU times of CFH applied to large linear interconnect networks.

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

One important advantage to CFH is that it can be readily given to parallel pro-

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

Table 6.1: CPU cost (SUN SPARC II) of CFH in some large transmission line networks.

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

Example 1

In Fig. 6.3 a small test network proposed by Chang[Cha91] is displayed. Two transmission lines (T_1, T_2) with different resistive parameters are cascaded. The parameters of T_1 are $Rd = 25\Omega$, $Gd = 0.005\Omega^{-1}$, $Ld = 0.1\mu H$, $Cd = 40pF$, where d is the length of the transmission line. The parameters of T_2 are the same except for

Figure 6.3: Example circuit proposed by Chang.

$R_d = 3.125\Omega$. A 1V step of 0.5 ns rise/fall time is placed at the input, and the output waveform was generated at the load resistance. CFH was applied to the network for analysis up to $\omega_{max} = 1e10$ or $f_{max} = 1.59GHz$. The analysis generated 28 poles using 7 hops. The pole distribution of the circuit is shown in Fig. 6.4 by the 'x'. The 'o' represent the approximate poles of the best moment-matching Padé approximation about $s = 0$. Clearly, one Padé approximation fails to obtain many of the high frequency poles very close to the imaginary axis.

Figure 6.4: Pole distribution (x) of Fig. 6.3. Approximate poles obtained at the origin indicated by (o).

Figure 6.5: Transient response of Fig. 6.3.

Figure 6.6: Frequency response of Fig. 6.3.

In Fig. 6.5 I compare the accurate FFT generated transient response using hundreds of frequency point analyses to the approximate transient response using one expansion point, as well as to the CFH response using multiple expansion points. The frequency response of CFH is compared to the actual frequency response in Fig. 6.6. The CFH frequency response matches the actual response up to f_{max} after which it drops off.

Figure 6.7: A lossy coupled transmission line network

Example 2

A small lossy coupled transmission line network is shown in Fig. 6.7. The network contains 2 coupled lines and 4 coupled lines and was analyzed at two outputs: V_1 and V_2 . An input pulse of 0.1 ns rise/fall time and 0.3 ns duration was placed at the input. First, one expansion at the origin was used and then several additional incremental hops were generated. The pole distribution of the network is displayed in Fig. 6.8 by 'x'. These poles were obtained by multiple hops. The approximate poles generated at the origin are marked by 'o'.

Figure 6.8: Poles of Fig. 6.7.

The waveforms at the nodes are compared with the accurate FFT generated one. At V_1 one expansion point at the origin produced the waveform displayed in Fig. 6.9. Another hop allowed the generation of 10 accurate poles and gave the waveform in Fig. 6.10. Two more hops gave a total of 22 poles and the almost converged waveform in Fig. 6.11.

The waveform at V_2 was more difficult to obtain because it is 3 orders of magnitude smaller than the input voltage and contains many high frequency components. One expansion point at the origin produced the waveform displayed in Fig. 6.12. Four hops produced 22 poles and gave the waveform in Fig. 6.13. Finally, the waveform is almost converged with six hops giving a total of 30 poles and the waveform in Fig. 6.14.

Figure 6.9: Transient response of Fig. 6.7 at V_1 (7 poles).

Figure 6.10: Transient response of Fig. 6.7 at V_1 (10 poles).

Figure 6.11: Transient response of Fig. 6.7 at V_1 (22 poles).

Figure 6.12: Transient response of Fig. 6.7 at V_2 (7 poles).

Figure 6.13: Transient response of Fig. 6.7 at V_2 (22 poles).

Figure 6.14: Transient response of Fig. 6.7 at V_2 (30 poles).

Example 3

A 160 node circuit with 96 capacitors, 84 inductors and 116 resistors as well as 8 lossy interconnect transmission lines, was simulated using CFH. The transmission lines were mismatched with respect to the terminations, giving rise to strong reflections. The voltage was sampled close to the source, immediately after the first transmission line. The input pulse was of 1V, 10 ps rise/fall time, and 30 ps duration. The highest frequency searched was up to 159 GHz. 12 hops were required with a simulation time of 7.72 seconds and a resulting 50 poles/residues. The difficult signal is compared with the FFT generated one in Fig. 6.15. The delay is clear at the beginning of the waveform after which the pulse follows. Past 70 ps however, the reflection from the rest of the circuit clearly affects the signal. Frequency response of CFH is compared to the actual frequency response in Fig. 6.16.

Example 4

CFH was applied to the lumped element network generated by a 3-D EM analyzer based on the partial element equivalent circuit (PEEC) model[Rue74] (see Chapter 9). It contained 2100 capacitances, 172 inductances, and 6990 mutual inductances. The output current of one of the internal inductors was simulated using AWE in [HRBR92] and obtained a maximum accuracy of less than 1 GHz. The AWE analysis required one expansion about the origin and one shift frequency away from the origin because the network is extremely stiff with several poles near the origin. The same inductor current was generated using CFH for 1000 frequency point analyses from DC to 4.8 GHz, which required 2 hours of CPU time.

Figure 6.15: Transient response of CFH (Example 3).

Figure 6.16: Frequency response of CFH (Example 3).

No.	Poles	Residues
1	-1.6761e3	-1.3572e-5
2	-1.5679e3	1.0628e-6
3	-7.2630e8	1.1304e6
4	-1.8323e5 $\pm$ 3.1735e9i	6.2670e1 $\pm$ 1.3766e1i
5	-2.5253e9 $\pm$ 7.6292e9i	-7.6084e5 $\pm$ 1.6670e7i
6	-3.1884e4 $\pm$ 7.8601e9i	4.2143e2 $\mp$ 2.0498e2i
7	-1.1037e6 $\pm$ 1.0577e10i	1.0766e3 $\mp$ 9.7822e3i
8	-3.3606e6 $\pm$ 1.1772e10i	-8.1455e3 $\mp$ 1.2229e4i
9	-9.4935e6 $\pm$ 1.3018e10i	-3.1854e4 $\pm$ 6.6067e3i
10	-5.1606e8 $\pm$ 1.6965e10i	-1.7086e6 $\pm$ 4.1978e5i
11	-6.1231e6 $\pm$ 1.7370e10i	3.6323e3 $\mp$ 2.7960e4i
12	-5.5969e7 $\pm$ 1.9140e10i	-2.5456e5 $\pm$ 1.0582e5i
13	-1.7241e9 $\pm$ 2.1928e10i	1.3712e6 $\mp$ 1.6280e7i
14	-8.7856e4 $\pm$ 2.2216e10i	-1.9031e4 $\pm$ 2.6503e3i
15	-1.0990e8 $\pm$ 2.5630e10i	5.4275e5 $\mp$ 1.1512e5i
16	-6.5555e7 $\pm$ 2.8092e10i	2.5208e5 $\mp$ 3.3975e5i
17	-4.0145e7 $\pm$ 3.1362e10i	2.3075e5 $\mp$ 8.2860e4i
18	-3.7002e7 $\pm$ 3.2879e10i	-2.5803e6 $\mp$ 4.7181e5i
19	-5.8861e8 $\pm$ 3.3715e10i	3.5055e7 $\mp$ 3.0539e7i
20	-9.0027e8 $\pm$ 3.3784e10i	-3.1768e7 $\pm$ 4.3046e7i

Table 6.2: The 37 poles and residues of Example 5 generated by a CFH search to a maximum frequency of 4.8 GHz.

CFH was applied to the circuit, first with a maximum pre-defined frequency of 1 GHz, (see Fig. 6.17). This analysis required three frequency hops and generated 9 poles. It ran over 300 times faster than SPICE on the same workstation. Though slower than an AWE analysis, it guaranteed the accuracy range to the pre-specified f_{max} . CFH was then applied to the same current output but with a maximum pre-defined frequency of 4.8 GHz. The algorithm generated 37 poles using 12 frequency hops, and was still about 85 times faster than the SPICE analysis. The 37 poles generated by CFH are shown in Table 6.2 and the frequency response is given in Fig. 6.18. It is to be noted that the complete frequency response given in Fig. 6.18 can be reconstructed by using (2.18) and the poles/residues given in Table 6.2. This is significant considering that the circuit has hundreds of poles.

Figure 6.17: PEEC Frequency domain CFH analysis using $f_{max}=1$ GHz.

Figure 6.18: PEEC Frequency domain CFH analysis using $f_{max}=4.8$ GHz.

Example 5

A small coupled lossless transmission line network was proposed by Cooke, Prince and Cangellaris in [CPC92] and illustrated in Fig. 6.19. The values for R , C and L are $27\ \Omega$, $1\ \text{pF}$ and $10\ \text{nH}$ respectively. There are two three-line lossless coupled transmission lines that are each 2.54cm in length. They have the following C and L matrices:

$$\mathbf{C} = \begin{bmatrix} 0.667 & -0.163 & -0.0145 \\ -0.163 & 0.722 & -0.163 \\ -0.0145 & -0.163 & 0.667 \end{bmatrix} \frac{\text{pF}}{\text{cm}}$$

$$\mathbf{L} = \begin{bmatrix} 5.033 & 1.734 & 0.818 \\ 1.734 & 4.972 & 1.734 \\ 0.818 & 1.734 & 5.033 \end{bmatrix} \frac{\text{nH}}{\text{cm}}$$

The center line is active and the two outer lines represent a shield line and a coupled load line. The active line is driven by a clocked source with a 10-ns period, a 50% duty cycle, and a 0.1-ns fall time. The simulation was first performed using a CFH search up to 1GHz, which required three hops and generated 9 poles. The resultant waveforms at selected nodes of the network are displayed in Fig. 6.20. A second simulation was performed using a search to 5 GHz. It needed twice the number of hops (6) and generated 21 poles. The accuracy is only slightly enhanced (Fig. 6.21) but compares very favorably with the waveforms given in [CPC92].

Figure 6.19: Cooke/Prince/Cangellaris coupled line example.

Figure 6.20: Transient simulation of Fig. 6.19 with $f_{max}=1$ GHz.

Figure 6.21: Transient simulation of Fig. 6.19 with $f_{max}=5$ GHz.

Chapter 7

Nonlinear Analysis

When modeling general interconnect networks, often, the drivers and receivers of the interconnects cannot be accurately modelled as linear elements. Driver and receiver modeling usually requires nonlinear equations due to the switching properties of the transistors that compose these elements. While the method of modeling terminations is not a direct subject matter discussed in this book, practical models can be incorporated into a moment-matching approximation[XN91],[XN93]. Terminations simulated can include nonlinear elements such as diodes and nonlinear resistors, capacitors and inductors. The method presented here is not intended to replace a full nonlinear simulator such as SPICE but is rather meant to handle large circuits with a few nonlinear terminations, i.e. sparsely nonlinear networks.

Nonlinear analysis of interconnect networks begins with a separation of the nonlinear elements from the linear elements into partitions (see example partition in Fig. 7.1). The linear partitions can be few or many, as can be the nonlinear partitions. Once a linear partition is formed, moment-matching may be applied to the partition to form a linear macromodel. A macromodel is also a linear circuit, but with generally less circuitry than the original approximated network of the partition. The poles of the macromodel are made to conform to the poles of the moment-matching approximation. The more accurately the poles of the approximation conform to the

Figure 7.1: Nonlinear/linear partition.

original network, the more the macromodel will act like the original partition.

7.1 Notation

Consider a nonlinear network π which contains linear lumped components, nonlinear components and N_s arbitrary linear partitions (subnetworks) ϕ_i . The linear lumped components may be described by equations in either the time or frequency domain while the nonlinear components may only be described by time domain equations. The arbitrary linear partitions may contain distributed elements that are best described in the frequency domain. The network π is described by N_π nodal variables not internal to the subnetworks, with linear subnetwork ϕ_i externally described by N_{ϕ_i} port currents and the same number of port voltages. The port voltages are contained in the N_π nodal variables of the network π .

Without loss of generality, the modified nodal admittance matrix equation of the network π can be described as,

$$\mathbf{C}_\pi \frac{d}{dt} \mathbf{v}_\pi(t) + \mathbf{G}_\pi \mathbf{v}_\pi(t) + \sum_{i=1}^{N_\pi} \mathbf{D}_{\phi_i} \mathbf{i}_{\phi_i}(t) + \mathbf{F}(\mathbf{v}_\pi(t)) - \mathbf{e}_\pi(t) = 0 \quad (7.1)$$

where,

$\mathbf{v}_\pi(t) \in \mathbb{R}^{N_\pi}$ is the vector of the N_π variables which include node voltage waveforms appended by independent voltage source current, linear inductor current, nonlinear capacitor charge and nonlinear inductor flux waveforms,

$\mathbf{e}_\pi(t) \in \mathbb{R}^{N_\pi}$ is the vector of source waveforms external to the subnetworks,

$\mathbf{C}_\pi \in \mathbb{R}^{N_\pi \times N_\pi}$, $\mathbf{G}_\pi \in \mathbb{R}^{N_\pi \times N_\pi}$ are constant matrices describing the lumped elements of network π external to the subnetworks,

$\mathbf{F}(\mathbf{v}) : \mathbb{R}^{N_\pi} \rightarrow \mathbb{R}^{N_\pi}$, is a function describing the nonlinear elements of the circuit and, $\mathbf{D}_{\phi_i} = [d_{i,j}]$ with $d_{i,j} \in \{0,1\}$ where $i \in \{1,2,\dots,N_\pi\}$, and $j \in \{1,2,\dots,N_{\phi_i}\}$, with a maximum of one non-zero in each row or column is a selector matrix that maps $\mathbf{i}_{\phi_i}(t) \in \mathbb{R}^{N_{\phi_i}}$ the vector of currents entering the i th linear subnetwork nodes, into the node space $\mathbb{R}^{N_\pi}$ of the nonlinear network π .

The first, second, fourth and fifth terms in (7.1) cover the lumped components and independent sources of the network π . The third term describes currents at the subnetwork terminals, and then maps them into the rest of the network through the matrices $\mathbf{D}_{\phi_i}$. A linear multi-terminal network ϕ_i can be characterized in the frequency domain by any of the usual port descriptions such as $\mathbf{Z}(s)$, $\mathbf{Y}(s)$, $\mathbf{H}(s)$ or $\mathbf{S}$ parameters. Without loss of generality, the relation between the terminal voltages of the linear subnetwork ϕ_i and its terminal currents can be written in the form,

$$\mathbf{Z}_{\phi_i}(s) \mathbf{I}_{\phi_i}(s) = \mathbf{V}_{\phi_i}(s) \quad (7.2)$$

where s is the Laplace operator,

$\mathbf{Z}_{\phi_i}(s)$ is the complex frequency domain impedance representation of the subnetwork ϕ_i ,

$\mathbf{V}_{\phi_i}(s)$ is the vector of terminal voltages of the nodes that connect the subnetwork ϕ_i to the network π and

$\mathbf{I}_{\phi_i}(s) = \mathcal{L}[\mathbf{i}_{\phi_i}(t)]$, where $\mathcal{L}$ denotes the Laplace transform operator.

Example of the formulation

Figure 7.2: Example nonlinear network and formulation.

To illustrate the formulation method, consider the network shown in Fig. 7.2. In this example there is only one linear subnetwork ϕ . The network can be described by (7.1) where,

$$\mathbf{v}_\pi = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ i_E \\ Q \end{bmatrix}, \quad \mathbf{F}(\mathbf{v}_\pi(t)) = \begin{bmatrix} -f_1(v_2 - v_1) \\ f_1(v_2 - v_1) \\ f_2(v_3) \\ 0 \\ 0 \\ -f_3(v_4) \end{bmatrix}, \quad \mathbf{e}_\pi = \begin{bmatrix} 0 \\ 0 \\ 0 \\ E(t) \\ 0 \end{bmatrix},$$

$$\mathbf{C}_\pi = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C' & -C' & 0 & 0 \\ 0 & 0 & -C' & C' & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{G}_\pi = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{i}_\pi = \begin{bmatrix} i_2 \\ i_3 \end{bmatrix}, \quad \mathbf{D}_\pi = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} \begin{bmatrix} i_2 \\ i_3 \end{bmatrix} = \begin{bmatrix} v_2 \\ v_3 \end{bmatrix}$$

The difficulty in solving interconnect structures formulated with (7.1) and (7.2) simultaneously is due to the fact that they implicitly contain a mixture of frequency and time domain representations. Moment methods have been shown to be an efficient way to analyze large interconnect structures containing mixed frequency/time domain formulations, often 2 orders of magnitude faster than conventional simulation.

7.2 Differential equations (SPICE) approach

Given one current input i and one output j , the input-output trans-impedance given by a q pole approximation may be described as,

$$Z^{(i,j)}(s) = e^{(i,j)} + \frac{\sum_{m=0}^{q-1} a_m^{(i,j)} s^{q-m-1}}{\sum_{m=0}^q b_m^{(i,j)} s^{q-m}} \quad (7.3)$$

or, by partial fraction decomposition,

$$Z^{(i,j)}(s) = e^{(i,j)} + \sum_{m=1}^q \frac{\hat{k}_j^{(i,j)}}{s - \hat{p}_m^{(i,j)}} \quad (7.4)$$

A Z-parameter macromodel of the of the linear network is then

$$\mathbf{Z}_\phi(s) = \begin{bmatrix} z^{(1,1)} & z^{(1,2)} & \dots & z^{(1,n_\phi)} \\ z^{(2,1)} & z^{(2,2)} & \dots & z^{(2,n_\phi)} \\ \vdots & \vdots & \ddots & \vdots \\ z^{(n_\phi,n_\phi)} & z^{(n_\phi,n_\phi)} & \dots & z^{(n_\phi,n_\phi)} \end{bmatrix} \quad (7.5)$$

In order to generate $\mathbf{Z}_\phi(s)$ for the partition ϕ , a unit impulse current is connected to each terminal i and the output voltage approximation at each node j is calculated, while keeping all other terminals open-circuited. Since, when using a moment-matching approximation, the moments are generated for every port j simultaneously for a given input voltage, there is only a need to vary the input currents i . Even here, the major task of an LU factorization has already been performed in the first calculation, and the only additional calculation is to start from $\mathbf{M}_0$ and recursively work up, doing this once for every input current. This is equivalent to calculating extra sets of moments in a single approximation.

Two methods of including the macromodel equations into conventional circuit simulators have been developed. The first method in (7.3) uses polynomial expressions for the impedance representation. The second method in (7.4) uses the pole/residue

(partial fraction) expressions. In either case, the denominator polynomial or the poles should be the same for every entry of the macromodel. If they are different for each entry, the CPU time will be increased. A partial Padé approximation (Chapter 5) may be used on poles or a denominator generated at the driving point(s). The residues of each of the output nodes may then be calculated using this denominator or these poles. While both methods for including the macromodel equations into a circuit simulator are described here in terms of Z-parameters there is no reason why Y-parameters could not be used.

7.2.1 Polynomial expression

Using (7.3), and assuming the same denominator for each input-output transfer function, a Z-parameter port description, $\mathbf{Z}_\phi(s)$ of an n_ϕ linear subnetwork ϕ can be given by, $\mathbf{Z}_\phi(s) =$

$$\begin{bmatrix} c^{(1,1)} + \frac{\sum_{m=0}^{q-1} a_m^{(1,1)} s^{q-m-1}}{\sum_{m=0}^q b_m s^{q-m}} & \dots & c^{(1,n_\phi)} + \frac{\sum_{m=0}^{q-1} a_m^{(1,n_\phi)} s^{q-m-1}}{\sum_{m=0}^q b_m s^{q-m}} \\ \vdots & \ddots & \vdots \\ c^{(n_\phi,1)} + \frac{\sum_{m=0}^{q-1} a_m^{(n_\phi,1)} s^{q-m-1}}{\sum_{m=0}^q b_m s^{q-m}} & \dots & c^{(n_\phi,n_\phi)} + \frac{\sum_{m=0}^{q-1} a_m^{(n_\phi,n_\phi)} s^{q-m-1}}{\sum_{m=0}^q b_m s^{q-m}} \end{bmatrix} \quad (7.6)$$

By introducing new state variables: $\{x_{j,i} | i = 1, 2, \dots, n_\phi; j = 1, 2, \dots, q\}$, (7.6) can be represented in the time domain by a set of differential equations in the form,

$$\begin{aligned} \frac{d}{dt} \mathbf{x}_i(t) - \mathbf{A}_i \mathbf{x}_i(t) - \mathbf{B}_i \mathbf{i}_\phi &= 0 \\ \mathbf{v}_{\phi,i}(t) - \mathbf{C}_i \mathbf{x}_i(t) - \mathbf{D}_i \mathbf{i}_\phi &= 0 \end{aligned} \quad (7.7)$$

where,

$$\mathbf{x}_i(t) = \begin{bmatrix} x_{1,i} \\ x_{2,i} \\ \dots \\ x_{q,i} \end{bmatrix}, \quad v_{\phi,i}(t) = [v_{\phi}(t)]_i$$

$$\mathbf{A}_i = \begin{bmatrix} 0 & 0 & \dots & 0 & -\frac{b_1}{b_0} \\ \frac{1}{b_{q-1}} & 0 & \dots & 0 & -\frac{b_1}{b_0} \\ 0 & \frac{b_{q-1}}{b_{q-2}} & \dots & 0 & -\frac{b_1}{b_0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \frac{b_2}{b_0} & -\frac{b_1}{b_0} \end{bmatrix} \quad \mathbf{B}_i = \frac{b_1}{b_0} \begin{bmatrix} \frac{a_{q-1}^{(i,1)}}{b_{q-1}} & \dots & \frac{a_{q-1}^{(i,n_{\phi})}}{b_{q-1}} \\ \frac{a_{q-2}^{(i,1)}}{b_{q-1}} & \dots & \frac{a_{q-2}^{(i,n_{\phi})}}{b_{q-1}} \\ \frac{a_{q-3}^{(i,1)}}{b_{q-2}} & \dots & \frac{a_{q-3}^{(i,n_{\phi})}}{b_{q-2}} \\ \vdots & \ddots & \vdots \\ \frac{a_0^{(i,1)}}{b_1} & \dots & \frac{a_0^{(i,n_{\phi})}}{b_1} \end{bmatrix}$$

$$\mathbf{C}_i = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

$$\mathbf{D}_i = \begin{bmatrix} c^{(i,1)} & c^{(i,2)} & \dots & c^{(i,n_{\phi})} \end{bmatrix} \quad (7.8)$$

Using (7.1) and (7.7) the time domain equations for the whole network can be written in the form,

$$\mathbf{C}_{\pi} \frac{d}{dt} \mathbf{v}_{\pi}(t) + \mathbf{G}_{\pi} \mathbf{v}_{\pi}(t) + \mathbf{D}_{\phi} \mathbf{i}_{\phi}(t) + \mathbf{F}(\mathbf{v}_{\pi}(t)) - \mathbf{b}_{\pi}(t) = 0$$

$$\frac{d}{dt} \mathbf{X} - \mathbf{A} \mathbf{X} - \mathbf{B} \mathbf{i}_{\phi} = 0$$

$$\mathbf{D}_{\phi}^T - \mathbf{C} \mathbf{X} - \mathbf{D} \mathbf{i}_{\phi} = 0 \quad (7.9)$$

where,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{n_{\phi}} \end{bmatrix} \quad \mathbf{v}_{\phi} = \begin{bmatrix} v_{\phi,1} \\ v_{\phi,2} \\ \vdots \\ v_{\phi,n_{\phi}} \end{bmatrix}$$

$$\begin{aligned}
 \mathbf{A} &= \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{n_\phi} \end{bmatrix} & \mathbf{B} &= \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_{n_\phi} \end{bmatrix} \\
 \mathbf{C} &= \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}_{n_\phi} \end{bmatrix} & \mathbf{D} &= \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \\ \vdots \\ \mathbf{D}_{n_\phi} \end{bmatrix}
 \end{aligned} \tag{7.10}$$

These equations can be inserted directly into a nonlinear differential equation solver such as SPICE[Kum91].

Example of the formulation

Consider the network shown in Fig. 7.2. Using $q = 3$, the time domain equivalent representation of the linear subnetwork ϕ can be described by (7.9) with,

$$\begin{aligned}
 \mathbf{x} &= \begin{bmatrix} x_{1,1} & x_{2,1} & x_{3,1} & x_{1,2} & x_{2,2} & x_{3,2} \end{bmatrix}^T \\
 \mathbf{A} &= \begin{bmatrix} 0 & 0 & -\frac{a_1}{a_0} & 0 & 0 & 0 \\ \frac{1}{a_2} & 0 & -\frac{a_1}{a_0} & 0 & 0 & 0 \\ 0 & \frac{a_2}{a_1} & -\frac{a_1}{a_0} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{a_1}{a_0} \\ 0 & 0 & 0 & \frac{1}{a_2} & 0 & -\frac{a_1}{a_0} \\ 0 & 0 & 0 & 0 & \frac{a_2}{a_1} & -\frac{a_1}{a_0} \end{bmatrix} & \mathbf{B} &= \begin{bmatrix} b_2^{1,1} & b_2^{1,2} \\ \frac{b_1^{1,1}}{a_2} & \frac{b_1^{1,2}}{a_2} \\ \frac{b_0^{1,1}}{a_1} & \frac{b_0^{1,2}}{a_1} \\ b_2^{2,1} & b_2^{2,2} \\ \frac{b_1^{2,1}}{a_2} & \frac{b_1^{2,2}}{a_2} \\ \frac{b_0^{2,1}}{a_1} & \frac{b_0^{2,2}}{a_1} \end{bmatrix} \\
 \mathbf{C} &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} & \mathbf{D} &= \begin{bmatrix} b_3^{1,1} & b_3^{1,2} \\ b_3^{2,1} & b_3^{2,2} \end{bmatrix}
 \end{aligned}$$

Figure 7.3: State space impedance model of a single pole.

7.2.2 Pole/residue expression

A pole/residue expression may be obtained either by decomposing the rational expression obtained using one expansion point, or by collecting poles and residues using hops. Using (7.4), and assuming the same roots for each input-output transfer function, a Z-parameter port description, $\mathbf{Z}_\phi(s)$ of an n_ϕ linear subnetwork ϕ can be given by,

$$\mathbf{Z}_\phi(s) = \begin{bmatrix} c^{(1,1)} + \sum_{j=1}^q \frac{\hat{k}_j^{(1,1)}}{s - \hat{p}_j} & \dots & c^{(1,n_\phi)} + \sum_{j=1}^q \frac{\hat{k}_j^{(1,n_\phi)}}{s - \hat{p}_j} \\ \vdots & \ddots & \vdots \\ c^{(n_\phi,1)} + \sum_{j=1}^q \frac{\hat{k}_j^{(n_\phi,1)}}{s - \hat{p}_j} & \dots & c^{(n_\phi,n_\phi)} + \sum_{j=1}^q \frac{\hat{k}_j^{(n_\phi,n_\phi)}}{s - \hat{p}_j} \end{bmatrix} \quad (7.11)$$

To model this transfer function in circuit elements consider the state space diagram of a single pole shown in Fig. 7.3 and its impedance circuit equivalence shown in Fig. 7.4. In actual implementation a resistor is used in parallel with the capacitance C to avoid a floating node in SPICE. A suitably large value that will have negligible impact on circuit operation should be used.

The transfer function shown in both Fig. 7.3 and Fig. 7.4 is

$$\frac{V}{I} = \frac{a}{s - b} \quad (7.12)$$

Complex values for a and b are not supported by any time domain simulators so complex poles are represented using the state-space diagram of Fig. 7.5 which can be realized using real-valued gains for a time domain simulator. Using real-valued gains,

Figure 7.4: Circuit representation of Fig. 7.3.

the transfer function of Fig. 7.5 can be given as.

$$\frac{V}{I} = \frac{ds + c}{s^2 - as - b} \quad (7.13)$$

A large equation such as (7.11) may be represented using multiple inputs and summing the operation of parallel-pole realizations as shown in Fig. 7.6. For realization in a circuit simulator complex poles must be merged in conjugate pairs and represented in the real-valued form of Fig. 7.5. A circuit representation similar to that of Fig. 7.4 can then be constructed.

Figure 7.5: State space impedance model of a complex pole using real valued gains.

Figure 7.6: State space model of the first row of the Z-parameter macro model of (7.11).

7.3 Recursive convolution approach

An alternative method to simulate a linear network with nonlinear terminations outside of the context of SPICE is to transform the macromodel equations into the time domain [CN92b]. The partial fraction decomposition of the macromodel in (7.11) is rewritten as an equivalent time domain model,

$$\mathbf{Z}_\phi(t) = \begin{bmatrix} c^{(1,1)}\delta(t) + \sum_{j=1}^q \hat{k}_j^{(1,1)} e^{\hat{p}_j t} & \dots & c^{(1,n_\phi)}\delta(t) + \sum_{j=1}^q \hat{k}_j^{(1,n_\phi)} e^{\hat{p}_j t} \\ \vdots & \ddots & \vdots \\ c^{(n_\phi,1)}\delta(t) + \sum_{j=1}^q \hat{k}_j^{(n_\phi,1)} e^{\hat{p}_j t} & \dots & c^{(n_\phi,n_\phi)}\delta(t) + \sum_{j=1}^q \hat{k}_j^{(n_\phi,n_\phi)} e^{\hat{p}_j t} \end{bmatrix} \quad (7.14)$$

or,

$$\mathbf{Z}_\phi(t) = \hat{\mathbf{C}}\delta(t) + \sum_{j=1}^q \hat{\mathbf{K}}_j e^{\hat{p}_j t} \quad (7.15)$$

A required output response $\hat{\mathbf{v}}(t)$ is given by a convolution of all input currents $\hat{\mathbf{i}}(t)$ and the impulse transfer function,

$$\hat{\mathbf{v}}(t) = \mathbf{Z}_\phi(t) * \hat{\mathbf{i}}(t) = \int_0^t \mathbf{Z}_\phi(t - \tau) \hat{\mathbf{i}}(\tau) d\tau \quad (7.16)$$

The input currents, in the case of nonlinear resistive terminations can be given in the simple case by a nonlinear function of the voltage nodes and general input sources.

$$\hat{\mathbf{i}}(t) = \mathbf{f}(\hat{\mathbf{v}}(t)) + \mathbf{e}(t) \quad (7.17)$$

Applying (7.15) and (7.16) we get,

$$\hat{\mathbf{v}}(t) = \int_0^t \left(\hat{\mathbf{C}}\delta(t - \tau) + \sum_{j=1}^q \hat{\mathbf{K}}_j e^{\hat{p}_j(t-\tau)} \right) \hat{\mathbf{i}}(\tau) d\tau \quad (7.18)$$

or,

$$\hat{\mathbf{v}}(t) = \hat{\mathbf{C}}\hat{\mathbf{i}}(t) + \sum_{j=1}^q \int_0^t \hat{\mathbf{K}}_j e^{\hat{p}_j(t-\tau)} \hat{\mathbf{i}}(\tau) d\tau = \hat{\mathbf{C}}\hat{\mathbf{i}}(t) + \sum_{j=1}^q \hat{\mathbf{v}}_j(t) \quad (7.19)$$

Therefore,

$$\hat{\mathbf{v}}_j(t + \Delta t) = \int_0^{t+\Delta t} \hat{\mathbf{K}}_j e^{\hat{p}_j(t+\Delta t-\tau)} \hat{\mathbf{i}}(\tau) d\tau \quad (7.20)$$

Expanding,

$$\dot{\mathbf{v}}_j(t + \Delta t) = \int_0^t \hat{\mathbf{K}}_j e^{\hat{\mathbf{p}}_j(t+\Delta t-\tau)} \dot{\mathbf{i}}(\tau) d\tau + \int_t^{t+\Delta t} \hat{\mathbf{K}}_j e^{\hat{\mathbf{p}}_j(t+\Delta t-\tau)} \dot{\mathbf{i}}(\tau) d\tau \quad (7.21)$$

Applying trapezoidal integration on the second term and pulling out the exponential from the integration in the first term,

$$\dot{\mathbf{v}}_j(t + \Delta t) = e^{\hat{\mathbf{p}}_j \Delta t} \dot{\mathbf{v}}_j(t) + \frac{\Delta t}{2} \hat{\mathbf{K}}_j \left(\dot{\mathbf{i}}(t + \Delta t) + e^{\hat{\mathbf{p}}_j \Delta t} \dot{\mathbf{i}}(t) \right) \quad (7.22)$$

The last equation is a recursive one which has effectively eliminated the convolution. It essentially says that every transient time point only depends on the previous time point. However, the calculations must be kept per pole per timepoint.

Using (7.17) for $\dot{\mathbf{i}}(t)$ and writing $\dot{\mathbf{v}}(t)$ in term of its components $\dot{\mathbf{v}}_j(t)$,

$$\begin{aligned} \dot{\mathbf{v}}(t + \Delta t) &= \hat{\mathbf{C}} \dot{\mathbf{i}}(t) + \sum_{j=1}^q e^{\hat{\mathbf{p}}_j \Delta t} \dot{\mathbf{v}}_j(t) \\ &+ \frac{\Delta t}{2} \sum_{j=1}^q \hat{\mathbf{K}}_j \left[\mathbf{f}(\dot{\mathbf{v}}(t + \Delta t) + \mathbf{e}(t + \Delta t)) + e^{\hat{\mathbf{p}}_j \Delta t} (\mathbf{f}(\dot{\mathbf{v}}(t) + \mathbf{e}(t))) \right] \end{aligned} \quad (7.23)$$

or,

$$\dot{\mathbf{v}}(t + \Delta t) = \mathbf{A}_1 + \mathbf{A}_2 \mathbf{f}(\dot{\mathbf{v}}(t + \Delta t)) \quad (7.24)$$

a nonlinear equation to be solved at each time point.

Nonlinear simulation highlights the problems of moment-matching in that, generally, more accurate models are needed than when simulating only the linear subnetworks. This is because nonlinear terminations have a tendency to spread the spectrum of the frequency response. So while the input may be, for example, 1 MHz, the output may contain frequency harmonics in the 2 MHz range or higher.

Because of this problem CFH is particularly appealing because it can ensure the accuracy of the frequency response of the linear subnetwork up to a pre-defined frequency of operation. The choice of f_{max} in a nonlinear circuit will depend mostly on the type of nonlinear elements used and therefore on the circuit designers.

Another very important property of CFH that becomes useful in macromodeling is that of pole selection. Because CFH generates the exact dominant poles of the

network, then the poles are the same at every output node and for every input. The only exception are poles that are not significantly excited (i.e. small residues) at a specific port and are therefore not detected. This means that any output selected yields generally the same results. This is in contrast to generating approximate poles in AWE and using poles generated at one node to obtain the residues at other nodes[XN93][CN92d][AGKP92]. The latter is a problem in that the approximate poles at one node may not be good for another node. The technique of [AGKP92] is best for using one expansion point but is insufficient in that it may not obtain all of the necessary dominant poles.

Example 1

An interconnect circuit which contains lossless transmission lines and nonlinear components is shown in Fig. 7.7. The inverter inverter driver/receiver is CMOS circuitry. The pure delays of the transmission lines are distributed between 161 ps and 255 ps. The linear subnetwork is replaced by a lumped component model derived from the differential equations obtained using a 6th order AWE. The combined network is simulated using HSPICE. The applied voltage pulse was 10 ps rise/fall time and 6 ns pulse width. The transient results are displayed in Fig. 7.8, Fig. 7.9 and Fig. 7.10, as well as the comparison with a full simulation using HSPICE.

Example 2

In this example, comparison between the CPU time of AWE method and that of HSPICE is made with three MCM interconnect circuits on SUN SPARC II workstation. Each circuit is constructed by cascading a number of linear subcircuits as shown in Fig. 7.11. The comparison results are listed in Table 7.1. The CPU time of the AWE method consists of two parts: the CPU time of running AWE to produce the approximate Z-parameters of the linear subnetwork and the CPU time to solve the combined differential equations using HSPICE.

Figure 7.7: Lossless transmission line network with nonlinear drivers/receivers.

Figure 7.8: Transient response of Fig. 7.7 at V_1 .

Figure 7.9: Transient response of Fig. 7.7 at V_2 .

Figure 7.10: Transient response of Fig. 7.7 at V_4 .

Figure 7.11: Large multi-chip interconnect network with nonlinear termination.

Example 3

Fig. 7.12 shows a circuit which consists of two lossy coupled transmission lines. The nonlinear components are the same as those in Example 1. Both lines are of the same length 0.1m. Line 1 has the following parameters:

$$\mathbf{L} = \begin{bmatrix} 494.6 & 63.3 \\ 63.3 & 494.6 \end{bmatrix} \frac{nH}{m} \quad \mathbf{C} = \begin{bmatrix} 62.8 & -1.9 \\ -1.9 & 62.8 \end{bmatrix} \frac{pF}{m}$$

$$\mathbf{R} = \begin{bmatrix} 75 & 15 \\ 15 & 75 \end{bmatrix} \frac{\Omega}{m} \quad \mathbf{G} = \begin{bmatrix} 0.1 & -0.01 \\ -0.01 & 0.1 \end{bmatrix} \frac{S}{m}$$

and the parameters for line 2 are,

$$\mathbf{L} = \begin{bmatrix} 494.6 & 63.3 & 7.8 & 0.0 \\ 63.3 & 494.6 & 63.3 & 7.8 \\ 7.8 & 63.3 & 494.6 & 63.3 \\ 0.0 & 7.8 & 63.3 & 494.6 \end{bmatrix} \frac{nH}{m}$$

Circuit No.	Number of Transmission Lines	Node Count	Lumped Element Count	CPU seconds	
				AWE	HSPICE
1	35	102	147	9.74	86.9
2	70	202	292	10.41	162.05
3	98	182	408	12.12	232.03

Table 7.1: CPU cost (SUN SPARC II) of nonlinear AWE compared to HSPICE.

$$\mathbf{C} = \begin{bmatrix} 62.8 & -4.9 & -0.3 & 0.0 \\ -4.9 & 62.8 & -4.9 & -0.3 \\ -0.3 & -4.9 & 62.8 & -4.9 \\ 0.0 & -0.3 & -4.9 & 62.8 \end{bmatrix} \frac{pF}{m}$$

$$\mathbf{R} = \begin{bmatrix} 50 & 10 & 1 & 0.0 \\ 10 & 50 & 10 & 1 \\ 1 & 10 & 50 & 10 \\ 0.0 & 1 & 10 & 50 \end{bmatrix} \frac{\Omega}{m}$$

$$\mathbf{G} = \begin{bmatrix} 0.1 & -0.01 & -0.001 & 0.0 \\ -0.01 & 0.1 & -0.01 & -0.001 \\ -0.001 & 0.01 & 0.1 & 0.0 \\ 0.0 & -0.001 & 0.01 & 0.1 \end{bmatrix} \frac{S}{m}$$

The applied voltage is a pulse with 10 ps rise/fall time and 6 nanoseconds pulse width. The output response obtained using the AWE method is shown in Fig. 7.13, Fig. 7.14, Fig. 7.15 and Fig. 7.16. Note the smooth switching that occurs even though the signal before the receivers has lost some of its integrity.

Figure 7.12: Coupled lossy line network with nonlinear terminations.

Example 4

A clock tree circuit (Fig. 7.18) containing 73 lossy lines, 2895 resistors, and 2777 capacitors and driven by two cascaded CMOS inverters was partitioned into linear and nonlinear subcircuits. The linear portion was analyzed by CFH. The resulting poles and residues were then used to form a nonlinear macromodel. The output was generated on two of the clock pins chosen at random. The poles on these two clock pins were generated separately to show the accuracy of the CFH algorithm in obtaining the same poles at different outputs. The poles are displayed in the complex plane in Fig. 7.17. A 5 V input signal with a rise time of 0.1 ns was placed on the input to the inverters. The resulting output at a clock line is shown in Fig. 7.19. The CPU time for the linear analysis was 33.4 seconds and the nonlinear analysis for 10 ns run with a step of 0.04 ns consumed 53.7 seconds on a SUN SPARC II. The result compared favorably with SPICE.

Figure 7.13: Transient response of Fig. 7.12 at V_1 .

Figure 7.14: Transient response of Fig. 7.12 at V_2 .

Figure 7.15: Transient response of Fig. 7.12 at V_3 .

Figure 7.16: Transient response of Fig. 7.12 at V_4 .

Figure 7.17: The dominant poles of the network at two clock pins as generated by CFH, 'x' for one output, 'o' for the other

Figure 7.18: A clock tree network driven by 2 CMOS inverters.

Figure 7.19: Transient response of Fig. 7.18.

Chapter 8

Other Applications

8.1 Eigenvalue calculation

In the case of a lumped element formulation, the poles of a network are the values of s that satisfy,

$$|\mathbf{G}_\phi + s\mathbf{C}_\phi| = 0 \quad (8.1)$$

This problem is known as the generalized eigenvalue problem. However, in the case of networks containing distributed components, the poles of the network are described by the values of s that satisfy,

$$|\tilde{\mathbf{Y}}_\phi(s)| = 0 \quad (8.2)$$

In this more general case, the number of poles that satisfy the equation can be infinite. That is $N_p = \infty$ in (2.14). I am not aware of any published algorithm for generating the poles of such a distributed network and a finite algorithm for generating all of the poles is not possible in the general case. CFH gives a general technique for calculating the dominant eigenvalues of a lumped or distributed linear network. This is significant that the poles of distributed systems are transcendental and infinite in number and there is no general algorithm for calculating them. Generally, known algorithms, such as the QR algorithm or the iteration (power) method with shifting[Wat82] are used

for lumped networks.

The QR algorithm is useful for calculating all the eigenvalues of a network and is practical for only relatively small networks. The inverse (or regular) iteration with shifting is a "power method". It finds a single smallest (largest) eigenvalue/eigenvector pair. It does not find multiple eigenvalues simultaneously at a given shift. CFH finds multiple eigenvalues at a given shift. The iteration method has an important weakening assumption. The eigenvalue modulus at the shift point must be the largest of all of the eigenvalues. This means that without a-priori knowledge of the eigenvalues, there is no guarantee of convergence. For example, the method will not be able to find a set of complex conjugate eigenvalues of a real matrix A , or eigenvalues of multiplicity. CFH has no problems with eigenvalues of same modulus because convergence is to M eigenvalues simultaneously. Convergence in the iteration method can be slow if the next smallest (largest) eigenvalues are comparable to the smallest (largest) eigenvalue. In CFH, convergence is not slowed down by this fact because the larger the M the faster the convergence. Once one eigenvalue is converged to, there is no criterion for choosing the shift in the iteration method for the next eigenvalue. Depending on where one shifts to, the same eigenvalue may result, or no eigenvalues may be found (due to equal modulus). CFH provides criteria for choosing the shifts and for minimizing them. Most importantly, CFH allows all of these points to be true in distributed networks, not possible by any other published algorithm. CFH generates only the dominant eigenvalues within a given search area and is therefore far less CPU intensive.

8.2 PEEC simulation

As switching speeds extend into the gigahertz range, two-dimensional transmission line interconnect simulation models become inadequate due to the spatial EM effects of the 3-dimensional structures. The 3-D structures can include microstrips, striplines,

vias, or planar structures such as patch antennas or ground planes. The modeling of these structures has been successfully accomplished in the past using partial element equivalent circuit (PEEC) models[Rue74]. PEEC models are RLC circuits where the individual resistances and capacitances are extracted from the geometry using a quasi-static (non retarded) solution of the Maxwell equations. The models, extracted from the underlying EM field effects, require solutions both in the frequency and time domains. Although the large resultant networks are solvable with general purpose circuit simulators like SPICE or ASTAP, the CPU time required is generally large.

PEEC models, like all circuit models, do not include the effect of the finite speed of light (retardation). Retardation must be included if the travel distance between the coupled elements is much larger than one tenth of the wavelength involved. This limitation is not important for geometries where the propagation is on wires with nearby ground planes. If there is no nearby ground plane and the frequencies involved are well above 100 MHz, large errors were shown to exist even for simple interconnect problems[Hee92]. The absence of ground planes is a common occurrence in packages like chip carriers. Another situation, where retardation is always required, is the EM interference problem[Pau92][Rue92].

The rPEEC algorithm addresses these problems by solving retarded PEEC models thereby turning PEEC into a *full-wave method*. Simulating PEEC and rPEEC models in frequency domain is time-consuming even when specialized circuit simulators are used because a full matrix representing the circuit with all cross couplings has to be solved for each frequency. If the resonances are important many frequency point solutions are necessary in order not to miss a resonance. In the time domain, rPEEC models are far more memory- and CPU-intensive to simulate than PEEC models because the voltage history for each node in the circuit has to be stored[Hee91]. It can also be shown that the errors caused by this type of simulation can grow without bounds unless even finer subdivisions of the original problem are used[Tij87]. Therefore the problem size that can be solved successfully in time domain has been

much smaller than the one for frequency domain. AWE has been applied in an attempt to solve this problem[HRBR92]. However AWE has not been accurate at all necessary frequency ranges and CFH has been applied to the problem[CHNR93]. CFH, as will be shown, can allow the extraction of accurate resonant frequencies often needed for a full 3-D analysis.

The time-domain modeling of 3-D interconnect structures is especially important in Mbit DRAM and SRAM cell plant and contact design[KRS90]. In these devices, it is important to minimize the delay not only between the cells but also inside the cell. Several 3-D wiring capacitance programs have been developed specifically for these networks, but they have been prevented from analyzing the resultant equations in the time domain due to the large CPU time required. I will describe here an application of AWE to analyze these equations for a transient 3-D analysis.

8.2.1 PEEC and rPEEC

The rPEEC formulation of a 3-D structure is given in this section in terms of a model representation which can be implemented with circuit elements. Beginning with the usual integral equation which forms the basis for several formulations like the method of moments[Har68] and the PEEC method, we have,

$$0 = \frac{J(x, t)}{\sigma} + \frac{\partial \bar{A}(x, t)}{\partial t} + \nabla \Phi(x, t) \quad (8.3)$$

$$A(x, t) = \frac{\mu_0}{4\pi} \int \frac{J(x', t')}{R} d^3x' \quad (8.4)$$

$$\Phi(x, t) = \frac{1}{4\pi\epsilon\epsilon_0} \int \frac{\rho(x', t')}{R} d^3x' \quad (8.5)$$

If the charged surfaces are subdivided into n elements each with a potential Φ_i and a charge Q_i , the following approximation of (8.5) results

$$\Phi_i(t) = \sum_{j=1}^n \mu_{ij} Q_j(t'_{ij}) \quad (8.6)$$

$$t'_{ij} = t - \frac{R_{ij}}{v} \quad (8.7)$$

where R_{ij} is the distance between conductor i and j and v is the speed of light. If $t'_{ij} = t$ (infinite speed of light) is assumed, the above equation can be inverted into the familiar capacitance equation. With retardation, (8.6) cannot be inverted and the circuit must therefore be expressed in terms of the *coefficients of potential* p_{ij} . the capacitances of an arbitrary circuit are replaced by *pseudo-capacitances* $c'_i = 1/p_{ii}$ and controlled voltage sources.

$$V_i^{(C)}(t) = \sum_{j \neq i} \frac{p_{ij}}{p_{jj}} \Phi'_j(t'_{ij}) \quad (8.8)$$

where Φ'_j is the potential of pseudo-capacitance j , as shown in Fig. 8.1. Mutual inductances can be replaced by a voltage source (in series with the corresponding inductance),

$$V_i^{(L)}(t) = \sum_{j \neq i} \frac{L_{pij}}{L_{pjj}} V'_j(t'_{ij}) \quad (8.9)$$

where V'_j is the voltage over inductance L_j and L_{pij} is the partial inductance between inductances i and j .

Figure 8.1: The PEEC voltage source model of two potential nodes.

In this formulation, *local* terms are included as inductances or (pseudo-) capacitances while all *non-local* interactions are represented as voltage controlled voltage sources. The local capacitances and inductances are not affected by retardation whereas the voltage sources depend on voltages *at past times* t'_{ij} .

In the frequency domain, retardation can be readily described. The Laplace transform of (8.7) and (8.8) is

$$U_i^{(C)}(s) = \sum_{j \neq i} \frac{p_{ij}}{p_{jj}} \Phi'_j(s) \exp\left(\frac{-sR_{ij}}{c}\right) \quad (8.10)$$

with s , the Laplace transform operator equal to $j\omega$. The retardation results in a *phase factor* multiplied into the non-retarded expression in the frequency domain. Similarly for (8.7) and (8.9),

$$U_i^{(L)}(s) = \sum_{j \neq i} \frac{L_{pij}}{L_{pjj}} V'_j(s) \exp\left(\frac{-sR_{ij}}{c}\right) \quad (8.11)$$

In order to obtain the frequency response for a frequency band, typically, hundreds of frequency point analyses are required. In the time-domain a “history” buffer of the capacitance potentials Φ'_i and the inductance voltages V'_i has to be stored, requiring a fixed number of memory cells. The history is kept in fixed time-intervals τ , identical for all circuit elements. This is analogous to the method required when simulating lossless delays in transmission lines. Due to the large *full* circuit matrices that have to be solved and, in retarded time domain, due to the history storage, PEEC and rPEEC methods are memory and CPU time intensive. Unless care is taken in retarded time domain simulations, the numerical interpolation errors can increase without bounds. To remedy this situation, *more* subdivisions are needed in retarded time domain than other PEEC methods. Because of this, many problems that were solved successfully with frequency domain rPEEC could not be solved in time domain using these methods. AWE and CFH however provide alternative techniques to simulating large PEEC networks.

We begin with the MNA formulation in (3.1), except that in the case of rPEEC,

the equations are best described as,

$$\mathbf{C}_\phi \frac{d}{dt} \mathbf{v}_\phi(t) + \mathbf{G}_\phi \mathbf{v}_\phi(t) + \sum_{i=1}^{N_c} \mathbf{D}_i^{(C)} \dot{V}_i^{(C)}(t) + \sum_{i=1}^{N_l} \mathbf{D}_i^{(L)} \dot{V}_i^{(L)}(t) - \mathbf{e}_\phi(t) = 0 \quad (8.12)$$

where,

N_c and N_l are the number of capacitive and inductive partial voltage source elements respectively.

$\mathbf{v}_\phi(t)$, $\mathbf{e}_\phi(t)$, $\mathbf{C}_\phi$ and $\mathbf{G}_\phi$ are as defined for (2.30).

$\mathbf{D}_k^{(C)} = [d_i]$ with elements $d_i \in \{0, 1\}$ and $i \in \{1, 2, \dots, N_\phi\}$ is a selector vector that maps $V_i^{(C)}(t)$ into the node space $\mathcal{R}^{N_\phi}$ of the network ϕ ; and similarly for $\mathbf{D}_k^{(L)}$ with respect to $V_i^{(L)}(t)$.

The $V_i^{(C)}(t)$ and $V_i^{(L)}(t)$ are, in turn, described in the frequency domain by (8.10) and (8.11) respectively, where the Φ'_j and V'_j are variables of the system included in $\mathbf{V}_\phi(s) = \mathcal{L}(\mathbf{v}_\phi(t))$.

Equations (8.10), (8.11) and (8.12) can, in turn be combined into a set of linear equations in the form of (3.7). In order to generate the moments of the network about $s = 0$, (3.10) and (3.11) are used. The derivatives of the stamps of $\tilde{\mathbf{Y}}(s)$ will yield the derivatives of the matrix. The derivatives of the capacitance retardation stamps can be generated using the relation,

$$\frac{p_{ij}}{p_{jj}} \exp\left(\frac{-sR_{ij}}{v}\right) = \frac{p_{ij}}{p_{jj}} \sum_{n=0}^{\infty} \frac{(-sR_{ij})^n}{n!v^n} \quad (8.13)$$

A similar relation holds with the inductance retardation stamps.

In order to expand about any other complex point $\alpha + j\beta$ in the frequency s plane, it is sufficient to shift the axis to the desired expansion point and expand it about $s' = 0$, where s' is the new shifted variable s . Replacing $s = s' + (\alpha + j\beta)$ in (8.13) we get,

$$\frac{p_{ij}}{p_{jj}} \exp\left(\frac{-(s' + \alpha + j\beta)R_{ij}}{v}\right) = \frac{p_{ij}}{p_{jj}} \exp\left(\frac{-(\alpha + j\beta)R_{ij}}{v}\right) \sum_{n=0}^{\infty} \frac{(-s'R_{ij})^n}{n!v^n} \quad (8.14)$$

This gives the Taylor's expansion about $s' = 0$ which in turn is used.

8.3 Resonant frequencies

Resonant frequencies in PEEC, rPEEC and other linear networks are difficult to detect using regular methods due to the fine frequency sweep that is required. The resonances are generally characterized by sharp and narrow peaks in the frequency response. In order to detect them a very fine frequency sweep must be carried out in the frequency domain, requiring hundreds of frequency point analyses.

However, resonant frequencies of a circuit are determined by actual system poles that are close to the imaginary axis. This can be seen by inspecting (2.14) (reproduced here):

$$H(s) = c + \sum_{j=1}^{N_p} \frac{k_j}{s - p_j} \quad (8.15)$$

If a pole of the network, $p_i = \alpha + j\beta$, is close to the imaginary axis, its real part, α will be quite small. Therefore at a frequency $s = j\beta$, one of the terms in the summation of (2.14) will become extremely large, signifying a highlight of that frequency. The closer the pole is to the axis, the faster the resonance will die down as the frequency is shifted from $j\beta$.

Typical PEEC networks contain many poles all along the imaginary axis giving several resonant frequencies. Since CFH guarantees the accuracy of all of the poles close to the imaginary axis, at the expense of a few frequency point analyses, all resonant frequencies can be determined without the large expense of performing a fine frequency scan, resonant frequencies. This is illustrated in the examples below. It is important to note that the method may be applied to searching a brief frequency band for resonant frequencies only in that band.

In Table 6.2 poles 4, 6-9, 11 and 14 cause sharp peaked resonances (Fig. 6.18) in the PEEC circuit due to their proximity to the imaginary axis. The resonant frequencies (in rad/sec) are determined by the imaginary parts (0.51, 1.25, 1.68, 1.87, 2.07, 2.76, 3.54 GHz).

Example

The same physical structure as in the PEEC example of Chapter 7 was modeled with 68 capacitive nodes, 112 inductances, 2278 capacitive and 2304 inductive couplings, *including retardation* between nodes. I applied CFH to the source current of the rPEEC model. The circuit was analyzed up to 5 GHz using 9 frequency hops yielding 42 poles/residues. The resultant CFH response is compared to the actual response generated using 1024 frequency point analyses produced with the fast rPEEC solver[Hee93] in Fig. 8.2. The transient response was calculated for a pulse of rise time/fall time of 0.35 ns, and pulse width of 1.05 ns. Since ASTAP based time domain rPEEC was not able to solve the problem the rPEEC results were calculated by applying FFT to 1024 frequency domain points. The results are given in Fig. 8.3.

Figure 8.2: CFH frequency analysis of a large rPEEC network.

Figure 8.3: CFH transient analysis of a large rPEEC network.

Chapter 9

Conclusion

As we have seen in this thesis, moment-matching techniques, such as Asymptotic Waveform Evaluation, have a wide range of application in linear electrical system analysis. They range from RLC lumped element model simulation in the time and frequency domain, to lossy, coupled transmission line analysis including nonuniform lines, to 3D full-wave electromagnetic simulation. However, several accuracy concerns of present moment-matching techniques needed to be addressed. In this thesis, two major numerical techniques were presented in order to address these concerns. First, complex frequency hopping, a multipoint moment-matching technique, guarantees the accuracy of an approximation over a pre-specified frequency range. This is in comparison to previously published single Padé approximation expansions where the accuracy range of an approximation was uncertain. Second, matrix exponential moment generation allows the accurate generation of the Taylor series coefficients of a set of lossy, coupled transmission lines without requiring the eigenvalues of the line propagation matrices. This method allows the accurate incorporation of transmission lines in either CFH or AWE techniques.

Several other benefits directly result from the aforementioned methods. CFH allows the extraction of the eigenvalues of linear network in a prespecified region of the complex plane. It enables the calculation of the eigenvalues of a distributed

network for which there is no available algorithm. Using CFH, one can extract the exact resonant frequencies of a network quickly and accurately, without requiring hundreds of frequency point analyses. Finally, CFH also enables the quick solution of rPEEC networks both in the time and frequency domain. Networks that are too large to be solved by the delay buffer mechanism can be solved by CFH.

In addition to the algorithmic solution of the problems of moment-matching, this thesis has attempted to give an overview of the properties of Padé approximations that are critical to these approaches. Padé approximations were seen to have empirically verifiable properties that could be used as rules of thumb to characterize approximations.

Even with these promising features of the algorithm, some problems and areas of exploration remain. At the present we can only characterize the poles/residues extracted by the error tolerance that we set upon them. The ideal situation would be for research to come up with an exact characterization of the error of Padé approximations as a function of frequency or in the time domain.

Another approach may be to use only part of the reasoning behind Padé approximations to generate a different, and better approximation. That is, one could try to generate a different function (rather than a rational function) so that its Taylor series expansion matches the system Taylor series expansion to an n -th moment. Such an approach is currently being explored by researchers at AT&T Bell Labs[FF94] with early signs showing it successful. The approximating function used is a matrix function rather than a simple rational polynomial. The method allows a broader region of accuracy about an expansion point potentially extending out to infinity. If this approach proves useful in the long run then one potential area of research exploration is the combining of CFH and this new method into the most efficient method possible.

One problem of CFH affecting its accuracy is the generation of residues once the poles have been collected. The present method uses convergence of the residues at the same time poles are converged to. This, can at times leave a DC offset error, as

all the poles and residues together might not match the DC solution. Other methods that can be explored is recalculating the residues once the poles have been collected matching them to an appropriate subset of moments from each of the hops. This could be done either with a matching set of equations or a least squares optimization approach.

Another problem using the method is that it currently relies on heuristic algorithms for finding the L and M of a particular Padé approximation, even with convergence. More research needs to be done as to how to select these variable more algorithmically.

Potentially, it is possible to speed up CFH even more. For circuit equations this is probably not necessary as the method works in terms of seconds or minutes, but when the method is applied to possibly much larger sets of equations this could be an advantage. Since each hop is essentially independent of each other hop after reordering, the method is well-given to parallelization. The approach that one could take would be to assign one expansion point per compute node and collect all of the information together. This is somewhat of a loss, since the matrix ordering information cannot be passed around to each compute node easily.

An area of exploration is applying CFH to extremely large systems of matrices such as those that are generated by full-wave methods such as the Finite Element Method (FEM) or the Finite Difference Time Domain (FDTD) method. These matrices can run into the sizes of hundreds of thousands or even millions. Using LU decomposition is not practical on these matrices since the reordering time would be prohibitive. Instead, methods based on Conjugate Gradient relaxation need to be used. The tradeoff would be that the cost of each moment would be the same as any other moment, at any particular hop. However, once poles and residues are obtained from these moments it would mean that the matrix had been effectively inverted. This is significant in that it is not popular to invert these matrices in typical numerical calculation due to the enormous cost. As a result, FDTD and FEM are typical solved

using forward integration methods which have problems with stability, accuracy and small step size.

CFH as applied to these problems would be excellent for parallel processing. This is because the solution cost is the same for each hop. In advance, the potential expansion points would be known on the frequency axis. If one knew the number of workstations, then one could plan a strategy. For instance if there are 4 workstations, 3 hops would be initially calculated: 0, f_{max} and $f_{max}/2$. If more information is needed, one would calculate mid points: $f_{max}/4$ and $3f_{max}/4$. Then, more midpoints would be used: $f_{max}/8$, $3f_{max}/8$, $5f_{max}/8$, and $7f_{max}/8$. At this point, 4 work stations would have been used and the time cost of 3 hops would have been consumed to produce the information for 9.

A further extension of CFH is possible in the area of transmission lines. For example, frequency-dependent transmission lines are difficult to simulate using standard techniques. However, as long as there is a way to extract the Taylor series coefficients from such line equations then CFH can be used to simulate them together with other linear/nonlinear components. Using polynomial functions has shown that such a method is potentially useful.

As well, measured networks can be incorporated in a moment-matching simulation if the measured network is a small part of a larger element-characterized one. This can be done without inverting the entire network at several frequency points. Preliminary work has shown this approach to be successful[SCNZ94].

Finally, at present, AWE is being applied to thermal analysis using equivalent RC-type models. However, it has been shown that thermal networks may have non-monotonic response implying higher frequency components. This may necessitate the need for CFH on these networks as well.

Appendix A

Transmission Line Equations

A.1 Partial differential equation description

The Telegrapher's equations describe a transmission line (Fig. A.1). They are generated by first assuming an even distribution of per-unit-length parameters of resistance, inductance, conductance and capacitance: R , L , G , and C . The line is discretized into infinitesimal sections of length Δx .

Figure A.1: Single transmission line.

Each section then includes a resistance $R\Delta x$, capacitance $C\Delta x$, conductance $G\Delta x$ and inductance $L\Delta x$ (see Fig. A.2). Using Kirchoff relationships yields the relation-

Figure A.2: Lumped equivalent of single transmission line.

ship,

$$v(x + \Delta x, t) = v(x, t) - i(x, t)R\Delta x - \frac{\partial}{\partial t}i(x, t)L\Delta x \quad (\text{A.1})$$

or,

$$\frac{v(x + \Delta x, t) - v(x, t)}{\Delta x} = -i(x, t)R - \frac{\partial}{\partial t}i(x, t)L \quad (\text{A.2})$$

Taking $\lim_{\Delta x \rightarrow 0}$ gives,

$$\frac{\partial v(x, t)}{\partial x} = -i(x, t)R - \frac{\partial}{\partial t}i(x, t)L \quad (\text{A.3})$$

Similarly for $i(x, t)$.

$$\frac{\partial i(x, t)}{\partial x} = -v(x, t)G - \frac{\partial}{\partial t}v(x, t)C \quad (\text{A.4})$$

In the case of a lossless line, the equation becomes,

$$\frac{\partial v(x, t)}{\partial x} = -\frac{\partial}{\partial t}i(x, t)L \quad (\text{A.5})$$

$$\frac{\partial i(x, t)}{\partial x} = -\frac{\partial}{\partial t}v(x, t)C \quad (\text{A.6})$$

The lossy equations have no closed form solution, while in the lossless case, one can solve for v by substitution.

$$\frac{\partial^2 v(x, t)}{\partial x^2} = LC \frac{\partial^2 v(x, t)}{\partial t^2} \quad (\text{A.7})$$

A similarly relation holds for i . This is the known wave equation with wave propagation speed v equal to $\sqrt{LC}$.

The equations for one line apply analogously for multiple coupled lines (Fig. 3.1), except that in that case, the per-unit-length parameters become matrices and the voltages and current become vectors of the multiple voltages/currents on the coupled lines. These equations are now formulated as in (3.12) and (3.13). In order to solve the lossy case, we take a Laplace transform as in (3.15) and (3.16) (reproduced here),

$$\frac{\partial}{\partial x} \mathbf{V}(x, s) = -(\mathbf{R} + s\mathbf{L})\mathbf{I}(x, s) \quad (\text{A.8})$$

$$\frac{\partial}{\partial x} \mathbf{I}(x, s) = -(\mathbf{G} + s\mathbf{C})\mathbf{V}(x, s) \quad (\text{A.9})$$

or,

$$\frac{\partial}{\partial x} \mathbf{V}(s) = -\mathbf{Z}_p \mathbf{I}(s) \quad (\text{A.10})$$

$$\frac{\partial}{\partial x} \mathbf{I}(s) = -\mathbf{Y}_p \mathbf{V}(s) \quad (\text{A.11})$$

where, $\mathbf{Z}_p = \mathbf{R} + s\mathbf{L}$ and $\mathbf{Y}_p = \mathbf{G} + s\mathbf{C}$. The two differential equations can be combined into a set of wave equations in the frequency domain,

$$\frac{\partial^2}{\partial x^2} \mathbf{V}(x, s) = \mathbf{Z}_p \mathbf{Y}_p \mathbf{V}(x, s) \quad (\text{A.12})$$

$$\frac{\partial^2}{\partial x^2} \mathbf{I}(x, s) = \mathbf{Y}_p \mathbf{Z}_p \mathbf{I}(x, s) \quad (\text{A.13})$$

which will have a solution of the form,

$$\mathbf{V}_m(x, s) = \mathbf{V}_m(0, s)e^{\pm \gamma_m(s)x} \quad (\text{A.14})$$

$$\mathbf{I}_m(x, s) = \mathbf{I}_m(0, s)e^{\pm \gamma_m(s)x} \quad (\text{A.15})$$

where γ_m is the complex propagation mode constant.

Substituting the solution form into the wave equation yields,

$$(\gamma_m^2 \mathbf{1} - \mathbf{Z}_p \mathbf{Y}_p) \mathbf{V}_m(0, s) = \mathbf{0} \quad (\text{A.16})$$

$\mathbf{V}_m(0, s)$ will have nontrivial solutions if γ_m^2 satisfies the eigenvalue problem

$$\det(\gamma_m^2 \mathbf{1} - \mathbf{Z}_p \mathbf{Y}_p) = 0 \quad (\text{A.17})$$

For inhomogeneous dielectrics, there exist in general n distinct eigenvalues $\gamma_m^2, m = 1, \dots, N$. Each eigenvalue has corresponding eigenvector $\underline{\mathbf{S}}_m$.

References

- [ABL⁺92a] M. M. Alaybeyi, J. E. Bracken, J. Y. Lee, V. Raghavan, R. J. Trihy, and R. A. Rohrer. Analysis of MCM's using asymptotic waveform evaluation (AWE). In *Proc. IEEE Multi-Chip Module Conference*, pages 48–51, March 1992.
- [ABL⁺92b] M. M. Alaybeyi, J. E. Bracken, J. Y. Lee, V. Raghavan, R. J. Trihy, and R. A. Rohrer. Exploiting partitioning in asymptotic waveform evaluation (AWE). In *Proc. IEEE Custom Integrated Circuit Conference*, pages 15.7.1–15.7.4, May 1992.
- [AGKP92] D. F. Anastasakis, N. Gopal, S. Y. Kim, and L. T. Pillage. On the stability of moment-matching approximations in asymptotic waveform evaluation. In *Proc. ACM/IEEE Design Automation Conference*, pages 207–212, June 1992.
- [Ale84] F. J. Alexandro Jr. Stable partial Padé approximation for reduced-order transfer functions. *IEEE Trans. on Automatic Control*, AC-29(2):159–162, February 1984.
- [ALR92] M. M. Alaybeyi, J. Y. Lee, and R. A. Rohrer. Numerical integration algorithms and asymptotic waveform evaluation (awe). In *Proc. IEEE International Conference on Computer Aided Design (ICCAD)*, pages 76–79, November 1992.

- [Bak75] G. A. Baker Jr. *Essential of Padé Approximants*. New York: Academic, 1975.
- [Bak90] H. B. Bakoglu. *Circuits, Interconnections and Packaging for VLSI*. Addison-Wesley, 1990.
- [BGM81] G. A. Baker Jr. and P. Graves-Morris. Padé approximants. In G. C. Rota, editor, *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, 1981. Vol 13 & 14.
- [BL72] M. J. Bosley and F. P. Lees. A survey of simple transfer-function derivations from high-order state-variable models. *Automatica*, 8:765-775, 1972.
- [Bra67] F. H. Branin Jr. Transient analysis of lossless transmission lines. *Proc. of the IEEE*, 55:2012-2013, 1967.
- [BRR92a] E. Bracken, V. Raghavan, and R.A. Rohrer. Simulating distributed elements with asymptotic waveform evaluation. In *Proc. IEEE MTT-S International Microwave Symposium*, pages 1337-1340. June 1992.
- [BRR92b] J. E. Bracken, V. Raghavan, and R. A. Rohrer. Interconnect simulation with asymptotic waveform evaluation (AWE). *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, CAS-39(11):869-878, November 1992.
- [CH87] C. Chu and M. Horowitz. Charge-sharing models for switch-level simulation. *IEEE Trans. Computer Aided Design*, pages 1053-1060, 1987.
- [Cha89] F. Y. Chang. The generalized method of characteristics for waveform relaxation analysis of lossy coupled transmission lines. *IEEE Trans. Microwave Theory and Tech.*, MTT-37:2028-2038, December 1989.

- [Cha91] F. Y. Chang. Waveform relaxation analysis of nonuniform lossy transmission lines characterized with frequency-dependent parameters. *IEEE Trans. on Circuits and Systems*, CAS-38:1484-1500, December 1991.
- [Cha92a] F. Y. Chang. Transient simulation of nonuniform coupled lossy transmission lines characterized with frequency-dependent parameters, Part I: Waveform relaxation analysis. *IEEE Trans. on Circuits and Systems*, 39:585-603, August 1992.
- [Cha92b] F. Y. Chang. Transient simulation of nonuniform coupled lossy transmission lines characterized with frequency-dependent parameters, Part II: discrete-time analysis. *IEEE Trans. on Circuits and Systems*, 39(11):907-927, November 1992.
- [CHNR93] E. Chiprout, H. Heeb, M. S. Nakhla, and A. E. Ruehli. Simulating 3-D retarded interconnect models using complex frequency hopping (CFH). In *IEEE International Conference on Computer Aided Design*, November 1993.
- [CK92] E. C. Chang and S. M. Kang. Computationally efficient simulation of a lossy transmission line with skin effect by using numerical inversion of Laplace transform. *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, CAS-39(11):861-868, November 1992.
- [CL75] L. O. Chua and P. Lin. *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [CN92a] E. Chiprout and M. S. Nakhla. Addressing high-speed interconnect issues in asymptotic waveform evaluation. In *Proc. ACM/IEEE Design Automation Conference*, pages 201-206, 1992.

- [CN92b] E. Chiprout and M. S. Nakhla. Fast nonlinear waveform estimation of large distributed networks. In *Proc. IEEE MTT-S International Microwave Symposium*, pages 1341–1344, 1992.
- [CN92c] E. Chiprout and M. S. Nakhla. Generalized moment matching methods for transient analysis of interconnect networks. In *Proc. ACM/IEEE Design Automation Conference*, pages 201–206, 1992.
- [CN92d] E. Chiprout and M. S. Nakhla. Optimal pole selection in asymptotic waveform evaluation. In *Proc. IEEE International Symposium on Circuits and Systems*, pages 1961–1964, June 1992.
- [CN93] E. Chiprout and M. S. Nakhla. Transient analysis of high-speed mcm networks using complex frequency hopping. In *Proc. IEEE Multi-Chip Module Conference*, pages 134–139, March 1993.
- [CPC'92] Bradly J. Cooke, John L. Prince, and Andreas C. Cangellaris. S-parameter analysis of multiconductor, integrated circuit interconnect systems. *IEEE Trans. Computer Aided Design*, 11(3):353–360, March 1992.
- [DSH86] A. R. Djordjević, T. K. Sarkar, and R. F. Harrington. Analysis of lossy transmission lines with arbitrary nonlinear terminal networks. *IEEE Trans. Microwave Theory and Tech.*, 34:660–666, June 1986.
- [Elm48] W. C. Elmore. The transient response of damped linear networks with particular regard to wideband amplifiers. *J. Appl. Phys.*, 19(1):55–63, 1948.
- [FF94] Peter Feldmann and Roland W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. In *European Design Automation Conference*, September 1994.

- [GCZN92] R. Griffith, E. Chiprout, Q. J. Zhang, and M. S. Nakhla. A CAD framework for simulation and optimization of high-speed VLSI interconnections. *IEEE Trans. on Circuits and Systems, CAS* 39(11):893-906, November 1992.
- [GN92] R. Griffith and M. S. Nakhla. Mixed frequency/time domain analysis of nonlinear circuits. *IEEE Trans. Computer Aided Design*, 10:1032-1043, August 1992.
- [GNP91] N. Gopal, D. P. Neikirk, and L. T. Pillage. Evaluating on-chip RC interconnect using moment-matching approximations. In *IEEE International Conference on Computer Aided Design*, pages 74-77, 1991.
- [Har68] R. F. Harrington. *Field Computation by Moment Methods*. Macmillan, 1968.
- [Hee91] H. Heeb, A. Ruehli. Retarded models for PCB board interconnects - or how the speed of light affects your SPICE circuit simulation. In *Proc. IEEE International Conference on Computer Aided Design (ICCAD)*, pages 70-73, November 1991.
- [Hee92] H. Heeb and A. E. Ruehli. Three-dimensional interconnect analysis using partial element equivalent circuits. *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, 39(11):974-982, November 1992.
- [Hee93] H. Heeb, S. Ponnappalli and A. E. Ruehli. Frequency domain microwave modeling using retarded partial element equivalent circuits. In *Proc. ACM/IEEE Design Automation Conference*, pages 702-706, June 1993.
- [HR92] H. Heeb and A. E. Ruehli. Three-dimensional interconnect analysis using partial element equivalent circuits. *IEEE Trans. on Circuits and Sys-*

3 of/de 3

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

PRECISIONSM RESOLUTION TARGET

tems I: Fundamental Theory and Applications, 39(11):974-982, November 1992.

- [HRB75] C. W. Ho, A. E. Ruehli, and P. A. Brennan. The modified nodal approach to network analysis. *IEEE Trans. on Circuits and Systems*, CAS-22:504-509, June 1975.
- [HRBR92] H. Heeb, A. E. Ruehli, J. E. Bracken, and R. A. Rohrer. Three dimensional circuit oriented electromagnetic modeling for VLSI interconnects. In *Proc. IEEE International Conference on Computer Design*, 1992.
- [HRR90] X. Huang, V. Raghavan, and R. A. Rohrer. AWESim: a program for the efficient analysis of linear(ized) circuits. In *Proc. IEEE International Conference on Computer-Aided Design (ICCAD)*, pages 534-537, November 1990.
- [Hua90] X. Huang. *Padé Approximation of Linear(ized) Circuit responses*. PhD thesis, Carnegie Mellon University, November 1990.
- [JM80] W. B. Jones and A. Magnus. Computation of poles of two-point Padé approximants and their limits. *Journal of Computational and Applied Mathematics*, 6(2):105-119, 1980.
- [Jou83] N. P. Jouppi. TV: an nMOS timing analyzer. In *Proc. 3rd Caltech Conf. on VLSI*, pages 57-69, March 1983.
- [Jou87] N. P. Jouppi. Timing analysis and performance improvement of MOS VLSI designs. *IEEE Trans. Computer Aided Design*, pages 650-665, 1987.
- [KRS90] S. Kumashiro, R. A. Rohrer, and A. J. Strojwas. A new efficient method for the transient simulation of three-dimensional interconnect structures. In *Technical Digest of the 1990 International Electron Devices Meeting*, pages 193-196, December 1990.

- [Kum91] K. B. Kumar. Novel techniques to solve sets of coupled differential equations with SPICE. *IEEE Trans. on Circuits and Devices*, 8(1):11-14, January 1991.
- [Kun86] K. S. Kundert. Sparse matrix techniques. In A. E. Ruehli, editor, *Circuit Analysis, Simulation and Design*. North-Holland, 1986. Vol 3, pt. 1.
- [Kun87] K. S. Kundert. Sparse 1.3 user's guide. Technical report, Univ. Calif. Berkeley, 1987.
- [LK92] S. Lin and E. S. Kuh. Transient simulation of lossy interconnects based on the recursive convolution formulation. *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, CAS-39(11):879-892, November 1992.
- [LM84] T. M. Lin and C. A. Mead. Signal delay in general RC networks. *IEEE Trans. Computer Aided Design*, 3:331-349, 1984.
- [LZN92] R. L. Liu, Q. J. Zhang, and M. S. Nakhla. A frequency domain approach to performance optimization of high-speed VLSI interconnects. *IEEE Trans. Microwave Theory and Tech.*, 40, December 1992.
- [Mat69] R. Matick. *Transmission Lines for Digital and Communication Networks*. McGraw-Hill, Inc., 1969.
- [McC75] J. H. McCabe. A formal extension of the Padé table to include two point Padé quotients. *Journal Inst. Maths Applics*, 15:363-372, 1975.
- [McC89] Steven Paul McCormick. *Modeling and Simulation of VLSI Interconnections with Moments*. PhD thesis, Massachusetts Institute of Technology, June 1989.
- [ML78] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *Siam Review*, 20(4):801-836, October 1978.

- [MNZ92] S. Manney, M. S. Nakhla, and Q. J. Zhang. Time domain analysis of nonuniform frequency dependent high-speed interconnects. In *Proc. IEEE International Conference on Computer Aided Design (ICCAD)*, pages 449-453, 1992.
- [Nag75] L. W. Nagel. SPICE2, a computer program to simulate semiconductor circuits. Technical Report ERL M520, UC-Berkeley, May 1975.
- [Ous83] J. K. Ousterhout. CRYSTAL: a timing analyzer for NMOS VLSI circuits. In *Proc. 3rd Caltech Conf. on VLSI*, pages 57-69, March 1983.
- [Ous84] J. K. Ousterhout. Switch-level delay models for digital MOS VLSI. In *Proc. ACM/IEEE Design Automation Conference*, pages 542-548, 1984.
- [OW86] P. O'Brien and J. L. Wyatt. Signal delay in ECL interconnect. In *Proc. IEEE International Symposium on Circuits and Systems*, May 1986.
- [Pau92] C. R. Paul. *Introduction to Electromagnetic Compatibility*. Wiley, 1992.
- [PHR91] L. T. Pillage, X. Huang, and R. A. Rohrer. asymptotic waveform evaluation for circuits containing floating nodes. In *Proc. IEEE Custom Integrated Circuit Conference*, pages 8.4.1-8.4.4, May 1991.
- [Pil89] L. T. Pillage. *Asymptotic Waveform Evaluation for timing analysis*. PhD thesis, Carnegie-Mellon University, 1989.
- [PL89] O. A. Palusinsky and A. Lee. Analysis of transients in nonuniform and uniform multiconductor transmission lines. *IEEE Trans. Microwave Theory and Tech.*, 37:127-138, January 1989.
- [PR90] L. T. Pillage and R. A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Trans. Computer Aided Design*, 9:352-366, April 1990.

- [RGP91] C. L. Ratzlaff, N. Gopal, and L. T. Pillage. RICE: rapid interconnect circuit evaluator. In *Proc. ACM/IEEE Design Automation Conference*, pages 555–560, June 1991.
- [Roh88] R. A. Rohrer. Circuit partitioning simplified. *IEEE Trans. on Circuits and Systems*, CAS-35(1):2–5, 1988.
- [RP84] J. Rubenstein and P. Penfield. Signal delay in RC tree networks. *IEEE Trans. Computer Aided Design*, 2:331–349, 1984.
- [Rue74] A. E. Ruehli. Equivalent circuit models for three dimensional multi-conductor systems. *IEEE Trans. Microwave Theory and Tech.*, MTT-22(3):216–221, March 1974.
- [Rue92] A. E. Ruehli and H. Heeb. Circuit models for three-dimensional geometries including dielectrics. *IEEE Trans. Microwave Theory and Tech.*, 40(7):1507–1516, July 1992.
- [SAM89] J. E. Schutt-Aine and R. Mitra. Nonlinear transient analysis of coupled transmission lines. *IEEE Trans. on Circuits and Systems*, CAS-36:959–967, July 1989.
- [SCNZ91] R. Sanaie, E. Chiprout, M. S. Nakhla, and Q. J. Zhang. Integrating subnetworks characterized by measured data into moment-matching simulations. In *Proc. IEEE Multi-Chip Module Conference*, pages 114–119, March 1991.
- [Tij87] A. G. Tijhuis. *Electromagnetic Inverse Profiling*, chapter 3, pages 189–269. VNU Science Press BV, Utrecht, The Netherlands, 1987.
- [TN90] T. Tang and M. S. Nakhla. Analysis of high-speed VLSI interconnects. In *Proc. IEEE International Conference on Computer-Aided Design (ICCAD)*, pages 341–352, November 1990.

- [TN92] T. Tang and M. S. Nakhla. Analysis of high-speed VLSI interconnect using the asymptotic waveform evaluation technique. *IEEE Trans. Computer Aided Design*, 11, March 1992.
- [VS83] J. Vlach and K. Singhal. *Computer Methods for Circuit Analysis and Design*. Van Nostrand Reinhold, 1983.
- [Wat82] David S. Watkins. Understanding the QR algorithm. *Siam Review*, 24(4):427-440, October 1982.
- [Wee73] W. T. Weeks, A. J. Jimenez, G. W. Mahoney, D. Mehta, H. Quasenzadeh and T. R. Scott. Algorithms for ASTAP - a network analysis program. *IEEE Trans. on Circ. Th.*, CT-20(11):628-634, November 1973.
- [Wol91] S. Wolfram. *Mathematica: A System for Doing Mathematics by Computer*. Addison-Wesley, 1991.
- [WW92] R. Wang and O. Wing. Transient analysis of dispersive VLSI interconnects terminated in nonlinear loads. *IEEE Trans. Computer Aided Design*, 11:1258-1277, October 1992.
- [Wya87] J. L. Wyatt. Signal delay in RC models for interconnect. In *Circuit Analysis, Simulation and Design*. North-Holland, Amsterdam, The Netherlands, 1987.
- [XN91] D. Xie and M. S. Nakhla. Delay and crosstalk simulation of high speed VLSI interconnects with nonlinear terminations. In *Proc. IEEE International Conference on Computer Aided Design (ICCAD)*, pages 66-69, November 1991.
- [XN93] D. Xie and M. S. Nakhla. Delay and crosstalk simulation of high speed VLSI interconnects with nonlinear terminations. *IEEE Trans. Computer Aided Design*, 12(11):1198-1811, November 1993.

END

2 5-0 5-9 5

FIN