

Abstract

CHRISTOFFERSEN, CARLOS ENRIQUE. Global Modeling of Nonlinear Microwave Circuits (Under the direction of Michael B. Steer.)

A global modeling concept for modeling microwave circuits is described. This concept allows the modeling of electromagnetic (EM) and thermal effects to be included in the simulation of electronic circuits, by viewing EM and thermal subsystems as subcircuits. Then, circuit analysis techniques are developed from a general state variable reduction formulation. This general formulation, based on the state variables of the nonlinear devices, allows the analysis of large microwave circuits because it reduces the size of the nonlinear system of equations to be solved. One of the derived analysis techniques is based on convolution and therefore provides modeling of frequency-defined network elements not present in conventional circuit simulators. Another analysis technique based on wavelets that would enable the multiresolution analysis of circuits is investigated. Also, a reduced state variable formulation using conventional time marching schemes is developed. It is shown that this can achieve more than an order of magnitude improvement in simulation speed compared to that of traditional circuit simulation methods. All these developments are implemented in a circuit simulator program, called Transim. This program provides unprecedented flexibility for the addition of new device models or circuit analysis algorithms. Transim supports the local reference concept, which is fundamental to the analysis of spatially distributed circuits and also to simultaneous thermal-electrical simulations. Transim is applied to the transient simulation of a 47-section nonlinear transmission line considering frequency dependent attenuation for the first time and the transient simulation, also for the first time, of two quasi-optical power amplifier arrays.

GLOBAL MODELING OF NONLINEAR MICROWAVE CIRCUITS

by

CARLOS ENRIQUE CHRISTOFFERSEN

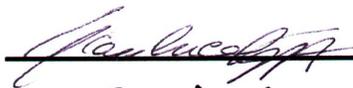
A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

ELECTRICAL ENGINEERING

Raleigh

2000

APPROVED BY:



Paul W. Lee

Griff Billro





Chair of Advisory Committee

Biographical Summary

Carlos E. Christoffersen was born in Santa Fe, Argentina in 1968. From 1991 to June 1996 he was a member of the research staff of the Laboratory of Microelectronics of the National University of Rosario, Argentina. He received the Electronic Engineer degree at the National University of Rosario, Argentina in 1993. From 1993 to 1995 he was a research fellow of the National Research Council of Argentina (CONICET). In 1996, he was awarded a Fulbright scholarship to pursue an M.S. degree in the USA. He received the M.S. degree in electrical engineering in 1998 from North Carolina State University. Currently he is pursuing his Ph.D. degree at the same university. Since 1996, he has been with the Department of Electrical and Computer Engineering as a research assistant. He is a member of the IEEE. His research interests include computer aided analysis of circuits and analog, RF and microwave circuit design.

Acknowledgments

I would like to express my gratitude to my advisor Dr. Michael Steer for his support and guidance during my graduate studies. It was a privilege to be part of his quasi-optical research group. I would also like to express my sincere appreciation to Dr. Amir Mortazawi, Dr. Griff Bilbro and Dr. Ilse Ipsen for showing interest in my research and serving on my Ph.D. committee. Also thanks to Dr. James Harvey, Dr. Gianluca Lazzi and Dr. James Mink.

A very big thanks go to my past and present graduate student and post-doctoral colleagues. Special thanks to Dr. Hector Gutierrez, Dr. Alexander Yakovlev, Mr. Mete Ozkar and Dr. William Batty. Finally, I wish to thank my wife and my parents for their support and encouragement.

Contents

List of Figures	vii
List of Tables	xi
List of Symbols	xii
List of Abbreviations	xv
1 Introduction	1
1.1 Motivations and Objectives of This Study	1
1.2 Review of Circuit Simulation Technology	2
1.2.1 Object-Oriented Circuit Simulators	5
1.3 Original Contributions	6
1.4 The Global Modeling Concept	8
1.5 Thesis Overview	9
1.6 Publications	10
1.6.1 Journals	10
1.6.2 Conferences	10
2 Literature Review	12
2.1 Transient Analysis of Microwave Circuits	12
2.1.1 Impulse Response and Convolution	13
2.1.2 Numerical Inversion of Laplace Transform Technique	13
2.1.3 Approximations using Pole-Zero Models	14
2.2 Multiresolution Analysis of Circuits	16
2.2.1 Adaptive Circuit Simulation Methods	16
2.2.2 Wavelets Applied to Solve Differential Equations	17
2.2.3 Spline Pseudo-Wavelet Collocation Method	18

2.3	Numerical Integration Using Time Marching Methods	28
2.3.1	Backward Euler Formula	29
2.3.2	Trapezoidal Rule	29
2.4	Parameterized Nonlinear Models	29
2.5	Multitone Frequency-Domain Simulation of Nonlinear Circuits	30
2.6	Summary	32
3	Generalized State Variable Reduction	33
3.1	Generalized Reduced Nonlinear Error Function Formulation .	33
3.1.1	Linear Network	34
3.1.2	Nonlinear Network	35
3.1.3	Error Function Formulation	35
3.2	Discussion	36
4	Convolution Transient Analysis	38
4.1	Introduction	38
4.2	Harmonic Balance Analysis	39
4.3	Formulation of the Convolution Transient	40
4.3.1	Initial Operating Point	41
4.4	Error Reduction Techniques	42
4.4.1	Impulse Response Determination	42
4.4.2	Correction of the DC Error	44
4.4.3	Convolution	45
4.5	Software Implementation	46
4.6	Nonlinear Transmission Line	47
4.7	Simulation Results and Discussion	48
5	Wavelet Transient Analysis	53
5.1	Introduction	53
5.2	Reduced Error Function Formulation	54
5.2.1	System Expansion	54
5.2.2	Initial Conditions in the State Variables	58
5.2.3	Solution of the Nonlinear System Considerations	58
5.2.4	Other Transformation Types	59
5.2.5	Resolution Limits	59
5.3	Implementation	60
5.4	Simulation Results and Discussion	63
5.5	Summary	65

6	Time Marching Transient Analysis	66
6.1	Introduction	66
6.2	Reduced Error Function Formulation	67
6.3	Implementation	68
6.4	Simulation Results and Discussion	70
7	Object Oriented Circuit Simulator	72
7.1	Introduction	72
7.2	The Network Package	74
7.3	The Analysis Classes	79
7.4	Nonlinear Elements	81
7.5	Example: Use of Polymorphism	84
7.6	Support Libraries	86
7.6.1	Solution of Sparse Linear Systems	86
7.6.2	Vectors and matrices	86
7.6.3	Solution of nonlinear systems	87
7.6.4	Fourier transform	88
7.6.5	Automatic differentiation	88
7.7	Summary	90
8	Modeling of Quasi-Optical Systems	92
8.1	Introduction	92
8.2	Implementation of Local Reference Nodes	93
8.2.1	Background	93
8.2.2	Handling of Local Reference Nodes	94
8.2.3	Implementation in Transim	97
8.3	Integration of Electromagnetic Structures	100
8.3.1	Frequency Domain and Time Domain Using Convolution	100
8.3.2	Time Domain using Pole-Zero Approximations	101
8.3.3	CPW Folded-Slot Active Antenna Array	104
8.3.4	2x2 Grid Amplifier	111
8.4	Electro-Thermal Modeling	120
8.4.1	Thermal Impedance Matrix	120
8.4.2	Implementation of Thermal Circuits and Elements in Transim	121
8.4.3	Electrothermal Modeling of a Quasi-Optical System . .	122
8.4.4	Simulation Results and Discussion	124

9	Conclusions and Future Research	129
9.1	Summary of Research and Original Contributions	129
9.2	Future Research	131
9.2.1	The Sliding Window Scheme	132
A	Transim's Graphical User Interface	134
A.1	Introduction	134
A.2	The Netlist Editor	134
A.3	The Analysis Window	135
A.4	The Output Viewer Window	135
B	Object Oriented Programming Basics	138
B.1	UML diagrams	140
B.1.1	Collaboration Diagrams	141

List of Figures

1.1	Relation between v and i in a diode.	4
1.2	Relation between x and i in a diode.	4
1.3	Relation between x and v in a diode.	5
1.4	Partition of a system into spatially distributed and lumped linear circuit, nonlinear network, and thermal parts.	9
2.1	The scaling function $\varphi(t)$	19
2.2	The boundary scaling function $\varphi_b(t)$	19
2.3	The wavelet function $\psi(t)$	20
2.4	The boundary wavelet function $\psi_b(t)$	21
2.5	Boundary spline: $\eta_1(t)$	23
2.6	Boundary spline: $\eta_2(t)$	23
2.7	Boundary wavelet function: ψ_{b0}	24
2.8	Boundary wavelet function: ψ_{b1}	25
2.9	Partition representation of the time-invariant harmonic balance method	30
3.1	Network with nonlinear elements.	34
3.2	A quasi-optical grid.	37
4.1	Augmentation and compensation network	43
4.2	Flow diagram of the analysis code	46
4.3	Model of the nonlinear transmission line.	47
4.4	Complete transient response for the soliton line.	48
4.5	Comparison between experimental data and simulations.	49
4.6	Magnitude of the harmonics of the output power.	50
4.7	Magnitude of the harmonics of the output power along the line. Blue is low and red is high power.	51

4.8	Comparison between simulations using different models for the transmission lines.	52
5.1	Scaling functions for $L = 6$	54
5.2	Wavelet functions in W_0 for $L = 6$	55
5.3	Representation of the nonzero elements of the transformation matrices for $L = 10$ and $J = 2$: (a) \mathbf{W} and (b) \mathbf{W}'	56
5.4	Linear system construction.	56
5.5	Number of time samples in the interval as a function of J for a small circuit.	60
5.6	Reciprocal of the condition number of \mathbf{M}_J in the interval as a function of J for a small circuit.	61
5.7	Class diagram for the wavelet-transient-based analysis in Transim.	62
5.8	Comparison of the voltage of the diode close to the generator (diode 1).	64
5.9	Comparison of the voltage of the diode close to the load (diode 47) of the nonlinear transmission line.	64
5.10	Coefficients of the state variable of diode 47.	65
6.1	Integration method interface.	69
6.2	Comparison of the voltage of the last diode.	70
7.1	The network package is the core of the simulator.	75
7.2	The NetListItem class.	76
7.3	The Element class.	76
7.4	Documentation generated for an element.	77
7.5	The Circuit class.	78
7.6	The Xsubckt class.	79
7.7	The analysis classes.	80
7.8	Dependency inversion was used to make the elements independent of the analysis classes.	82
7.9	Class diagram for an element using generic evaluation.	83
7.10	Nonlinear element function evaluation in convolution transient.	85
7.11	Implementation of automatic differentiation.	89
8.1	Equivalent circuit of a two-port distributed element.	95
8.2	Generic spatially distributed circuit.	96

8.3	The locally referenced groups are isolated, therefore the solution is unchanged if the local references are connected.	96
8.4	Illegal connection between groups.	97
8.5	Circuit containing an illegal element. The numbers at the nodes indicate the order in which terminals are discovered. . .	98
8.6	Collaboration diagram showing the violation detection algorithm.	99
8.7	Sample frequency-domain y parameter file.	101
8.8	Comparison between the original and the approximated y_{11} for the 2x2 grid.	102
8.9	Sample time-domain y parameter file.	103
8.10	Unit cell of the CPW antenna array.	105
8.11	Circuit model of the unit cell. The diamond symbol indicates a local reference node.	106
8.12	Effective isotropic power gain as a function of frequency. . . .	106
8.13	2x2 CPW antenna array.	107
8.14	2x2 CPW antenna array equivalent circuit.	108
8.15	Netlist for a 2x2 CPW antenna array.	109
8.16	Output currents for the 2x2 antenna array.	110
8.17	Setup for measurement characterization of the quasi-optical amplifier system.	111
8.18	Layout for 2x2 quasi-optical grid amplifier.	112
8.19	Full spectrum for output voltage.	113
8.20	Spectral regrowth in one of the amplifiers.	114
8.21	Transient response at one of the MMIC output ports using a large time step.	115
8.22	Zoom of the transient response at one of the MMIC output ports using a small time step.	116
8.23	Output transient starting with the circuit already biased. . . .	117
8.24	Comparison of the steady-state output waveforms.	118
8.25	Transient response of one amplifier with RF excitation.	119
8.26	Amplifier circuit used.	122
8.27	Thermal circuit model for the 2x2 CPW array.	123
8.28	Comparison of the steady-state output voltage with and without thermal effects.	124
8.29	Steady-state temperature at the MESFET and the substrate. . .	125
8.30	Detail of the MESFET temperature.	125
8.31	Effect of the temperature on the ACPR of the output voltage. .	126

8.32	Transient of the MESFET temperature. The temperature shown is the difference with respect to the ambient temperature (300 K).	127
8.33	Transient of the substrate temperature. The temperature shown is the difference with respect to the ambient temperature (300 K).	128
9.1	The sliding window scheme.	132
A.1	Netlist Editor window.	135
A.2	Analysis window.	136
A.3	Output Viewer window.	136
A.4	Plot window.	137
B.1	Notation for a class diagram.	141

List of Tables

6.1	Comparison of simulation times for the soliton line	71
8.1	Comparison of the simulation times for the 2 by 2 grid array .	118

List of Symbols

$\Re()$	– Real part.
ω	– Angular frequency.
t	– Time.
$H(s)$	– Transfer function in the Laplace domain.
$H^2(I)$	– Sobolev space.
$\varphi(t)$	– Scaling function.
$\varphi_b(t)$	– Boundary scaling function.
Z	– The set of integer numbers.
J	– Wavelet subspace level.
$\psi(t)$	– Wavelet function.
$\psi_b(t)$	– Boundary wavelet function.
$\psi_{b0}(t)$	– Boundary wavelet function.
$\psi_{b1}(t)$	– Boundary wavelet function.
$\eta_1(t)$	– Boundary spline function.
$\eta_2(t)$	– Boundary spline function.
$\hat{\mathbf{f}}()$	– Wavelet coefficient vector of function sample vector \mathbf{f}
$\mathbf{x}(t)$	– Nonlinear element state variable vector.
$\mathbf{v}_{\text{NL}}(t)$	– Nonlinear element port voltage vector.
$\mathbf{i}_{\text{NL}}(t)$	– Nonlinear element port current vector.
n_t	– Number of tones.
C	– Nonlinearity order.
\mathbf{G}	– Frequency-independent component of the MNAM.
\mathbf{C}	– Frequency-dependent component of the MNAM.
\mathbf{u}	– Vector of nodal unknowns and selected variables.
\mathbf{s}	– Source vector in MNAM system.
\mathbf{T}	– Incidence matrix.
n_s	– Number of state variables.
n_m	– Size of the MNAM.

- Ω – Frequency domain derivation operator.
- \mathbf{S}_{SV} – Frequency-domain compressed source vector.
- \mathbf{M}_{SV} – Frequency domain compressed impedance matrix.
- \mathbf{m}_{SV} – Compressed impulse response impedance matrix.
- \mathbf{W}_J – Wavelet transformation matrix up to subspace level J .
- \mathbf{W}'_J – Wavelet derivative transformation matrix up to subspace level J .
- \mathbf{x}_J – State variable vector at all collocation points.
- $\hat{\mathbf{x}}_J$ – State variable wavelet coefficient vector.
- \mathbf{M}_J – Wavelet-expanded MNAM.
- $\mathbf{s}_{f,J}$ – Wavelet-expanded source vector.
- $\mathbf{M}_{sv,J}$ – Wavelet-expanded compressed impedance matrix.
- $\mathbf{s}_{sv,J}$ – Wavelet-expanded compressed source vector.
- \mathbf{M}_{sv} – Time marching compressed impedance matrix.
- $\mathbf{s}_{sv,n}$ – Time marching compressed source vector at time step n .

List of Abbreviations

- ACPR – Adjacent Channel Power Regrowth.
- AWE – Asymptotic Waveform Evaluation.
- BE – Backward Euler.
- CAE – Computer Aided Engineering.
- CPW – Coplanar Waveguide.
- DFT – Direct Fourier Transform.
- EM – Electro Magnetic.
- FDTD – Finite Difference Time Domain.
- FFT – Fast Fourier Transform.
- HB – Harmonic Balance.
- I-FFT – Inverse Fast Fourier Transform.
- IRC – Impulse Response and Convolution.
- KCL – Kirchoff Current Law.
- LRG – Local Reference Group.
- LRN – Local Reference Node.
- MMIC – Monolithic Microwave Integrated Circuit.
- MNAM – Modified Nodal Admittance Matrix.
- MRA – Multiresolution Analysis.
- NILT – Numerical Inverse Laplace Transform.
- NLTL – Nonlinear Transmission Line.
- ODE – Ordinary Differential Equation.
- OO – Object Oriented.
- PDE – Partial Differential Equation.
- QO – Quasi-Optical.
- RF – Radio Frequency.
- RLCG – Resistor, Inductor, Capacitor, Conductor.
- STL – Standard Template Library.
- UML – Unified Modeling Language.

Chapter 1

Introduction

1.1 Motivations and Objectives of This Study

The almost overwhelming use of the circuit abstraction in electronic engineering has enabled the design of surprisingly complex systems [1]. There are three important reasons for simulating electronic circuits: to understand the physics of a complex system of interacting elements; to test new concepts; and to optimize designs. Millimeter-wave circuits are becoming ever more commercially and militarily viable and, coupled with large scale production, design practices must evolve to be more sophisticated in order to handle more complex relationships between constituent elements. Also, as the frequency of RF circuits extends beyond a gigahertz to tens and hundreds of gigahertz, device and circuit dimensions become large with respect to wavelength and the three dimensional electro magnetic (EM) environment becomes more significant. If reliable, high yielding, optimized designs of microwave and millimeter-wave circuits are to be achieved, the interrelated effects of the EM field, the linear and nonlinear circuit elements and the thermal subsystem must be self-consistently modeled.

This work attempts to develop a unified circuit view for integrating linear and nonlinear devices with electromagnetic and thermal effects. The basic idea is to convert the EM and thermal problems into circuit abstraction. Another objective of this study is to develop efficient circuit analysis techniques that can be used to simulate the whole system.

In this thesis, a general state variable reduction error function formulation is presented. This formulation is numerically efficient and allows a

very flexible and convenient software implementation in a circuit simulator. Several circuit transient analysis methods are derived from this formulation; based on convolution, wavelets and time-marching schemes. It is also shown that the harmonic balance analysis developed in [2] is a particular case of the general state variable error function formulation. The state-variable-based convolution analysis was used here to perform a transient simulation of a 47-section nonlinear transmission line (NLTL) with frequency-dependent attenuation for the first time. The combination of the state variable reduction developed here with time marching schemes achieves more than an order of magnitude improvement in simulation speed comparing with traditional circuit simulation methods.

All these developments are implemented in a circuit simulator program, called *Transim*. The novel techniques used in the design of this program are described. Two quasi-optical power amplifier arrays are modeled and simulated. The results for electrical-only and simultaneous thermal-electrical simulations are presented and commented.

1.2 Review of Circuit Simulation Technology

The most widespread [3] method of nonlinear circuit analysis is time-domain analysis (also called transient analysis) using programs like Spice. Such programs use numerical integration to determine the circuit response at one instance of time given the circuit's response at a previous instance of time. The success of Spice is that the associated modeling approach works extremely well. In Spice the circuit model is discretized in time and merged with a Newton iteration step to form what is called an associated discrete model. This associated discrete model is linear and so the mathematical formulation and solution is linear. If an error, such as a Kirchoff's current law error, is detected the associated model is updated and solved again. It is not obvious that this is a better way of solving the circuit problem than solving the coupled nonlinear integro-differential problem directly as would be required without introducing the associated model. However it is extremely robust and numerically efficient.

However, it is not always practical to simulate analog circuits with sinusoidal excitation using numerical integration. Application of nonlinear analysis using analysis domains other than the time domain is a current area of research. There are many other ways of thinking about circuit analysis.

Some are ideally suited to particular applications. An example is *harmonic balance* (HB) analysis of RF and microwave circuits. Harmonic balance differs [4] from traditional transient analysis in two fundamental ways. These differences allow harmonic balance to compute periodic and quasi-periodic solutions directly and in certain circumstances give the method significant advantages in terms of accuracy and efficiency. The first difference between harmonic balance and transient analysis is that harmonic balance uses a linear combination of sinusoids to build the solution. Thus, it approximates naturally the periodic and quasi-periodic signals found in a steady-state response. Harmonic balance also differs from traditional time domain methods in that time domain simulators represent waveforms as a collection of samples whereas harmonic balance represents them using the coefficients of the sinusoids.

Rizzoli [5] proposed a state variable approach in the HB simulation context (described in Section 2.4) which provides great flexibility for the design of nonlinear device models. The state variables can be chosen to achieve robust numerical characteristics. As an example we present a parameterized diode model as described in [5]. The full model includes capacitances and nonlinear resistance, but here a simple case is shown for didactic purposes. The conventional current equation for the diode is

$$i(t) = I_s(\exp(\alpha v(t)) - 1)$$

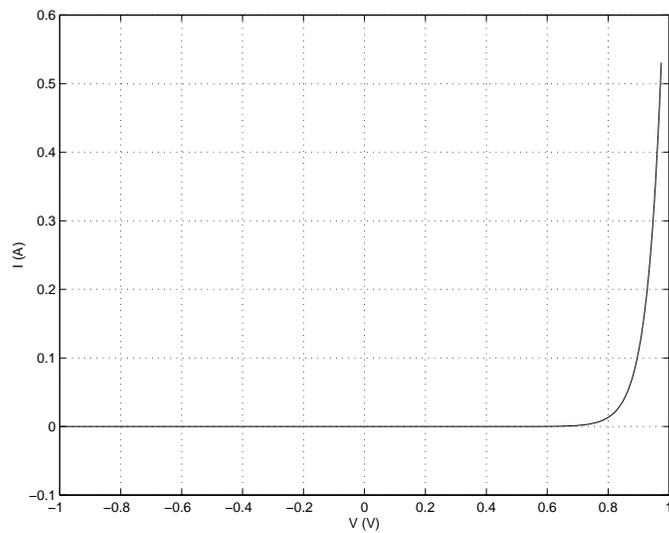
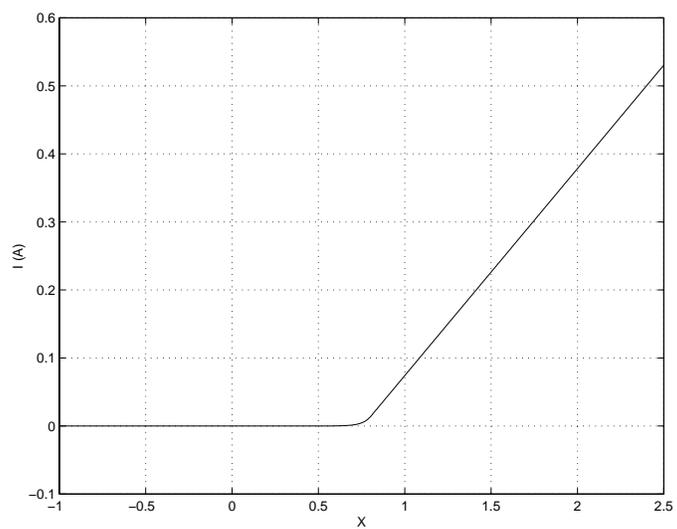
and, based on previous work [5], the parametric model is as follows

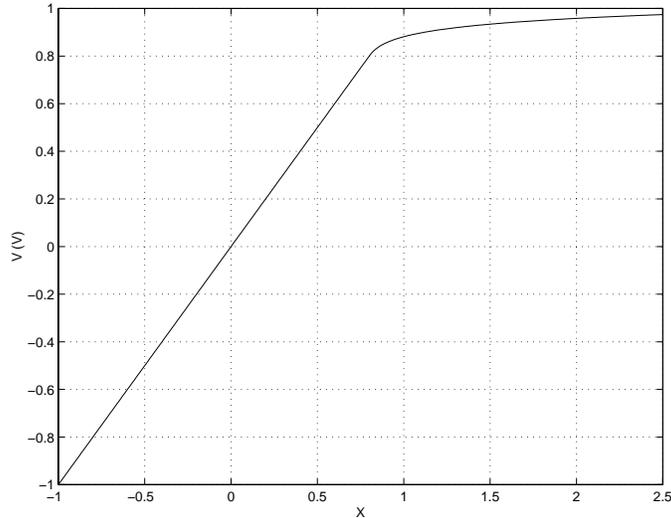
$$\begin{aligned} v(t) &= \begin{cases} x(t) & \text{if } x(t) \leq V_1 \\ V_1 + \frac{1}{\alpha} \ln(1 + \alpha(x(t) - V_1)) & \text{if } x(t) > V_1 \end{cases} \\ i(t) &= \begin{cases} I_s(\exp(\alpha x(t)) - 1) & \text{if } x(t) \leq V_1 \\ I_s \exp(\alpha V_1)(1 + \alpha(x(t) - V_1)) - I_s & \text{if } x(t) > V_1 \end{cases} \end{aligned}$$

where V_1 is some threshold value.

The plots of Figs. 1.1, 1.2 and 1.3 illustrate the improvement in the regular nonlinear behavior of the model when using the state variable approach.

In a diode the current has an exponential dependence on voltage (Fig. 1.1). This causes convergence problems when the voltage is updated during nonlinear iteration as in normal circuit simulation. At voltages greater than the threshold, small voltage increments can result in large current changes and hence large changes in the error function that must be solved to simulate the circuit. In a Spice circuit simulator the amount that the voltage can

Figure 1.1: Relation between v and i in a diode.Figure 1.2: Relation between x and i in a diode.

Figure 1.3: Relation between x and v in a diode.

change is generally limited so that there can be no more than 10% change in i or v . The possibility of large changes is eliminated through the use of parameterization which ensures smooth, well behaved current, voltage and error function variations when the state variable is updated. Thus the $i-x$ and $v-x$ functions are well behaved and thus circuit analysis via nonlinear iterations is also well behaved. See Figures 1.2 and 1.3.

One of the most significant developments relevant to microwave computer aided engineering is the rise of object oriented design practice. In the last years, some circuit simulators have been designed using *object oriented* (OO) techniques. A review of the key developments follows.

1.2.1 Object-Oriented Circuit Simulators

*APLAC*¹ [6, 7] is a significant achievement in the development of object-oriented circuit simulators with the object orientation implemented in the standard C language using macros. The most important feature of *APLAC* is that every circuit element is modeled internally using independent and voltage-controlled current sources. Since all models in *APLAC* are eventually mapped to current sources, the simple nodal linear DC analysis, $\mathbf{Gu} = \mathbf{j}$,

¹<http://www.aplac.hut.fi/aplac/main.html>

is all that is required to realize nonlinear DC, AC, transient and harmonic balance analyses. Here \mathbf{G} , \mathbf{u} and \mathbf{j} denote conductance matrix, nodal voltages and the independent source currents, respectively. The cost of this approach is reduced speed. In part this is because the C language is not optimal for OO applications but also because the high level of abstraction introduces overhead. However the objective of providing great functionality to enable experimentation with new element types and analysis techniques was achieved. The current version of the program incorporates many advanced features including electro-thermal analysis and is commercially available.

Other OO circuit simulators are *CODECS* [8], *ACS²* [9] and *Sframe* [10] and these adopted a common interface for all the circuit elements. In this way, all the code related to one element is separated from the rest of the program. In other words, the main program does not have dependencies on individual elements. The result is that the programming effort required to add new elements and algorithms is greatly reduced. ACS and Sframe are written in C++ and among other features, they both allow one element to be composed of other basic elements. The underlying algorithms in ACS are the same as those in Spice. As well as the flexibility introduced by the OO design there are memory savings in storing the circuit.

Sframe incorporates several novel features including automatic differentiation. In this simulator C++ is used as the circuit description language rather than, say, a Spice netlist. This arrangement yields a level of flexibility difficult to achieve using a netlist parser or graphical interface. On the other hand, the netlist must be compiled and the simulator linked for each circuit, and the user must be aware of the subtle details of the C++ syntax.

Other considerations about the OO design of circuit simulators are presented in [11].

1.3 Original Contributions

The simulator engine of the pre-existing Transim simulator program [2] was re-designed and coded from scratch using object oriented techniques. Only the parser and the output routines from the original program were kept, and they had been significantly modified to work with the new simulator engine and support new features. One of the outstanding new features is the *generic*

²<http://www.geda.seul.org/dist/>

element evaluation mechanism, described in Section 7.4. No other circuit simulator provides the same level of flexibility for the addition of new nonlinear device models and circuit analysis algorithms. Transim also supports the local reference node concept [12]. The conventional nodal specification enables circuit elements to be connected in any possible combination and only one reference node (commonly called the global reference node or simply ground) is used. With spatially distributed circuits it is possible to make non-physical connections such as connecting a non-spatially distributed element, say a resistor, across two physically separated (relative to the wavelength) parts of the circuit, *e.g.* the opposite ends of a transmission line. The local reference concept (described in Section 8.2) avoids this kind of problem and therefore is fundamental for the analysis of spatially distributed circuits as well as for simultaneous thermal-electrical simulations.

Another original contribution is the circuit formulation of Chapter 3, which was developed as a generalization of the harmonic balance method presented by the author in [2]. All the nonlinear circuit analysis techniques presented in this thesis are derived from the generic formulation in Chapter 3. This formulation provides a tool for the derivation of new circuit analysis techniques. It uses Rizzoli's state variable concept discussed in Section 1.2 and thus inherits the associated advantages of this approach. The number of nonlinear unknowns resulting from our formulation is generally much smaller than the number of unknowns in conventional circuit analysis.

The convolution transient method developed by Ozkar in [13] was enhanced and re-implemented more efficiently in the newly designed Transim (Chapter 4). Convolution transient provides modeling of frequency-defined network elements with more accuracy than other methods, but it will be shown that it is useful when the number of time steps to be simulated is not very large. The implementation of this circuit analysis technique was the first chronologically and allowed the transient simulation for the first time of a NLTL considering frequency-dependent attenuation (Section 4.7) and, also for the first time, the transient simulation of a quasi-optical grid amplifier (Section 8.3.4).

The wavelet transient formulation presented in Chapter 5 is another contribution of this thesis. Wavelet basis functions are ideally suited to expanding the response of systems with an overall coarse response but fine behavior in some regions as higher order and more localized basis functions can be concentrated on the regions where the response varies rapidly. The wavelet transient analysis in Transim is the most advanced wavelet-based

nonlinear circuit simulator developed to date. The simulation examples using the NLTL and the quasi-optical (QO) grid amplifier presented here are the most complex circuits ever simulated using wavelets. Nevertheless, as it will be shown, this circuit analysis technique requires more research to become efficient.

The state variable transient analysis based on time marching integration methods (Chapter 6), both formulation and implementation in Transim is original from this thesis. It is shown that this transient analysis offers more than an order of magnitude reduction (and it has the potential for even better performance) in the transient simulation time compared with Spice for some circuits. When combined with pole-zero modeling of EM structures, the simulations of QO systems using the method of Chapter 6 are several orders of magnitude faster than the simulations using convolution transient. Therefore, this is the preferred circuit analysis method if accurate pole-zero models for QO are available.

Other contributions include the integration of thermal circuit elements based on the *Leeds' thermal impedance matrix model* to enable electro-thermal simulations in Transim (Section 8.4). This was a joint effort with Dr. William Batty of Leeds University, in the UK. Also the combination and the actual coding in Transim of the frequency spectrum generation technique of Section 2.5 with the frequency mapping technique in harmonic balance to allow the simulation of spectral regrowth using HB.

1.4 The Global Modeling Concept

Fig. 1.4 illustrates the model integration concept. In this circuit-oriented approach the high-level circuit abstraction is retained, and the results of EM analysis of the spatially distributed circuit are incorporated into the circuit framework. The following analogy is used to represent thermal systems as circuits. The temperature of a surface where heat is being exchanged is a potential, equivalent to voltage in an electrical circuit, and the heat exchanged is a flux, equivalent to a current in an electrical circuit. Thus we have a general unifying concepts of fluxes and potentials which leads to a universal error formulation: the sum of fluxes at a node must be zero and all potentials calculated for a particular node must be the same. This concept is expanded on later in this dissertation. This analogy is used to model the thermal subsystem as a circuit. The different physical thermal components are thus

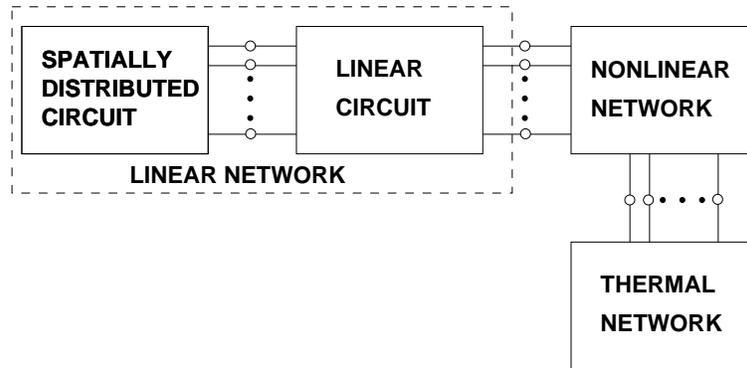


Figure 1.4: Partition of a system into spatially distributed and lumped linear circuit, nonlinear network, and thermal parts.

modeled independently and connected to form a thermal circuit. Note that the link between the electrical and thermal parts is through the nonlinear elements. This is so because the electrical power dissipation is always a nonlinear process. The spatially distributed circuit could certainly be directly connected to the nonlinear devices and this is consistent with this model.

The analysis strategy described in Chapter 3 shows that the complete nonlinear system can be simulated by knowing only the state of the nonlinear network. However, since the thermal subsystem is often nonlinear, part of it must be embedded into the nonlinear network to achieve this.

1.5 Thesis Overview

The second Chapter is dedicated to the literature review. Chapter 3 describes the generalized state variable analysis used in this work to analyze circuits and systems. Then there is one Chapter for each concrete circuit analysis derived from the general approach. Each method is formulated, implemented and tested. The pros and cons of each method are discussed. Convolution transient is treated in Chapter 4, wavelet transient in Chapter 5 and timemarching transient in Chapter 6. The system used for testing is a nonlinear transmission line described in Section 4.6.

Chapter 7 describes the architecture of the circuit simulator developed here, Transim, used to implement all the ideas described here. The architecture of Transim is in itself a piece of innovation because several new concepts

were used in its design.

Chapter 8 presents modeling issues specific to quasi-optical systems, including the need for local reference nodes, the modeling of electromagnetic structures and thermal effects. Two different structures are simulated using different methods and the results are commented.

Finally, Chapter 8 contains general conclusions and ideas for continuation of this research.

The first appendix describes the *Graphical User Interface* (GUI) developed to make the use of Transim easier. The second presents a very short and incomplete introduction to object oriented programming that might be nevertheless useful to understand Chapter 7.

1.6 Publications

1.6.1 Journals

1. C. E. Christoffersen, U. A. Mughal and M. B. Steer, “Object Oriented Microwave Circuit Simulation,” *Int. J. of RF and Microwave Computer-Aided Engineering*, Vol. 10, Issue 3, 2000, pp. 164–182.
2. C. E. Christoffersen, M. Ozkar, M. B. Steer, M. G. Case and M. Rodwell, “State variable-based transient analysis using convolution,” *IEEE Transactions on Microwave Theory and Techniques*, Vol. 47, June 1999, pp. 882–889.
3. C. E. Christoffersen and M. B. Steer “Implementation of the local reference concept for spatially distributed circuits,” *Int. J. of RF and Microwave Computer-Aided Eng.*, vol. 9, No. 5, 1999, pp. 376–384.
4. M. B. Steer, J. F. Harvey, J. W. Mink, M. N. Abdulla, C. E. Christoffersen, H. M. Gutierrez, P. L. Heron, C. W. Hicks, A. I. Khalil, U. A. Mughal, S. Nakazawa, T. W. Nuteson, J. Patwardhan, S. G. Skaggs, M. A. Summers, S. Wang, and A. B. Yakovlev, “Global modeling of spatially distributed microwave and millimeter-wave systems,” *IEEE Trans. Microwave Theory Techniques*, June 1999, pp. 830–839.

1.6.2 Conferences

1. C. E. Christoffersen, S. Nakazawa, M. A. Summers, and M. B. Steer,

- “Transient analysis of a spatial power combining amplifier”, *1999 IEEE MTT-S Int. Microwave Symp. Dig.*, June 1999, pp. 791–794.
2. W. Batty, C. E. Christoffersen, S. David, A. J. Panks, R. G. Johnson, C. M. Snowden and M. B. Steer, “Predictive microwave device design by coupled electro-thermal simulation based on a fully physical thermal model,” *EDMO 2000*, Glasgow UK, November 2000.
 3. W. Batty, C. E. Christoffersen, S. David, A. J. Panks, R. G. Johnson, C. M. Snowden and M. B. Steer, “Steady-state and transient electro-thermal simulation of power devices and circuits based on a fully physical thermal model,” *THERMINIC 2000 Digest*, Budapest, September 2000.
 4. H. Gutierrez, C. E. Christoffersen and M. B. Steer, “An integrated environment for the simulation of electrical, thermal and electromagnetic interactions in high-performance integrated circuits,” *Proc. IEEE 6th Topical Meeting on Electrical Performance of Electronic Packaging*, Sept. 1999, pp. 217–220.
 5. M. B. Steer, C. E. Christoffersen, H. Gutierrez, S. Nakazawa, M. Abdulla, and T. W. Nutesson, “Modelling of Large Non-Linear Systems Integrating Thermal and Electromagnetic Models,” *European Gallium Arsenide and related III-V Compounds Application Symposium*, October 1998, pp. 169–174.

Chapter 2

Literature Review

First, microwave circuit transient analysis techniques are reviewed. These are relevant to the new techniques proposed in the following chapters. Section 2.2 is dedicated to the application of wavelets to the simulation of circuits and a description of the wavelet basis functions used to implement the method in Chapter 5. Section 2.3 briefly presents a review of time marching integration methods. We will use these methods to derive a new type of circuit analysis in Chapter 6. Section 2.4 presents the state variable approach adopted here to analyze nonlinear circuits. Section 2.5 is dedicated to the review of a method to produce all the intermodulation products present in a circuit given certain spectrum of the exciting tones. This novel technique was applied combined with harmonic balance (possibly for the first time) in Transim to evaluate spectral regrowth in power combiners.

2.1 Transient Analysis of Microwave Circuits

Distributed linear microwave circuits are described by frequency-dependent network parameters and only a few methods are available to accommodate these circuits in transient simulation. These methods include Impulse Response and Convolution (IRC), Laplace Inversion and the approximation of network functions using pole-zero models, which be subdivided in Asymptotic Waveform Evaluation (AWE) and Interpolation Methods. We briefly describe these methods in the following sections.

2.1.1 Impulse Response and Convolution

One of the first implementations of IRC to distributed microwave circuits was by Djordjevic and Sarkar [14] in 1987. Since that time there have been several efforts to reduce the significant aliasing errors that result from the Inverse Fast Fourier Transform (I-FFT) operation.

Minimization of aliasing in the I-FFT requires that the imaginary part of the frequency response be zero at the maximum frequency. Low pass filtering has been used to achieve this [15]. The introduction of a small time delay also achieves this result with presumably less distortion [16]. Insertion of an augmentation network in the linear network at the interface with the non-linear network achieves the same result for special types of circuits [17]. The effect of the augmentation network is compensated in the nonlinear iteration scheme. This, however, is not a general technique. It is also important to limit the length of the impulse response to reduce memory requirements and the resistive augmentation achieves this result [17]. Augmentation is also effectively achieved using scattering (S) parameters [18], where the reference impedance effectively dampens multiple reflections.

Distributed networks are characterized partly by reflections so that an impulse response tends to have regions of low value between regions of rapid change. In this case thresholding greatly reduces the number of impulse response discretizations that need to be retained [17]. In high speed digital interconnect circuitry this can reduce the number of significant impulse responses by a factor of 10 to 100, depending on the desired accuracy [17].

Convolution as generally implemented uses a rectangular integration scheme (essentially an impulse response is treated as being constant in a time step interval). P. Stenius *et al.* [19] developed a trapezoidal form of the convolution integration which should have superior convergence properties than the previous block integration.

2.1.2 Numerical Inversion of Laplace Transform Technique

This technique does not have aliasing problems since it does not assume that the function is periodic. The inverse transform exists for both periodic and non-periodic functions. There is no causality problem for double sided Laplace Transforms, either. Unlike FFT methods, the desired part of the response can be achieved without doing tedious and unnecessary calculations

for the other parts of the response. Laplace techniques suffer from the series approximations and the nonlinear iterations involved. If only $Y(j\omega)$ is available (*e.g.* through measurements) the I-FFT must be used instead of the NILT. The advantages and the limitations of the Inverse Laplace Methods are discussed in detail in [20].

2.1.3 Approximations using Pole-Zero Models

Asymptotic Waveform Evaluation

The frequency dependent network parameters can be modeled using frequency independent primitives (resistors, inductors and capacitors) if a rational polynomial transfer function is fitted to the network parameters. In practice, this procedure results in an impossibly large circuit. The AWE method addresses this problem by reducing the dimension of the rational polynomial while minimizing distortion [21,22]. This method works well for interconnects in digital systems and lower frequency microwave circuits [23,24]. However, higher frequency circuits can only be modeled approximately using AWE due to the infinite number of poles and zeros of such a circuit.

Most of the AWE methods use the Padé approximation and this and other approximations used can have stability problems [21].

Interpolation Methods

Besides AWE, there are other techniques to approximate network parameters using a rational polynomial transfer function. Here we will briefly summarize the method that will be used in Section 8.3.2. The procedure in detail was presented by Beyene *et al.* in [25].

A network function $Y(s)$ can be approximated by the rational function

$$H(s) = \frac{q_0 + q_1s + q_2s^2 + \dots + q_\xi s^\xi}{1 + r_1s + r_2s^2 + \dots + r_\vartheta s^\vartheta}, \quad (2.1)$$

The method reviewed here utilizes the special properties of network functions to provide an accurate approximation given a set of data points of the original function. For instance, the coefficients of the polynomials in the rational function must be real and all roots in the denominator polynomial must have zero or negative real parts. In addition, network functions are analytic functions of a complex variable; hence their real and imaginary parts are related.

Let ω_{min} and ω_{max} be the lowest and highest frequencies of the data sample set, respectively. The first step in this method is to normalize and shift frequency points to map $[\omega_{min}, \omega_{max}]$ onto $[-1, 1]$.

Then, the real part of the original function $Y(s)$ is fitted to the real part of the rational polynomial function $H(s)$, leading to the following form

$$\Re(H(s)) = \frac{C(s)}{D(s)} = \frac{c_0 + c_1s^2 + c_2s^4 + \dots + c_\xi s^{2\xi}}{1 + d_1s^2 + d_2s^4 + \dots + d_\vartheta s^{2\vartheta}} \quad (2.2)$$

Let y_i be

$$y_i = Y(j\omega_i) \quad (2.3)$$

i.e., the value of the network function at $s = j\omega_i$. Then, from Equation (2.2), making $\Re(H(j\omega_i)) = \Re(y_i)$,

$$C(j\omega_i) - D(j\omega_i)\Re(y_i) = 0 \quad (2.4)$$

A linear system of equations is built by applying Equation (2.4) at each frequency sample. There are some considerations about how many sample points are necessary. At least $n = \xi + \vartheta + 1$ are needed. If there are more samples, [25] proposes to use *the method of averages* [26]. Briefly, the equations (more than n) are arranged in n groups, and the equations in each group are added so the final system is square. Note that then the approximation will no longer be an interpolation.

The linear system of equations is solved for the polynomial coefficients. The authors of [25] suggest using QR decomposition instead of Gaussian elimination in order to avoid the *growth factor* present in Gaussian elimination. They also state that for this ill-conditioned problem, the orthogonal method gives an added measure of reliability. The frequency normalization also helps to reduce the ill conditioning of the matrix.

Once the coefficients d_i are known, it is possible to extract the poles of the real part by calculating the roots of $D(s)$. Only the poles in the left-hand plane are kept and the rest is discarded. Pure-imaginary complex conjugate poles are taken only if they are double. The extracted poles are denoted p_i .

The solution for a given pair ξ, ϑ may not be acceptable. If so, a new order of approximation is selected and the previous calculation repeated.

Write $H(s)$ in (2.1) in pole-residue form

$$H(s) = k_\infty + \sum_{i=1}^{\vartheta'} \frac{k_i}{s - p_i} \quad (2.5)$$

where $\vartheta' \leq \vartheta$ and $\vartheta - \vartheta'$ is the number of rejected purely imaginary poles. k_i denotes the residues and k_∞ represents any direct coupling between input and output. From this form, it is possible to formulate another linear system to find the residues and finally obtain the approximating rational function.

2.2 Multiresolution Analysis of Circuits

In this section we review methods for the simulation of circuits with nonuniform meshes. We concentrate in techniques based on wavelets. Since the simulation of circuits consist in solving coupled algebraic and *ordinary differential equations* (ODEs), Section 2.2.2 reviews wavelet methods applied to the solution of differential equations. One particular method is reviewed in section 2.2.3. The implementation of the wavelet transient analysis presented in Chapter 5 is based on this.

2.2.1 Adaptive Circuit Simulation Methods

Adaptive circuit simulation methods attempt to adapt the position and the number of time samples of the different circuit variables in order to save computation time and memory. The difference between this approach and conventional time-marching methods is that the position of time samples must not necessarily be shared by all circuit variables. One natural approach to achieve this is to use wavelets. Wavelet methods for the solution of PDEs have direct application in the analysis of electromagnetic structures and in the modeling of semiconductor devices. The transient analysis of general circuits involves the solution of a set of nonlinear coupled *algebraic* and ODEs. Methods developed with PDEs in mind can be modified to solve circuit equations, however its use is not widespread.

Wavelets have been used to speed up the calculation of the convolution operation in the transient analysis of circuits [27, 28]. The basic idea of the method is as follows. Both the signal and the impulse response are decomposed using an orthonormal wavelet base. The convolution integration is carried out independently at each resolution level. The final result is exactly the convolution of the original input signal and the impulse response function. However, the perfect reconstruction does not save any computational time. As a result the elements of the sub-band signals that are less than a preset tolerance are filtered out and a sparse storage scheme is applied. This

method was shown to speed up the convolution operation by around two to six times comparing with the conventional FFT method.

Other applications include, but they are not limited to, analysis of transient signals in circuits ([29,30] are some examples) and the analysis of linear time-variant electrical networks [31].

Zhou *et al.* presented the pseudo-wavelet collocation method developed in [32] modified to solve linear ordinary differential equations (ODEs) in [33] and nonlinear ODEs in [34]. The same method and examples were presented with some more detail using the boundary wavelets proposed in [35] in [36] and [37]. This method is described in Section 2.2.3.

Wenzler *et al.* [38] propose an adaptive algorithm based on interpolating splines. The interpolation points are adjusted to minimize the error. This method does not use wavelets.

2.2.2 Wavelets Applied to Solve Differential Equations

Wavelets offer a means of approximating functions that allows selective grid refinement [39]. If regions of an image or a signal have exceptionally large variations, one need only store a set of coefficients, determined by function values in the neighborhoods of those regions, in order to reproduce these variations accurately. In this way, one can have approximations of functions in terms of a basis that has spatially varying resolution. This approach reduces the memory storage required to represent functions and may be used for data compression.

Wavelet approximations [32] have attracted much attention as a potentially efficient numerical technique for solving partial differential equations. Because of their advantageous properties of localizations in both space and frequency domains, wavelets seem to be great candidate for adaptive and multiresolution schemes to obtain solutions which vary dramatically both in space and time and develop singularities. On one hand, these methods allow one to design methods of arbitrarily high order, and on the other hand they display a local behavior. For introductory material on wavelets the reader is referred to [40,41].

Collocation methods are considered to be more efficient for the treatment of nonlinear problems with nonuniform grids [42] since all calculations are performed in the physical space and there is no need to evaluate integrals. The idea behind collocation methods is quite simple. The unknown function is approximated by a wavelet series expansion. The expansion coefficients

must verify the equation exactly at some *collocation points*.

2.2.3 Spline Pseudo-Wavelet Collocation Method

This method is of special interest since it has been applied to circuit transient simulation by Zhou *et al.* in [33, 36]. The Transim implementation of the wavelet transient formulation presented in Chapter 5 is also based on this.

Scaling Spline Wavelet Functions

First a short introduction to multiresolution analysis (MRA) is presented based on the material in [32]. Refer to the original article for mathematical proofs. For simplicity, we will use the term ‘wavelet’ in this discussion with the understanding that it is different from the usual wavelet which has a non vanishing moment.

Let I denote a finite interval $I = [0, L]$, L is a positive integer, $L > 4$ and $H^2(I)$ and $H_0^2(I)$ denote the following two Sobolev spaces:

$$H^2(I) = \{f(t), t \in I \mid \|f^{(i)}\|_2 < \infty, i = 0, 1, 2\} \quad (2.6)$$

$$H_0^2(I) = \{f(t) \in H^2(I) \mid f(0) = f'(0) = f(L) = f'(L) = 0\} \quad (2.7)$$

$H_0^2(I)$ is a Hilbert space equipped with inner product

$$\langle f, g \rangle = \int_I f''(t)g''(t)dt \quad (2.8)$$

thus

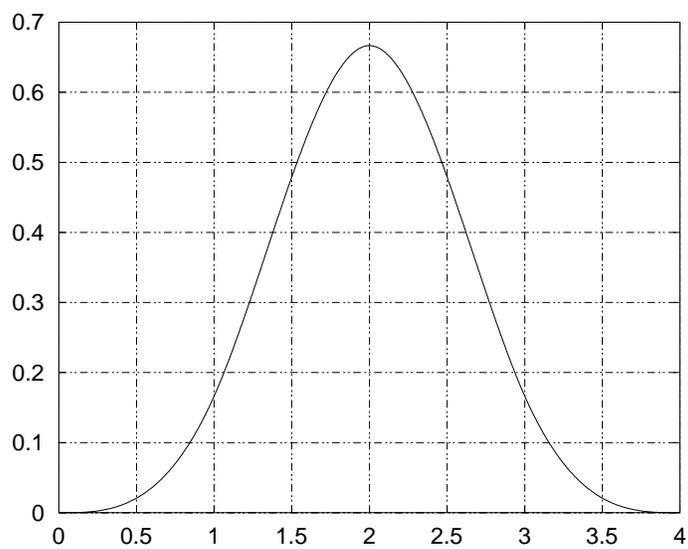
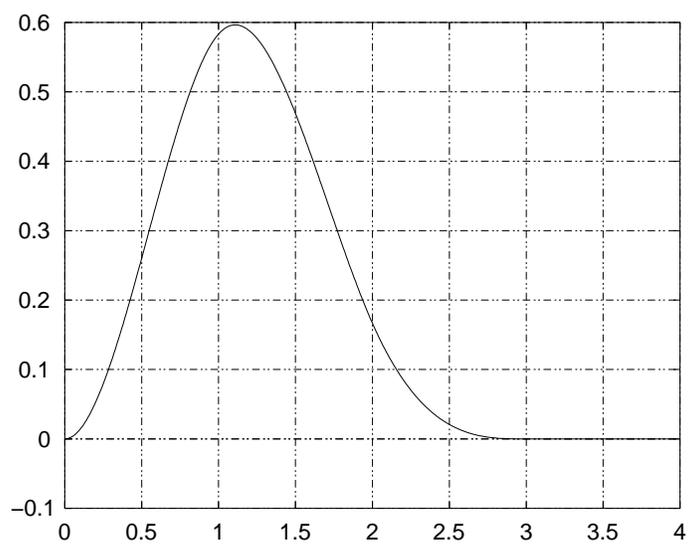
$$\|f\| = \sqrt{\langle f, f \rangle} \quad (2.9)$$

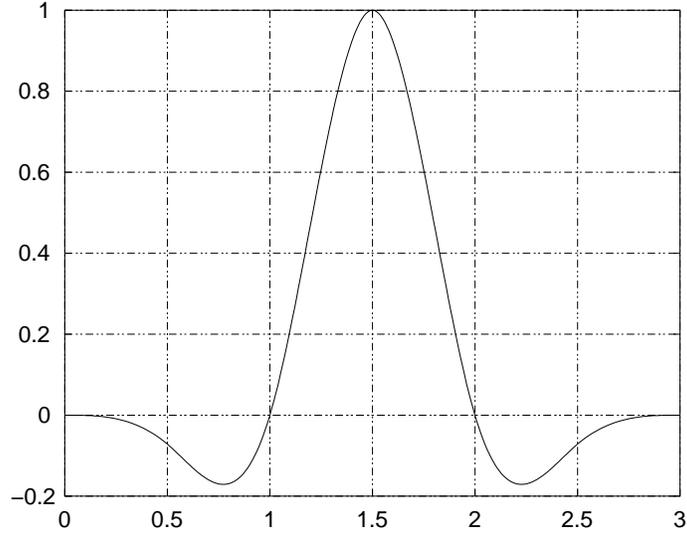
provides a norm for $H_0^2(I)$.

Approximation of a Function in $H_0^2(I)$

To generate an MRA for the Sobolev space $H_0^2(I)$, we consider two scaling functions; an interior scaling function $\varphi(t)$ and a boundary scaling function $\varphi_b(t)$ (see Figs. 2.1 and 2.2)

$$\begin{aligned} \varphi(t) &= N_4(t) = \frac{1}{6} \sum_{j=0}^4 \binom{4}{j} (-1)^j (t-j)_+^3 \\ \varphi_b(t) &= \frac{3}{2}t_+^2 - \frac{11}{12}t_+^3 + \frac{3}{2}(t-1)_+^3 - \frac{3}{4}(t-2)_+^3 \end{aligned}$$

Figure 2.1: The scaling function $\varphi(t)$.Figure 2.2: The boundary scaling function $\varphi_b(t)$.

Figure 2.3: The wavelet function $\psi(t)$.

$N_4(t)$ is the fourth-order B-spline and for any real number n ,

$$x_+^n = \begin{cases} x^n & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

For any $j, k \in \mathbb{Z}$, we define

$$\begin{aligned} \varphi_{j,k}(t) &= \varphi(2^j t - k) \\ \varphi_{b,j}(t) &= \varphi_b(2^j t) \end{aligned}$$

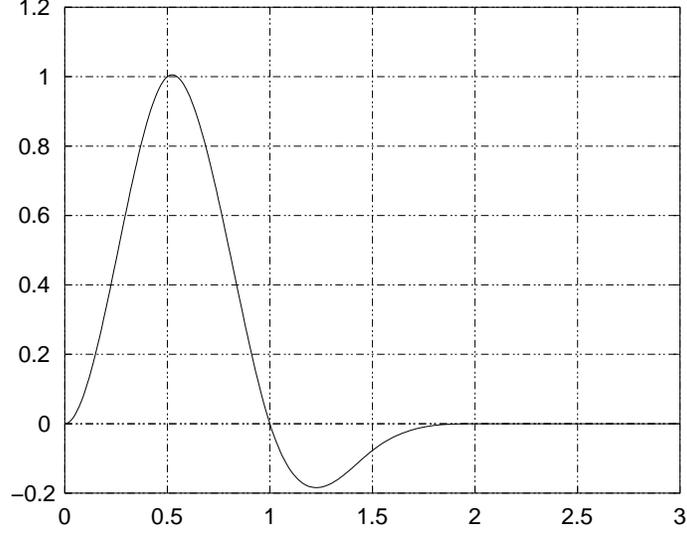
and let V_j be

$$V_j = \text{span}\{\varphi_{j,k}(t) | \varphi_{b,j}(t), \varphi_{b,j}(L - t) : 0 \leq k \leq 2^j L - 4\} \quad (2.10)$$

Let V_j , $j \in \mathbb{Z}^+$ be the linear span of (2.10). Then V_j forms an MRA in $H_0^2(I)$ equipped with norm, Equation (2.9), and for each j , $\{\varphi_{j,k}(t), \varphi_{b,j}(t), \varphi_{b,j}(L - t)\}$ is a basis of V_j .

To construct a wavelet decomposition of Sobolev space $H_0^2(I)$ under the inner product of Equation (2.8), we consider the following two wavelet functions (see Figs. 2.3 and 2.4)

$$\begin{aligned} \psi(t) &= -\frac{3}{7}\varphi(2t) + \frac{12}{7}\varphi(2t - 1) - \frac{3}{7}\varphi(2t - 2) \in V_1 \\ \psi_b(t) &= \frac{24}{13}\varphi_b(2t) - \frac{6}{13}\varphi_b(2t) \in V_1 \end{aligned}$$

Figure 2.4: The boundary wavelet function $\psi_b(t)$.

It can be verified that both $\psi(t)$ and $\psi_b(t)$ belong to V_1 and

$$\psi(n) = \psi_b(n) = 0 \text{ for all } n \in Z$$

Now define

$$\begin{aligned} \psi_{j,k}(t) &= \psi(2^j t - k) \\ \psi_{b,j}^l(t) &= \psi_b(2^j t) \\ \psi_{b,j}^r(t) &= \psi_b(2^j(L - t)) \end{aligned}$$

where $j \geq 0, k = 0, \dots, n_j - 3$ and $n_j = 2^j L$. For simplicity, we will adopt the following notation

$$\begin{aligned} \psi_{j,-1}(t) &= \psi_{b,j}^l(t) \\ \psi_{j,n_j-2}(t) &= \psi_{b,j}^r(t) \end{aligned}$$

so, when $k = -1$ or $n_j - 2$, wavelet functions $\psi_{j,k}$ will denote the two boundary wavelet functions, which can not be obtained by translating and dilating $\psi(t)$.

Finally, for each $j \geq 0$ define

$$W_j = \text{span}\{\psi_{j,k}(t) | k = -1, \dots, n_j - 2\} \quad (2.11)$$

The W_j , $j \geq 0$ as defined in Equation (2.11) is the orthogonal complement of V_j in V_{j+1} under the inner product of Equation (2.8). Therefore,

$$\begin{aligned} W_j &\perp W_{j+1}, \quad j \in Z^+ \\ H_0^2(I) &= V_0 \bigoplus_{j \in Z^+} W_j \end{aligned}$$

where the symbol \perp indicates orthogonality under the inner product of Equation (2.8) and $V_{j+1} = V_j \oplus W_j$ indicates $V_j \perp W_j$ and $V_{j+1} = V_j + W_j$. As a consequence of this, any function $f(t) \in H_0^2(I)$ can be approximated as closely as needed by a function $f_j(t) \in V_j = V_0 \oplus W_0 \oplus \dots \oplus W_{j-1}$ for a sufficiently large j , and $f_j(t)$ has a unique orthogonal decomposition

$$f_j(t) = f_0 + g_0 + \dots + g_{j-1} \quad (2.12)$$

where $f_0 \in V_0$ and $g_i \in W_i$, $0 \leq i \leq j-1$.

Approximation of a Function in $H^2(I)$

Consider the following two splines (Figs. 2.5 and 2.6)

$$\begin{aligned} \eta_1(t) &= (1-t)_+^2 \\ \eta_2(t) &= 2t_+ - 3t_+^2 + \frac{7}{6}t_+^3 - \frac{4}{3}(t-1)_+^2 + \frac{1}{6}(t-2)_+^3 \end{aligned}$$

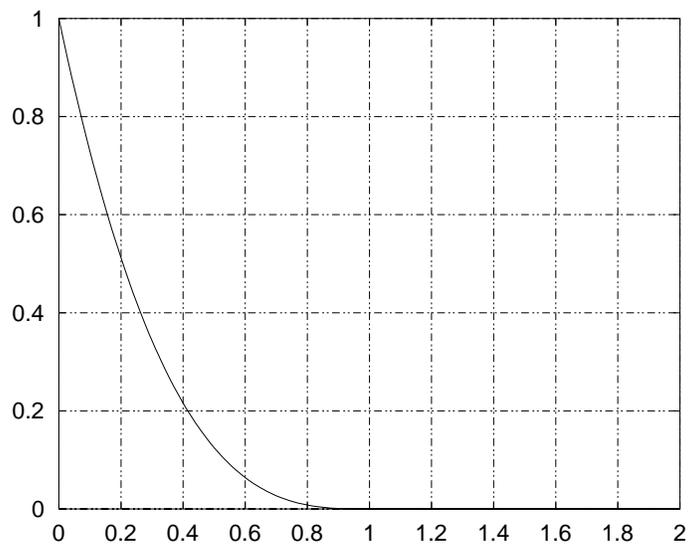
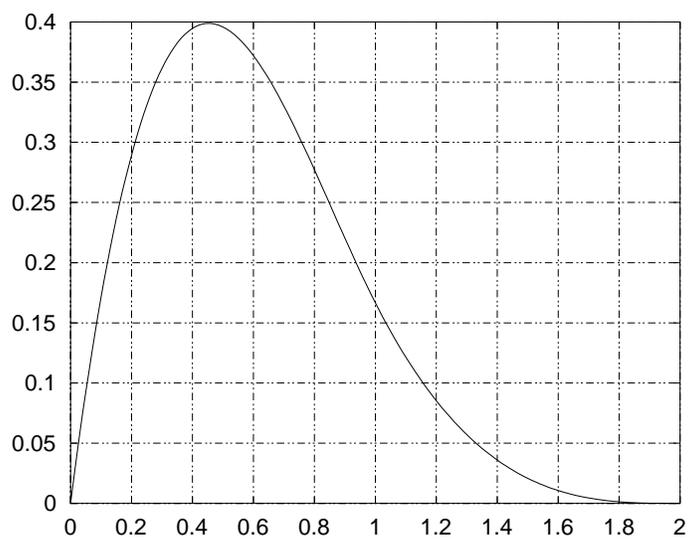
For any function $f(t) \in H^2(I)$, we can define the following interpolating spline $\mathbf{I}_{b,j}f(t)$, $j \geq 0$, expected to approximate the non homogeneities of the function $f(t)$ at the boundaries.

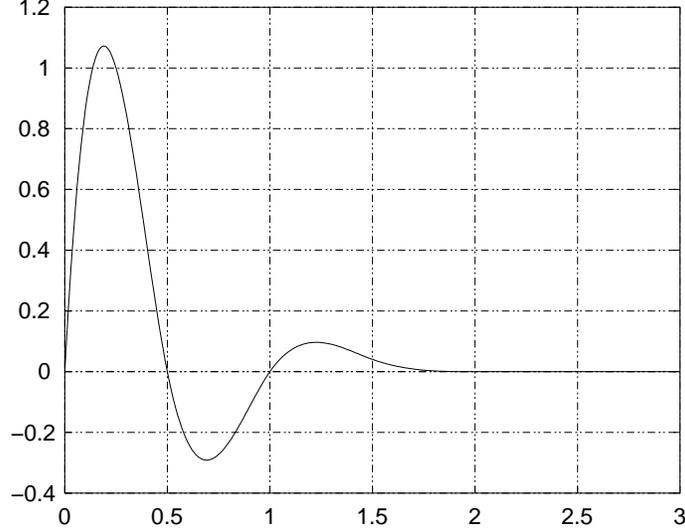
$$\mathbf{I}_{b,j}f(t) = \alpha_1\eta_1(2^j t) + \alpha_2\eta_2(2^j t) + \alpha_3\eta_2(2^j(L-t)) + \alpha_4\eta_1(2^j(L-t))$$

The coefficients $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, must be chosen such that

$$\begin{aligned} \mathbf{I}_{b,j}f(0) &= f(0) \\ \mathbf{I}_{b,j}f(L) &= f(L) \\ (\mathbf{I}_{b,j}f)'(0) &= f'(0) \\ (\mathbf{I}_{b,j}f)'(L) &= f'(L) \end{aligned}$$

The derivative values $f'(0)$ and $f'(L)$ can be approximated using finite differences while preserving the exact order of accuracy of the cubic spline approximation as shown in [32].

Figure 2.5: Boundary spline: $\eta_1(t)$.Figure 2.6: Boundary spline: $\eta_2(t)$.

Figure 2.7: Boundary wavelet function: ψ_{b0} .

Now we have $f(t) - \mathbf{I}_{b,j}f(t) \in H_0^2(I)$ and the decomposition of (2.12) can be applied. Finally, for any function $f(t) \in H^2(I)$, we can find a function $f_j(t)$ in the form of

$$f_j(t) = \mathbf{I}_{b,j}f(t) + f_0 + g_0 + \dots + g_{j-1} \quad (2.13)$$

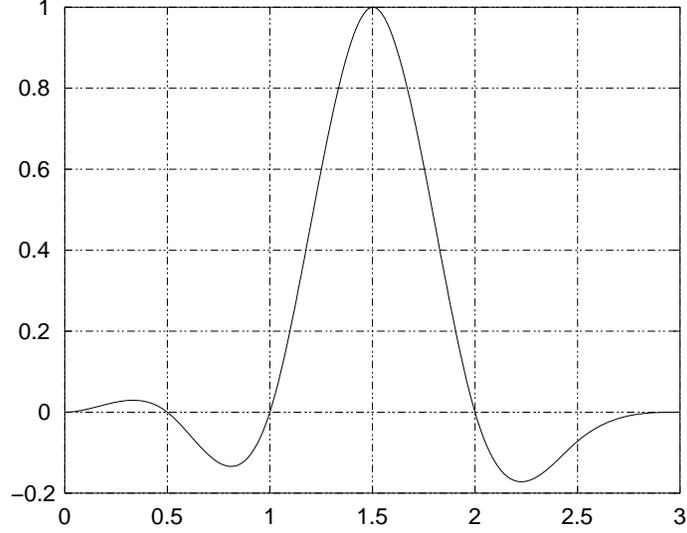
which approximates $f(t)$ as closely as needed provided that j is large enough.

Not-a-Knot Conditions

In many applications, the solutions vary dramatically near the boundary, and the finite difference approximation suggested in the previous section could result in large errors. To overcome this problem, [32] proposes that $\mathbf{I}_{b,j}f(t)$ agrees with $f(t)$ at one additional point near each boundary. Reference [35] also proposes the following modifications. The boundary wavelet, ψ_b , is replaced with (Figs. 2.7 and 2.8) and

$$\begin{aligned} \psi_{b0}(t) &= -\frac{56}{99}(\psi_{0,-1}(t) + 14\psi_{0,-2}(t)) \\ \psi_{b1}(t) &= \frac{182}{181}(\psi(t) + \frac{1}{13}(\psi_{0,-1}(t) + \psi_{0,-2}(t))) \end{aligned}$$

where $\psi_{0,-1}(t) = \psi(t+1)$ and $\psi_{0,-2}(t) = \psi(t+2)$. They also consider a different set of collocation points from those defined in [32] in order to reflect

Figure 2.8: Boundary wavelet function: ψ_{b1} .

the extra boundary functions. First, the scaling space V_0 and wavelet space W_j are redefined. For any $j, k \in Z$, we have

$$\varphi_{0,k}(t) = \varphi(t - k), \quad 0 \leq k \leq L - 4 \quad (2.14)$$

$$\varphi_{0,-3}(t) = \eta_1(t) \quad (2.15)$$

$$\varphi_{0,-2}(t) = \eta_2(t) \quad (2.16)$$

$$\varphi_{0,-1}(t) = \varphi_b(t) \quad (2.17)$$

$$\varphi_{0,L-3}(t) = \varphi_b(L - t) \quad (2.18)$$

$$\varphi_{0,L-2}(t) = \eta_2(L - t) \quad (2.19)$$

$$\varphi_{0,L-1}(t) = \eta_1(L - t) \quad (2.20)$$

and

$$\begin{aligned} \psi_{j,k}(t) &= \psi(2^j t - k) \quad j \leq 0, k = 1, 2, \dots, n_j - 4 \\ \psi_{j,-1}(t) &= \psi_{b0}(2^j t) \\ \psi_{j,0}(t) &= \psi_{b1}(2^j t) \\ \psi_{j,n_j-3}(t) &= \psi_{b1}(2^j(L - t)) \\ \psi_{j,n_j-2}(t) &= \psi_{b0}(2^j(L - t)) \end{aligned}$$

The new spaces V_0 and W_j are then

$$V_0 = \text{span}\{\varphi_{0,k}(t) \mid -3 \leq k \leq L - 1\} \quad (2.21)$$

$$W_j = \text{span}\{\psi_{j,k}(t) \mid -1 \leq k \leq n_j - 2, n_j = 2^j L\} \quad (2.22)$$

It can be checked that $\dim V_0 = L + 3$ and $\dim W_j = n_j$. The collocation points are

$$T^{(-1)} = \left\{0, \frac{1}{2}, 1, 2, \dots, L-1, L - \frac{1}{2}, L\right\} = \left\{x_k^{(-1)}\right\}_{k=-1}^{L+3} \quad (2.23)$$

for V_0 , and

$$T^{(j)} = \left\{\frac{1}{2^{j+2}}, \frac{3}{2^{j+1}}, \dots, \frac{2k+3}{2^{j+1}}, \dots, L - \frac{3}{2^{j+1}}, L - \frac{1}{2^{j+2}}\right\} = \left\{x_k^{(j)}\right\}_{k=-1}^{n_j-2} \quad (2.24)$$

The modification on the boundary scaling functions partially destroys the orthogonality of the bases in Equations (2.15) to (2.20) with respect to the wavelet spaces. However, it can still be proven [35] that $\{\psi_{j,k}(t)\}$ forms a hierarchical basis function over the collocation points in (2.23)–(2.24).

Summarizing, for any function $f(t) \in H^2(I)$, we have an approximate function $f_J(t)$ in the form of

$$f_J(t) = f_0 + g_0 + g_1 + \dots + g_J \quad (2.25)$$

where $f_0 \in V_0$, $g_j \in W_j$ and $0 \leq j \leq J$.

Application to the Numerical Solution of ODEs

The MRA just described is now applied to solve ODEs. Consider the following system of nonlinear differential equations:

$$\begin{cases} \frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}) \\ \mathbf{x}(0) = \mathbf{x}_{t_0} \end{cases} \quad (2.26)$$

where $\mathbf{x}(t)$ is an unknown vector function defined in a finite interval $I = [0, L]$, $L > 4$ and $\mathbf{f}(t, \mathbf{x})$ is a given nonlinear function. A scaling operation is applied to I to cover any interval.

Let $\hat{\mathbf{x}}_J$ be the vector with the wavelet coefficients of $\mathbf{x}(t)$:

$$\hat{\mathbf{x}}_J = (\hat{x}_{-1,-3}, \dots, \hat{x}_{-1,L-1}, \hat{x}_{0,-1}, \dots, \hat{x}_{0,n_0-2}, \hat{x}_{J-1,-1}, \dots, \hat{x}_{J-1,n_{J-1}-2})$$

and will be determined by satisfying interpolating conditions at the collocation points. Each component $x_i(t)$ of $\mathbf{x}(t)$ in Equation (2.26) is replaced by its wavelet expansion form at each collocation point. Therefore we obtain the following nonlinear algebraic system

$$A\hat{\mathbf{x}}_J = \hat{\mathbf{f}}(\hat{\mathbf{x}}_J) \quad (2.27)$$

where A is a constant matrix that performs the derivative operation and $\hat{\mathbf{f}}()$ is a nonlinear vector function.

The method to solve differential equations just described provides a *uniform error distribution* on the interval [33]. By this property, large time steps can be used without introducing significant phase shifting between the approximate and exact solutions. The minimization of the error over an interval is the significant departure from conventional transient analysis where error is minimized at one time point at a time.

Adaptive Techniques

In wavelet analysis, computations are carried out from lower subspaces to higher ones. If the wavelet coefficients at the highest subspace level are negligible, then stop, otherwise recalculate using an additional level [33]. The same reference remarks that even in the same subspace level, there is no need to keep all the coefficients, but only the coefficients where the tolerance has not been satisfied. Reference [34] proposes the application of time windowing, a well-known technique also used in waveform relaxation [43]. Then, the original interval is divided into sub-intervals, and the method is applied using different resolutions in each time window.

Iterative Scheme to Solve the Nonlinear Equations

Zhou *et al.* [34] propose to solve the resulting system of algebraic nonlinear equations using the following fixed-point scheme. Derived from Equation (2.27),

$$\hat{\mathbf{x}}^{(k+1)} = \hat{\mathbf{x}}^k - \rho \left(A\hat{\mathbf{x}}^k - \hat{\mathbf{f}}(\hat{\mathbf{x}}^k) \right) \quad (2.28)$$

where $0 < \rho < 1$ is a relaxation factor determined by a one-dimensional search at each iterative step. References [34, 37] recommend the iteration scheme of Equation (2.28) since it is a convergent scheme. However, [37] comments that depending on the initial guess $\hat{\mathbf{x}}^0$ of the coefficients, the scheme may not converge at all.

2.3 Numerical Integration Using Time Marching Methods

The most widespread method of nonlinear circuit analysis is time-domain analysis using numerical integration to determine the circuit response at one instance of time given the circuit's response at a previous instance of time. In this section we review some of the numerical integration methods. Consider the following differential equation

$$x' = f(x, t)$$

where x is a unknown function, t is time, and $f(x, t)$ is a given function.

The equivalent integral equation is [3]

$$x(t_1) = x(t_0) + \int_{t_0}^{t_1} f(x, t).dt$$

where t_1 and t_0 are two times. If $t_1 - t_0$ is small the integral equation can be discretized as

$$x(t_1) \approx x(t_0) + x'(t_1 - t_0)$$

with $x_1 \approx x(t_1)$ but $x_0 = x(t_0)$. That is,

$$x_1 = x_0 + hx'$$

with $h = t_1 - t_0$ the size of the time step. Rewriting this formula at a generic time step t_n we obtain

$$x_n = x_{n-1} + hx' \tag{2.29}$$

This development indicates the relatively straightforward way an integral equation is discretized. The importance of (2.29) is that the future value of a quantity can be predicted from the current value of the quantity. The different integration formulas differ by the method used to estimate x' . In general, many integration formulas¹ can be reduced to this form

$$x'_n = ax_n + b_{n-1}, \tag{2.30}$$

where a is a constant and b_{n-1} depends on the previous history of x .

¹All the implicit integration formulas, including multi-step formulas

2.3.1 Backward Euler Formula

This is the most simple of the implicit methods. Setting $x' = x'_n$ in (2.29),

$$x_n = x_{n-1} + hx'_n$$

The coefficients in Equation (2.30) are then

$$\begin{aligned} a &= \frac{1}{h} \\ b_{n-1} &= -\frac{1}{h}x_{n-1} \end{aligned}$$

2.3.2 Trapezoidal Rule

Setting $x' = (x'_n + x'_{n-1})/2$, the discretized numerical integration Equation (2.29) becomes

$$x_n = x_{n-1} + \frac{h}{2}(x'_{n-1} + x'_n)$$

The coefficients in Equation (2.30) are then

$$\begin{aligned} a &= \frac{2}{h} \\ b_{n-1} &= -\frac{2}{h}x_{n-1} - x'_{n-1} \end{aligned}$$

2.4 Parameterized Nonlinear Models

In this section we present a way to formulate the equations of the nonlinear devices inside a circuit. This concept was originally applied for the piecewise harmonic balance circuit analysis in reference [5]. The idea of piecewise harmonic balance was proposed by Nakhla and Vlach [44]. This approach is based on partitioning the linear and nonlinear portions of a circuit as shown in Figure 2.9.

Let the nonlinear subnetwork be described by the following generalized parametric equations [5]:

$$\mathbf{v}_{\text{NL}}(t) = u[\mathbf{x}(t), \frac{d\mathbf{x}}{dt}, \dots, \frac{d^m\mathbf{x}}{dt^m}, \mathbf{x}_D(t)] \quad (2.31)$$

$$\mathbf{i}_{\text{NL}}(t) = w[\mathbf{x}(t), \frac{d\mathbf{x}}{dt}, \dots, \frac{d^m\mathbf{x}}{dt^m}, \mathbf{x}_D(t)] \quad (2.32)$$

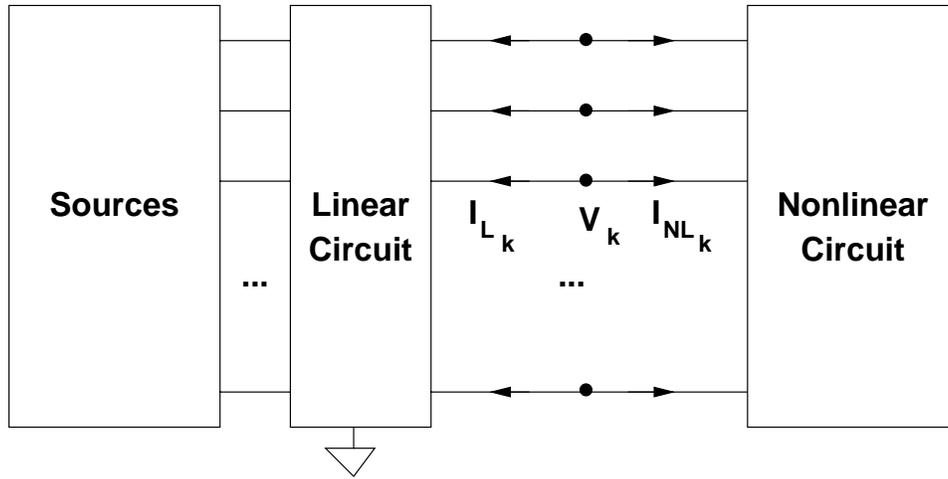


Figure 2.9: Partition representation of the time-invariant harmonic balance method

where $\mathbf{v}_{\text{NL}}(\mathbf{t})$, $\mathbf{i}_{\text{NL}}(\mathbf{t})$ are vectors of voltages and currents at the common ports, $\mathbf{x}(t)$ is a vector of state variables and $\mathbf{x}_D(t)$ a vector of time-delayed state variables, i.e., $[x_D(t)]_i = x_i(t - \tau_i)$. The time delays τ_i may be functions of the state variables. All vectors in Equations (2.31) and (2.32) have the same size equal to the number of common (device) ports. This kind of representation is convenient from the physical viewpoint, as it is equivalent to a set of implicit integro-differential equations in the port currents and voltages. This allows an effective minimization of the number of subnetwork ports, and what is more important, results in complete generality in device modeling. For example, it is no longer necessary to express nonlinear elements as voltage controlled current sources.

2.5 Multitone Frequency-Domain Simulation of Nonlinear Circuits

This section is dedicated to the review of a method which has application in frequency domain steady-state analysis of circuits. The reader is encouraged to consult references [3, 45] for a review of them.

One of the most important open problems in microwave circuit simulation [46] is the prediction of strong nonlinear regimes when the input signal is

composed of a large number of noncommensurate tones. Multi-tone harmonic balance is a mature technique, but when the number of independent tones considered, n_t , is greater than 3, conventional algorithms [4, 45] produce too many mixing components. The number of components is $(2C + 1)^{n_t}$, where C is the maximum nonlinear order considered. When n_t is greater than 2, conventional algorithms may create distinct places of equal frequency in the index vector.

Two alternative approaches have been proposed to analyze strong nonlinear regimes in the frequency domain when the input signal is composed of a large number of noncommensurate tones. The first of them was proposed by Carvalho *et al.* in [46] applied to spectral balance, but it is also valid for harmonic balance, as it will be shown later in Section 8.3.4.

If the input signal spectrum is regular, composed of a set of input frequencies ω_i

$$\omega_i = \omega_0 + k_i \Delta\omega$$

Where ω_0 is the first component, $k_i = 0, 1, \dots, n_t - 1$ and $\Delta\omega$ is the frequency step, then the output spectrum is composed of separate bands with equally spaced tones in them. The center frequency for each band is calculated

$$f_{\text{central}} = \begin{cases} \left(\frac{\omega_m + \omega_{m+1}}{2}\right) c & \text{If } n_t \text{ is even} \\ \omega_m c & \text{If } n_t \text{ is odd} \end{cases} \quad (2.33)$$

where ω_m is the input central frequency and $c = 0, 1, \dots, C$ corresponds to the band being considered.

The spectral width of each band is given by N_c tones separated by the frequency step, $\Delta\omega$. Thus

$$N_c = \begin{cases} Cn_t - C + 1 & \text{If } c \text{ is even} \\ (C - 1)n_t - C + 2 & \text{If } c \text{ is odd} \end{cases} \quad (2.34)$$

Using this formulation, the index vector contains approximately only $n_t C^2 + n_t C$ frequencies instead of the previous $(2C + 1)^{n_t}$, therefore the computational effort of the analysis is greatly reduced.

The second approach is due to Borich *et al.* in [47]. They propose a special kind of Fourier transform to handle regular spectrums and then apply it in a harmonic balance algorithm. But the frequency mapping technique described in [4] can handle the same spectrum and it is more general and easier to implement.

2.6 Summary

In this chapter we reviewed the concepts and techniques that will be incorporated in the work described in the following chapters. Where necessary the concepts are further expanded.

Chapter 3

Generalized State Variable Reduction

This chapter outlines a method of analyzing circuits with the minimum number of unknowns and error functions starting from a *modified nodal admittance matrix* (MNAM) of the linear part of the circuit. This approach has several advantages. The resulting system of nonlinear equations is generally much smaller than the nonlinear system resulting from applying conventional formulations. Also the flexibility of the modified nodal admittance matrix is kept as well as the robustness provided by the state variable approach. This formulation provides a tool for the derivation of new circuit analysis techniques. All the nonlinear circuit analysis techniques presented in this thesis are derived from the generic formulation in this chapter.

In Section 3.1 the circuit is partitioned into a linear and a nonlinear subnetworks. Then the subnetworks are analyzed separately in order to formulate the general equations. Section 3.2 discuss the significance of this formulation.

3.1 Generalized Reduced Nonlinear Error Function Formulation

The formulation of the system equations begins with the partitioned network of Fig. 3.1 with the nonlinear elements replaced by variable voltage or current sources [2]. For each nonlinear element one terminal is taken as the reference and the element is replaced by a set of sources connected to the reference

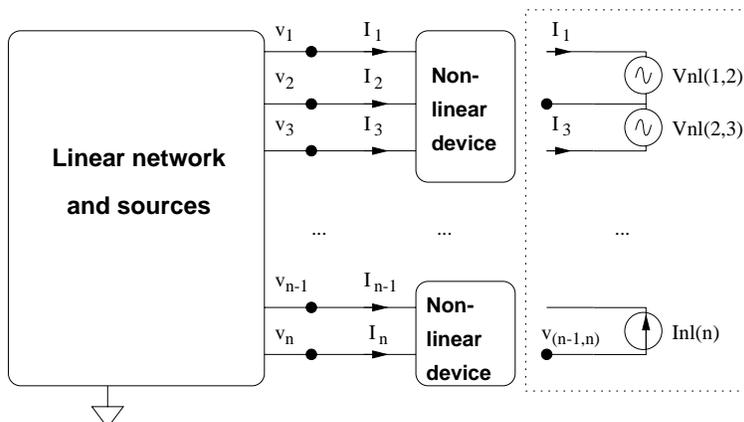


Figure 3.1: Network with nonlinear elements.

terminal. Both voltage and current sources are valid replacements for the nonlinear elements, but current sources are more convenient because they yield a smaller *modified nodal admittance matrix* (MNAM).

3.1.1 Linear Network

The MNAM of the linear subcircuit is formulated as follows. Define two matrices \mathbf{G} and \mathbf{C} of equal size n_m , where n_m is equal to the number of non-reference nodes in the circuit plus the number of additional required variables [48]. Define a vector \mathbf{s} of size n_m for the right hand side of the system. The contributions of the fixed sources and the nonlinear elements (which depend on the time t) will be entered in this vector. All conductors and frequency-independent MNAM stamps arising in the formulation will be entered in \mathbf{G} , whereas capacitor and inductor values and other values that are associated with dynamic elements will be stored in matrix \mathbf{C} . The linear system obtained is the following.

$$\mathbf{G}\mathbf{u}(t) + \mathbf{C}\frac{d\mathbf{u}(t)}{dt} = \mathbf{s}(t), \quad (3.1)$$

where \mathbf{u} is the vector of the nodal voltages and required currents. \mathbf{s} is composed of an independent component \mathbf{s}_f and a component \mathbf{s}_v that depends on the state variables, as in the HB case [2].

$$\mathbf{s}(t) = \mathbf{s}_f(t) + \mathbf{s}_v(t) \quad (3.2)$$

The \mathbf{s}_f vector is due to the independent sources in the circuit. The \mathbf{s}_v vector is the contribution of the currents injected into the linear circuit by the nonlinear network.

3.1.2 Nonlinear Network

The concept of state variable used in this work is the one defined in Section 2.4. The Equations (2.31) and (2.32) are rewritten here for convenience

$$\mathbf{v}_{NL}(t) = u[\mathbf{x}(t), \frac{d\mathbf{x}}{dt}, \dots, \frac{d^m \mathbf{x}}{dt^m}, \mathbf{x}_D(t)] \quad (3.3)$$

$$\mathbf{i}_{NL}(t) = w[\mathbf{x}(t), \frac{d\mathbf{x}}{dt}, \dots, \frac{d^m \mathbf{x}}{dt^m}, \mathbf{x}_D(t)] \quad (3.4)$$

The error function of an arbitrary circuit is developed using connectivity information (described by an incidence matrix and constitutive relations describing the nonlinear elements). The incidence matrix, \mathbf{T} , is built as follows. The number of columns is n_m , and the number of rows is equal to the number of state variables, n_s . In each row, enter “+1” in the column corresponding to the positive terminal of the row nonlinear element port and “-1” in the column corresponding to the negative terminal (the local reference of the port). Then, each row of \mathbf{T} has at most 2 nonzero elements and the number of nonzero elements is at most $2n_s$.

The following equations are then true for all t :

$$\mathbf{v}_L(t) = \mathbf{T}\mathbf{u}(t) \quad (3.5)$$

$$\mathbf{s}_v(t) = \mathbf{T}^T \mathbf{i}_{NL}(t), \quad (3.6)$$

where $\mathbf{v}_L(t)$ is the vector of the port voltages of the nonlinear elements calculated from the nodal voltages of the linear network.

3.1.3 Error Function Formulation

Now we have all the equations necessary to build a nonlinear error function for the entire circuit. Combining Equations (3.1), (3.2), (3.6), the general equation for the linear network is obtained

$$\mathbf{G}\mathbf{u}(t) + \mathbf{C} \frac{d\mathbf{u}(t)}{dt} = \mathbf{s}_f(t) + \mathbf{T}^T \mathbf{i}_{NL}(t) \quad (3.7)$$

The reduced error function $\mathbf{f}(t)$ is defined as follows

$$\mathbf{f}(t) = \mathbf{v}_L(t) - \mathbf{v}_{NL}(t) = 0$$

Replacing $\mathbf{v}_L(t)$ from Equation (3.5),

$$\mathbf{f}(t) = \mathbf{T}\mathbf{u}(t) - \mathbf{v}_{NL}(t) = 0 \quad (3.8)$$

Equations (3.3), (3.4), (3.7) and (3.8) conform the generalized state variable reduction formulation. The error function in Equation (3.8) only depends on the state variables and the time,

$$\mathbf{f}[\mathbf{x}(t), \frac{d\mathbf{x}}{dt}, \dots, \frac{d^n\mathbf{x}}{dt^n}, \mathbf{x}_D(t), t] = 0 \quad (3.9)$$

The dimension of the error function and the number of unknowns are equal to n_s , and this number is the minimum necessary to solve the equations of a circuit without any loss of information. This formulation is very general and can be applied to derive several types of analysis as it will be shown in the following sections of this chapter. The method to approximate $\mathbf{x}(t)$ and $\mathbf{u}(t)$ and their derivatives will determine the type of analysis, as will be shown in the following chapters.

3.2 Discussion

The formulation described in this chapter constitutes a reduction method because, after the method to represent $\mathbf{x}(t)$ and $\mathbf{u}(t)$ (and derivatives) is defined, the size of the resulting square nonlinear system of equations is equal to the number of state variables n_s . In contrast, traditional methods to solve nonlinear circuit equations require to solve for n_m unknowns, and n_m is typically much greater than n_s in microwave circuits. The generality of this formulation allows the generic evaluation mechanism (described in Section 7.4) to be implemented. It is also a theoretical tool that can be used to readily derive a set of equations for a particular type of analysis.

To illustrate the reduction in the number of unknowns achieved by this formulation, consider the quasi-optical grid of Figure 3.2. Let's assume that there are two 3-terminal active devices and a total of 10 nodes (plus reference) per unit cell. Then, the formulation of the resulting circuit using a conventional approach would result $100 \times 10 = 1000$ nonlinear unknowns.

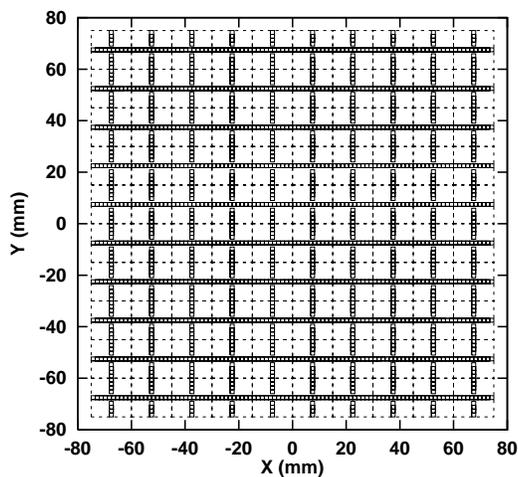


Figure 3.2: A quasi-optical grid.

On the other hand, this state variable approach would need only two state variables per active device, therefore the number of nonlinear unknowns is reduced to 200. If the grid is modeled using rational function approximations, the advantage of our formulation is even better, with one or two order of magnitude (depending on the order of the rational approximation) reduction in the number of nonlinear unknowns. Simulations in Chapters 6 and 8 provide concrete examples of this.

Chapter 4

Convolution Transient Analysis

This chapter develops the equations and describes the software implementation of a transient analysis based on convolution. After the introduction, section 4.2 presents the derivation of the harmonic balance (HB) equations. These are the base for the convolution transient of section 4.3. Section 4.5 presents the implementation of this circuit analysis in Transim. The nonlinear transmission line used to test the simulator is presented in section 4.6. Finally, the simulation results and comments are in section 4.7.

4.1 Introduction

Transient analysis of distributed microwave circuits is complicated by the inability of frequency independent primitives (such as resistors, inductors and capacitors) to model distributed circuits. More generally, the linear part of a microwave circuit is described in the frequency domain by network parameters especially where numerical field analysis is used to model a spatially distributed structure. Inverse Fourier transformation of these network parameters yields the impulse response of the linear circuit. This has been used in the past with convolution to achieve transient analysis of distributed circuits [16, 17, 49, 50].

The major drawback of convolution analysis has been the large time and memory requirements that result from recording and convolving the nodal voltages of the nonlinear elements. The situation worsens because of convergence considerations which require small time steps. This chapter develops a convolution-based transient analysis which uses state variables instead of

node voltages to capture the nonlinear response. This has two desirable effects. First, the number of state variables required is less than the number of nodes of the nonlinear elements and so fewer quantities need to be recorded and convolved. Second, parameterized device models can be used as the nonlinearity is not restricted to the form of current as a nonlinear function of voltage. Parameterized device models result in stable transient analysis for larger time steps and so less discretized time history is required. Also with improved stability constant time steps can be used further improving the robustness of the convolution.

4.2 Harmonic Balance Analysis

The state variable-based convolution transient analysis presented here is based on the harmonic balance analysis as presented in [51], which in turn can be easily derived from Equations (3.7) and (3.8). Expressed in the frequency domain, the equations become

$$\begin{aligned}\mathbf{G}\mathbf{U}(f) + \mathbf{C}\boldsymbol{\Omega}(f)\mathbf{U}(f) &= \mathbf{S}_f(f) + \mathbf{T}^T\mathbf{I}_{NL}(\mathbf{X}(f)) \\ \mathbf{F}(\mathbf{X}(f), f) &= \mathbf{T}\mathbf{U}(\mathbf{X}(f), f) - \mathbf{V}_{NL}(\mathbf{X}(f)) = 0\end{aligned}$$

where f is the frequency and $\boldsymbol{\Omega}$ is the frequency domain derivation operator. Uppercase letters have been used to represent quantities in the frequency domain.

$$\boldsymbol{\Omega} = j 2\pi f \mathbf{I},$$

Here j is the imaginary unit and \mathbf{I} is the identity matrix. It can then readily be shown that

$$\mathbf{F}(\mathbf{X}(f), f) = \mathbf{S}_{SV}(f) + \mathbf{M}_{SV}(f)\mathbf{i}_{NL}(\mathbf{X}(f)) - \mathbf{V}_{NL}(\mathbf{X}(f), f) = 0,$$

where

$$\mathbf{S}_{SV}(f) = \mathbf{T}[\mathbf{G} + \mathbf{C}\boldsymbol{\Omega}(f)]^{-1}\mathbf{S}_f(f) \quad (4.1)$$

and

$$\mathbf{M}_{SV}(f) = \mathbf{T}[\mathbf{G} + \mathbf{C}\boldsymbol{\Omega}(f)]^{-1}\mathbf{T}^T \quad (4.2)$$

4.3 Formulation of the Convolution Transient

The system being modeled is partitioned into linear and nonlinear subnetworks¹ with the linear subnetwork comprising the spatially distributed network characterized by EM analysis. This yields a port-based admittance matrix which is augmented by other linear circuit elements. Sources are included in the nonlinear subnetworks as they must be treated in the time domain. This implies $\mathbf{S}_{SV} = 0$. As voltages and currents at the element's reference nodes are not considered, the number of state variables, n_s , is equal to the number of interface ports. The frequency domain voltages at the interface due to the linear network is

$$\mathbf{V}_L(\mathbf{X}, f) = \mathbf{M}_{SV} \mathbf{I}_{NL}(\mathbf{X}, f) \quad (4.3)$$

After doing discrete Fourier transformation the i th component of $\mathbf{V}_L(f)$ at the discretized time step n_t , $((\mathbf{v}_L)(n_t))_i$ is

$$((\mathbf{v}_L)(n_t))_i = \sum_{j=1}^{n_s} (\mathbf{v}_L(n_t))_{i,j}$$

where $((\mathbf{v}_L)(n_t))_{i,j}$ is defined as

$$((\mathbf{v}_L)(n_t))_{i,j} = \mathcal{F}^{-1}[(\mathbf{M}_{SV})_{i,j}(\mathbf{I}_{NL})_j]$$

Here \mathcal{F}^{-1} denotes the inverse fast Fourier transformation (I-FFT). For the purposes of this transient analysis, $((\mathbf{v}_L)(n_t))_{i,j}$ must be evaluated in the discrete time domain. The multiplication in the frequency domain becomes a convolution operation in the time domain. Then, the contribution of the current at the j th interface node to the voltage at the i th state variable is, assuming for now that the circuit is initially not biased (*i.e.*, zero initial conditions),

$$((\mathbf{v}_L)(\mathbf{x}, n_t))_{i,j} = \begin{cases} (\mathbf{m}_{SV}(0))_{i,j} (\mathbf{i}_{NL}(\mathbf{x}, n_t))_j \\ + \sum_{n_\tau=1}^{n_t-1} (\mathbf{m}_{SV}(n_\tau))_{i,j} (\mathbf{i}_{NL}(n_t - n_\tau))_j, & \text{if } n_t < N_T \\ (\mathbf{m}_{SV}(0))_{i,j}(0) (\mathbf{i}_{NL}(\mathbf{x}, n_t))_j \\ + \sum_{n_\tau=1}^{N_T-1} (\mathbf{m}_{SV}(n_\tau))_{i,j} (\mathbf{i}_{NL}(n_t - n_\tau))_j, & \text{if } n_t \geq N_T \end{cases} \quad (4.4)$$

¹The linear/nonlinear classification is not strict here since some 'linear' elements such as sources or sometimes inductors and capacitors can be treated in the time domain, for example to avoid problems with the impulse response.

where $(\mathbf{m}_{SV}(n_t))_{ij}$ is the I-FFT of $(\mathbf{M}_{SV}(f))_{i,j}$ with duration N_T and is the discrete impulse response contribution of the current at the j th interface node to the voltage at the i th node. Note that $(\mathbf{m}_{SV}(n_t))_{ij} = 0$ for $n_t < 0$ and for $n_t \geq N_T$. Equation (4.4) provides a means to evaluate the response of the linear network in the time domain. Then the error function vector $\mathbf{f} = [f_1 \dots f_i \dots f_{n_s}]^T$ can be formed at each time step, where

$$(f(\mathbf{x}, n_t))_i = \sum_{j=1}^{n_s} (\mathbf{v}_L(\mathbf{x}, n_t))_{i,j} - (\mathbf{v}_{NL}(\mathbf{x}, n_t))_i = 0 \quad (4.5)$$

The transient analysis proceeds as follows. Equation (4.5) is solved for \mathbf{x} at each time step n_t . The value of $\mathbf{v}_{NL}(\mathbf{x}, n_t)$ is obtained directly from the nonlinear device models. The value of $\mathbf{v}_L(\mathbf{x}, n_t)$ is calculated from Equation (4.4). The vector $\mathbf{i}_{NL}(\mathbf{x}, n_t)$ must be saved to be used in the summation in Equation (4.4).

4.3.1 Initial Operating Point

The quickest way to determine the steady state response is to avoid the initial transient which largely determines the dc operating point with a superimposed time-varying response. So, we now drop the assumption that the circuit is not biased (*i.e.* the initial conditions are zero) and assume that the circuit is in its dc operating point at $n_t = 0$. Then $(\mathbf{v}_L(0))_{i,j}$ can be evaluated directly in the frequency domain

$$(\mathbf{v}_L(0))_{i,j} = (\mathbf{M}_{SV}(0))_{i,j} (\mathbf{I}_{NL}(0))_j \quad (4.6)$$

$\mathbf{I}_L(0)$ is the dc current flowing into the nonlinear circuit at the j th nonlinear element port. In time domain

$$(\mathbf{v}_L(0))_{i,j} = (\mathbf{m}_{DC})_{i,j} (\mathbf{I}_{NL}(0))_j$$

where $(\mathbf{m}_{DC})_{i,j}$ is a resistance defined as

$$(\mathbf{m}_{DC})_{i,j} = \sum_{n_\tau=0}^{N_T-1} (\mathbf{m}_{SV}(n_\tau))_{i,j}$$

The initial condition of the state variable vector \mathbf{x} is found by solving (4.5) after replacing the value of $(\mathbf{v}_L(0))_i$ calculated from Equation (4.6). That

vector (\mathbf{x}) is then used as initial condition in solving for the steady-state response with the same effect as establishing the voltage and current conditions before time 0. The initial condition must slowly be removed as the simulation progresses. For $n_t < N_T$ the initial condition contribution $(\mathbf{v}_{L,DC}(n_t))_{i,j}$ is,

$$(\mathbf{v}_{L,DC}(n_t))_{i,j} = (\mathbf{I}_{NL}(0))_j \sum_{n_\tau=n_t}^{N_T-1} (\mathbf{m}_{SV}(n_\tau))_{i,j}$$

This can also be evaluated as

$$(\mathbf{v}_{L,DC}(n_t))_{i,j} = \begin{cases} (\mathbf{m}_{DC})_{i,j} (\mathbf{I}_{NL}(0))_j, & \text{if } n_t = 0 \\ (\mathbf{v}_{L,DC}(n_t - 1))_{i,j} - (\mathbf{I}_{NL}(0))_j (\mathbf{m}_{SV}(n_t - 1))_{i,j} (n_t - 1), & \text{if } 1 \leq n_t < N_T \\ 0, & \text{if } n_t \geq N_T \end{cases}$$

Now the previous expression for $v_{Li,j}$ (4.4) with the addition of the term considering the initial dc operating point becomes

$$(\mathbf{v}_L(\mathbf{x}, n_t))_{i,j} = \begin{cases} (\mathbf{m}_{SV}(0))_{i,j} (\mathbf{i}_{NL}(\mathbf{x}, n_t))_j + (\mathbf{v}_{L,DC}(n_t))_{i,j} \\ + \sum_{n_\tau=1}^{n_t-1} (\mathbf{m}_{SV}(n_\tau))_{i,j} (\mathbf{i}_{NL}(n_t - n_\tau))_j, & \text{if } n_t < N_T \\ (\mathbf{m}_{SV}(0))_{i,j} (\mathbf{i}_{NL}(\mathbf{x}, n_t))_j \\ + \sum_{n_\tau=1}^{N_T-1} (\mathbf{m}_{SV}(n_\tau))_{i,j} (\mathbf{i}_{NL}(n_t - n_\tau))_j, & \text{if } n_t \geq N_T \end{cases} \quad (4.7)$$

Thus, the DC initial condition only adds one term to the convolution sum.

4.4 Error Reduction Techniques

Discretization errors and aliasing errors arise in developing the discrete impulse response. The following sections present some techniques used in Transim to reduce these errors.

4.4.1 Impulse Response Determination

The impulse response is obtained using the inverse real Fourier transformation of each element of \mathbf{m}_{SV} . The transform requires special characteristics of the frequency domain variables at high frequencies so that aliasing is minimized. Problems occur with ideal inductors and capacitors as the

matrix elements can go to infinity. This can be corrected by cascading a resistive augmentation circuit with the linear circuit and so ensure finite parameters [13, 17]. The use of scattering parameters achieves similar resistive augmentation [18]. Being resistive, the effect of the augmentation network can be removed in transient simulation as the resistors are unaffected by the Fourier transformation. The current work uses the resistive augmentation network shown in Fig. 4.1. The advantage of this topology is that no extra nodes are added to the circuit and the size of the MNAM is not changed. The augmentation network provides a better conditioning for the I-FFT operation at the expense of increased error due to finite numerical accuracy. This is because it prevents the impedance of a port to go to infinity. Ideally, the effect of the augmentation would be exactly compensated. The resistive

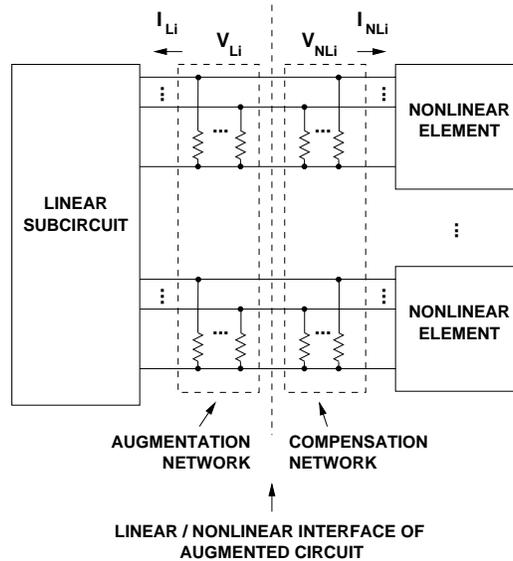


Figure 4.1: Augmentation and compensation network

augmentation network also serves to limit the extent of the impulse response as energy is absorbed and multiple reflections are damped. Again, note this effect is fully compensated.

Before converting the frequency domain \mathbf{M}_{SV} (impedance-like) matrix to the time domain, the imaginary part of the frequency response needs to be band-limited so that the function has a periodic (or circular) frequency response. This significantly reduces the aliasing errors in the I-FFT operation.

In Transim, frequency response limiting is implemented by filtering the last part of the frequency spectrum of each matrix element so the imaginary part goes to zero and the real part goes to the DC value. In this way, the frequency response is periodic. This is in contrast to low pass filtering [17] which zeroes the high frequency response and introducing additional time-delay to take the imaginary response to zero [49]. The filtering factor k_l is

$$k_l = \left(\frac{n-l}{n-l_0} \right)^2$$

where n is the number of discrete frequencies, l is the frequency index and l_0 is the frequency index at which filtering starts. For all l greater than l_0 , the filtering operation on each state variable impedance matrix entry is

$$\begin{aligned} \Re\{(\mathbf{M}_{SV})_l\} &= k_l \Re\{(\mathbf{M}_{SV})_l\} + (1 - k_l) (\mathbf{M}_{SV})_0 \\ \Im\{(\mathbf{M}_{SV})_l\} &= k_l \Im\{(\mathbf{M}_{SV})_l\} \end{aligned}$$

The amount of filtering is an analysis option since in some cases it is not necessary. By default, only the last 3% of the spectrum is filtered to achieve an acceptable alias reduction. If this is not done the correct impulse response is not assured as the continuity condition is inherent in the I-FFT (or FFT) operation.

Fourier transformation is implemented using a real I-FFT algorithm from the FFTW library [52]. This algorithm requires only the positive frequency samples since the negative frequency samples are the complex conjugate of the corresponding positive frequencies. The FFTW library is a comprehensive collection of C routines for computing the discrete Fourier transform in one or more dimensions, of both real and complex data, and of arbitrary input size.

4.4.2 Correction of the DC Error

The finite length discrete impulse response has inherent errors and this is particularly problematic in determining the dc response. The following procedure eliminates the dc error of the finite convolution by conveniently scaling the impulse response. The linear network is characterized by its transient response

$$R_i^{STEP} = \sum_{j=1}^{n_s} (\mathbf{m}_{DC})_{i,j}$$

and its dc response (which is known to be more accurate)

$$R_i^{DC} = \sum_{j=1}^{n_s} (\mathbf{M}_{SV}(0))_{i,j}$$

The dc error is characterized by an absolute error

$$E_i = R_i^{DC} - R_i^{STEP}$$

The dc error of the time domain solution can be corrected by multiplying each row i of impulse responses of \mathbf{m}_{SV} by the following factor

$$f_{Ei} = \frac{R_i^{DC}}{R_i^{STEP}}$$

which is ideally 1. When the dc response R_i is zero or close to zero a correction would be erroneous. The importance is taken as R_i^{DC} relative to E_i . Now the error correction algorithm is for each i

$$\begin{array}{ll} \text{if } |R_i^{DC}| > |E_i| & (\mathbf{m}_{SV}(n_t))_{i,j} = f_{Ei}(\mathbf{m}_{SV}(n_t))_{i,j} \\ \text{else if } R_i^{DC} \neq 0 & (\mathbf{m}_{SV}(n_t))_{i,j} = (\mathbf{m}_{SV}(n_t))_{i,j} + E_i/(n_s N_T) \\ \text{else} & \text{do nothing} \end{array}$$

4.4.3 Convolution

Convolution of the impulse response of the linear circuit with the outputs of the nonlinear elements has been proposed for transient analysis of distributed circuits with [15–18, 49, 50, 53]. The major problem identified with this type of analysis is the rapid growth in computation and memory requirements. The convolution integral, which becomes a convolution sum in computer implementations, is $O(n_{MAX}^2 n_s^2)$ where n_{MAX} is the length of the impulse response and n_s is the number of state variables. Memory usage is $O(n_{MAX} n_s^2)$, principally due to storage of the matrix impulse response. Thus the approach adopted here of reducing the number of state variables, n_s , (i.e., using the minimum number of state variables rather than the voltages at all nodes of the nonlinear network) dramatically reduces memory and compute time. As state variables permit the use of parameterized device models, the stability of the convolution analysis is improved allowing larger time steps thus reducing n_{MAX} .

4.5 Software Implementation

Equation (4.5) combined either with equation (4.4) or (4.7) resulted in a standard algebraic nonlinear problem at each time step that can be solved using a Newton method or quasi-Newton method. As the error function changes only slightly from time step to time step, efficient iterative matrix solve schemes can be used as a very good preconditioner is available from the previous time step. The general flow of the analysis is shown in Figure 4.2 and was implemented in the Transim program. After the netlist is parsed,

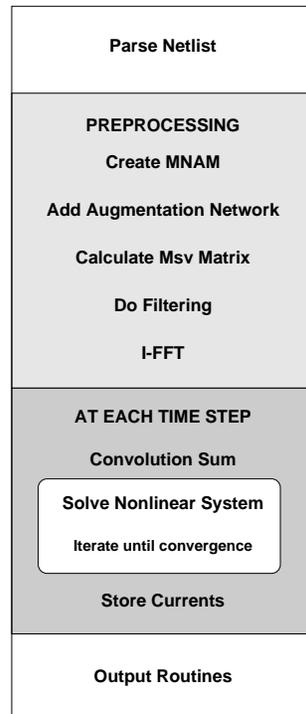


Figure 4.2: Flow diagram of the analysis code

the frequency-domain MNAM of the linear network is created at a set of frequencies given as a netlist parameter. The augmentation network is added to the MNAMs. After that, a set of \mathbf{M}_{SV} matrices (one at each frequency) is created from the augmented MNAMs. The \mathbf{M}_{SV} are filtered and converted via I-FFT into an impulse response matrix, \mathbf{m}_{SV} . If dc correction is desired, it is performed at this time. The calculation of each transient time step begins. At each time step, the convolution sum (summation terms of

Equations (4.4) or (4.7), depending on the initial conditions) is performed first and then the nonlinear algebraic system of equations (4.5) is solved. The nonlinear currents \mathbf{i}_{NL} are stored to be used in the convolution for the next step. Once all the time steps are calculated, the requested currents and voltages are saved in files.

4.6 Nonlinear Transmission Line

A nonlinear transmission line is regarded by many in the field as an extreme test of the performance of transient and steady-state simulators. Nonlinear transmission lines (NLTLs) find applications in a variety of high speed, wide bandwidth systems including picosecond resolution sampling circuits, laser and switching diode drivers, test waveform generators, and mm-wave sources [54]. They have three fundamental characteristics: nonlinearity, dispersion and dissipation. The NLTL consist of coplanar waveguides (CPWs) periodically loaded with reverse biased Schottky diodes. Diode-based NLTL used for pulse generation are extremely nonlinear circuits and are being used to test the robustness of circuit simulators. The NLTL considered here was designed with a balance between the nonlinearity of the loaded nonlinear elements and the dispersion of the periodic structure which results in the formation of a stable soliton [55, 56]. The nonlinearity of NLTLs is principally due to the voltage dependent capacitance of the diodes and the dissipation is due to the conductor losses in the CPWs.

In this work, the NLTL was modeled using generic transmission lines with frequency-dependent loss and Schottky diodes [57]. Skin effect was taken into account in the modeling of the transmission lines. The NLTL model is shown in Figure 4.3 and is excited by a 9 GHz sinusoid. The NLTL was designed for

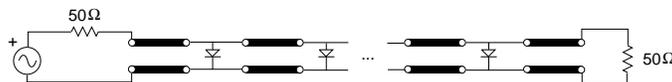


Figure 4.3: Model of the nonlinear transmission line.

a 24 GHz initial Bragg frequency, 225 GHz final Bragg frequency, 0.952097 tapering rule, and 120 ps total compression. It contains 48 sections of CPW transmission lines and 47 diodes. The drive is a 27 dBm sine wave with -3 V dc bias.

4.7 Simulation Results and Discussion

Fig. 4.4 shows the transient response of the soliton line simulated using Transim, including frequency dependent (skin effect) loss of the transmission lines. The simulation time was 3 hours on an Pentium III Xeon workstation, clocked at 550 MHz. The number of frequencies used to obtain the impulse response was 32768 and the time step was 0.04ps. The value of the compensation resistor was 120Ω . The comparison among the measured output voltage

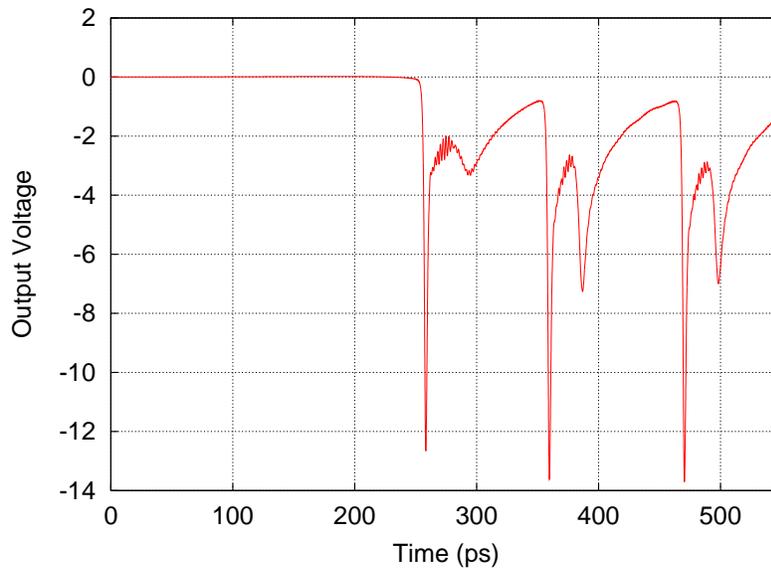


Figure 4.4: Complete transient response for the soliton line.

across the load [54] and transient and harmonic balance simulation (also using Transim) is shown in Fig. 4.5. The harmonic balance analysis used the state variable approach described in [2], using 100 frequencies. The prediction of the main soliton amplitude is correct, although the impulse width is slightly smaller in the simulations. Note that the secondary soliton predicted by the simulations is not observed in the measurements. This is probably due to neglecting higher order parasitic effects of the interconnections. In both simulations, a parasitic inductance of 21.8 pH modeled the connection between each diode and the CPW line.

Other harmonic balance simulation results, using 40 frequencies instead of 100, are shown in Figs. 4.6 and 4.7. Figure 4.7 shows how the harmonic

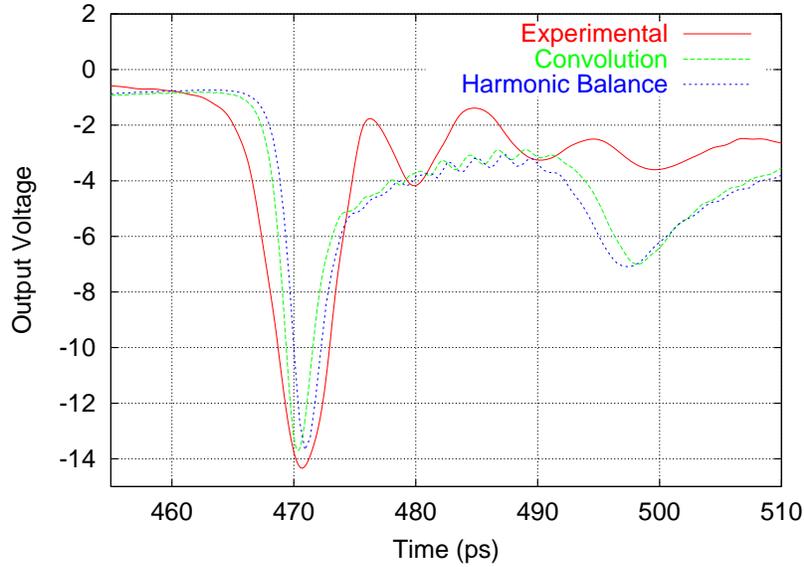


Figure 4.5: Comparison between experimental data and simulations.

content increases towards the end of the line. The convergence rate was increased and the number of harmonics reduced using the filtering technique described in [58], starting at harmonic number 30. The total number of frequencies was 40, and the simulation time was 30 hours on an UltraSparc 1.

In Fig. 4.8 the effect of frequency dependent transmission line attenuation, principally due to the skin effect, is compared to frequency-independent attenuation, an RLGC discrete model and no attenuation. The frequency independent attenuation of the transmission lines is the attenuation at 10 GHz. Frequency dependent attenuation has only a small effect on the depth and width of the soliton. The importance of modeling is borne out by these results. Rather than focusing on the previously deleterious effect of frequency dependent loss, better modeling of the parasitics of the NLTL is required to provide a better basis for computer aided design.

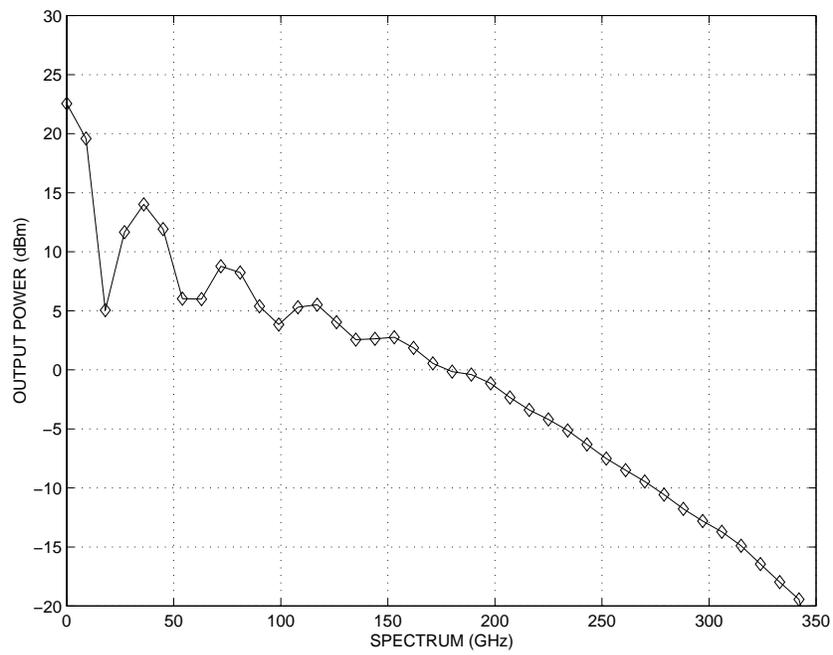


Figure 4.6: Magnitude of the harmonics of the output power.

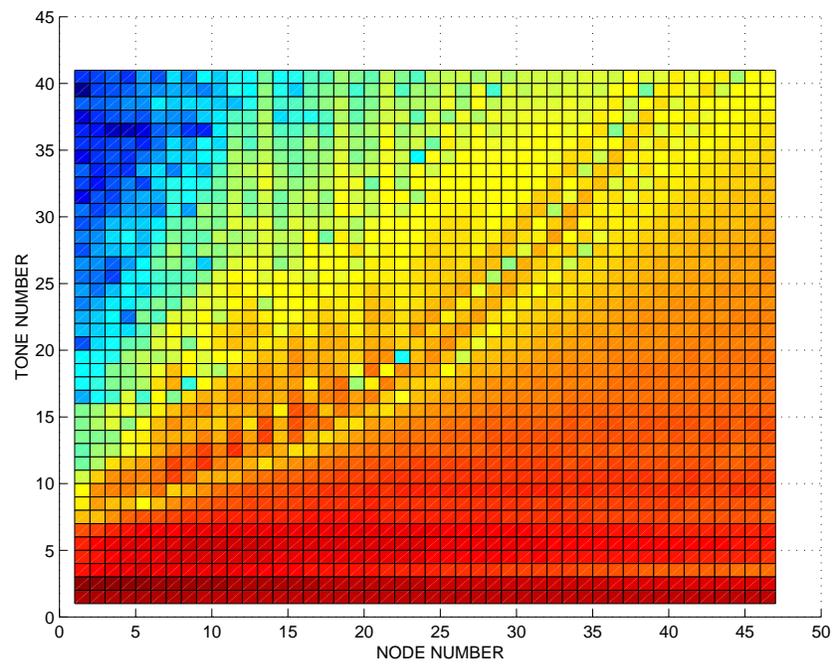


Figure 4.7: Magnitude of the harmonics of the output power along the line. Blue is low and red is high power.

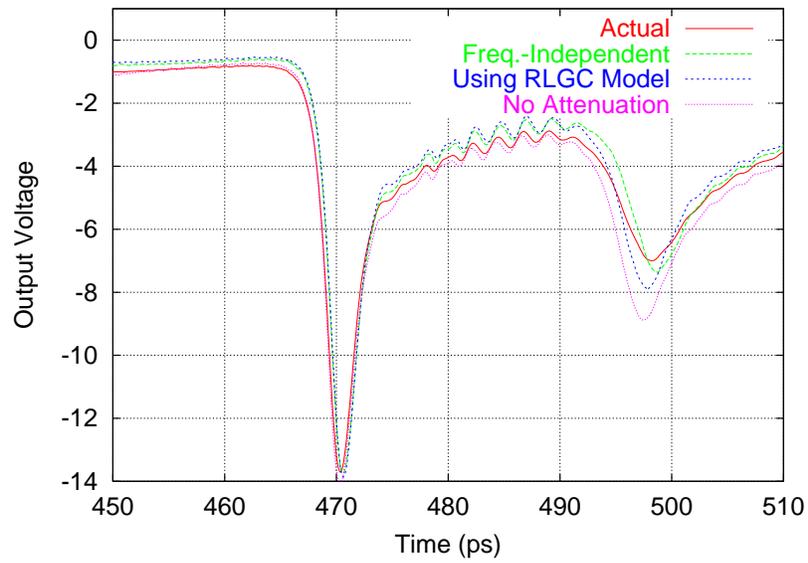


Figure 4.8: Comparison between simulations using different models for the transmission lines.

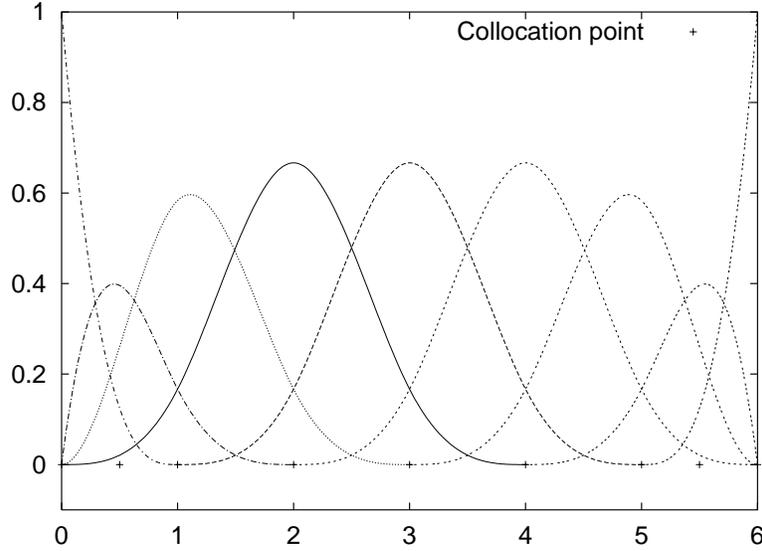
Chapter 5

Wavelet Transient Analysis

This chapter is dedicated to the development of a state-variable-based circuit transient analysis using wavelets. After a short introduction, section 5.2 presents the formulation of the equations of the transient analysis. Section 5.3 shows the implementation of the analysis in Transim. Finally, simulation results and a discussion is given in section 5.4.

5.1 Introduction

Multiresolution analysis has generated considerable excitement in the engineering community because of its potential to efficiently model large systems with an overall coarse response but fine behavior in some regions. Wavelet basis functions are ideally suited to expanding such a response as higher order and more localized basis functions can be concentrated on the regions where the response varies rapidly. Multiresolution analysis has been used with a wide variety of modeling problems including signal processing and electromagnetics. It is important to know where wavelet analysis is applicable as this guides future development. This chapter presents, for the first time, wavelet-based transient analysis incorporated in a general purpose circuit simulator. In circuits, voltage and current changes vary with time and location (*e.g.* node index) and so they can be modeled with few state variables by using variable resolution. In contrast, in conventional transient simulation the same fine time step is used at every node. The state variable formulation to analyze nonlinear circuits presented here can be used along with many transformations, in particular using wavelets.

Figure 5.1: Scaling functions for $L = 6$.

5.2 Reduced Error Function Formulation

In the wavelet collocation method [36] the unknown function x is expanded in a wavelet series which must fit the circuit response at a number of collocation points. The scaling and wavelet functions used in this work are shown in Figs. 5.1 and 5.2, respectively, and are used to model the response over a time interval. In this section, we will develop a set of equations combining the generalized state variable approach of Chapter 3 with the wavelet collocation method presented in [36] and reviewed in this work in section 2.2.3.

5.2.1 System Expansion

Wavelets are introduced by considering the function $g(t)$ defined in I . The following square matrices \mathbf{W}_J and \mathbf{W}'_J can be defined:

$$\mathbf{g} = \mathbf{W}_J \hat{\mathbf{g}}_J, \quad \mathbf{g}' = \mathbf{W}'_J \hat{\mathbf{g}}_J \quad (5.1)$$

where \mathbf{g} , \mathbf{g}' are vectors whose elements are the function and derivatives values, respectively, at the collocation points and $\hat{\mathbf{g}}_J$ is the vector of the corresponding coefficients. J is the maximum subspace level being considered.

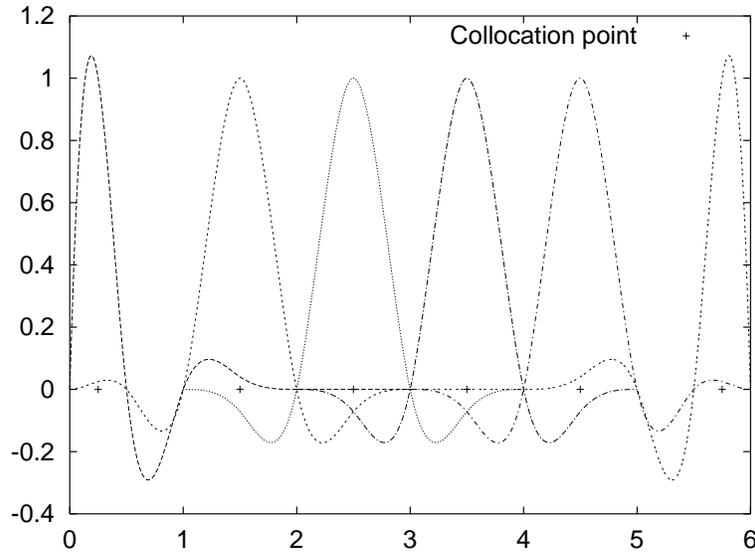


Figure 5.2: Wavelet functions in W_0 for $L = 6$.

Figure 5.3 shows the nonzero structure of \mathbf{W}_J and \mathbf{W}'_J for the particular wavelet basis functions used in this work.

The matrices \mathbf{G} and \mathbf{C} in Equation (3.7) are now expanded using \mathbf{W}_J and \mathbf{W}'_J , respectively, reduced by removing the first row. The term ‘expanded’ here means that each element in the original matrix is replaced by the product of the element and the expanding matrix. The size of the resulting matrix is the product of the sizes of the original matrix and the expanding matrix. Additional equations are obtained by noting that the product of the coefficients of each electrical variable times the first row of \mathbf{W}_J is equal to the initial condition for that variable. The extra rows from these equations plus the sum of the expanded matrices result in a sparse matrix \mathbf{M}_J , which is invertible for ‘normal’ circuits.

The wavelet formulation of the source vector $\mathbf{s}_J = \mathbf{s}_{f,J} + \mathbf{s}_{v,J}$ is obtained by expanding each element of $\mathbf{s}_{f,J}$ into the set of time samples corresponding to the collocation points. The first time sample of the source vector is replaced by the corresponding initial value. The matrix \mathbf{T} (in (3.7)) must also be expanded by replacing each $+1$ by \mathbf{W}_J , and each -1 by $-\mathbf{W}_J$ (in both cases \mathbf{W}_J is reduced by removing the first column and the first row is replaced by zeros). This matrix is denoted $\mathbf{T}_{2,J}$. The transpose of \mathbf{T} is similarly

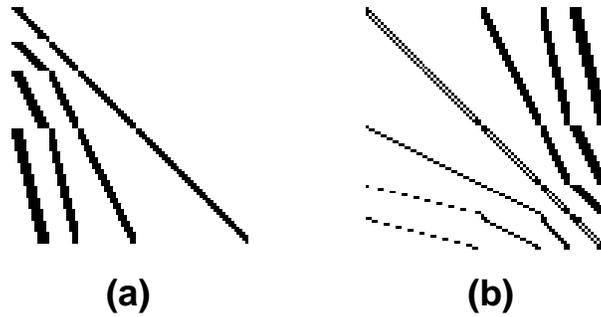


Figure 5.3: Representation of the nonzero elements of the transformation matrices for $L = 10$ and $J = 2$: (a) \mathbf{W} and (b) \mathbf{W}' .

expanded by replacing each $+1$ by a matrix \mathbf{I}_r , and each -1 by $-\mathbf{I}_r$, where \mathbf{I}_r is an identity matrix of size $m \times m$ reduced by removing the first row. This matrix is denoted $\mathbf{T}_{1,J}$. The final linear circuit equation is

$$\mathbf{M}_J \hat{\mathbf{u}}_J = \mathbf{s}_{f,J} + \mathbf{T}_{1,J} \mathbf{i}_{NL,J}(\hat{\mathbf{x}}_J) \tag{5.2}$$

where $\hat{\mathbf{u}}_J$ is the vector of the wavelet coefficients of the unknown circuit variables. Figure 5.4 shows the structure of this linear system

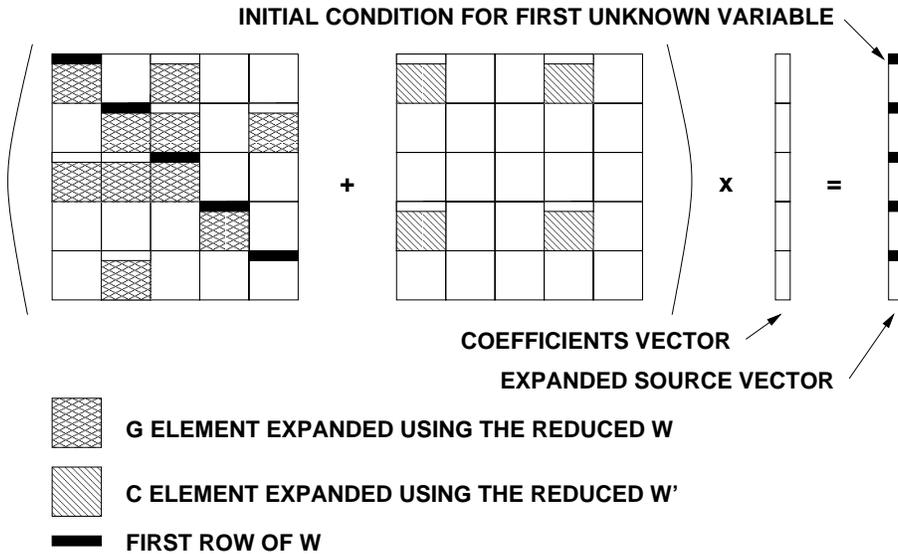


Figure 5.4: Linear system construction.

Let \mathbf{x}_J be the state variable vector (with the state variable concept as

defined in equations (3.3) and 3.4) at all collocation points and $\hat{\mathbf{x}}_J$ the corresponding vector of coefficients in the transform domain. The first transform coefficient is excluded from the unknowns since it can be derived from the initial condition. Then we denote by $\mathbf{v}_{NL,J}(\hat{\mathbf{x}}_J)$ and $\mathbf{i}_{NL,J}(\hat{\mathbf{x}}_J)$ the vectors of voltages and currents at the ports of the nonlinear devices at all the collocation points but the first. The error function $\mathbf{F}(\hat{\mathbf{x}}_J)$ is then, expanding (3.8)

$$\mathbf{F}(\hat{\mathbf{x}}_J) = \mathbf{T}_{2,J}\hat{\mathbf{u}}_J - \mathbf{v}_{NL,J}(\hat{\mathbf{x}}_J) = 0$$

Combining the preceding with Equation (5.2),

$$\begin{aligned} \mathbf{F}(\hat{\mathbf{x}}_J) &= \mathbf{T}_{2,J}\mathbf{M}_J^{-1}\mathbf{s}_{f,J} \\ &\quad + \mathbf{T}_{2,J}\mathbf{M}_J^{-1}\mathbf{T}_{1,J}\mathbf{i}_{NL,J}(\hat{\mathbf{x}}_J) - \mathbf{v}_{NL,J}(\hat{\mathbf{x}}_J) \end{aligned}$$

which can be expressed as

$$\mathbf{F}(\hat{\mathbf{x}}_J) = \mathbf{s}_{sv,J} + \mathbf{M}_{sv,J}\mathbf{i}_{NL,J}(\hat{\mathbf{x}}_J) - \mathbf{v}_{NL,J}(\hat{\mathbf{x}}_J) = 0 \quad (5.3)$$

Here $\mathbf{s}_{sv,J}$ is the compressed source vector (the initial conditions of the entire linear subcircuit are embedded in it) and $\mathbf{M}_{sv,J}$ is the compressed impedance matrix. They are defined as

$$\begin{aligned} \mathbf{s}_{sv,J} &= \mathbf{T}_{2,J}\mathbf{M}_J^{-1}\mathbf{s}_{f,J} \\ \mathbf{M}_{sv,J} &= \mathbf{T}_{2,J}\mathbf{M}_J^{-1}\mathbf{T}_{1,J} \end{aligned}$$

The system of nonlinear algebraic equations (5.3) is solved using globally convergent quasi-Newton methods. The size of $\mathbf{M}_{sv,J}$ is $(m-1)n_s \times (m-1)n_s$, where m is the number of collocation points.

This wavelet transient formulation (Equation (5.3)) could easily be modified to produce a formulation to find the periodic steady-state of a circuit. The only modification is that the equations relating the initial conditions are replaced by boundary condition equations. But this work will focus on the transient problem.

The nonlinear error function can be expressed in terms of the wavelet coefficients of the port voltages. This yields a similar error function where $\mathbf{v}_{NL,J}(\hat{\mathbf{x}}_J)$ must be transformed from the physical to the coefficient space (using \mathbf{W}_J^{-1}). At first, this approach would seem to be less efficient, since the resulting compressed matrix is not sparse in general and it requires the implementation of ‘inverse’ transformation. Nevertheless, this approach would allow the reduction of the linear system size if some coefficients of the port voltages are known to be zero, which can not be done in the physical space. This alternative will not be developed here.

5.2.2 Initial Conditions in the State Variables

For each state variable, the wavelet coefficients are not completely independent. There is a constraint imposed by the initial condition. Therefore, the first transform coefficient is excluded from the unknowns. Given the initial condition x_0 and the remaining coefficients $\hat{\mathbf{x}}$, it is possible to obtain the rest of the samples \mathbf{x} as follows

$$\mathbf{x} = (\mathbf{W}_r - \frac{\mathbf{w}_{c0}}{w_{0,0}}\mathbf{w}_{r0})\hat{\mathbf{x}} + \frac{\mathbf{w}_{c0}}{w_{0,0}}x_0 \quad (5.4)$$

where \mathbf{W}_r is equal to \mathbf{W} reduced by the first row and column, \mathbf{w}_{c0} and \mathbf{w}_{r0} are the first column and row of \mathbf{W} , respectively, excluding the first element $w_{0,0}$.

A similar expression can be obtained for $\dot{\mathbf{x}}$, namely

$$\dot{\mathbf{x}} = (\mathbf{W}'_r - \frac{\mathbf{w}'_{c0}}{w_{0,0}}\mathbf{w}_{r0})\hat{\mathbf{x}} + \frac{\mathbf{w}'_{c0}}{w_{0,0}}x_0 \quad (5.5)$$

Higher order derivatives were not used in the present work.

5.2.3 Solution of the Nonlinear System Considerations

If the time interval to be simulated requires many collocation points, the nonlinear system to be solved becomes very large. One way to overcome this problem is to divide the total simulation time interval into smaller *windows*. Then solve one time window at a time. The final time sample at each window becomes the initial condition for the next and the method is applied for all windows. The approach becomes thus an hybrid between collocation and time marching methods.

The nonlinear system of equations in Equation (5.3) can be solved using different numerical methods. Zhou *et al.* [34,37] used a fixed point scheme. We have chosen to use Newton and quasi-Newton methods because of the higher convergence rate and the global convergence properties of some variants such as the line search method [59].

There are two main issues in solving the nonlinear system. Obtaining a good initial guess and reducing the number of unknowns.

If the circuit being simulated has a periodic excitation, the time window size can be chosen equal to the period. Then the solution for a given time window can be used as a good guess for the solution at the next window.

Unfortunately, for some circuits this would imply a time window too large to be handled efficiently.

An adaptive scheme could be used to somewhat reduce the number of unknowns. It involves solving a nonlinear system several times with increasing resolution. When some coefficients are detected to be smaller than a given tolerance, they are discarded from the set of unknowns, and the resolution is only increased where the corresponding coefficients are significant. The problem with this approach is that it greatly increases the implementation complexity and it only attenuates the problem of having to solve for many unknowns at a time.

5.2.4 Other Transformation Types

The formulation in Equation (5.3) is quite general, and can be applied not only with wavelet transformation, but with other transformations as well. As an example, when using the following set of matrices,

$$\mathbf{W} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

and

$$\mathbf{W}' = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

the formulation becomes backward Euler method. In this case the linear system to be solved is twice as large as the original MNAM (because the transformations matrices are 2×2) and the size of the nonlinear system resulting from Equation (5.3) is equal to the number of state variables. Multi-step time-marching schemes and harmonic balance can also be implemented by introducing some variations in the formulation (the initial conditions considerations do not apply for harmonic balance). However, it is in general not practical to use the formulation of Equation (5.3) in those cases because simpler and more efficient formulations exist as it is shown in Chapter 6 in this work.

5.2.5 Resolution Limits

Figure 5.6 shows that the increase in the number of time samples (by increasing J) in the interval produces an increase in the condition number of

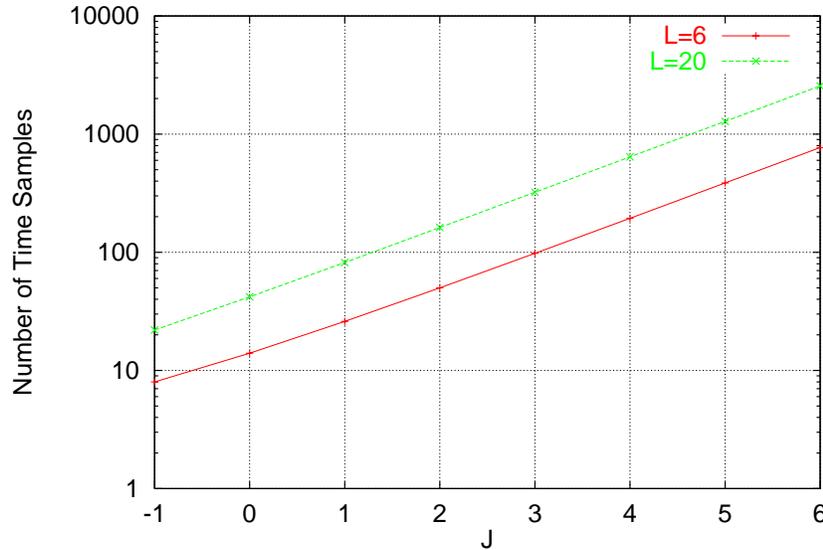


Figure 5.5: Number of time samples in the interval as a function of J for a small circuit.

\mathbf{M}_J . The corresponding number of time samples in the interval are shown in Figure 5.5. This sets an upper limit in the maximum level J that can be practically be used, because there is some value where the corresponding matrix \mathbf{M}_J is so ill-conditioned that it can not be factored reliably.

5.3 Implementation

The complexity of the state variable wavelet transient requires a more elaborate implementation with several classes in our circuit simulator, Transim. The main analysis class (**WavTran**) and related classes are shown in the diagram in Fig. 5.7. For a brief explanation of the class diagram syntax, see Appendix B. More details about the architecture of Transim are given in Chapter 7. This is a modular design. Each component can eventually be replaced by alternative implementations. All the details about the wavelet base and collocation method used are encapsulated inside the **WaveletBase** class. In this way, the rest of the code is independent of the particular choice of wavelet base. As an example of the flexibility of this approach, the normal **WaveletBase** class code implementing Cai's [35] wavelets was replaced by

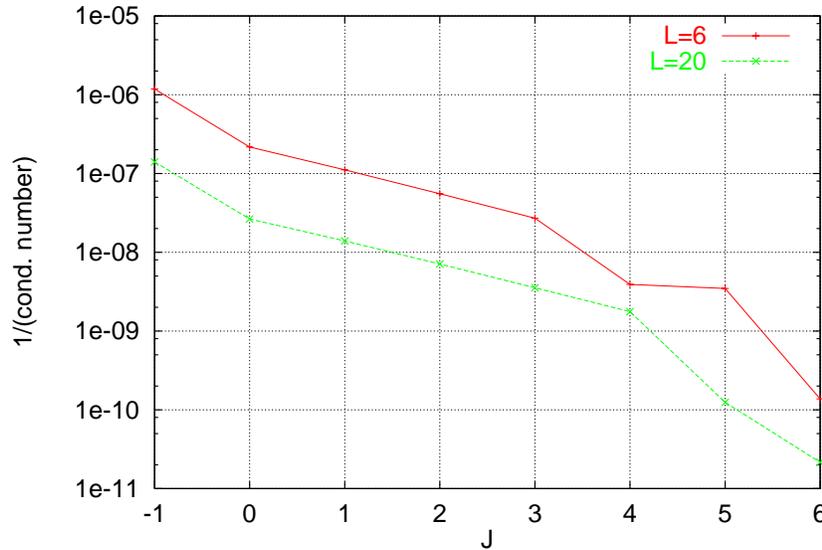


Figure 5.6: Reciprocal of the condition number of \mathbf{M}_J in the interval as a function of J for a small circuit.

another **WaveletBase** implementation with the backward Euler matrices of section 5.2.4. After recompilation of Transim, the BE simulation results shown in section 5.4 were produced. No modifications were needed in the other components of Transim, so in this aspect the design was successful.

The **WTLinear** class encapsulates everything related with the linear part of the solution. The SuperLU library is used by this class to solve the linear system. **WTNonLinear** is a class used to solve the nonlinear system for one time window. It uses the existing nonlinear solver routines integrated in Transim to find the solution of the equations. There is a need to share data between the linear/nonlinear modules. That is the reason for the **WTDBase** class. Its function is to store the result of calculations. The **WavTran** class is the top-level analysis class that controls the analysis flow. The original idea was that **WavTran** could create several instances of **WTLinear** and **WTNonLinear** according to the desired resolution level. Currently, the resolution level is fixed at the starting of the calculation.

The current implementation supports time windows of variable size. Then simulation starts with a given resolution and time window size. If there is no convergence, the time window is divided by two and a new calculation is

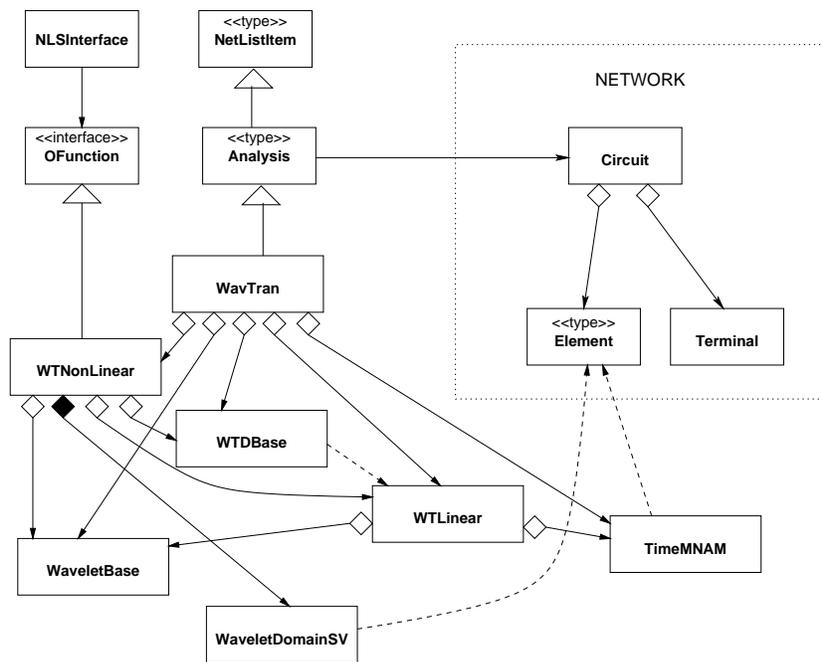


Figure 5.7: Class diagram for the wavelet-transient-based analysis in Transim.

attempted. This is usually better than increasing the resolution level because in this way the number of simultaneous unknowns remains constant and the overall simulation speed is faster.

5.4 Simulation Results and Discussion

Consider the modeling of the 47-section nonlinear transmission line described in Section 4.6. Since the method considered here is a time domain method, the transmission lines are modeled using RLGC sections. The attenuation factor in the following wavelet and Spice simulations is thus frequency-independent.

Figures 5.8 and 5.9 compare the results using wavelet transient, convolution and Spice3f5 simulations. The differences are greater at the last diode (diode 47) before the load (Fig. 5.9). The small difference in the response between wavelets and Spice is due to slightly different treatments of the diode model in Transim and Spice. A large number of time windows (see [37]) was needed to reduce the number of unknowns to be solved simultaneously. Since the results using wavelets and Spice are very close, we can conclude the wavelet analysis (at least in this simulation) behaves like a time-marching scheme, possibly due to fact that the circuit was analyzed using small time intervals at a time. There are clear tradeoffs involved. In wavelet transient analysis the error is minimized over a time interval and there are many more unknowns than when the error is minimized at a single time point. However since error is minimized over a range and because of the $O(h^4)$ convergence rate of the wavelet basis used, where h is the time step, there are generally many fewer time points than required in a Spice-like analysis. Fig. 5.10 shows the wavelet coefficients for the state variable of the last diode. Many of these are close to zero, and could be removed from the calculation using an adaptive scheme.

More development is needed before this method reaches its full potential. In particular dynamic variation of resolution including variable resolution at different circuit nodes are required. However it is unlikely that this kind of wavelet formulation can be made faster than conventional time marching methods, because this method requires to solve many more simultaneous unknowns.

It is possible to formulate this problem without the reduction using an approach similar to associate discrete modeling [3] and keep in this way the

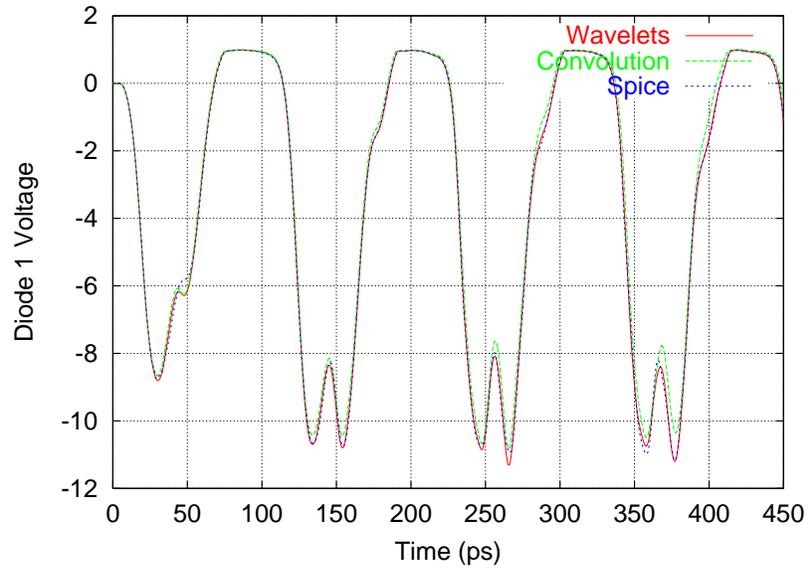


Figure 5.8: Comparison of the voltage of the diode close to the generator (diode 1).

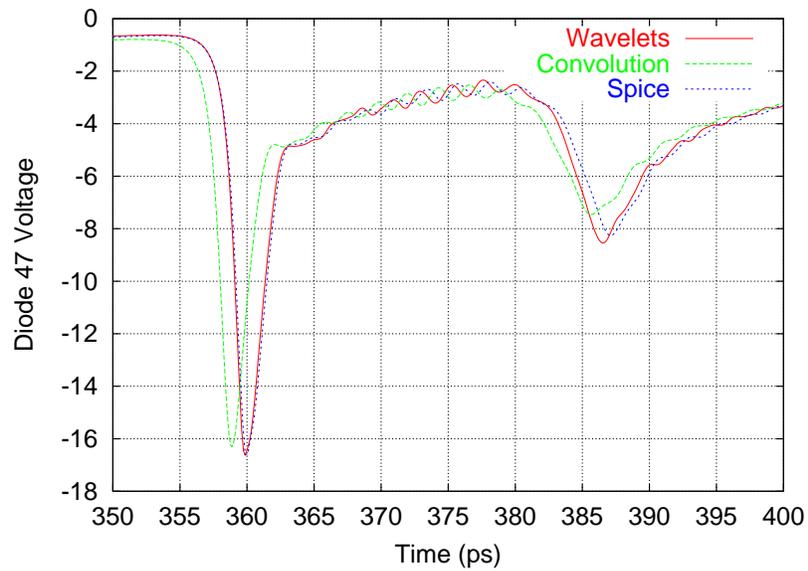


Figure 5.9: Comparison of the voltage of the diode close to the load (diode 47) of the nonlinear transmission line.

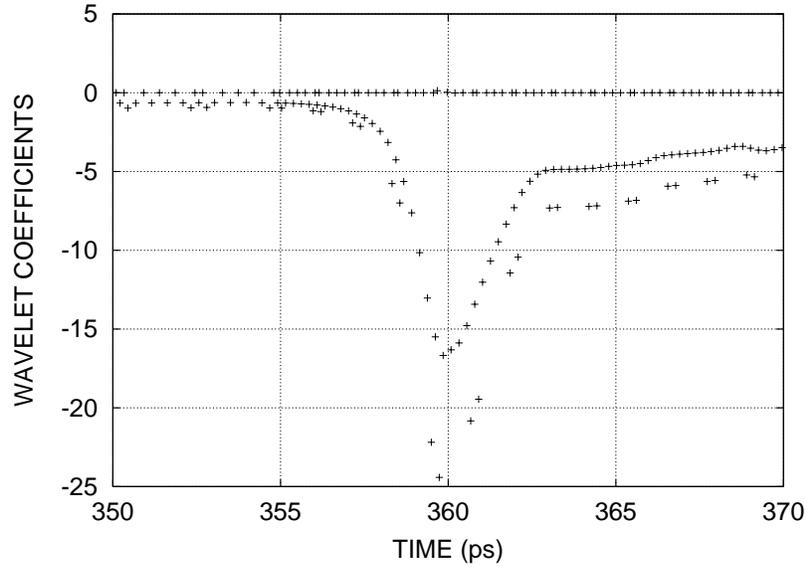


Figure 5.10: Coefficients of the state variable of diode 47.

sparsity of the original system (5.2). However the disadvantage of having to find many simultaneous unknowns would still remain. This is left for future research.

5.5 Summary

A wavelet based transient analysis was implemented in a circuit simulator for the first time. The analysis formulation is based on the state variables of the nonlinear devices and therefore the resulting nonlinear system of algebraic equations is much smaller than the system resulting from the wavelet formulations in the literature.

The formulation described in this chapter is very general and does not depend on a particular type of wavelet basis function. It was shown that our formulation can be used with other types of transformations to produce other circuit analysis types.

Chapter 6

Time Marching Transient Analysis

This chapter develops the equations and describes the software implementation for a state variable transient analysis based on time marching integration methods. After the introduction, section 6.2 presents the formulation of this transient analysis. Then, the implementation of this transient analysis in Transim is commented in Section 6.3. Finally results and discussion of the simulation of the nonlinear transmission line described in Section 4.6 are given Section 6.4.

6.1 Introduction

The simulation of the NLTL (Section 4.6) using the wavelet-based transient analysis of Chapter 5 took several hours of computer time. In contrast, when Transim was recompiled using the matrices to implement the backward Euler integration method (Section 5.2.4) the simulation time of the same circuit dropped to little more than 10 minutes. Given the this promising result, and since neither the wavelet transient formulation nor its software implementation was optimized for time marching methods, we decided to develop an efficient state-variable transient analysis based on general time marching integration methods.

Time marching methods have been used in circuit simulation for a long time. What is new here is that the formulation presented in this chapter is state variable based and performs a reduction in the size of the nonlinear

algebraic system of equations resulting from the discretization of the circuit equations. In some circuits, the application of this new formulation results in a dramatic reduction of the total simulation time.

6.2 Reduced Error Function Formulation

The equations for the transient analysis are derived from the equations in Chapter 3. The basic idea is to convert the differential equations in an algebraic system of nonlinear equations using time marching integration methods.

First, Equations (3.3) and (3.4) are expressed using the discretized time

$$\mathbf{v}_{NL}(\mathbf{x}_n) = u[\mathbf{x}_n, \mathbf{x}'_n, \dots, \mathbf{x}_n^{(m)}, \mathbf{x}_{D,n}] \quad (6.1)$$

$$\mathbf{i}_{NL}(\mathbf{x}_n) = w[\mathbf{x}_n, \mathbf{x}'_n, \dots, \mathbf{x}_n^{(m)}, \mathbf{x}_{D,n}] \quad (6.2)$$

where $\mathbf{x}_n = \mathbf{x}(t_n)$, $\mathbf{x}'_n = \mathbf{x}'(t_n)$, $(\mathbf{x}_{D,n})_i = \mathbf{x}_i(t_n - \tau_i)$, and t_n is the current time. Discretization to Equation (3.7) yields

$$\mathbf{G}\mathbf{u}_n + \mathbf{C}\mathbf{u}'_n = \mathbf{s}_{f,n} + \mathbf{T}^T \mathbf{i}_{NL}(\mathbf{x}_n) \quad (6.3)$$

The time marching integration approximation is introduced by applying Equation (2.30) to calculate the \mathbf{u}'_n vector

$$\mathbf{u}'_n = a\mathbf{u}_n + \mathbf{b}_{n-1}$$

where \mathbf{b}_{n-1} is a vector of the same dimension as \mathbf{u}_n . Replacing \mathbf{u}'_n in (6.3),

$$\mathbf{G}\mathbf{u}_n + \mathbf{C}[a\mathbf{u}_n + \mathbf{b}_{n-1}] = \mathbf{s}_{f,n} + \mathbf{T}^T \mathbf{i}_{NL}(\mathbf{x}_n)$$

Now we can solve for \mathbf{u}_n ,

$$\mathbf{u}_n = [\mathbf{G} + a\mathbf{C}]^{-1}[\mathbf{s}_{f,n} - \mathbf{b}_{n-1} + \mathbf{T}^T \mathbf{i}_{NL}(\mathbf{x}_n)] \quad (6.4)$$

Finally, the error function $\mathbf{f}(\mathbf{x}_n)$ is obtained by discretizing Equation (3.8) and replacing \mathbf{u}_n from Equation (6.4),

$$\mathbf{f}(\mathbf{x}_n) = \mathbf{s}_{sv,n} + \mathbf{M}_{sv} \mathbf{i}_{NL}(\mathbf{x}_n) - \mathbf{v}_{NL}(\mathbf{x}_n) = 0 \quad (6.5)$$

where $\mathbf{s}_{sv,n}$ and \mathbf{M}_{sv} are defined as

$$\begin{aligned} \mathbf{s}_{sv,n} &= \mathbf{T}[\mathbf{G} + a\mathbf{C}]^{-1}[\mathbf{s}_{f,n} - \mathbf{C}\mathbf{b}_{n-1}] \\ \mathbf{M}_{sv} &= \mathbf{T}[\mathbf{G} + a\mathbf{C}]^{-1}\mathbf{T}^T \end{aligned}$$

Once again the size of the resulting algebraic system of nonlinear equations is $n_s \times n_s$. \mathbf{M}_{sv} is constant as long as the time step h is constant. $\mathbf{s}_{sv,n}$ changes at each time step. This is in contrast with traditional circuit simulators, where the number of simultaneous unknowns is equal to the number of non-reference nodes in the circuit plus the number of additional required variables (n_m). In many microwave circuits, $n_s \ll n_m$, therefore the convenience of this new formulation is evident.

6.3 Implementation

Equation (6.5) resulted in a standard algebraic nonlinear problem at each time step that can be solved using a Newton method or quasi-Newton method. As the error function changes only slightly from time step to time step, a very good guess of the unknowns is obtained from the previous time step solution.

The transient analysis resulting from Equation (6.5) was implemented in Transim. The interface with the nonlinear elements is the same used for the convolution transient (Chapter 4) analysis type, and the time-domain MNAM is handled by the same class as in the wavelet transient analysis (Chapter 5). Therefore, most of the necessary code was already implemented. Some improvements to the nonlinear element interface were made in order to increase the flexibility (*i.e.*, allow different integration methods at run time). These enhancements are therefore now available for the other analysis types.

The integration method is specified in the input netlist file at run time. Currently the backward Euler (BE) and the trapezoidal methods are available, but more methods can be added with no modifications to the main routines. Figure 6.1 shows the interface used for the integration methods. These are purely abstract classes. **NIntegMethod** is the interface class used (indirectly) to obtain the derivatives and delays in the nonlinear device routines. **LIntegMethod** is used in the main analysis routine to build $a\mathbf{C}$ and $\mathbf{s}_{f,n} - \mathbf{Cb}_{n-1}$, both of which depend on the integration method being used.

The standard Transim libraries were used. The resulting implementation is quite efficient as it will be shown in the comparison of run times with other analysis types and other simulators.

```
class NLIntegMethod
{
public:

    virtual double derivX(const int& index) = 0;
    virtual double getdx_dtFactor() = 0;
    virtual double delayX(const int& index, const double& t) = 0;
    virtual double getDelayXFactor(const double& t) = 0;
    virtual void store() {}
};

class LIntegMethod
{
public:

    virtual void buildMd(SparseMatrix& M, const double& h) = 0;
    virtual void buildSf(DenseVector& s1, const double& ctime) = 0;
    virtual void store() {}
};
```

Figure 6.1: Integration method interface.

6.4 Simulation Results and Discussion

The circuit of Section 4.6 is again used to test the transient analysis type developed in this chapter. Fig. 6.2 compares the voltage at the last diode obtained using BE and trapezoidal integration and the Spice simulation. The fixed time step used in Transim was .01 ps for BE and .1 ps for trapezoidal. Spice uses a variable time step. Note that even when using a very small time step, the numerical damping introduced by the BE formula attenuates the small oscillations in the waveform. Table 6.1 compares the simulation times

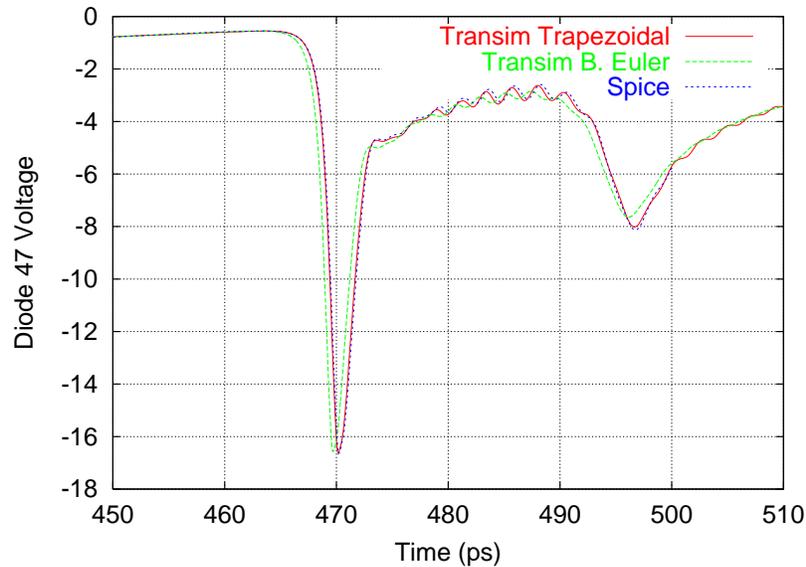


Figure 6.2: Comparison of the voltage of the last diode.

of Transim using trapezoidal integration and Spice, running in a Pentium III Xeon clocked at 550MHz. Transim is always faster. The speedup is greater when the simulated time is longer in this circuit because Transim does not yet implement variable time step and thus can not use long time steps at the beginning of the simulation. Note that the speedup is significant even when Transim is internally using a high level of abstraction which introduces some overhead, such as automatic differentiation, run-time decision of the integration method and general-purpose math libraries. The reduction in the computation can be explained as follows. Traditional circuit simulators such as Spice, must perform a big sparse matrix (size $n_m \times n_m$) decomposition at

Table 6.1: Comparison of simulation times for the soliton line

Simulated Time (ns)	Transim Time (s)	Spice Time (s)	Speedup
.55	57	249	4.3
1.5	157	2596	16.5

each iteration of the method to solve nonlinear equations [3]. In contrast, the state variable based time marching transient requires the factorization of matrix $\mathbf{G} + a\mathbf{C}$ (also of size $n_m \times n_m$) only when the h is modified. The nonlinear system of equations in the state variable based time marching transient is solved using quasi-Newton methods. Therefore, a Jacobian (here called the *reduced Jacobian*) of size $n_s \times n_s$ must be factored every a few nonlinear iterations, but this is not expensive if n_s is not large (say, $n_s < 500$). Thus, the state variable based time marching transient seems to be ideal for up to medium-sized microwave circuits (up to a few hundred nonlinear devices) with many linear devices so $n_s \ll n_m$. The same accuracy of Spice is obtained (this is due to the definition of the equations) and all the circuit variables are available as simulation results.

However, when the circuit being analyzed is large, things change. The original MNAM was sparse, and the reduced Jacobian is not. Then, the time to decompose the MNAM is $O(c^2 n_m)$, where c is a factor typically around 10 for this kind of problem [60] and n_m is the size of the MNAM. On the other hand, factoring of the reduced Jacobian matrix takes $O(n_s^3)$. Based on this reasoning and the available simulation results, we conclude that, the state variable based time marching transient seems more efficient than the conventional Spice analysis if (1) n_s is considerably smaller than the number of nodes and (2) n_s is not more than a few hundred. The nonlinear transmission line analyzed here and many quasi-optical systems satisfy those requirements. But this method would not be convenient to analyze digital VLSI circuits, because typically none of the enumerated conditions are satisfied in this kind of circuit.

Chapter 7

Object Oriented Circuit Simulator

7.1 Introduction

The rapid rate of innovation of microwave and millimeter wave systems requires the development of an easily extensible and modifiable microwave computer aided engineering (CAE) environment. While great strides have been made in the flexibility of commercial CAE tools, these sometimes prove inadequate in modeling advanced systems. As with virtually all aspects of electronic engineering the abstraction level of RF and microwave theory and techniques has increased dramatically. In particular, large systems are being designed with attention given to the interaction of components at many levels. One of the most significant developments relevant to microwave computer aided engineering is the rise of object oriented (OO) design practice [61–66]¹. While it is normal to think of OO-specific programming languages as being the main technology for implementing OO design, good OO practice (with limitations) can be implemented in more conventional programming languages such as C. However OO-specific languages foster code reuse and have constructs that facilitate object manipulation. The OO abstraction is well suited to modeling electronic systems, for example, circuit elements are already viewed as discrete objects and at the same time as an integral part of a (circuit) continuum. The OO view is a unifying concept that maps extremely well onto the way humans perceive the world around them. Non-OO

¹These references are available on line at <http://www.objectmentor.com/>

circuit simulators always become complicated with many layers of special cases. Referring to circuit elements again, traditional simulation implementations have many “if-then” like statements and individually identify every element in many places for special handling.

Traditionally papers on advances in circuit simulation technology have presented algorithmic developments which usually resulted in more robust and speedier circuit modeling technologies. In some cases additional capability is presented in that strategies for simulating electronic systems that could not be modeled previously are presented. In contrast, what this chapter does more than anything else is present a level of abstraction higher than that previously reported for modeling microwave circuits and systems.

There are a few key premises that drove the work reported here. One of these is the adoption of a very strong OO paradigm throughout to obtain a modular design. Also, an integral part of the various high performance computing initiatives is the separation of the core components embodying numerical methods from the modeling and solver formulation process with the result that numerical techniques developed by computer scientists and mathematicians can be formulated using formal correctness procedures. Thus, what is adopted here, is that the circuit abstraction is adapted so that highly reliable and efficient pre-developed libraries can be used.

C++ was once considered slow for scientific applications. Advances in compilers and programming techniques, however, have made this language attractive and in some benchmarks C++ outperforms Fortran [67,68]. Several OO numerical libraries have been developed [69]. Of great importance to the work described here is the incorporation of the standard template library (STL) [70]. The STL is a C++ library of container classes, algorithms, and iterators; it provides many of the basic algorithms and data structures of computer science. The STL is a generic library, meaning that its components are heavily parameterized: almost every component in the STL is a template. The current ISO/ANSI C++ standard [71] has not been fully implemented and C++ compilers support a variable subset of the standard. The biggest areas of noncompliance being the templates and the standard library.

In this chapter the focus is on the design of the object-oriented structure of the program. The goal in design was to obtain speed in development, to use ‘off the shelf’ advanced numerical techniques, and to allow easy expansion and testing of new models and numerical methods.

The circuit simulator implementing the ideas presented here is called Transim. We present for the first time a circuit simulator using some of the

recently developed formal OO techniques and C++ features. The design intent was to combine the advantages of previous OO circuit simulators with these new developments as well as expanding capability. Transim uses C++ libraries [72, 73] and several written in C or Fortran [52, 59, 74]. In the following the specific OO programming construction used in the current work is described. Then an example is presented integrating electromagnetic and circuit analysis in modeling a microwave CPW active antenna.

7.2 The Network Package

The network package is the core of the simulator. All the elements and the analysis classes are built upon it as shown in the class diagram of Figure 7.1. (See Appendix 1 for a description of the class diagram syntax.) Following the suggestion made in [9], there is a **NetListItem** class that is the base for all classes of objects that appear in the input netlist. This is the base class that handles parameters. Figure 7.2 shows some of the methods provided by this class. All the netlist items share a common syntax so that the element model developer does not need to worry about the details of element parsing and there is no need to modify the parser to add new elements. For compatibility reasons, Spice-type syntax (which does not have a consistent grammar) is supported by the parser outside the network package.

The **Element** class contains basic methods common to all elements as well as the interface methods for the evaluation routines. Some of the methods of this class (Figure 7.3) need to be overridden by the derived classes. For example, in class **Diode**, **svTran()** is intended to contain the code to evaluate the time domain response of a diode. This function is used by DC and transient analyses. The same happens with **svHB()** and **fillMNAM()**. The overhead imposed by these virtual functions is small compared to the time spent evaluating the functions themselves and so this approach is a good compromise between flexibility and efficiency. This idea has been used in [8–10]. Transim also offers a more elaborate mechanism for nonlinear element evaluation functions which will be described in Section 7.4.

The **ElementManager** class is mainly responsible for keeping a catalog of all the existing elements. Note that this class is the only one that ‘knows’ about each and every type of element but this dependency is weak. The element list is included from an automatically generated file. **ElementManager** is also used to automatically generate the element catalog doc-

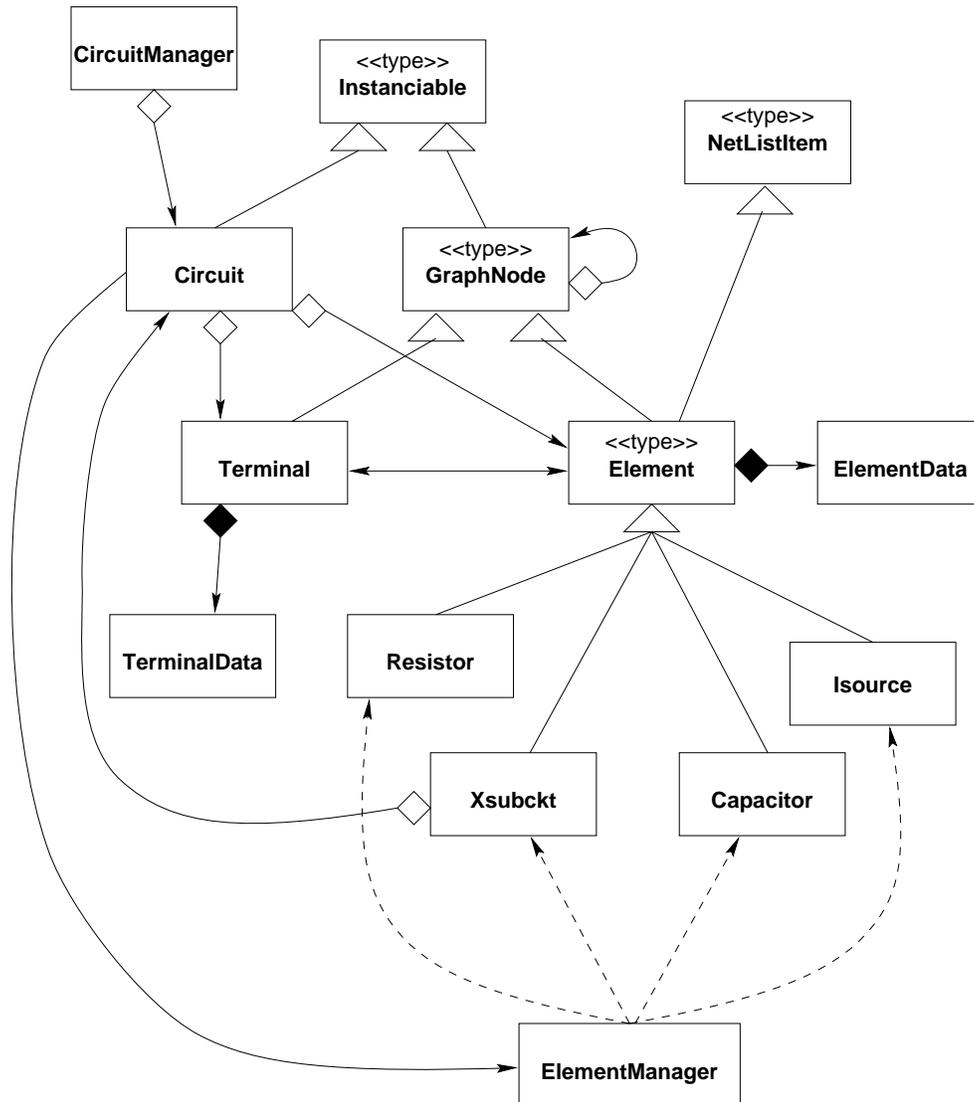
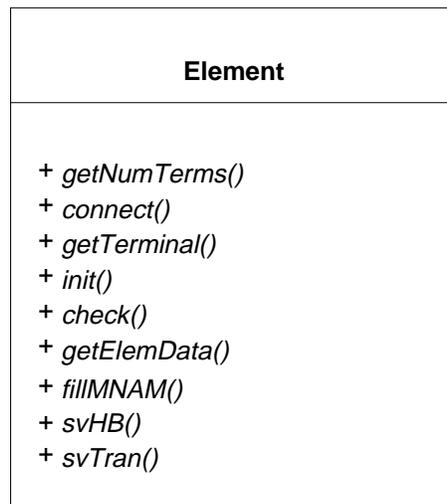


Figure 7.1: The network package is the core of the simulator.

Figure 7.2: The **NetListItem** class.Figure 7.3: The **Element** class.

4 terminal physical transmission line

Authors: Carlos E. Christoffersen, Mete Ozkar

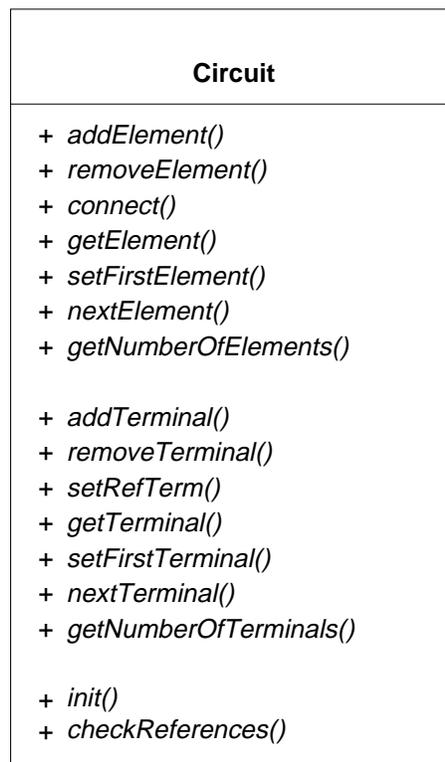
Multi-referenced element.

Usage:

tlinp4:<instance name> **n1 n2 n3 n4** <parameter list>

Parameter	Type	Default value	Required?
k : Effective dielectric constant	DOUBLE	1	no
alpha : Attenuation (dB/m)	DOUBLE	0.1	no
z0mag : Magnitude of characteristic impedance (ohms)	DOUBLE	n/a	yes
fscale : Scaling frequency for attenuation (Hz)	DOUBLE	0	no
tand : Loss tangent	DOUBLE	0	no
length : Line length (m)	DOUBLE	n/a	yes

Figure 7.4: Documentation generated for an element.

Figure 7.5: The **Circuit** class.

umentation in html format, see (Figure 7.4). The **Circuit** class represents either a main circuit or a subcircuit as a collection of elements and terminals. It provides methods to add, remove and find elements and terminals using different criteria and it also provides methods related to circuit topology. More details about this class are given in Figure 7.5. All the **Element** and **Terminal** instances must be stored in data structure inside the **Circuit** instance and the *map* container of the STL [70]. The map is a *Sorted Associative Container* that associates objects of type *Key* with objects of type *Data*. Here *Data* is either **Element** or **Terminal** and *Key* is *int* (the ID number). This is an example of where the features of C++ are used to reduce development time. This is achieved at no overhead as an optimum implementation of these concepts is embedded in the compiler.

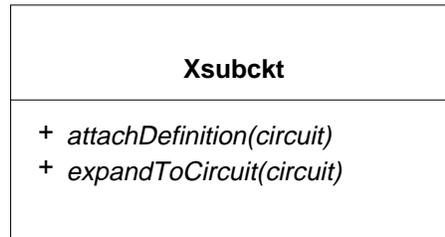


Figure 7.6: The **Xsubckt** class.

Subcircuit instances are represented by the **Xsubckt** class, see Figure 7.6. The method `attachDefinition()` is used to associate a **Circuit** instance where the actual subcircuit is stored to the particular **Xsubckt** instance and `expandToCircuit()` takes a **Circuit** pointer as argument and expands the subcircuit. Note that before expansion the complete hierarchy of a circuit is available in memory, so this engine could eventually be used to perform hierarchical simulation.

7.3 The Analysis Classes

Figure 7.7 shows the relation between the network package, the elements and the analysis classes. Each of these classes stores analysis-specific data that would traditionally be global. This leads to the key desired attribute of flexibility in incorporating a new type of analysis, or even different implementations of the same analysis type. Examples are the **SVTr** and **SVHB**

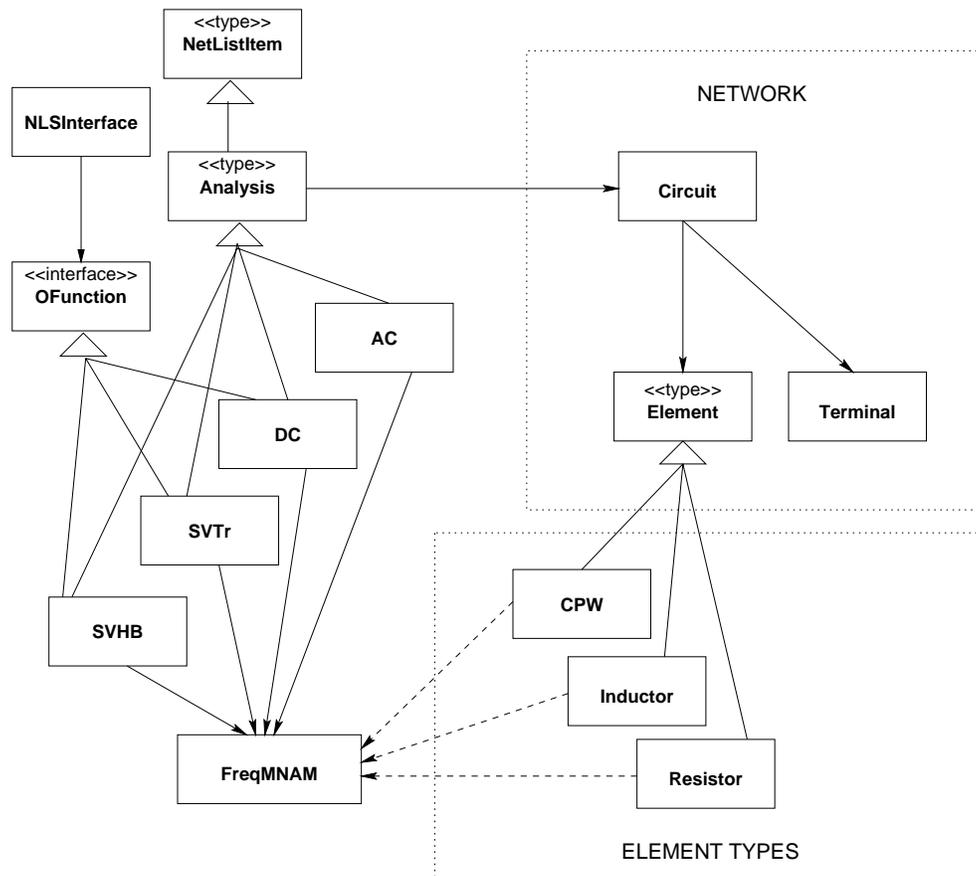


Figure 7.7: The analysis classes.

classes which contain the state-variable- convolution transient and harmonic balance analyses described in Chapter 4 and Reference [2], respectively.

There are some components common to two or more analysis types. The natural way of handling these components is by creating a class which is shared by the different analysis types. For example, the **FreqMNAM** class handles a Modified Nodal Admittance Matrix (MNAM) in the frequency domain. In a microwave simulator, the frequency domain admittance matrix is a key element for most analysis types. Since it is used so often, special care was taken to optimize efficiency. The elements fill the matrix directly without the need for intermediate storage of the element stamp. They do that by means of in-line functions to reduce function call overhead. Elements can fill the source vector in a similar way. The elements depend on the **FreqMNAM** methods, but this is not a problem since the interface is very unlikely to change. The current implementation of MNAM uses the Sparse library, described in the next Section, and is completely encapsulated inside the **FreqMNAM** class. In the same way, the **NLSInterface** class encapsulates the nonlinear solver routines. Therefore it is possible to replace the underlying libraries, if that is desired, without the need for any code modification outside the wrapper classes. A final observation is that it is possible to add any kind of analysis type provided that the appropriate interface is defined and the member functions are written for each element type.

7.4 Nonlinear Elements

Nonlinear elements often use service routines provided by the analysis classes. In order to maximize code reuse and to avoid the dependence of the element code on a particular analysis routines, in Transim elements depend on interface classes (Figure 7.8). The concept is similar to the *dependency inversion* [62]. This is a technique which relies on interface classes (normally implemented using abstract classes in C++) to make different parts of a program independent of each other. They only depend on the interfaces. To achieve greater efficiency, in Transim the dependency inversion is implemented using a concrete class with heavy in-lining and pass-by-reference.

In this way, the element routines and the analysis depend on an interface class, **TimeDomainSV** (not shown in Figure 7.1 for clarity). **TimeDomainSV** is a class that is used to exchange information between an element and a state variable based time domain analysis. It also provides some basic

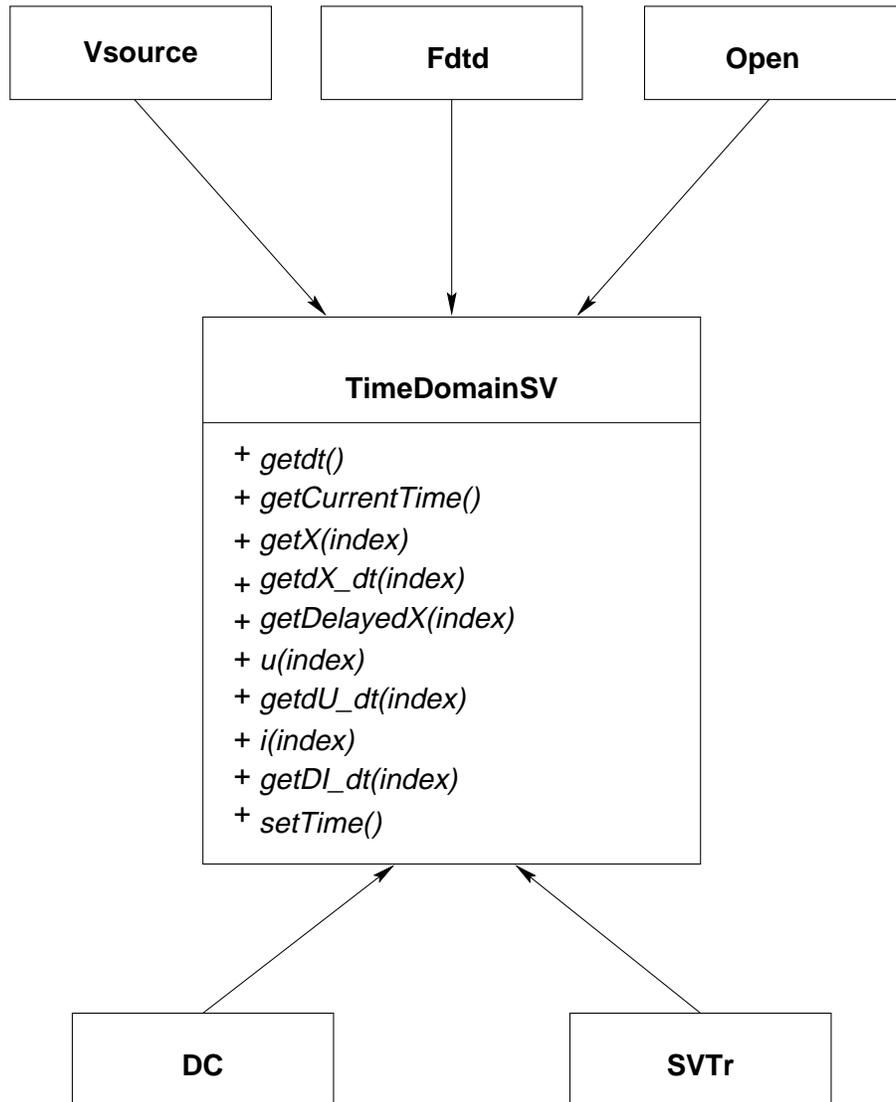


Figure 7.8: Dependency inversion was used to make the elements independent of the analysis classes.

algorithms such as time differentiation methods. This approach enables the element routines to be reused by several analysis types without the need to modify the element code (as long as the new analysis is state variable-based). For example, the **DC** analysis uses the same interface element as the **SVTr**.

Transim offers a more refined way to implement nonlinear elements, based in the state variable definition of Equations (3.3) and (3.4). By implementing those parametric equations using a special syntax in only one function, Transim can obtain the analysis functions `svHB()`, `svTran()` and derivatives automatically. This mechanism is termed *generic evaluation*.

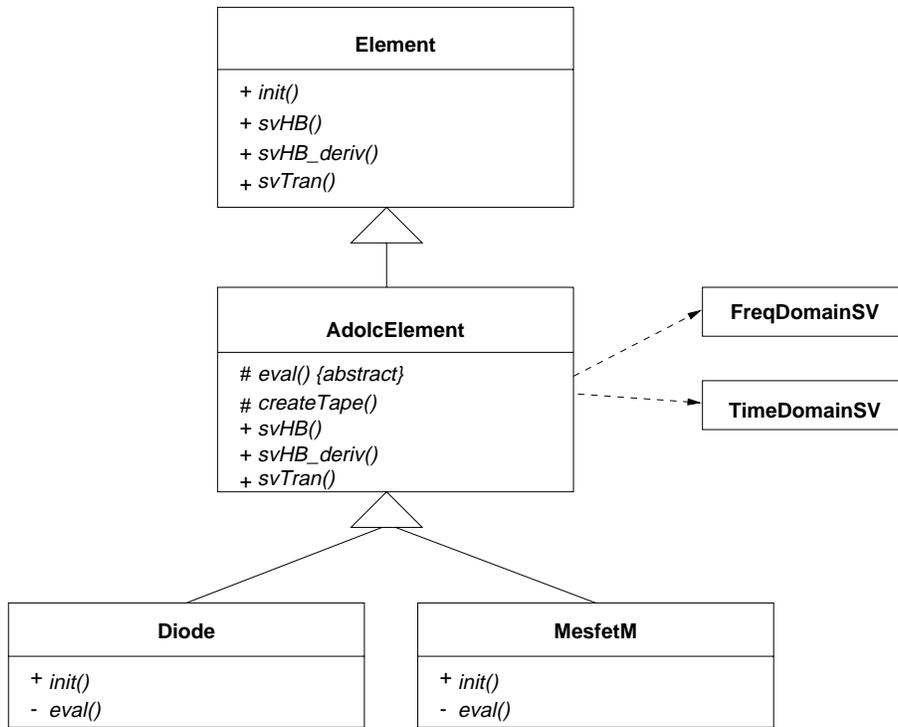


Figure 7.9: Class diagram for an element using generic evaluation.

Figure 7.9 shows the class diagram for an element using this feature. Note that the **Diode** class is derived from a class (**AdolcElement**) which itself provides the analysis routines and deals with the analysis interfaces. The **Diode** class only needs to implement the `eval()` function with the parametric equations.

AdolcElement uses the Adol-C library (see Section 7.6.5 for a detailed

explanation) to evaluate the parametric function and derivatives, but the concept is independent of automatic differentiation. If automatic differentiation were not used, then the derived class (*e.g.* the **Diode** class) would have to provide the Jacobian for the parametric Equations, (2.31) and (2.32).

The idea is in a way similar to the one used in [7], *i.e.* the primitive equations are ‘wrapped’ in analysis-specific generic functions and so there is no need to write a separate routine for each analysis type. In the current work there are two additional features. The first is that the generic evaluation is combined with the state variable concept of Section 2.4 and automatic differentiation. This provides unprecedented simplicity to create nonlinear element models. The second is that a single mechanism is not mandatory. There are cases where it may not be practical to use this approach, or the overhead involved (which is completely acceptable even for simple models such as a diode) may not be properly amortized. In those cases, the element can implement the analysis functions directly, without using the **AdolcElement** class.

It is important to remark that generic evaluation is implemented efficiently so there are no superfluous calculations. The current implementation supports elements with any number of state variables. Each element selects the input variables as a subset of the following: the state variables, the first derivatives, the second derivatives and a time delayed version of the variables (the delay may be different for each). No derivation, time delaying nor transformation is performed on the unselected inputs.

For example, an element with only an algebraic nonlinearity (such as the VCT: voltage controlled transducer) only selects the state variables without any derivatives or delays. As another example, the **MesfetM** class implements the Materka-Kacprzac model for a MESFET. It requires two state variables, but only one of them needs to be delayed.

A consequence of having a library of elements using generic evaluation is that it is possible to add a new analysis type by just adding the appropriate evaluation routines to the **AdolcElement** class. Thus, the maintenance and expansion of the simulator is simplified.

7.5 Example: Use of Polymorphism

The concept of polymorphism is briefly explained in the introduction to object oriented programming in appendix B. The scheme presented in Section

7.4 constitutes a good example of the use of polymorphism to solve a complex problem. Consider the segment of code of Figure 7.10. This code evaluates the nonlinear element functions in the state variable convolution transient. `elem_vec` is a vector of Element pointers implemented using the vector container of the C++ STL. There is no need to keep the size of the vector

```
// Number of elements
int n_elem = elem_vec.size();
int i = 0;
// Go through all the nonlinear elements
for (int k = 0; k < n_elem; k++) {
    // Set base index in interface object
    tdsv->setIBase(i);
    // nonlinear element evaluation
    elem_vec[k]->svTran(tdsv);
    i += elem_vec[k]->getNumberOfStates();
}
```

Figure 7.10: Nonlinear element function evaluation in convolution transient.

in a separate variable as `elem_vec.size()` returns the size of the vector. Also, the memory management of the vector is dynamic and automatic; and `elem_vec[k]` returns the Element pointer at position k in the vector.

Each pointer inside `elem_vec` points to different kinds of elements. For the transient routine the actual type of each element does not matter. The line containing `elem_vec[k]->svTran(tdsv)` will call the appropriate evaluation routine depending on the actual type of Element pointer. The element may implement the routine directly or through generic evaluation, but it makes no difference for the analysis routine.

The resulting code is therefore simple and there is no need for lists of “if-then” statements. Those lists would be very difficult to maintain, because each time an element is added or removed, all the lists would have to be updated.

7.6 Support Libraries

A large number of support libraries are available (many of them freely) and some of these are used in Transim. The various libraries, which should be of general interest to the microwave modeling community, are described below.

7.6.1 Solution of Sparse Linear Systems

*Sparse 1.3*² [74] is a flexible package of subroutines written in C used to numerically solve large sparse systems of linear equations. The package is able to handle arbitrary real and complex square matrix equations. Besides being able to solve linear systems, it is also able to quickly solve transposed systems, find determinants, and estimate errors due to ill-conditioning in the system of equations and instability in the computations. Sparse also provides a test program that is able to read matrix equation from a file, solve it, and print useful information (such as condition number of the matrix) about the equation and its solution. Sparse was originally written for use in circuit simulators and is well adapted to handling nodal- and modified-nodal admittance matrices.

*SuperLU*³ is used in the wavelet and time marching transient analyses. It contains a set of subroutines to numerically solve a sparse linear system $\mathbf{Ax} = \mathbf{b}$. It uses Gaussian elimination with partial pivoting (GEPP). The columns of \mathbf{A} may be reordered before factorization; the reordering for sparsity is completely separate from the factorization. SuperLU is implemented in ANSI C. It provides support for both real and complex matrices, in both single and double precision.

7.6.2 Vectors and matrices

Most of the vector and matrix handling in Transim uses *MV++*⁴ [73]. This is a small set of vector and simple matrix classes for numerical computing written in C++. It is not intended as a general vector container class but rather designed specifically for optimized numerical computations on RISC and pipelined architectures which are used in most new computer architectures. The various MV++ classes form the building blocks of larger

²<http://www.netlib.org/sparse/>

³<http://www.nersc.gov/xiaoye/SuperLU/>

⁴<http://math.nist.gov/mv++/>

user-level libraries. The MV++ package includes interfaces to the computational kernels of the Basic Linear Algebra Subprograms package (BLAS) which includes scalar updates, vector sums, and dot products. The idea is to utilize vendor-supplied, or optimized BLAS routines that are fine-tuned for particular platforms.

The *Matrix Template Library* (MTL)⁵ is a high-performance generic component library that provides comprehensive linear algebra functionality for a wide variety of matrix formats. It is used in the wavelet and time marching transient analyses.

As with the STL, MTL uses a five-fold approach, consisting of generic functions, containers, iterators, adaptors, and function objects, all developed specifically for high performance numerical linear algebra. Within this framework, MTL provides generic algorithms corresponding to the mathematical operations that define linear algebra. Similarly, the containers, adaptors, and iterators are used to represent and to manipulate matrices and vectors.

7.6.3 Solution of nonlinear systems

Nonlinear systems of equations in Transim are solved using the *NNES*⁶ [59] library. This package is written in Fortran and provides Newton and quasi-Newton methods with many options including the use of analytic Jacobian or forward, backwards or central differences to approximate it, different quasi-Newton Jacobian updates, or two globally convergent methods, etc. This library is used through an interface class (**NLSInterface**), so it is possible to install a different routine to solve nonlinear systems if desired by just replacing the interface (four different nonlinear solvers have already been used). The Fortran routine *NLEQ1* (Numerical solution of nonlinear (NL) equations (EQ)⁷) can also be used as a compile option.

⁵<http://www.lsc.nd.edu/research/mtl/>

⁶<http://www.netlib.org/opt/>

⁷Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB). Contact: Lutz Weimann, ZIB, Division Scientific Computing, Department Scientific Software, e-mail: weimann@zib.de

7.6.4 Fourier transform

Fourier transformation is implemented in Transim using the *FFTW*⁸ library [52]. FFTW is a C subroutine library for computing the Discrete Fourier Transform (DFT) in one or more dimensions, of both real and complex data, and of arbitrary input size. Benchmarks, performed on a variety of platforms show that FFTW's performance is typically superior to that of other publicly available FFT software. Moreover, FFTW's performance is portable: the program performs well on most computer architectures without modification.

7.6.5 Automatic differentiation

Most nonlinear computations require the evaluation of first and higher derivatives of vector functions with m components in n real or complex variables [72]. Often these functions are defined by sequential evaluation procedures involving many intermediate variables. By eliminating the intermediate variables symbolically, it is theoretically always possible to express the m dependent variables directly in terms of the n independent variables. Typically, however, the attempt results in unwieldy algebraic formulae, if it can be completed at all. Symbolic differentiation of the resulting formulae will usually exacerbate this problem of *expression swell* and often entails the repeated evaluation of common expressions.

An obvious way to avoid such redundant calculations is to apply an optimizing compiler to the source code that can be generated from the symbolic representation of the derivatives in question. Exactly this approach was investigated by Speelpenning during his Ph.D. research [75] at the University of Illinois from 1977 to 1980. Eventually he realized that at least in the cases $n = 1$ and $m = 1$, the most efficient code for the evaluation of derivatives can be obtained directly from the evaluation of the underlying vector function. In other words, he advocated the differentiation of evaluation algorithms rather than formulae. In his thesis he made the particularly striking observation that the gradient of a scalar-valued function (*i.e.* $m = 1$) can always be obtained for no more than five times the operations count of evaluating the function itself. This bound is completely independent of n , the number of independent variables, and allows the row-wise computation of Jacobians for at most $5m$ times the effort of evaluating the underlying vector function.

⁸<http://www.fftw.org>

Given a code for a function $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$, automatic differentiation (AD) uses the chain rule successively to compute the derivative matrix. AD has two basic modes, forward mode and reverse mode [76]. The difference between these two is the way the chain rule is used to propagate the derivatives.

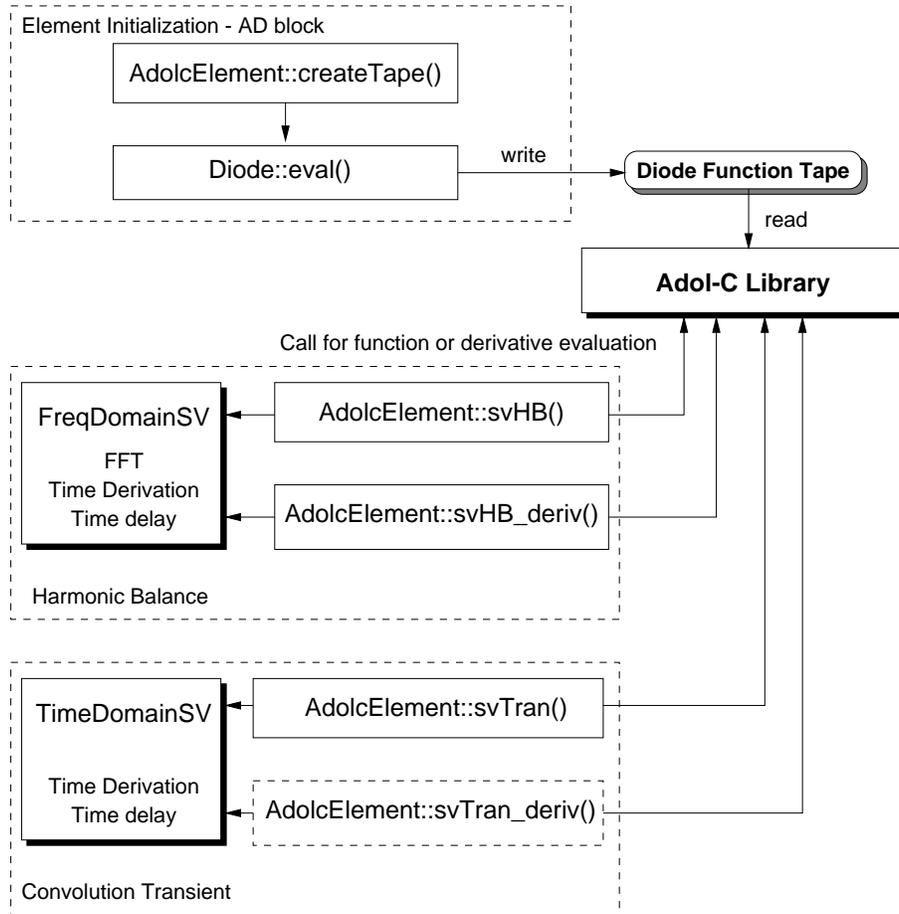


Figure 7.11: Implementation of automatic differentiation.

A versatile implementation of the AD technique is *Adol-C*⁹ [72], a software package written in C and C++. The numerical values of derivative vectors (required to fill a Jacobian in Harmonic Balance analysis [2], see Figure 7.11) are obtained free of truncation errors at a small multiple of the run time required to evaluate the original function with little additional memory

⁹<http://www.math.tu-dresden.de/~adol-c/>

required. It is important to note that AD is not numerical differentiation and the same accuracy achieved by evaluating analytically developed derivatives is obtained.

The `eval()` method of the nonlinear element class is executed at initialization time and so the operations to calculate the currents and voltages of each element are recorded by Adol-C in a *tape* which is actually an internal buffer. After that, each time that the values or the derivatives of the nonlinear elements are required, an Adol-C function is called and the values are calculated using the tapes. This implementation is efficient because the taping process is done only once (this almost doubles the speed of the calculation compared to the case where the functions are taped each time they are needed). When the Jacobian is needed, the corresponding Adol-C function is called using the same tape. We have tested the program with large circuits with many tones, and the function or Jacobian evaluation times are always very small compared with the time required to solve the matrix equation (typically some form of Newton's method) that uses the Jacobian. The conclusion is that there is little detriment to the performance of the program introduced by using automatic differentiation. However the advantage in terms of rapid model development is significant. The majority of the development time in implementing models in simulators, particularly harmonic balance simulators, is in the manual development of the derivative equations. Unfortunately the determination of derivatives using numerical differences is not sufficiently accurate for any but the simplest circuits. With Adol-C full 'analytic' accuracy is obtained and the implementation of nonlinear device models is dramatically simplified. From experience the average time to develop and implement a transistor model is an order of magnitude less than deriving and coding the derivatives manually. Note that time differentiation, time delay and transformations are left outside the automatic differentiation block. The calculation speed achieved is approximately ten times faster than the speed achieved by including time differentiation, time delay and transformations inside the block.

7.7 Summary

Object oriented techniques offer significant advantages for the design of circuit simulators. Great design flexibility is obtained without compromising efficiency. This is due to advances in both programming techniques and com-

puler technology. However careful analysis of the problem and programming discipline are required. The use of ‘off-the-shelf’ libraries permits rapid development, higher quality of the code, and as they implement modern numerical methods which are regularly updated, currency is maintained. The simulator was developed to model spatially distributed circuits, and in particular spatial, power combining systems where electromagnetics, circuit and thermal interactions are important. The simulator architecture presented here is suited to the experimentation of new simulation algorithms and element models.

Chapter 8

Modeling of Quasi-Optical Systems

This chapter describes a modeling strategy that allows port-based EM and thermal models to be integrated with a nodal electrical circuit model. After the introduction in Section 8.1, the concept of local reference nodes, which is necessary to model spatially distributed structures, is described along with its implementation in Transim in Section 8.2. Section 8.3 presents the modeling of electromagnetic structures in Transim. The simulation of two power combiner systems are also presented in this section. Finally, the modeling of the thermal subsystem along with electrothermal simulation examples are presented in section 8.4.

8.1 Introduction

A continuing issue in regard to microwave and higher frequencies is the development of high power solid-state sources [1]. This is an enabling technology for a variety of military and commercial systems. Given the current state of maturity of microwave integrated circuit (MMIC) technology, the quest to produce high powers from solid state devices requires the investigation of novel semiconductor materials and of power combining architectures. The interrelated effects of the EM field, the linear and nonlinear circuit elements and the thermal subsystem present in quasi-optical systems are self-consistently modeled and simulated in the following sections.

8.2 Implementation of Local Reference Nodes

The distributed nature of many microwave and millimeter-wave circuits necessitates electromagnetic modeling. Thus the full application of computer-aided engineering (CAE) to these circuits requires the integration of electromagnetic models of distributed structures with conventional circuit analysis. In the formative stages of the CAE of microwave circuits, ports were used in specifying the interconnectivity of networks. The utilization of ports avoided the issue of specifying reference nodes. In analysis, using matrix manipulations or signal flow graphs, one of the terminals of a port was used implicitly as a reference node and generally ignored in formulating the mathematical model. The connection of discrete elements is specified nodally but at the highest level of the hierarchy port-based descriptions are used. While it is possible to analyze any circuit with this arrangement it becomes increasingly difficult to specify the connections of large spatially distributed circuits, and also to specify and extract desired output quantities. The alternative to using port-only descriptions is to exclusively use the nodal connectivity description, the only method used in general purpose circuit simulators. The conventional nodal specification enables circuit elements to be connected in any possible combination and only one reference node (commonly called the global reference node or simply ground) is used. With spatially distributed circuits it is possible to make illegal connections such as connecting a non-spatially distributed element, say a resistor, across a spatially distributed element, *e.g.* a transmission line. The purpose of this section is to develop and describe the implementation of a scheme for checking the legal connections of a network with mixed spatially distributed and conventional elements.

8.2.1 Background

A circuit is a graphical construct (which can be specified in textual form as a netlist) for coupling together algebraic and first order differential equations. These equations arise from the constitutive relations of the individual elements, which specify the actual form of the individual equations, and from Kirchoff's current law which specifies how the equations are coupled. With a few modifications, the analysis thus summarized is called nodal admittance analysis. It is important to note that circuits have no sense of space—a circuit is defined as though it existed at a point. The problem is then how to incorporate an electromagnetic model of a structure that is inherently dis-

tributed in space. Of two approaches, one is to insert the device equations into an appropriate time-stepping electromagnetic simulator such as a finite difference time domain (FDTD) simulator [77, 78]. This reduces the level of abstraction of the ‘circuit’ and embeds the constitutive relations of the conventional circuit elements into the analysis grid of the FDTD method. An alternative is to retain the high-level circuit abstraction and incorporate the results of a field analysis (the spatially distributed circuit) into a circuit structure [79–82]. Thus the problem is one of taking an electromagnetic solution and inserting it as a circuit element.

At this point it is useful to review modified nodal admittance (MNA) analysis to indicate the basis for the work presented here. MNA analysis was developed to handle elements that do not have nodal admittance descriptions. For each such element one or more additional equations are added to the nodal admittance equations and these new equations become additional rows and columns in the evolving matrix system of equations. A similar approach can be followed for the electromagnetic elements. The process is a little more sophisticated as it is no longer sufficient to add additional rows. Instead the concept of local reference nodes was developed [12] as a generalization of the compression matrix approach [82]. This concept provides another way to incorporate alternate equations in the evolving MNA matrix. However, rather than adding additional constitutive relations, the local reference node concept changes the way the port-based parameters are used. Figure 8.2 shows a spatially distributed circuit with local reference nodes indicated by the diagonal symbol. In a conventional circuit only one reference node (ground) is possible so that application of KCL to the global reference node introduces just one additional redundant row and column in the indefinite form of the MNA matrix. For a spatially distributed circuit, KCL is applied to each locally referenced group one at a time and, as the local reference nodes are electrically connected only through a spatially distributed element, each application results in a redundant row and column in the MNA matrix. This is a particular property of the port-based characterization of the spatially distributed element [12]. The mathematical description of the required extension to the nodal admittance analysis is given in [12].

8.2.2 Handling of Local Reference Nodes

The basis of this approach is that in a multi-port element, the terminals of ports with different local reference nodes can be considered isolated. For

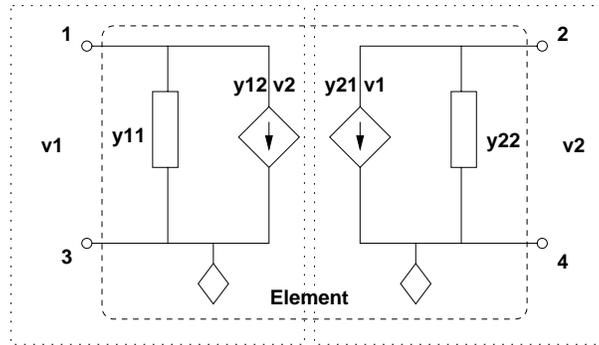


Figure 8.1: Equivalent circuit of a two-port distributed element.

example, if we write the equivalent circuit of a two-element derived from the port-based admittance (y) parameters,

$$\begin{bmatrix} i1 \\ i2 \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \begin{bmatrix} v1 \\ v2 \end{bmatrix}$$

we obtain the equivalent circuit in Fig. 8.1. Note that no current can flow between the two ports. This type of circuit model can be generalized for a multiport element where external and internal local reference nodes are defined. The local references shown in Fig. 8.1 are internal because they are used internally in the element to measure the voltages at its ports. An external local reference terminal, on the other hand, is an arbitrarily chosen terminal from a locally referenced group. In general, for any spatially distributed circuit (Fig. 8.2) there is no current between port groups inside the spatially-distributed element. Thus the circuit can be divided into sub-circuits, each with a local reference terminal, as shown in Fig. 8.3. Each subcircuit is isolated with respect to the others, so there is no change in the circuit behavior if all the reference nodes are connected together. The difference is that now there is only one global reference node for all the circuit, and standard nodal methods can be used to formulate the circuit equations.

The problem becomes that of detecting violations to the assumption that each locally referenced group is isolated from the others and that there is exactly one reference terminal for each one. Such a situation is shown in Fig. 8.4. The effect of the connection of the lumped element between the two locally referenced groups in the figure is the creation of a current loop that is non-physical since the two circuits cannot be connected instantaneously.

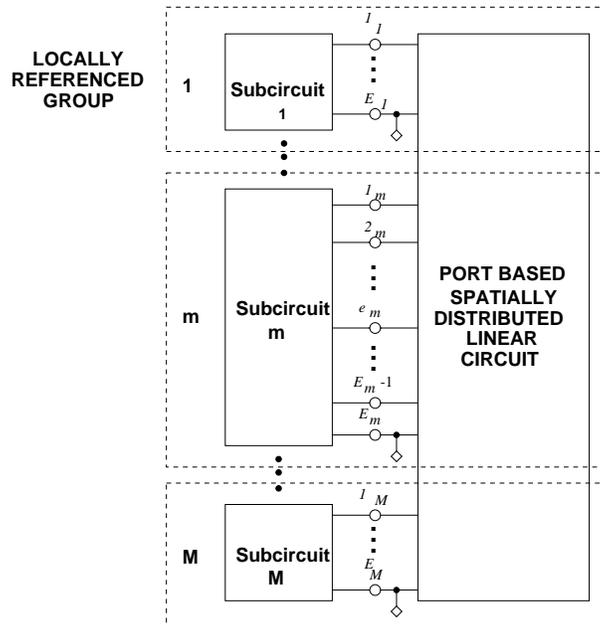


Figure 8.2: Generic spatially distributed circuit.

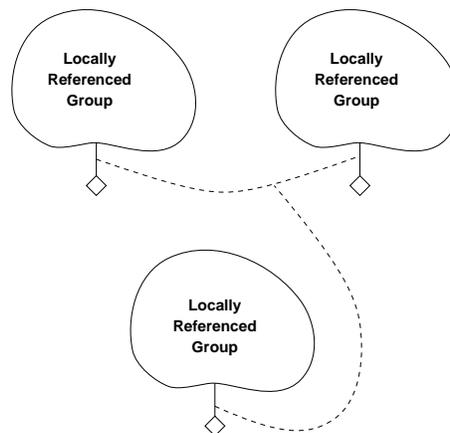


Figure 8.3: The locally referenced groups are isolated, therefore the solution is unchanged if the local references are connected.

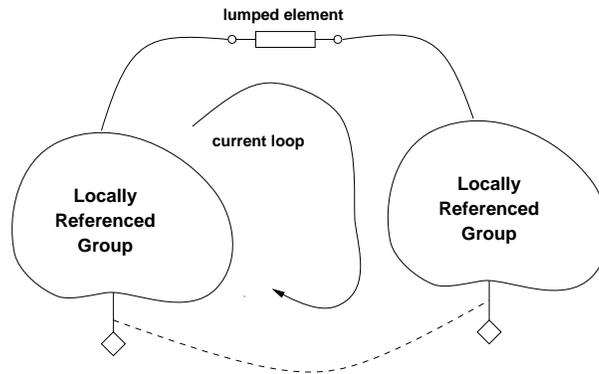


Figure 8.4: Illegal connection between groups.

On the other hand, it is valid for two subcircuits to be connected by more than one spatially distributed element. It is also possible for two ports of a spatially distributed element to be connected to the same locally referenced group (for example a delay line). In this case, the external local reference node would be the same for both ports of the line, but internally, the line still has two local reference nodes. After all the checking is done, all the local reference nodes are merged into a single global reference node, and the solution of the circuit is found by standard procedures. The nodal voltages found are coincident with the voltages referred to each local reference node.

8.2.3 Implementation in Transim

The implementation of this approach can be described using a conventional procedural approach or an object oriented approach. The object oriented view maps onto the circuit analysis problem cleanly. A circuit is a collection of objects (resistors, inductors, transmission lines, etc.) that are related to each other (they share nodes). With the introduction of spatially distributed elements along with the conventional elements there are certain rules that describe allowable interconnections. The rule is

There must be one and only one reference node for each locally referenced group.

In Transim, each element and each terminal is a node in a graph structure. The algorithm to detect violations is a variation of the depth-search algorithm and is best illustrated using what is called a collaboration diagram. The al-

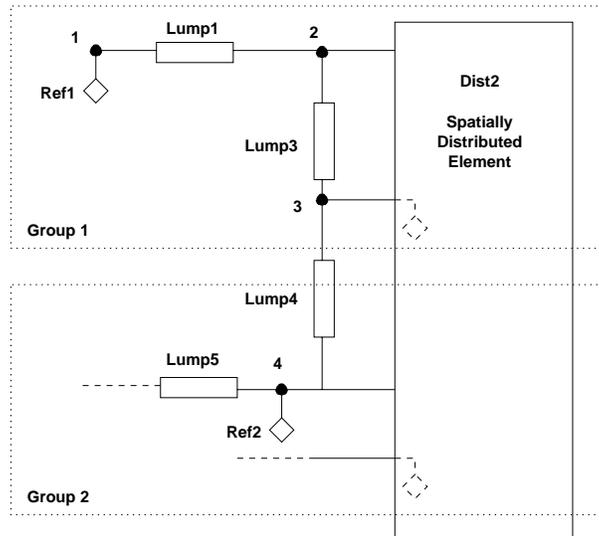


Figure 8.5: Circuit containing an illegal element. The numbers at the nodes indicate the order in which terminals are discovered.

gorithm could also be implemented using conventional procedural programming [83]. The collaboration diagram [84] is part of the Unified Modeling Language (UML), a language for specifying, visualizing, and constructing the artifacts of software systems as well as for business modeling. The UML represents a collection of ‘best engineering practices’ that have proven successful in the modeling of large and complex systems. Behavior is implemented by sets of objects that exchange messages within an overall interaction to accomplish a purpose. To understand the mechanisms used in a design, it is important to see only the objects and the messages involved in accomplishing a purpose or a related set of purposes, projected from the larger system of which they are part for other purposes. Such a static construct is called a collaboration. A collaboration is a set of participants and relationships that are meaningful for a given set of purposes. The identification of participants and their relationships does not have global meaning. In the collaboration diagram, each box represents an object (in this case, either a terminal or an element). The lines between boxes represent associations and the arrows are messages. The order in which messages are sent is shown by the numbers.

Figure 8.5 shows the circuit view corresponding to the collaboration diagram of Figure 8.6. The terminal numbers indicate the order in which

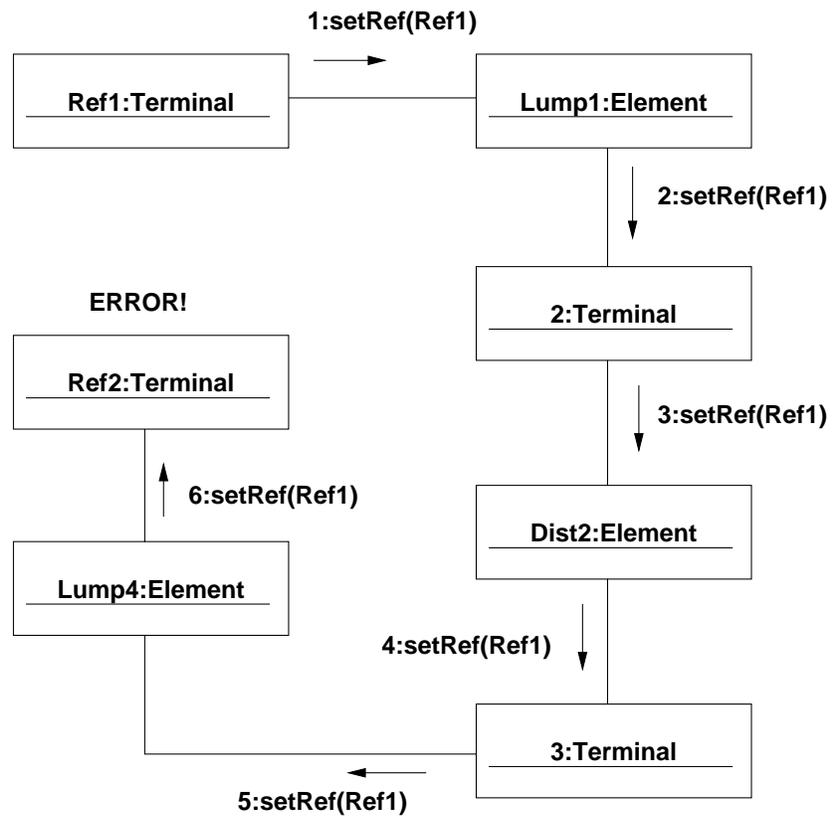


Figure 8.6: Collaboration diagram showing the violation detection algorithm.

terminals are discovered by the algorithm. Note also that the internal local reference terminals of the spatially distributed element do not need to be coincident with the external references. The search begins at each local reference terminal. The local reference terminal sends a message to propagate its identifier (id) to all the adjacent elements. If the element that receives the message has only one local reference terminal (lumped), it continues the propagation to the other adjacent terminals. Otherwise it only propagates the id to the element terminals with the same internal reference. If an element attempts to propagate a reference id to a terminal already marked with another reference id, then a violation is detected: there is an illegal connection between two locally referenced groups. A check to detect floating parts of the circuit can be made by finding any terminal that does not belong to any locally referenced group. All these checks are implemented in Transim. Note that the bias circuit often violates the condition that the locally referenced groups are isolated between them. To overcome this either the distributed elements should include the DC characteristics or an independent DC source should be considered in each group. In the input netlist, the local reference nodes are identified with the command `.ref <terminal>`. By default the nodes named 0 and gnd are assumed to be reference nodes.

8.3 Integration of Electromagnetic Structures

8.3.1 Frequency Domain and Time Domain Using Convolution

EM modeling yields port-based y parameters of the structures at each frequency of interest. The transfer of data between the EM and circuit simulators (typically a file) includes a header with port grouping information (a port grouping includes terminals associated with a specific local reference node). This is required by the circuit simulator in order to expand the port-based matrix into nodal form and also to check the connectivity of the spatially-distributed circuit.

Figure 8.7 shows a sample data file. Each port belongs to a different group, so the element has four terminals. After reading the header the circuit simulator knows the number of elements of the matrix and the port number and local reference corresponding to each row and column of the matrix. The rest of the file consists of a list of frequencies and the associated matrix

```

# port:group
1:1
1:2
# GHZ Y RI R 50
4
0.00355603      -0.0233196
-0.00121905    -0.00496212
-0.00121905    -0.00496212
0.00355603      -0.0233196
...

```

Figure 8.7: Sample frequency-domain y parameter file.

elements.

Once read by the simulator, the parameters can be directly used in HB and Convolution Transient analysis. However further processing is required to use this data in a time domain analysis using a Pole-Zero approximation.

8.3.2 Time Domain using Pole-Zero Approximations

The set of frequency samples of the y parameters for the EM structure are used as input to a program (written using *Octave*¹, a free program similar to *Matlab*) which generates a rational function approximation using a method similar to the one described in Section 2.1.3. As an illustration of the kind of approximation currently achieved, Figure 8.8 shows a comparison between the original and the approximated y_{11} for the 2x2 grid described in Section 8.3.4. The polynomial coefficients are written to a file as the one shown in Figure 8.9. The header is similar to the one used for the frequency domain file. f_{max} is the normalizing frequency. There is one rational function for each element in the y matrix. The polynomial coefficients are given for the normalized frequency $\omega_{norm} = 2f/f_{max}$, where f is the actual frequency.

The rational function of Equation (2.1) with $H(s) = I(s)/V(s)$ is expressed in the time domain using the inverse Laplace transformation as follows

$$i = q_0v + q_1v' + q_2v'' + \dots + q_\xi v^{(\xi)} - r_1i' - r_2i'' - \dots - r_\theta i^{(\theta)} \quad (8.1)$$

¹ftp://ftp.che.wisc.edu/pub/octave

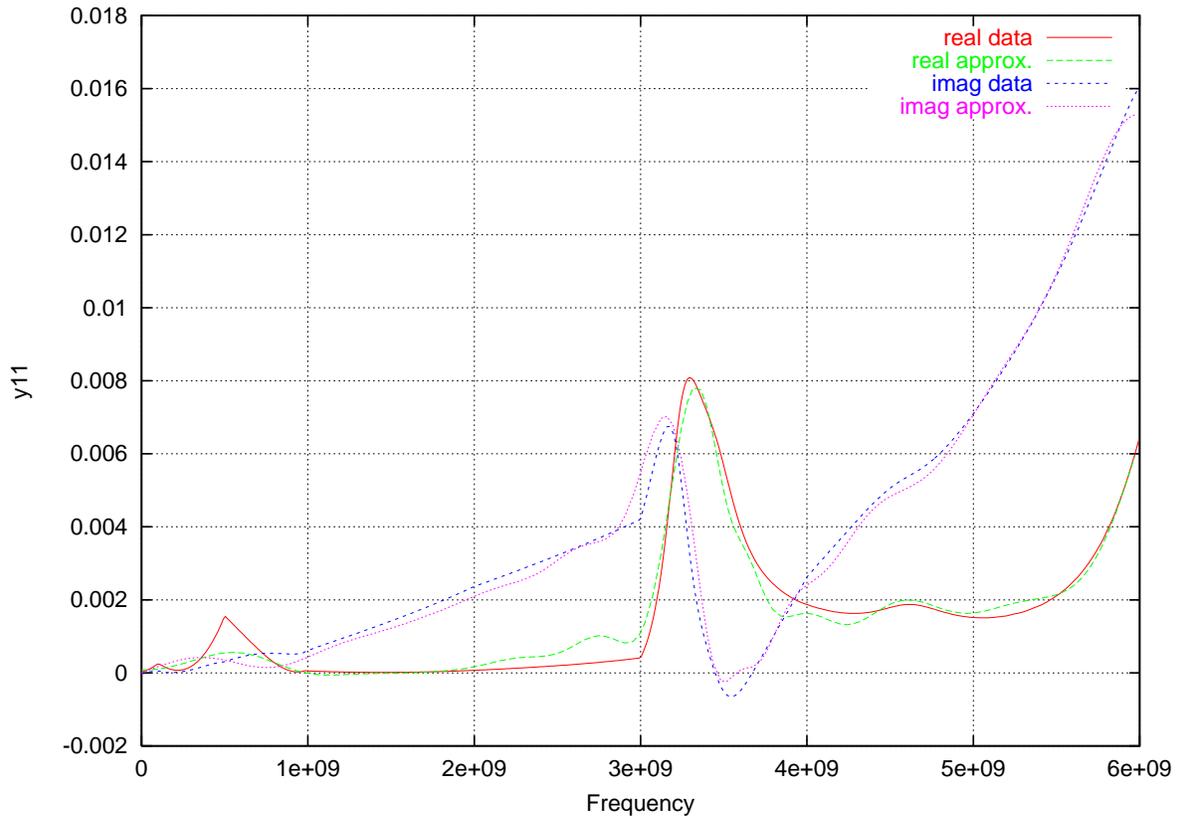


Figure 8.8: Comparison between the original and the approximated y_{11} for the 2x2 grid.

```
# port:group
1:1
2:1
3:1
4:1
1:2
2:2
3:2
4:2
1:3
2:3
3:3
4:3
1:4
2:4
3:4
4:4

fmax = 6e9

! Row 1 numerator
Degree: 15
-4.37708776601776e-02
-2.18430714850605e-01
-1.66834816830034e-01
...
+2.12677671834220e-01
+6.74241821653293e-02
+5.47259429373396e-03
+0.00000000000000e+00
! Row 1 denominator
Degree: 16
+9.16023582743810e-02
+1.66900260707466e+00
+1.40732624093744e+01
+7.31251775538846e+01
+2.63196514090180e+02
...
```

Figure 8.9: Sample time-domain y parameter file.

This equation can be entered directly into the MNAM of the circuit by defining the derivatives of the current i and the voltage v as additional variables. This is what the current Transim implementation does. It is not optimal because the wide varying order of magnitude of the coefficients cause ill-conditioning of the MNAM when the number of poles is large (in practice, greater than 30). This could be avoided by expressing the transfer functions in the pole-residue form, as shown in Equation (2.5). Future implementations will use this idea.

8.3.3 CPW Folded-Slot Active Antenna Array

In this section we show the modeling and the simulation results of a CPW folded-slot active antenna array as an example of the integration of electromagnetic structures in Transim and the use of local reference nodes. The system unit cell is considered first. Then we consider a 2 by 2 array.

Unit Cell

Consider the CPW folded-slot active antenna [85] shown in Figure 8.10. This is a component of a spatial power combining circuit. The unit cell amplifier is excited by an incident horizontally polarized field and radiates an amplified vertically polarized field. Complete analysis of this structure uses electromagnetic characterization as well as circuit simulation. In Figure 8.10 the two orthogonal folded-slots are connected to each other by a CPW with an inserted MMIC amplifier. The system is modeled using the circuit of Figure 8.11 and electromagnetic modeling of the structure is discussed in [86–88]. Note that three different local reference nodes, indicated by the diamond shaped node symbol, are required. EM modeling yields port-based y parameters of the antennas at each frequency of interest. The currents calculated by this simulation are used by an EM program to evaluate the effective isotropic power gain of the amplifier cell. The comparison between the simulated results and the measurements presented in [85] are shown in Figure 8.12.

2x2 Array

The 2x2 antenna array shown in Figure 8.13 is modeled using the equivalent circuit in Figure 8.14. A multi-port distributed element is used to model the

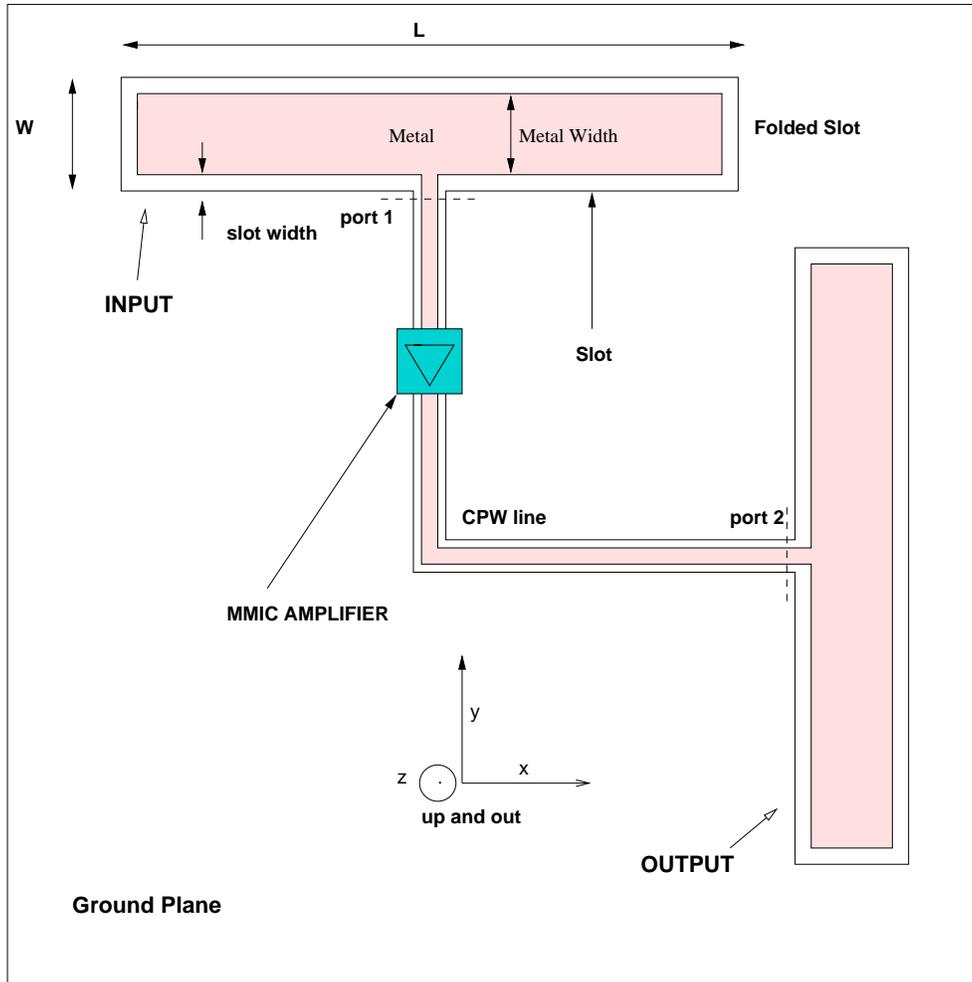


Figure 8.10: Unit cell of the CPW antenna array.

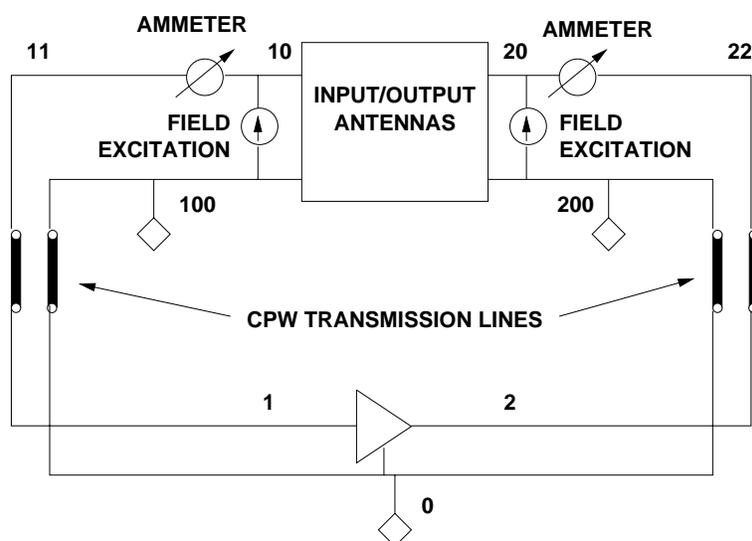


Figure 8.11: Circuit model of the unit cell. The diamond symbol indicates a local reference node.

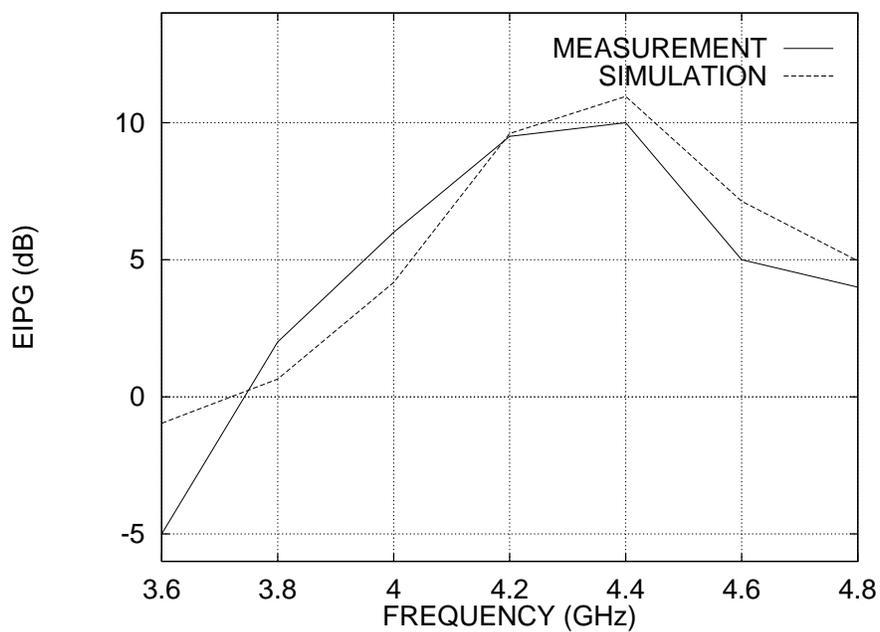


Figure 8.12: Effective isotropic power gain as a function of frequency.

antenna array so the electromagnetic interactions of the antennas of different cells is taken into account.

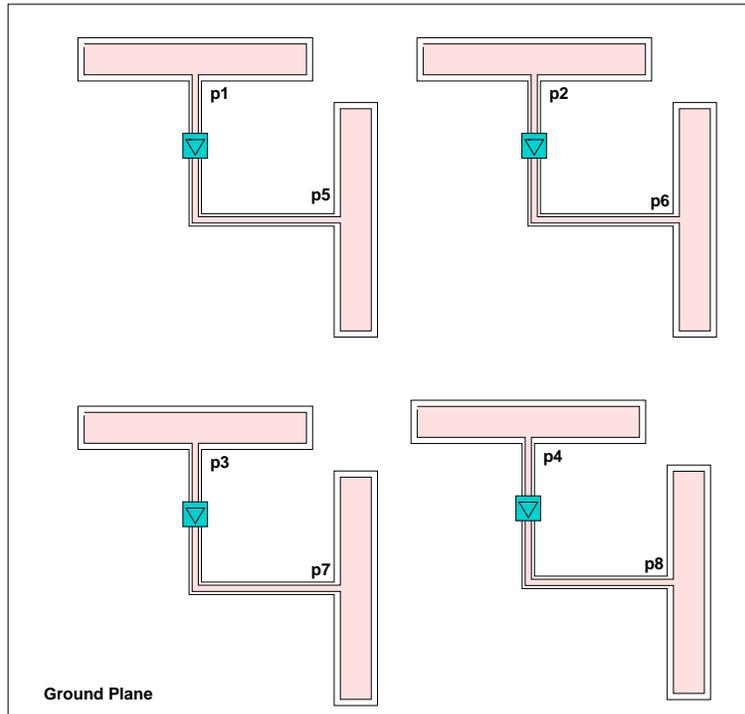


Figure 8.13: 2x2 CPW antenna array.

The netlist (Figure 8.15) shows the syntax of subcircuits and of local reference nodes. The `.model` statements can be used in conjunction with any element type and here it is used for the transmission lines. The output statement (`.out`) provide a calculator to process output data. The output currents are plotted in Figure 8.16. As expected, the currents are all different since the system is not symmetric.

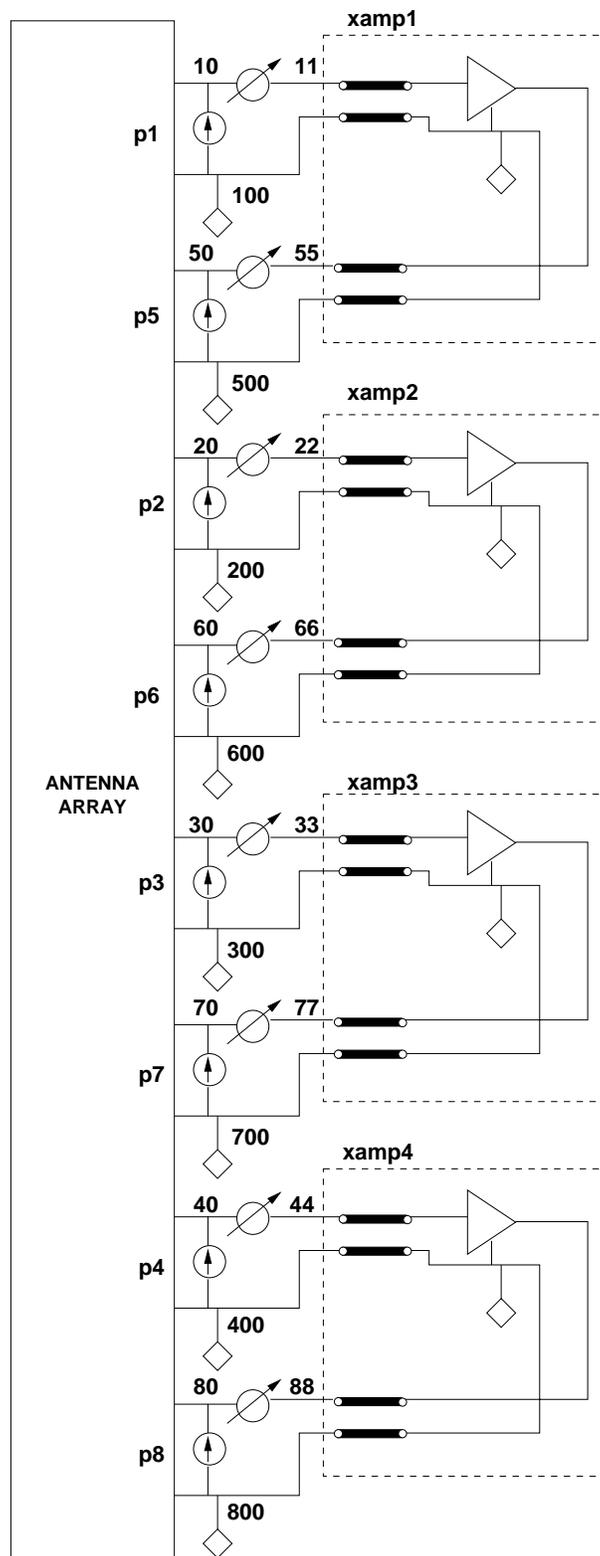


Figure 8.14: 2x2 CPW antenna array equivalent circuit.

```

.ac start = 3.6GHz stop = 5.0GHz n_freqs = 50
* Subcircuit with cpwlines and amplifier model
.subckt amp_lines 11 100 22 200
.ref "a_gnd"
* Transistor small signal model
nport:amplifier 1 2 "a_gnd" filename = "ne3210s1.yp"
* CPW Transmission lines
.model fsa1 cpw (s=.369m w=1m t=10u er=10.8 tand=.001)
cpw:t1 11 100 1 "a_gnd" model="fsa1" length=8.5m
cpw:t2 22 200 2 "a_gnd" model="fsa1" length=17.5m
.ends
* Local reference nodes
.ref 100
.ref 200
.ref 300
.ref 400
.ref 500
.ref 600
.ref 700
.ref 800
* Antenna array
nport:cpw_2 10 20 30 40 50 60 70 80
+ 100 200 300 400 500 600 700 800 filename = "2x2cell.yp"
* Field excitation
gridex:iin 10 100 20 200 30 300 40 400
+ 50 500 60 600 70 700 80 800 ifilename = "2x2cell.i"
+ efilename = "dummy.e"
* Amplifier instances
xamp1 11 100 55 500 amp_lines
xamp2 22 200 66 600 amp_lines
xamp3 33 300 77 700 amp_lines
xamp4 44 400 88 800 amp_lines
* Current meters
vsource:amp1 10 11
vsource:amp2 20 22
vsource:amp3 30 33
vsource:amp4 40 44
vsource:amp5 50 55
vsource:amp6 60 66
vsource:amp7 70 77
vsource:amp8 80 88
* Plot amplifier output currents
.out plot element "vsource:amp5" 0 if mag in "i5.outm"
.out plot element "vsource:amp6" 0 if mag in "i6.outm"
.out plot element "vsource:amp7" 0 if mag in "i7.outm"
.out plot element "vsource:amp8" 0 if mag in "i8.outm"
.end

```

Figure 8.15: Netlist for a 2x2 CPW antenna array.

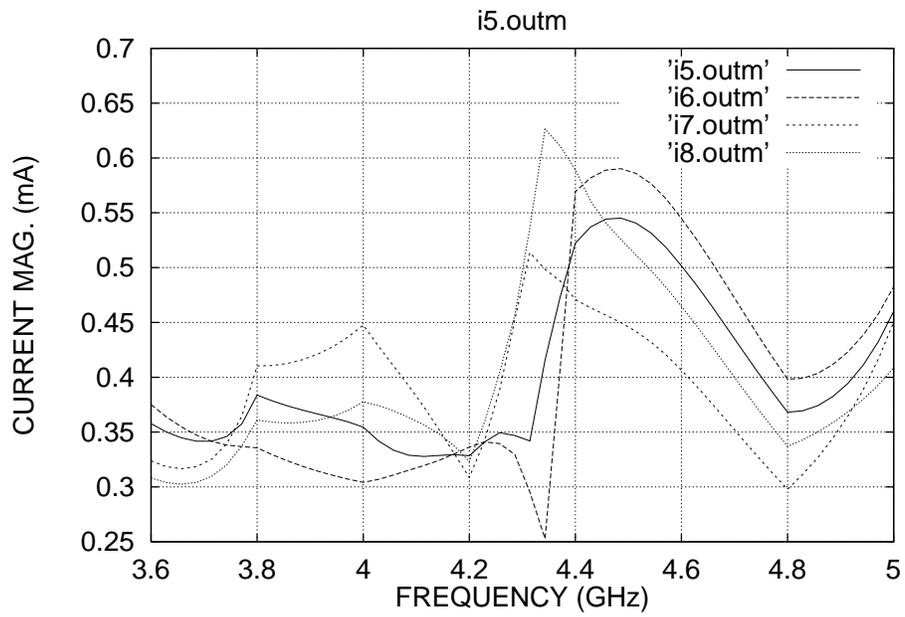


Figure 8.16: Output currents for the 2x2 antenna array.

8.3.4 2x2 Grid Amplifier

Transim is used in this section to model the nonlinear performance of a 2 by 2 quasi-optical grid amplifier system [89]. The purpose of this system was to provide experimental verification data, and so compromises were made in the complexity of the system. Furthermore the small size of the system does not efficiently launch a tightly focused quasi-optical beam. However, this does not affect the quality of the modeling. The 2 by 2 grid amplifier system is constructed and measured [89] as shown in Fig. 8.17. The layout and bias network are shown in Fig. 8.18. A horn-to-horn simulation of the

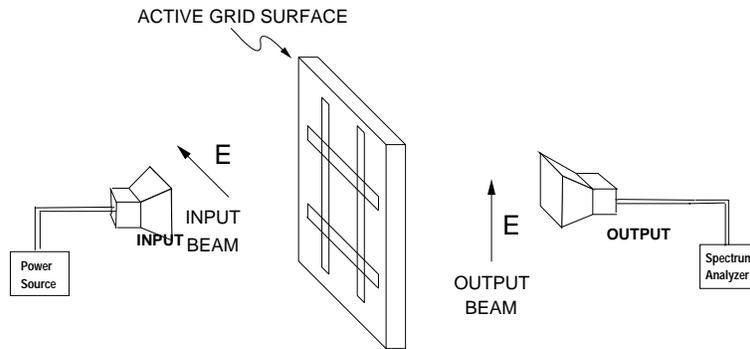


Figure 8.17: Setup for measurement characterization of the quasi-optical amplifier system.

grid amplifier system described above was performed. The grid structure was modeled using a MOM field simulator [80] to generate the multi-port admittance matrix and excitation currents for the grid structure. The horns are included in the simulation by modeling the horns using their far-field gain. The gain of the horns is used to calculate the field incident on the grid and the power received from the field radiated by the grid amplifier.

Spectral Regrowth

The first nonlinear simulation of the grid amplifier presented here is a spectral regrowth simulation using harmonic balance and the technique reviewed in Section 2.5. The grid was excited using 5 tones starting at frequency 5.4 GHz and separated by a frequency $\Delta f = 5$ MHz. The maximum distortion order considered was 5. The size of the resulting frequency vector applying formulas (2.33) and (2.34) was 106. The modifications made to Transim to support

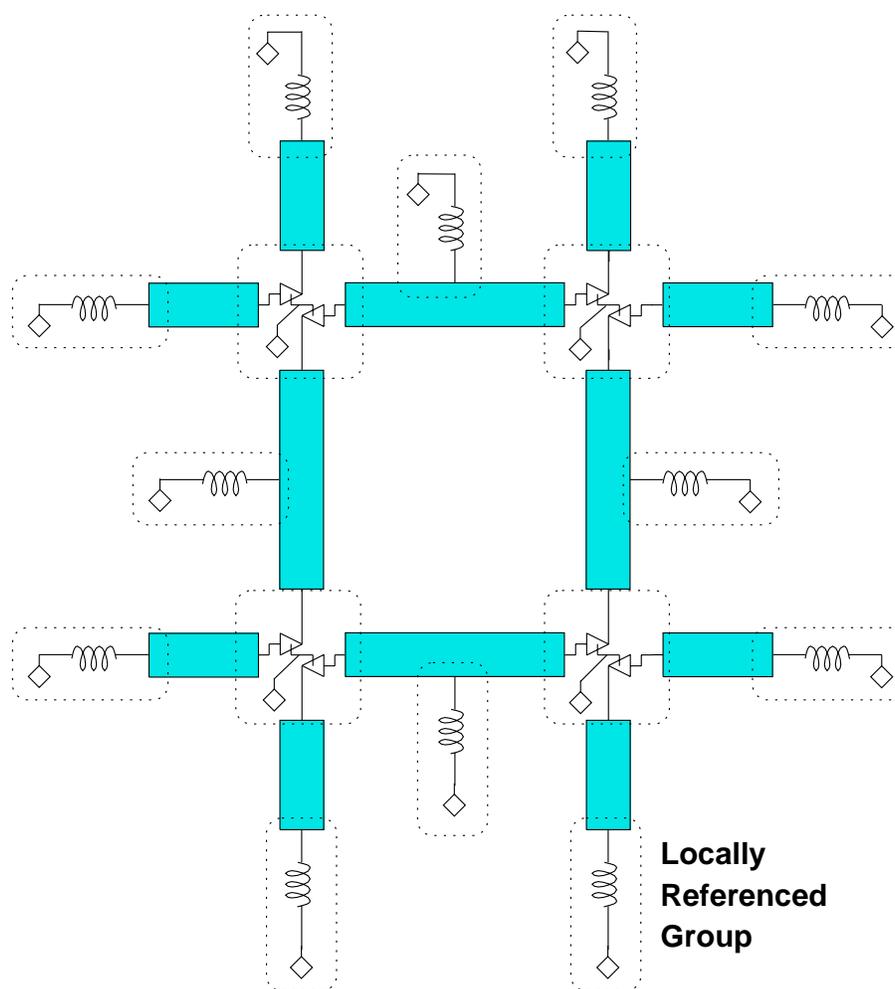


Figure 8.18: Layout for 2x2 quasi-optical grid amplifier.

this type of spectrum in the harmonic balance analysis were minimal because it is only a trivial extension of the frequency mapping technique described in [4]. Other harmonic balance simulators based on the multidimensional FFT can not handle this [46].

Figure 8.19 shows all the generated output frequency components. The spectral regrowth is shown in Fig. 8.20. The adjacent channel power regrowth is clear in this picture. Total simulation time was 10 hours in an

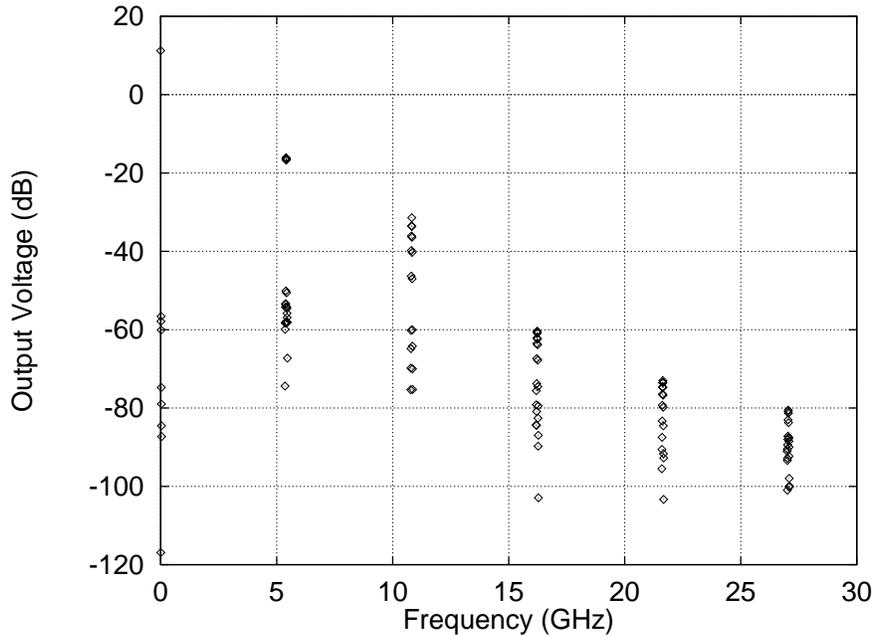


Figure 8.19: Full spectrum for output voltage.

UltraSparc 1. This time could be greatly reduced if a matrix-free harmonic balance algorithm were implemented in Transim.

Transient Analysis

The Transim simulation turn-on transient response of the grid amplifier without excitation is shown in Fig. 8.21. This system takes a very long time to settle down because of the RF chokes and the DC capacitors in the voltage regulation circuitry. Some special consideration had to be taken to perform the convolution simulation. For this system, some of the capacitors and inductors of the amplifier model had to be treated in the time domain (with the

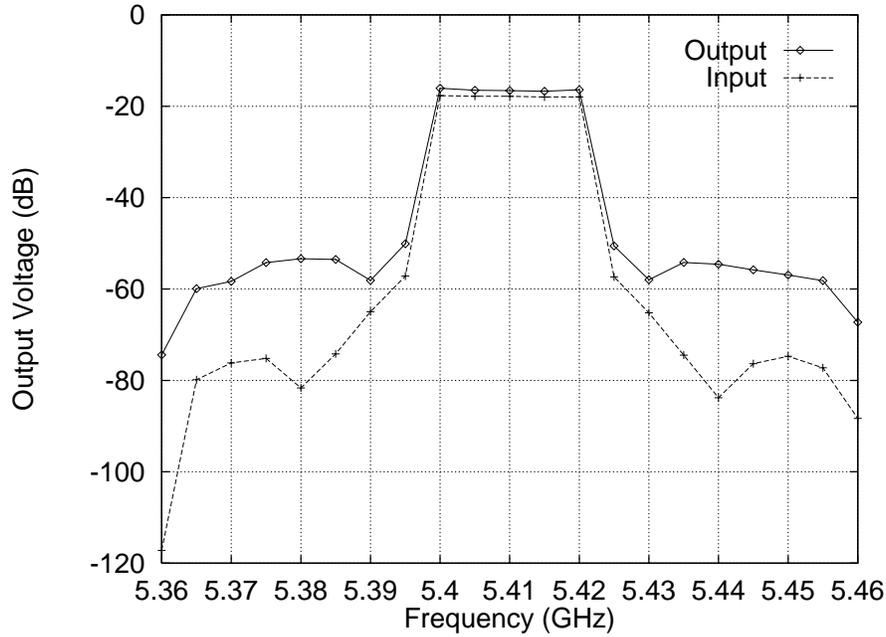


Figure 8.20: Spectral regrowth in one of the amplifiers.

nonlinear elements). The advantage of doing so is that the impulse response of the remaining linear subcircuit is shorter and thus the analysis requires less memory and computation time. On the other hand, this increases the number of state variables, thus increasing memory usage and computation time. Therefore there is a tradeoff between length of the impulse response and number of state variables.

Even though it is time consuming, the simulation of this initial power-on transient is essential to ensure initial stability. In Figure 8.21, the response calculated using convolution starts from a value below zero. This is the consequence of using a very long time step to store the impulse response, so it could fit in memory. Backward Euler integration yielded better results than trapezoidal because the system being simulated is very stiff.

Detail of the initial transient is given in Fig. 8.22. The excitation is applied at $t = 2$ ns. Note the two different convolution results. If the convolution analysis is performed using the pole-zero model of the grid, the agreement with the other transient simulations is much better. We can conclude that the pole zero modeling must be improved.

The third run used the initialization procedure outlined in section 4.3.1

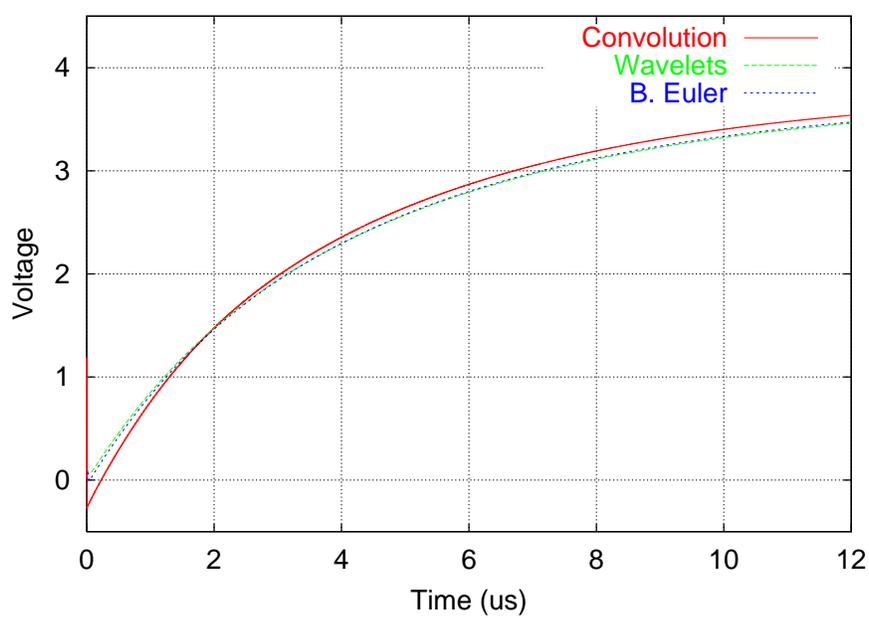


Figure 8.21: Transient response at one of the MMIC output ports using a large time step.

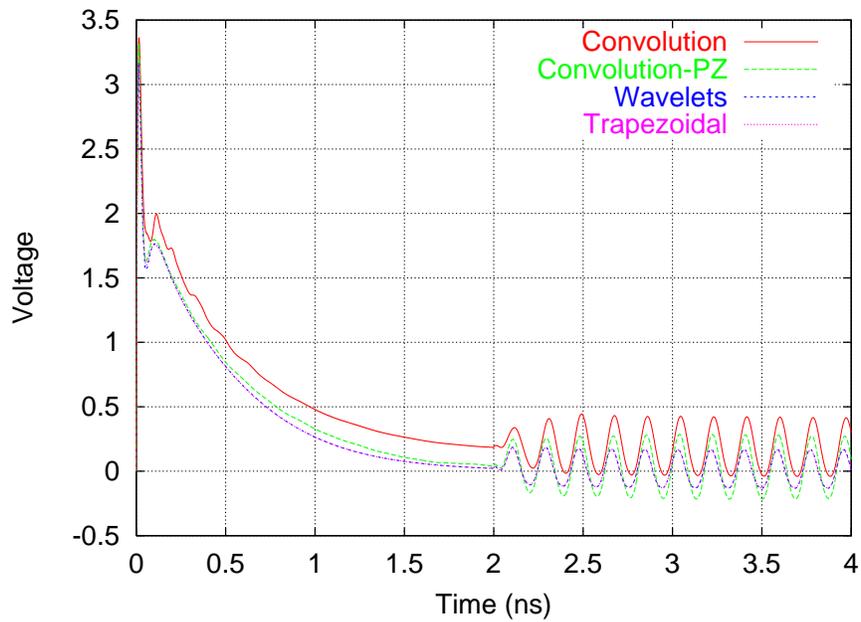


Figure 8.22: Zoom of the transient response at one of the MMIC output ports using a small time step.

for the convolution simulation. First the DC operating point was determined using the step response but solving for the correct nonlinear characteristics of the MMICs. Then transient analysis was begun with the initial conditions set, as shown in Figure 8.23. This makes it possible to determine the steady

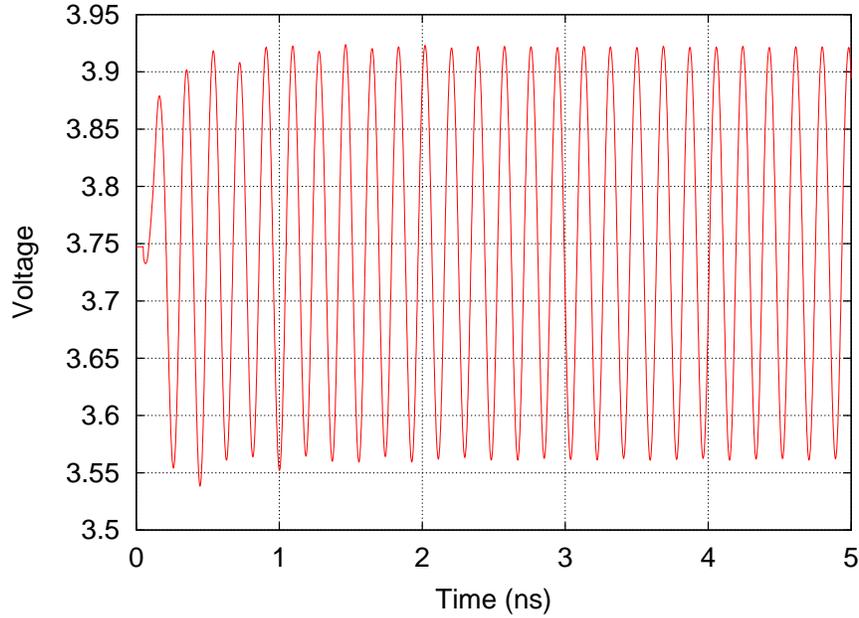


Figure 8.23: Output transient starting with the circuit already biased.

state response using convolution, for validation purposes. A simulation using HB would be preferred if modeling of non-periodic behavior was not required. Convolution, time marching and harmonic balance results are shown in Fig. 8.24. The result for the time marching simulation was obtained by simulating the complete transient ($30 \mu\text{s}$) and shifting the waveform an integer number of periods in order to make the comparison (this is because nonzero initial conditions are not implemented in this transient analysis yet). The results of the harmonic balance analysis have been previously verified [89], where measurements are compared with the results of harmonic balance and transient simulation. The differences between the simulations can be attributed to the effect of aliasing errors inherent to the generation of the impulse response for convolution and the pole-zero approximation for the time marching scheme. Harmonic balance uses fewer approximations and thus it is the most reliable result. No wavelet simulation is shown because the current wavelet imple-

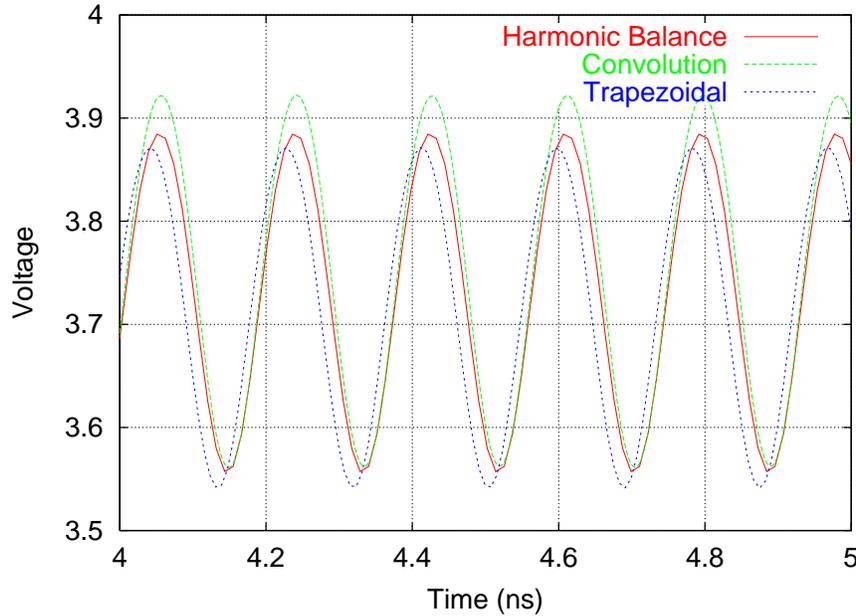


Figure 8.24: Comparison of the steady-state output waveforms.

mentation would use too much memory in a full transient with excitation. The transient using wavelets and trapezoidal integration for the first $4 \mu\text{s}$ are compared in Figure 8.25. The agreement between simulations is good.

A comparison of the run times is given in Table 8.1. Convolution transient is always the slowest method. Nevertheless, this method can potentially achieve the best accuracy. Transient based on wavelets is faster than convolution, but it is orders of magnitude slower than the time marching scheme. This suggests that a possible solution to achieve the best accuracy with a reasonable computation speed would be the combination of the time march-

Table 8.1: Comparison of the simulation times for the 2 by 2 grid array

Description	Convolution (h:m:s)	Wavelets (h:m:s)	Time Marching (h:m:s)
Bias-on ($12 \mu\text{s}$)	16:15:00	00:00:37	00:00:02
Bias + Excitation ($4 \mu\text{s}$)	-	18:24:00	00:11:36
Bias + Excitation (4 ns)	00:09:34	00:04:40	00:00:01

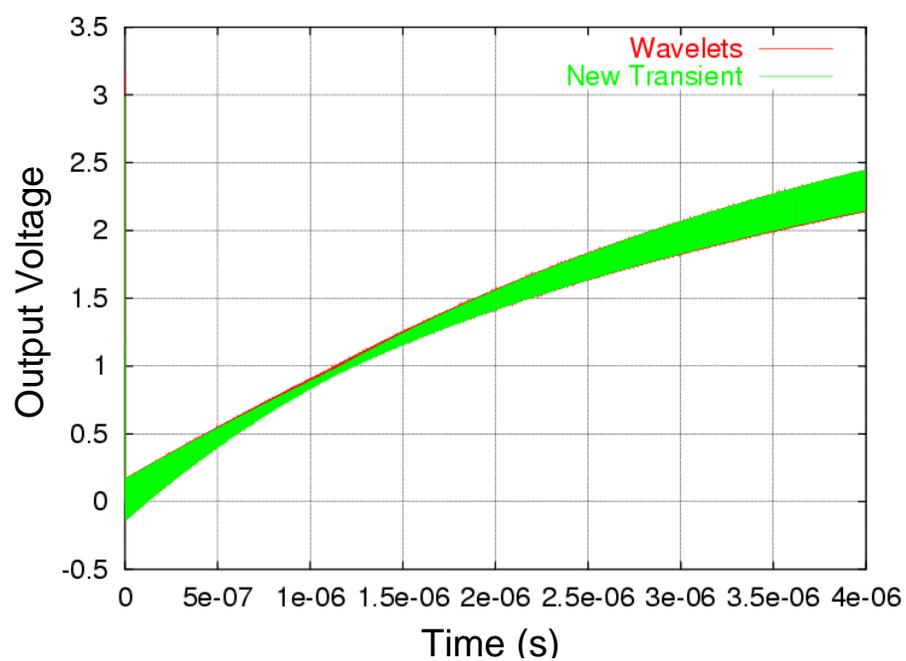


Figure 8.25: Transient response of one amplifier with RF excitation.

ing scheme with the convolution performed only in the grid model. For very long transient simulations, however, the only viable alternative seems to be the approximation based on rational functions.

8.4 Electro-Thermal Modeling

In this section, we will outline how thermal modeling is implemented in Transim and present the simultaneous thermal-electrical simulation of a quasi-optical system as an example. One way of incorporating thermal effects in a circuit simulator [90] is to make the thermal model look like an electrical circuit. The thermal and electrical circuits are then solved simultaneously as if they were one large electrical circuit. Power dissipated in active devices is represented as a heat current source referenced to thermal ground. One problem with this strategy is to provide separate circuits for the electrical and thermal parts. This has been addressed by the concept of local reference nodes, as described earlier in this chapter in Section 8.2. The use of local reference nodes guarantees that there is no mixing of electric and thermal currents.

8.4.1 Thermal Impedance Matrix

The thermal modeling of complex 3-dimensional structures can be achieved by standard numerical techniques [91], and solutions of the heat diffusion equation for complex 3-dimensional systems are commonly based on finite volume, finite element, finite difference or boundary element methods. All of these approaches require construction of a volume or surface mesh. They are computationally intensive and therefore generally too slow for direct coupling to electronic device and circuit simulators.

The thermal approach implemented in Transim has been presented in [91–94] and it is a new fully physical spectral domain decomposition technique with essentially no model reduction. This approach is called the *Leeds thermal impedance matrix model*. It is modular and it can describe simultaneously all device details, from surface metalization, bias and substrate thinning in power FETs and MMICs, through (actively cooled) MMIC on substrate arrays, up to MCMs and circuit board level.

The nonlinear thermal problem is converted into a linear problem by means of two transformations [91]. The Kirchoff transformation is used to

correct the steady state error of the thermal response and a time transformation is used to obtain accurate modeling of the time constants. Currently, Transim implements the Kirchoff transformation only.

8.4.2 Implementation of Thermal Circuits and Elements in Transim

Thermal elements in Transim are implemented as single- or multi-port blocks. Each port in a block corresponds to a surface where heat is exchanged. Each thermal element provides a routine to fill the corresponding frequency-domain MNAM elements (used in harmonic balance) and a time-domain routine used in transient analysis. The same routine is used in both convolution and time-marching transient analyses, as it happens with all nonlinear elements in Transim.

The following is the list of available thermal elements:

- Thermalmmic: heat sink mounted MMIC. One-port thermal element with a single averaged surface heating element.
- Thermalgrid: grid array substrate. One-port thermal element with a single averaged surface heating element.
- Thermal2: two-port MMIC die with variable base temperature (for interface matching), and a single averaged surface heating element.
- Thermalnport: N-port grid array substrate with NxN surface heating elements.

For the harmonic balance analysis in Transim, the thermal elements are modeled in the frequency domain. The thermal impedance matrix is directly entered into the modified nodal admittance matrix of the circuit at each frequency. Thermal elements are then ‘embedded’ in the linear part of the HB formulation. Therefore thermal elements do not increase the size of the nonlinear system of equations. The Kirchoff transformation is included in the nonlinear active device models, so the potential in the thermal circuit is then not the actual temperature, T , but the transformed temperature, θ .

Thermal elements are considered as a nonlinear elements in Transim’s transient analysis. Therefore they are treated in time domain. The thermal impedance matrix approach [91] is used to calculate the resulting temperature

at each iteration to solve the nonlinear system. Transient analysis require the precomputation of the thermal resistance matrix in all thermal elements for the complete set of time points of the simulation.

8.4.3 Electrothermal Modeling of a Quasi-Optical System

The 2x2 CPW folded-slot active antenna array system presented earlier in Section 8.3.3 is now modeled using nonlinear amplifiers. One amplifier is shown in Figure 8.26. The extra terminals in the MESFET schematic

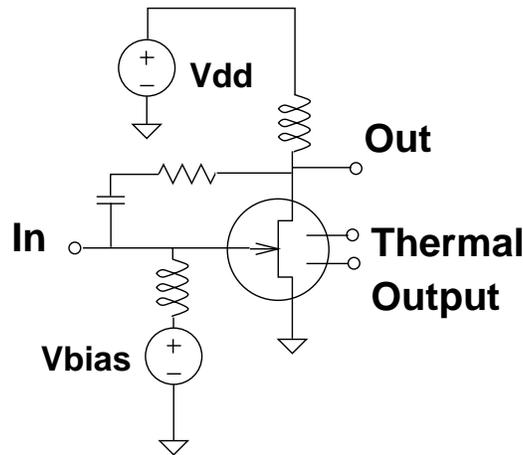


Figure 8.26: Amplifier circuit used.

(Fig 8.26) represent the thermal connections. From the thermal circuit point of view, the MESFET behaves as a current source that applies some heat flow to the thermal circuit and this increases the temperature of the MESFET device. Some MESFET model variables are affected by the increase in temperature, adjusting thus the behavior of the electrical circuit.

The Thermal2 element in Transim is used to model the GaAs die of each amplifier and a ThermalNport element is used for the substrate as shown in Figure 8.27. The ambient temperature is modeled by a potential source (300 K in this case).

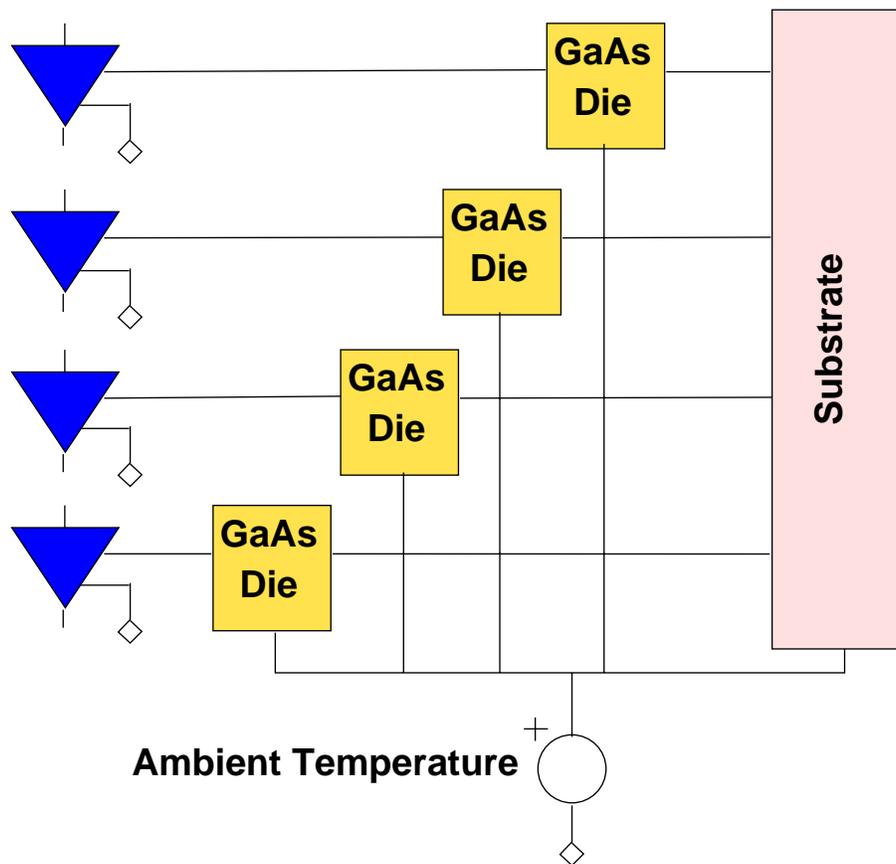


Figure 8.27: Thermal circuit model for the 2x2 CPW array.

8.4.4 Simulation Results and Discussion

The Transim steady-state simulation results are shown in Figures 8.28 thru 8.30. In Fig. 8.28 the simulated output voltage at one of the the amplifiers is shown, both with and without considering thermal effects. The results show that the self-heating effect in the MESFET modifies the electrical behavior of the circuit. The operating temperatures at the GaAs dies and the substrate

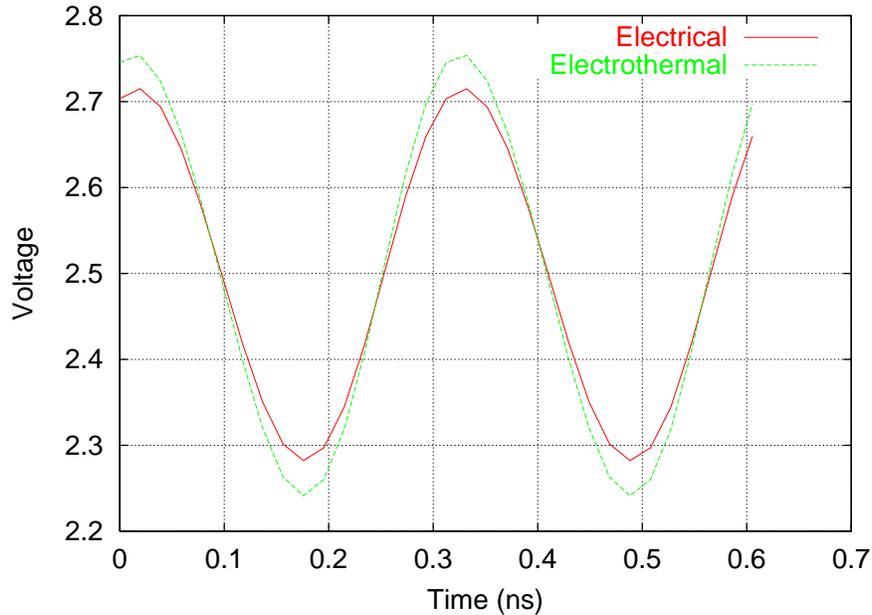


Figure 8.28: Comparison of the steady-state output voltage with and without thermal effects.

are shown in Figure 8.29. The die temperature appears constant here, but a change in scale in Figure 8.30 reveals an oscillation at the excitation frequency. This oscillation has a very small amplitude. Therefore, for practical purposes, the temperature of the die in steady state can be considered as a constant. Figure 8.31 illustrates the potential of the model for prediction of thermal effects on spectral regrowth and ACPR. The electrothermal simulation exhibits a greater ACPR compared with the electrical-only simulation. This simulation was performed by exciting the grid with 5 fundamental frequencies starting at 3.2 GHz and spaced by 5 MHz. Simulation time was 5 hours for the electrothermal case in a Pentium III Xeon 550MHz.

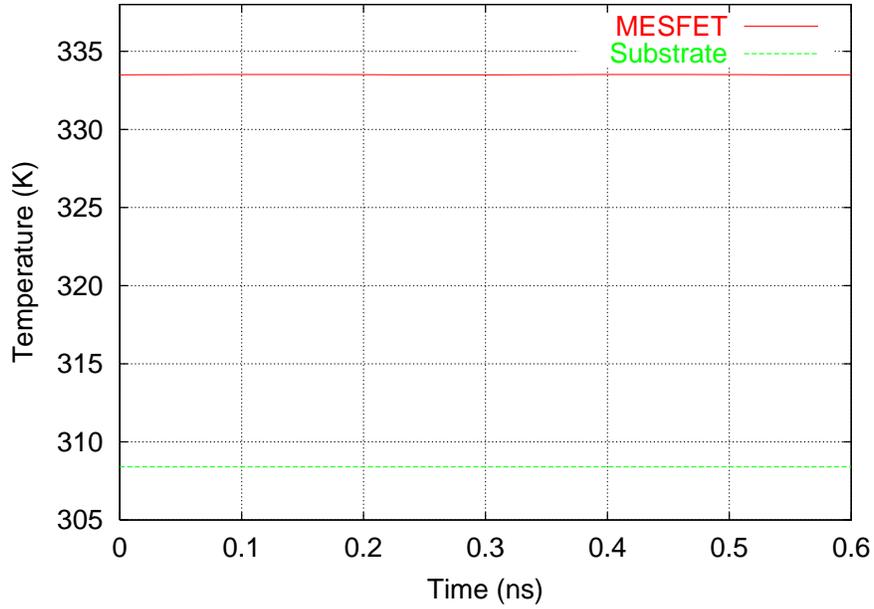


Figure 8.29: Steady-state temperature at the MESFET and the substrate.

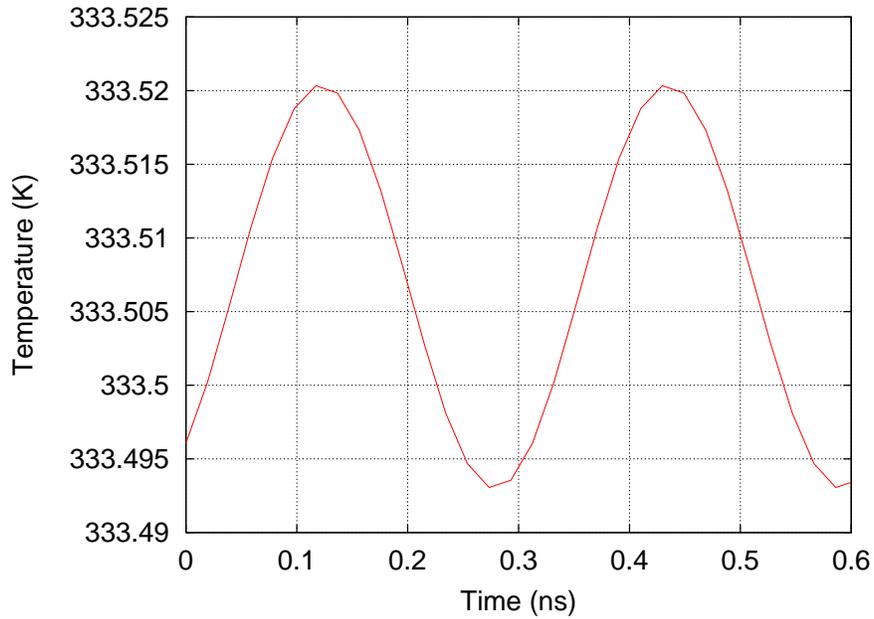


Figure 8.30: Detail of the MESFET temperature.

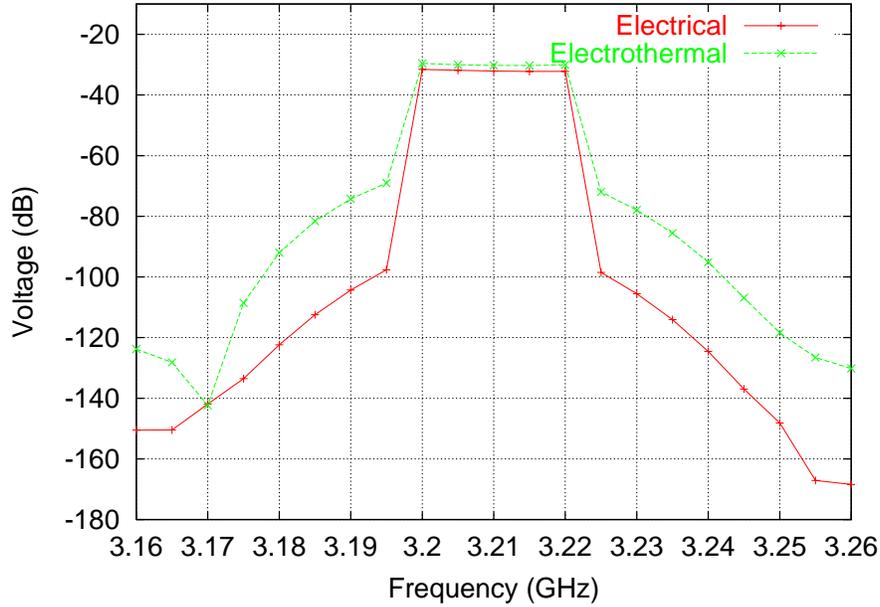


Figure 8.31: Effect of the temperature on the ACPR of the output voltage.

Figures 8.32 and 8.33 show part of the thermal transient (excitation off). The substrate temperature is far from reaching the steady-state value, and so is the die temperature. Currently, the precomputation of the thermal resistance matrix takes most of the simulation time. A complete transient simulation with microwave excitation would require the precomputation (and storage) of so many samples of the thermal resistance matrix that it is currently impossible to achieve. Research is under way to alleviate this problem. Note however that the Leeds thermal impedance matrix approach still is computationally more efficient than mesh-based thermal modeling, because only the power and temperatures at the interface points are calculated. One of the main issues of electrothermal transient simulation is that the time constants are extremely wide varying. One solution to this problem is to separate the system in blocks with similar time constants and run coupled simulations with different time steps for each block. This and other possibilities will be investigated in the future.

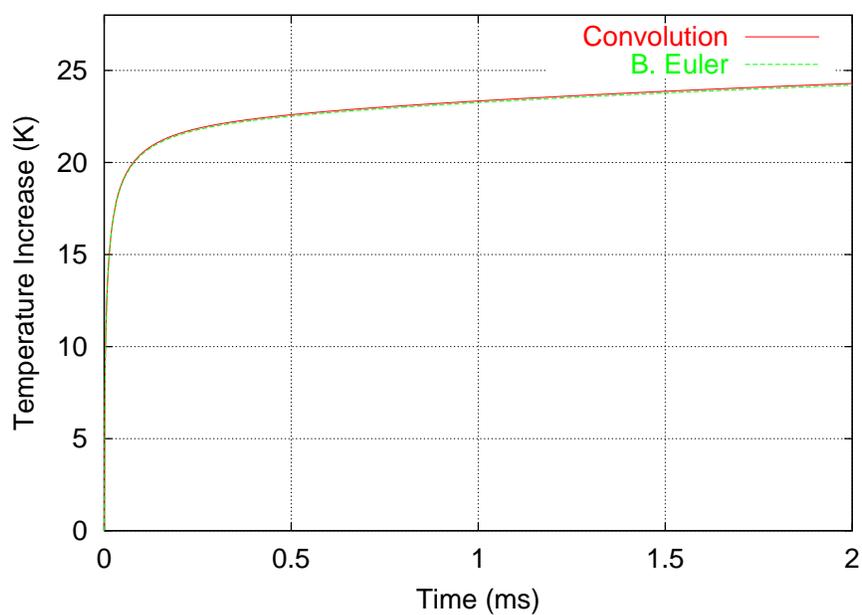


Figure 8.32: Transient of the MESFET temperature. The temperature shown is the difference with respect to the ambient temperature (300 K).

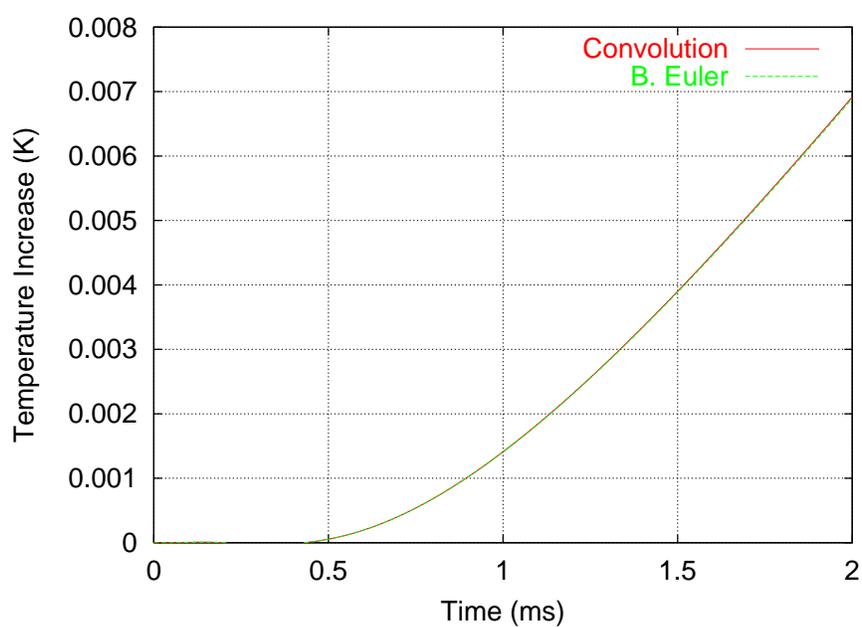


Figure 8.33: Transient of the substrate temperature. The temperature shown is the difference with respect to the ambient temperature (300 K).

Chapter 9

Conclusions and Future Research

9.1 Summary of Research and Original Contributions

Possibly one of the most important contributions of this work is the generalized state variable reduction formulation. This unified modeling view — along with the local reference node concept also implemented here for the first time — allows the simulation of QO systems integrating nonlinear circuits with electromagnetic structures and thermal modeling. Also, the state variable formulation adopted here enables an efficient, robust and very flexible software implementation, *i.e.* the generic evaluation technique, which was described in Section 7.4. This capacity is fundamental to reduce the development time of new system component models. All this was accomplished by re-designing the simulator engine of the preexisting Transim simulator program [2] using object oriented techniques. Only the parser and the output routines from the original program were kept, and they had been significantly modified to work with the new simulator engine and support new features. To this date, no other circuit simulator provides the same level of flexibility for the addition of new nonlinear device models and circuit analysis algorithms. The support for the local reference node concept for the first time in a circuit simulator described in Section 8.2 is also a contribution of this thesis. The local reference concept is fundamental for the analysis of spatially distributed circuits as well as for simultaneous thermal-electrical

simulations.

Another original contribution is the circuit formulation of Chapter 3. All the nonlinear circuit analysis techniques presented in this thesis are derived from the generic formulation in Chapter 3. This formulation provides a tool for the derivation of new circuit analysis techniques. It uses Rizzoli's state variable concept and thus inherits the associated advantages of this approach. The number of nonlinear unknowns resulting from our formulation is generally much smaller than the number of unknowns in conventional circuit analysis as it is shown in Section 3.2.

The convolution transient method developed by Ozkar in [13] was enhanced and re-implemented more efficiently in the newly designed Transim (Chapter 4). Convolution transient provides modeling of frequency-defined network elements with more accuracy than other methods, but as it was shown it is useful when the number of time steps to be simulated is not very large. The implementation of this circuit analysis technique was the first chronologically and allowed the transient simulation for the first time of a NLTL considering frequency-dependent attenuation in Section 4.7 and, also for the first time, the transient simulation of a quasi-optical grid amplifier in Section 8.3.4.

The wavelet approach developed in Chapter 5 is unique, and the software implementation in Transim is the most advanced wavelet-based nonlinear circuit simulator developed to date. Wavelet basis functions are ideally suited to expanding the response of systems with an overall coarse response but fine behavior in some regions as higher order and more localized basis functions can be concentrated on the regions where the response varies rapidly. The simulation examples using the NLTL and the QO grid amplifier are the most complex circuits ever simulated using wavelets. Nevertheless, our final conclusion is that circuit simulation wavelet techniques still require more research before they can achieve the same efficiency as time marching techniques.

Another contribution is the reduced state variable time marching scheme developed in Chapter 6. Both formulation and implementation in Transim is original from this thesis. It was shown that with this transient analysis it is possible to achieve more than an order of magnitude speedup in the simulation time compared with Spice. When combined with pole-zero modeling of EM structures, the simulations of QO systems using the method of Chapter 6 are several orders of magnitude faster than the simulations using convolution transient. Therefore, this is the preferred circuit analysis method if accurate pole-zero models for QO are available. We can conclude that the reduced

state variable time marching transient may become a standard method for the simulation of up to medium-sized microwave circuits given its definitive advantages for that type of problem.

Other contributions include the integration of thermal circuit elements based on the *Leeds thermal impedance matrix model* to enable electro-thermal simulations in Transim (Section 8.4). This was a joint effort with Dr. William Batty of Leeds University, in the UK. The simulation results in Section 8.4 show that thermal effects can not be neglected in quasi-optical systems.

Also the combination and the actual coding in Transim of the frequency spectrum generation technique of Section 2.5 with the frequency mapping technique in harmonic balance to allow the simulation of spectral regrowth using HB.

Other minor contributions include the combined use of OO techniques and off-the-shelf libraries to build a flexible and efficient circuit simulator and the combination and the actual coding in Transim of the frequency spectrum generation technique of Section 2.5 with the frequency mapping technique in harmonic balance to allow the simulations of spectral regrowth of QO systems using HB in Chapter 8. This was independently proposed as an alternative, but not implemented, in Reference [47].

9.2 Future Research

Most of the topics treated in this work are left open to new research. One possible topic for future research is the development and implementation in Transim of a Spice-like state variable transient analysis. The idea is to combine the robustness and flexibility of the state variable approach with the associated discrete modeling [3] used in Spice. This would allow the simulation of very large circuits, of the same order that Spice can handle and use the same model library already developed for the reduced formulations. The treatment of very large QO arrays will require this kind of analysis.

Another important topic for research is the solution to the problem of the transient analysis of circuits with widely different time constants (such as thermal-electrical). One alternative is to develop in Transim the capability to perform two or more coupled time marching transient simulations, each corresponding to a different section of the circuit with similar time constants. The time step in each coupled simulation is adjusted according to the time constants of the corresponding circuit section.

The issue of approximating frequency-defined functions in the time domain requires more research too. Possible next steps are to integrate and test more algorithms to perform the pole-zero approximations. Also, there are cases when the extra accuracy provided by convolution techniques is needed and the simulation times are acceptable. Then it is advisable to investigate other ways of performing convolution (perhaps using wavelets, such as in [27, 28]), in a way compatible with time-marching analysis in Transim.

The remaining of this section presents one idea to improve the circuit analysis using wavelets.

9.2.1 The Sliding Window Scheme

The sliding window scheme attempts to avoid the problem of obtaining a good guess for the state variable coefficients and at the same time reduce the number of unknowns that must be solved simultaneously. This is just an idea and more research and testing are needed to actually produce an efficient scheme. The idea is to ‘slide’ the time window in such a way that

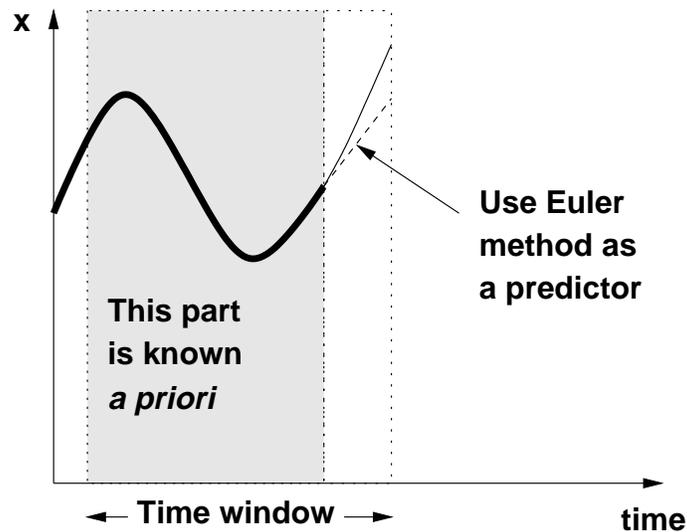


Figure 9.1: The sliding window scheme.

most of the solution inside the window is known, so the error function needs to be evaluated at a few points, see Fig. 9.1. The unknown part may be estimated using a predictor such as the BE method. The initial guess is then in general close to the final solution.

An outline of the complete method is as follows.

- Create the MNAM for $J = J_{max}$.
- Build the linear system in the wavelet domain for only the last samples of the window. $\mathbf{M}\mathbf{sv}$ will thus be reduced to a small fraction of the size if all the samples are considered.
- The unknowns are the last time samples of the state variables, the error function is built comparing the last coefficients of the port voltages.
- Assuming all state variables are zero before $t = 0$, start the calculation with $J = -1$ in all variables.
- For each circuit variable, verify that the coefficients in the $J + 1$ subspace level are zeros by calculating them assuming the coefficients in the other levels are unchanged. If they are not zero, increase J and recalculate all coefficients until this happens. If a circuit variable has a level of all-zero coefficients, decrease J there.
- Slide the window one time step. Calculate the new wavelet coefficients. Use a time marching method to produce an initial guess for them.
- Repeat the last two steps until all the simulation interval is covered.

The advantages of this method are:

- The nonlinear system to solve simultaneously is small, then
- the memory requirements are a lot smaller than for the previous method.
- It provides independent resolution at each port.
- It should be faster and more robust than the current approach.

There are many details that must be taken care of in order to implement this in Transim. For example, this method would require bidirectional wavelet transformations and storage of variables at different resolutions among others.

Appendix A

Transim's Graphical User Interface

A.1 Introduction

The simulation engine in Transim can be used in the traditional way as a stand-alone program, for example in batch jobs. In this mode, the program reads an input netlist, process its contents and writes the requested output files.

Transim also provides a Graphical User Interface (GUI), which is more convenient for interactive use of the program. This GUI is written using the Java language, so it can be used in every system where Java is supported. In this chapter we describe the different components of the GUI.

A.2 The Netlist Editor

The netlist editor is a simple text editor combined with a simulation manager. The editor window is shown in Figure A.1. Besides the normal editing commands, the editor provides buttons and keyboard shortcuts to analyze the netlist being edited and see the output of the simulation.

The editor can edit several files and handle multiple simulations at once by spawning multiple windows.

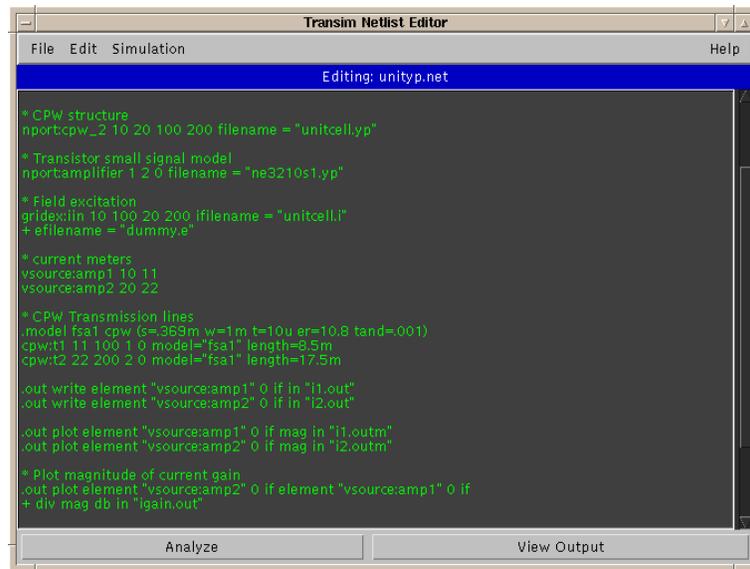


Figure A.1: Netlist Editor window.

A.3 The Analysis Window

The analysis window is used to show the progress of a simulation (Figure A.2). The upper sub-window displays important messages such as when the program starts or stops, and also warnings and errors that may occur during the simulation. The lower sub-window shows the progress of the simulation.

The buttons are self-explaining. The “Analyze” button changes to “Stop” when the engine is running.

A.4 The Output Viewer Window

This window is perhaps the most useful of all. It is shown in Figure A.3. The output file is displayed at the left. This file contains detailed information about the simulation.

At the right there is a list of files available for plotting. After selecting one or more of these files and depressing the “Plot” button, a plot window appears showing the desired data (see Figure A.4). Any number of plots can be requested. Also, the plot data is kept in memory by the plot window, so it is possible to re-run a simulation with different parameters and compare

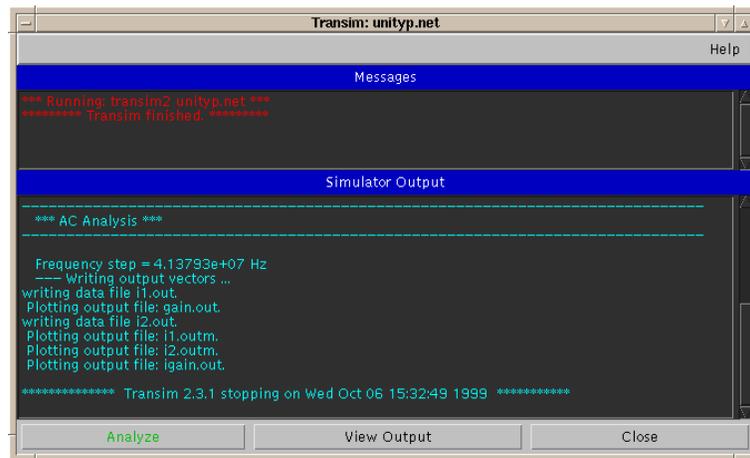


Figure A.2: Analysis window.

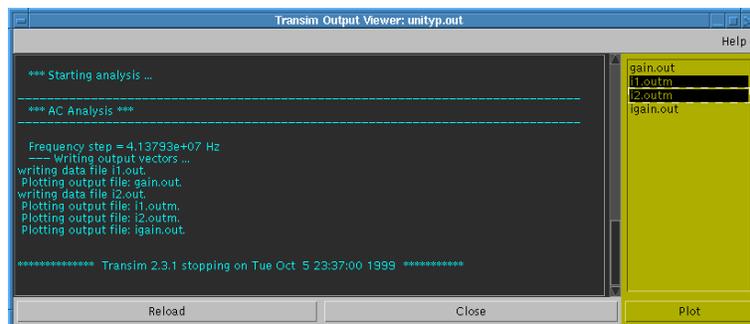


Figure A.3: Output Viewer window.

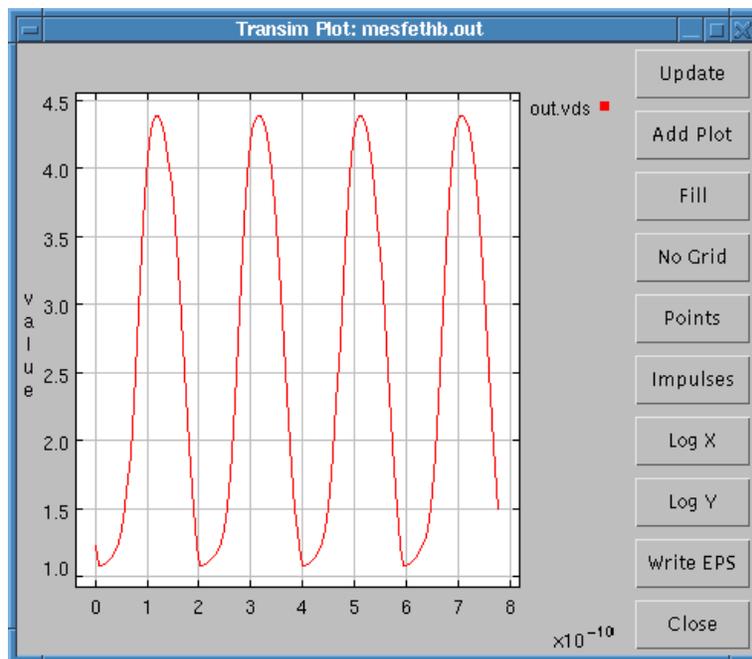


Figure A.4: Plot window.

the new and old graphs on the screen. An encapsulated postscript file can be generated pressing the corresponding button. There are also buttons to change to logarithmic scales and change the style of the graph.

There are several features provided in the plot window. For example, it is possible to zoom in or out the graph by dragging the left or right mouse button, respectively.

The plotting facility is provided by the `ptplot` [95] library.

Appendix B

Object Oriented Programming Basics

Object oriented programming (OOP) [61] provides a means for abstraction in both programming and design. OOP does not deal with programming in the sense of developing algorithms or data structures but it must be studied as a means for the organization of programs and, more generally, techniques for designing programs.

As the primary means for structuring a program or design, OOP provides *objects*. Objects may model real life entities, may function to capture abstractions of arbitrary complex phenomena, or may represent system artifacts such as stacks or graphics. Operationally, objects control the computation. From the perspective of program development, however, the most important characteristic of objects is not their behavior as such, but the fact that the behavior of an object may be described by an abstract characterization of its interface. Having such a characterization suffices for the design. The actual behavior of the object may be implemented later and refined according to the need. A *class* specifies the behavior of the objects which are its instances. Also, classes act as templates from which actual objects may be created. An *instance* of a class is an object belonging to that class. A procedure (or function) inside an object is called a *method*. A *message* to an object is a request to execute a method.

Inheritance is the mechanism which allows the reuse of class specifications. The use of inheritance results in a class hierarchy that, from an operational point of view, decides what is the method that will be selected in response to a message.

Finally, an important feature of OO languages is their support for *polymorphism*. This makes it possible to hide different implementations behind a common interface.

The notion of flow diagram in procedural programming is replaced in OOP by a set of objects which interact by sending messages to each other.

We will briefly review what are traditionally considered to be features and benefits of OOP. Both *information hiding* (also known as *encapsulation*) and *data abstraction* relieve the task of the programmer using existing OO code, since with these mechanisms the programmer's attention is no longer distracted by irrelevant implementation details. The flexible dispatching behavior of objects that lends them their polymorphic behavior is due to the dynamic binding of methods to messages. For the C++ language, polymorphic object behavior is effected by using *virtual functions*, for which in contrast to ordinary functions, the binding to an actual function takes place at run time and not at compile time.

Encapsulation promotes *modularity*, meaning that objects may be regarded as the building blocks of a complex system. Another advantage often attributed to the OOP is code reuse. Inheritance is an invaluable mechanism in this respect, since it enables the programmer to modify the behavior of a class of objects without requiring access to the source code.

Although an object oriented approach to program development indeed offers great flexibility, some of the problems it addresses are intrinsically difficult and cannot really be solved by mechanisms alone. For example, it is more likely to achieve a stable modularization when shifting focus from programming to design.

C++ virtual functions [71] can have big performance penalties such as extra memory accesses or the possibility of an unpredictable branch so pipelines can grind to a halt (note however, that some architectures have branch caches which can avoid this problem). There are several research projects which have demonstrated success at replacing virtual function calls with direct dispatches. Note however, that virtual functions are not always bad when it comes to performance. The important questions are: How much code is inside the virtual function? How often is it used? If there is a lot of code (*i.e.* more than 25 flops), then the overhead of the virtual function will be insignificant. But if there is a small amount of code and the function is called very often (*e.g.* inside a loop), then the overhead can be critical.

B.1 UML diagrams

The Unified Modeling Language (UML) [66, 84] is a language for specifying, visualizing, and constructing the artifacts of software systems as well as for business modeling. The goal of UML is to become a common language for creating models of object oriented computer software. It is used here to graphically illustrate the relationship of classes using what is called a class diagram. A class diagram is a graph of Classifier elements with connections indicating by their various static relationships (Figure 7.1). (Note that a “class” diagram may also contain interfaces, packages, relationships, and even instances, such as objects and links. Perhaps a better name would be “static structural diagram” but “class diagram” is shorter and its use is well established.) A class is drawn as a solid-outline rectangle with 3 compartments separated by horizontal lines. The top name compartment holds the class name and other general properties of the class (including stereotype); the middle list compartment holds a list of attributes; the bottom list compartment holds a list of operations. Either or both of the attribute and operation compartments may be suppressed. A separator line is not drawn for a missing compartment. If a compartment is suppressed, no inference can be drawn about the presence or absence of elements in it.

Each instance of type **Element**, for example, seems to contain an instance of type **ElementData**. This class relationship is indicated by the joining line. The relationship is composition — indicated by the solid diamond symbol. The arrowhead denotes that the relationship is navigable in only one direction, *i.e.*, **ElementData** does not know anything about **Element**. The inheritance relationship in UML is depicted by the triangular arrowhead and points to the base class. A line from the base of the arrowhead connects it to the derived classes, e.g. **Element** is derived from **NetListItem**.

Other forms of containment do not have whole/part implications and are called association relationships indicated by a line drawn between the participating classes. (This relationship will almost certainly be implemented using pointers unless it is very weak.) If the relationship between two classes is very weak (*i.e.* very little data is shared) then a dashed line is used. For example, in Figure 7.1, **ElementManager** somehow depends upon **Diode**. (In C++ the weak relationship is almost always implemented using an `#include`.)

An illustration showing examples for the notation is given in Figure B.1.

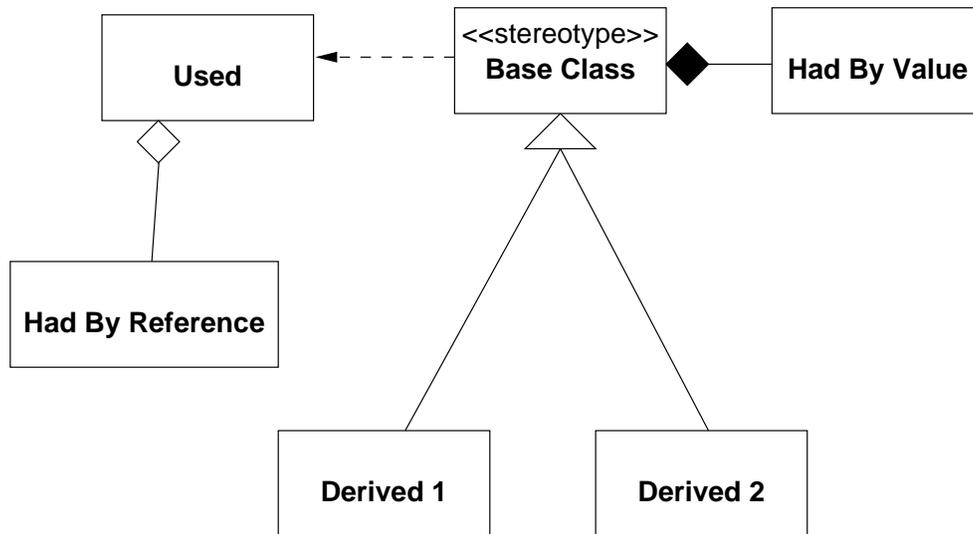


Figure B.1: Notation for a class diagram.

B.1.1 Collaboration Diagrams

The collaboration diagram [84] is part of the Unified Modeling Language (UML), a language for specifying, visualizing, and constructing the artifacts of software systems as well as for business modeling. The UML represents a collection of ‘best engineering practices’ that have proven successful in the modeling of large and complex systems. Behavior is implemented by sets of objects that exchange messages within an overall interaction to accomplish a purpose. To understand the mechanisms used in a design, it is important to see only the objects and the messages involved in accomplishing a purpose or a related set of purposes, projected from the larger system of which they are part for other purposes. Such a static construct is called a collaboration. A collaboration is a set of participants and relationships that are meaningful for a given set of purposes. The identification of participants and their relationships does not have global meaning. In the collaboration diagram, each box represents an object (in this case, either a terminal or an element). The lines between boxes represent associations and the arrows are messages. The order in which messages are sent is shown by the numbers.

Bibliography

- [1] M. B. Steer, J. F. Harvey, J. W. Mink, M. N. Abdulla, C. E. Christoffersen, H. M. Gutierrez, P. L. Heron, C. W. Hicks, A. I. Khalil, U. A. Mughal, S. Nakazawa, T. W. Nuteson, J. Patwardhan, S. G. Skaggs, M. A. Summers, S. Wang, and A. B. Yakovlev, "Global modeling of spatially distributed microwave and millimeter-wave systems," *IEEE Trans. Microwave Theory Tech.*, June 1999, pp. 830–839.
- [2] C. E. Christoffersen, M. B. Steer and M. A. Summers, "Harmonic balance analysis for systems with circuit-field interactions," *1998 IEEE Int. Microwave Symp. Dig.*, June 1998, pp. 1131–1134.
- [3] M. B. Steer, *Transient and Steady-State Analysis of Nonlinear RF and Microwave Circuits*, ECE603 class notes, August 15, 1996.
- [4] K. S. Kundert, J. K. White and A. Sangiovanni-Vincentelli, *Steady-state methods for simulating analog and microwave circuits*, Boston, Dordrecht, Kluwer Academic Publishers, 1990.
- [5] V. Rizzoli, A. Lipparini, A. Costanzo, F. Mastri, C. Ceccetti, A. Neri and D. Masotti, "State-of-the-Art Harmonic-Balance Simulation of Forced Nonlinear Microwave Circuits by the Piecewise Technique," *IEEE Trans. on Microwave Theory and Tech.*, Vol. 40, No. 1, Jan 1992.
- [6] M. Valtonen and T. Veijola, "A microcomputer tool especially suited for microwave circuit design in frequency and time domain," *Proc. URSI/IEEE National Convention on Radio Science*, Espoo, Finland, 1986, p. 20,
- [7] M. Valtonen, P. Heikkilä, A. Kankkunen, K. Mannersalo, R. Niutanen, P. Stenius, T. Veijola and J. Virtanen, "APLAC - A new approach to

- circuit simulation by object orientation,” *10th European Conference on Circuit Theory and Design Dig.*, 1991.
- [8] K. Mayaram and D. O. Pederson, “CODECS: an object-oriented mixed-level circuit and device simulator,” *1987 IEEE Int. Symp. on Circuits and Systems Digest*, 1987, pp 604–607.
- [9] A. Davis, “An object-oriented approach to circuit simulation,” *1996 IEEE Midwest Symp. on Circuits and Systems Dig.*, 1996, pp 313–316.
- [10] B. Melville, P. Feldmann and S. Moinian, “A C++ environment for analog circuit simulation,” *1992 IEEE Int. Conf. on Computer Design: VLSI in Computers and Processors*.
- [11] P. Carvalho, E. Ngoya, J. Rousset and J. Obregon, “Object-oriented design of microwave circuit simulators,” *1993 IEEE MTT-S Int. Microwave Symp. Digest*, June 1993, pp 1491–1494.
- [12] A. I. Khalil and M. B. Steer “Circuit theory for spatially distributed microwave circuits,” *IEEE Trans. on Microwave Theory and Techn.*, Vol. 46, Oct. 1998, pp 1500–1503.
- [13] M. Ozkar, *Transient Analysis of Spatially Distributed Microwave Circuits Using Convolution and State Variables*, M. S. Thesis, Department of Electrical and Computer Engineering, North Carolina State University.
- [14] A. R. Djordjevic and T. K. Sarkar, “Analysis of time response of lossy multiconductor transmission line networks,” *IEEE Trans. on Microwave Theory and Techn.*, Vol. MTT-35, Oct. 1987, pp. 898–908.
- [15] D. Winkelstein, R. Pomerleau and M. B. Steer, “Transient simulation of complex, lossy, multi-port transmission line networks with nonlinear digital device termination using a circuit simulator,” *Conf. Proc. IEEE SOUTHEASTCON*, Vol. 3, pp. 1239–1244.
- [16] T. J. Brazil, “Causal convolution—a new method for the transient analysis of linear systems at microwave frequencies,” *IEEE Trans. on Microwave Theory and Techn.*, Vol. 43, Feb. 1995, pp. 315–23.

- [17] M. S. Basel, M. B. Steer and P. D. Franzon, "Simulation of high speed interconnects using a convolution-based hierarchical packaging simulator," *IEEE Trans. on Components, Packaging, and Manufacturing Techn.*, Vol. 18, February 1995, pp. 74–82.
- [18] J. E. Schutt-Aine and R. Mittra, "Nonlinear transient analysis of coupled transmission lines," *IEEE Trans. on Circuits and Systems*, Vol. 36, Jul. 1989, pp. 959–967.
- [19] P. Stenius, P. Heikkilä and M. Valtonen, "Transient analysis of circuits including frequency-dependent components using transgyrator and convolution," *Proc. of the 11th European Conference on Circuit Theory and Design*, Part II, 1993, pp. 1299–1304.
- [20] R. Griffith and M. S. Nakhla, "Mixed frequency/time domain analysis of nonlinear circuits," *IEEE Trans. on Computer Aided Design*, Vol. 11, Aug. 1992, pp. 1032–43.
- [21] P. K. Chan, Comments on "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. on Computer Aided Design*, Vol. 10, Aug. 1991, pp. 1078–79.
- [22] M. Celik, O. Ocali, M. A. Tan, and A. Atalar, "Pole-zero computation in microwave circuits using multipoint Padé approximation," *IEEE Trans. on Circuits and Systems*, Jan. 1995, pp. 6–13.
- [23] E. Chiprout and M. Nakhla, "Fast nonlinear waveform estimation for large distributed networks," *1992 IEEE MTT-S Int. Microwave Symp. Digest*, Vol. 3, Jun. 1992, pp. 1341–1344.
- [24] R. J. Trihy and Ronald A. Rohrer, "AWE macromodels for nonlinear circuits," *Proceedings of the 36th Midwest Symposium on Circuits and Systems*, Vol. 1, Aug. 1993, pp. 633–636.
- [25] W. T. Beyene and J. E. Schutt-Aineé, "Efficient Transient Simulation of High-Speed Interconnects Characterized by Sampled Data", *IEEE Trans. on Components, Packaging and Manufacturing Techn. — Part B*, Vol. 21, No. 1, Feb. 1998, pp. 105–114.

- [26] D. Borisovich and J. E. Schutt-Aineé, “Optimal Transient Simulation of Transmission Lines,” *IEEE Trans. on Circuits and Systems—I: Fundamental Theory and Applications*, Vol. 43, No. 2, Feb. 1996, pp. 110–121.
- [27] W. Leung and F. Chang, “Transient analysis via fast wavelet-based convolution,” *1995 ISCAS Symp. Digest*, Vol. 3, pp. 1884–1887, 1995.
- [28] W. Leung and F. Chang, “Wavelet-based waveform relaxation simulation of lossy transmission lines,” *1996 ISCAS Symp. Digest*, Vol. 4, pp. 739–742, 1996.
- [29] A. Al-Rawi and M. Devaney, “Wavelets and power system transient analysis,” *1998 IEEE Instr. and Meas. Techn. Conf. Digest*, Vol. 2, pp. 1331–1334, 1998.
- [30] T. Hisakado and K. Okumura, “Steady states prediction in nonlinear circuit by wavelet transform,” *1999 ISCAS Symp. Digest*, 1999.
- [31] C. M. Arturi, A. Gandelli, S. Leva, S. Marchi and A. P. Morando, “Multiresolution analysis of time-variant electrical networks,” *1999 ISCAS Symp. Digest*, 1999.
- [32] W. Cai and J. Wang, “Adaptive multiresolution collocation methods for initial boundary value problems of nonlinear PDEs,” *SIAM J. Numer. Anal.*, Vol. 33, No. 3, pp. 937–970, June 1996.
- [33] D. Zhou, N. Chen and W. Cai, “A fast wavelet collocation method for high-speed VLSI circuit simulation,” *1995 IEEE/ACM ICCAD Symp. Digest*, pp 115–122, 1995.
- [34] D. Zhou, X. Li, W. Zhang and W. Cai, “Nonlinear circuit simulation based on adaptive wavelet method,” *1997 ISCAS Symp. Digest*, Vol. 3, pp. 1720–1723, 1997.
- [35] W. Cai and J. Wang, “An adaptative spline wavelet ADI (SW-ADI) method for two-dimensional reaction-diffusion equations,” *J. of Computational Physics*, No. 139, pp. 92–126, 1998.
- [36] D. Zhou and W. Cai, “A fast wavelet collocation method for high-speed VLSI circuit simulation,” *IEEE Trans. on Circuits and Systems—I: Fundamental Theory and Appl.*, Vol. 46, pp 920–930, Aug. 1999.

- [37] D. Zhou, W. Cai and W. Zhang, "An adaptive wavelet method for nonlinear circuit simulation," *IEEE Trans. on Circuits and Systems—I: Fundamental Theory and Appl.*, Vol. 46, pp 931–938, Aug. 1999.
- [38] A. Wenzler and E. Lueder, "Analysis of the periodic steady-state in nonlinear circuits using an adaptive function base," *1999 IEEE Int. Symposium on Circuits and Systems Digest*.
- [39] R. A. Lippert, T. A. Arias and A. Edelman, "Multiscale computation with interpolating wavelets," *J. of Computational Physics*, No. 140, pp. 278–310, 1998.
- [40] A. Graps, "An introduction to wavelets," *IEEE Computational Science and Engineering*, Vol. 2, No. 2, pp. 50–61, Summer 1995.
- [41] I. Daubechies, "Ten Lectures on Wavelets," *SIAM Publication*, Philadelphia, 1992.
- [42] S. Bertoluzza, "An adaptive collocation method based on interpolating wavelets," *Multiscale Wavelet Methods for Partial Differential Equations*, Academic Press, 1997.
- [43] P. Debeve F. Odeh and A. E. Ruehli, "Waveform Techniques" *Circuit Analysis, Simulation and Design*, North-Holland, 1994.
- [44] M. S. Nakhla and J. Vlach, "A Piecewise Harmonic Balance Technique for Determination of Periodic Response of Nonlinear Systems," *IEEE Trans. on Circuits and Systems*, Vol CAS-23, No. 2, Feb 1976.
- [45] M. B. Steer, C. Chang and G. W. Rhyne, "Computer-Aided Analysis of Nonlinear Microwave Circuits Using Frequency-Domain Nonlinear Analysis Tech.: The State of the Art," *Int. Journal of Microwave and Millimeter-Wave Computer-Aided Engineering*, Vol. 1, No. 2, 181–200, 1991.
- [46] N. Borges de Carvalho and J. C. Pedro, "Multitone frequency-domain simulation of nonlinear circuits in large- and small-signal regimes," *IEEE MTT*, Vol. 46, no. 12, pp. 2016–2024, 1998.
- [47] V. Borich, J. East and G. Haddad, "An efficient Fourier transform algorithm for multitone harmonic balance," *IEEE Transactions on Microwave Theory and Tech.*, Vol. 47, no. 2, Feb. 1999, pp 182–188.

- [48] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, Van Nostrand Reinhold, 1994.
- [49] T. J. Brazil, "A new method for the transient simulation of causal linear systems described in the frequency domain," *1992 IEEE MTT-S Int. Microwave Symp. Digest*, June 1992, pp. 1485–1488.
- [50] P. Perry and T. J. Brazil, "Hilbert-transform-derived relative group delay," *IEEE Trans. on Microwave Theory and Techn.*, Vol 45, Aug. 1997, pt. 1, pp. 1214–1225.
- [51] C. E. Christoffersen, S. Nakazawa, M. A. Summers, and M. B. Steer, "Transient analysis of a spatial power combining amplifier", *1999 IEEE MTT-S Int. Microwave Symp. Dig.*, June 1999, pp. 791–794.
- [52] M. Frigo and S. G. Johnson, *FFTW User's Manual*, Massachusetts Institute of Technology, September 1998.
- [53] C. Gordon, T. Blazeck and R. Mittra, "Time domain simulation of multiconductor transmission lines with frequency-dependent losses," *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, Vol. 11, Nov. 1992 pp. 1372–87.
- [54] M. G. Case, *Nonlinear transmission lines for picosecond pulse, impulse and millimeter-wave harmonic generation*, Ph.D Dissertation, Department of Electrical and Computer Engineering, University of California, Santa Barbara, California, U.S.A., 1993.
- [55] M. J. W. Rodwell, M. Kamegawa, R. Yu, M. Case, E. Carman and K. S. Giboney, "GaAs nonlinear transmission lines for picosecond pulse generation and millimeter-wave sampling," *IEEE Trans. on Microwave Theory and Techn.*, Vol. 39, July 1991, pp. 1194–1204.
- [56] H. Shi, C. W. Domier and N. C. Luhmann, "A monolithic nonlinear transmission line system for the experimental study of lattice solutions," *J. of Applied Physics*, Vol 4., August 1995, pp. 2558–64.
- [57] Compact Software, *Microwave Harmonica Elements Library*, (1994).
- [58] A. Brambilla, D. D'Amor and M. Pillan, "Convergence improvements of the harmonic balance method," *Proceedings IEEE Int. Symposium on*

Circuits and Systems, Vol. 4 1993, Publ. by IEEE, IEEE Service Center, Piscataway, NJ, USA. p 2482–2485.

- [59] R. S. Bain, *NNES user's manual*, 1993.
- [60] P. J. C. Rodrigues, *Computer Aided Analysis of Nonlinear Microwave Circuits*, Artech House, 1998.
- [61] A. Eliëns, *Principles of object-oriented software development*, Addison-Wesley, 1995.
- [62] R. C. Martin. “The dependency inversion principle,” *C++ Report*, May 1996.
- [63] R. C. Martin, “The Open Closed Principle,” *C++ Report*, Jan. 1996.
- [64] R. C. Martin, “The Liskov Substitution Principle,” *C++ Report*, March 1996.
- [65] R. C. Martin, “The Interface Segregation Principle,” *C++ Report*, Aug 1996.
- [66] R. C. Martin, “UML Tutorial: Part 1 — Class Diagrams,” Engineering Notebook Column, *C++ Report*, Aug. 1997.
- [67] A. D. Robison, “C++ Gets Faster for Scientific Computing,” *Computers in Physics*, Vol. 10, pp. 458–462, 1996.
- [68] J. R. Cary and S. G. Shasharina, “Comparison of C++ and Fortran 90 for Object-Oriented Scientific Programming,” Available from Los Alamos National Laboratory as Report No. LA-UR-96-4064.
- [69] The Object Oriented Numerics Page, <http://oonumerics.org/>.
- [70] Silicon Graphics, *Standard Template Library Programmer's Guide*, <http://www.sgi.com/Technology/STL/>.
- [71] T. Veldhuizen, Tech. for Scientific C++ - Version 0.3, Indiana University, Computer Science Department, 1999. (<http://extreme.indiana.edu/tveldhui/papers/Tech./>)

- [72] A. Griewank, D. Juedes and J. Utke, "Adol-C: A Package for the Automatic Differentiation of Algorithms Written in C/C++," *ACM TOMS*, Vol. 22(2), pp. 131–167, June 1996.
- [73] R. Pozo, MV++ v. 1.5a, *Reference Guide*, National Institute of Standards and Technology, 1997.
- [74] K. S. Kundert and A. Songiovanni-Vincentelli, *Sparse user's guide - a sparse linear equation solver*, Dept. of Electrical Engineering and Computer Sciences, University of California, Berkeley, Calif. 94720, Version 1.3a, Apr 1988.
- [75] B. Speelpenning. *Compiling Fast Partial Derivatives of Functions Given by Algorithms*, Ph.D. thesis (Under the supervision of W. Gear), Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana-Champaign, Ill., January 1980.
- [76] T. F. Coleman and G. F. Jonsson, "The Efficient Computation of Structured Gradients using Automatic Differentiation," *Cornell Theory Center Technical Report CTC97TR272*, April 28, 1997
- [77] S. M. S. Imtiaz and S. M. El-Ghazaly, "Global modeling of millimeter-wave circuits: electromagnetic simulation of amplifiers," *IEEE Trans. on Microwave Theory and Tech.*, vol 45, pp. 2208–2217. Dec. 1997.
- [78] C.-N. Kuo, R.-B. Wu, B. Houshmand, and T. Itoh, Modeling of microwave active devices using the FDTD analysis based on the voltage-source approach, *IEEE Microwave Guided Wave Lett.*, Vol. 6, pp. 199–201, May 1996.
- [79] E. Larique, S. Mons, D. Baillargeat, S. Verdeyme, M. Aubourg, P. Guillon, and R. Quere, "Electromagnetic analysis for microwave FET modeling," *IEEE microwave and guided wave letters*, Vol 8, pp. 41–43, Jan. 1998.
- [80] T. W. Nuteson, H. Hwang, M. B. Steer, K. Naishadham, J.W.Mink, and J. Harvey, "Analysis of finite grid structures with lenses in quasi-optical systems," *IEEE Trans. Microwave Theory Tech.*, pp. 666–672, May 1997.

- [81] M. B. Steer, M. N. Abdullah, C. Christoffersen, M. Summers, S. Nakazawa, A. Khalil, and J. Harvey, "Integrated electro-magnetic and circuit modeling of large microwave and millimeter-wave structures," *Proc. 1998 IEEE Antennas and Propagation Symp.*, pp. 478–481, June 1998.
- [82] J. Kunisch and I. Wolff, "Steady-state analysis of nonlinear forced and autonomous microwave circuits using the compression approach," *Int. J. of Microwave and Millimeter-Wave Computer-Aided Engineering*, Vol. 5, No. 4, pp. 241–225, 1995
- [83] T. H. Cormen, C. E. Leiserson and R. L. Rivest *Introduction to Algorithms*, The MIT Press, McGraw-Hill Book Company, 1990.
- [84] Rational Software, UML Resources, <http://www.rational.com/>.
- [85] H. S. Tsai, M. J. W. Rodwell and R. A. York, "Planar amplifier array with improved bandwidth using folded-slots," *IEEE Microwave and Guided Wave Letters*, Vol. 4, April 1994, pp. 112–114.
- [86] M. B. Steer, M. N. Abdullah, C. Christoffersen, M. Summers, S. Nakazawa, A. Khalil, and J. Harvey, "Integrated electro-magnetic and circuit modeling of large microwave and millimeter-wave structures," *Proc. 1998 IEEE Antennas and Propagation Symp.*, pp. 478–481, June 1998.
- [87] M. N. Abdulla, U.A. Mughal, and M B. Steer, "Network Characterization for a Finite Array of Folded-Slot Antennas for Spatial Power Combining Application," *Proc. 1999 IEEE Antennas and Propagation Symp.*, July 1999.
- [88] U. A. Mughal, "Hierarchical approach to global modeling of active antenna arrays," *M.S. Thesis*, North Carolina State University, 1999.
- [89] M. A. Summers, C. E. Christoffersen, A. I. Khalil, S. Nakazawa, T. W. Nuteson, M. B. Steer and J. W. Mink, "An integrated electromagnetic and nonlinear circuit simulation environment for spatial power combining systems," *1998 IEEE MTT-S Int. Microwave Symp. Dig.*, June 1998, pp. 1473–1476.

- [90] H. Gutierrez, C. E. Christoffersen and M. B. Steer, "An integrated environment for the simulation of electrical, thermal and electromagnetic interactions in high-performance integrated circuits," *Proc. IEEE 6th Topical Meeting on Electrical Performance of Electronic Packaging*, Sept. 1999.
- [91] W. Batty, C. E. Christoffersen, S. David, A. J. Panks, R. G. Johnson, C. M. Snowden and M. B. Steer, "Electro-thermal cad of power devices and circuits with fully physical time-dependent thermal modelling of complex 3-d systems," submitted to the *IEEE Trans. on Component and Packaging Technologies*.
- [92] W. Batty, C. E. Christoffersen, S. David, A. J. Panks, R. G. Johnson, C. M. Snowden and M. B. Steer, "Fully physical, time-dependent thermal modelling of complex 3-dimensional systems for device and circuit level electro-thermal CAD," submitted to *Semi-Therm XVII*, San Jose, March 2001.
- [93] W. Batty, C. E. Christoffersen, S. David, A. J. Panks, R. G. Johnson, C. M. Snowden and M. B. Steer, "Predictive microwave device design by coupled electro-thermal simulation based on a fully physical thermal model," *EDMO 2000*, Glasgow UK, November 2000.
- [94] W. Batty, C. E. Christoffersen, S. David, A. J. Panks, R. G. Johnson, C. M. Snowden and M. B. Steer, "Steady-state and transient electro-thermal simulation of power devices and circuits based on a fully physical thermal model," *THERMINIC 2000 Digest*, Budapest, September 2000.
- [95] Ptplot. <http://ptolemy.eecs.berkeley.edu/java/ptplot>.