

Nonlinear Transient Analysis Based on Power Waves and State Variables

by
Muhammad Ershadul Kabir

A Thesis
Presented to Lakehead University
in Partial Fulfilment of the Requirement for the Degree of
Master of Science
in
Electrical and Computer Engineering

Thunder Bay, Ontario, Canada

May 2010



Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-71769-1
Our file *Notre référence*
ISBN: 978-0-494-71769-1

NOTICE:

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

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

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

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

AVIS:

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

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

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

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


Canada

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

In the name of Allah, the beneficent the merciful

Abstract

A new approach for transient analysis of nonlinear circuits based on the nonlinear device state variables and power waves at their ports is described and implemented in a free circuit simulator named *fREEDA*. The analysis is formulated in terms of power wave quantities instead of voltages and currents. The circuit is partitioned into linear and nonlinear parts and a relaxation approach is used to decouple the calculations in each part. The wave equations are combined with the parameterized nonlinear device models already available in *fREEDA*. A direct consequence of iterating power waves is that iterations can never diverge to infinity. Another potential advantage of this approach is that no large matrix decompositions are necessary at every time step. Many of the calculations in this approach can be performed concurrently, so the method could be implemented in a parallel computer system easily.

This thesis covers the theoretical background, literature review and formulation of the proposed method. Convergence conditions are analyzed. Modifications to improve the robustness and convergence rate of the relaxation approach are investigated. Simulation results of several circuits containing both single and multiple port nonlinear devices are used to evaluate the performance of the proposed approach. Finally the road map for future work is presented.

Biographical Summary

Muhammad Ershadul Kabir was born in Chittagong, Bangladesh on July 15, 1982. He received the Bachelor of Science in Electrical and Electronic Engineering from Bangladesh University of Engineering and Technology (BUET) in Dhaka, Bangladesh in June, 2005. From July, 2005 to August, 2008 he worked in Motorola Telecommunication Bangladesh Pvt. Ltd. as System Engineer. During this time, he designed many SDH and PDH communication networks for different Wireless and PSTN operators in Bangladesh.

In September 2008 he enrolled in the Masters program of Electrical and Computer Engineering in Lakehead University and move in Canada along with his wife. His research interests include Computer Aided Design (CAD) of Circuit and systems, Simulation Techniques and Algorithms, Parallel computing system and parallel Implementation of CAD tools, Implementation of CAD tools in Graphics processing unit (GPU), Analog and mixed-signal circuit design, VLSI circuit design. He is a student member of the Institute of Electrical and Electronics Engineers (IEEE).

Acknowledgements

I would like to utter my deep gratitude and appreciation to my supervisor Dr. Carlos E. Christoffersen for his magnanimous support and patient guidance throughout my graduate studies at Lakehead University. It is my privilege to study in his research group.

I would also like to express my sincere admiration to Dr. Ehsan Atoofian for his co-supervision in my research work. Appreciations are also due to Dr. Rachid Benlamri, Dr. Ali Manzak and Dr. Samuel Pichardo for their enthusiastic instructions during my course study.

A very thanks to all my graduate student colleagues. First to Mr. Ryan Plater for helping me in my research work, then to everyone in the research group and pursued graduate courses with me.

Finally, I would like to thank my wife, Nusrat Sadia for her immense love and inspiration. Without her support it was not possible to continue my graduate studies. Many love to our newborn baby girl Sunehra. And many thanks to my parents for their heartiest support and encouragement.

Muhammad Ershadul Kabir

mekabir@lakeheadu.ca

Contents

List of Figures	viii
List of Tables	xii
List of Algorithms	xiii
List of Symbols	xiv
List of Abbreviations	xiv
1 Introduction	2
1.1 Objectives and Motivations of The Presented Study	2
1.2 Original Contributions	3
1.3 Thesis Outline	4
2 Literature Review	5
2.1 Basic Theory of Waves	5
2.1.1 Voltage Wave	6
2.1.2 Power Wave	7
2.2 Basic Concepts of Wave Digital Filter	7
2.2.1 Signal-Flow Graph	8
2.2.2 Frequency Variable	10
2.2.3 Signal Parameters	12
2.3 Building Blocks	12
2.3.1 Transformation of Elements and Sources	12

2.3.2	Circulators	15
2.3.3	Interconnections and Adaptors	15
2.3.4	Scattering Matrix	21
2.3.5	Simulation of a simple <i>RLC</i> circuit	22
2.3.6	General procedure	27
2.4	Single Nonlinearity	28
2.4.1	Nonlinear Resistances	28
2.4.2	Nonlinear Elements Treated as Switch	32
2.5	Multiple Nonlinearity	34
2.5.1	Jacobi's Method to cut DFL	35
2.5.2	Newton's Method for strong nonlinearity	38
2.6	Relaxation-Based Electrical Analysis	42
2.6.1	Timing Simulation	45
2.6.2	Iterated Timing Analysis	45
2.6.3	Waveform Relaxation Techniques	47
2.6.4	Simulation of a simple <i>RLC</i> circuit	48
2.7	Minimum Polynomial Extrapolation	52
3	Formulation	56
3.1	Generalized Circuit Formulation	56
3.1.1	Linear Network	56
3.1.2	Nonlinear Network	58
3.1.3	Error Function Formulation	59
3.2	Transformation to Power Waves	59
3.3	Fixed-Point Iterative scheme	61
3.3.1	Nonlinearity	61
3.3.2	Conditions for Local Convergence	64
3.4	Treating Nonlinearity by Newton-Jacobi method	65
3.5	Pseudo-Transient Approach	67
3.5.1	Timing Simulation of Pseudo-Circuit	67
3.5.2	Relaxation Factor	71

3.6	Application of MPE in relaxation method	71
3.6.1	Simplified Pseudo-code	74
3.7	Reflected Power Considerations	74
4	Simulation Results and Discussions	76
4.1	Single MESFET Amplifier	77
4.1.1	Simulation Results and Outputs	77
4.1.2	Convergence Rate Analysis	80
4.2	X-band MMIC LNA	84
4.2.1	Simulation Results and Outputs	85
4.2.2	Convergence Rate Analysis	89
4.3	Colpitts Oscillator	90
4.3.1	Simulation Results and Outputs	90
4.3.2	Convergence Rate Analysis	93
4.4	Soliton Line, a Nonlinear Transmission Line	97
4.4.1	Simulation Results and Outputs	99
4.4.2	Convergence Rate Analysis	99
4.5	Multi-MMIC-Soliton Line	104
4.5.1	Simulation Results and Outputs	104
4.5.2	Convergence Rate Analysis	114
4.6	Summing Amplifier	115
4.6.1	Simulation Results and Outputs	115
4.6.2	Steady-state Oscillations	118
5	Conclusion and Future Research	122
5.1	Summary of Research Works and Original Contributions	122
5.2	Future Work	123
5.2.1	Waveform Relaxation of Power Waves	124
A	Newton's Method	126
B	Fixed Point Iteration	128

C State Variable Approach	129
D Modified Nodal Analysis	132
E Example Circuit Netlists	136
E.1 Single MESFET Amplifier	136
E.2 X-band MMIC LNA	137
E.3 Colpitts Oscillator	139
E.4 Soliton Line	140
E.5 Summing Amplifier	145

List of Figures

2.1	A Terminated Lossless Transmission Line	6
2.2	SFG of an inductor and a capacitor	9
2.3	DFL by a LC filter	9
2.4	Mapping of frequencies in different domain	11
2.5	Some major one port elements and their WDF domain realization	14
2.6	WDF realization of a four port circulator	15
2.7	WDF domain realization of 3 port parallel adaptor	16
2.8	WDF domain realization of m port parallel adaptor	18
2.9	WDF domain realization of a parallel adaptor with n_p port <i>reflection-free</i>	19
2.10	WDF domain realization of m port series adaptor	20
2.11	A simple RLC circuit to demonstrate the procedure of WDF realization	22
2.12	WDF realization of the simple RLC circuit with adaptors	23
2.13	WDF realization of a simple RLC circuit with Scattering Matrix	26
2.14	WDF realization of a circuit with Nonlinear Resistor [30]	31
2.15	<i>current – voltage</i> characteristics of Nonlinear resistor [30]	31
2.16	Plot of wave characteristics of Nonlinear resistor defined by Eq. (2.38).	32
2.17	WDF realization of a diode treated as switch	33
2.18	WDF model for iterative solving of DFL	35
2.19	Impedence to cut DFL	36
2.20	Lossless nonlinear transmission line	37
2.21	WDF realization of nonlinear transmission line by modified Jacobi’s method	38
2.22	Circuit partitioning	39
2.23	WR example with a simple RLC circuit	49

2.24	Outputs for the example in Fig. 2.23	51
3.1	Network Partition	57
3.2	Transformation to power waves	60
3.3	Iterative scheme to calculate the value of waves	62
3.4	Circuit partition with pseudo-circuit	68
3.5	Discretized model of a linear capacitor using backward Euler formula	69
3.6	Actual time step and pseudo-time points in timing analysis of pseudo-circuit . . .	70
3.7	Convergence improvement with relaxation factor over plain relaxation	72
4.1	Single MESFET amplifier	78
4.2	Outputs from Wave_Transient method single MESFET amplifier	80
4.3	Outputs from Wave_Transient2 method for single MESFET amplifier	81
4.4	Outputs from Pseudo_Transient method for single MESFET amplifier	82
4.5	Convergence comparison from Wave_Transient method for single MESFET amplifier	82
4.6	Convergence comparison from Wave_Transient2 method for single MESFET am- plifier	83
4.7	Convergence comparison from Pseudo_Transient method for single MESFET am- plifier	83
4.8	X-band MMIC LNA (LMA 411)	84
4.9	Outputs from Wave_Transient method for X-band MMIC LNA	86
4.10	Outputs from Wave_Transient2 method for X-band MMIC LNA	87
4.11	Outputs from Pseudo_Transient method for X-band MMIC LNA	88
4.12	Convergence comparison from Wave_Transient method for X-band MMIC LNA . .	89
4.13	Convergence comparison from Wave_Transient2 method for X-band MMIC LNA .	89
4.14	Convergence comparison from Pseudo_Transient method for X-band MMIC LNA .	90
4.15	Colpitts Oscillator	91
4.16	Outputs from Wave_Transient method for Colpitts Oscillator	93
4.17	Outputs from Wave_Transient2 method for Colpitts Oscillator	94
4.18	Outputs from Pseudo_Transient method for Colpitts Oscillator	95
4.19	Convergence comparison from Wave_Transient method for Colpitts Oscillator . . .	95

4.20	Convergence comparison from Wave_Transient2 method for Colpitts Oscillator . . .	96
4.21	Convergence comparison from Pseudo_Transient method for Colpitts Oscillator . . .	96
4.22	Model of the Soliton Line	97
4.23	Outputs from Wave_Transient method for Soliton Line	100
4.24	Outputs from Wave_Transient2 method for Soliton Line	101
4.25	Outputs from Pseudo_Transient method for Soliton Line	102
4.26	Convergence comparison from Wave_Transient method for Soliton Line	103
4.27	Convergence comparison from Wave_Transient2 method for Soliton Line	103
4.28	Convergence comparison from Pseudo_Transient method for Soliton Line	104
4.29	Multi MMIC Soliton Line	105
4.30	Output2 from Wave_Transient method without the clamping circuit	107
4.31	Output2 from Wave_Transient method for Multi-MMIC-Soliton Line	108
4.32	Output5 from Wave_Transient method for Multi-MMIC-Soliton Line	109
4.33	Output2 from Wave_Transient2 method for Multi-MMIC-Soliton Line	110
4.34	Output5 from Wave_Transient2 method for Multi-MMIC-Soliton Line	111
4.35	Output2 from Pseudo_Transient method for Soliton Line	112
4.36	Output5 from Pseudo_Transient method for Soliton Line	113
4.37	Convergence comparison from Wave_Transient method for Multi-MMIC-Soliton Line	114
4.38	Convergence comparison from Wave_Transient2 method for Multi-MMIC-Soliton Line	114
4.39	Convergence comparison from Pseudo_Transient method for Multi-MMIC-Soliton Line	115
4.40	741 Operational Amplifier Circuit and Summing Amplifier	116
4.41	Outputs from Wave_Transient method for Summing Amplifier	118
4.42	Outputs from Wave_Transient2 method for Summing Amplifier	119
4.43	Outputs from Pseudo_Transient method for Summing Amplifier	120
4.44	Steady-state oscillation from different sequences	121
C.1	Relation between v_d and i_d	130
C.2	Relation between x_{sv} and i_d	130

C.3	Relation between x_{sv} and v_d	131
D.1	Example circuit to explain MNA	133

List of Tables

4.1	Summary of Simulation Results for Single MESFET amplifier	79
4.2	Summary of Simulation Results for X-band MMIC LNA	85
4.3	Summary of Simulation Results for Colpitts Oscillator	92
4.4	Summary of Simulation Results for Soliton Line	98
4.5	Summary of Simulation Results for Multi-MMIC-Soliton Line	106
4.6	Summary of Simulation Results for Summing Amplifier	117

List of Algorithms

2.6.1 Nonlinear Gauss-Jacobi Algorithm	44
2.6.2 Nonlinear Gauss-Seidel Algorithm	44
2.6.3 WR Gauss-Seidel Algorithm for Solving Eq. (2.53)	48
3.5.1 Algorithm for Timing Analysis of Power Waves using Relaxation Factor	72
3.6.1 Algorithm of Fixed-point Iterative Scheme of Power Waves	74

List of Symbols

- [**S**] – Scattering Parameter Matrix.
- Z_L – Load impedance.
- Z_0 – Reference Impedance.
- β – Phase Constant.
- Γ – Reflection Coefficient.
- v^+ – instantaneous voltage wave incident to the system.
- v^- – instantaneous voltage wave reflected from the system.
- a – Incident power wave.
- b – Reflected power wave.
- \mathbf{a} – Vector of incident power waves.
- \mathbf{b} – Vector of reflected power waves.
- [**J_F**] – Jacobian Matrix of nonlinear function $\mathbf{F}(\cdot)$.
- ε – Predefined tolerance.
- $x(t)$ – State variable.
- \mathbf{x} – Vector of state variables.
- \mathbf{x}_D – Vector of time delayed state variable.
- s – Complex frequency in reference domain.
- σ – Real part of complex frequency in reference domain.
- ω – Actual frequency in reference domain.
- ψ – Transferred frequency in reference domain.
- ϕ – Actual frequency in transferred reference domain.
- F – Sampling Frequency.
- T – Sampling Interval.
- j – Imaginary axis.
- r – Used to represent any arbitrary port.
- γ_r – Parallel adaptor coefficient for port r .
- ς_r – Series adaptor coefficient for port r .
- [**Q**] – Cut Set Matrix.

- [**B**] – Loop Set Matrix.
- [**G**] – Diagonal matrix with reciprocal of port impedances.
- \mathbf{v}^+ – Vector of incident waves.
- \mathbf{v}^- – Vector of reflected waves.
- [**S_K**] – Scattering Matrix for the adaptors.
- [**I**] – Identity Matrix.
- χ – Relaxation factor or scaling factor.
- λ – Eigenvalue of the matrix [**J_FS**].
- ξ – Perturbation or error for the solution.
- E_A – Total energy stored in the nonlinear capacitors and inductors.
- h – Time step size.
- [**M**] – Modified Nodal Admittance Matrix.
- \mathbf{u} – Vector of node voltages or selected currents.
- \mathbf{S}_s – Source vector.
- \mathbf{S}_f – Source vector for independent sources.
- \mathbf{S}_v – Source vector for the current contribution from the nonlinear devices.
- \mathbf{t} – Discrete time points.
- \mathbf{w}_b – Difference vector used in MPE.
- [**W_b**] – Difference matrix used in MPE.
- [**D**] – Diagonal matrix with square root of reference port impedances.
- [**J_I**] – Jacobian matrix of \mathbf{i}_{NL} .
- [**J_V**] – Jacobian matrix of \mathbf{v}_{NL} .
- [**C**] – Matrix containing capacitances and inductances of the circuit.
- n_{un} – Total number of unknown variables.
- n_{nd} – Total number of unknown variables.
- n_{ad} – Number of additional variables to be solved.
- [**M_C**] – Matrix containing the values of **M** and **C** after time discretization.
- \mathbf{S}_{ni} – Source contribution from previous history after time discretization.
- $\mathbf{v}_{NL}(t)$ – Vector of voltages at nonlinear device ports.
- $\mathbf{i}_{NL}(t)$ – Vector of currents at nonlinear device ports.
- n_{nl} – Number of nonlinear device ports.
- [**T**] – Incidence matrix.
- n_{st} – Number of state variables.
- [**M_{SV}**] – Constant impedance matrix.
- \mathbf{S}_{SV} – Vector accounts for sources and previous history of linear part.

- Z – Characteristic impedance of ports of nonlinear device network.
- \mathbf{a}_0 – Source contribution of incident waves.
- \mathbf{n} – Iteration number of relaxation of power waves.
- $\mathbf{F}(\)$ – Nonlinear vector function.
- $\mathbf{F}_v(\)$ – Error function for node voltages.
- $\mathbf{F}_i(\)$ – Error function for port currents.
- k – Newton iteration number.
- $[\mathbf{S}_B]$ – Block-diagonal matrix derived from scattering matrix, $[\mathbf{S}]$.
- $[\mathbf{S}_O]$ – Rest of the elements of $[\mathbf{S}]$ *i.e.*, $[\mathbf{S}_O] = [\mathbf{S}] - [\mathbf{S}_B]$.
- t_s – Pseudo-time step.
- n_{pseudo} – Total number of pseudo-time step.
- v_{C_p} – Voltage across the capacitor placed between linear and nonlinear network.
- \mathbf{v}_{C_p} – Vector of voltages across the capacitors placed between linear and nonlinear network.
- i_{C_p} – Current through the capacitor placed between linear and nonlinear network.
- h_p – Time step size used for pseudo-transient.
- g_p – Conductance from the discretization of capacitance used in pseudo-transient.
- i_p – Current source from the discretization of capacitance used in pseudo-transient.
- C_s – Capacitance of the capacitor used in pseudo-transient.
- d – Pseudo-transient iteration number.
- $[\mathbf{G}_p]$ – Diagonal matrix containing the conductances from the discretization of capacitance used in pseudo-transient.
- P_{max} – Maximum power that can be transmitted from nonlinear device.
- L_{Sc} – Scaler used in the reflected power consideration of nonlinear devices.
- \mathbf{b}_{sol} – Solution of Eq. (3.22) assumed in theory of MPE.
- K – Number of vectors to extrapolate.
- \mathbf{b}_P – Fixed-point of the sequences generated from Eq. (3.22) to extrapolate.
- q_b – Coefficient of minimal polynomial of matrix $[\mathbf{J}_F\mathbf{S}]$.
- \mathbf{q}_b – Vector of coefficients of minimal polynomial of matrix $[\mathbf{J}_F\mathbf{S}]$.
- t_{end} – End time for simulation.

List of Abbreviations

BJT	– Bipolar Junction Transistor.
CPU	– Central Processing Unit.
CPW	– Coplanar Waveguides.
DDEQ	– Difference-Differential Equation.
DFL	– Delay Free Loop.
FLOPS	– Floating Point Operations per Second.
GFLOPS	– Giga Floating Point Operations per Second.
GPU	– Graphics Processing Unit.
GPGPU	– General Purpose computing on Graphics Processing Units.
HB	– Harmonic Balance.
ITA	– Iterated Timing Analysis.
KCL	– Kirchhoff’s Current Law.
KVL	– Kirchhoff’s Voltage Law.
LNA	– Low Noise Amplifier.
MMIC	– Monolithic Microwave Integrated Circuit.
MNA	– Modified Nodal Analysis.
MNAM	– Modified Nodal Admittance Matrix.
MPE	– Minimum Polynomial Extrapolation.
MPI	– Message Passing Interface.
MESFET	– Metal Semiconductor Field Effect Transistor.
NLTL	– Nonlinear Transmission Line.
OpenMP	– Open Multi-Processing.
PC	– Personal Computer.
pHEMT	– pseudomorphic High Electron Mobility Transistor.
RMS	– Root Mean Square.
SFG	– Signal-Flow Graph.
WDF	– Wave Digital Filter.
WR	– Waveform Relaxation.

Chapter 1

Introduction

1.1 Objectives and Motivations of The Presented Study

Circuit-level simulation of large circuits is a challenging task in terms of memory and CPU time. Nonlinear circuit elements add more complexity to the simulation because of the solution of system of nonlinear algebraic equations and/or differential equations. It is thus of great interest to find efficient simulation methods. The objective of the research presented in this thesis is to investigate the performance of a transient analysis based on fixed-point iterations of wave quantities. His long-term objectives are to develop a robust and efficient parallel circuit simulator for nonlinear circuits that can exploit both computer clusters and multi-processors for higher performance over sequential circuit simulators.

Transient Analysis employs time-domain techniques to simulate circuits. The utmost advantage of transient analysis compared to other techniques is its capability to handle very strong nonlinearities of large circuits. The fact that small time steps can be used in time-domain integration makes this analysis robust [50]. So transient analysis is a promising start to evaluate the theory of nonlinear circuit analysis based on power waves.

The transient analysis formulation presented in this work is based on fixed-point iterations of *power waves* at the nonlinear device ports. The greatest advantage of this approach compared to the traditional fixed-point iteration approaches using voltages and currents is that iterations can not diverge to infinity. This property is the direct consequence of using power waves and is very convenient because it assures physically meaningful excitations to the device models and iterations never go out of control. Steady-state oscillations in the error function are observed

if iterations are not convergent. One objective of this work is to demonstrate this property for circuits with different nonlinear elements. Another objective is to make those sequences showing steady-state oscillations convergent to the appropriate solution *i.e.* guarantee the local convergence.

Numerical methods based on Newton's method require the solution of a set of simultaneous linear equations [3]. A matrix decomposition of a large Jacobian matrix is required at each Newton iteration. In his current implementation, it is not required to decompose a large Jacobian matrix in each iteration, instead, system equations are decoupled to each nonlinear device and the numerical calculations for this dissociated approach can be performed concurrently, so the proposed approach can readily be implemented in parallel computer system to reduce the simulation time to a greater extent. Thus the proposed approach could be efficient for large circuits.

1.2 Original Contributions

Previously, the proposed techniques in the literature use wave quantities instead of voltages and currents for transient simulation of circuits in the framework of Wave Digital Filter (WDF) theory [1]. Fiedler *et al* in [4] employed this approach of WDF theory to simulate power electronic circuits. Some other related works are [30, 55, 5]. But this theory was implemented most of the time for linear circuit elements and a small collection of simple nonlinear elements. C. E. Christoffersen in [12] proposed a transient analysis formulation based on relaxation of power waves at the ports of nonlinear devices, which allows easy inclusion of complex multi-port nonlinear devices. The use of power waves guarantees that at all iterations, nonlinear devices are excited with a physically meaningful input, *i.e.*, the amount of power transmitted to nonlinear devices is bounded. There is little work in the literature exploring this idea. In the presented work the method proposed in [12] is further developed for faster convergence in a wider variety of circuits with parameterized nonlinear device models and implemented in the fREEDA simulator. The introduction of device modelling using parameterization developed with fREEDA can be found in [20]. This is the first time implementation of a relaxation based approach of power waves in a circuit simulator for transient analysis of nonlinear circuits. An approach of dissociation of the system of nonlinear power wave equations into each nonlinear device is utilized in the presented research. The idea of decoupling system equations is common in relaxation of voltages

and currents (*e.g.*, [3]), but quite new for relaxation of power waves. This idea is important for developing a parallel algorithm to solve a system of equations with nonlinearity. The idea of parameterization of nonlinear device models using state variables was used previously for circuit analysis in [38], this concept is employed in the wave approach for the first time. Finally a vector extrapolation method [16] is used for the first time in circuit analysis to improve the convergence of relaxation methods.

1.3 Thesis Outline

Chapter 2 presents the review of all topics that are relevant for the work in this thesis. The formulation of circuit equations using different methods along with the convergence analysis of the methods and convergence improvements are illustrated in Chapter 3. Description of circuits used in this work and all the simulation results with individual discussion are presented in Chapter 4. General discussions and summary of the presented approach with the conclusion and the road map for the future works are included in Chapter 5.

Chapter 2

Literature Review

This chapter presents an overview of existing works relevant to this thesis. The basic theory of waves, passivity and a review of the WDF concept along with all other attempts of circuit analysis using WDF theory are presented first. Next, the principles of relaxation-based transient analysis of circuits are presented, followed by the theory for minimum polynomial extrapolation (MPE).

2.1 Basic Theory of Waves

The theory of transmission line demonstrates the basic conception of waves. His presented research work is based on power waves, so the transformation of voltages and currents to power waves is explored here briefly. Besides, previous works based on waves actually used voltage waves as parameter, so the transformation for voltage waves is also depicted here.

Let us consider a lossless transmission line with reference impedance Z_{B0} terminated in a load of impedance Z_{BL} and a sinusoidal voltage source. The sinusoidal wave travelling through positive z direction will be reflected from the termination. So the voltages and currents at each point of transmission line will contain both positive and negative waves. Hence,

$$V(z) = V^+ e^{-j\beta z} + V^- e^{+j\beta z} \quad (2.1)$$

$$I(z) = \frac{V^+}{Z_{B0}} e^{-j\beta z} - \frac{V^-}{Z_{B0}} e^{+j\beta z} \quad (2.2)$$

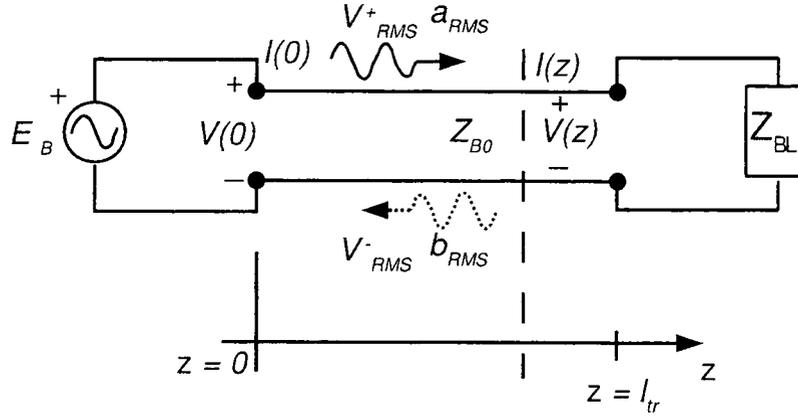


Figure 2.1: A Terminated Lossless Transmission Line

and reflection coefficient,

$$\begin{aligned}\Gamma_B &= \frac{V^+ e^{-j\beta l_{tr}}}{V^- e^{+j\beta l_{tr}}} \\ &= \frac{Z_{BL} - Z_{B0}}{Z_{BL} + Z_{B0}}\end{aligned}\quad (2.3)$$

where, V^+ and V^- are the amplitude of voltage waves through positive and negative z direction respectively, V and I are the steady-state voltage and current of the transmission line, β is the phase constant and l_{tr} is the length of transmission line.

If V^+_{RMS} and V^-_{RMS} are the RMS values of the travelling voltage wave, then $|V^+_{RMS}| = \frac{|V^+|}{\sqrt{2}}$ and $|V^-_{RMS}| = \frac{|V^-|}{\sqrt{2}}$. So the power flowing through positive z direction P_B can be expressed by the following equation:

$$P_B = \frac{|V^+_{RMS}|^2}{Z_{B0}} - \frac{|V^-_{RMS}|^2}{Z_{B0}}\quad (2.4)$$

2.1.1 Voltage Wave

Now voltage and current at $z = l_{tr}$,

$$\begin{aligned}V(l_{tr}) &= V^+ e^{-j\beta l_{tr}} + V^- e^{+j\beta l_{tr}} \\ I(l_{tr}) &= \frac{V^+}{Z_{B0}} e^{-j\beta l_{tr}} - \frac{V^-}{Z_{B0}} e^{+j\beta l_{tr}}\end{aligned}$$

Considering zero length transmission line,

$$\begin{aligned} V &= V^+ + V^- \\ &= \sqrt{2}(V_{RMS}^+ + V_{RMS}^-) \end{aligned} \quad (2.5)$$

$$\begin{aligned} I &= \frac{V^+}{Z_{B0}} - \frac{V^-}{Z_{B0}} \\ &= \frac{\sqrt{2}}{Z_{B0}}(V_{RMS}^+ - V_{RMS}^-) \end{aligned} \quad (2.6)$$

steady-state voltages and currents at any node can be converted to voltage waves considering a zero length fictitious transmission line and these voltage wave quantities can be used as parameters for circuit analysis. Most of the previous works exploring the idea of using wave quantities instead of voltages and currents employed voltage waves.

2.1.2 Power Wave

Using power waves is more convenient than using voltage waves in terms of nonlinearity of the circuit. Let us consider a_{AMP} and b_{AMP} are the amplitude of incident and reflected power waves to the load respectively in such a way that $|a_{AMP}|^2 = \frac{|V^+|^2}{Z_{B0}}$ and $|b_{AMP}|^2 = \frac{|V^-|^2}{Z_{B0}}$. Then voltage and current can be expressed as follows

$$\begin{aligned} V &= \sqrt{Z_{B0}}(a_{AMP} + b_{AMP}) \\ &= \sqrt{2Z_{B0}}(a_{RMS} + b_{RMS}) \end{aligned} \quad (2.7)$$

$$\begin{aligned} I &= \frac{1}{\sqrt{Z_{B0}}}(a_{AMP} - b_{AMP}) \\ &= \frac{\sqrt{2}}{\sqrt{Z_{B0}}}(a_{RMS} - b_{RMS}) \end{aligned} \quad (2.8)$$

where, a_{RMS} and b_{RMS} are the RMS value of incident and reflected power waves respectively. Steady-state voltages and currents can be transformed to power waves using these equations. The idea of transformation can be extended for the instantaneous value of the quantities also.

2.2 Basic Concepts of Wave Digital Filter

WDF represents one class of digital filters that are closely related to the classical filter networks. WDF have discrete structures that emulate an analog reference circuit which is not required to be a filter and thus WDF theory can be applied to model any circuit. The equivalence

between a WDF and its reference circuit from which it is derived is based on the idea of not using voltages and currents as signal parameters but instead using so-called wave quantities for circuit analysis [1]. WDF preserve losslessness and passivity of the reference circuit. So WDF theory validate the application of wave quantities for circuit analysis without changing the behaviour of the circuit. WDF are less sensitive to parameter quantization than discretization based on voltage and current [5]. The transformation of analog reference circuit from so-called reference domain to WDF domain is described in [1].

2.2.1 Signal-Flow Graph

The conception of signal-flow graph (SFG) is important for WDF representation of the reference circuit, that is why a brief description of SFG is provided here. For example, SFG of an inductor and a capacitor is presented. Continuous and discrete time equations (derived from trapezoidal rule) for an inductor and a capacitor are as follows:

$$\begin{aligned} v_{L1}(t) &= L_1 \frac{di_{L1}(t)}{dt} \\ \implies i_{L1}(p) &= i_{L1}(p-1) + \frac{h_{sfg}}{2L_1} (v_{L1}(p) + v_{L1}(p-1)) \end{aligned}$$

$$\begin{aligned} i_{C1}(t) &= C_1 \frac{dv_{C1}(t)}{dt} \\ \implies v_{C1}(p) &= v_{C1}(p-1) + \frac{h_{sfg}}{2C_1} (i_{C1}(p) + i_{C1}(p-1)) \end{aligned}$$

where, L_1 and C_1 are the inductance and capacitance of the inductor and capacitor respectively, v_{L1} , i_{L1} and v_{C1} , i_{C1} are the voltages and currents across the inductor and capacitor respectively, h_{sfg} is the time step size used for discretization and p is the discretized time points. Corresponding SFGs are shown in Fig. 2.2.

Delay free loop (DFL) is a loop of SFG where no delay is involved. DFL is an important term in SFG as well as WDF theory. Here, DFL is explained with a simple LC filter (Fig. 2.3). Continuous time differential equations and corresponding discrete time equations are as follows:

$$\begin{aligned} v_{L2}(t) &= L_2 \frac{di_2(t)}{dt} \\ \implies i_2(p) &= i_2(p-1) + \frac{h_d}{2L_2} (v_{L2}(p) + v_{L2}(p-1)) \end{aligned}$$

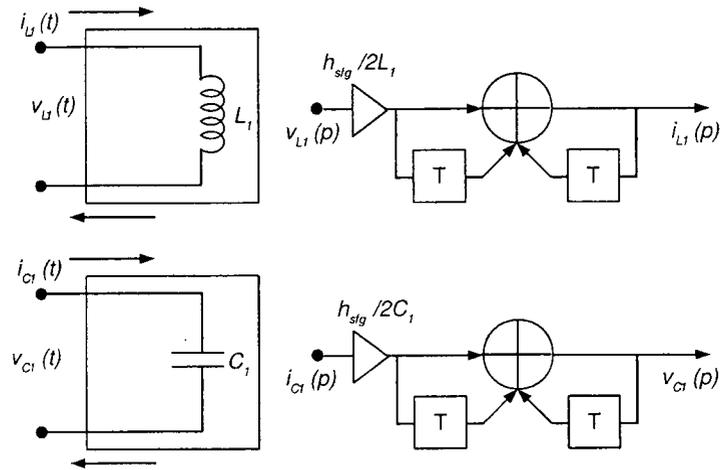


Figure 2.2: SFG of an inductor and a capacitor

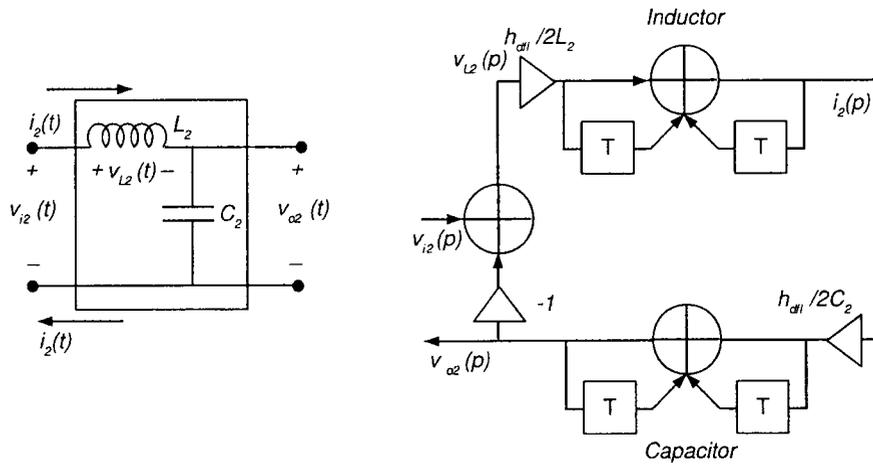


Figure 2.3: DFL by a LC filter

$$\begin{aligned}
i_2(t) &= C_2 \frac{dv_{o2}(t)}{dt} \\
\implies v_{o2}(p) &= v_{o2}(p-1) + \frac{h_d}{2C_2} (i_2(p) + i_2(p-1))
\end{aligned}$$

$$\begin{aligned}
v_{L2}(t) &= v_{i2}(t) - v_{o2}(t) \\
\implies v_{L2}(p) &= v_{i2}(p) - v_{o2}(p)
\end{aligned}$$

The SFG for this *LC* filter (Fig. 2.3) does not contain any delay in the path from input to output, thus SFG for this circuit creates a DFL from input to output that prohibits this implementation from being realizable.

THEOREM 2.2.1. [1] *The SFG of a proper digital filter is realizable at a sampling frequency $F_1 = \frac{1}{T_1}$ (T_1 is the sampling interval) if and only if it satisfies the following conditions:*

1. *It does not contain any DFL.*
2. *The total delay in any loop (directed or not) is equal to a multiple (zero, positive, or negative) of T_1 .*

2.2.2 Frequency Variable

Though this rule is basically used in HB simulation using waves, this idea (described in [1, 4]) is presented to depict some important properties of transformation. Let, $s = \sigma + j\omega$ be the complex frequency for reference circuit which is transferred into another complex frequency variable, ψ . The simplest choice of ψ is the well-known bilinear transform of the z -variable [17].

$$\begin{aligned}
\psi &= \frac{z-1}{z+1} \\
&= \tanh\left(\frac{sT}{2}\right) \\
\text{and } z &= e^{sT}
\end{aligned} \tag{2.9}$$

where $F = \frac{1}{T}$ is the sampling frequency and T is the sampling interval for the digital filter. So WDF domain referred to z -domain. From Eq. (2.9). it can be easily shown that

$$s = \frac{1}{T} \ln z$$

and from first-order bilinear approximation,

$$s \approx \frac{2z-1}{Tz+1} \quad (2.10)$$

From frequency warping concept it can be shown that every point on the unit circle in z -plane, ($z = e^{j\omega T}$) is mapped to a point on the $j\omega$ axis on the s -plane and it is mapped to ψ -domain as follows:

$$\phi = \tan\left(\frac{\omega T}{2}\right), \quad s = j\omega, \quad \psi = j\phi \quad (2.11)$$

and the mapping in different domain,

$$\begin{aligned} \operatorname{Re}(\psi) > 0 &\iff \operatorname{Re}(s) > 0 \iff |z| > 1 \\ \operatorname{Re}(\psi) < 0 &\iff \operatorname{Re}(s) < 0 \iff |z| < 1 \end{aligned} \quad (2.12)$$

where, $\operatorname{Re}(\cdot)$ and $\operatorname{Im}(\cdot)$ denote the real and imaginary axes respectively. Frequencies on the imaginary axes (both ψ -domain and s -domain) are mapped to the unity circle at z -domain, left half plane and right half plane frequencies are mapped inside and outside the unity circle respectively (Fig. 2.4).

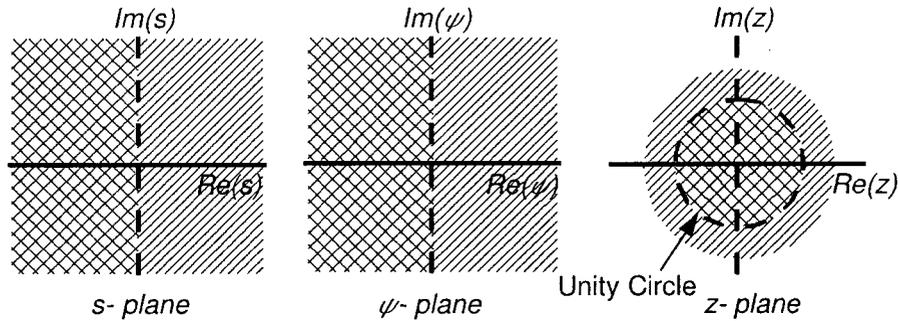


Figure 2.4: Mapping of frequencies in different domain

Property (2.11). implies that real frequencies in the reference domain (*i.e.* imaginary values of ψ) correspond to real frequencies in the WDF domain (*i.e.* to imaginary s) and *vice versa*. property (2.12). indicates that stable reference circuits correspond to stable and causal digital filters and *vice versa*.

2.2.3 Signal Parameters

In classical circuit, a port is characterized by a voltage and a current and a reference port impedance Z_a can be assumed for that port. Let, the instantaneous quantities for voltages and currents are $v = v(t)$ and $i = i(t)$ respectively. Then instantaneous voltage waves (wave quantities) are defined as follows:

$$\left. \begin{array}{l} v^+ = v + Z_a i, \\ v^- = v - Z_a i. \end{array} \right\} \iff \left\{ \begin{array}{l} v = \frac{v^+ + v^-}{2}, \\ i = \frac{v^+ - v^-}{2Z_a}. \end{array} \right. \quad (2.13)$$

where v^+ and v^- are the instantaneous incident and reflected voltage wave quantities. These are the extension of the definition of voltage wave depicted in Eqs. (2.5), and (2.6). This is the basic equation of circuit analysis using voltage wave quantities, explored in all previous works based on voltage waves.

2.3 Building Blocks

WDF realization of reference circuit is derived block by block *i.e.* all elements and interconnections are treated individually. The transformation theory of elements and interconnections for classical circuit from analog reference domain to WDF domain is explored in most of previous works, an elaborate description of these transformations would be found on [1]. The basic idea of building blocks of WDF is mentioned here to understand the procedure of circuit simulation using this theory.

2.3.1 Transformation of Elements and Sources

Reflections from some common circuit elements are calculated in terms of incident voltage waves in this Subsection. Reflection coefficient, defined in Eq. (2.3). will be used for this purpose. Let us consider a resistor, a capacitor and an inductor of values R_3 , C_3 and L_3 respectively connected to a transmission line of reference impedance Z_{E0} , then the load impedances across these elements are R_3 , $\frac{1}{sC_3}$ and sL_3 respectively. Now the reflection coefficients from these

elements are given as follows:

$$\text{Resistor, } \Gamma_R(s) = \frac{R_3 - Z_{E0}}{R_3 + Z_{E0}} \quad (2.14)$$

$$\text{Capacitor, } \Gamma_C(s) = \frac{\frac{1}{sC_3} - Z_{E0}}{\frac{1}{sC_3} + Z_{E0}} \quad (2.15)$$

$$\text{Inductor, } \Gamma_L(s) = \frac{sL_3 - Z_{E0}}{sL_3 + Z_{E0}} \quad (2.16)$$

Now using bilinear transform in Eq. (2.14).

$$\frac{v_R^-(p)}{v_R^+(p)} = \frac{1 - \frac{Z_{E0}}{R_3}}{1 + \frac{Z_{E0}}{R_3}}$$

where, v_R^+ and v_R^- are the incident and reflected voltage waves for the resistor and p is the discretized time points. If it is assumed reference port impedance, $Z_{E0} = \text{load impedance}$, R_3 , then

$$v_R^-(p) = 0 \quad (2.17)$$

So, a resistor absorbs all the waves incident to it. Similarly using bilinear transform in Eq. (2.15).

$$\frac{v_C^-(p)}{v_C^+(p)} = \frac{1 - Z_{E0}C_3s}{1 + Z_{E0}C_3s} = \frac{1 - Z_{E0}C_3\frac{2}{T_3}\frac{z-1}{z+1}}{1 + Z_{E0}C_3\frac{2}{T_3}\frac{z-1}{z+1}} = \frac{(1+z^{-1}) - Z_{E0}C_3\frac{2}{T_3}(1-z^{-1})}{(1+z^{-1}) + Z_{E0}C_3\frac{2}{T_3}(1-z^{-1})}$$

where, v_C^+ and v_C^- are the incident and reflected voltage waves for the capacitor and T_3 is the sampling interval. If it is assumed reference port impedance, $Z_{E0} = \frac{T_3}{2C_3}$, then

$$v_C^-(p) = v_C^+(p-1) \quad (2.18)$$

So incident waves are delayed by one time step after reflection from a capacitor. In the same way, using bilinear transform in Eq. (2.16).

$$\frac{v_L^-(p)}{v_L^+(p)} = \frac{s - \frac{Z_{E0}}{L_3}}{s + \frac{Z_{E0}}{L_3}} = \frac{\frac{2}{T_3}\frac{z-1}{z+1} - \frac{Z_{E0}}{L_3}}{\frac{2}{T_3}\frac{z-1}{z+1} + \frac{Z_{E0}}{L_3}} = \frac{(1-z^{-1}) - \frac{Z_{E0}T_3}{2L_3}(1+z^{-1})}{(1-z^{-1}) + \frac{Z_{E0}T_3}{2L_3}(1+z^{-1})}$$

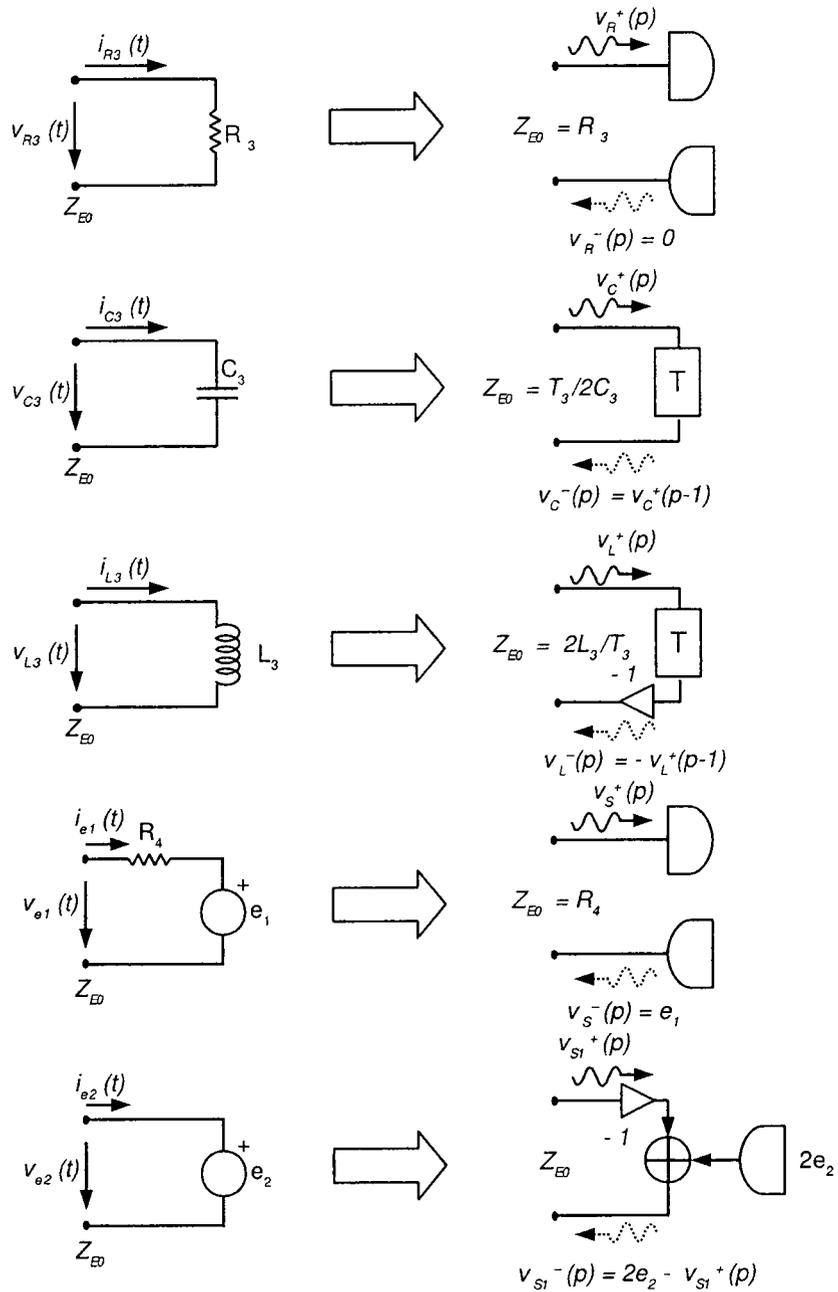


Figure 2.5: Some major one port elements and their WDF domain realization

where, v_L^+ and v_L^- are the incident and reflected voltage waves for the inductor. Assuming reference port impedance, $Z_{E0} = \frac{2L_3}{T_3}$,

$$v_L^-(p) = -v_L^+(p-1) \tag{2.19}$$

So incident waves are inverted and delayed by one time step after reflection from an inductor. Realization of some major one port elements in WDF domain is depicted in Fig. 2.5.

2.3.2 Circulators

Circulator is a nonreciprocal n_c port ($n_c \geq 3$) element and best described in terms of wave quantities as follows

$$\begin{aligned} v_{u1}^- &= v_{u n_c}^+ \\ v_{u2}^- &= v_{u1}^+ \\ v_{u n_c}^- &= v_{u n_c-1}^+ \end{aligned}$$

where, $v_{u n_c}^+$ and $v_{u n_c}^-$ are the incident wave to and reflected wave from port n_c respectively. A

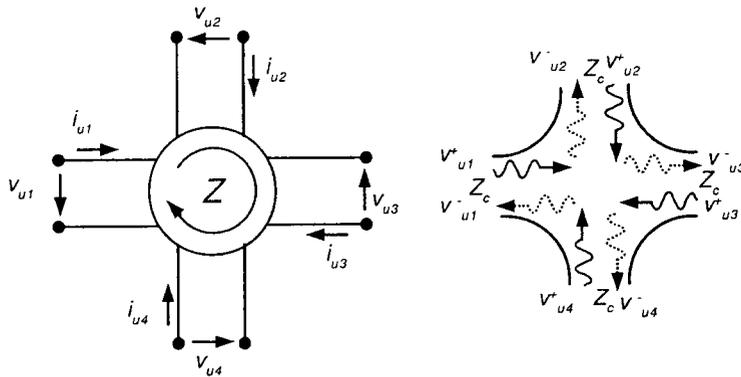


Figure 2.6: WDF realization of a four port circulator

four port circulator is shown in Fig. 2.6.

2.3.3 Interconnections and Adaptors

In this Subsection the procedures are shown to transform the interconnections or the topological rules (Kirchhoff's laws) from classical network to WDF domain. Adaptors are defined for parallel and series connections.

Parallel Adaptors

A three port parallel adaptor (Fig. 2.7) is used to depict the concept first and then the formula is generalized for multiple ports. v^+ is used as the incident wave to the device and v^- as the reflected wave from the device, but from the adaptor point of view v^- is the incident wave and v^+ is the reflected wave, so the solid and dashed waves are interchanged here. The equations for incident and reflected waves at port 1 are as follows:

$$\begin{aligned} v_{p1}^- &= v_{p1} + Z_{p1} i_{p1} \\ v_{p1}^+ &= v_{p1} - Z_{p1} i_{p1} \end{aligned} \quad (2.20)$$

where, v_{p_r} is the voltage across port r and i_{p_r} is the current through that port of parallel connection, Z_{p_r} is the reference impedance of that port, $v_{p_r}^+$ and $v_{p_r}^-$ are the incident and reflected waves at port r of the parallel adaptor. Similarly other equations are as follows:

$$\begin{aligned} v_{p2}^- &= v_{p2} + Z_{p2} i_{p2} \\ v_{p2}^+ &= v_{p2} - Z_{p2} i_{p2} \end{aligned} \quad (2.21)$$

$$\begin{aligned} v_{p3}^- &= v_{p3} + Z_{p3} i_{p3} \\ v_{p3}^+ &= v_{p3} - Z_{p3} i_{p3} \end{aligned} \quad (2.22)$$

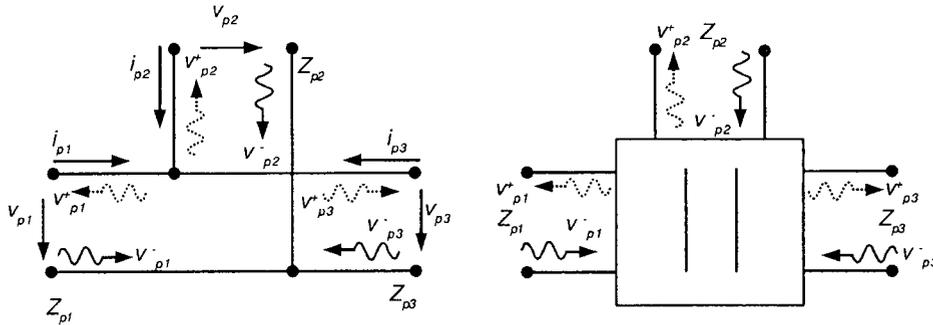


Figure 2.7: WDF domain realization of 3 port parallel adaptor

From Kirchhoff's law, it can be expressed that $v_{p1} = v_{p2} = v_{p3}$ and $i_{p1} + i_{p2} + i_{p3} = 0$. Eliminating v_{p1} , v_{p2} , v_{p3} , i_{p1} , i_{p2} and i_{p3} from Eqs. (2.20), (2.21), and (2.22), the expressions

for the reflected waves can be found from the adaptor,

$$v_{p_1}^+ = \frac{\frac{2}{Z_{p_1}}}{\frac{1}{Z_{p_1}} + \frac{1}{Z_{p_2}} + \frac{1}{Z_{p_3}}} v_{p_1}^- + \frac{\frac{2}{Z_{p_2}}}{\frac{1}{Z_{p_1}} + \frac{1}{Z_{p_2}} + \frac{1}{Z_{p_3}}} v_{p_2}^- + \frac{\frac{2}{Z_{p_3}}}{\frac{1}{Z_{p_1}} + \frac{1}{Z_{p_2}} + \frac{1}{Z_{p_3}}} v_{p_3}^- - v_{p_1}^- \quad (2.23)$$

$$v_{p_2}^+ = \frac{\frac{2}{Z_{p_1}}}{\frac{1}{Z_{p_1}} + \frac{1}{Z_{p_2}} + \frac{1}{Z_{p_3}}} v_{p_1}^- + \frac{\frac{2}{Z_{p_2}}}{\frac{1}{Z_{p_1}} + \frac{1}{Z_{p_2}} + \frac{1}{Z_{p_3}}} v_{p_2}^- + \frac{\frac{2}{Z_{p_3}}}{\frac{1}{Z_{p_1}} + \frac{1}{Z_{p_2}} + \frac{1}{Z_{p_3}}} v_{p_3}^- - v_{p_2}^- \quad (2.24)$$

and

$$v_{p_3}^+ = \frac{\frac{2}{Z_{p_1}}}{\frac{1}{Z_{p_1}} + \frac{1}{Z_{p_2}} + \frac{1}{Z_{p_3}}} v_{p_1}^- + \frac{\frac{2}{Z_{p_2}}}{\frac{1}{Z_{p_1}} + \frac{1}{Z_{p_2}} + \frac{1}{Z_{p_3}}} v_{p_2}^- + \frac{\frac{2}{Z_{p_3}}}{\frac{1}{Z_{p_1}} + \frac{1}{Z_{p_2}} + \frac{1}{Z_{p_3}}} v_{p_3}^- - v_{p_3}^- \quad (2.25)$$

The concept can be generalized for multiple port parallel adaptor (Fig. 2.8). According to Kirchhoff's law,

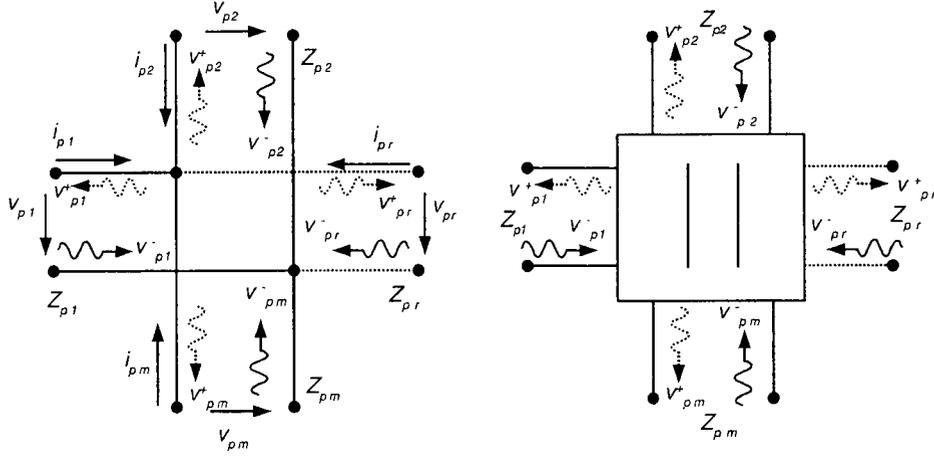
$$v_{p_1} = v_{p_2} = \cdots = v_{p_r} = \cdots = v_{p_m}, \quad i_{p_1} + i_{p_2} + \cdots + i_{p_r} + \cdots + i_{p_m} = 0$$

where $r = 1, 2, 3, \dots, m$ and m is the number of ports in the adaptor.

So the reflection from port r in parallel adaptor can be expressed as follows:

$$v_{p_r}^+ = v_{p_0}^- - v_{p_r}^- \quad (2.26)$$

$$v_{p_0}^- = \gamma_1 v_{p_1}^- + \gamma_2 v_{p_2}^- + \cdots + \gamma_r v_{p_r}^- + \cdots + \gamma_m v_{p_m}^-$$

Figure 2.8: WDF domain realization of m port parallel adaptor

where, γ_r is the adaptor coefficient for port r in m parallel ports and defined as,

$$\gamma_r = \frac{2}{\frac{1}{Z_{p1}} + \frac{1}{Z_{p2}} + \cdots + \frac{1}{Z_{pm}}}$$

An important property of parallel adaptor is that any one port of a parallel adaptor can be made *reflection-free*. Let's say port m is reflection free, *i.e.*,

$$\gamma_m = 1 \quad (2.27)$$

So port impedance Z_{pm} can be expressed as follows

$$\frac{1}{Z_{pm}} = \frac{1}{Z_{p1}} + \frac{1}{Z_{p2}} + \cdots + \frac{1}{Z_{pm-1}} \quad (2.28)$$

The corresponding reflection from port m of parallel adaptor (*i.e.*, incident wave to the device connected to port m) can be derived From Eq. (2.26).

$$v_{pm}^+ = \gamma_1 v_{p1}^- + \gamma_2 v_{p2}^- + \cdots + \gamma_{m-1} v_{p_{m-1}}^-$$

The reflection from that port of parallel adaptor, v_{pm}^+ is independent of incident wave v_{pm}^- (*or*, actual reflection from the device connected to port m) and corresponding port m is called *reflection-free* and the adaptor is said to be *constrained*. Symbolically this absence of reflection

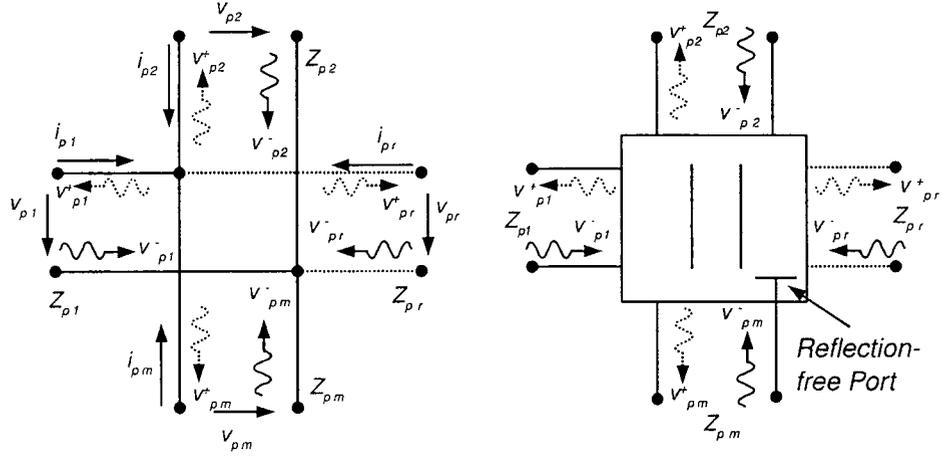


Figure 2.9: WDF domain realization of a parallel adaptor with n_p port *reflection-free*

is represented by a stroke at the output (Fig. 2.9). Physically Eq. (2.28). can be interpreted to express that Z_{p_m} , is equal to the input impedance at port m if the other ports are terminated by their respective port impedances.

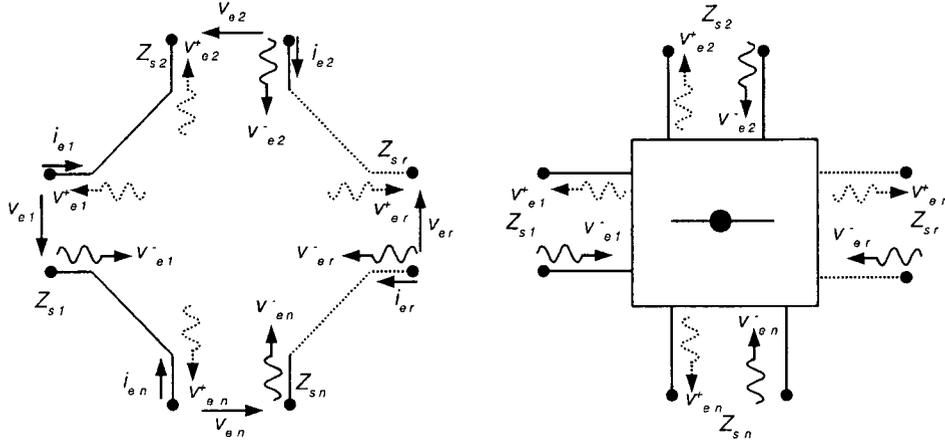
Series Adaptors

For the series adaptor (Fig. 2.10) Kirchoff's laws can be employed as follows:

$$v_{e1} + v_{e2} + \dots + v_{er} + \dots + v_{en} = 0, \quad i_{e1} = i_{e2} = \dots = i_{er} = \dots = i_{en}$$

where, v_{er} is the voltage across the port r and i_{er} is the current through that port of series connection, $r = 1, 2, 3, \dots, n$ and n is the number of ports in the adaptor. Equations for incident waves and reflected waves in terms of node voltages and currents are,

$$\begin{aligned} v_{e1}^- &= v_{e1} + Z_{s1}i_{e1}, & v_{e1}^+ &= v_{e1} - Z_{s1}i_{e1} \\ v_{e2}^- &= v_{e2} + Z_{s2}i_{e2}, & v_{e2}^+ &= v_{e2} - Z_{s2}i_{e2} \\ &\vdots & &\vdots \\ v_{er}^- &= v_{er} + Z_{sr}i_{er}, & v_{er}^+ &= v_{er} - Z_{sr}i_{er} \\ &\vdots & &\vdots \\ v_{en}^- &= v_{en} + Z_{sn}i_{en}, & v_{en}^+ &= v_{en} - Z_{sn}i_{en} \end{aligned}$$

Figure 2.10: WDF domain realization of m port series adaptor

where, Z_{sr} is the reference port impedance at port r . Eliminating voltages and currents from the above equations, the generalized equation for the wave reflection from a series adaptor can be found.

$$v_{e_r}^+ = v_{e_r}^- - \varsigma_r v_{e_0}^- \quad (2.29)$$

where,

$$v_{e_0}^- = v_{e_1}^- + v_{e_1}^- + \cdots + v_{e_r}^- + \cdots + v_{e_n}^-$$

$$\varsigma_r = \frac{2Z_{sr}}{Z_{s1} + Z_{s2} + \cdots + Z_{sn}}$$

One of the ports in series adaptor can be made *reflection-free* too. Let's say port n is the port, so the reflection coefficient of that port is

$$\varsigma_n = 1 \quad (2.30)$$

so port impedance Z_{sn} can be expressed as follows

$$Z_{sn} = Z_{s1} + Z_{s2} + \cdots + Z_{sn-1} \quad (2.31)$$

Similar to the parallel adaptor, reflection from that port of series adaptor, $v_{e_n}^+$ is independent of incident wave $v_{e_n}^-$, so corresponding port n is called *reflection-free* and the adaptor is said to be

constrained. Symbolically this absence of reflection is represented by the same way as parallel adaptor. Physically Eq. (2.31). can be interpreted to express that Z_{sn} , is equal to the input impedance at port n if the other ports are terminated by their respective port impedances.

2.3.4 Scattering Matrix

There is an alternative way to calculate the reflected waves from interconnections directly using so-called *scattering matrix*. It is good to realize all the connections from classical circuits by adaptor concept as a clear view of the connections can be obtained, but for larger circuits this concept is inefficient because a lot of calculations involved in this concept and DFLs may arise in the WDF realization that have to be taken care of individually. The scattering matrix concept avoids this problems. Reflections from the connections could be found in just one step using this concept. The idea of scattering matrix is used in several works in the literature (*e.g.*, [4, 12]). This is described here because a similar idea is used to formulate the circuit equations in his proposed approach. For calculating this matrix all the series and parallel adaptors are combined and adopt any method to write *Kirchhoff's Current Law* (KCL) and *Kirchhoff's Voltage Law* (KVL) equations in vector form, then node voltages and port currents are transformed to incident and reflected waves, this vector form of wave equations are then used to calculate the reflections. The methods, adopted here to find the KCL and KVL equations are KCL based on *cut sets* and KVL based on *loop sets* [43]. If \mathbf{i}_q is the vector of currents and \mathbf{v}_q is the vector of voltages, KCL and KVL for any set of adaptors can be derived as follows:

$$[\mathbf{Q}] \mathbf{i}_q = 0 \quad (2.32)$$

$$[\mathbf{B}] \mathbf{v}_q = 0 \quad (2.33)$$

where $[\mathbf{Q}]$ and $[\mathbf{B}]$ are cut set and loop set matrices respectively. As incident waves and reflected waves for individual ports can be obtained from individual voltages and currents, vectors of waves can be defined as follows:

$$[\mathbf{QG}] (\mathbf{v}_q^- - \mathbf{v}_q^+) = 0 \quad (2.34)$$

$$[\mathbf{B}] (\mathbf{v}_q^- + \mathbf{v}_q^+) = 0 \quad (2.35)$$

where \mathbf{v}_q^- and \mathbf{v}_q^+ are the vectors of incident and reflected voltage waves respectively and $[\mathbf{G}]$ is a diagonal matrix containing the reciprocal of port impedances. Relating \mathbf{v}_q^+ and \mathbf{v}_q^- ,

$$\begin{aligned} \begin{bmatrix} \mathbf{QG} \\ -\mathbf{B} \end{bmatrix} \mathbf{v}_q^+ &= \begin{bmatrix} \mathbf{QG} \\ \mathbf{B} \end{bmatrix} \mathbf{v}_q^- \\ \Rightarrow \mathbf{v}_q^+ &= \begin{bmatrix} \mathbf{QG} \\ -\mathbf{B} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{QG} \\ \mathbf{B} \end{bmatrix} \mathbf{v}_q^- \\ \therefore \mathbf{v}_q^+ &= [\mathbf{S}_K] \mathbf{v}_q^- \end{aligned} \quad (2.36)$$

where, $[\mathbf{S}_K]$ is the corresponding scattering matrix. $[\mathbf{Q}]$, $[\mathbf{B}]$ and $[\mathbf{G}]$ are sparse matrices (matrix mostly contains zeros).

2.3.5 Simulation of a simple *RLC* circuit

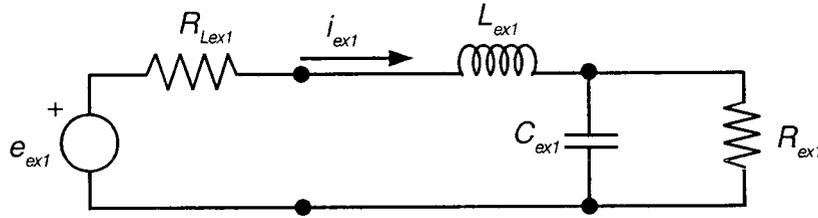


Figure 2.11: A simple *RLC* circuit to demonstrate the procedure of WDF realization

To summarize the approach, WDF realization of a simple *RLC* circuit (Fig. 2.11) is demonstrated. First, graphical means of calculation of adaptor coefficients are employed and then scattering matrix approach is applied to solve the equations derived from topological rules.

Topological rules replaced by adaptors

The circuit is converted to SFG as shown in Fig. 2.12. Signal parameters are calculated from node voltages and currents following the procedures mentioned in Subsection 2.2.3. Sources and elements are realized in WDF domain by the approach stated in Subsection 2.3.1. Two adaptors (one series and another parallel) are identified in this circuit, adaptor coefficients are calculated

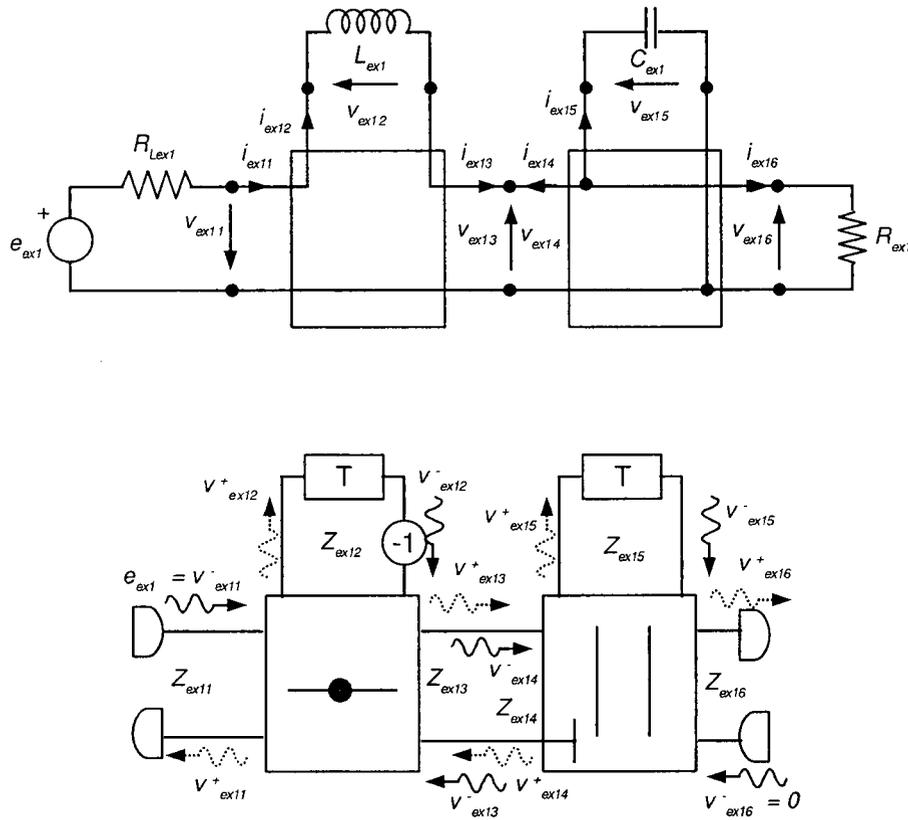


Figure 2.12: WDF realization of the simple RLC circuit with adaptors

following the procedure as depicted in Subsection 2.3.3. Calculations of reflected waves for adaptors are shown below:

From sources and elements,

$$\begin{aligned} v_{ex11}^-(p) &= e_{ex1} \\ v_{ex12}^-(p) &= -v_{ex12}^+(p-1) \\ v_{ex15}^-(p) &= v_{ex15}^+(p-1) \\ v_{ex16}^-(p) &= 0 \end{aligned}$$

where, p is the discretized time points. From series adaptor,

$$\begin{aligned} v_{ex11}^+(p) &= v_{ex11}^-(p) - \varsigma_1 v_{ex10}^- \\ &= v_{ex11}^-(p) - \frac{2Z_{ex11}}{Z_{ex11} + Z_{ex12} + Z_{ex13}} (v_{ex11}^-(p) + v_{ex12}^-(p) + v_{ex13}^-(p)) \\ &= e_{ex1}(p) - \frac{2Z_{ex11}}{Z_{ex11} + Z_{ex12} + Z_{ex13}} (e_{ex1}(p) - v_{ex12}^+(p-1) + v_{ex14}^+(p)) \end{aligned}$$

$$\begin{aligned} v_{ex12}^+(p) &= v_{ex12}^-(p) - \varsigma_2 v_{ex10}^- \\ &= v_{ex12}^-(p) - \frac{2Z_{ex12}}{Z_{ex11} + Z_{ex12} + Z_{ex13}} (v_{ex11}^-(p) + v_{ex12}^-(p) + v_{ex13}^-(p)) \\ &= -v_{ex12}^+(p-1) - \frac{2Z_{ex12}}{Z_{ex11} + Z_{ex12} + Z_{ex13}} (e(p) - v_{ex12}^+(p-1) + v_{ex14}^+(p)) \end{aligned}$$

$$\begin{aligned} v_{ex13}^+(p) &= v_{ex13}^-(p) - \varsigma_3 v_{ex10}^- \\ &= v_{ex13}^-(p) - \frac{2Z_{ex13}}{Z_{ex11} + Z_{ex12} + Z_{ex13}} (v_{ex11}^-(p) + v_{ex12}^-(p) + v_{ex13}^-(p)) \\ &= v_{ex14}^+(p) - \frac{2Z_{ex13}}{Z_{ex11} + Z_{ex12} + Z_{ex13}} (e_{ex1}(p) - v_{ex12}^+(p-1) + v_{ex14}^+(p)) \end{aligned}$$

where, ς_1 , ς_2 and ς_3 are the adaptor coefficients for the series adaptors. From parallel adaptor,

$$\begin{aligned} v_{ex14}^+(p) &= \gamma_4 v_{ex14}^- + \gamma_5 v_{ex15}^- + \gamma_6 v_{ex16}^- - v_{ex14}^- \\ &= \gamma_5 v_{ex15}^-(p) + 0 \quad [\because \gamma_4 = 1 \text{ and } v_{ex16}^-(p) = 0] \\ &= \frac{2}{\frac{1}{Z_{ex14}} + \frac{1}{Z_{ex15}} + \frac{1}{Z_{ex16}}} v_{ex15}^+(p-1) \end{aligned}$$

$$\begin{aligned}
v_{ex15}^+(p) &= \gamma_4 v_{ex14}^- + \gamma_5 v_{ex15}^- + \gamma_6 v_{ex16}^- - v_{ex15}^- \\
&= \frac{\frac{2}{Z_{ex14}}}{\frac{1}{Z_{ex14}} + \frac{1}{Z_{ex15}} + \frac{1}{Z_{ex16}}} v_{ex14}^-(p) + \frac{\frac{2}{Z_{ex15}}}{\frac{1}{Z_{ex14}} + \frac{1}{Z_{ex15}} + \frac{1}{Z_{ex16}}} v_{ex15}^-(p) + 0 - v_{ex15}^-(p) \\
&= \frac{\frac{2}{Z_{ex14}}}{\frac{1}{Z_{ex14}} + \frac{1}{Z_{ex15}} + \frac{1}{Z_{ex16}}} v_{ex13}^+(p) + \frac{\frac{2}{Z_{ex15}}}{\frac{1}{Z_{ex14}} + \frac{1}{Z_{ex15}} + \frac{1}{Z_{ex16}}} v_{ex15}^+(p-1) - v_{ex15}^+(p-1)
\end{aligned}$$

$$\begin{aligned}
v_{ex16}^+(p) &= \gamma_4 v_{ex14}^- + \gamma_5 v_{ex15}^- + \gamma_6 v_{ex16}^- - v_{ex16}^- \\
&= \frac{\frac{2}{Z_{ex14}}}{\frac{1}{Z_{ex14}} + \frac{1}{Z_{ex15}} + \frac{1}{Z_{ex16}}} v_{ex14}^-(p) + \frac{\frac{2}{Z_{ex15}}}{\frac{1}{Z_{ex14}} + \frac{1}{Z_{ex15}} + \frac{1}{Z_{ex16}}} v_{ex15}^-(p) + 0 - 0 [\because v_{ex16}^-(p) = 0] \\
&= \frac{\frac{2}{Z_{ex14}}}{\frac{1}{Z_{ex14}} + \frac{1}{Z_{ex15}} + \frac{1}{Z_{ex16}}} v_{ex13}^+(p) + \frac{\frac{2}{Z_{ex15}}}{\frac{1}{Z_{ex14}} + \frac{1}{Z_{ex15}} + \frac{1}{Z_{ex16}}} v_{ex15}^+(p-1)
\end{aligned}$$

where, γ_4 , γ_5 and γ_6 are the adaptor coefficients for the parallel adaptors.

Topological rules replaced by scattering matrix

Applying the *Tree* concept described in Chapter 4 of [43] the following linearly independent cut set and loop set equations are found,

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} i_{ex11} \\ i_{ex12} \\ i_{ex15} \\ i_{ex16} \end{bmatrix} = 0, \quad \therefore [\mathbf{Q}_{ex1}] = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} v_{ex11} \\ v_{ex12} \\ v_{ex15} \\ v_{ex16} \end{bmatrix} = 0, \quad \therefore [\mathbf{B}_{ex1}] = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

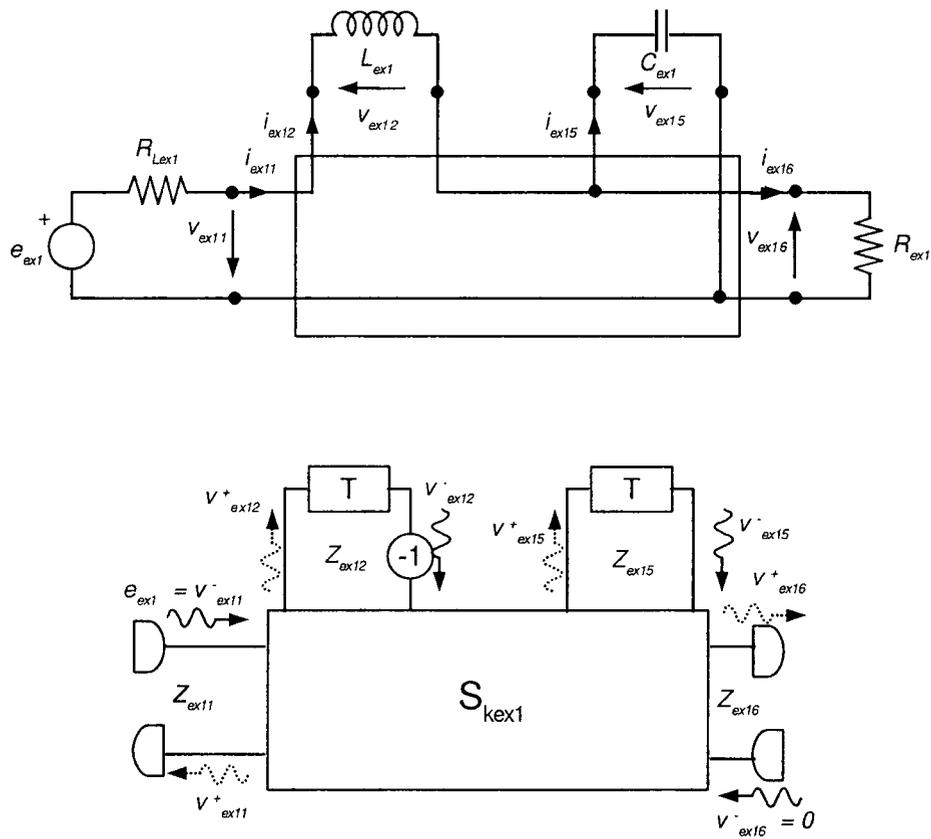


Figure 2.13: WDF realization of a simple *RLC* circuit with Scattering Matrix

where, $[\mathbf{Q}_{ex1}]$ and $[\mathbf{B}_{ex1}]$ are the cut set and loop set matrices respectively. Scattering Matrix $[\mathbf{S}_{K_{ex1}}]$, can be calculated as follows,

$$[\mathbf{S}_{K_{ex1}}] = \begin{bmatrix} \mathbf{Q}_{ex1} \mathbf{G}_{ex1} \\ -\mathbf{B}_{ex1} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{Q}_{ex1} \mathbf{G}_{ex1} \\ \mathbf{B}_{ex1} \end{bmatrix} \quad \text{where } [\mathbf{G}_{ex1}] = \begin{bmatrix} \frac{1}{Z_{ex11}} & 0 & 0 & 0 \\ 0 & \frac{1}{Z_{ex12}} & 0 & 0 \\ 0 & 0 & \frac{1}{Z_{ex15}} & 0 \\ 0 & 0 & 0 & \frac{1}{Z_{ex16}} \end{bmatrix}$$

So the equations for reflected waves are given by

$$\begin{bmatrix} v_{ex11}^+(p) \\ v_{ex12}^+(p) \\ v_{ex15}^+(p) \\ v_{ex16}^+(p) \end{bmatrix} = [\mathbf{S}_{K_{ex1}}] \begin{bmatrix} v_{ex11}^-(p) \\ v_{ex12}^-(p) \\ v_{ex15}^-(p) \\ v_{ex16}^-(p) \end{bmatrix} \\ = [\mathbf{S}_{K_{ex1}}] \begin{bmatrix} e_{ex1}(p) \\ -v_{ex12}^+(p-1) \\ v_{ex15}^+(p-1) \\ 0 \end{bmatrix}$$

2.3.6 General procedure

The general procedures for simulation of simple linear circuits by WDF theory can be summarized as mentioned in [4]:

1. Initialize the devices L, C, R, \dots describing the circuit.
2. Calculate port impedances (Z_1, Z_2, \dots) .
3. Calculate adaptor coefficients γ and ς , or cut sets $[\mathbf{Q}]$ and loop sets $[\mathbf{B}]$ matrices.
4. Initialize the signal variables (*i.e.*, initial values of incident and reflected waves) with appropriate values.
5. Simulate the circuit.

2.4 Single Nonlinearity

Inclusion of nonlinear elements add complexity to WDF realization of circuits. The methods of WDF realization described above are not sufficient for covering most of the nonlinear cases, especially when nonlinearity is demonstrated by differential equations rather than algebraic equations [5]. Solution of multiple port nonlinear elements may lead to DFLs that must be solved in this case also. Some considerable research has been performed to solve the nonlinear equations by waves. First some of the works considering only one nonlinear device will be described briefly then multiple nonlinearities will be illustrated.

2.4.1 Nonlinear Resistances

In [30], representation of a nonlinear resistance in terms of wave variables is shown by Meerkötter *et al.* In this method either current in a nonlinear resistance is expressed as a nonlinear function of voltage or voltage is expressed as nonlinear function of current. This topic is included in literature review as the same idea of expressing nonlinearity by a nonlinear function is used in his proposed work.

1. Current as function of voltage

Let us assume that current through the nonlinear resistor can be expressed as a single valued function of the voltage across it, *i.e.*,

$$i_\omega = f_{\omega_i}(v_\omega)$$

where, i_ω is the current through the resistor and v_ω is the voltage across it which can be any real value. So the incident and reflected waves at this voltage controlled resistance can be expressed as follows,

$$\begin{aligned} v_\omega^+ &= f_{1v^+}(v_\omega) \\ &= v_\omega + Z_\omega f_{\omega_i}(v_\omega) \end{aligned}$$

and

$$\begin{aligned} v_\omega^- &= f_{1v^-}(v_\omega) \\ &= v_\omega - Z_\omega f_{\omega_i}(v_\omega) \end{aligned}$$

where, Z_ω is the reference port impedance. If f_{1v+} has an inverse, f_{1v+}^{-1} defined on the real axis, then reflected waves can be expressed as function of incident waves as follows,

$$\begin{aligned} v_\omega^- &= g_{1v-}(v_\omega^+) \\ &= f_{1v-}(f_{1v+}^{-1}(v_\omega^+)) \end{aligned}$$

The unique invertibility of f_{1v+} , *i.e.* the existence of f_{1v+}^{-1} , is not only a sufficient but also a necessary condition to express v_ω^- as a function of v_ω^+ . In fact, if $f_{1v+}(v_{\omega 1}) = f_{1v+}(v_{\omega 2})$ did not indicate $v_{\omega 1} = v_{\omega 2}$, it would be found that $f_{1v-}(v_{\omega 1}) \neq f_{1v-}(v_{\omega 2})$. f_{1v+}^{-1} will exist if f_{1v+} is strictly monotonic. To guarantee the strict monotonicity of f_{1v+} , the reference impedance, Z_ω has to be chosen such that

either

$$\frac{1}{Z_\omega} > - \inf_{v_{\omega 1} \neq v_{\omega 2}} \frac{f_{\omega i}(v_{\omega 2}) - f_{\omega i}(v_{\omega 1})}{v_{\omega 2} - v_{\omega 1}}$$

or

$$\frac{1}{Z_\omega} < - \sup_{v_{\omega 1} \neq v_{\omega 2}} \frac{f_{\omega i}(v_{\omega 2}) - f_{\omega i}(v_{\omega 1})}{v_{\omega 2} - v_{\omega 1}}$$

2. Voltage as function of current

Similarly, assuming that voltage v_ω can be expressed as a single valued function of the current i_ω , *i.e.*,

$$v_\omega = f_{\omega v}(i_\omega)$$

where i_ω can be any real value. So the incident and reflected waves at this current controlled resistor can be expressed as follows,

$$\begin{aligned} v_\omega^+ &= f_{2v+}(i_\omega) \\ &= f_{\omega v}(i_\omega) + Z_\omega i_\omega \end{aligned}$$

and

$$\begin{aligned} v_\omega^- &= f_{2v-}(i_\omega) \\ &= f_{\omega v}(i_\omega) - Z_\omega i_\omega \end{aligned}$$

If f_{2v^+} has an inverse, $f_{2v^+}^{-1}$ is defined on the real axis, then reflected waves can be expressed as function of incident waves as follows,

$$\begin{aligned} v_{\omega}^{-} &= g_{2v^-}(v_{\omega}^{+}) \\ &= f_{2v^-}(f_{2v^+}^{-1}(v_{\omega}^{+})) \end{aligned}$$

Similarly, the strict monotonicity of f_{2v^+} and thus it's invertibility will be guaranteed by meeting the following criteria,

either

$$Z_{\omega} > - \inf_{i_{\omega 1} \neq i_{\omega 2}} \frac{f_{\omega v}(i_{\omega 2}) - f_{\omega v}(i_{\omega 1})}{i_{\omega 2} - i_{\omega 1}}$$

or

$$Z_{\omega} < - \sup_{i_{\omega 1} \neq i_{\omega 2}} \frac{f_{\omega v}(i_{\omega 2}) - f_{\omega v}(i_{\omega 1})}{i_{\omega 2} - i_{\omega 1}}$$

For example, digital model of Chua's circuit (Fig 2.14) is presented in [30]. The only nonlinear element in this circuit is a nonlinear resistor (R_{Ch}) having the following *current – voltage* characteristics (shown in Fig. 2.15):

$$i_{Ch} = B_1 v_{Ch} + \frac{1}{2}(B_2 - B_1)(|v_{Ch} + v_{Ch0}| - |v_{Ch} - v_{Ch0}|) \quad (2.37)$$

where, i_{Ch} is the current through the nonlinear resistor, v_{Ch} is the voltage across the resistor.

$B_1 = -500 \mu S$, $B_2 = -800 \mu S$ and $v_{Ch0} = 1 V$. Values of the linear elements are given by $C_{Ch2} = 5.5 nF$, $R_{Ch3} = 1.428K \Omega$, $L_{Ch4} = 7.07 mH$, $C_{Ch5} = 49.5 nF$. WDF realization of this circuit is shown in Fig 2.14. Port impedances and Reflection coefficients are calculated according to the procedure mentioned previously by taking $T_{Ch} = 10 \mu S$.

Port impedance of the port connected to nonlinear resistor is calculated such a way to make it *reflection-free*. The calculated value is given by

$$Z_{Ch} = 569.2 \Omega$$

using all the impedance values the expression found for the nonlinear resistor from Eq. (2.37). is given by

$$\begin{aligned} v_{Ch}^{-} &= \varrho(v_{Ch}^{+}) \\ &= \varrho_1 v_{Ch}^{+} + \frac{1}{2}(\varrho_2 - \varrho_1)(|v_{Ch}^{+} + v_{Ch0}^{+}| - |v_{Ch}^{+} - v_{Ch0}^{+}|) \end{aligned} \quad (2.38)$$

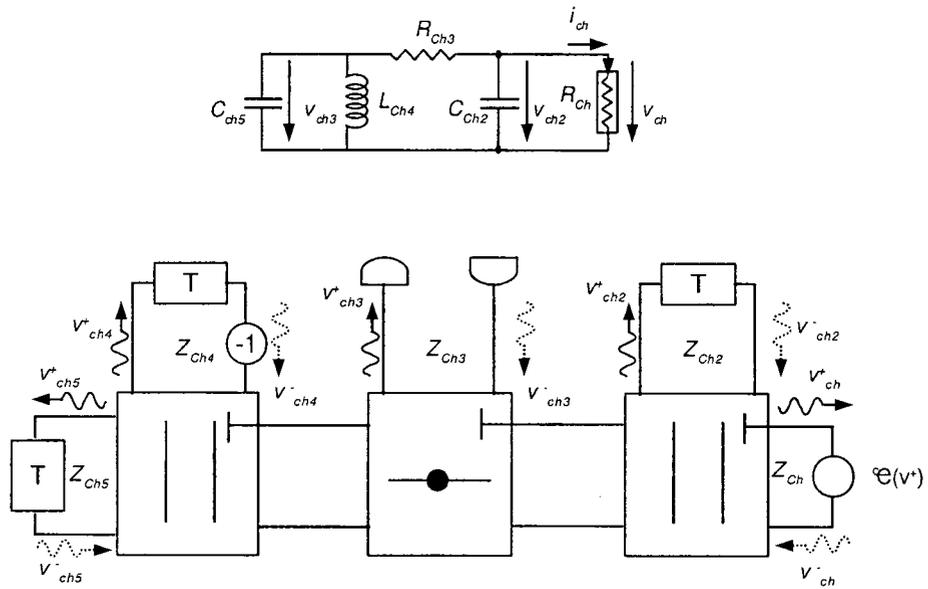


Figure 2.14: WDF realization of a circuit with Nonlinear Resistor [30]

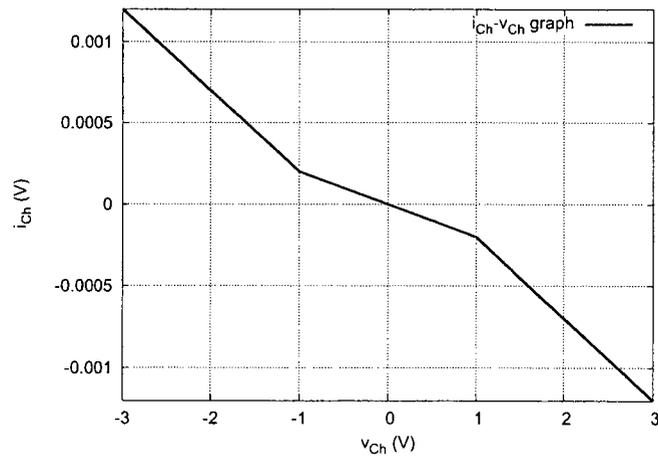


Figure 2.15: current - voltage characteristics of Nonlinear resistor [30]

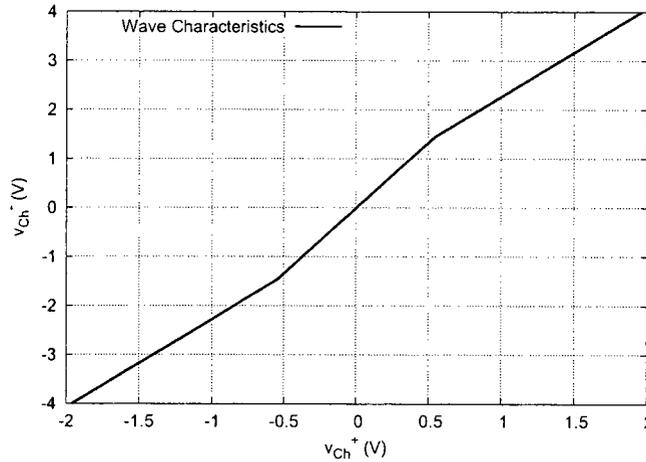


Figure 2.16: Plot of wave characteristics of Nonlinear resistor defined by Eq. (2.38).

where,

$$\varrho_1 = \frac{1 - B_1 Z_{Ch1}}{1 + B_1 Z_{Ch1}} = 1.7956, \quad \varrho_2 = \frac{1 - B_2 Z_{Ch1}}{1 + B_2 Z_{Ch1}} = 2.6722, \quad v_{Ch0}^+ = v_{Ch0}(1 + B_2 Z_{Ch1}) = 0.5447V$$

2.4.2 Nonlinear Elements Treated as Switch

In Ref [4] nonlinear devices are treated as switch because for many Power Electronics applications only switches are important as nonlinear device. This topic is included here to demonstrate the efforts to simulate the nonlinear devices using WDF concept. Switches are distinguished between controlled and uncontrolled devices as mentioned in Ref [4].

1. Transistors like insulated gate bipolar transistors (IGBTs) or MOSFETs are the members of controlled devices. These elements are used as fast switches and probably described as time varying but linear resistor.
2. Most important uncontrolled devices in power electronics are diodes. External switching signals can not control their states (on or off).

The following function is used to describe the rectifying property of a diode,

$$i_z = f_{Di}(v_z)$$

where, i_z is the current through the diode, v_z is the voltage across the diode and rectifying property is denoted by function f_{Di} . The diode is assumed as *incremental passive* [2] here.

For simplicity the function describing a diode is chosen as

$$i_z = f_{Di}(v_z) = \frac{v_z}{R_{Di}}$$

$$R_{Di} = \begin{cases} R_{LD} \geq 0 & \text{if } v_z \geq 0 \\ R_{SD} \leq \infty & \text{if } v_z < 0 \end{cases} \quad R_{LD} \leq R_{SD}$$

Thus a diode is expressed as a resistance R_{Di} with a low value R_{LD} when conducting (*i.e.*, voltage across the diode is positive) and if the diode cuts off R_{Di} is set to a resistance R_{SD} of high value. In practise, this model is too simple for the network containing many diodes.

For example, let us consider a single diode (Fig. 2.17).

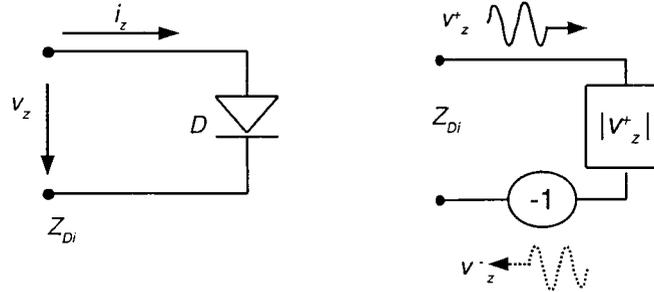


Figure 2.17: WDF realization of a diode treated as switch

$$v_z^+ = v_z + Z_{Di}i_z$$

$$= v_z + Z_{Di}f_{Di}(v_z)$$

$$= \begin{cases} v_z + \frac{Z_{Di}}{R_{LD}}v_z & \text{if } v_z \geq 0 \\ v_z + \frac{Z_{Di}}{R_{SD}}v_z & \text{if } v_z < 0 \end{cases}$$

and

$$\begin{aligned}
 v_z^- &= v_z - Z_{Di}i_z \\
 &= v_z - Z_{Di}f_{Di}(v_z) \\
 &= \begin{cases} v_z - \frac{Z_{Di}}{R_{LD}}v_z & \text{if } v_z \geq 0 \\ v_z - \frac{Z_{Di}}{R_{SD}}v_z & \text{if } v_z < 0 \end{cases}
 \end{aligned}$$

where, Z_{Di} is the reference port impedance. From the above two equations the expressions for the reflection can be found,

$$\begin{aligned}
 v_z^- &= \begin{cases} \frac{R_{LD} - Z_{Di}}{R_{LD} + Z_{Di}}v_z^+ & \text{if } v_z^+ \geq 0 \\ \frac{R_{SD} - Z_{Di}}{R_{SD} + Z_{Di}}v_z^+ & \text{if } v_z^+ < 0 \end{cases} \\
 &= -|v_z^+|, \quad \text{if } R_{LD} = 0 \text{ and } R_{SD} \rightarrow \infty
 \end{aligned}$$

2.5 Multiple Nonlinearity

In the previous section, some explorations on WDF realization concerning nonlinearity of circuit elements are demonstrated. But all these methods are restricted to single nonlinearities only. The delay-free feedback of the nonlinear element may result in DFLs for multiple nonlinearities that must be avoided or solved for the realization of whole circuit in WDF domain. Methods were proposed in [53] and [22] to eliminate DFLs caused from multiple nonlinearities. These methods are based on simultaneous computation (or precomputation) of all possible reflections given all possible incident waves from all nonlinear device ports. These approaches are useful for circuits with small number of nonlinear elements only. To avoid these precomputations an iterative method is required to solve the equations arisen from the nonlinearities. A relaxation approach was proposed by Meerkötter *et al.* in [31] to eliminate DFLs. This approach is always convergent for circuits that contain only locally passive [29] nonlinear elements. Felderhoff in [55] further investigated this approach and proved that it is possible to cut DFLs and split the computation in independent blocks suitable for parallel processing without losing convergence. C. E. Christoffersen in [12] proposed a wave-based transient analysis approach that solves DFL through an iterative procedure and has better convergence properties than previous works.

This approach allows easy inclusion of complex multiple port nonlinear devices formulated in the Kirchhoff domain and only one large (sparse) matrix decomposition is required for a given time step size. Some of the approaches to solve the multiple nonlinearity in the circuit will be described here briefly.

2.5.1 Jacobi's Method to cut DFL

To derive the WDF realization of nonlinear circuit elements either implicit equations have to be solved by guaranteeing the unambiguity of the solutions (proposed in [31]), or WDF circuit has to be modified in such a way that an iterative formulation can be used to cut DFLs. In [55] a method similar to the Jacobi's method [36, 3] was presented for solving a system of equations numerically. It is possible to cut all DFLs with the aid of this method and a massive parallel algorithm can be generated by the systematical use of the method. This topic is included in this chapter because an iterative way is used to solve the system of equations seems like a tautology used in his work.

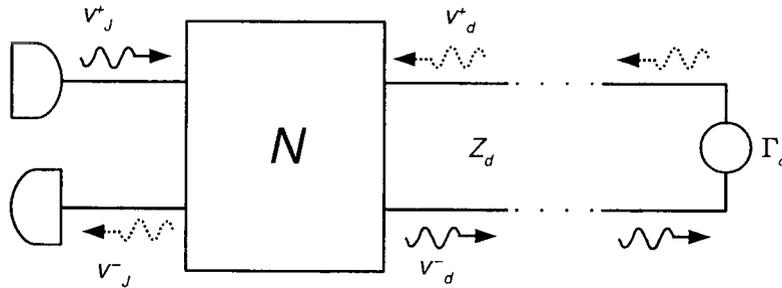


Figure 2.18: WDF model for iterative solving of DFL

One path of every DFL is shown out of the remaining non-energized WDF network N in Fig. 2.18. Vector of incident waves to DFLs is \mathbf{v}_d^- and \mathbf{v}_d^+ is the vector of reflected waves. Then the network can be described as

$$\begin{bmatrix} \mathbf{v}_J^- \\ \mathbf{v}_d^- \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{D11} & \mathbf{S}_{D12} \\ \mathbf{S}_{D21} & \mathbf{S}_{D22} \end{bmatrix} \begin{bmatrix} \mathbf{v}_J^+ \\ \mathbf{v}_d^+ \end{bmatrix}$$

where $[\mathbf{S}_{Drl}]$, $r, l \in \{1, 2\}$ are the scattering matrices describing both the adaptors and DFLs, \mathbf{v}_J^+

and \mathbf{v}_j^- are the vectors of incident and reflected waves to the network respectively. Felderhoff in [55] proposed a reflectance, Γ_d that ensures there are no DFLs between \mathbf{v}_d^- and \mathbf{v}_d^+ any longer. So it is assumed that

$$\Gamma_d(\psi)|_{\psi=1} = 0$$

where, ψ is the frequency. The steady state solution has to fulfil $\mathbf{v}_d^- = \mathbf{v}_d^+$, so the reflectance has to satisfy the following also

$$\Gamma_d(\psi)|_{\psi=0} = 1$$

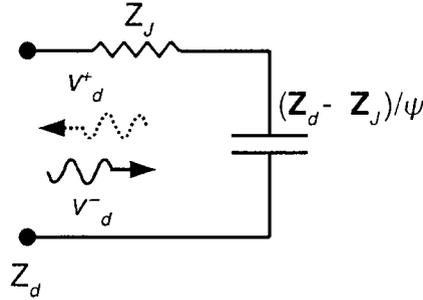


Figure 2.19: Impedance to cut DFL

A simple impedance derived in [55] is shown in Fig. 2.19, where $[Z_d]$, $[Z_J]$ and $\left[\frac{Z_d - Z_J}{\psi}\right]$ are positive diagonal matrices. Iterative equation to solve \mathbf{v}_d^+ is given by

$$\mathbf{v}_{d_j}^+ = [\kappa]\mathbf{v}_{d_{j-1}}^+ + [\mathbf{I} - \kappa]([\mathbf{S}_{D22}]\mathbf{v}_{d_{j-1}}^+ + [\mathbf{S}_{D21}]\mathbf{v}_j^+) \quad (2.39)$$

where j is the iteration number and $[\kappa] = [Z_d^{-1}Z_J]$. This iteration has to be done for each sampling point. For iteration number, $j \rightarrow \infty$ the equation is given by

$$\begin{aligned} \mathbf{v}_d^+ &= \mathbf{v}_d^- \\ &= [\mathbf{S}_{D21}]\mathbf{v}_j^+ + [\mathbf{S}_{D22}]\mathbf{v}_d^+ \\ \therefore [\mathbf{I} - \mathbf{S}_{D22}]\mathbf{v}_d^+ &= [\mathbf{S}_{D21}]\mathbf{v}_j^+ \end{aligned}$$

Using Jacobi's over relaxation method [7] for solving the equation it is found that

$$\mathbf{v}_d^+ = [\kappa]\mathbf{v}_d^+ + [\mathbf{I} - \kappa][\mathbf{M}_D^{-1}][[\mathbf{M}_{LR}]\mathbf{v}_d^+ + [\mathbf{S}_{D21}]\mathbf{v}_j^+] \quad (2.40)$$

where $[\mathbf{I} - \mathbf{S}_{D22}] = [\mathbf{M}_D - \mathbf{M}_{LR}]$, $[\mathbf{M}_D]$ is a diagonal matrix and $[\mathbf{M}_{LR}]$ is a lower and upper triangular matrix.

There is a significant similarity between Eqs. (2.39). and (2.40). So the method proposed is a modified Jacobi's method that use Eq. (2.39). to calculate the waves in DFLs. Stability of the iteration is guaranteed as the circuit elements are passive.

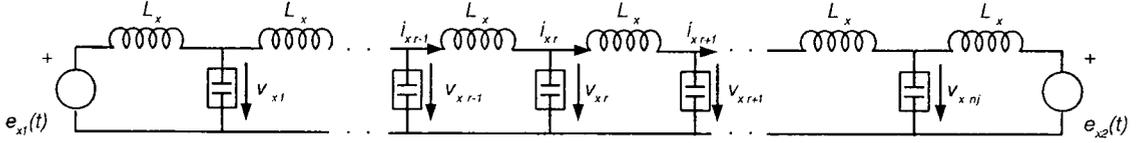


Figure 2.20: Lossless nonlinear transmission line

For example, a lossless nonlinear transmission line (Fig. 2.20), investigated in several other works [31, 48, 49, 24, 32, 39] is presented. Inductances are assumed to be linear and capacitances are nonlinear. Voltages across the capacitances are characterized by the following difference-differential equation (DDEQ)

$$v_{xr+1} - 2v_{xr} + v_{xr-1} = L_x \frac{d}{dt} \left(\frac{C_{x0}}{1 + \frac{v_{xr}}{V_{x0}}} \frac{dv_{xr}}{dt} \right)$$

where, C_{x0} and V_{x0} are constant. One set of solutions to this DDEQ is given by so-called solitons [39]

$$v_{xr}(t) = J_1 \operatorname{sech}^2 \left[J_2 \left(t \mp \frac{r}{\nu} \right) \right]$$

where J_1 is an arbitrary positive constant and

$$J_2 = \sqrt{\frac{J_1/V_{x0}}{L_x C_{x0}}} \quad \text{and} \quad \nu = \frac{J_2}{\operatorname{arcsinh} \sqrt{\frac{V_{\text{con}}}{V_{x0}}}}$$

Solutions can travel in both directions with a velocity ν that increases with J_1 . The following values of the elements are used for the simulation [39]: $L_x = 1.38 \mu H$, $C_{x0} = 224.9 pF$ and $V_{x0} = 3.73 V$.

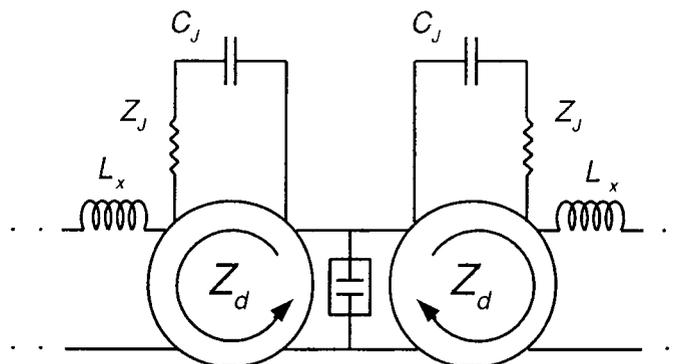


Figure 2.21: WDF realization of nonlinear transmission line by modified Jacobi's method

Each nonlinear capacitance is modelled by a nonlinear voltage-controlled ideal transformer connected with a linear capacitance [31]. Impedances shown in Fig. 2.19 are inserted to the nonlinear transmission line by terminating three-port circulators and modified Jacobi's method shown in Eq. (2.39). is used to solve the wave quantities. This modification not only cuts all DFLs but also leads to a realizable massive parallel algorithm as the calculation in one part (inductance, circulator, impedance and capacitance) is independent of the rest of the network. A small portion of modified transmission line is depicted in Fig. 2.21, where

$$C_J = \frac{T_J}{2(Z_d - Z_J)} \text{ and } Z_d = \sqrt{\frac{L_x}{C_x(0)}}$$

T_J denotes the sampling time.

2.5.2 Newton's Method for strong nonlinearity

Power waves instead of voltage waves was proposed as analysis parameter in [12] first by C. E. Christoffersen. The proposed method has better numerical properties associated with the WDF approach and permits inclusion of complex nonlinear device models. Most attractive feature of this approach is that only one matrix decomposition is required for a given time step size in

this method. This method is further developed in his proposed approach. Formulation of circuit equations by this method starts with the partitioned network as shown in Fig. 2.22.

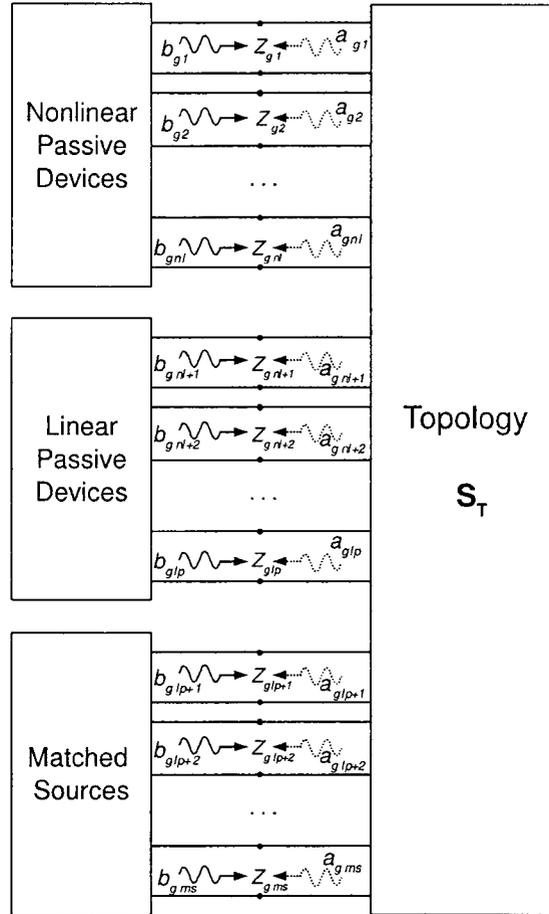


Figure 2.22: Circuit partitioning

Linear and nonlinear devices are assumed to be passive and sources are assumed to be matched to the impedances of their respective ports. Voltages and currents at port r can be expressed in terms of power waves the same way as shown in Eqs. (2.7), and (2.8).

$$v_{g_r} = \sqrt{Z_{g_r}}(a_{g_r} + b_{g_r}) \quad (2.41)$$

$$i_{g_r} = \frac{a_{g_r} - b_{g_r}}{\sqrt{Z_{g_r}}} \quad (2.42)$$

where, v_{g_r} and i_{g_r} are the instantaneous voltage and current at port r , a_{g_r} and b_{g_r} are the instantaneous incident and reflected power waves as seen from device and Z_{g_r} is the reference impedance of port r and it is pure real as only time domain method is considered.

Topological rules can be expressed by scattering matrix the same way as shown in Subsection 2.3.4 (Eqs. (2.32),-(2.36).)

$$\begin{aligned} \mathbf{a}_g &= \begin{bmatrix} \mathbf{Q}_g \mathbf{D}_g^{-1} \\ \mathbf{B}_g \mathbf{D}_g \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{Q}_g \mathbf{D}_g^{-1} \\ -\mathbf{B}_g \mathbf{D}_g \end{bmatrix} \mathbf{b}_g \\ \therefore \mathbf{a}_g &= [\mathbf{S}_T] \mathbf{b}_g \end{aligned} \quad (2.43)$$

where $[\mathbf{S}_T]$ is the scattering matrix of the topology, $[\mathbf{Q}_g]$ and $[\mathbf{B}_g]$ represent the full cut-set and loop-set matrices, $[\mathbf{D}_g]$ is the diagonal matrix with square root of reference port impedances, \mathbf{a}_g and \mathbf{b}_g are the vector of incident and reflected waves. Scattering matrix equation representing the topology can be rearranged as

$$\begin{bmatrix} \mathbf{a}_{g_{NL}} \\ \mathbf{a}_{g_L} \\ \mathbf{a}_{g_S} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_{T11} & \mathbf{S}_{T12} & \mathbf{S}_{T13} \\ \mathbf{S}_{T21} & \mathbf{S}_{T22} & \mathbf{S}_{T23} \\ \mathbf{S}_{T31} & \mathbf{S}_{T32} & \mathbf{S}_{T33} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{g_{NL}} \\ \mathbf{b}_{g_L} \\ \mathbf{b}_{g_S} \end{bmatrix}$$

where $\mathbf{a}_{g_{NL}}$, \mathbf{a}_{g_L} and \mathbf{a}_{g_S} are the vectors of waves incident to devices and sources respectively and $\mathbf{b}_{g_{NL}}$, \mathbf{b}_{g_L} and \mathbf{b}_{g_S} are the same quantities defined for reflected waves. Reference impedances at sources and linear devices are chosen in such a way that there is no DFL from the device back to the network [4]. So \mathbf{b}_{g_L} is constant for one time step as all loops contain at least one delay.

Both static and dynamic nonlinear devices cause DFLs in WDF realization. Newton's method is used to calculate the reflection from nonlinear device ports. From previous works, it is known that currents in nonlinear device ports can be expressed as nonlinear function of voltages (*i.e.*, $i_{g_r} = f_{g_i}(v_{g_r})$). An error function $E_g(a_{g_r})$ can be defined as follows,

$$E_g(a_{g_r}) = \sqrt{Z_{g_r}} f_{g_i}(\sqrt{Z_{g_r}}(a_{g_r} + b_{g_r})) - a_{g_r} + b_{g_r} = 0$$

This error function can be generalized for both static and dynamic multi-port nonlinear devices. Calculations for Newton's method performed for each nonlinear device is independent of the rest of the network and require a small Jacobian matrix factorization. This approach allows straightforward treatment of complex nonlinear models.

The nonlinear equation to be solved by Newton's method is given by

$$\mathbf{b}_{g_{NL}} = \mathbf{F}_g([\mathbf{S}_{\mathbf{T}11}]\mathbf{b}_{g_{NL}} + \mathbf{a}_{g_0}) \quad (2.44)$$

where $\mathbf{a}_{g_0} = \mathbf{S}_{\mathbf{T}12}\mathbf{b}_{g_L} + \mathbf{S}_{\mathbf{T}13}\mathbf{b}_{g_S}$ being constant for a given time step and $\mathbf{F}_g(\cdot)$ is a nonlinear vector function that represents the contribution of nonlinear passive devices. The proposed fixed-point iterative scheme to solve Eq. (2.44). is as follows,

$$\mathbf{b}_{g_{NL}}^{\rho+1} = [\mathbf{I} - \mathbf{M}_S^\rho]\mathbf{b}_{g_{NL}}^\rho + [\mathbf{M}_S^\rho]\mathbf{F}_g([\mathbf{S}_{\mathbf{T}11}]\mathbf{b}_{g_{NL}}^\rho + \mathbf{b}_{g_0}) \quad (2.45)$$

where ρ denotes the iteration number, $[\mathbf{I}]$ is the identity matrix and $[\mathbf{M}_S^\rho]$ is a matrix of size $nl \times nl$ (nl is the number of nonlinear device ports) that may be a constant or updated at each iteration.

Different selections of $[\mathbf{M}_S^\rho]$ will lead to different method for solving Eq. (2.45). With $[\mathbf{M}_S^\rho] = [\mathbf{I}]$ Eq. (2.45). is equivalent to just propagating reflections along the DFLs in the circuit (*i.e.*, plain relaxation). $[\mathbf{M}_S^\rho] = \chi_j[\mathbf{I}]$ (χ_j is a scalar and $0 < \chi_j \leq 1$) will lead Eq. (2.45). to the same approach proposed in [55]. Iterations converge to the desired solution if the spectral radius [46] of $[\mathbf{J}_{F_g}\mathbf{S}_{\mathbf{T}11}]$ is less than one, where $[\mathbf{J}_{F_g}]$ is the Jacobian matrix of function $\mathbf{F}_g(\cdot)$. This condition is satisfied if all nonlinear devices are locally passive [29] (e.g., diodes). So local convergence can be obtained by the following condition:

$$[\mathbf{M}_S^\rho] = [\mathbf{I} - \mathbf{J}_{F_g}\mathbf{S}_{\mathbf{T}11}]^{-1} \quad (2.46)$$

Eq. (2.45). is equivalent to Newton's method. It is possible to re-order Eq. (2.43). and (2.45). to perform one sparse matrix decomposition per iteration, but this is more expensive than the backward substitution of the relaxation method. It is possible to avoid the factorization of a large matrix at each iteration by making $[\mathbf{M}_S^\rho]$ artificially sparse. If the elements of $[\mathbf{S}_{\mathbf{T}11}]$ in Eq. (2.46). are set to zero to get the same block-diagonal pattern as in $[\mathbf{J}_{F_g}]$, the derived $[\mathbf{M}_S^\rho]$ matrix is also block-diagonal. Precise knowledge of $[\mathbf{S}_{\mathbf{T}11}]$ is required for this which in turn requires nl backward substitutions of the decomposed matrix originated from Eq. (2.43). but this is required to be performed once only. The resulting iterative scheme is known as *Newton-Jacobi* algorithm [27]. Local convergence is no longer guaranteed for this approach but it is shown in the work that this modification is an improvement over the relaxation approach. For smaller circuits containing few nonlinear devices, Newton's method is faster than Newton-Jacobi

approach but this could be faster however for large circuits as the cost of iterations does not grow as much with the circuit size.

2.6 Relaxation-Based Electrical Analysis

The *relaxation method* is a method to obtain numerical approximations to the solutions of system of equations both linear and nonlinear. Solving a linear system of equations requires LU decomposition of a sparse matrix in direct method of solution and solving a nonlinear system of equations by Newton's method requires LU decomposition of a Jacobian matrix. Solution of a system of equations by relaxation method involves decoupling of equations (*i.e.*, splitting the matrix to decompose). Though the convergence rate is slower for relaxation method it has an advantage over the traditional approaches as it is not required to decompose a large matrix in each iteration. A good introduction to the relaxation-based electrical analysis would be found in [3], most of the topics, demonstrated in this Section are cited from [3].

Nodal equations of an electrical circuit under certain assumptions mentioned in [3] can be expressed as follows:

$$\begin{aligned} [\mathbf{C}_R(\mathbf{v}_R(t), \mathbf{s}_R(t))] \dot{\mathbf{v}}_R(t) &= -\mathbf{I}_R(\mathbf{v}_R(t), \mathbf{s}_R(t)), & 0 \leq t \leq T_R \\ \mathbf{v}_R(0) &= V_R \end{aligned} \quad (2.47)$$

where $\mathbf{v}_R(t) \in \mathbb{R}^{n_n}$ is the vector of node voltages at time t and n_n is the total number of nodes in the circuit; $\dot{\mathbf{v}}_R(t) \in \mathbb{R}^{n_n}$ is the vector of time derivatives of $\mathbf{v}_R(t)$; $\mathbf{s}_R(t) \in \mathbb{R}^{n_n}$ is the input vector at time t , $[\mathbf{C}_R] : \mathbb{R}^{n_n} \rightarrow \mathbb{R}^{n_n \times n_n}$ represents the nodal capacitance matrix, $\mathbf{I}_R : \mathbb{R}^{n_n} \times \mathbb{R}^{n_n} \rightarrow \mathbb{R}^{n_n}$, and

$$\mathbf{I}_R(\mathbf{v}_R(t), \mathbf{s}_R(t)) = [i_{R1}(\mathbf{v}_R(t), \mathbf{s}_R(t)), i_{R2}(\mathbf{v}_R(t), \mathbf{s}_R(t)), \dots, i_{Rn_n}(\mathbf{v}_R(t), \mathbf{s}_R(t))]^T$$

where $i_{Rr}(\mathbf{v}_R(t), \mathbf{s}_R(t))$ is the sum of the currents charging the capacitors connected to node r . Expressing the time dependency implicitly Eq. (2.47), can be expressed as follows:

$$[\mathbf{C}_R(\mathbf{v}_R, \mathbf{s}_R)] \dot{\mathbf{v}}_R = -\mathbf{I}_R(\mathbf{v}_R, \mathbf{s}_R), \quad (2.48)$$

After numerical integration Eq. (2.48). can be expressed as a set of nonlinear, algebraic difference equations of the form

$$\mathbf{F}_R(\mathbf{y}) = 0 \quad (2.49)$$

where, $\mathbf{F}_R(\cdot)$ denotes the nonlinear vector function and $\mathbf{y} \in \mathbb{R}^{n_n}$ is the vector of node voltages at time $t = t_R + 1$. Newton's method yields a set of sparse linear equations from Eq. (2.49). of the form,

$$\begin{aligned} [\mathbf{J}_{F_R}(\mathbf{y}^\iota)] \mathbf{y}^{\iota+1} &= \mathbf{d}_R(\mathbf{y}^\iota) \\ \implies [\mathbf{J}_{F_R}] \mathbf{y}^{\iota+1} &= \mathbf{d}_R \end{aligned} \quad (2.50)$$

where, ι is the Newton iteration number, $[\mathbf{J}_{F_R}] \in \mathbb{R}^{n_n \times n_n}$ is the Jacobian matrix of $\mathbf{F}_R(\cdot)$ and $\mathbf{d}_R \in \mathbb{R}^{n_n}$ is a vector that depends on the value from previous Newton iteration. Eq. (2.50). has to be solved in each iteration of Newton's method and is usually solved using direct methods, such as *LU* decomposition or Gaussian Elimination.

Relaxation methods offer alternatives to solve Eq. (2.47), Relaxation methods can be applied in a number of ways. In all cases, they do not require the direct solution of a large system of equations and permit the simulator to exploit the latency efficiently.

1. Linear Relaxation Methods

Relaxation methods can be applied to Eq. (2.50). to avoid doing the *LU* decomposition in Newton iterations. Two most common relaxation methods used in electrical simulation are the Gauss-Jacobi method and the Gauss-Seidel method. Jacobian matrix $[\mathbf{J}_{F_R}]$ is divided into $[\mathbf{J}_{F_R}^{LT} + \mathbf{J}_{F_R}^D + \mathbf{J}_{F_R}^{UT}]$ where $[\mathbf{J}_{F_R}^{LT}] \in \mathbb{R}^{n_n}$ is strictly lower triangular, $[\mathbf{J}_{F_R}^D] \in \mathbb{R}^{n_n}$ is diagonal and $[\mathbf{J}_{F_R}^{UT}] \in \mathbb{R}^{n_n}$ is strictly upper triangular. In the Gauss-Jacobi method Eq. (2.50). is solved by the following equation

$$[\mathbf{J}_{F_R}^D] \mathbf{y}^{\eta+1} = - [\mathbf{J}_{F_R}^{LT} + \mathbf{J}_{F_R}^{UT}] \mathbf{y}^\eta + \mathbf{d}_R \quad (2.51)$$

And in Gauss-Seidel method the following equation is used

$$[\mathbf{J}_{F_R}^{LT} + \mathbf{J}_{F_R}^D] \mathbf{y}^{\eta+1} = - [\mathbf{J}_{F_R}^{UT}] \mathbf{y}^\eta + \mathbf{d}_R \quad (2.52)$$

where, η is the iteration number. It may require more than one iterations to solve Eq. (2.50). by Gauss-Jacobi or Gauss-Seidel method, whereas using direct methods only one step is required.

As relaxation methods are iterative methods, it is important to explore the conditions to guarantee the convergence of the methods. The iterations are not well defined if $[\mathbf{J}_{F_R}^D]$ is

singular (*i.e.*, if there is zero on main diagonal of $[\mathbf{J}_{F_R}]$). It is shown in [3] that if $[\mathbf{J}_{F_R}]$ is strictly diagonally dominant, then both Gauss-Jacobi and the Gauss-Seidel iteration converge to the solution of Eq. (2.50). independent of the initial guess.

Another important property of iterative methods is the rate of convergence. It is shown in [3] that if the Gauss-Jacobi and Gauss-Seidel iteration converge, they converge at least linearly. That is, the error at each iteration decreases according to the following relation after a sufficiently large number of iterations,

$$\|\mathbf{y}^{\eta+1} - \mathbf{y}_{sol}\| \leq \varepsilon^{R_i} \|\mathbf{y}^{\eta} - \mathbf{y}_{sol}\|$$

where, \mathbf{y}_{sol} is the solution of Eq. (2.50). and ε^{R_i} is the error at each iteration of iterative relaxation methods.

2. Nonlinear Relaxation Methods

To enhance the Newton's method, relaxation methods can also be used at the nonlinear equation solution level. Let \mathbf{y}^{η} denote the value of \mathbf{y} at the η^{th} iteration. When applied to Eq. (2.49). the Gauss-Jacobi and Gauss-Seidel algorithms have the following forms:

Algorithm 2.6.1 Nonlinear Gauss-Jacobi Algorithm

```
repeat
  forall (r in  $n_n$ )
    Solve  $F_{R_r}(y_1^{\eta}, y_2^{\eta}, \dots, y_r^{\eta+1}, \dots, y_{n_n}^{\eta}) = 0$  for  $y_r^{\eta+1}$ 
until ( $\|\mathbf{y}^{\eta+1} - \mathbf{y}^{\eta}\| \leq \varepsilon^R$ )
```

where, F_{R_r} denotes the r^{th} function in vector \mathbf{F}_R and ε^R is any predefined error. The **forall** (p_R in K_R) construct specifies that the computations for all values of p_R in the set K_R may proceed concurrently and in any order.

Algorithm 2.6.2 Nonlinear Gauss-Seidel Algorithm

```
repeat
  foreach (r in  $n_n$ )
    Solve  $F_{R_r}(y_1^{\eta+1}, y_2^{\eta+1}, \dots, y_r^{\eta+1}, \dots, y_{n_n}^{\eta}) = 0$  for  $y_r^{\eta+1}$ 
until ( $\|\mathbf{y}^{\eta+1} - \mathbf{y}^{\eta}\| \leq \varepsilon^R$ )
```

The **foreach** (p_R in K_R) construct specifies that the computations for each value of p_R in the ordered set K_R must proceed sequentially and in the order specified by the set. The actual order for this method in which the node equations are solved may be determined either statically or dynamically.

As the equations shown in Algs. (2.6.1) and (2.6.2) are nonlinear, there is no hope of solving the equations exactly in finite time. So an iterative method is required for this purpose. In general, Newton's method is used to solve these equations. Note that, n_n decoupled equations, each in one unknown must be solved for each relaxation iteration. Thus Newton's method is straightforward to implement here. These *composite* methods are called Gauss-Jacobi-Newton and Gauss-Seidel-Newton method to specify that Newton's method is employed inside the nonlinear Gauss-Jacobi and Gauss-Seidel iteration respectively.

Several electrical analysis techniques have been developed based on relaxation methods. Some analysis techniques are presented here as some of the numerical properties from those are similar to that of his proposed approach.

2.6.1 Timing Simulation

Timing Simulation is the first successful application of relaxation methods for electrical circuit analysis. Only one relaxation iteration is carried out per time step while one or more Newton iterations may be performed to solve each nodal equation in timing simulation. A small time step must be used to bound local errors as the relaxation loop is not taken to convergence. A major drawback with timing analysis is that it may cause severe inaccuracies or even instability for tightly coupled feedback loop or bidirectional circuit elements during the analysis. This ascertains that timing algorithms do not inherit the numerical properties of the discretization method used to approximate the time derivative (*i.e.*, timing simulation may cause instability though the integration method used is stable) [3]. This idiosyncrasy is observed due to the fact that only one sweep of relaxation iteration is taken. These problems hamper the use of timing simulation as a standard simulation tool.

2.6.2 Iterated Timing Analysis

Iterated Timing Analysis (ITA) is derived from timing analysis but instead of a single relaxation iteration per time point, the relaxation process is continued to convergence at a time point in the ITA approach. Starting point of equation formulation for ITA is same as Eq. (2.48). and

(2.49). then an iterative relaxation method (Gauss-Jacobi or Gauss-Seidel) is used to solve them. Only one Newton iteration is performed to approximate the solution of each nodal equation per relaxation iteration. So the mathematical framework of ITA is the nonlinear Gauss-Jacobi-Newton or Gauss-Seidel-Newton method presented before.

Since in ITA the nonlinear circuit equations are solved by an iterative method until satisfactory convergence is achieved, the numerical properties of the integration methods, used for discretization of the circuit equations, are conserved. So the stability problem that may occur in timing simulation is not an issue for ITA, the only concern is that whether the relaxation iteration will converge at each time point when solving the discretized circuit equations or not. It is already shown that relaxation iterations are guaranteed to converge if the Jacobian matrix of the discretized nonlinear equations is diagonally dominant. Circuit equations can be formulated the same way as Eq. (2.48).

$$[\mathbf{C}_R(\mathbf{v}_R, \mathbf{s}_R)]\dot{\mathbf{v}}_R + \mathbf{I}_R(\mathbf{v}_R, \mathbf{s}_R) = 0, \quad \mathbf{v}_R(0) = V_R \quad (2.53)$$

where, $[\mathbf{C}_R] : \mathbb{R}^{n_n} \times \mathbb{R}^{n_n} \rightarrow \mathbb{R}^{n_n \times n_n}$ is a symmetric diagonally dominant matrix-value function in which $\sphericalangle C_{R\tau l}(\mathbf{v}_R, \mathbf{s}_R)$; $\tau \neq l$ is the total floating capacitance between node τ and l , $C_{R\tau\tau}(\mathbf{v}_R, \mathbf{s}_R)$ is the sum of the capacitances of all the capacitors connected to node τ and $\mathbf{I}_R : \mathbb{R}^{n_n} \times \mathbb{R}^{n_n} \rightarrow \mathbb{R}^{n_n}$ is a continuous function, each component of which represents the net current charging the capacitor at a node due to other conductive elements. If capacitance matrix $[\mathbf{C}_R(\mathbf{v}_R, \mathbf{s}_R)]$ is assumed to be symmetric and positive definite, the Jacobian matrix of discretized nonlinear equations is diagonally dominant provided that the time step is small enough. In fact, the time step appears here as a scaling parameter that increases the role of the capacitance matrix in the Jacobian matrix, when it is discretized. If $[\mathbf{C}_R(\mathbf{v}_R, \mathbf{s}_R)]$ is a matrix of real numbers *i.e.*, the capacitors present in the circuit are all linear, then the discretized equations become,

$$[\mathbf{C}_R](\mathbf{v}_R^{t_R+1} - \mathbf{v}_R^{t_R}) - h_R^{t_R+1} \mathbf{I}_R(\mathbf{v}_R^{t_R+1}, \mathbf{s}_R^{t_R+1}) = 0 \quad (2.54)$$

where, $h_R^{t_R+1}$ is the time step selected at time $t = t_R$. The following theorem has been proven in [18].

THEOREM 2.6.1. [3] *There exists a time step \hat{h}_R strictly positive such that for all $h_R^{t_R+1} \leq \hat{h}_R$, the nonlinear Gauss-Jacobi and nonlinear Gauss-Seidel iteration applied to Eq. (2.54). converge to solution of discretized circuit equations independent of the initial guess.*

2.6.3 Waveform Relaxation Techniques

Timing analysis and iterated timing analysis are based on the application of relaxation techniques to the solution of circuit equations at the nonlinear algebraic equation level, whereas *Waveform Relaxation* (WR) is based on the application of family of relaxation techniques applied to differential equation level. While relaxation techniques in the linear and nonlinear algebraic equation level deal with vectors in \mathbb{R}^{n_n} , WR techniques deal with elements in function spaces *i.e.* *waveforms*. The basic idea of WR techniques is explained here with Gauss-Seidel algorithm.

Let us consider the first-order differential equation in $y_R(t) \in \mathbb{R}^2$ on $t \in [0, T_W]$,

$$\dot{y}_{R_1}(t) = f_{w1}(y_{R_1}, y_{R_2}, t), \quad y_{R_1}(0) = Y_{R10} \quad (2.55)$$

$$\dot{y}_{R_2}(t) = f_{w2}(y_{R_1}, y_{R_2}, t), \quad y_{R_2}(0) = Y_{R20} \quad (2.56)$$

where, y_R is an arbitrary variable and T_W is the corresponding time period. The basic idea is to fix the waveform $y_{R_2} : [0, T_W] \rightarrow \mathbb{R}$ and solve Eq. (2.55), as one-dimensional differential equation in $y_{R_1}(\cdot)$. The solution thus obtained for y_{R_1} is substituted into Eq. (2.56). which will then reduce to another first-order differential equation in one variable, y_{R_2} . Eq. (2.55), is then resolved with the new solution found for $y_{R_2}(t)$ and the process will be repeated.

An iterative algorithm has been developed in this fashion. This simplifies the problem of solving a differential equation in two variables by solving a sequence of differential equations in one variable. Here, the unknowns to be solved are the waveforms (elements of a function space), rather than real variables. In this sense, the Gauss-Seidel WR algorithm is a technique for time-domain decoupling of differential equations.

Circuit equations for the Gauss-Seidel WR technique can be formulated the same way as ITA shown in Eq. (2.53). A simplified algorithm to simulate the circuit with this technique is presented in Alg. (2.6.3):

Here, $C_{Rr,l}(\cdot)$; $r \neq l$ is the total floating capacitance between nodes r and l , I_{R_r} is the r^{th} function of the vector function $\mathbf{I}_R(\cdot)$, v_{R_r} is the r^{th} node of the circuit, η is the WR iteration number and ε^R is the predefined error. The only unknown variable is $\mathbf{v}_{R_r}^\eta$, the variables $\mathbf{v}_{R_{r+1}}^{\eta-1}, \dots, \mathbf{v}_{R_{n_n}}^{\eta-1}$ are known from the previous iteration and $\mathbf{v}_{R_1}^\eta, \dots, \mathbf{v}_{R_{r-1}}^\eta$ already been computed. The presented Gauss-Seidel algorithm can easily be modified to obtain Gauss-Jacobi version of WR technique

Algorithm 2.6.3 WR Gauss-Seidel Algorithm for Solving Eq. (2.53)

```

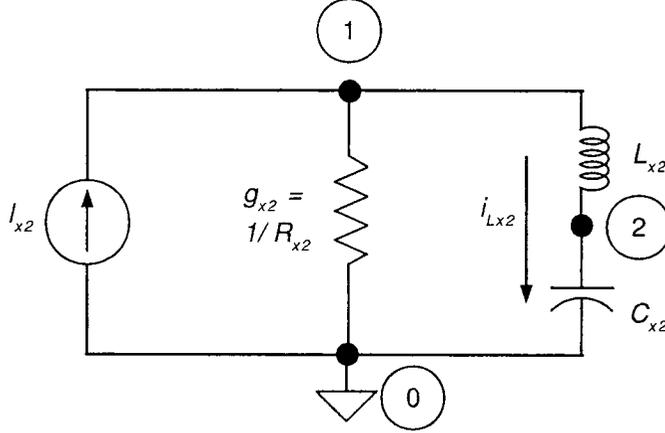
 $\eta \leftarrow 0$ 
guess waveform  $\mathbf{v}_R^0(t)$ ;  $t \in [0, T_W]$  such that  $\mathbf{v}_R^0(0) = V_R$ 
  {for example, set  $\mathbf{v}_R^0(t) = V_R, t \in [0, T_W]$ }
repeat
   $\eta \leftarrow \eta + 1$ 
  foreach ( $r$  in  $n_n$ )
    Solve  $\sum_{l=1}^r C_{Rrl} \left( v_{R_1}^\eta, \dots, v_{R_r}^\eta, v_{R_{r+1}}^{\eta-1}, \dots, v_{R_{n_n}}^{\eta-1}, \mathbf{s}_R \right) \dot{v}_{R_l}^\eta +$ 
       $\sum_{l=r+1}^{n_n} C_{Rrl} \left( v_{R_1}^\eta, \dots, v_{R_r}^\eta, v_{R_{r+1}}^{\eta-1}, \dots, v_{R_{n_n}}^{\eta-1}, \mathbf{s}_R \right) \dot{v}_{R_l}^{\eta-1} +$ 
       $I_{Rr} \left( v_{R_1}^\eta, \dots, v_{R_r}^\eta, v_{R_{r+1}}^{\eta-1}, \dots, v_{R_{n_n}}^{\eta-1}, \mathbf{s}_R \right)$ 
  until ( $\max_{1 \leq r \leq n_n} \max_{t \in [0, T_W]} |v_{R_r}^\eta(t) - v_{R_r}^{\eta-1}(t)| \leq \varepsilon^R$ )
```

by simply replacing **foreach** statement with **forall** and adjusting the indices as mentioned in Alg. (2.6.1).

2.6.4 Simulation of a simple RLC circuit

Let us consider a simple *RLC* circuit (Fig. 2.23) to explain the WR algorithms. Here, $I_{x2} = 0$ at $t = 0$ and $I_{x2} = 1$ at $t > 0$, time step, $h_{x2} = 10 \mu s$ and time period, $T_{x2} = 8 ms$. Circuit element values are, $g_{x2} = \frac{1}{R_{x2}} = \frac{1}{10 \Omega}$, $L_{x2} = 10 mH$ and $C_{x2} = 100 \mu F$. The unknowns in the circuit are voltage at node 1, v_{x21} , voltage at node 2, v_{x22} and current through the inductor, i_{Lx2} . Using the *Modified Nodal Analysis* (MNA) (will be illustrated in App. D) to formulate the circuit equations, the following equations are obtained,

$$\begin{aligned}
 & \left[\begin{array}{cc|c} g_{x2} & 0 & 1 \\ 0 & 0 & -1 \\ -1 & +1 & 0 \end{array} \right] \left[\begin{array}{c} v_{x21} \\ v_{x22} \\ i_{Lx2} \end{array} \right] + \left[\begin{array}{cc|c} 0 & 0 & 0 \\ 0 & C_{x2} & 0 \\ 0 & 0 & L_{x2} \end{array} \right] \left[\begin{array}{c} \dot{v}_{x21} \\ \dot{v}_{x22} \\ \dot{i}_{Lx2} \end{array} \right] = \left[\begin{array}{c} I_{x2} \\ 0 \\ 0 \end{array} \right] \\
 & \implies [\mathbf{M}_{x2}] \left[\begin{array}{c} v_{x21} \\ v_{x22} \\ i_{Lx2} \end{array} \right] + [\mathbf{C}_{x2}] \left[\begin{array}{c} \dot{v}_{x21} \\ \dot{v}_{x22} \\ \dot{i}_{Lx2} \end{array} \right] = \left[\begin{array}{c} I_{x2} \\ 0 \\ 0 \end{array} \right] \tag{2.57}
 \end{aligned}$$

Figure 2.23: WR example with a simple RLC circuit

where, $[\mathbf{M}_{x2}]$ is the modified nodal admittance matrix (MNAM) and $[\mathbf{C}_{x2}]$ is the matrix containing the capacitor and inductor values.

1. Gauss-Seidel Algorithm

Applying WR scheme using this algorithm and time discretization, three equations in Eq (2.57). can be expressed as follows:

$$\begin{aligned}
 M_{x2}^{11} v_{x21}^{\eta+1} + M_{x2}^{12} v_{x22}^{\eta} + M_{x2}^{13} i_{Lx2}^{\eta} + C_{x2}^{11} \left(\frac{v_{x21}^{\eta+1} - v_{x21}^{\eta+1} t_{R-1}}{h_{x2}} \right) + \\
 C_{x2}^{12} \left(\frac{v_{x22}^{\eta} - v_{x22}^{\eta} t_{R-1}}{h_{x2}} \right) + C_{x2}^{13} \left(\frac{i_{Lx2}^{\eta} - i_{Lx2}^{\eta} t_{R-1}}{h_{x2}} \right) = I_{x2} t_R \\
 \implies g_{x2} v_{x21}^{\eta+1} = I_{x2} t_R - i_{Lx2}^{\eta}
 \end{aligned} \tag{2.58}$$

where, η is the WR iteration number, M_{x2}^{rl} and C_{x2}^{rl} are the entries of r^{th} row and l^{th} column

of matrix $[M_{x2}]$ and $[C_{x2}]$ respectively and t_R is the time point used for discretization.

$$\begin{aligned}
& M_{x2}^{21} v_{x21 t_R}^{\eta+1} + M_{x2}^{22} v_{x22 t_R}^{\eta+1} + M_{x2}^{23} i_{Lx2 t_R}^{\eta} + C_{x2}^{21} \left(\frac{v_{x21 t_R}^{\eta+1} - v_{x21 t_{R-1}}^{\eta+1}}{h_{x2}} \right) + \\
& C_{x2}^{22} \left(\frac{v_{x22 t_R}^{\eta+1} - v_{x22 t_{R-1}}^{\eta+1}}{h_{x2}} \right) C_{x2}^{23} \left(\frac{i_{Lx2 t_R}^{\eta} - i_{Lx2 t_{R-1}}^{\eta}}{h_{x2}} \right) = 0 \\
& \implies \frac{C_{x2}}{h_{x2}} v_{x22 t_R}^{\eta+1} = \frac{C_{x2}}{h_{x2}} v_{x21 t_{R-1}}^{\eta+1} + i_{Lx2 t_R}^{\eta}
\end{aligned} \tag{2.59}$$

$$\begin{aligned}
& M_{x2}^{31} v_{x21 t_R}^{\eta+1} + M_{x2}^{32} v_{x22 t_R}^{\eta+1} + M_{x2}^{33} i_{Lx2 t_R}^{\eta+1} + C_{x2}^{31} \left(\frac{v_{x21 t_R}^{\eta+1} - v_{x21 t_{R-1}}^{\eta+1}}{h_{x2}} \right) + \\
& C_{x2}^{32} \left(\frac{v_{x22 t_R}^{\eta+1} - v_{x22 t_{R-1}}^{\eta+1}}{h_{x2}} \right) C_{x2}^{33} \left(\frac{i_{Lx2 t_R}^{\eta+1} - i_{Lx2 t_{R-1}}^{\eta+1}}{h_{x2}} \right) = 0 \\
& \implies \frac{L_{x2}}{h_{x2}} i_{Lx2 t_R}^{\eta+1} = \frac{L_{x2}}{h_{x2}} i_{Lx2 t_{R-1}}^{\eta+1} + v_{x21 t_R}^{\eta+1} - v_{x22 t_R}^{\eta+1}
\end{aligned} \tag{2.60}$$

Note that Eqs. (2.58), (2.59), (2.60). has to be solved sequentially, *i.e.*, value of $v_{x21 t_R}^{\eta+1}$ calculated from Eq. (2.58), is used to find the value of $v_{x22 t_R}^{\eta+1}$ in Eq. (2.59), and both the values are used to calculate the value of $i_{Lx2 t_R}^{\eta+1}$ in Eq. (2.60). and the process is repeated until convergence is achieved.

2. Gauss-Jacobi Algorithm

Similarly Gauss-Jacobi algorithm can be used to solve the equations as follows:

$$\begin{aligned}
& M_{x2}^{11} v_{x21 t_R}^{\eta+1} + M_{x2}^{12} v_{x22 t_R}^{\eta} + M_{x2}^{13} i_{Lx2 t_R}^{\eta} + C_{x2}^{11} \left(\frac{v_{x21 t_R}^{\eta+1} - v_{x21 t_{R-1}}^{\eta}}{h_{x2}} \right) + \\
& C_{x2}^{12} \left(\frac{v_{x22 t_R}^{\eta} - v_{x22 t_{R-1}}^{\eta}}{h_{x2}} \right) C_{x2}^{13} \left(\frac{i_{Lx2 t_R}^{\eta} - i_{Lx2 t_{R-1}}^{\eta}}{h_{x2}} \right) = I_{x2 t_R} \\
& \implies g_{x2} v_{x21 t_R}^{\eta+1} = I_{x2 t_R} - i_{Lx2 t_R}^{\eta}
\end{aligned} \tag{2.61}$$

$$\begin{aligned}
& M_{x2}^{21} v_{x21 t_R}^{\eta} + M_{x2}^{22} v_{x22 t_R}^{\eta+1} + M_{x2}^{23} i_{Lx2 t_R}^{\eta} + C_{x2}^{21} \left(\frac{v_{x21 t_R}^{\eta} - v_{x21 t_{R-1}}^{\eta}}{h_{x2}} \right) + \\
& C_{x2}^{22} \left(\frac{v_{x22 t_R}^{\eta+1} - v_{x22 t_{R-1}}^{\eta+1}}{h_{x2}} \right) C_{x2}^{23} \left(\frac{i_{Lx2 t_R}^{\eta} - i_{Lx2 t_{R-1}}^{\eta}}{h_{x2}} \right) = 0 \\
& \implies \frac{C_{x2}}{h_{x2}} v_{x22 t_R}^{\eta+1} = \frac{C_{x2}}{h_{x2}} v_{x21 t_{R-1}}^{\eta} + i_{Lx2 t_R}^{\eta}
\end{aligned} \tag{2.62}$$

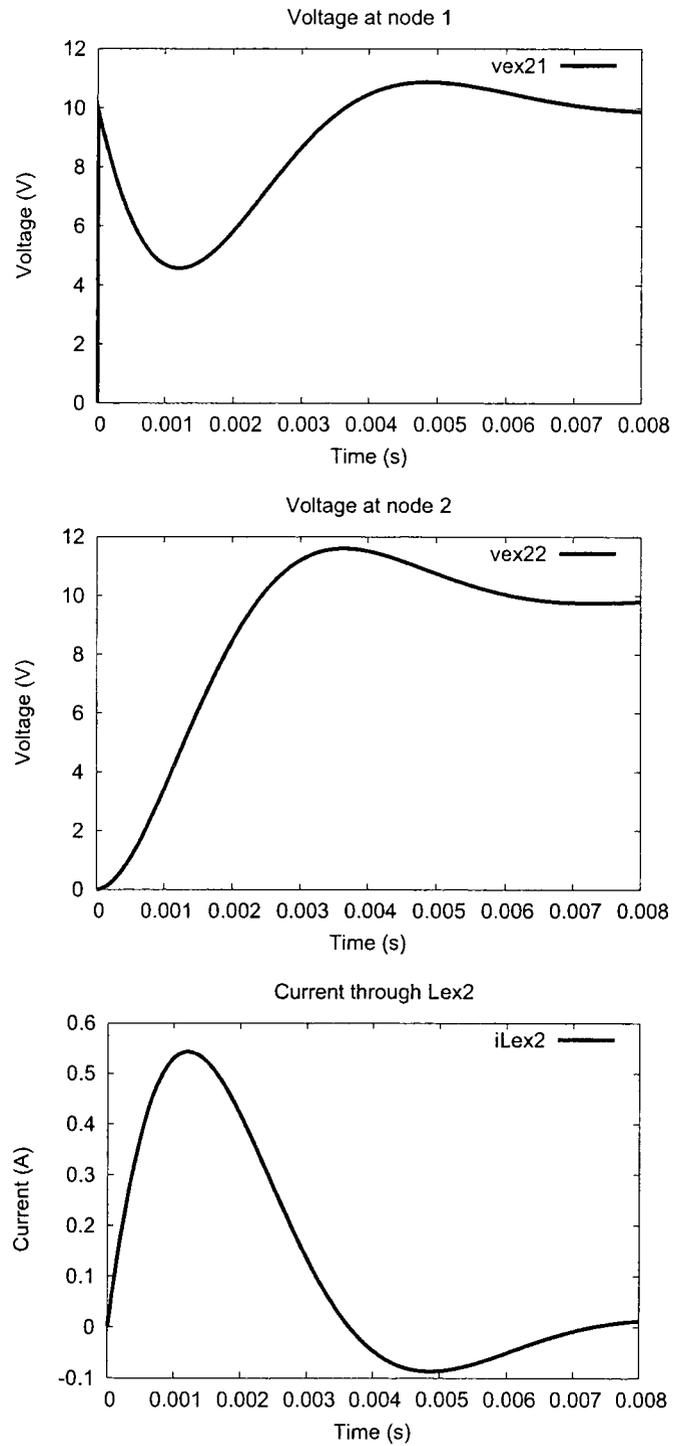


Figure 2.24: Outputs for the example in Fig. 2.23

$$\begin{aligned}
& M_{x2}^{31} v_{x21t_R}^\eta + M_{x2}^{32} v_{x22t_R}^\eta + M_{x2}^{33} i_{Lx2t_R}^{\eta+1} + C_{x2}^{31} \left(\frac{v_{x21t_R}^\eta - v_{x21t_{R-1}}^\eta}{h_{x2}} \right) + \\
& C_{x2}^{32} \left(\frac{v_{x22t_R}^\eta - v_{x22t_{R-1}}^\eta}{h_{x2}} \right) C_{x2}^{33} \left(\frac{i_{Lx2t_R}^{\eta+1} - i_{Lx2t_{R-1}}^\eta}{h_{x2}} \right) = 0 \\
& \implies \frac{L_{x2}}{h_{x2}} i_{Lx2t_R}^{\eta+1} = \frac{L_{x2}}{h_{x2}} i_{Lx2t_{R-1}}^\eta + v_{x21t_R}^\eta - v_{x22t_R}^\eta
\end{aligned} \tag{2.63}$$

Here, Eqs. (2.61), (2.62), (2.63). depends only on the values from previous iteration and thus can be solved concurrently. The outputs are shown in Fig. 2.24.

2.7 Minimum Polynomial Extrapolation

Minimum polynomial extrapolation (MPE) is a sequence transformation method used for convergence acceleration of vector sequences or transforming divergent vector sequences to convergent. This method does not require precise knowledge of sequence generators but computation is performed directly from the terms of the sequences. Furthermore, MPE tends to converge quadratically when applied iteratively to a sequence generated nonlinearly [16]. MPE can be employed to both linear and nonlinear sequences. The theory of MPE applied to linear sequences as presented in Reference [16] is explored here.

Let, a sequence of vectors is generated linearly from a starting point \mathbf{c}_0

$$\mathbf{c}_{r+1} = [\mathbf{M}_f] \mathbf{c}_r + \mathbf{d} \tag{2.64}$$

where $r = 0, 1, 2, \dots$, $[\mathbf{M}_f]$ is a fixed matrix and \mathbf{d} is a fixed vector. if the eigenvalue of $[\mathbf{M}_f]$, $\lambda_m \neq 1$ and $||[\mathbf{M}_f]|| < 1$ then iterative sequence 2.64 has a unique fixed point,

$$\begin{aligned}
\mathbf{c}_P &= [\mathbf{M}_f] \mathbf{c}_P + \mathbf{d}, \\
\implies \mathbf{c}_P &= [\mathbf{I} - \mathbf{M}_f]^{-1} \mathbf{d}
\end{aligned} \tag{2.65}$$

where \mathbf{c}_P is the fixed point of the sequence. If all eigenvalues of $[\mathbf{M}_f]$, $|\lambda_m| < 1$ than

$$\mathbf{c}_P = \lim_{r \rightarrow \infty} \mathbf{c}_r$$

\mathbf{c}_P is called *limit* when the sequence converges and is called *anti-limit* when the sequence diverges. The objective is to determine \mathbf{c}_P from a finite number of terms and limit the number as few as possible without requiring any additional information about $[\mathbf{M}_f]$ and without inverting $n_m \times n_m$ matrix (n_m is the size of the vectors). Typically, n_m may be quite large relative to the number

of eigenvalues λ_m with $|\lambda_m|$ near 1 (causing slow convergence) or greater than 1 (usually causing divergence) and expensive to compute (The vector \mathbf{d} can always be computed by starting the iteration at $\mathbf{c}_0 = 0$).

The extrapolation method to be derived is based on differences of the vectors. The following difference vectors can be defined,

$$\begin{aligned}\mathbf{w}_r &= \Delta \mathbf{c}_r \\ &= \mathbf{c}_{r+1} - \mathbf{c}_r\end{aligned}\tag{2.66}$$

and from the above equations the differences for each value of r can be related,

$$\begin{aligned}\mathbf{w}_{r+1} &= \mathbf{c}_{r+2} - \mathbf{c}_{r+1} \\ &= [\mathbf{M}_f] \mathbf{c}_{r+1} + \mathbf{d} - [\mathbf{M}_f] \mathbf{c}_r - \mathbf{d} \\ &= [\mathbf{M}_f] (\mathbf{c}_{r+1} - \mathbf{c}_r) \\ &= [\mathbf{M}_f] \mathbf{w}_r\end{aligned}\tag{2.67}$$

If the initial difference vector is \mathbf{w}_0 , then \mathbf{w}_{r+1} can be expressed as follows:

$$\begin{aligned}\mathbf{w}_{r+1} &= [\mathbf{M}_f] \mathbf{w}_r \\ &= [\mathbf{M}_f] [\mathbf{M}_f] \mathbf{w}_{r-1} \\ &\vdots \\ &= [\mathbf{M}_f]^{r+1} \mathbf{w}_0\end{aligned}\tag{2.68}$$

For a fixed integer K_m , a matrix of size $n_m \times K_m$ can be defined whose columns are the vectors of differences

$$[\mathbf{W}] \equiv [\mathbf{W}_{K_m}] = [\mathbf{w}_0, \mathbf{w}_1, \dots, \mathbf{w}_{K_m-1}]\tag{2.69}$$

In this method \mathbf{c}_P will be calculated as a weighted average of \mathbf{c}_r s, with weights determined by the coefficients of *minimal polynomial* [21] $\mathbf{P}(\lambda_m)$ of $[\mathbf{M}_f]$ with respect to \mathbf{w}_0 *i.e.*, the unique *monic polynomial* [25] of least degree such that

$$\mathbf{P}([\mathbf{M}_f])\mathbf{w}_0 = 0\tag{2.70}$$

Here, K_m is assumed as the degree of \mathbf{P} or number of roots of *characteristic polynomial* of $[\mathbf{M}_f]$ or the number of eigenvalues of $[\mathbf{M}_f]$. If $K_m \leq n_m$ it can written,

$$\mathbf{P}(\lambda_m) = \sum_{r=0}^{K_m} q_r \lambda_m^r, \quad q_{K_m} = 1$$

where, q_r ; ($r = 0, 1, 2, \dots, K_m - 1$) is the coefficient of r^{th} degree of λ_m of the polynomial. From Eq. (2.68). and (2.70). it follows that

$$\begin{aligned} \mathbf{P}([\mathbf{M}_f])\mathbf{w}_0 &= 0 \\ \implies \sum_{r=0}^{K_m} q_r [\mathbf{M}_f]^r \mathbf{w}_0 &= 0 \\ \therefore \sum_{r=0}^{K_m} q_r \mathbf{w}_r &= 0 \end{aligned} \quad (2.71)$$

Expanding this equation, a system of equations is found, solution of that is sufficient to find the value of coefficients. From Eq. (2.71). it follows,

$$\begin{aligned} \sum_{r=0}^{K_m-1} q_r \mathbf{w}_r + q_{K_m} \mathbf{w}_{K_m} &= 0 \\ \implies [\mathbf{W}] \mathbf{q} &= -\mathbf{w}_{K_m} \end{aligned} \quad (2.72)$$

so the vector $\mathbf{q} = [q_0, q_1, \dots, q_{K_m-1}]^T$ of unknown coefficients of \mathbf{P} is a solution of the system of equations.

The unique solution of Eq. (2.72). can be written as

$$\mathbf{q} = -[\mathbf{W}]^+ \mathbf{w}_{K_m} \quad (2.73)$$

where $[\mathbf{W}]^+$ is the pseudoinverse [14] (or Moore-Penrose generalized inverse [6]) of $[\mathbf{W}]$. In principle, computation of $[\mathbf{W}]^+$ require the inversion of $K_m \times K_m$ matrix. To compute the fixed-point \mathbf{c}_P a Theorem regarding this will be demonstrated.

THEOREM 2.7.1. [16] For $K_{th} + 1$ consecutive terms ($K_{th} < \text{size of the vector}, n_m$) of the sequence, say $\mathbf{c}_{thm}, \mathbf{c}_{thm+1}, \dots,$

\mathbf{c}_{thm+K_m} , it can be stated that

$$\sum_{r=0}^{K_{th}} q_{thr} \mathbf{c}_{thm+r} = \left(\sum_{r=0}^{K_{th}} q_{thr} \right) \mathbf{c}_{Pth}$$

where, q_{th} is the coefficients as defined above and \mathbf{c}_{Pth} is the fixed-point of the sequence.

So the fixed-point of the sequence defined by Eq. (2.64). can be determined by the following equation:

$$\begin{aligned} \sum_{r=0}^{K_m-1} q_r \mathbf{c}_r &= \left(\sum_{r=0}^{K_m-1} q_r \right) \mathbf{c}_P \\ \therefore \mathbf{c}_P &= \frac{\sum_{r=0}^{K_m-1} q_r \mathbf{c}_r}{\sum_{r=0}^{K_m-1} q_r} \end{aligned} \quad (2.74)$$

MPE extrapolation method can be summarized for a given sequence generator of the form Eq. (2.64). and a starting point \mathbf{c}_0

1. generate $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{K_m+1}$;
2. compute $[\mathbf{W}]$, \mathbf{w}_{K_m} as in Eq. (2.66), and (2.69).
3. compute \mathbf{q} as in Eq. (2.72), (2.73). and set $q_{K_m} = 1$;
4. compute \mathbf{c}_P from Eq. (2.74).

Chapter 3

Formulation

3.1 Generalized Circuit Formulation

Formulation of system equations begins with the partitioned network shown in Fig. 3.1 with the nonlinear elements replaced by variable voltage or current sources [38]. For each nonlinear element, ports are defined with one terminal taken as the reference and the elements are replaced by a set of sources connected to the reference terminal. Both voltage and current sources can be used to replace the nonlinear elements but current sources are preferable because they provide a smaller modified nodal admittance matrix (MNAM) [38].

3.1.1 Linear Network

Modified Nodal Analysis (App. D) can be utilized to formulate the linear portion and sources (both independent and dependent) of the network shown in Fig. 3.1. So generally the contribution of linear network and sources can be expressed as follows:

$$[\mathbf{M}]\mathbf{u}(t) + [\mathbf{C}]\frac{d\mathbf{u}(t)}{dt} = \mathbf{S}_s(t) \quad (3.1)$$

The matrices $[\mathbf{M}]$ and $[\mathbf{C}]$ are of size $n_{un} \times n_{un}$, where $n_{un} = (n_{nd} - 1) + n_{ad}$ (n_{nd} is the total number of nodes and n_{ad} is the number of additional variables) is the total number of variables to be solved. $[\mathbf{M}]$ contains all conductors and frequency-independent MNAM stamps arising in the formulation, whereas $[\mathbf{C}]$ consists of capacitor and inductor values and other values that are associated with dynamic elements. \mathbf{S}_s is the source vector of size n_{un} for the right hand side of the system that include the contributions of the independent sources and the nonlinear elements (which depend on the time t). \mathbf{u} is the vector of nodal voltages and selected currents. Source

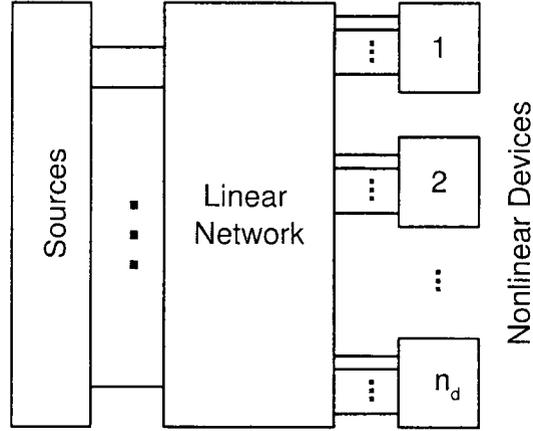


Figure 3.1: Network Partition

vector \mathbf{S}_s can be expanded as follows:

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

where \mathbf{S}_f is the vector of independent sources and \mathbf{S}_v is the current contribution of nonlinear elements (nonlinear elements are replaced by a number of current sources). So Eq. (3.1). can be rewritten as follows:

$$[\mathbf{M}]\mathbf{u}(t) + [\mathbf{C}]\frac{d\mathbf{u}(t)}{dt} = \mathbf{S}_f(t) + \mathbf{S}_v(t) \quad (3.3)$$

Discretization of Eq. (3.3). by *backward Euler* formula,

$$\begin{aligned} [\mathbf{M}]\mathbf{u}^t + [\mathbf{C}]\frac{\mathbf{u}^t - \mathbf{u}^{t-1}}{h} &= \mathbf{S}_f^t + \mathbf{S}_v^t \\ \Rightarrow \left([\mathbf{M}] + \left[\frac{\mathbf{C}}{h}\right]\right)\mathbf{u}^t &= \mathbf{S}_f^t + \mathbf{S}_v^t + \left[\frac{\mathbf{C}}{h}\right]\mathbf{u}^{t-1} \end{aligned} \quad (3.4)$$

where superscript t denotes discretized time points and h is the time step size. Discretization of Eq. (3.3). by *trapezoidal rule*,

$$\begin{aligned} [\mathbf{M}]\mathbf{u}^t + [\mathbf{C}]\left\{\frac{2}{h}(\mathbf{u}^t - \mathbf{u}^{t-1}) - \mathbf{u}^{t-1}\right\} &= \mathbf{S}_f^t + \mathbf{S}_v^t \\ \Rightarrow \left([\mathbf{M}] + \left[\frac{2\mathbf{C}}{h}\right]\right)\mathbf{u}^t &= \mathbf{S}_f^t + \mathbf{S}_v^t + \left(\left[\frac{2\mathbf{C}}{h}\right]\mathbf{u}^{t-1} + \mathbf{u}^{t-1}\right) \end{aligned} \quad (3.5)$$

Eq. (3.4). and (3.5). can be expressed in a generalized form as follows:

$$\begin{aligned} [\mathbf{M}_C]\mathbf{u}^t &= \mathbf{S}_f^t + \mathbf{S}_v^t + \mathbf{S}_{ni}^t \\ \Rightarrow [\mathbf{M}_C]\mathbf{u}^t &= \mathbf{S}_l^t + \mathbf{S}_v^t \end{aligned} \quad (3.6)$$

where $[\mathbf{M}_C]$ contains both the elements of $[\mathbf{M}]$ and $[\mathbf{C}]$ matrices and contributions from previous time step can be expressed in another source vector, \mathbf{S}_{ni}^t . \mathbf{S}_l^t include the source contribution from linear side (\mathbf{S}_f^t and \mathbf{S}_{ni}^t). Eq. (3.6), is the generalized form of the system of equations derived from linear side of the network.

3.1.2 Nonlinear Network

In FREEDA, the nonlinear subnetwork is described by the following parametric equations [57]:

$$\mathbf{v}_{NL}(t) = \mathbf{v} \left(\mathbf{x}(t), \frac{d\mathbf{x}}{dt}, \dots, \frac{d^{\text{od}}\mathbf{x}}{dt^{\text{od}}}, \mathbf{x}_D(t) \right) \quad (3.7)$$

$$\mathbf{i}_{NL}(t) = \mathbf{i} \left(\mathbf{x}(t), \frac{d\mathbf{x}}{dt}, \dots, \frac{d^{\text{od}}\mathbf{x}}{dt^{\text{od}}}, \mathbf{x}_D(t) \right) \quad (3.8)$$

where $\mathbf{v}_{NL}(t)$, $\mathbf{i}_{NL}(t)$ are vectors of voltages and currents at the ports of the nonlinear network, $\mathbf{x}(t)$ is a vector of state variables $\text{od} = 1, 2, 3, \dots$ is the order of derivatives of state variables and $\mathbf{x}_D(t)$ is a vector of time-delayed state variables, *i.e.*, $[x_D(t)]_i = x_i(t - \tau_i)$. All vectors in equations. (3.7), and (3.8), have the same size equal to the number of ports of the nonlinear network (n_{nl}). Representation of this kind is expedient from the physical point of view, as it is analogous to a set of implicit integro-differential equations (*i.e.*, equation with both integrals and derivatives of an unknown function) in the port currents and voltages. This brings in complete generality in device modelling. For example, nonlinear elements are no longer necessary to be expressed as voltage controlled current sources.

Connectivity information described by an incidence matrix and constitutive relations describing the nonlinear elements is used to derive the error function of an arbitrary circuit. The incidence matrix $[\mathbf{T}]$ is used to relate the vectors of the linear and the nonlinear network equations as follows:

$$\mathbf{v}_{NL}(t) = [\mathbf{T}] \mathbf{u}(t) \quad (3.9)$$

$$\mathbf{S}_v(t) = [\mathbf{T}]^T \mathbf{i}_{NL}(t) \quad (3.10)$$

after discretization Eq. (3.9), and (3.10), become,

$$\mathbf{v}_{NL}(\mathbf{x}^t) = [\mathbf{T}] \mathbf{u}^t \quad (3.11)$$

$$\mathbf{S}_v^t = [\mathbf{T}]^T \mathbf{i}_{NL}(\mathbf{x}^t) \quad (3.12)$$

The matrix $[\mathbf{T}]$ contains n_{un} columns and the number of rows is equal to the number of state variables, n_{st} (number of state variables is equal to the number of nonlinear device ports, *i.e.*, $n_{nl} = n_{st}$). In each row, enter “+1” in the column corresponding to the positive terminal of nonlinear element port and “-1” in the column corresponding to the negative terminal (the local reference of the port). So each row of $[\mathbf{T}]$ has at most 2 nonzero elements and total number of nonzero elements is at most $2n_{nl}$.

3.1.3 Error Function Formulation

Combining Eq. (3.6), (3.11), and (3.12), the general equation for the network is as follows:

$$\begin{aligned} \mathbf{v}_{NL}(\mathbf{x}^t) &= [\mathbf{T}] [\mathbf{M}_C]^{-1} \mathbf{S}_i^t + [\mathbf{T}] [\mathbf{M}_C]^{-1} [\mathbf{T}]^T \mathbf{i}_{NL}(\mathbf{x}^t) \\ \implies [\mathbf{T}] [\mathbf{M}_C]^{-1} \mathbf{S}_i^t + [\mathbf{T}] [\mathbf{M}_C]^{-1} [\mathbf{T}]^T \mathbf{i}_{NL}(\mathbf{x}^t) - \mathbf{v}_{NL}(\mathbf{x}^t) &= 0 \\ \therefore \mathbf{S}_{SV}^t + [\mathbf{M}_{SV}] \mathbf{i}_{NL}(\mathbf{x}^t) - \mathbf{v}_{NL}(\mathbf{x}^t) &= 0 \end{aligned} \quad (3.13)$$

where $[\mathbf{M}_{SV}]$ is a constant impedance matrix of size $n_{nl} \times n_{nl}$ and \mathbf{S}_{SV}^t is a vector of size n_{nl} and accounts for the source components and the previous history of the linear part. Eq. (3.13). is the reduced *error function* for the system. Important observation regarding to this error function is that the dimension of the error function and the number of unknowns is equal to n_{nl} though the formulation started from n_{un} unknowns ($n_{nl} < n_{un}$), and this number is the minimum necessary to solve the equations of a circuit without any loss of information.

3.2 Transformation to Power Waves

Let us imagine that each nonlinear device port is connected to the linear network by a zero-length lossless transmission line with characteristic impedance Z_r , with r being the port number. Since the transmission line does not really exist, Z_r will be referred as the port reference

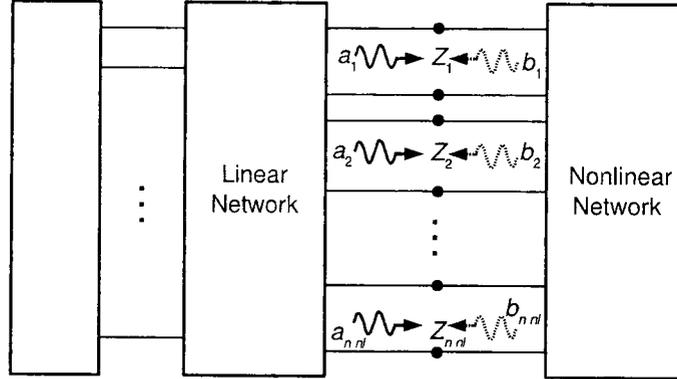


Figure 3.2: Transformation to power waves

impedance. Power waves (a_r and b_r) can be defined in that transmission line as follows,

$$a_r = \frac{v_r^+}{\sqrt{Z_r}},$$

$$b_r = \frac{v_r^-}{\sqrt{Z_r}}$$

where v_r^+ and v_r^- are the values of incident and reflected voltage waves in the transmission line as seen from the nonlinear device side, *i.e.*, the convention adopted here is that a_r is incident and b_r is reflected wave from the nonlinear device. Note that the instantaneous power flow to the nonlinear device in port r is equal to $a_r^2 - b_r^2$. The node voltage (v_r) and current (i_r) at port r can be expressed as follows:

$$v_r = \sqrt{Z_r}(a_r + b_r),$$

$$i_r = \frac{a_r - b_r}{\sqrt{Z_r}}$$

Thus power wave vectors at a given time step (\mathbf{a}^t and \mathbf{b}^t) must satisfy:

$$\mathbf{v}_{NL}(\mathbf{x}^t) = [\mathbf{D}](\mathbf{a}^t + \mathbf{b}^t) \quad (3.14)$$

$$\mathbf{i}_{NL}(\mathbf{x}^t) = [\mathbf{D}]^{-1}(\mathbf{a}^t - \mathbf{b}^t) \quad (3.15)$$

where $[\mathbf{D}]$ is a diagonal matrix with the square root of reference port impedances. Error function is transformed to power waves by combining Eq. (3.13), (3.14), and (3.15).

$$\begin{aligned}
& \mathbf{S}_{SV}^t + [\mathbf{M}_{SV}] [\mathbf{D}]^{-1} (\mathbf{a}^t - \mathbf{b}^t) - [\mathbf{D}] (\mathbf{a}^t + \mathbf{b}^t) = 0, \\
\Rightarrow & [\mathbf{M}_{SV} \mathbf{D}^{-1} - \mathbf{D}] \mathbf{a}^t - [\mathbf{M}_{SV} \mathbf{D}^{-1} + \mathbf{D}] \mathbf{b}^t + \mathbf{S}_{SV}^t = 0, \\
\Rightarrow & [\mathbf{M}_{SV} \mathbf{D}^{-1} - \mathbf{D}] \mathbf{a}^t = [\mathbf{M}_{SV} \mathbf{D}^{-1} + \mathbf{D}] \mathbf{b}^t - \mathbf{S}_{SV}^t, \\
\Rightarrow & \mathbf{a}^t = [\mathbf{M}_{SV} \mathbf{D}^{-1} - \mathbf{D}]^{-1} [\mathbf{M}_{SV} \mathbf{D}^{-1} + \mathbf{D}] \mathbf{b}^t - [\mathbf{M}_{SV} \mathbf{D}^{-1} - \mathbf{D}]^{-1} \mathbf{S}_{SV}^t \quad (3.16)
\end{aligned}$$

with

$$\begin{aligned}
[\mathbf{S}] &= [\mathbf{M}_{SV} \mathbf{D}^{-1} - \mathbf{D}]^{-1} [\mathbf{M}_{SV} \mathbf{D}^{-1} + \mathbf{D}], \\
\mathbf{a}_0^t &= -[\mathbf{M}_{SV} \mathbf{D}^{-1} - \mathbf{D}]^{-1} \mathbf{S}_{SV}^t
\end{aligned}$$

Eq. (3.16). becomes:

$$\mathbf{a}^t = [\mathbf{S}] \mathbf{b}^t + \mathbf{a}_0^t \quad (3.17)$$

where, $[\mathbf{S}]$ is the scattering matrix of the linear network and \mathbf{a}_0^t is the source contribution to the incident waves. Here, the matrix, $[\mathbf{M}_{SV} \mathbf{D}^{-1} - \mathbf{D}]$ is a square and nonsingular matrix of size n_{nl} , so the inverse exists.

3.3 Fixed-Point Iterative scheme

The proposed fixed-point iterative scheme is based on propagating reflections of power waves between the linear and nonlinear subnetworks. Assume an initial vector of reflected waves (\mathbf{b}_n^t) is known, where \mathbf{n} denotes the iteration number. The corresponding waves sent by the linear network (\mathbf{a}_{n+1}^t) can be calculated by Eq. (3.17). which is rewritten as:

$$\mathbf{a}_{n+1}^t = [\mathbf{S}] \mathbf{b}_n^t + \mathbf{a}_0^t \quad (3.18)$$

3.3.1 Nonlinearity

Wave equations for the nonlinear device side are needed to complete the scheme. Under certain conditions (provided originally in [30], described in Subsection 2.4.1 also), the effect of nonlinear devices can be expressed as a nonlinear vector function $\mathbf{F}(\cdot)$,

$$\mathbf{b}_{n+1}^t = \mathbf{F}(\mathbf{a}_{n+1}^t) \quad (3.19)$$

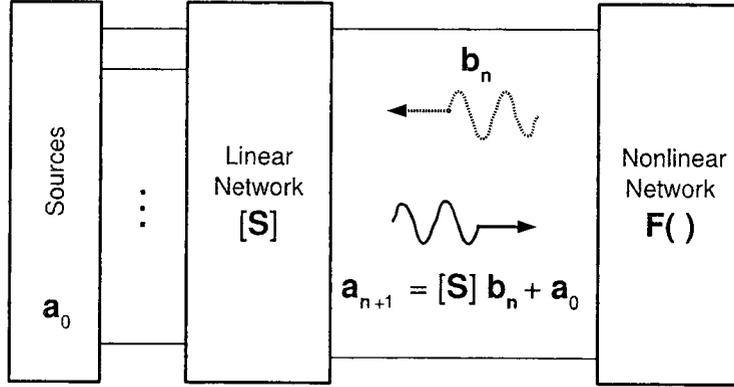


Figure 3.3: Iterative scheme to calculate the value of waves

Eq. (3.19). is not explicitly available in the simulator. A new method is adopted in this work to solve for \mathbf{b}_{n+1}^t . System of equations is separated into the nonlinear devices and numerical methods are adopted to solve this set of decoupled equations. This is an important contribution for developing a parallel algorithm to solve a system of equations with strong nonlinearity. Two different methods are employed to solve this decoupled set of equations, (1) Newton's method and (2) Newton-Jacobi method. Rearranging Eqs. (3.14), and (3.15), the error functions $\mathbf{F}_v(\cdot)$ and $\mathbf{F}_i(\cdot)$ can be obtained:

$$\mathbf{F}_v(\mathbf{x}_{n+1}^t, \mathbf{b}_{n+1}^t) = [\mathbf{D}]^{-1} \mathbf{v}_{NL}(\mathbf{x}_{n+1}^t) - \mathbf{a}_{n+1}^t - \mathbf{b}_{n+1}^t \quad (3.20)$$

$$\mathbf{F}_i(\mathbf{x}_{n+1}^t, \mathbf{b}_{n+1}^t) = [\mathbf{D}] \mathbf{i}_{NL}(\mathbf{x}_{n+1}^t) - \mathbf{a}_{n+1}^t + \mathbf{b}_{n+1}^t \quad (3.21)$$

which must be equal to zero. Total number of equations and unknowns for this system of equations is $2n_{nl}$. The unknowns are vectors \mathbf{b}^t and \mathbf{x}^t as the vector of incident waves, \mathbf{a}^t is known from Eq. (3.18). Note that this system of equations is decoupled for each nonlinear device, thus the Jacobian matrix arising from it is block-diagonal with small diagonal blocks, typically no more than 6×6 elements. The good numerical properties given by the parametric formulation of the nonlinear device equations [38] are retained at the expense of having to solve for the state variable vector (\mathbf{x}^t). The relaxation scheme is summarized by combining Eqs. (3.18), and (3.19).

$$\mathbf{b}_{n+1}^t = \mathbf{F}([\mathbf{S}] \mathbf{b}_n^t + \mathbf{a}_0^t) \quad (3.22)$$

Newton's method

To solve the decoupled system of equations shown in Eqs. (3.20), and (3.21). by Newton's method the formulation the iterative equations are as follows:

$$\begin{aligned} & \begin{bmatrix} \frac{\partial \mathbf{F}_v}{\partial \mathbf{x}} & \frac{\partial \mathbf{F}_v}{\partial \mathbf{b}} \\ \frac{\partial \mathbf{F}_i}{\partial \mathbf{x}} & \frac{\partial \mathbf{F}_i}{\partial \mathbf{b}} \end{bmatrix} \begin{bmatrix} \mathbf{x}^{k+1} - \mathbf{x}^k \\ \mathbf{b}^{k+1} - \mathbf{b}^k \end{bmatrix} = - \begin{bmatrix} \mathbf{F}_v(\mathbf{x}^k, \mathbf{b}^k) \\ \mathbf{F}_i(\mathbf{x}^k, \mathbf{b}^k) \end{bmatrix} \\ \Rightarrow & \begin{bmatrix} \frac{\partial \mathbf{F}_v}{\partial \mathbf{x}} & \frac{\partial \mathbf{F}_v}{\partial \mathbf{b}} \\ \frac{\partial \mathbf{F}_i}{\partial \mathbf{x}} & \frac{\partial \mathbf{F}_i}{\partial \mathbf{b}} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}^{k+1} \\ \Delta \mathbf{b}^{k+1} \end{bmatrix} + \begin{bmatrix} \mathbf{F}_v(\mathbf{x}^k, \mathbf{b}^k) \\ \mathbf{F}_i(\mathbf{x}^k, \mathbf{b}^k) \end{bmatrix} = 0 \end{aligned} \quad (3.23)$$

where k denotes the Newton's iteration number. All the calculations are performed in $\mathbf{n} + 1$ relaxation iteration and time point \mathbf{t} , these indices are omitted in the equations for Newton iteration. Eq. (3.23). is a general representation of the system of equations containing all the nonlinear devices for the calculation during Newton iteration. But the calculation is limited to corresponding set of variables for each nonlinear device, as the Jacobian matrices are all block diagonal based on the nonlinear devices.

Taking the partial differentiation of Eq. (3.20), including all the nonlinear devices it can be found:

$$\begin{aligned} \frac{\partial \mathbf{F}_v}{\partial \mathbf{x}} &= [\mathbf{D}]^{-1} \frac{\partial \mathbf{v}_{NL}(\mathbf{x})}{\partial \mathbf{x}} \\ &= [\mathbf{D}]^{-1} [\mathbf{J}_V] \end{aligned} \quad (3.24)$$

and

$$\frac{\partial \mathbf{F}_v}{\partial \mathbf{b}} = -[\mathbf{I}] \quad (3.25)$$

Similarly taking the partial differentiation of Eq. (3.21). It can be found:

$$\begin{aligned} \frac{\partial \mathbf{F}_i}{\partial \mathbf{x}} &= [\mathbf{D}] \frac{\partial \mathbf{i}_{NL}(\mathbf{x})}{\partial \mathbf{x}_d} \\ &= [\mathbf{D}] [\mathbf{J}_I] \end{aligned} \quad (3.26)$$

and

$$\frac{\partial \mathbf{F}_i}{\partial \mathbf{b}} = [\mathbf{I}] \quad (3.27)$$

Here $[\mathbf{J}_I]$ and $[\mathbf{J}_V]$ are the block-diagonal Jacobian Matrices of \mathbf{v}_{NL} and \mathbf{i}_{NL} , respectively. Note that, both the Jacobian matrices are explicitly available in FREEDA. From Eq. (3.23). It can be found:

$$\begin{aligned} \frac{\partial \mathbf{F}_i}{\partial \mathbf{x}} \Delta \mathbf{x}^{k+1} + \frac{\partial \mathbf{F}_i}{\partial \mathbf{b}} \Delta \mathbf{b}^{k+1} + \mathbf{F}_i &= 0 \\ \Rightarrow [\mathbf{D}] [\mathbf{J}_I] \Delta \mathbf{x}^{k+1} + \mathbf{I} \Delta \mathbf{b}^{k+1} + \mathbf{F}_i &= 0 \\ \therefore \Delta \mathbf{b}^{k+1} &= -\mathbf{F}_i - [\mathbf{D}] [\mathbf{J}_I] \Delta \mathbf{x}^{k+1} \end{aligned} \quad (3.28)$$

and

$$\begin{aligned} \frac{\partial \mathbf{F}_v}{\partial \mathbf{x}} \Delta \mathbf{x}^{k+1} + \frac{\partial \mathbf{F}_v}{\partial \mathbf{b}} \Delta \mathbf{b}^{k+1} + \mathbf{F}_v &= 0 \\ \Rightarrow [\mathbf{D}]^{-1} [\mathbf{J}_V] \Delta \mathbf{x}^{k+1} - [\mathbf{I}] \Delta \mathbf{b}^{k+1} + \mathbf{F}_v &= 0 \end{aligned}$$

using the value of $\Delta \mathbf{b}^{k+1}$ found from Eq. (3.28).

$$\begin{aligned} [\mathbf{D}]^{-1} [\mathbf{J}_V] \Delta \mathbf{x}^{k+1} + \mathbf{F}_i + [\mathbf{D}] [\mathbf{J}_I] \Delta \mathbf{x}^{k+1} + \mathbf{F}_v &= 0 \\ \Rightarrow [\mathbf{D}^{-1} \mathbf{J}_V + \mathbf{D} \mathbf{J}_I] \Delta \mathbf{x}^{k+1} &= -\mathbf{F}_v - \mathbf{F}_i \\ \therefore \Delta \mathbf{x}^{k+1} &= [\mathbf{D}^{-1} \mathbf{J}_V + \mathbf{D} \mathbf{J}_I]^{-1} (-\mathbf{F}_v - \mathbf{F}_i) \end{aligned} \quad (3.29)$$

The updates from each Newton iteration are calculated by Eqs. (3.29), and (3.28), respectively. Matrix decomposition of the corresponding block of Jacobian matrix sum (shown in Eq. (3.29),) has to be performed in each Newton iteration.

3.3.2 Conditions for Local Convergence

Conditions for local convergence of the relaxation method will be presented in the following. Let's assume the solution of Eq. (3.22), is \mathbf{b}_{sol} and iterations are started at $\mathbf{b}_{sol} + \xi_0$, with ξ_0 being an initial perturbation or error vector. By performing Taylor expansion of Eq. (3.22), around \mathbf{b}_{sol} , it follows that the vector of errors at iteration $\mathbf{n} + 1$, $\xi_{\mathbf{n}+1}$ is given by

$$\xi_{\mathbf{n}+1} = [\mathbf{J}_F \mathbf{S}] \xi_{\mathbf{n}}, \quad (3.30)$$

where $[\mathbf{J}_F]$ is the Jacobian matrix of function $\mathbf{F}(\cdot)$ in Eq. (3.22), and represents the small-signal scattering matrix of the nonlinear network. $[\mathbf{J}_F]$ is a block-diagonal matrix since $[\mathbf{J}_I]$ and $[\mathbf{J}_V]$ are block-diagonal. Then the square of 2-norm of $\xi_{\mathbf{n}+1}$ is given by

$$|\xi_{\mathbf{n}+1}|_2 = \xi_{\mathbf{n}}^T [\mathbf{S}]^T [\mathbf{J}_F]^T [\mathbf{J}_F \mathbf{S}] \xi_{\mathbf{n}},$$

where $[\mathbf{J}_F]^T$ denotes the transpose of $[\mathbf{J}_F]$. One sufficient condition for local convergence is that $[\mathbf{J}_F \mathbf{S}]$ corresponds to a passive network. In that case $[\mathbf{I} - \mathbf{S}^T \mathbf{J}_F^T \mathbf{J}_F \mathbf{S}]$ is a *positive semidefinite matrix* [41] ($[\mathbf{I}]$ is the corresponding identity matrix). It follows that $|\xi_{n+1}|_2$ decreases with \mathbf{n} and thus iteration converges eventually. $[\mathbf{S}]$ already corresponds to a passive network, thus his proposed approach is always convergent for any circuit containing only locally passive nonlinear devices. Diodes are known to be locally passive, unfortunately transistors may not be locally passive depending on the bias point.

There is a way to overcome this problem. All practical nonlinear devices have internal parasitic capacitors in parallel with their ports. After time discretization, capacitors appear as conductances and have the effect of *passivizing* the nonlinear devices. Those conductances are inversely proportional to the time step. It is always possible to reduce the time step until all devices become locally passive (a similar reasoning can be made with parasitic inductors in series). This result is somewhat similar to the condition stated in Thm. 2.6.1 for ITA.

Steady state oscillations are observed when the sequences are not convergent. This property is not present in relaxation methods based on voltages and currents.

3.4 Treating Nonlinearity by Newton-Jacobi method

As discussed in Subsection 2.5.2 *Newton-Jacobi method* is based on deriving a block-diagonal pattern of scattering matrix $[\mathbf{S}]$. Block-diagonal matrix, $[\mathbf{S}_B]$ is derived based on each nonlinear device:

$$\begin{bmatrix} S_{11} & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} = \begin{bmatrix} \text{shaded } S_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{shaded} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \text{shaded} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \text{shaded} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \text{shaded} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & \text{shaded} \\ 0 & 0 & 0 & \text{shaded} \\ 0 & 0 & 0 & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & 0 & 0 & \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & 0 & 0 & \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} & 0 & 0 & 0 & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} & 0 & 0 & 0 & \text{shaded} \\ \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} & \text{shaded} & 0 & 0 & 0 & \text{shaded} \end{bmatrix}$$

$$\Rightarrow [\mathbf{S}] = [\mathbf{S}_B] + [\mathbf{S}_O]$$

where, $[\mathbf{S}_O]$ is the part of scattering matrix with the rest of the elements. Thus the waves sent by the linear network (\mathbf{a}_{n+1}^t) are calculated as follows:

$$\mathbf{a}_{n+1}^t = [\mathbf{S}_O] \mathbf{b}_n^t + \mathbf{a}_0^t \quad (3.31)$$

The relaxation scheme based on Newton-Jacobi method is summarized by combining Eqs. (3.19). and (3.31).

$$\mathbf{b}_{n+1}^t = \mathbf{F}([\mathbf{S}_O] \mathbf{b}_n^t + \mathbf{a}_0^t) \quad (3.32)$$

As only one nonlinear device is treated in each iteration of Newton's method, the error functions shown in Eqs. (3.20), and (3.21), are modified as follows:

$$\mathbf{F}_v(\mathbf{x}_{n+1}^t, \mathbf{b}_{n+1}^t) = [\mathbf{D}]^{-1} \mathbf{v}_{NL}(\mathbf{x}_{n+1}^t) - (\mathbf{a}_{n+1}^t + [\mathbf{S}_B] \mathbf{b}_{n+1}^t) - \mathbf{b}_{n+1}^t \quad (3.33)$$

$$\mathbf{F}_i(\mathbf{x}_{n+1}^t, \mathbf{b}_{n+1}^t) = [\mathbf{D}] \mathbf{i}_{NL}(\mathbf{x}_{n+1}^t) - (\mathbf{a}_{n+1}^t + [\mathbf{S}_B] \mathbf{b}_{n+1}^t) + \mathbf{b}_{n+1}^t \quad (3.34)$$

The values of derivatives will be modified as follows:

$$\frac{\partial \mathbf{F}_v}{\partial \mathbf{b}} = -[\mathbf{I}] - [\mathbf{S}_B] \quad (3.35)$$

and

$$\frac{\partial \mathbf{F}_i}{\partial \mathbf{b}} = [\mathbf{I}] - [\mathbf{S}_B] \quad (3.36)$$

From Eq. (3.34), it can be found:

$$\begin{aligned} & \frac{\partial \mathbf{F}_i}{\partial \mathbf{x}} \Delta \mathbf{x}^{k+1} + \frac{\partial \mathbf{F}_i}{\partial \mathbf{b}} \Delta \mathbf{b}^{k+1} + \mathbf{F}_i = 0 \\ \implies & [\mathbf{D}][\mathbf{J}_I] \Delta \mathbf{x}^{k+1} + [\mathbf{I} - \mathbf{S}_B] \Delta \mathbf{b}^{k+1} + \mathbf{F}_i = 0 \\ \therefore & \Delta \mathbf{b}^{k+1} = [\mathbf{I} - \mathbf{S}_B]^{-1} (-\mathbf{F}_i - [\mathbf{D}][\mathbf{J}_I] \Delta \mathbf{x}^{k+1}) \end{aligned} \quad (3.37)$$

and

$$\begin{aligned} & \frac{\partial \mathbf{F}_v}{\partial \mathbf{x}} \Delta \mathbf{x}^{k+1} + \frac{\partial \mathbf{F}_v}{\partial \mathbf{b}} \Delta \mathbf{b}^{k+1} + \mathbf{F}_v = 0 \\ \implies & [\mathbf{D}]^{-1} [\mathbf{J}_V] \Delta \mathbf{x}^{k+1} - [\mathbf{I} + \mathbf{S}_B] \Delta \mathbf{b}^{k+1} + \mathbf{F}_v = 0 \\ \implies & [\mathbf{I} - \mathbf{S}_B] [\mathbf{D}]^{-1} [\mathbf{J}_V] \Delta \mathbf{x}^{k+1} - [\mathbf{I} - \mathbf{S}_B] [\mathbf{I} + \mathbf{S}_B] \Delta \mathbf{b}^{k+1} + [\mathbf{I} - \mathbf{S}_B] \mathbf{F}_v = 0 \end{aligned}$$

as $[\mathbf{I} - \mathbf{S}_B] [\mathbf{I} + \mathbf{S}_B] = [\mathbf{I} - \mathbf{S}_B] [\mathbf{I} + \mathbf{S}_B] = [\mathbf{I} - \mathbf{S}_B^2]$, the equation can be rewritten as follows,

$$[\mathbf{I} - \mathbf{S}_B] [\mathbf{D}]^{-1} [\mathbf{J}_V] \Delta \mathbf{x}^{k+1} - [\mathbf{I} + \mathbf{S}_B] [\mathbf{I} - \mathbf{S}_B] \Delta \mathbf{b}^{k+1} + [\mathbf{I} - \mathbf{S}_B] \mathbf{F}_v = 0$$

using the value of $\Delta \mathbf{b}^{k+1}$ found from Eq. (3.37).

$$\begin{aligned}
& [\mathbf{I} - \mathbf{S}_B] [\mathbf{D}]^{-1} [\mathbf{J}_V] \Delta \mathbf{x}^{k+1} - [\mathbf{I} + \mathbf{S}_B] (-\mathbf{F}_i - [\mathbf{D}] [\mathbf{J}_I] \Delta \mathbf{x}^{k+1}) + [\mathbf{I} - \mathbf{S}_B] \mathbf{F}_v = 0 \\
\Rightarrow & \quad [(\mathbf{I} - \mathbf{S}_B) \mathbf{D}^{-1} \mathbf{J}_V + (\mathbf{I} + \mathbf{S}_B) \mathbf{D} \mathbf{J}_I] \Delta \mathbf{x}^{k+1} = -[\mathbf{I} + \mathbf{S}_B] \mathbf{F}_i - [\mathbf{I} - \mathbf{S}_B] \mathbf{F}_v \\
\Rightarrow & \quad \Delta \mathbf{x}^{k+1} = [(\mathbf{I} - \mathbf{S}_B) \mathbf{D}^{-1} \mathbf{J}_V + (\mathbf{I} + \mathbf{S}_B) \mathbf{D} \mathbf{J}_I]^{-1} \\
& \quad \quad \quad \{-[\mathbf{I} + \mathbf{S}_B] \mathbf{F}_i - [\mathbf{I} - \mathbf{S}_B] \mathbf{F}_v\} \tag{3.38}
\end{aligned}$$

The updates from each iteration are calculated by Eqs. (3.38), and (3.37). Here, one additional matrix decomposition is required for calculating $\Delta \mathbf{b}^{k+1}$.

3.5 Pseudo-Transient Approach

Pseudo-Transient method was originally used in *Advanced Statistical Analysis Program (ASTAP)* [59], [26]. A transient simulation on a pseudo-circuit is performed instead of the original circuit, the pseudo-circuit may be an inductor connected in series or a capacitor connected in parallel [54]. Iterations for the original circuit converge when the pseudo-transient reaches steady-state. It is proved in Section 3.3.2 that his proposed approach of transient analysis based on power waves is always convergent for circuits containing only locally passive nonlinear elements but circuit elements are not always guaranteed to be passive and thus the convergence is not always assured. Capacitors have the effect of *passivizing* the nonlinear devices, because they appear as conductances after time discretization. Capacitors are inserted in each port of nonlinear devices as pseudo-circuit to exploit this property. Convergence improvement with accuracy is achieved up to a certain level over the plain relaxation method but unfortunately convergence assurance is not obtained.

3.5.1 Timing Simulation of Pseudo-Circuit

In pseudo-transient approach, timing simulation (described in Subsection 2.6.1) is employed to perform the transient analysis of the pseudo-circuit. One relaxation iteration is performed per pseudo-time step, this process is repeated until steady-state voltage across the pseudo-circuit is achieved. As this transient analysis upon the pseudo-circuit *i.e.*, the transition from the zero solution to the final solution of voltage across this circuit is of no interest in this actual analysis, the truncation error can be ignored as long as the solution converges to the correct equilibrium

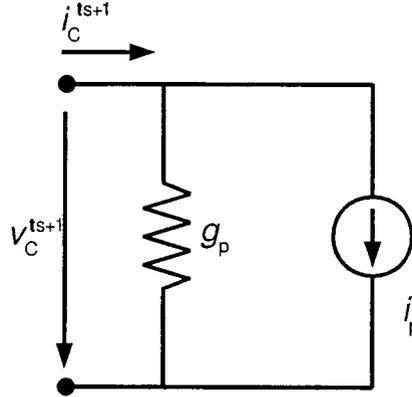


Figure 3.5: Discretized model of a linear capacitor using backward Euler formula

Note that, a constant conductance is assumed throughout the pseudo-transient analysis, *i.e.*, both capacitance C_s and time step h_p are assumed constant. As timing simulation is performed until steady state is achieved, the total number of iterations for one actual time step is equal to the total number of iterations (pseudo-time points) in one pseudo timing simulation. Timing simulation trajectory across any arbitrary pseudo-capacitor is shown in Fig. 3.6 (MESFET circuit is used to generate the trajectory). Note that, this trajectory is shown just for the better understanding of the concept of pseudo-transient analysis, it is not a part of actual simulation process.

Modification of System Equations

Newton's method is applied as before to solve the decoupled system of equations. Equations are slightly modified to include the effect of the capacitors. Error function in Eq. (3.21), will be rewritten as:

$$\mathbf{F}_i(\mathbf{x}_{d+1}^t, \mathbf{b}_{d+1}^t) = [\mathbf{D}] (\mathbf{i}_{NL}(\mathbf{x}_{d+1}^t) + [\mathbf{G}_p] \mathbf{v}_{NL}(\mathbf{x}_{d+1}^t) + \mathbf{i}_p) - \mathbf{a}_{d+1}^t + \mathbf{b}_{d+1}^t \quad (3.41)$$

here, d is the iteration number in timing simulation of pseudo-circuit, $[\mathbf{G}_p]$ is the diagonal matrix with the conductances and \mathbf{i}_p is the vector of dependent current sources (i_p) that depends on the

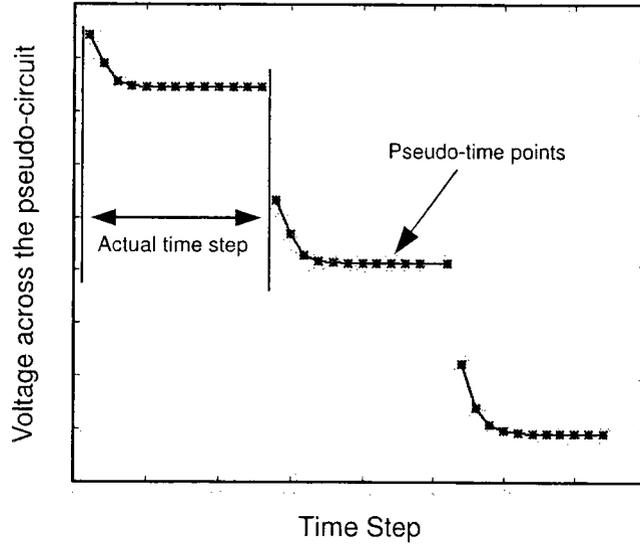


Figure 3.6: Actual time step and pseudo-time points in timing analysis of pseudo-circuit

voltages across the pseudo-circuit at d iteration *i.e.*, for $d + 1$ iteration this vector is treated as constant. The derivative from Eq. (3.26), will be modified as follows:

$$\begin{aligned}
 \frac{\partial \mathbf{F}_i}{\partial \mathbf{x}} &= [\mathbf{D}] \frac{\partial}{\partial \mathbf{x}} (\mathbf{i}_{NL}(\mathbf{x}_{d+1}^t) + [\mathbf{G}_p] \mathbf{v}_{NL}(\mathbf{x}_{d+1}^t) + \mathbf{i}_p) \\
 &= [\mathbf{D}] \frac{\partial \mathbf{i}_{NL}(\mathbf{x}_{d+1}^t)}{\partial \mathbf{x}} + [\mathbf{D} \mathbf{G}_p] \frac{\partial \mathbf{v}_{NL}(\mathbf{x}_{d+1}^t)}{\partial \mathbf{x}} + 0 \\
 &= [\mathbf{D} \mathbf{J}_I] + [\mathbf{D} \mathbf{G}_p \mathbf{J}_V]
 \end{aligned} \tag{3.42}$$

Voltage across the capacitor is updated as follows:

$$\mathbf{v}_C^{d+1} = [\mathbf{D}]([\mathbf{S}] \mathbf{b}_{d+1}^t + \mathbf{a}_0^t + \mathbf{b}_{d+1}^t) \tag{3.43}$$

where, \mathbf{v}_C is the vector of voltages across the capacitors. The vector of current sources is updated as follows,

$$\begin{aligned}
 \mathbf{i}_p^{d+1} &= [\mathbf{G}_p] \mathbf{v}_C^{d+1} \\
 &= [\mathbf{G}_p \mathbf{D}]([\mathbf{S}] \mathbf{b}_{d+1}^t + \mathbf{a}_0^t + \mathbf{b}_{d+1}^t)
 \end{aligned} \tag{3.44}$$

All other formulations will be the same as mentioned in Subsection 3.3.1.

3.5.2 Relaxation Factor

Modified Newton's Method with *relaxation factor* is used for solving nonlinear problems in different fields, such as magnetic field analysis [42] and heat transfer problems [9]. This method is also used to accelerate the convergence characteristics of a nonlinear finite-element in magnetic field analysis [11]. This idea of relaxation factor is used to modify Newton's method for handling strong nonlinearity and to improve the convergence as well. This idea works well with *Pseudo-Transient approach*, so his implementation is limited to this approach only. Modification with relaxation factor of Eq. (3.22). is given by

$$\mathbf{b}_{n+1}^t = (\mathbf{I} - \chi)\mathbf{b}_n^t + \chi\mathbf{F}([\mathbf{S}] \mathbf{b}_n^t + \mathbf{a}_0) \quad (3.45)$$

where χ is the relaxation factor and $0 < \chi < 1$, different names are used for this factor in different fields, the name, *relaxation factor* is used as in the references mentioned in this section. Eq. (3.45). is in principle similar to Eq. (2.45). when $[\mathbf{M}_S^p] = \chi_j[\mathbf{I}]$. One case of convergence improvement with relaxation factor is depicted in Fig. 3.7. Original sequence generated by timing analysis of the voltages across the pseudo-circuit is not convergent, so steady state oscillation is observed. Selection of relaxation factor along with MPE extrapolation of reflected waves make this sequence convergent.

Simplified Pseudo-code with Relaxation Factor

In the present implementation, the iteration is reset when Newton iteration of decoupled equations or the original timing iteration is not convergent. Exploration of a method to find the optimum value of χ may save some simulation time here. A simplified Pseudo-code of the simulator with relaxation factor is presented here,

3.6 Application of MPE in relaxation method

It is shown in Section 2.6 that relaxation methods converge linearly when may converge to solution. Fixed-point iterative scheme of power waves conserves this convergence property too. Eq. (3.30), can be rewritten as

$$\begin{aligned} (\mathbf{b}_{n+1} - \mathbf{b}_{sol}) &= [\mathbf{J}_F\mathbf{S}] (\mathbf{b}_n - \mathbf{b}_{sol}), \\ \implies \mathbf{b}_{n+1} &= [\mathbf{J}_F\mathbf{S}] \mathbf{b}_n + [\mathbf{I} - \mathbf{J}_F\mathbf{S}] \mathbf{b}_{sol} \end{aligned} \quad (3.46)$$

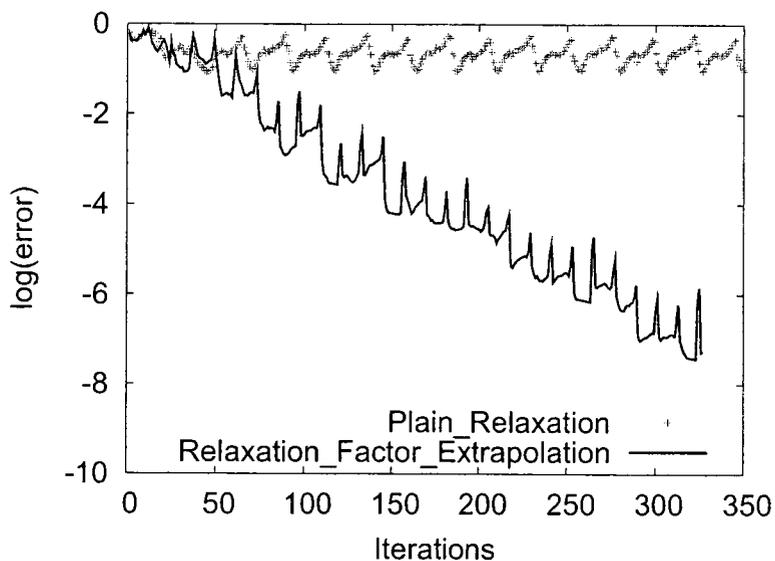


Figure 3.7: Convergence improvement with relaxation factor over plain relaxation

Algorithm 3.5.1 Algorithm for Timing Analysis of Power Waves using Relaxation Factor

repeat

χ is set to 1 or some optimum value

save the results from previous time step and use those as initial guess

repeat

apply Eq. (3.45). for K iterations

apply MPE extrapolation

if (Newton's method or timing analysis is not convergent) **then**

reset the timing analysis and reset all vectors to initial guess

reduce the value of χ

end if

until ($error \leq \epsilon$)

increase time, $t = t + h$

until ($t < t_{end}$)

where \mathbf{b}_{sol} is the solution of the Eqs. (3.22), or (3.32). \mathbf{n} is the iteration number. In Eq. (3.46). iterative sequence \mathbf{b}_n is expressed in the same way as \mathbf{c}_r stated in Eq. (2.64). Here, $[\mathbf{J}_F\mathbf{S}]$ is a fixed matrix and $[\mathbf{I} - \mathbf{J}_F\mathbf{S}] \mathbf{b}_{sol}$ is a vector. To employ MPE, K terms of the sequence generated from Eq. (3.46) are used.

$$\begin{aligned} \mathbf{b}_{m+1} &= [\mathbf{J}_F\mathbf{S}] \mathbf{b}_m + [\mathbf{I} - \mathbf{J}_F\mathbf{S}] \mathbf{b}_{sol}, \\ &\vdots \\ \mathbf{b}_{m+K} &= [\mathbf{J}_F\mathbf{S}] \mathbf{b}_{m+K-1} + [\mathbf{I} - \mathbf{J}_F\mathbf{S}] \mathbf{b}_{sol} \end{aligned} \quad (3.47)$$

And the difference vectors are defined as follows:

$$\begin{aligned} \mathbf{w}_{br} &= \Delta \mathbf{b}_r \\ &= \mathbf{b}_{m+r+1} - \mathbf{b}_{m+r} \end{aligned}$$

If the eigenvalue of $[\mathbf{J}_F\mathbf{S}]$, $\lambda \neq 1$ and $\|[\mathbf{J}_F\mathbf{S}]\| < 1$ then iterative sequence stated by Eq. (3.47), has a unique fixed point,

$$\begin{aligned} \mathbf{b}_P &= [\mathbf{J}_F\mathbf{S}] \mathbf{b}_P + [\mathbf{I} - \mathbf{J}_F\mathbf{S}] \mathbf{b}_{sol}, \\ \implies \mathbf{b}_P &= [\mathbf{I} - \mathbf{J}_F\mathbf{S}]^{-1} [\mathbf{I} - \mathbf{J}_F\mathbf{S}] \mathbf{b}_{sol} \end{aligned} \quad (3.48)$$

where \mathbf{b}_P is the fixed point of the sequence. Note that, number of vectors to extrapolate (K) is set as simulation parameter in his proposed approach. And the vector $\mathbf{q}_b = [q_{b0}, q_{b1}, \dots, q_{bK-1}]^T$ of unknown coefficients of polynomial of $[\mathbf{J}_F\mathbf{S}]$ is a solution of the system of equations

$$[\mathbf{W}_b] \mathbf{q}_b = -\mathbf{w}_{bK} \quad (3.49)$$

where $[\mathbf{W}_b]$ is the difference matrix of size $n_{nl} \times K$ (n_{nl} is the number of nonlinear device ports), \mathbf{w}_{bK} is the K^{th} difference vector which is of size n_{nl} . Now multiplying transpose $[\mathbf{W}_b]^T$ to both side of the Eq. (3.49), it is found,

$$\begin{aligned} [\mathbf{W}_b]^T [\mathbf{W}_b] \mathbf{q}_b &= -[\mathbf{W}_b]^T \mathbf{w}_{bK} \\ \implies [\mathbf{W}_b^{sq}] \mathbf{q}_b &= -\mathbf{w}_{bK}^{sq} \end{aligned} \quad (3.50)$$

where $[\mathbf{W}_b^{sq}]$ is square matrix of size $K \times K$ and \mathbf{w}_{bK}^{sq} is a vector of size K . So the solution of Eq. (3.50). is given by

$$\mathbf{q}_b = -[\mathbf{W}_b^{sq}]^{-1} \mathbf{w}_{bK}^{sq} \quad (3.51)$$

The inverse does not exist if $K > n_{nl}$, but in the presented work K is always smaller than n_{nl} . The fixed-point can be calculated the same way as mentioned in Eq. (2.74).

$$\mathbf{b}_P = \frac{\sum_{r=m+1}^{m+K} q_{b_r} \mathbf{b}_r}{\sum_{r=m+1}^{m+K} q_{b_r}} \quad (3.52)$$

The computational cost of MPE is roughly equivalent to the cost of decomposing a square matrix of size K . For medium and large circuits K is smaller than the number of nonlinear ports. MPE converges quadratically if the vector sequence is close to the solution. Thus the combination of fixed-point plus extrapolation could achieve in principle a performance similar to Newton's method.

3.6.1 Simplified Pseudo-code

A simplified Pseudo-code of the simulator is presented in Algs. (3.6.1), the sequences are similar for both *fixed-point iteration with Newton's method* and *fixed-point iteration with Newton-Jacobi method*.

Algorithm 3.6.1 Algorithm of Fixed-point Iterative Scheme of Power Waves

```

repeat
  initial guess is set to the results in previous step
  repeat
    apply Eq. (3.22), or Eq. (3.32), for  $K$  iterations
    apply MPE extrapolation
  until ( $error \leq \epsilon$ )
  increase time,  $t = t + h$ 
until ( $t < t_{end}$ )

```

3.7 Reflected Power Considerations

Pure relaxation approach stated by Eq. (3.22), is guaranteed not to diverge to infinity, even if the initial guess is far from the solution. This property is a direct consequence of using power waves and is important because it ensures that device models always have a physically meaningful excitation and thus numerical problems can be avoided. The total reflected power at

each iteration is given by $|\mathbf{b}_{n+1}^t|^2$, where the bars denote the Euclidean Norm, and is bounded by

$$|\mathbf{b}_{n+1}^t|^2 < P_{max} + L_{Sc} |\mathbf{S} \mathbf{b}_n^t + \mathbf{a}_0|^2 \quad (3.53)$$

where L_{Sc} is a scalar and $0 < L_{Sc} < 1$ for locally passive nonlinear devices and P_{max} is the maximum power that can be transmitted from the nonlinear devices to the network during one time step and is given by

$$P_{max} = \frac{E_A}{h} \quad (3.54)$$

where E_A is the total energy stored in nonlinear capacitors and inductors and h is the time step. If it is assumed that the upper bound is propagated from the first iteration, the result obtained (originally shown in [12]) is as follows:

$$\lim_{n \rightarrow \infty} |\mathbf{b}_{n+1}^t|^2 < \frac{1}{1 - L_{Sc}} (P_{max} + L_{Sc} |\mathbf{a}_0|^2) \quad (3.55)$$

Chapter 4

Simulation Results and Discussions

Transient simulations of several circuits performed using the different methods proposed in this work are presented in this chapter. Nonlinear device models are modelled accurately including nonlinear capacitors. This is the first time a wave-based transient analysis is used for circuits of this complexity. Results from the standard state-variable transient analysis in fREEDA are assumed to be correct, as this analysis has been previously verified against measurements and results from other circuit simulators [51, 10] on most of the circuits considered in this chapter. All simulations use fixed time step and the same absolute tolerance equal to 10^{-8} . Reference port impedance is considered as constant throughout the simulation.

All the analysis methods in fREEDA are saved in $\sim \backslash freeda - 1.x \backslash simulator \backslash analysis$ directory. The regular transient analysis in fREEDA is named as *tran2* (source file is SV-Tran.cc). The Transient analysis methods, developed for this research are *wavetran* (source file is WaveTran.cc), *wavetran2* (source file is WaveTran2.cc) and *wavetranpseudo* (source file is WaveTranPseudo.cc). *wavetran* is the transient analysis based on relaxation of power waves and Newton's method, *wavetran2* is the transient analysis based on relaxation of power waves and Newton-Jacobi method and *wavetranpseudo* is the transient analysis based on timing analysis of power waves across pseudo-circuit. All the codes are written by the explicit use of Matrix Template Library (MTL). Some of the common simulation parameters are

1. "tstop" – Stop time (s),
2. "tstep" – Time step (s),

3. “im” – Integration method,
4. “out_steps” – Number of steps skipped for output simulation progress,
5. “zref” – Reference nonlinear port impedance (Ω),
6. “tol” – Tolerance for nonlinear iterations.

In this Chapter, *Wave_Transient* is used to denote *wavetran*, *Wave_Transient2* represents *wavetran2* and *Pseudo_Transient* represents *wavetranpseudo*. Finally *Regular_Transient* is used to denote *tran2*.

4.1 Single MESFET Amplifier

The single MESFET amplifier circuit used in the simulations, is shown in Fig. 4.1. The MESFET is modelled using the Curtice-Ettemberg cubic model with symmetric diodes and capacitances [13]. Element values are as follows: $R_{cr12} = R_{cr1L} = 50 \Omega$, $L_{cr12} = 1 \text{ nH}$, $C_{cr12} = 200 \text{ pF}$, $R_{cr11} = 100 \Omega$, $L_{cr11} = L_{cr13} = 15 \text{ nH}$, $R_{cr13} = 10 \Omega$, $R_{cr14} = 1.144 \Omega$ and $C_{cr1L} = 20 \text{ pF}$. The source values are $V_{cr1BIAS} = -0.4 \text{ V}$, $V_{cr1dd} = 3 \text{ V}$. Amplitude and frequency of V_{cr1S} are 1 V and 5.1 GHz respectively and $zref = 200 \Omega$.

4.1.1 Simulation Results and Outputs

Simulation results for single MESFET amplifier is shown in Table 4.1, all the results presented here are in average per time step except *Newton Iteration*, this is the average value per relaxation iteration and the term *plain* refer to the methods without MPE. This convention will be followed for other circuits also. The methods are slower than the regular transient analysis in FREEDA as expected, because the total number of equations and unknowns to be solved is twice than that of in regular transient and relaxation method is slower than Newton’s method. Though the proposed approaches are slower for the MESFET amplifier and the next three circuits, the results are presented here mainly to demonstrate the correctness of the methods. For all the methods on average one Newton iteration is required, so overhead for iterating waves is not expensive. Extrapolation improves the performance significantly for *Wave_Transient* and *Pseudo_Transient*. *Wave_Transient2* converges to the solution as soon as the Newton-Jacobi iteration converges when there is only one nonlinear device (the whole scattering matrix in considered inside nonlinear iteration loop). But sometimes it takes more than 2 iterations to

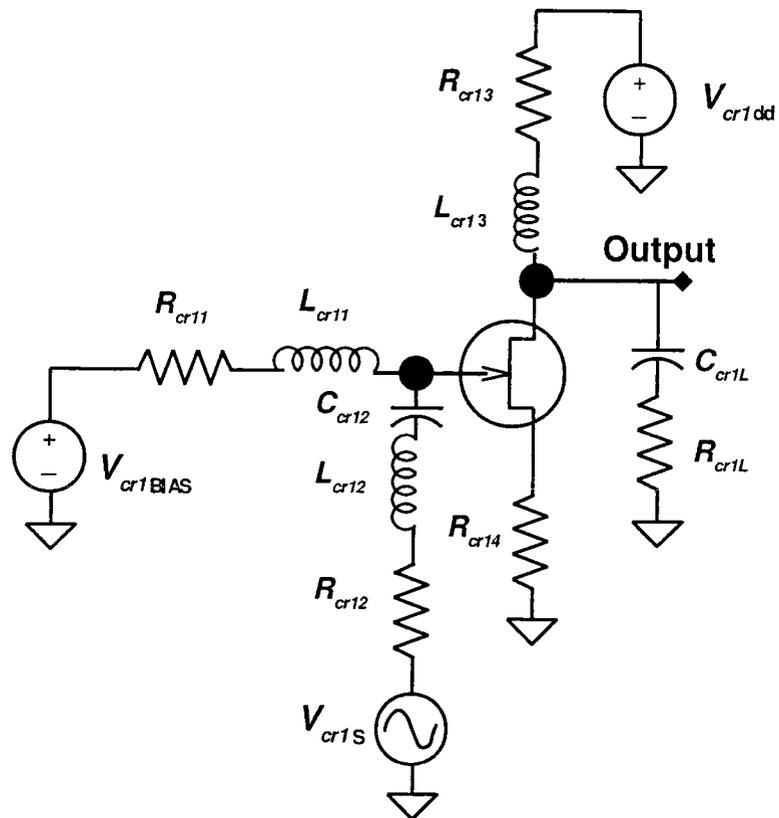


Figure 4.1: Single MESFET amplifier

Table 4.1: Summary of Simulation Results for Single MESFET amplifier

Methods	Wave_Transient	Wave_Transient2	Pseudo_Transient
Nonlinear Ports	3		
Time steps	200		
Iterations (plain)	29	4	22.25
Iterations (MPE)	12	4	12.29
K	4		
Avg. MPE	1.74627	0.0895522	1.66667
Newton Iteration (MPE)	1.15964	1.29461	1.19732
Simulation Time (plain)	0.67 s	0.26 s	0.69 s
Simulation Time (MPE)	0.36 s	0.25 s	0.43 s
Simulation Time (Regular_Transient)	0.07 s		

converge because of the larger value of tolerance of nonlinear iteration loop than the actual relaxation loop (as just the approximation from the nonlinear iteration is needed). For MESFET circuit, Wave_Transient2 is faster than other two approaches.

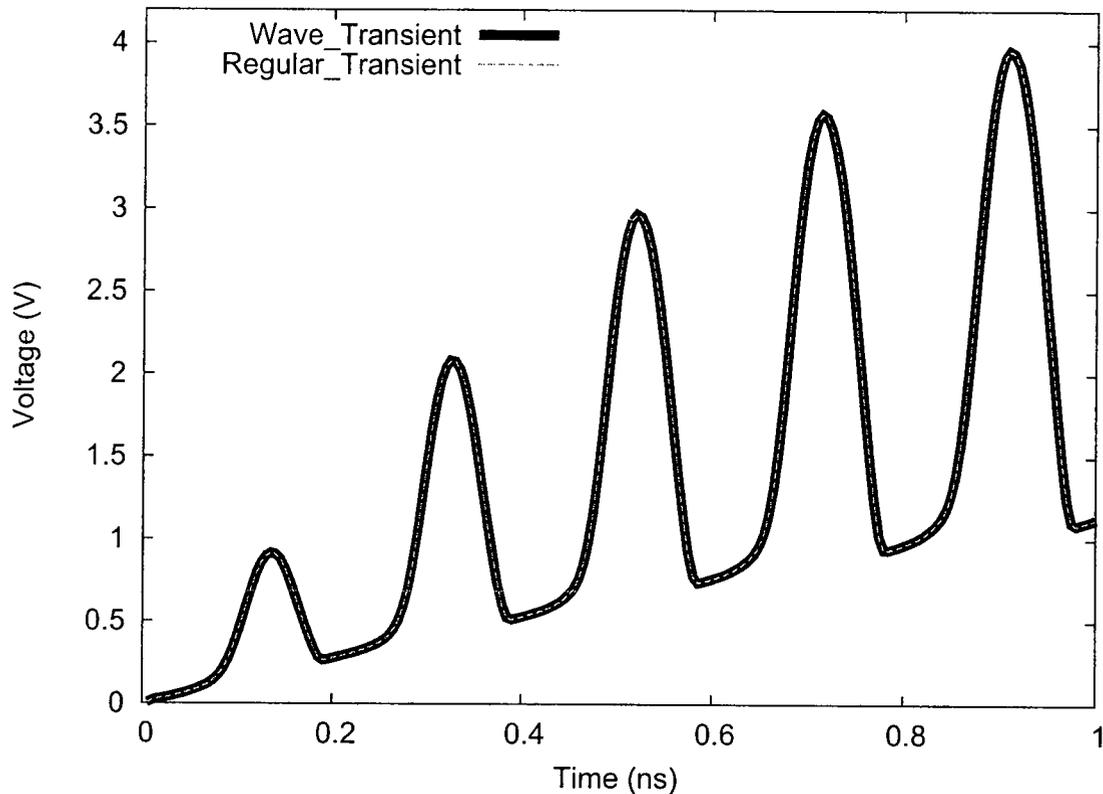


Figure 4.2: Outputs from Wave_Transient method single MESFET amplifier

Outputs for MESFET amplifier from different methods are shown in Fig. 4.2, 4.3 and 4.4 respectively. All outputs from his proposed methods coincide with the output from regular transient, so his methods converge to correct solution for single MESFET circuit.

4.1.2 Convergence Rate Analysis

Convergence improvements with extrapolation are shown in Fig. 4.5, 4.6 and 4.7 respectively. All the convergence plots are for some representative time step. Convergence rate is more or less linear for Wave_Transient and Pseudo_Transient methods without extrapolation, which turns into *superlinear* (*i.e.*, faster than linear but slower than quadratic) with the aid of MPE extrapolation. Wave_Transient2 has better convergence rate than the other two and there is no considerable

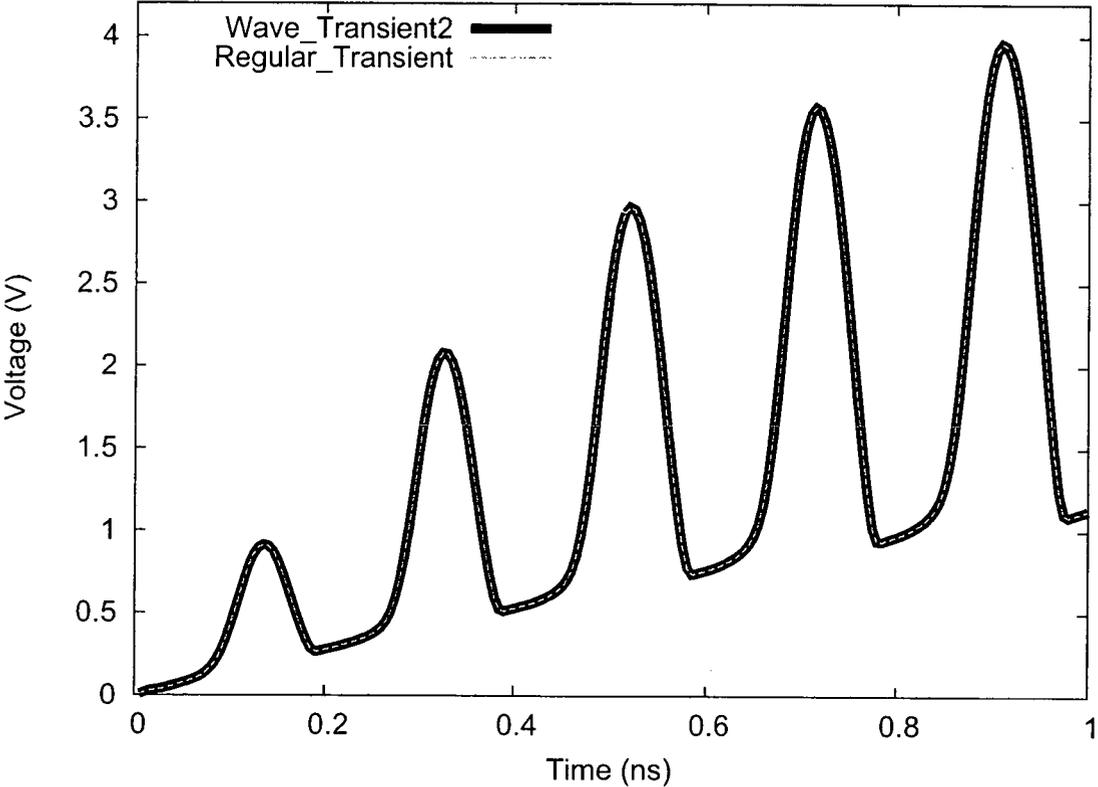


Figure 4.3: Outputs from Wave_Transient2 method for single MESFET amplifier

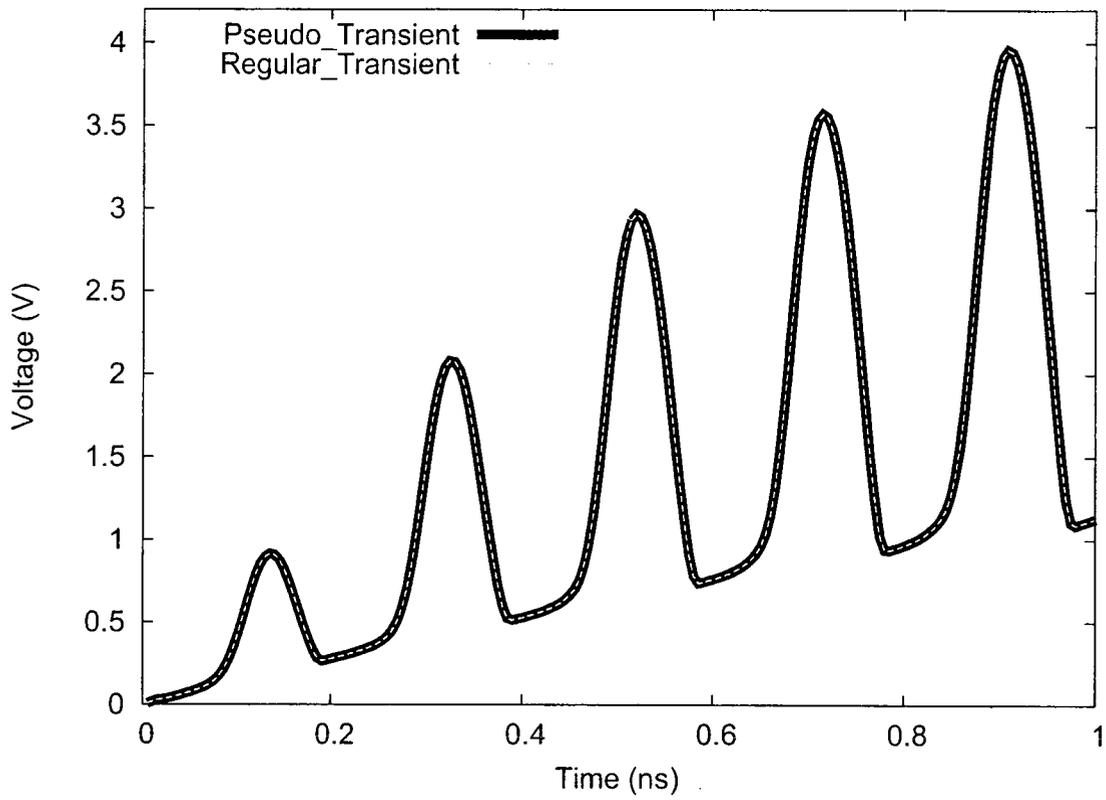


Figure 4.4: Outputs from Pseudo_Transient method for single MESFET amplifier

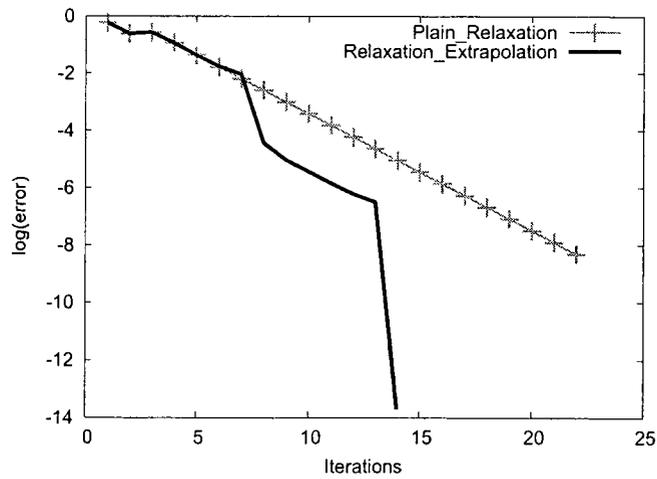


Figure 4.5: Convergence comparison from Wave_Transient method for single MESFET amplifier

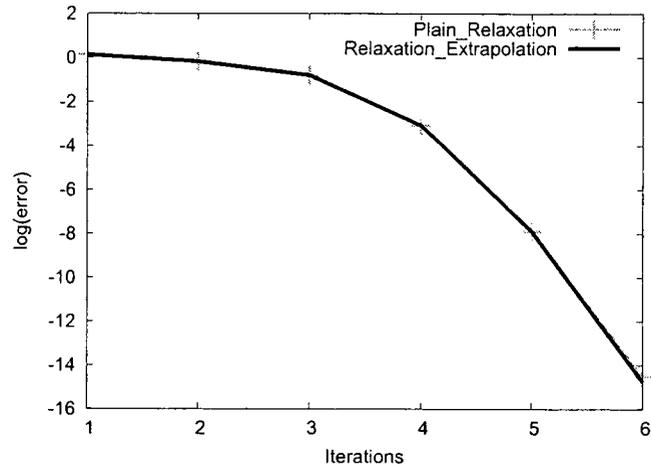


Figure 4.6: Convergence comparison from Wave_Transient2 method for single MESFET amplifier

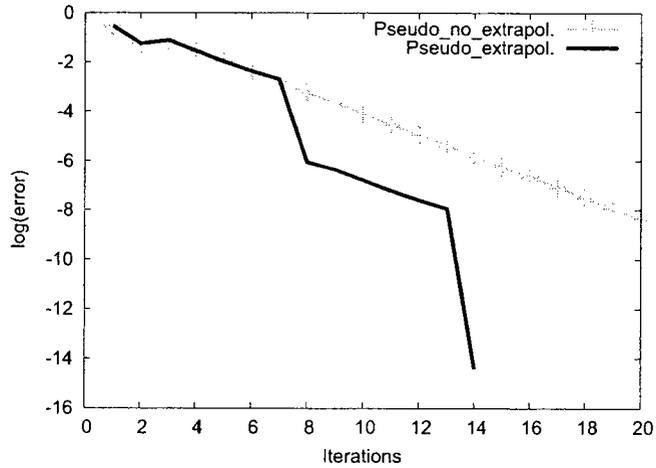


Figure 4.7: Convergence comparison from Pseudo_Transient method for single MESFET amplifier

improvement with MPE on that analysis.

4.2 X-band MMIC LNA

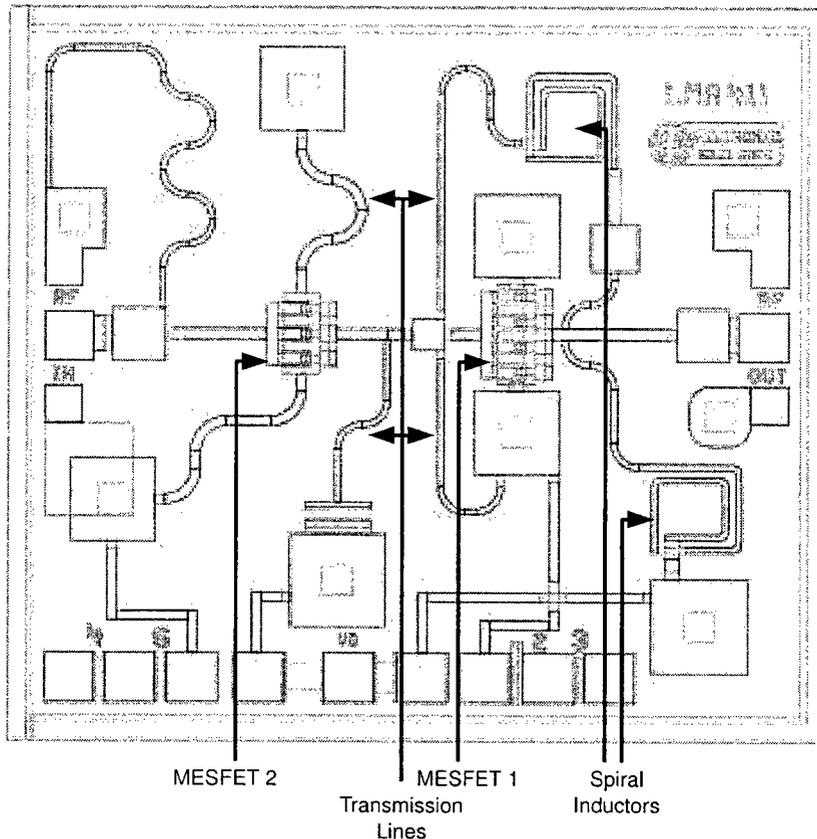


Figure 4.8: X-band MMIC LNA (LMA 411)

Analog circuits called *Monolithic Microwave ICs* (MMICs) are usually used to downconvert a modulated microwave signal to a baseband signal or upconvert the baseband signal to a microwave signal. Several sub-circuit types, such as low noise amplifiers, mixers, power amplifiers, oscillators, and switches belong to MMIC family [35]. Fig. 4.8 is an X-band MMIC low noise amplifier (LNA) (LMA411). This is a low noise pHEMT LNA using $0.25 \mu m$ technology. First amplifier (MESFET 1) is $300 \mu m$ ($6 \times 50 \mu m$) wide and second is $400 \mu m$ wide, GaAs substrate thickness is $100 \mu m$ with silicon nitride passivation and DC supply required is 6 volts. Reference port impedance, $z_{ref} = 50 \Omega$.

Table 4.2: Summary of Simulation Results for X-band MMIC LNA

Methods	Wave_Transient	Wave_Transient2	Pseudo_Transient
Nonlinear Ports	4		
Time steps	1000		
Iterations (plain)	15	42	16.63
Iterations (MPE)	7	7	11.51
K	4		
Avg. MPE	0.999001	0.999001	1.54845
Newton Iteration (MPE)	0.999875	0.999875	0.999865
Simulation Time (plain)	4.24 s	16.51 s	5.56 s
Simulation Time (MPE)	3.08 s	4.12 s	4.47 s
Simulation Time (Regular_Transient)	1.68 s		

4.2.1 Simulation Results and Outputs

Simulation results for X-band MMIC LNA is shown in Table 4.2. The methods are slower than the regular transient analysis in FREEDA for same reason explained for previous circuit. On average almost one Newton iteration is required for X-band MMIC LNA. Tolerance for the Newton iteration is greater than the actual tolerance of relaxation loop because an approximation of the solution is needed, not exact solution. When the iterations are close to the solution, sometimes the error for Newton iteration is greater than the tolerance itself, so iterations do not enter the Newton iteration loop. That is why, the average Newton iteration is slightly less than 1. Extrapolation improves the performance significantly for all the methods especially for Wave_Transient2. Wave_Transient2 without MPE is much slower than other two methods because of a dominant element in $[S_O]$ which slows down the overall relaxation scheme, but in Wave_Transient and Pseudo_Transient $[S]$ is taken into account as a whole.

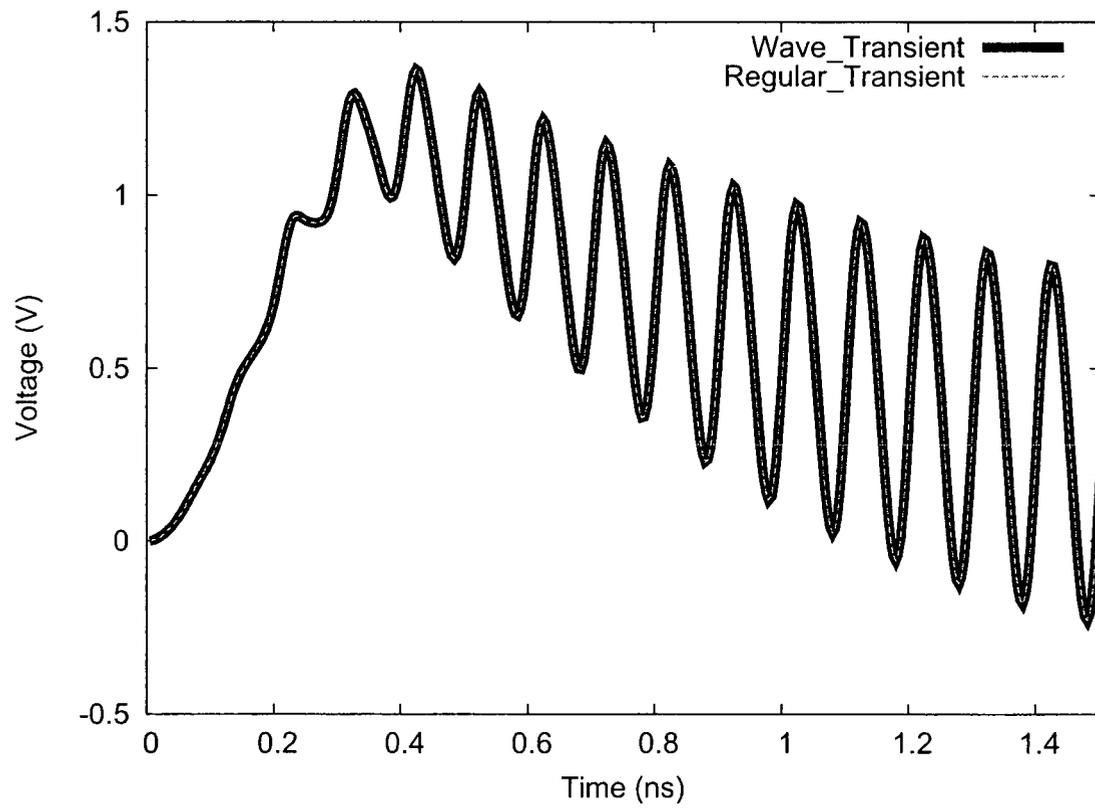


Figure 4.9: Outputs from Wave_Transient method for X-band MMIC LNA

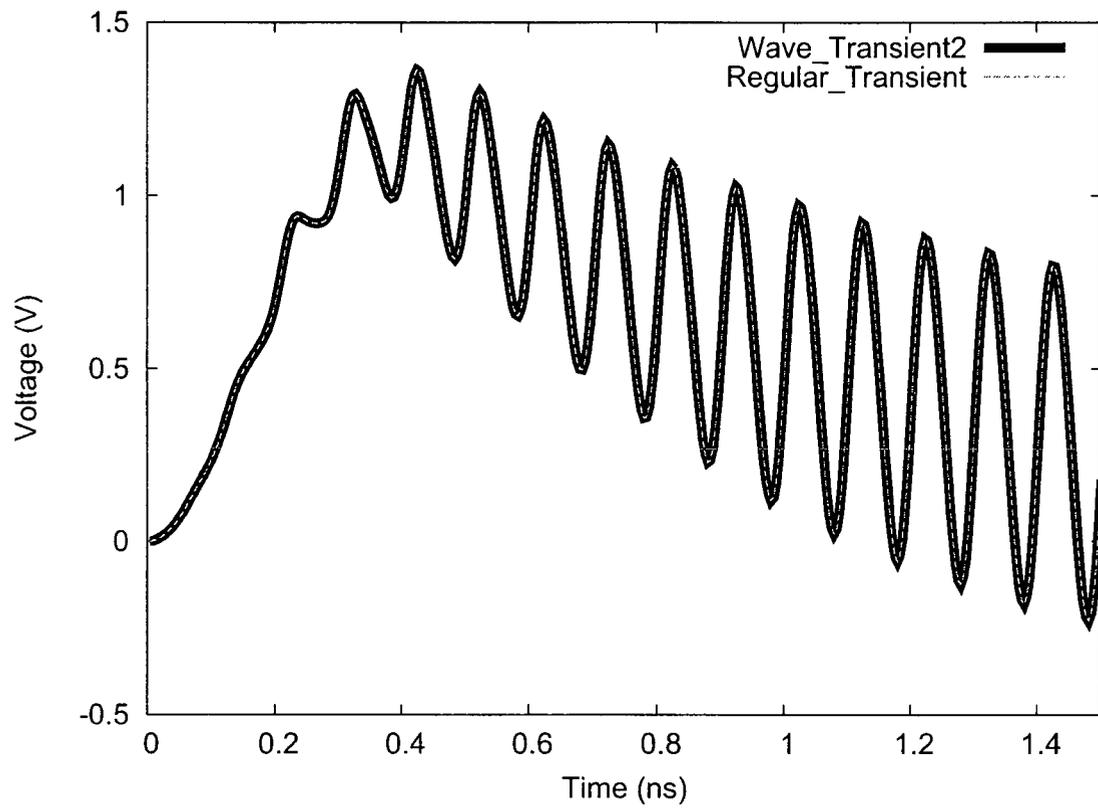


Figure 4.10: Outputs from Wave_Transient2 method for X-band MMIC LNA

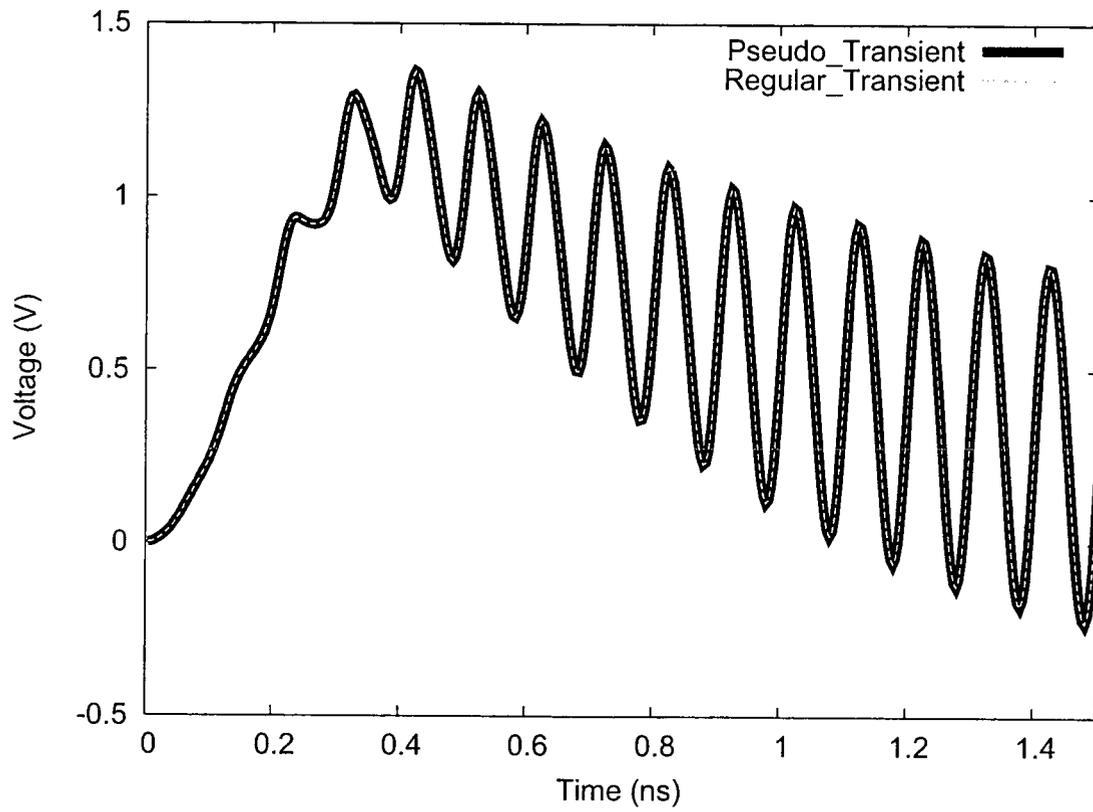


Figure 4.11: Outputs from Pseudo_Transient method for X-band MMIC LNA

Outputs for X-band MMIC LNA from different methods are shown in Fig. 4.9, 4.10 and 4.11 respectively. As shown all the methods converge to correct solution for this circuit.

4.2.2 Convergence Rate Analysis

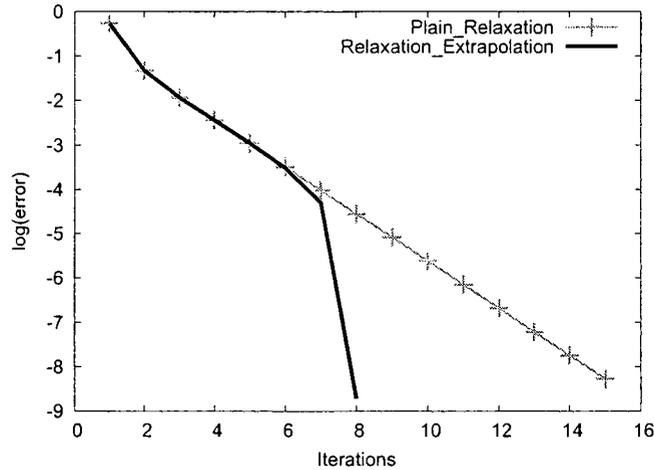


Figure 4.12: Convergence comparison from Wave_Transient method for X-band MMIC LNA

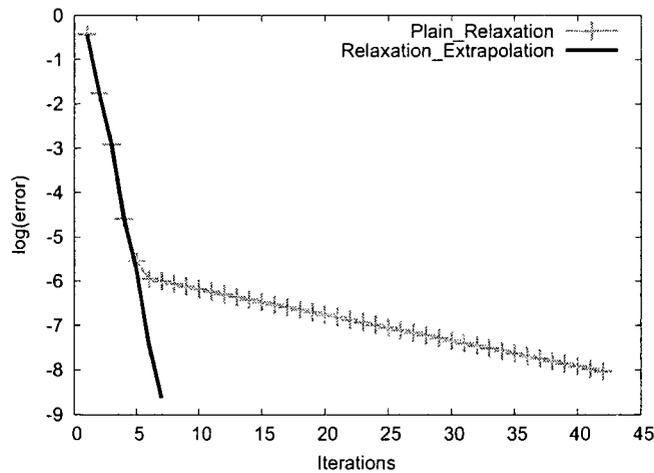


Figure 4.13: Convergence comparison from Wave_Transient2 method for X-band MMIC LNA

Convergence improvements are shown in Fig. 4.12, 4.13 and 4.14 respectively. Like the previous circuit, all the convergence plots are for some representative time step. Convergence rate is more or less slow for all the methods without extrapolation, which turns into a faster one

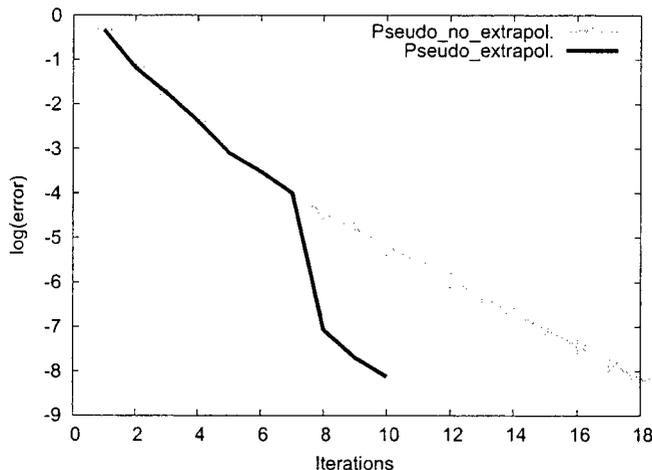


Figure 4.14: Convergence comparison from Pseudo-Transient method for X-band MMIC LNA

with the aid of MPE extrapolation. Convergence rate for Wave-Transient2 without MPE is faster for some initial iterations but slows down after that which gains the same rate of convergence with the help of MPE extrapolation.

4.3 Colpitts Oscillator

A Colpitts oscillator is one of a number of designs for electronic oscillator circuits using the combination of an inductor (L) with a capacitor (C) for frequency determination. This type of oscillator is simple in design and robust, as only a single inductor is needed. The Colpitts oscillator used here, is originally taken from [47] which used a capacitive voltage divider in the LC tank circuit. The oscillator circuit is shown in Fig. 4.15 and the element values are the same as that used in [34]: $C_{cr21} = C_{cr22} = 2 \text{ pF}$, $C_{cr2C} = 400 \text{ pF}$, $C_{cr2E} = 100 \text{ pF}$, $L_{cr21} = 1 \text{ } \mu\text{H}$, $R_{cr21} = 8 \text{ k}\Omega$, $R_{cr22} = 2 \text{ k}\Omega$, $R_{cr2C} = 2.4 \text{ k}\Omega$, $R_{cr2E} = 1.3 \text{ k}\Omega$, $V_{cr2CC} = 11 \text{ V}$, $BF_{cr2} = 100$, $BR_{cr2} = 1$ and $zref = 50 \text{ } \Omega$.

4.3.1 Simulation Results and Outputs

Simulation results for Colpitts Oscillator is shown in Table 4.3. The methods are slower than the regular transient analysis for the same reason explained for previous circuits. On average almost one Newton iteration is required for Colpitts Oscillator and it's slightly less than 1 due to the same reason explained for previous circuit. Extrapolation improves the performance significantly for Wave-Transient and Pseudo-Transient. Wave-Transient2 is much faster than

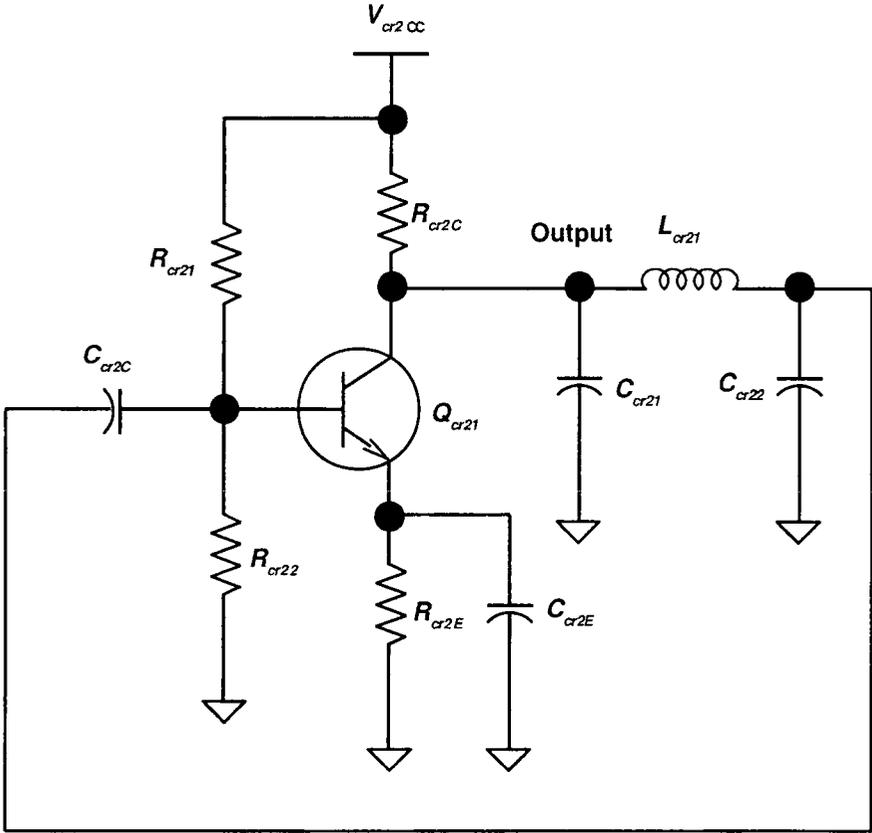


Figure 4.15: Colpitts Oscillator

Table 4.3: Summary of Simulation Results for Colpitts Oscillator

Methods	Wave_Transient	Wave_Transient2	Pseudo_Transient
Nonlinear Ports	3		
Time steps	7500		
Iterations (plain)	45	2	34.38
Iterations (MPE)	10	2	9.27
K	4	2	4
Avg. MPE	1.34262	0.020264	1.21504
Newton Iteration (MPE)	0.999867	0.999867	0.999867
Simulation Time (plain)	67.84 s	8.1 s	64.96 s
Simulation Time (MPE)	16.38 s	8.07 s	18.36 s
Simulation Time (Regular_Transient)	1.88 s		

other two methods for Colpitts Oscillator due to the same reason mentioned for MESFET circuit and MPE does not improve the performance as the iterations per time step is only 2.

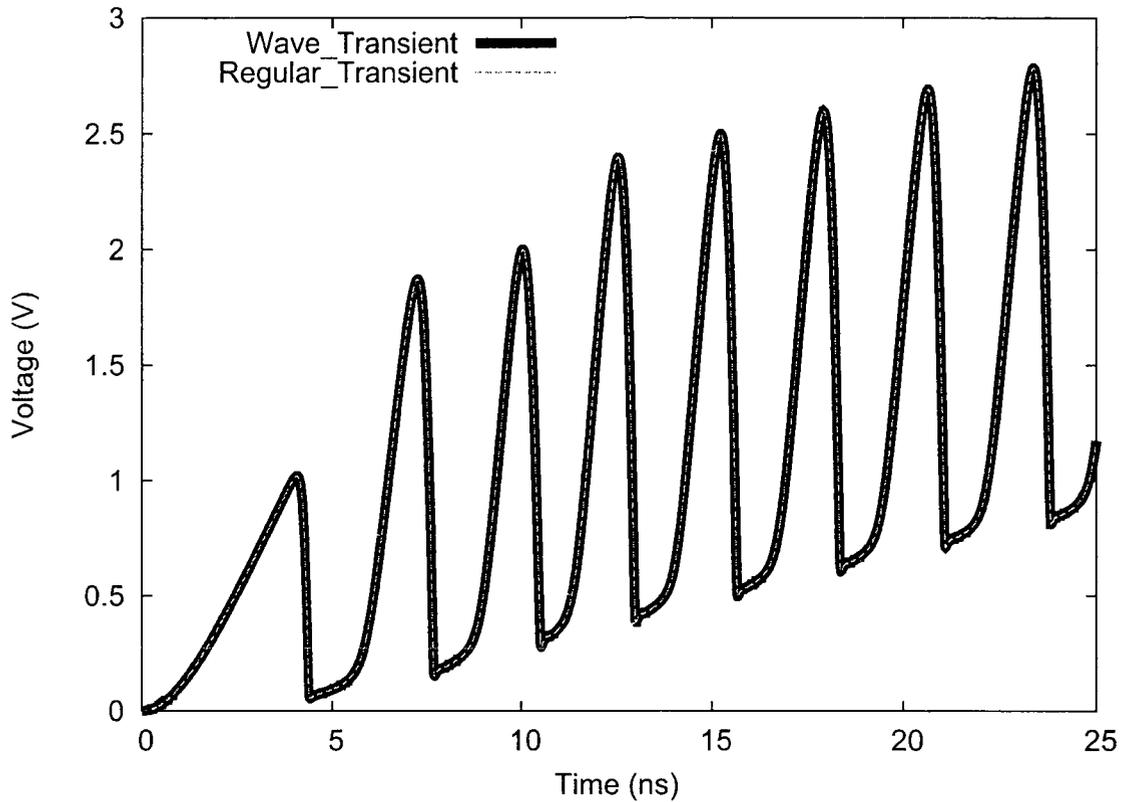


Figure 4.16: Outputs from Wave_Transient method for Colpitts Oscillator

Outputs for Colpitts Oscillator from different methods are shown in Fig. 4.16, 4.17 and 4.18 respectively. As shown all the methods converge to correct solution.

4.3.2 Convergence Rate Analysis

Convergence improvements are shown in Fig. 4.19, 4.20 and 4.21 respectively. Similar to previous circuits, all the convergence plots are for some representative time step. MPE does not influence the convergence rate for Wave_Transient2 at all. For other two methods rate of convergence is slower without extrapolation, which turns into a faster one with the aid of MPE extrapolation.

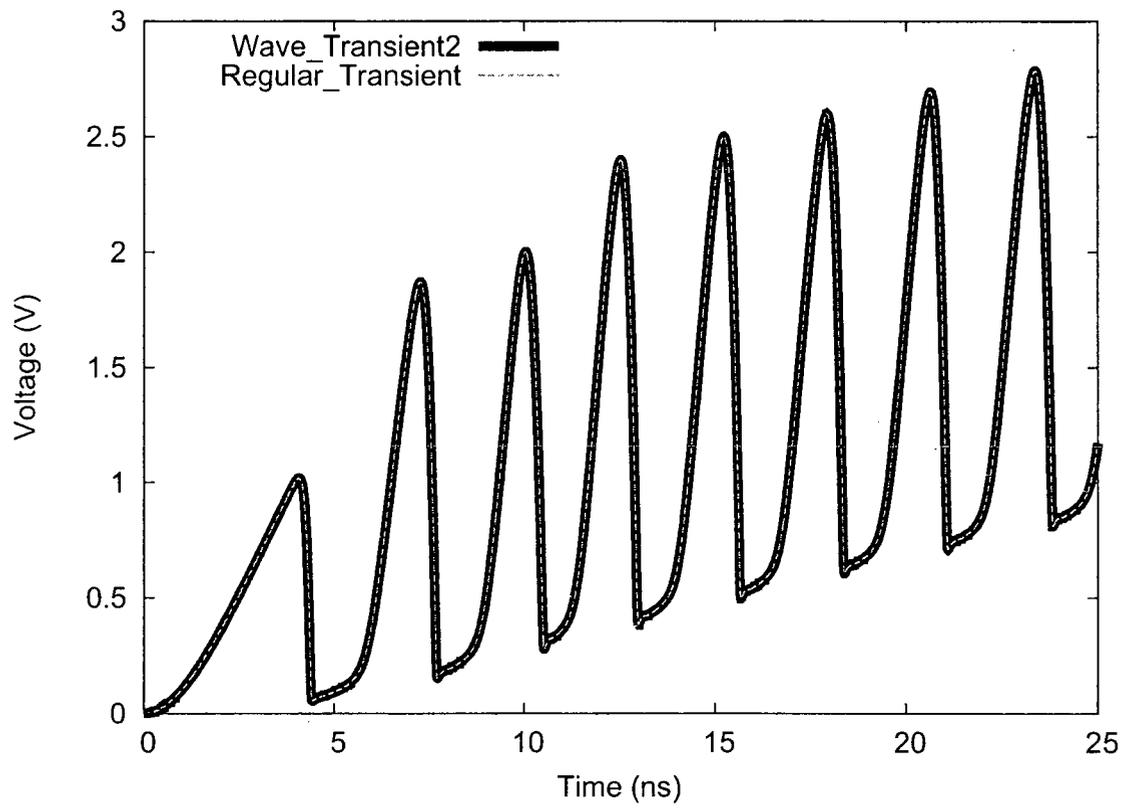


Figure 4.17: Outputs from Wave_Transient2 method for Colpitts Oscillator

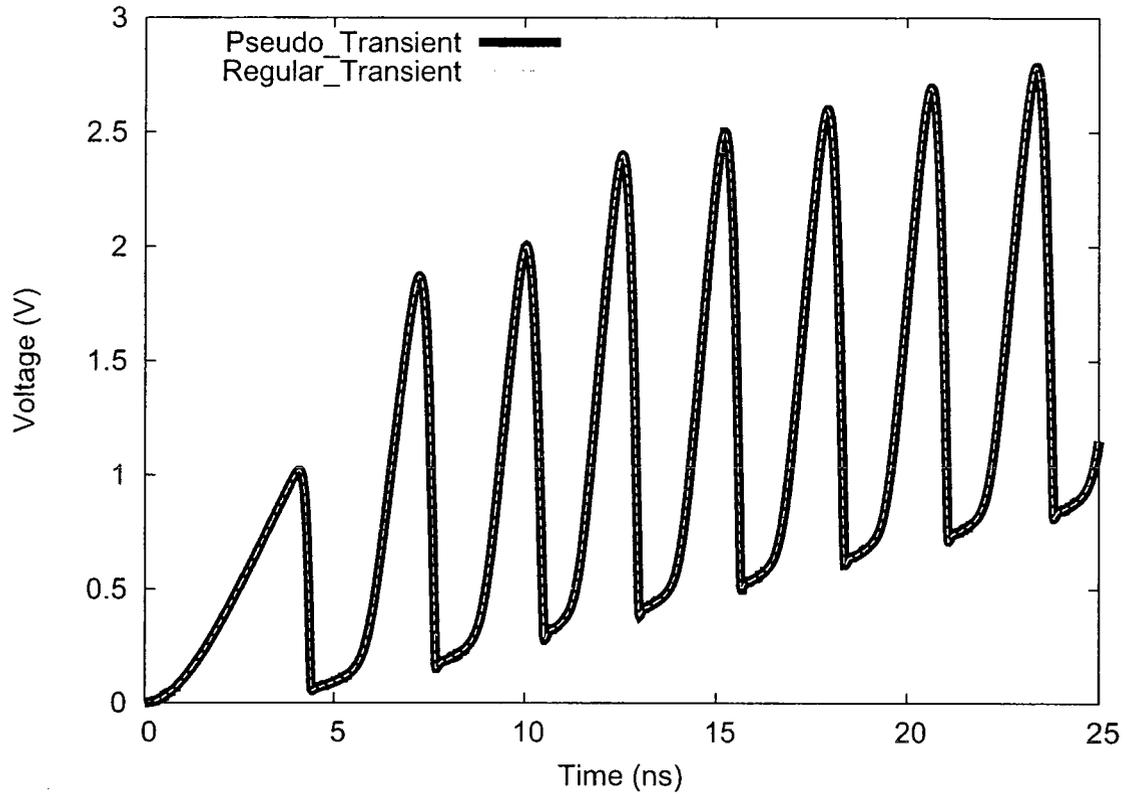


Figure 4.18: Outputs from Pseudo_Transient method for Colpitts Oscillator

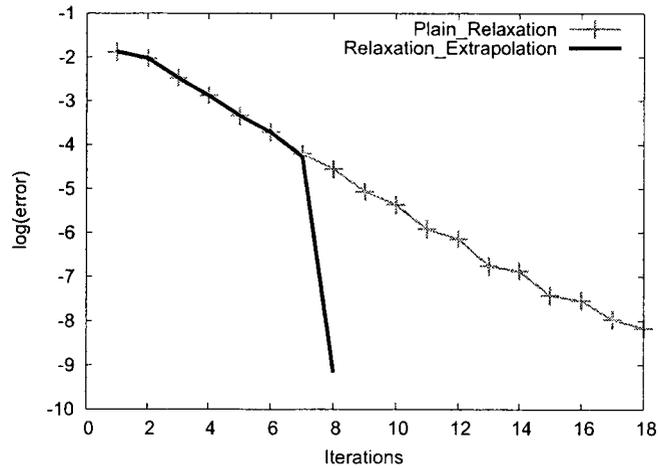


Figure 4.19: Convergence comparison from Wave_Transient method for Colpitts Oscillator

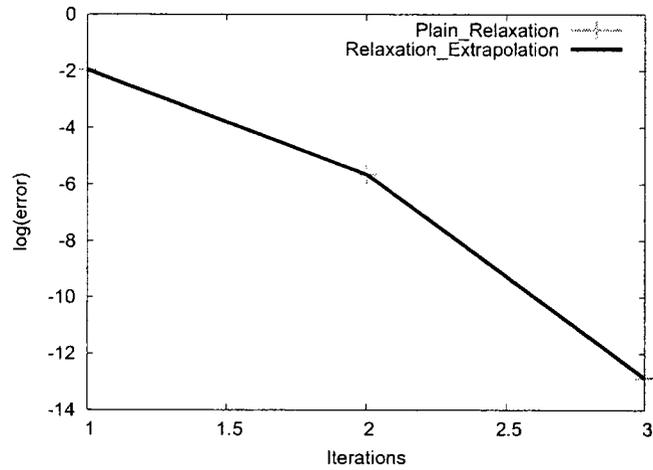


Figure 4.20: Convergence comparison from Wave_Transient2 method for Colpitts Oscillator

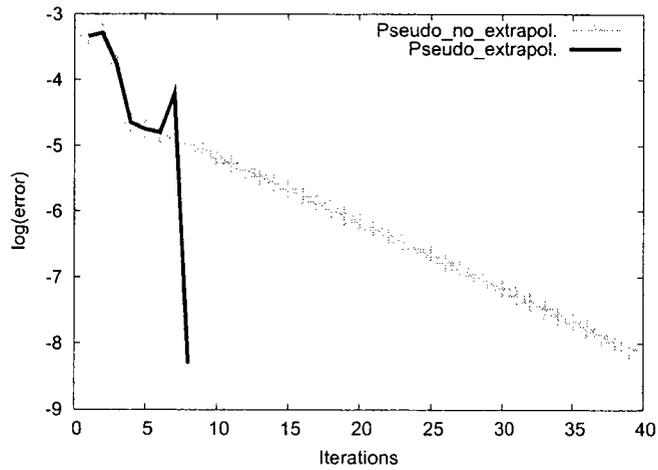


Figure 4.21: Convergence comparison from Pseudo_Transient method for Colpitts Oscillator

4.4 Soliton Line, a Nonlinear Transmission Line

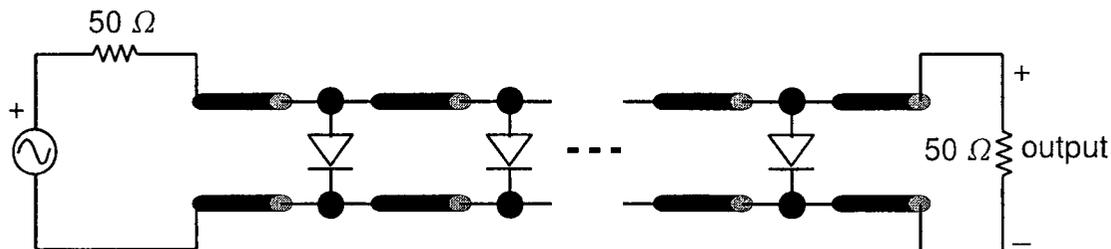


Figure 4.22: Model of the Soliton Line

A nonlinear transmission line is considered in many works as an extreme test of the performance of transient and steady-state simulators [8]. Nonlinear transmission lines (NLTLs) are used in a variety of high speed, wide bandwidth systems including picoseconds resolution sampling circuits, laser and switching diode drivers, test waveform generators, and mm-wave sources [37]. Nonlinearity, dispersion and dissipation are the three fundamental characteristics of NLTLs. The NLTL, presented here, consists of coplanar waveguides (CPWs) periodically loaded with reverse biased Schottky diodes. Diode-based NLTLs (used for pulse generation) are extremely nonlinear which is used to test the robustness of circuit simulators. The NLTL regarded here is 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 [40], [23]. So the NLTL is entitled as *Soliton Line* in this work.

The Soliton Line is modelled using generic transmission lines with frequency-dependent loss and Schottky diodes [13]. Skin effect is also taken into account in the modelling of the transmission lines. The Soliton Line model is shown in Fig. 4.22 and is excited by a 9 GHz sinusoid. The Soliton Line is designed for 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. Reference port impedance, $z_{ref} = 440 \Omega$

Table 4.4: Summary of Simulation Results for Soliton Line

Methods	Wave_Transient	Wave_Transient2	Pseudo_Transient
Nonlinear Ports	47		
Time steps	5000		
Iterations (plain)	6	3	6.59
Iterations (MPE)	5	3	6.13
K	2		
Avg. MPE	0.993601	0.779044	0.994401
Newton Iteration (MPE)	1.08818	1.13317	1.10208
Simulation Time (plain)	146.98 s	144.82 s	171.89 s
Simulation Time (MPE)	137.72 s	147.37 s	166.56 s
Simulation Time (Regular_Transient)	79.84 s		

4.4.1 Simulation Results and Outputs

Simulation results for Soliton Line is shown in Table 4.4. Difference between the simulation time for the proposed methods and the standard method is decreasing with the increase of the number of nonlinear elements. Extrapolation does not effect the simulation time considerably, even sometimes it has negative impact also. For Wave_Transient2 the cost of extrapolation is worse than it's actual improvement. Newton-Jacobi method improves the rate of convergence of the fixed-point iteration but the cost of calculations effects the simulation time.

Outputs for Soliton Line from different methods are shown in Fig. 4.23, 4.24 and 4.25 respectively, some zoom in plots of strongly nonlinear parts are presented also. As presented all the methods converge to correct solution.

4.4.2 Convergence Rate Analysis

Convergence enhancements with MPE are shown in Fig. 4.26, 4.27 and 4.28 respectively. Similar to previous circuits, all the convergence plots are for some representative time step. The value of K and the number of iterations without MPE is very close, that is why convergence improvement is not achieved considerably, sometimes the cost of computation for MPE is expensive than its actual improvement. So both the curves are close to each other (especially for Wave_Transient2) in all graphs and the rate of convergence is linear.

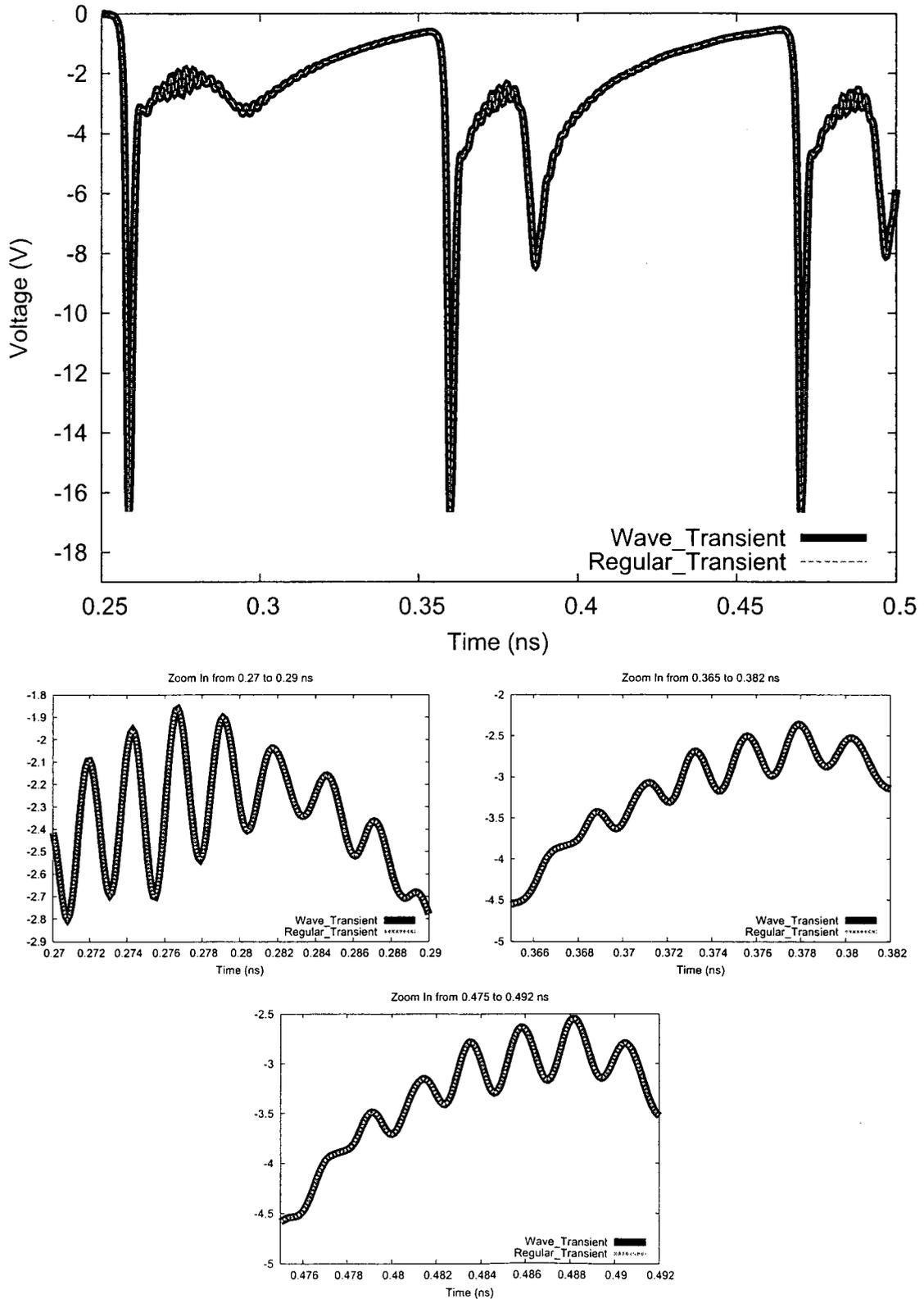


Figure 4.23: Outputs from Wave_Transient method for Soliton Line

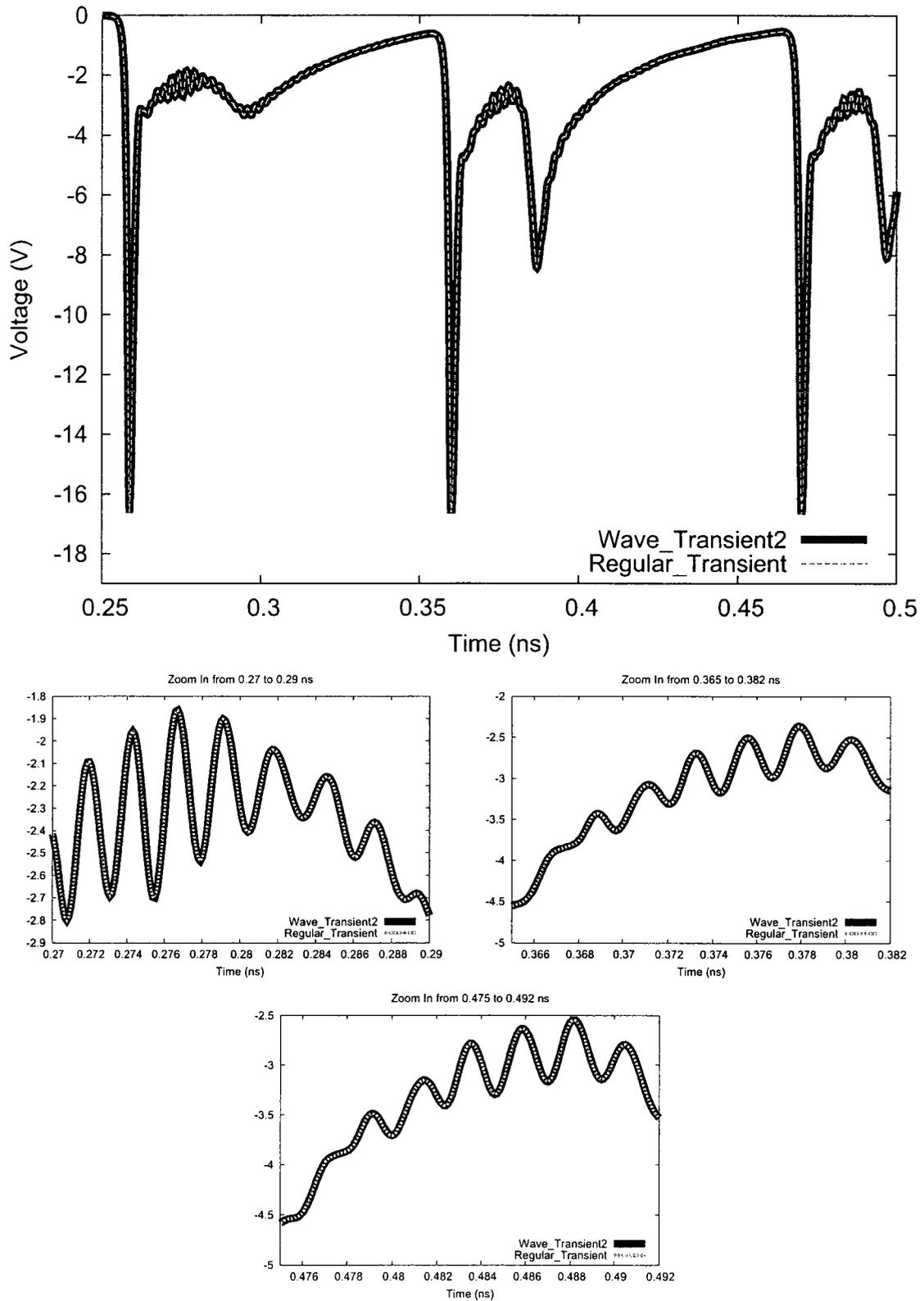


Figure 4.24: Outputs from Wave_Transient2 method for Soliton Line

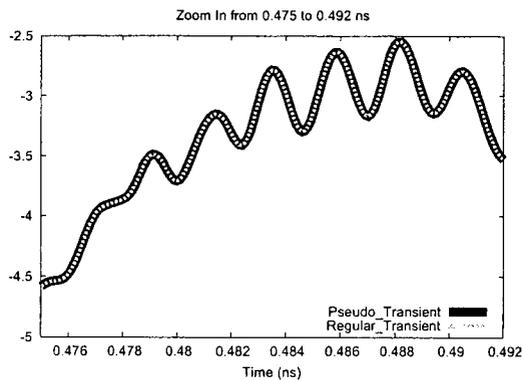
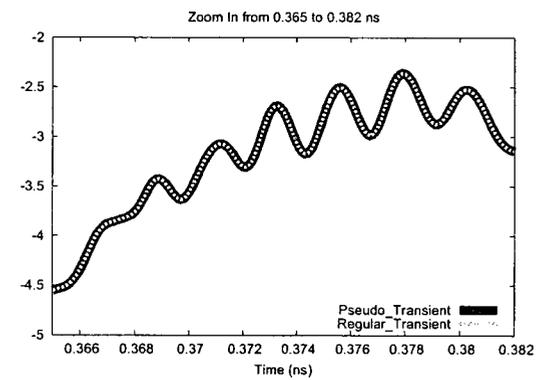
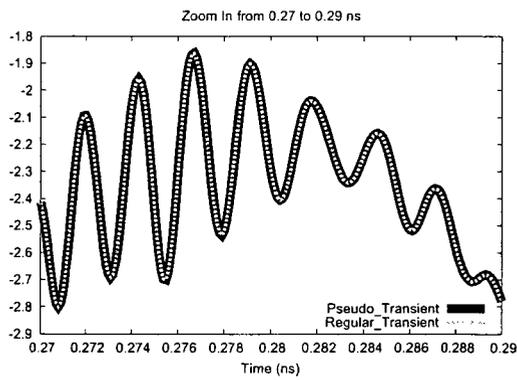
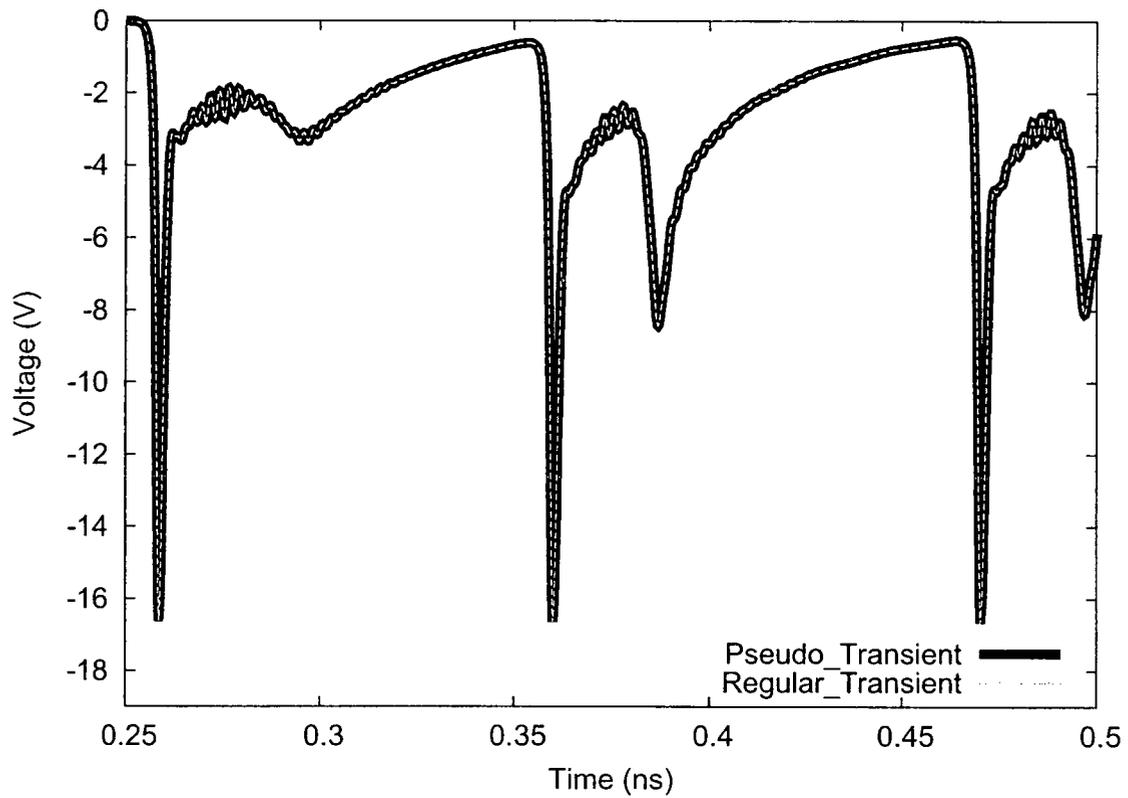


Figure 4.25: Outputs from Pseudo_Transient method for Soliton Line

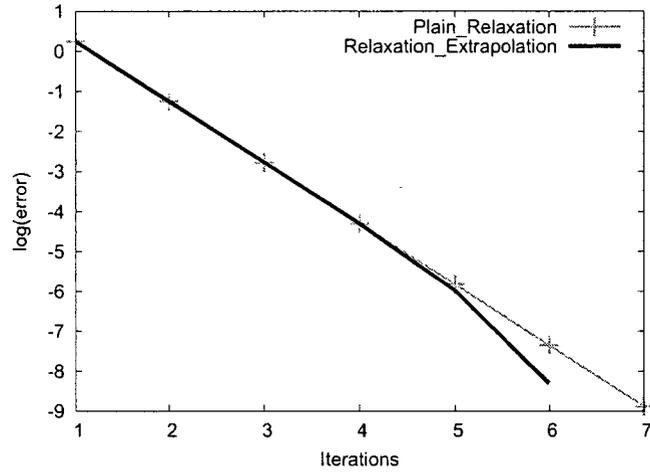


Figure 4.26: Convergence comparison from Wave_Transient method for Soliton Line

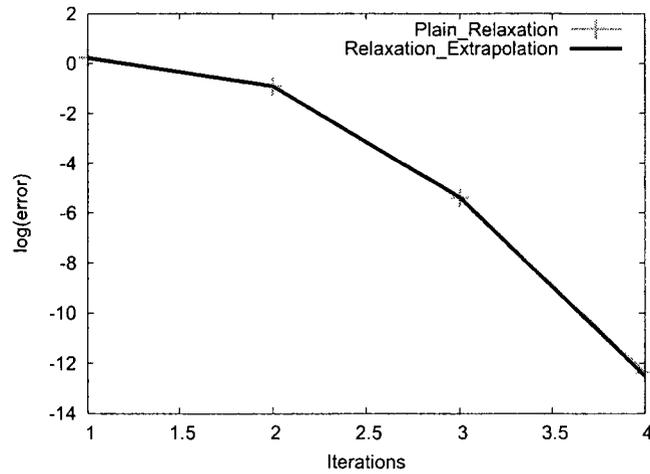


Figure 4.27: Convergence comparison from Wave_Transient2 method for Soliton Line

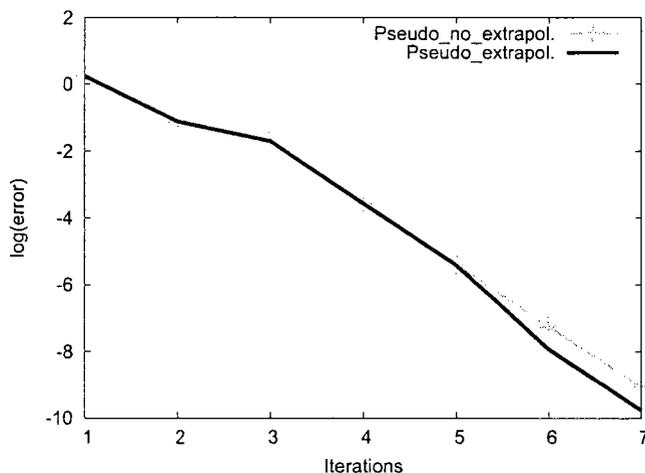


Figure 4.28: Convergence comparison from Pseudo_Transient method for Soliton Line

4.5 Multi-MMIC-Soliton Line

The Multi-MMIC-Soliton Line, shown in Fig. 4.29 is composed of 5 MMIC LNA each connected to a Soliton Line at the output stage, each LNA fed with sinusoid of different frequency. Clamping of the MMIC output is required to feed to soliton line otherwise the output behaviour will be different from the output of MMIC, output of the soliton line without clamping is shown in Fig. 4.30. A $-6 V$ DC supply is used to clamp the output of MMIC. Clamper circuit parameters and resistor values are as follows: $C_{cr4} = 20 pF$ and $L_{cr4} = 20 nH$, $R_{cr4L} = 100\Omega$ and $R_{cr4P} = 80 \Omega$.

4.5.1 Simulation Results and Outputs

Simulation results for Multi-MMIC-Soliton Line is shown in Table 4.5. Fixed-point iteration with decoupled approach of handling nonlinearity overcomes the regular approach of transient analysis for this size and type of circuits as it is not required to decompose a big matrix. For this circuit Wave_Transient2 is faster than Wave_Transient due to the dominant elements in the block diagonal pattern of scattering matrix. MPE improves the performance for all methods, considerable improvements are found in Wave_Transient2 and Pseudo_Transient.

Outputs for Multi-MMIC-Soliton Line from different methods are shown in Figs. 4.31, 4.32; 4.33, 4.34; 4.35, 4.36 respectively, some zoom in plots are presented also. All the methods converge to the correct solution.

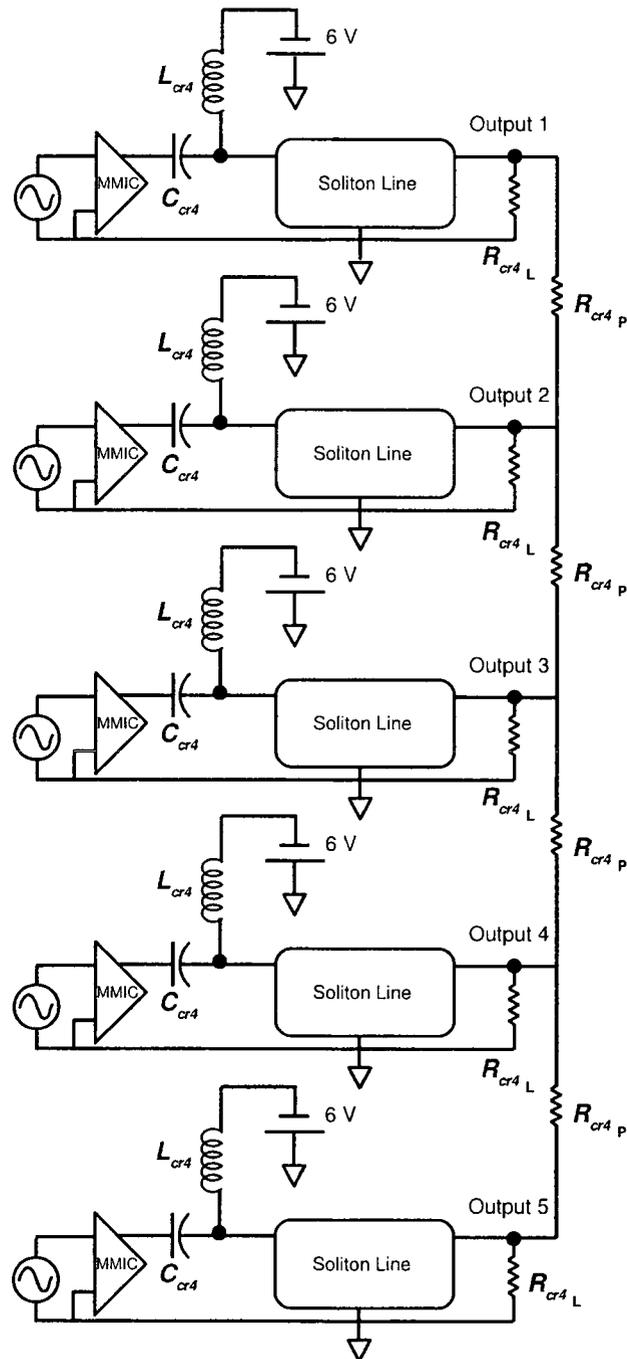


Figure 4.29: Multi MMIC Soliton Line

Table 4.5: Summary of Simulation Results for Multi-MMIC-Soliton Line

Methods	Wave_Transient	Wave_Transient2	Pseudo_Transient
Nonlinear Ports	255		
Time steps	4000		
Iterations (plain)	17	19	23.27
Iterations (MPE)	13	9	15.63
K	4		
Avg. MPE	1.90302	0.971757	2.02549
Newton Iteration (MPE)	0.99975	0.99975	0.99975
Simulation Time (plain)	2858.58 s	3944.49 s	4134.04 s
Simulation Time (MPE)	2268.04 s	2102.4 s	2941.18 s
Simulation Time (Regular_Transient)	2423.32 s		

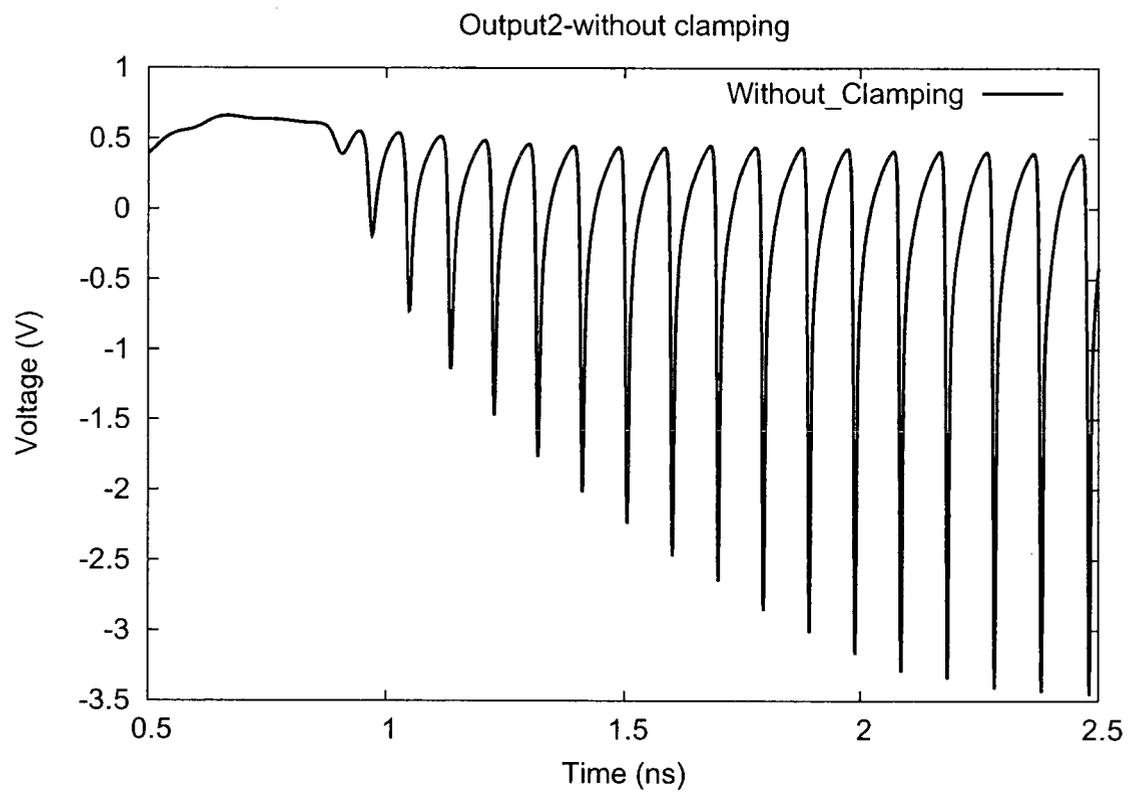


Figure 4.30: Output2 from Wave_Transient method without the clamping circuit

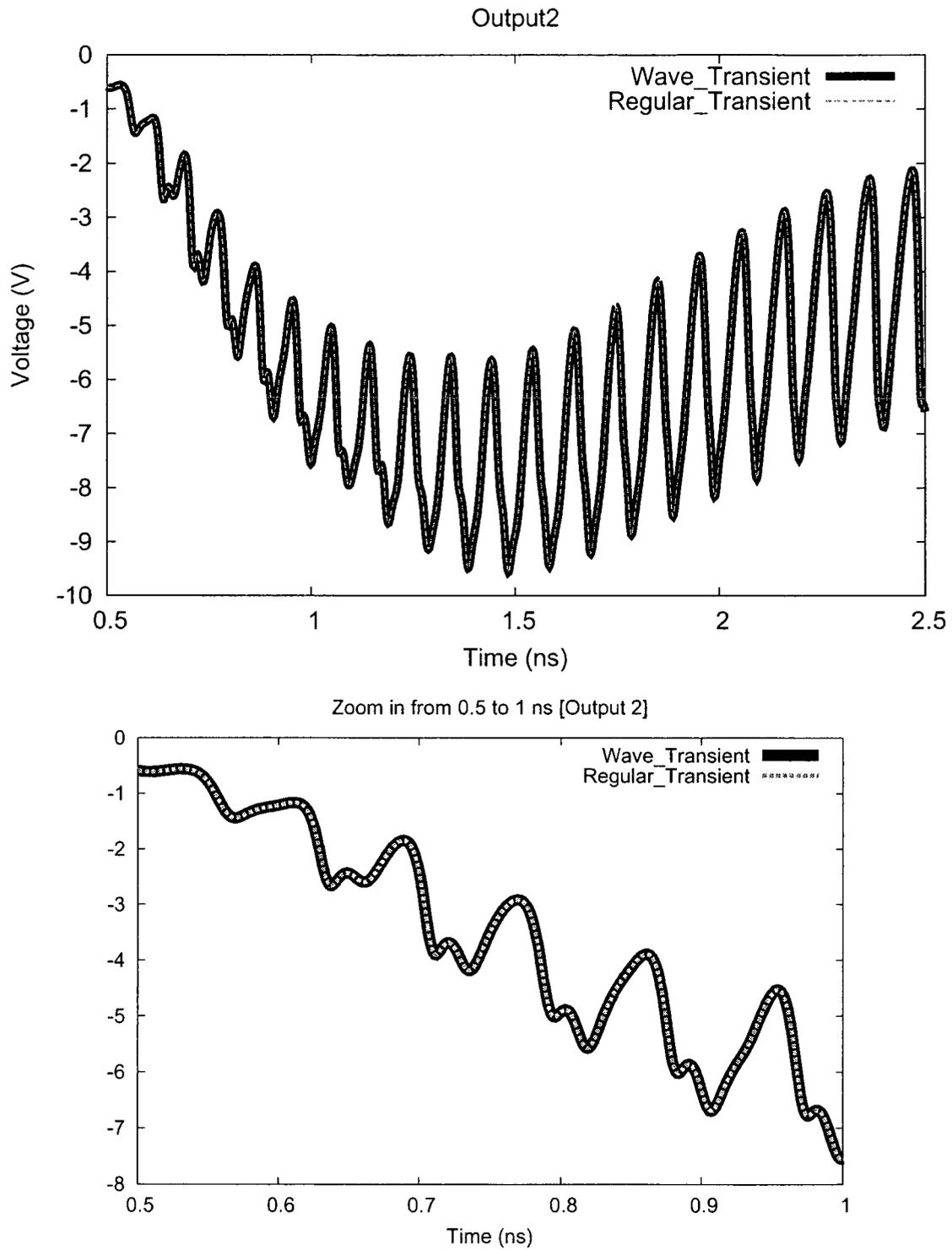


Figure 4.31: Output2 from Wave_Transient method for Multi-MMIC-Soliton Line

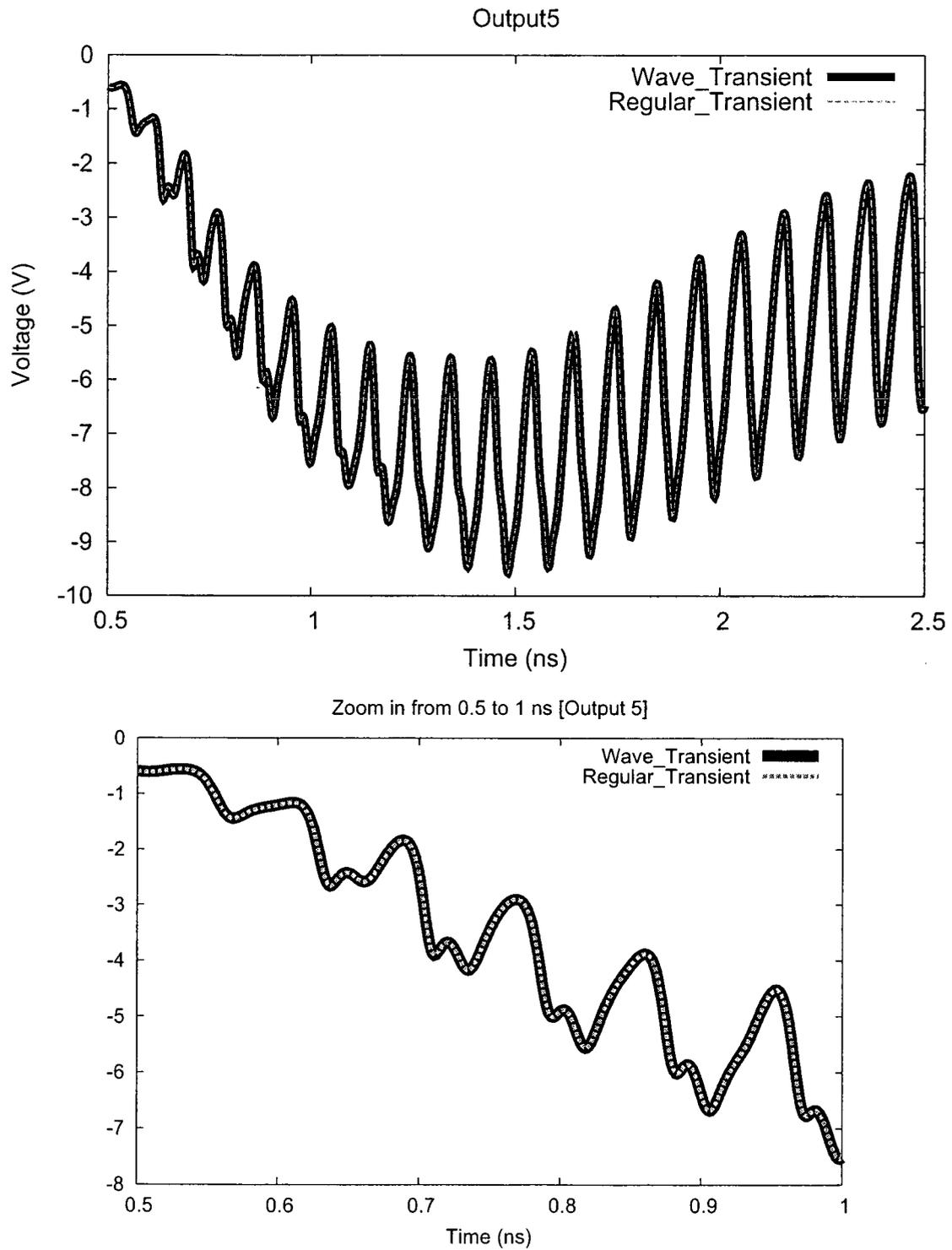


Figure 4.32: Output5 from Wave_Transient method for Multi-MMIC-Soliton Line

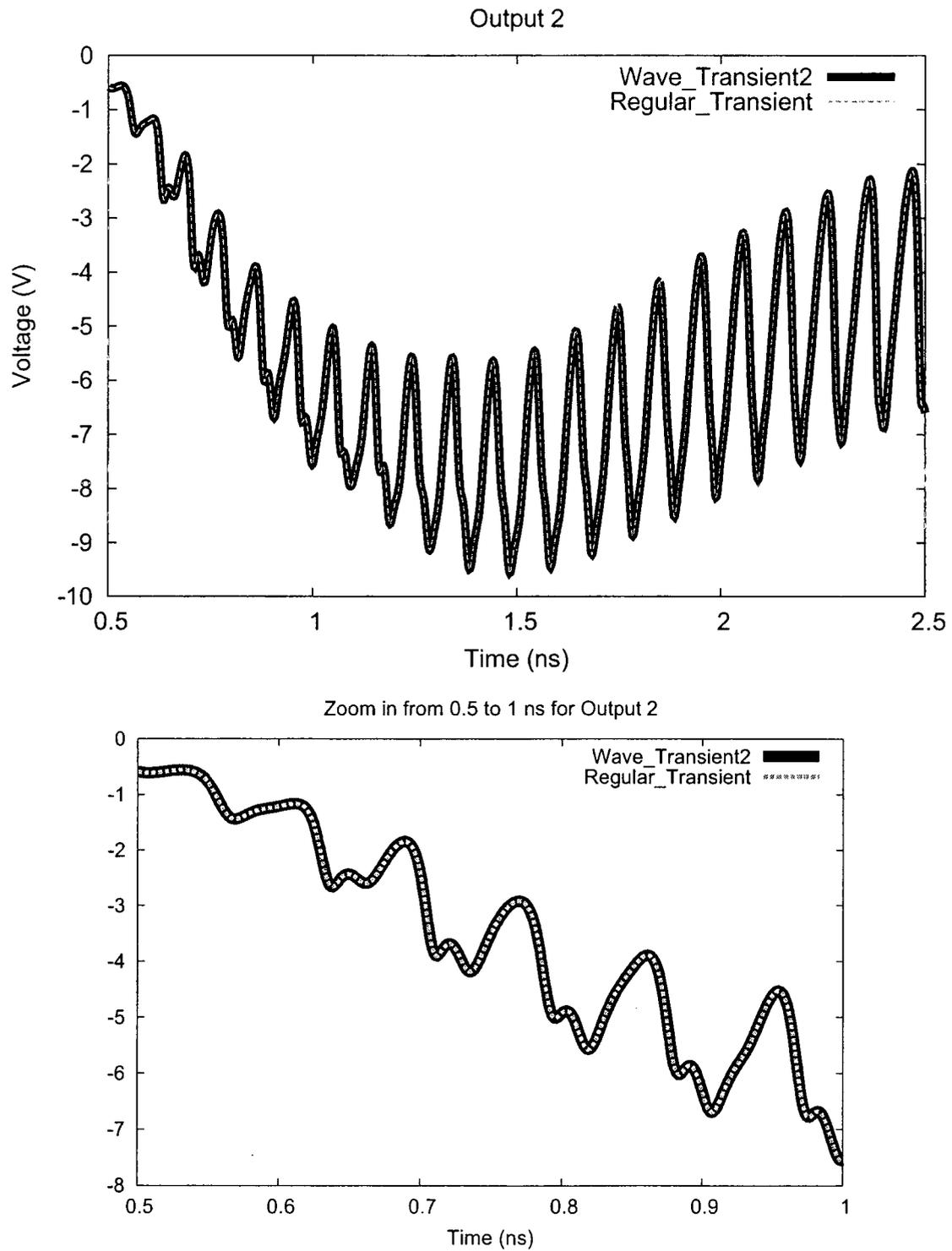


Figure 4.33: Output2 from Wave_Transient2 method for Multi-MMIC-Soliton Line

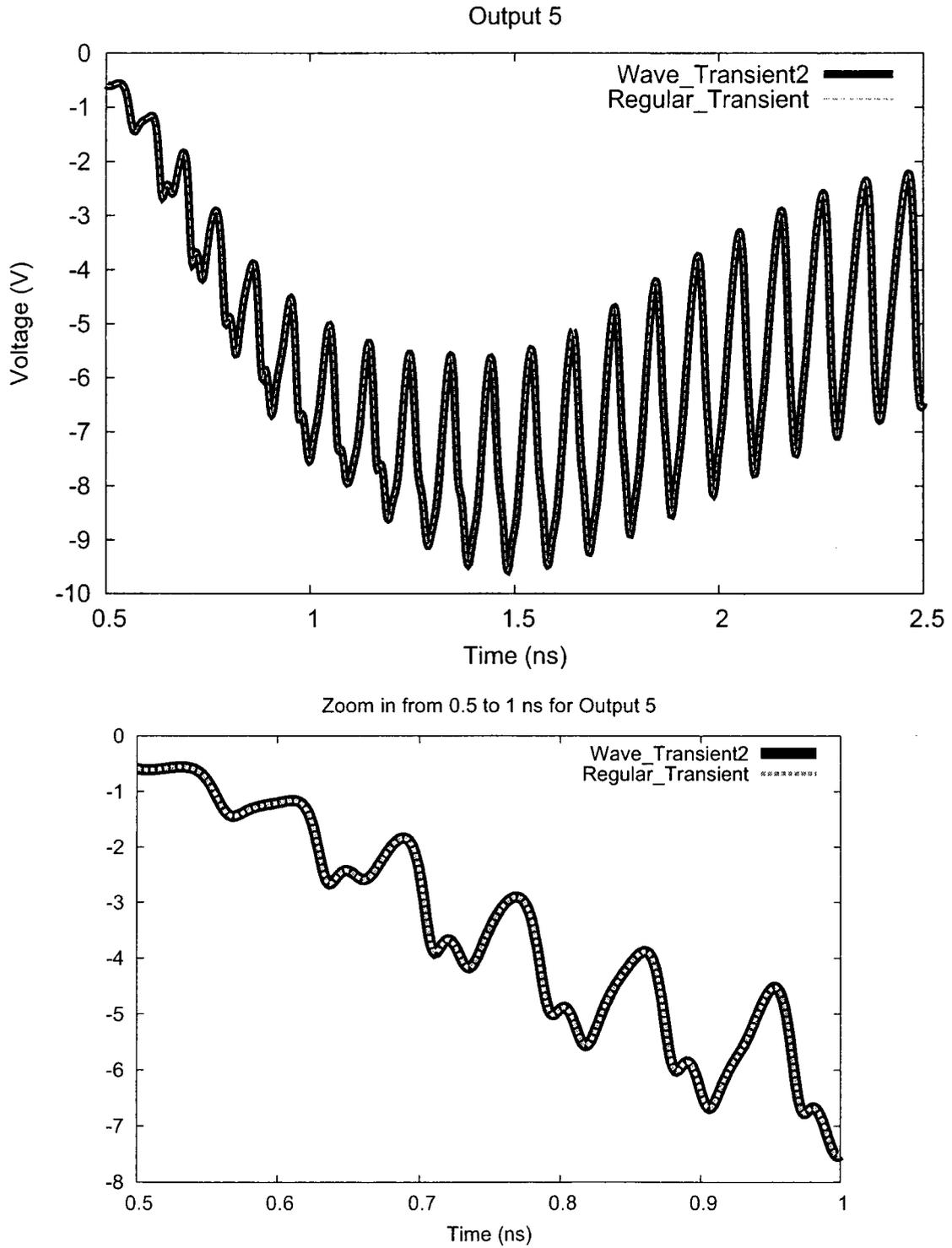


Figure 4.34: Output5 from Wave_Transient2 method for Multi-MMIC-Soliton Line

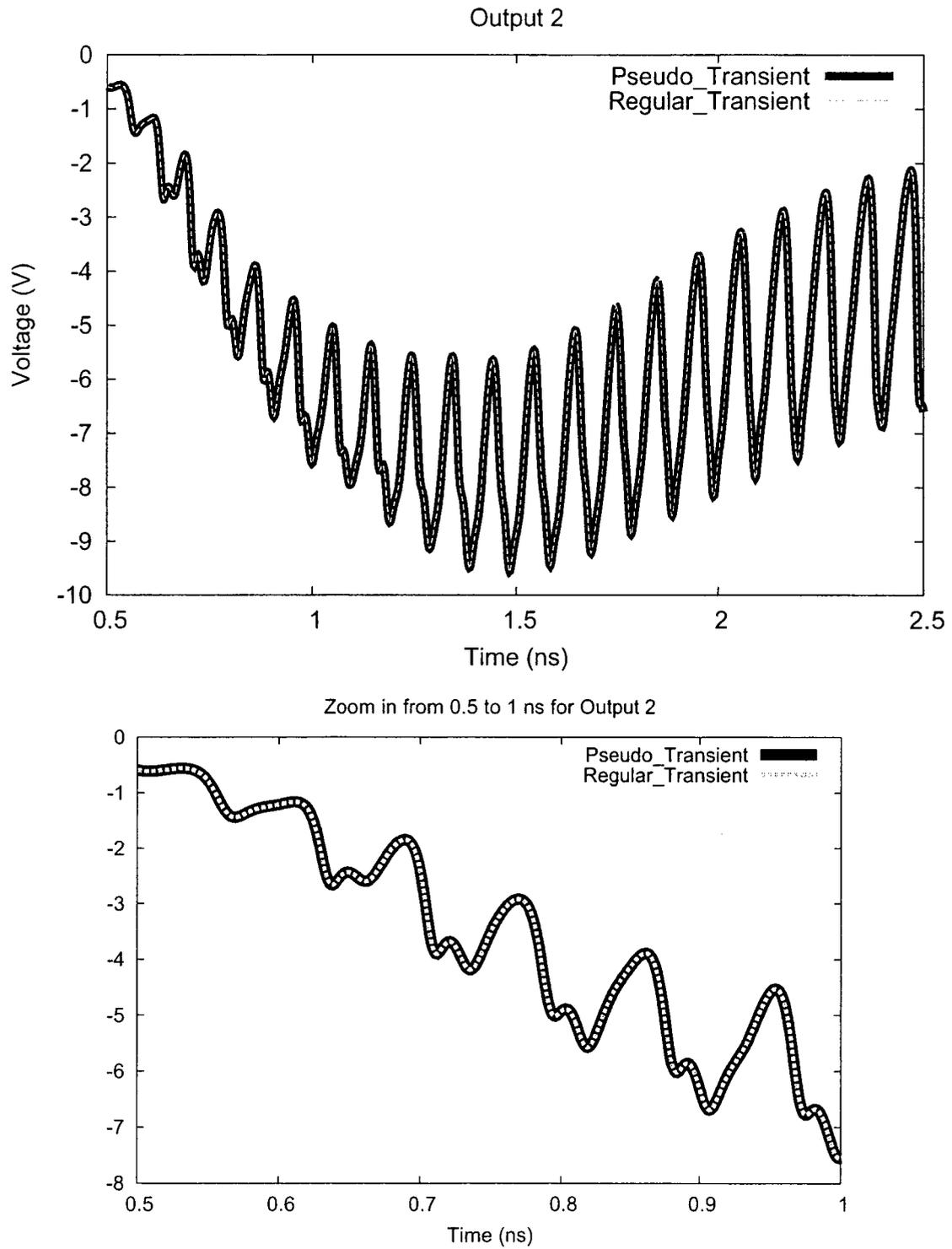


Figure 4.35: Output2 from Pseudo_Transient method for Soliton Line

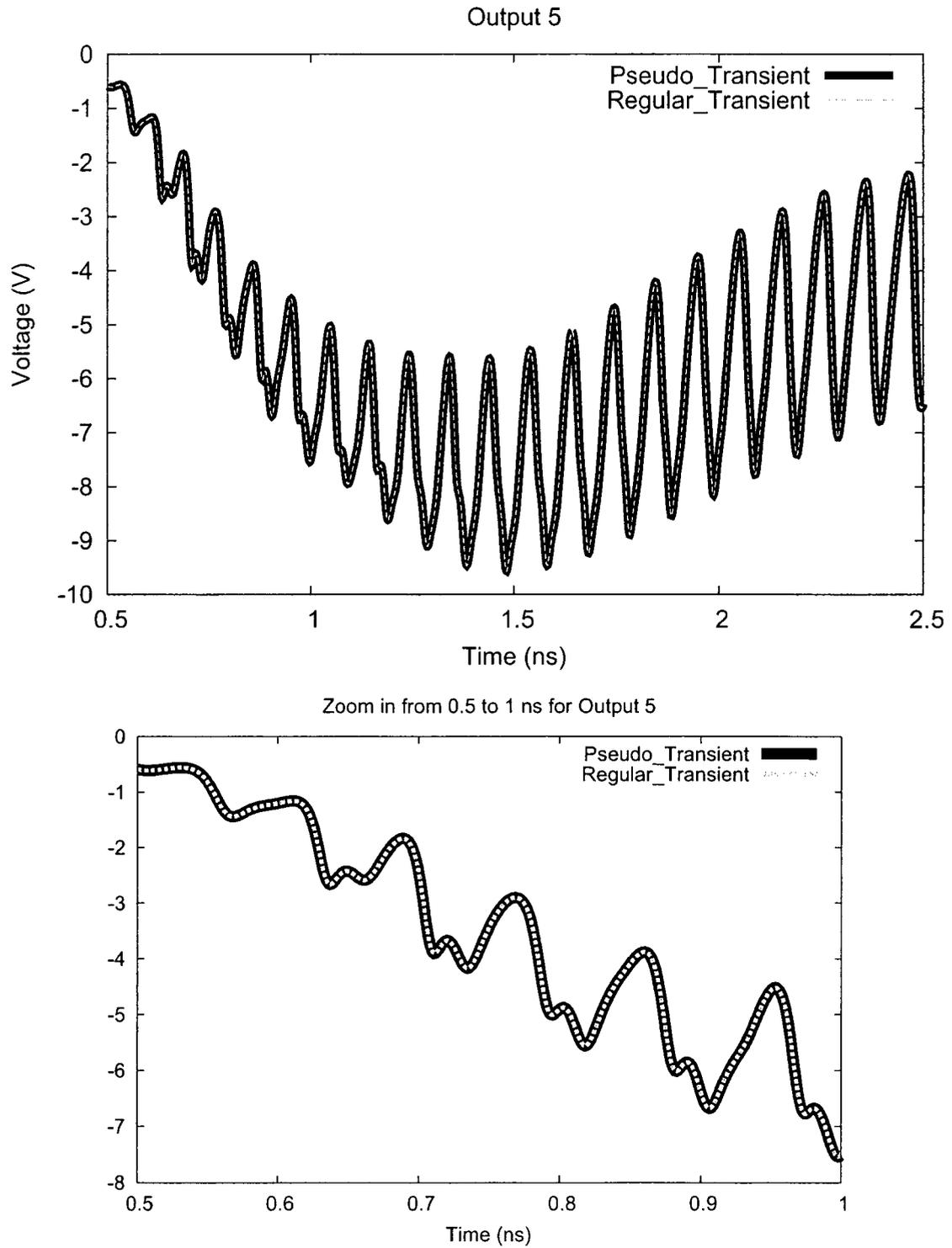


Figure 4.36: Output5 from Pseudo-Transient method for Soliton Line

4.5.2 Convergence Rate Analysis

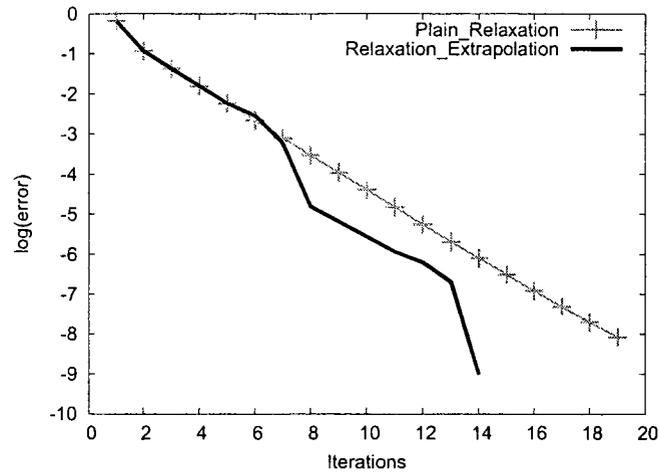


Figure 4.37: Convergence comparison from Wave_Transient method for Multi-MMIC-Soliton Line

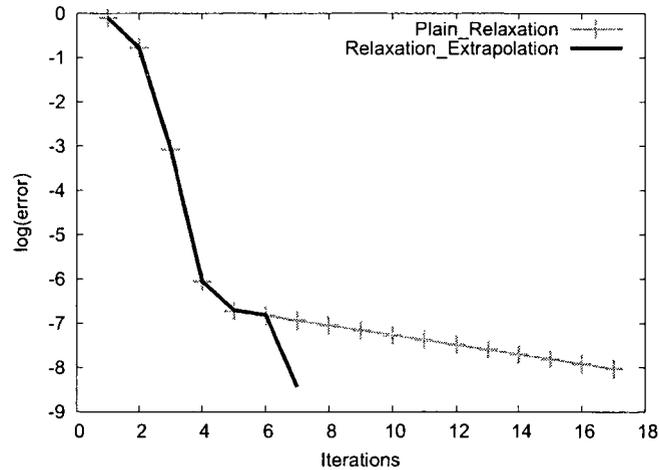


Figure 4.38: Convergence comparison from Wave_Transient2 method for Multi-MMIC-Soliton Line

Convergence improvements with MPE are shown in Fig. 4.37, 4.38 and 4.39 respectively. Like the other circuits, all the convergence plots are for some representative time step. As shown, considerable convergence improvement is found by using MPE. For Wave_Transient2 iterations move toward the solution in a faster rate but slow down after certain point, the rate is further improved by using MPE.

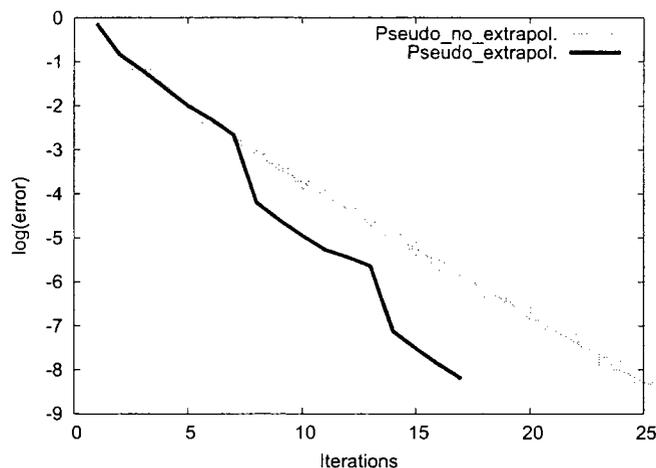


Figure 4.39: Convergence comparison from Pseudo-Transient method for Multi-MMIC-Soliton Line

4.6 Summing Amplifier

Summing amplifier shown in Fig. 4.40 is implemented with a 741 operational amplifier (26 BJTs) and a resistive feedback network. This circuit contains BJTs that are not locally passive all the time and it has a feedback connection. This circuit is used to test the correctness of solution using the proposed methods for circuit containing large number of nonlinear elements that are not locally passive all the time. 15 V rail to rail supply is used for the operational amplifier and the resistor values are as follows: $R_{cr61} = 5\text{ k}\Omega$, $R_{cr62} = 20\text{ k}\Omega$, $R_{cr64} = 20\text{ k}\Omega$ and $R_{cr63} = 3.33\text{ k}\Omega$. Amplitude and frequency for V_{cr61} are 0.1 V and 10 KHz respectively and for V_{cr62} the values are 1 V and 100 KHz respectively.

4.6.1 Simulation Results and Outputs

Simulation results for summing amplifier is shown in Table 4.6. Here, all the approaches are compared with Regular-Transient using different time step, Wave-Transient is compared using 5 ns time step, Wave-Transient2 is compared using 4 ns time step and Pseudo-Transient is compared using 0.2 μs . Fixed-point iterative scheme is considerably slower than the regular transient analysis approach for this summing amplifier though MPE improves the performance markedly. Most of the BJTs are not locally passive during the analysis, so convergence is not guaranteed. Local convergence can be achieved by using a smaller time step as mentioned in Section 3.3.2, but required simulation time is increased in significant extent. For strongly coupled

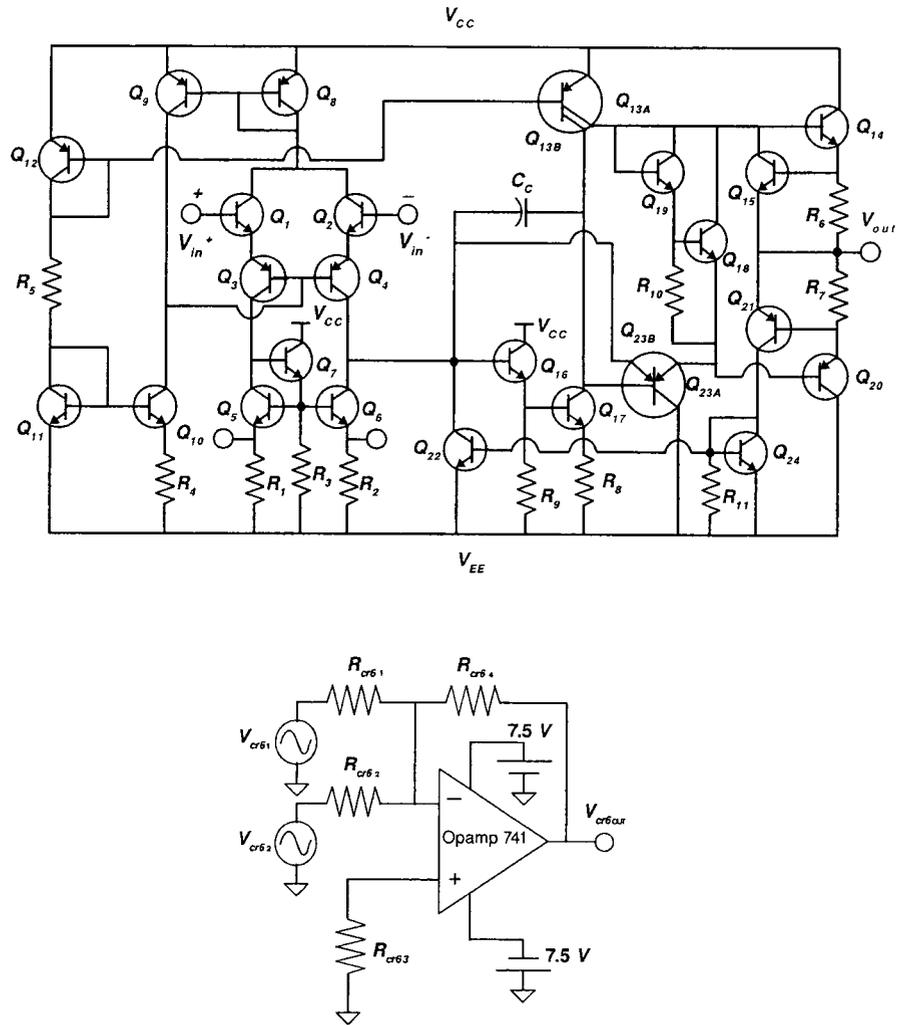


Figure 4.40: 741 Operational Amplifier Circuit and Summing Amplifier

Table 4.6: Summary of Simulation Results for Summing Amplifier

Methods	Wave_Transient	Wave_Transient2	Pseudo_Transient
Nonlinear Ports	78		
Step size	5 ns	4 ns	0.2 μ s
Time steps	8000	10000	200
Iterations (plain)	112	> 150	N/C
Iterations (MPE)	69	63	253.23
K	10		
Avg. MPE	5.50906	5.0366	20.9403
Newton Iteration (MPE)	0.999875	0.9999	0.995248
Simulation Time (plain)	6331.09 s	> 10000 s	N/C
Simulation Time (MPE)	4041.62 s	7341.96 s	500.66 s
Reset Time	N/A	N/A	58.49 s
Simulation Time (Regular_Transient)	183.22 s	226.53 s	6.63 s

circuit with feedback connections iterated timing analysis (`Wave_Transient` and `Wave_Transient2`) and timing analysis (`Pseudo_Transient`) is noticeably slower than the regular approach using Newton's method [3]. With `Pseudo_Transient` method time step size can be increased up to a certain point by resetting the iterations and scale down the reflected waves by relaxation factor at the expense of resetting time. But without MPE timing analysis of the pseudo-circuit is not convergent for all the time steps.

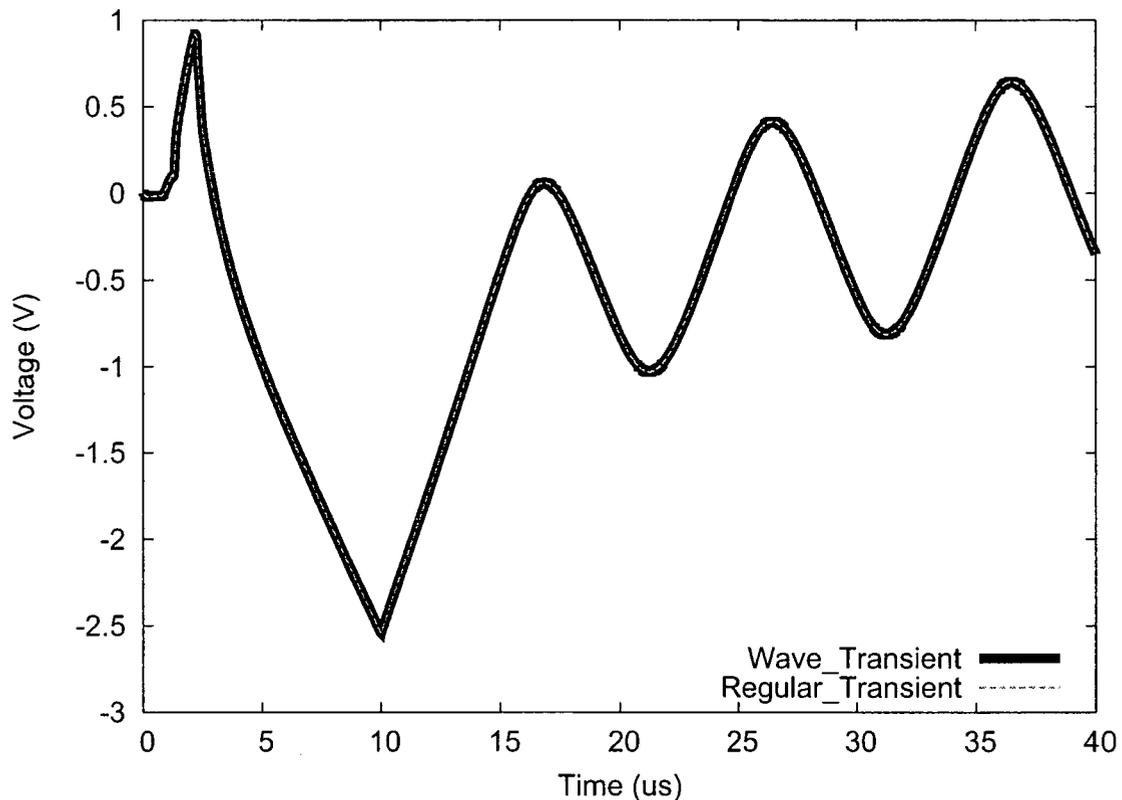


Figure 4.41: Outputs from `Wave_Transient` method for Summing Amplifier

Outputs for summing amplifier from different methods are shown in Fig. 4.41, 4.42 and 4.43 respectively. As shown, all the methods converge to correct solution for this circuit.

4.6.2 Steady-state Oscillations

Some steady-state oscillations obtained for the summing amplifier are shown in Fig. 4.44. Some representative time steps (greater than the time step for which the sequences converge) are used to generate the sequences using `Wave_Transient` and `Wave_Transient2` methods.

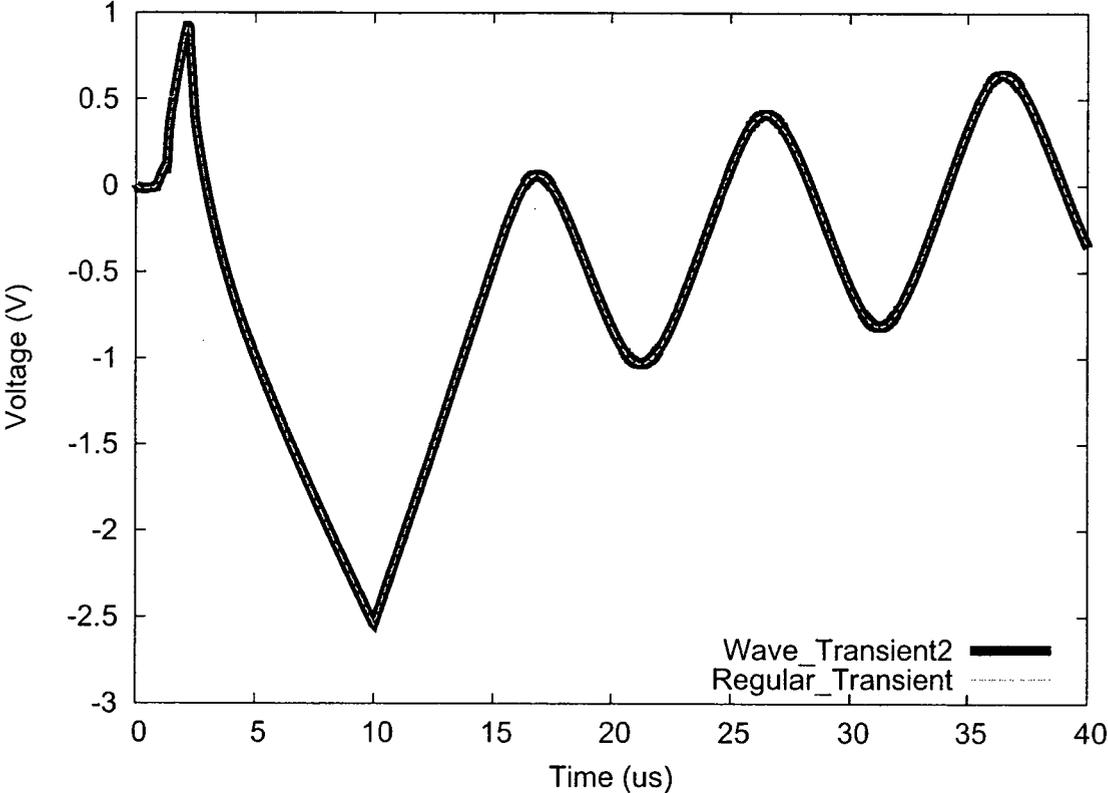


Figure 4.42: Outputs from Wave_Transient2 method for Summing Amplifier

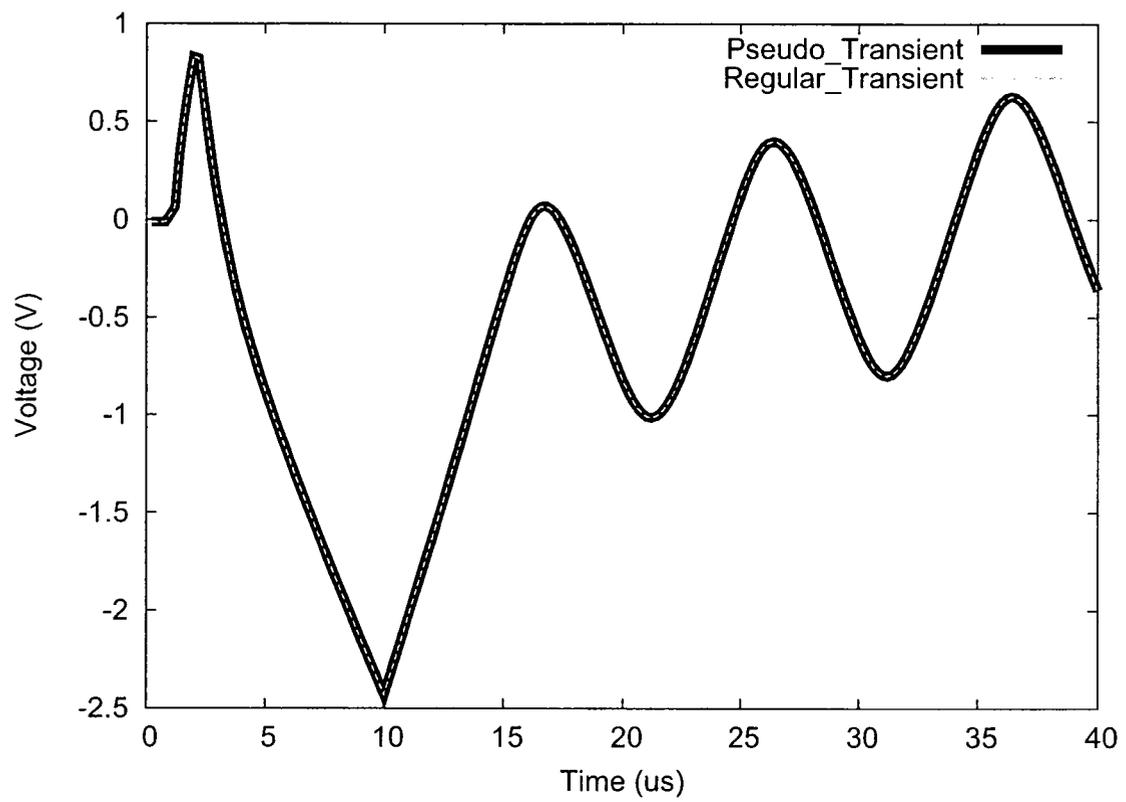


Figure 4.43: Outputs from Pseudo_Transient method for Summing Amplifier

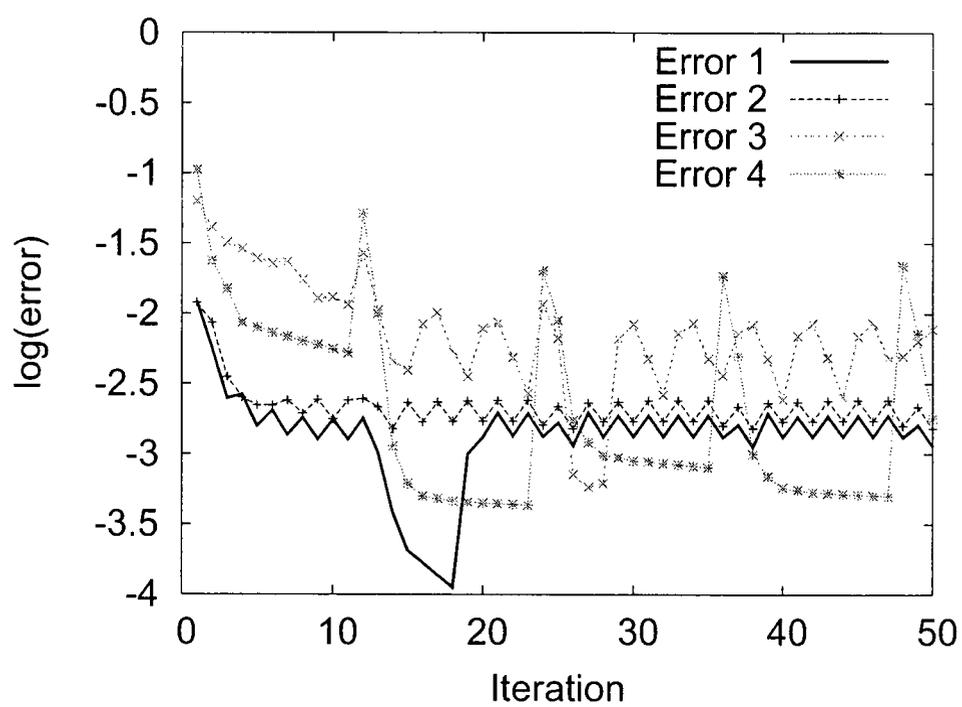


Figure 4.44: Steady-state oscillation from different sequences

Chapter 5

Conclusion and Future Research

5.1 Summary of Research Works and Original Contributions

A nonlinear transient analysis based on state variables and relaxation of power waves has been presented for the first time. Three variations of the approach have been demonstrated: plain relaxation of waves, block Newton-Jacobi with wave variables and a Pseudo-Transient analysis at each time step. Newton-Jacobi and Pseudo-Transient approaches were proposed to make the method convergent for a wider variety of circuits. In addition these methods have been combined with MPE extrapolation to improve the convergence rate of these formulations. This was also reported here for the first time. Finally the proposed methods have been demonstrated with nonlinear circuits of different size and variety.

The well-behaved numerical properties given by the parametric formulation of the nonlinear device equations are attained at the expense of having to solve for state variables. The plain relaxation and Pseudo-Transient approaches using power waves can not diverge to infinity. This assures physically meaningful excitation to nonlinear device models and thus numerical ill-conditions can be circumvented. The block Newton-Jacobi presented here would have to be modified to have this property. Indicative improvement of convergence of relaxation methods is achieved by the utilization of MPE extrapolation. The demonstrated approach is always convergent for any circuit containing only locally passive nonlinear devices regardless of the initial guess. Furthermore, for circuits that contain mostly locally passive nonlinear devices and transistors separated by important parasitic passive networks the proposed method is usually convergent. As the size of the circuit increases the proposed method becomes more efficient because it is

not required to decompose a large matrix in each iteration. Since the only step that can not be computed locally is a matrix-vector multiplication ($[\mathbf{S}] \times \mathbf{F}(\)$), the method may be suitable to develop a parallel algorithm.

Despite these attractive properties several issues remain to be solved to make the approach practical for general circuit simulation. For smaller circuits, the presented approach is slower than the regular methods. One of the reasons for that is to solve for twice as many variables at the nonlinear device level (the state variables plus the reflected waves). Another reason is the slower convergence rate of the approach compared to Newton's method. Finally, the algorithm was not implemented with efficiency as the main priority. The proposed relaxation method is significantly slower than the regular methods for circuits containing high gain, strongly coupled nonlinear devices and feedback connections (*e.g.*, summing amplifier with operational amplifier 741). Additionally, convergence is not guaranteed for the circuit that contains nonlinear elements that are not locally passive. In presence of parasitic capacitor or inductor convergence can be achieved by reducing the time step size, but total simulation time increases in a great extent with this smaller time step. Though the proposed algorithm can readily be implemented in a parallel computer system, the cost of communication may cancel any performance gain of the approach.

5.2 Future Work

There are several directions left open in this research for future work. The major one is the exploration of different numerical methods and techniques to assure convergence for any circuit, this would be a remarkable contribution. There are ample opportunities to improve the performance of circuit simulators in a greater extent by implementing those in parallel computer systems. By the development of multi core processors, the implementation is not limited to cluster computers only. It is possible to utilize both in a single program if the computers in a cluster are equipped with multi core processors. The proposed approach can be modified easily to develop a parallel algorithm. Most of the data type used in the simulator is double-precision 64-bit floating point, so all the operations to simulate a circuit are mainly double-precision floating point operations. GPU (Graphics Processing Unit) is more powerful than the PC (Personal Computer) processors in terms of floating point operation. As of 2008, the fastest PC processor

Intel Core i7 965 XE (quad-core) perform over 70 GFLOPS (Giga Floating Point Operations per Second) in double-precision floating point whereas AMD ATI Radeon HD 5970 is capable of 928 GFLOPS in double precision [60]. The performance is even better for single-precision floating point operation. The technique of using a GPU to perform computation in applications traditionally handled by the CPU is termed as GPGPU (General Purpose computing on Graphics Processing Units). The idea of GPGPU in circuit simulation is quite new, there are some works related to the evaluation of transistor model equations using a GPU [45, 33]. So an attempt could be made to use GPU for performing the calculations for each nonlinear device, this will improve the performance of the approach considerably. A greater portion of current simulation time can be saved by using variable time step. Before attaining the steady-state, smaller time step is required, but after that time step size can be made longer. As discussed in [12], the time step can be made variable in the proposed approach at the expense of one matrix decomposition (calculation of scattering matrix, $[S]$) each time the step size is changed.

5.2.1 Waveform Relaxation of Power Waves

The fixed-point iterative scheme, developed in this work, can be implemented in parallel in principle. But the communication cost in each iteration will reduce the performance of the simulator while implemented in parallel in cluster computers (*i.e.* using MPI-Message Passing Interface). Though Multi-Thread or Multi-Processor implementation (*i.e.* using OpenMP-Open Multi-Processing) may improve the performance, more efficient parallel algorithms can be developed using WR of power waves to utilize both computer clusters and multi-thread in the same simulator. In parallel WR approach, iterates are communicated between processors only after having been computed over a time interval [56, 28, 19].

The basic idea would be the propagation of the power waves between linear and nonlinear subnetworks, the same as implemented here. But instead of the time discretized value of the vectors, the whole waveform of reflected and incident waves have to be calculated at each iteration. Eq. (3.18). can be rewritten as follows:

$$\mathbf{a}_{wr+1}(t) = [S] \mathbf{b}_{wr}(t) + \mathbf{a}_0(t) \quad (5.1)$$

where, wr denotes the iteration number in WR method and the power wave vectors defined as

follows:

$$\mathbf{a}(t) = \begin{bmatrix} a_1(t) \\ a_2(t) \\ \vdots \\ a_{n_s}(t) \end{bmatrix}, \mathbf{b}(t) = \begin{bmatrix} b_1(t) \\ b_2(t) \\ \vdots \\ b_{n_s}(t) \end{bmatrix} \text{ and } \mathbf{a}_0(t) = -[\mathbf{M}_{SV} \mathbf{D}^{-1} - \mathbf{D}]^{-1} \mathbf{S}_{SV}(t)$$

Similarly Eq. (3.22). can be expressed as follows:

$$\mathbf{b}_{wr+1}(t) = \mathbf{F}([\mathbf{S}] \mathbf{b}_{wr}(t) + \mathbf{a}_0(t)) \quad (5.2)$$

The WR method is planned to implement using the same procedure mentioned in Subsection 2.6.3. There are different convergence improvement methods for WR, yet to be investigated, for this implementation.

Appendix A

Newton's Method

Newton's method (also known as the Newton-Raphson method) is a well-known technique in Numerical Analysis used for finding successively better approximations to the roots of a real-valued function. Let, $f_a(y)$ is a function of y and $f'_a(y)$ be its derivative, our initial guess of the root be y_0 . A better approximation of the root y_1 (provided that the function is reasonably well-behaved) is

$$y_1 = y_0 - \frac{f_a(y_0)}{f'_a(y_0)}$$

this successive approximation is repeated until a sufficient accuracy is achieved.

$$y_{nt+1} = y_{nt} - \frac{f_a(y_{nt})}{f'_a(y_{nt})} \quad (\text{A.1})$$

where nt is the iteration number. This method generally with some modifications is used to solve the systems of nonlinear equations and most notorious in circuit analysis also. Let $\mathbf{F}_a(\mathbf{Y}) = 0$ is the vector notation of \mathbf{n}_a nonlinear equations in \mathbf{n}_a variables where $\mathbf{Y} \in \mathbb{R}^{\mathbf{n}_a}$ and $\mathbf{F}_a : \mathbb{R}^{\mathbf{n}_a} \rightarrow \mathbb{R}^{\mathbf{n}_a}$ are defined as,

$$\mathbf{F}_a(\mathbf{Y}) = \begin{bmatrix} f_{a1}(y_1, \dots, y_{\mathbf{n}_a}) \\ f_{a2}(y_1, \dots, y_{\mathbf{n}_a}) \\ \vdots \\ f_{a_{\mathbf{n}_a}}(y_1, \dots, y_{\mathbf{n}_a}) \end{bmatrix}$$

If $\mathbf{F}_a : \mathbb{R}^{\mathbf{n}_a} \rightarrow \mathbb{R}^{\mathbf{n}_a}$ is differentiable then Jacobian matrix, $[\mathbf{J}_{F_a}]$ of function $\mathbf{F}_a(\mathbf{Y})$ is defined as follows

$$[\mathbf{J}_F(\mathbf{Y})] = \begin{bmatrix} \frac{\partial f_{a_1}(\mathbf{Y})}{\partial y_1} & \frac{\partial f_{a_1}(\mathbf{Y})}{\partial y_2} & \dots & \frac{\partial f_{a_1}(\mathbf{Y})}{\partial y_{n_a}} \\ \frac{\partial f_{a_2}(\mathbf{Y})}{\partial y_1} & \frac{\partial f_{a_2}(\mathbf{Y})}{\partial y_2} & \dots & \frac{\partial f_{a_2}(\mathbf{Y})}{\partial y_{n_a}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{a_{n_a}}(\mathbf{Y})}{\partial y_1} & \frac{\partial f_{a_{n_a}}(\mathbf{Y})}{\partial y_2} & \dots & \frac{\partial f_{a_{n_a}}(\mathbf{Y})}{\partial y_{n_a}} \end{bmatrix}$$

So, Newton's method is generalized to find the solutions of vector function $\mathbf{F}_a(\mathbf{Y}) = 0$ as follows:

$$\mathbf{Y}_{nt+1} = \mathbf{Y}_{nt} - [\mathbf{J}_{F_a}(\mathbf{Y}_{nt})]^{-1} [\mathbf{F}_a(\mathbf{Y}_{nt})] \quad (\text{A.2})$$

The most important observation is that Newton's method is based on decomposition of Jacobian matrix, $[\mathbf{J}_{F_a}(\mathbf{Y})]$ of size $n_a \times n_a$ in each iteration.

Appendix B

Fixed Point Iteration

y_P is a fixed point of a function $g(y)$ if and only if $g(y_P) = y_P$. Let us say, we are attempting to solve the equation $f(y) = 0$. Now if we rewrite the equation in the form to find the fixed point, $y = g(y)$ and finding a value of y for which $y = g(y)$ is thus equivalent to finding a solution of the equation $f(y) = 0$. Fixed point iteration is a method of computing fixed points of a function in an iterative approach. Suppose our initial assumption y_0 lies near a fixed point, y_P of function $g(y)$, iterative scheme under appropriate circumstances to find the fixed point of the function can be exploited as follows:

$$y_{f+1} = g(y_f) \quad \text{until } |y_{f+1} - y_f| \leq \varepsilon(f_p)$$

where $f = 0, 1, 2, \dots$ is the successive iteration number and $\varepsilon(f_p)$ is a predefined tolerance.

Appendix C

State Variable Approach

Currents are directly expressed as a function of voltages in a traditional way of circuit analysis. But in the *state variable approach*, an independent variable $x_{sv}(t)$ is introduced to express both currents and voltages. This approach was first proposed for Harmonic Balance (HB) simulation in [57]. The state variables may be selected to attain robust numerical characteristics that provides great flexibility for the design of nonlinear device models [8]. For example, we will present the parameterized diode model as depicted in [57]. Conventional current equation for diode (for didactic purpose we are not including the full model) is as follows:

$$i_d(t) = I_{sat}(e^{\alpha_s v_d(t)} - 1)$$

where, i_d is the current and v_d is the voltage across a p-n junction diode, I_{sat} is the saturation current or scale current (typical value 10^{-12} A) and $\alpha_s = \frac{1}{m_i V_T}$, V_T is the thermal voltage (26 mV at normal temperature) and m_i is the diode ideality factor (for silicon diodes m_i is approximately 1 to 2).

As proposed in [57], the parametric model of diode is as follows:

$$v_d(t) = \begin{cases} x_{sv}(t) & \text{if } x_{sv}(t) \leq V_1 \\ V_1 + \frac{1}{\alpha_s} \ln(1 + \alpha_s(x_{sv}(t) - V_1)) & \text{if } x_{sv}(t) > V_1 \end{cases}$$
$$i_d(t) = \begin{cases} I_{sat}(e^{\alpha_s x_{sv}(t)} - 1) & \text{if } x_{sv}(t) \leq V_1 \\ I_{sat} e^{\alpha_s V_1} (1 + \alpha_s(x_{sv}(t) - V_1)) - I_{sat} & \text{if } x_{sv}(t) > V_1 \end{cases}$$

where V_1 is some threshold value.

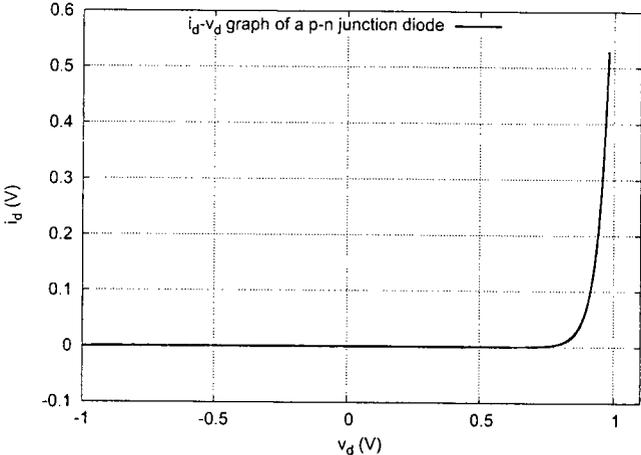


Figure C.1: Relation between v_d and i_d

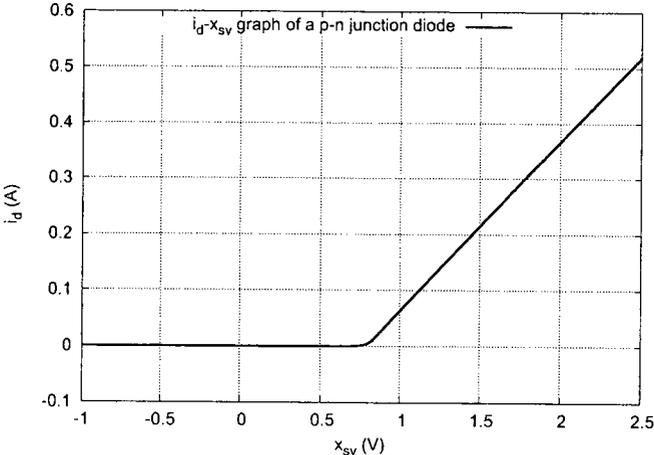


Figure C.2: Relation between x_{sv} and i_d

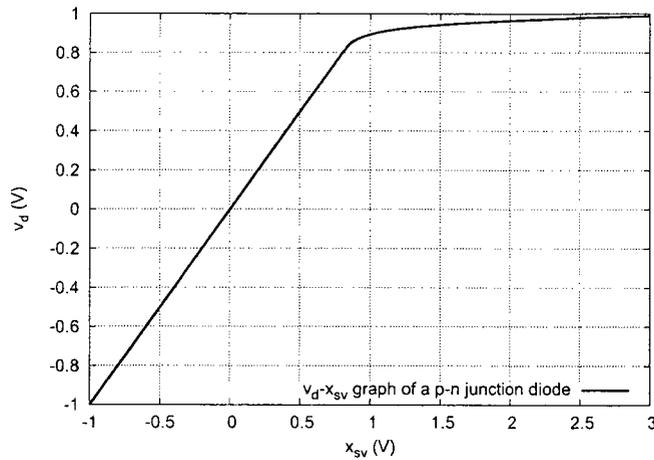


Figure C.3: Relation between x_{sv} and v_d

Current has an exponential dependence on voltage in a diode (Fig. C.1) that is a strong source of numerical shortcoming as this exponential function creates convergence difficulties when the voltage is updated during nonlinear iteration as in normal circuit simulation. A small change in voltage can result a large current changes after the threshold voltage, so large variations in error function must be solved to simulate the circuit. Special consideration in voltage changes is required to handle this type of functions so that there can only be a small change in i_d . parameterization annihilates the contingency of large changes that assures smooth, well behaved current (Fig. C.2), voltage (Fig. C.3) and error function variations when the state variable is updated.

Appendix D

Modified Nodal Analysis

Modified Nodal Analysis (MNA) is a powerful method of circuit analysis. The basic idea of MNA is to add to the set of variables used in *nodal analysis* (*i.e.*, the node voltages) just enough variables to make the solution of the circuit possible. So MNA often results in larger system of equations than the other methods, but this method of circuit analysis is easier to implement algorithmically for computer aided analysis of circuits. To use MNA for the analysis of circuits usually following steps are performed:

1. Select one node as reference and name that node as node '0' and name other nodes as '1', '2', ..., ' $n_{mna} - 1$ ' where n_{mna} is the number of nodes in the circuit.
2. Current through the voltage sources (independent/dependent) and inductors are assigned a name and passive convention is followed for the flow of current (*i.e.*, currents are assumed to flow from positive to negative node).
3. Apply KCL in each node and currents out of the nodes are taken to be positive.
4. Write an equation for voltage sources and inductor currents.
5. Solve the system of $n_{mna} - 1$ unknowns.

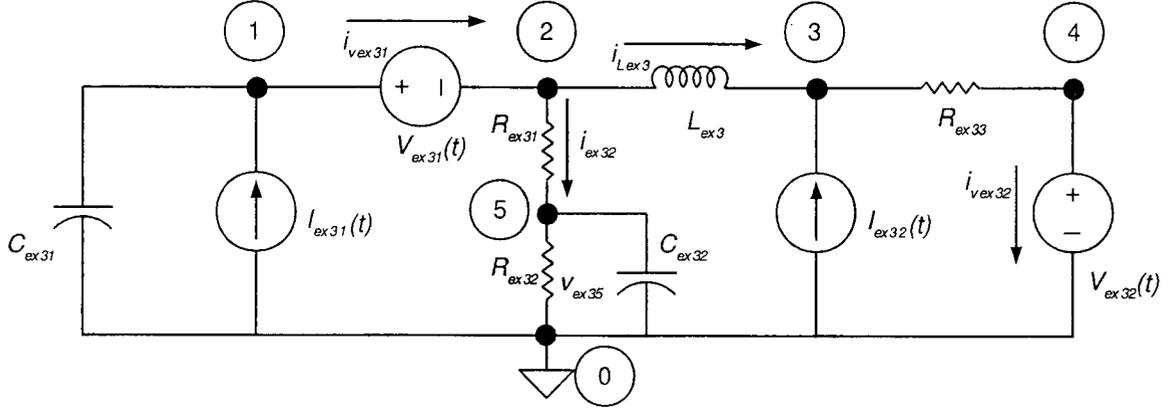


Figure D.1: Example circuit to explain MNA

For example, let us consider the circuit in Fig. D.1. The equations we have to consider for analyzing the circuit are as follows:

$$\begin{aligned}
 \text{Node 1:} & \quad C_{ex31} \dot{v}_{ex31} + i_{v_{ex31}} & = I_{ex31}(t) \\
 \text{Node 2:} & \quad \frac{(v_{ex32} - v_{ex35})}{R_{ex31}} - i_{v_{ex31}} + i_{L_{ex3}} & = 0 \\
 \text{Node 3:} & \quad \frac{(v_{ex33} - v_{ex34})}{R_{ex33}} - i_{L_{ex3}} & = I_{ex32}(t) \\
 \text{Node 4:} & \quad \frac{(v_{ex34} - v_{ex33})}{R_{ex33}} + i_{v_{ex32}} & = 0 \\
 \text{Node 5:} & \quad \frac{(v_{ex35} - v_{ex32})}{R_{ex31}} + \frac{v_{ex35}}{R_{ex32}} + C_{ex32} \dot{v}_{ex35} & = 0 \\
 \text{Voltage Source 1:} & \quad v_{ex31} - v_{ex32} & = V_{ex31}(t) \\
 \text{Inductor 1:} & \quad v_{ex32} - v_{ex33} - L_{ex3} \dot{i}_{L_{ex3}} & = 0 \\
 \text{Voltage Source 2:} & \quad v_{ex34} & = V_{ex32}(t)
 \end{aligned}$$

where, v_{ex3r} is the node voltage at node r . The equations in matrix form are as follows:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{1}{R_{ex31}} & 0 & 0 & -\frac{1}{R_{ex31}} & -1 & 1 & 0 \\ 0 & 0 & \frac{1}{R_{ex33}} & -\frac{1}{R_{ex33}} & 0 & 0 & -1 & 0 \\ 0 & 0 & -\frac{1}{R_{ex33}} & \frac{1}{R_{ex33}} & 0 & 0 & 0 & 1 \\ 0 & -\frac{1}{R_{ex31}} & 0 & 0 & \frac{1}{R_{ex31}} + \frac{1}{R_{ex32}} & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{ex31} \\ v_{ex32} \\ v_{ex33} \\ v_{ex34} \\ v_{ex35} \\ i_{vex31} \\ i_{L_{ex3}} \\ i_{vex32} \end{bmatrix}$$

$$+ \begin{bmatrix} C_{ex31} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{ex32} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -L_{ex3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{ex31} \\ v_{ex32} \\ v_{ex33} \\ v_{ex34} \\ v_{ex35} \\ 0 \\ i_{L_{ex3}} \\ 0 \end{bmatrix} = \begin{bmatrix} I_{ex31}(t) \\ 0 \\ I_{ex32}(t) \\ 0 \\ 0 \\ V_{ex31}(t) \\ 0 \\ V_{ex32}(t) \end{bmatrix}$$

$$\Rightarrow [\mathbf{M}_{mna}] \mathbf{u}_{mna}(t) + [\mathbf{C}_{mna}] \frac{d\mathbf{u}_{mna}(t)}{dt} = \mathbf{S}_{mna}(t)$$

where $[\mathbf{M}_{mna}]$ is the MNAM of the circuit, $[\mathbf{C}_{mna}]$ is a matrix containing the capacitances and inductances, \mathbf{S}_{mna} is the source vector and \mathbf{u}_{mna} is a vector of nodal voltages and selected currents.

Highlighted portion of $[\mathbf{M}_{mna}]$ $((n_{mna} - 1) \times (n_{mna} - 1))$ matrix only contains the conductances from the resistors. Besides the highlighted portion is symmetric with positive values along the diagonal, and only negative (or zero) values for the off-diagonal terms. Elements connected to ground (e.g., R_{ex32}) only appear along the diagonal whereas a non-grounded (e.g. R_{ex31} , R_{ex33}) appears both on and off the diagonal. Rest of the terms (non-highlighted) of $[\mathbf{M}_{mna}]$ is the contributions of the selective currents (inductor currents, current through the voltage sources) and contain positive ones, negative one or zeros. And $[\mathbf{C}_{mna}]$ contains the the contributions of dynamic elements in the circuit and appear same way as $[\mathbf{M}_{mna}]$. Empty entries (i.e., zeros) in

the matrices indicate that the circuit element does not contribute to the related position. One advantage of MNA is that all the entries of both $[M_{mna}]$ and $[C_{mna}]$ matrices can be written by inspection which can easily be implemented algorithmically.

Appendix E

Example Circuit Netlists

The netlists of the circuits used in this research are provided here for future reference.

E.1 Single MESFET Amplifier

```
.options f0 = 5.1e9 method = 2 jupdm = 1
***Element Values***
inductor:l1 1 2 l=1e-9
capacitor:c1 2 3 c=20e-11
inductor:l2 3 7 l=15e-9
resistor:r2 7 8 r=100
***Nonlinear Element***
mesfetm:m1 3 4 123 idss = 0.06 vp0 = -1.906 gama = -0.015 e = 1.8
+ sl = 0.0676 kg = 1.1 t = 7.0e-12 ss = 1.666e-3 ig0 = 7.13e-6
+ afag = 38.46 r10 = 3.5 kr = 1.111 vbc = 12 ib0 = 7.13e-6 afab = 38.46
+ c10 = 0.42e-12 k1 = 1.282 cf0 = 0.02e-12 kf = 1.282
resistor:rs 123 0 r=1.144
inductor:l3 4 5 l=15e-9
resistor:r3 5 6 r=10
capacitor:cload 4 9 c=20e-12
resistor:rload 9 0 r=50.
***Sources***
vsource:vbias 8 0 vdc = -.4
```

```

vsource:vdrain 6 0 vdc = 3.
resistor:rin 11 1 r = 50
vsource:vs 11 0 f = f0 vac = 1.
***Analysis***
.wavetran tstop=1e-9 tstep=.005e-9 zref=200 tol=1e-8 out_steps=100 im=1
***Outputs***
.out plot term 4 vt term 123 vt sub in "vds.wave.node"
.end

```

E.2 X-band MMIC LNA

```

.options f0=10e9
***Models***
.model m_line1 tlinp4 (z0mag = 95.7 k=7.55 fscale=10e9 alpha=773 nsect=20 foft=10e9 tand=0.006) .model
m_line2 tlinp4 (z0mag= 81.9 k=7.73 fscale=10e9 alpha=78 nsect=20 foft=10e9 tand=0.006) .model m_line3
tlinp4 (z0mag = 76.2 k=7.82 fscale=10e9 alpha= 156 nsect=20 foft=10e9 tand=0.006)
c1 2 3 6e-12
tlinp4:t1 3 0 0 0 model="m_line1" length = 1194u
tlinp4:t2 3 0 4 0 model="m_line2" length = 183u
mesfetc:m1 42 51 62 A0=0.09910 A1=0.08541 A2=-0.02030 A3=-0.01543
+ BETA=0.01865 GAMA=0.8293
+ VDS0=6.494 VT0=-1.2 VBI=0.8 CGD0=3f CGS0=528.2f IS=3e-12 NR=1.2 T=1e-12 vbd=12
resistor:rg1 41 42 r=0.83
resistor:rd1 5 51 r=0.83
resistor:rs1 61 62 r=0.33
l:lg1 4 41 l=7e-12
l:ls1 6 61 l=11e-12
tlinp4:t3 6 0 8 0 model="m_line2" length=391u
tlinp4:t4 6 0 7 0 model="m_line2" length=401u
c:c_via4 7 0 c=17e-12
resistor:r_via4 7 0 r=6
c:c_via3 8 0 c=17e-12

```

```
tlinp4:t5 5 0 9 0 model="m_line2" length=102u
tlinp4:t6 9 0 10 0 model="m_line1" length=368u
resistor:r1 10 11 r=10.53
resistor:r2 11 12 r = 61.53
c:c_s6 11 0 c=17e-12
tlinp4:t7 9 0 13 0 model="m_line2" length=33u
c:c2 13 14 c=2e-12
tlinp4:t8 14 0 15 0 model="m_line1" length=705u
tlinp4:t9 14 0 0 0 model="m_line2" length=419u
tlinp4:t10 14 0 17 0 model="m_line2" length=58u
mesfetc:m2 172 192 182
+ A0=0.1321 A1=0.1085 A2=-0.04804 A3=-0.03821
+ BETA=0.03141 GAMA=0.7946 VDS0=5.892 VT0=-1.2 VBI=1.5
+ CGD0=4e-15 CGS0=695.2f
+ IS=4e-12 N=1.2 T=1e-12 vbd=12
resistor:rg2 171 172 r=0.63
resistor:rd2 191 192 r=0.63
resistor:rs2 181 182 r=0.25
l:lg2 17 171 l=16e-12
l:ld2 19 191 l=11e-12
l:ls2 18 181 l=11e-12
c:c_via8 18 0 c=17p
resistor:r_via8 18 0 r=8
tlinp4:t11 19 0 20 0 model="m_line1" length=138u
c:cfb 20 21 c=4.28e-12
resistor:rfb 21 22 r=237.4
l:lfb 22 15 l=1.268n int_res=9.55
tlinp4:t12 19 0 23 0 model="m_line1" length=313u
l:lp 23 24 l=1.268n int_res=1.55
resistor:rpads 24 29 r=24.93
```

```
vsourc:v2 29 0 vdc=6
c:c_vial2 24 0 c=17e-12
tlinp4:t13 19 0 25 0 model="m_line3" length=229u
c:cload 25 26 c=6e-12
resistor:r50 26 0 r=50
***Sources***
vsourc:vin 2 211 f=10e9 vac=0.1 phase=-90
vsourc:vin2 211 0 f=11e9 vac=0.001 phase=-90
vsourc:v1 12 0 vdc=6
***Analysis***
.wavetran tstop=4ns tstep=4ps im=1 zref=50 tol=1e-8 out_steps=100
***Outputs***
.out plot term 26 vt in "mmic.wavetran"
.end
```

E.3 Colpitts Oscillator

```
***Sources***
vpulse:vs 1 0 v1=0 v2=11 tr=.01e-6 tf=.01e-6 per=10e-3 pw=3e-3
***Elements***
resistor:rc 1 2 r=2.4e3
capacitor:c1 2 0 c=2e-12
inductor:l1 2 5 l=1e-6
capacitor:c2 5 0 c=2e-12
resistor:re 4 0 r=1.3e3
capacitor:ce 4 0 c=100e-12
bjtnpn:q1 2 3 4 0 bf=100 br=1 re=1 rc=1
resistor:r1 1 3 r=8e3
resistor:r2 3 0 r=2e3
capacitor:cc 5 3 c=400e-12
***Analysis***
```

```
.wavetran tstop=30ns tstep=4ps zref=50 tol=1e-8 out_steps=500
***Options***
.options ftol=1e-8
***outputs***
.out plot term 2 vt in "osc.wavetran"
.end
```

E.4 Soliton Line

```
***Options*** .options freq=9.GHz nonlin=4 rtol=1e-4 ftol=rtol maxit=100
***Analysis*** .wavetran tstop=.5e-9 tstep=.1ps im=1 savenode=0 zref=440 tol=1e-8 out_steps=500
***Sources***
vsource:l 201 0 vac = 14. vdc = -6. f = freq phase=90 tr=.1e-9
resistor:rs 201 202 r=50.
***Models*** .model carlos diode ( js=2.24e-12 alfa=21.13 e=10 ct0=1.32767e-15 r0=171.9
+ fi=1.27517 gama=0.810205 jb=1.e-5 vb=-16. )
***Transmission line parameters***
.model c_line tlinp4 ( z0mag=75.00 k=7 fscale=10.e9 alpha = 59.9
+ nsect = 20 foft=10e9)
***Diodes***
diode:d1 101 0 model = "carlos" area=271.64
diode:d2 102 0 model = "carlos" area=258.63
diode:d3 103 0 model = "carlos" area=246.24
diode:d4 104 0 model = "carlos" area=234.45
diode:d5 105 0 model = "carlos" area=223.21
diode:d6 106 0 model = "carlos" area=212.52
diode:d7 107 0 model = "carlos" area=202.34
diode:d8 108 0 model = "carlos" area=192.65
diode:d9 109 0 model = "carlos" area=183.42
diode:d10 110 0 model = "carlos" area=174.63
diode:d11 111 0 model = "carlos" area=166.27
```

diode:d12 112 0 model = "carlos" area=158.3
diode:d13 113 0 model = "carlos" area=150.72
diode:d14 114 0 model = "carlos" area=143.5
diode:d15 115 0 model = "carlos" area=136.63
diode:d16 116 0 model = "carlos" area=130.08
diode:d17 117 0 model = "carlos" area=123.85
diode:d18 118 0 model = "carlos" area=117.92
diode:d19 119 0 model = "carlos" area=112.27
diode:d20 120 0 model = "carlos" area=106.89
diode:d21 121 0 model = "carlos" area=101.77
diode:d22 122 0 model = "carlos" area=96.89
diode:d23 123 0 model = "carlos" area=92.25
diode:d24 124 0 model = "carlos" area=87.83
diode:d25 125 0 model = "carlos" area=83.63
diode:d26 126 0 model = "carlos" area=79.62
diode:d27 127 0 model = "carlos" area=75.81
diode:d28 128 0 model = "carlos" area=72.18
diode:d29 129 0 model = "carlos" area=68.72
diode:d30 130 0 model = "carlos" area=65.43
diode:d31 131 0 model = "carlos" area=62.29
diode:d32 132 0 model = "carlos" area=59.31
diode:d33 133 0 model = "carlos" area=56.47
diode:d34 134 0 model = "carlos" area=53.76
diode:d35 135 0 model = "carlos" area=51.19
diode:d36 136 0 model = "carlos" area=48.73
diode:d37 137 0 model = "carlos" area=46.4
diode:d38 138 0 model = "carlos" area=44.18
diode:d39 139 0 model = "carlos" area=42.06
diode:d40 140 0 model = "carlos" area=40.05
diode:d41 141 0 model = "carlos" area=38.13

```
diode:d42 142 0 model = "carlos" area=36.3
diode:d43 143 0 model = "carlos" area=34.56
diode:d44 144 0 model = "carlos" area=32.91
diode:d45 145 0 model = "carlos" area=31.33
diode:d46 146 0 model = "carlos" area=29.83
diode:d47 147 0 model = "carlos" area=28.4
***Parasitic inductors*** inductor:i1 1 101 l=21.8pH
inductor:i2 2 102 l=21.8pH
inductor:i3 3 103 l=21.8pH
inductor:i4 4 104 l=21.8pH
inductor:i5 5 105 l=21.8pH
inductor:i6 6 106 l=21.8pH
inductor:i7 7 107 l=21.8pH
inductor:i8 8 108 l=21.8pH
inductor:i9 9 109 l=21.8pH
inductor:i10 10 110 l=21.8pH
inductor:i11 11 111 l=21.8pH
inductor:i12 12 112 l=21.8pH
inductor:i13 13 113 l=21.8pH
inductor:i14 14 114 l=21.8pH
inductor:i15 15 115 l=21.8pH
inductor:i16 16 116 l=21.8pH
inductor:i17 17 117 l=21.8pH
inductor:i18 18 118 l=21.8pH
inductor:i19 19 119 l=21.8pH
inductor:i20 20 120 l=21.8pH
inductor:i21 21 121 l=21.8pH
inductor:i22 22 122 l=21.8pH
inductor:i23 23 123 l=21.8pH
inductor:i24 24 124 l=21.8pH
```

```
inductor:i25 25 125 l=21.8pH
inductor:i26 26 126 l=21.8pH
inductor:i27 27 127 l=21.8pH
inductor:i28 28 128 l=21.8pH
inductor:i29 29 129 l=21.8pH
inductor:i30 30 130 l=21.8pH
inductor:i31 31 131 l=21.8pH
inductor:i32 32 132 l=21.8pH
inductor:i33 33 133 l=21.8pH
inductor:i34 34 134 l=21.8pH
inductor:i35 35 135 l=21.8pH
inductor:i36 36 136 l=21.8pH
inductor:i37 37 137 l=21.8pH
inductor:i38 38 138 l=21.8pH
inductor:i39 39 139 l=21.8pH
inductor:i40 40 140 l=21.8pH
inductor:i41 41 141 l=21.8pH
inductor:i42 42 142 l=21.8pH
inductor:i43 43 143 l=21.8pH
inductor:i44 44 144 l=21.8pH
inductor:i45 45 145 l=21.8pH
inductor:i46 46 146 l=21.8pH
inductor:i47 47 147 l=21.8pH
***Transmission lines** tlinp4:t0 202 0 1 0 model = "c_line" length=501.29u
tlinp4:t1 1 0 2 0 model = "c_line" length=978.57u
tlinp4:t2 2 0 3 0 model = "c_line" length=931.69u
tlinp4:t3 3 0 4 0 model = "c_line" length=887.06u
tlinp4:t4 4 0 5 0 model = "c_line" length=844.57u
tlinp4:t5 5 0 6 0 model = "c_line" length=804.11u
tlinp4:t6 6 0 7 0 model = "c_line" length=765.59u
```

tlinp4:t7 7 0 8 0 model = "c_line" length=728.92u
tlinp4:t8 8 0 9 0 model = "c_line" length=694.00u
tlinp4:t9 9 0 10 0 model = "c_line" length=660.75u
tlinp4:t10 10 0 11 0 model = "c_line" length=629.10u
tlinp4:t11 11 0 12 0 model = "c_line" length=598.97u
tlinp4:t12 12 0 13 0 model = "c_line" length=570.27u
tlinp4:t13 13 0 14 0 model = "c_line" length=542.96u
tlinp4:t14 14 0 15 0 model = "c_line" length=516.95u
tlinp4:t15 15 0 16 0 model = "c_line" length=492.18u
tlinp4:t16 16 0 17 0 model = "c_line" length=468.61u
tlinp4:t17 17 0 18 0 model = "c_line" length=446.16u
tlinp4:t18 18 0 19 0 model = "c_line" length=424.79u
tlinp4:t19 19 0 20 0 model = "c_line" length=404.44u
tlinp4:t20 20 0 21 0 model = "c_line" length=385.06u
tlinp4:t21 21 0 22 0 model = "c_line" length=366.62u
tlinp4:t22 22 0 23 0 model = "c_line" length=349.05u
tlinp4:t23 23 0 24 0 model = "c_line" length=332.33u
tlinp4:t24 24 0 25 0 model = "c_line" length=316.41u
tlinp4:t25 25 0 26 0 model = "c_line" length=301.26u
tlinp4:t26 26 0 27 0 model = "c_line" length=286.83u
tlinp4:t27 27 0 28 0 model = "c_line" length=273.09u
tlinp4:t28 28 0 29 0 model = "c_line" length=260.00u
tlinp4:t29 29 0 30 0 model = "c_line" length=247.55u
tlinp4:t30 30 0 31 0 model = "c_line" length=235.69u
tlinp4:t31 31 0 32 0 model = "c_line" length=224.40u
tlinp4:t32 32 0 33 0 model = "c_line" length=213.65u
tlinp4:t33 33 0 34 0 model = "c_line" length=203.42u
tlinp4:t34 34 0 35 0 model = "c_line" length=193.67u
tlinp4:t35 35 0 36 0 model = "c_line" length=184.39u
tlinp4:t36 36 0 37 0 model = "c_line" length=175.56u

```
tlinp4:t37 37 0 38 0 model = "c_line" length=167.15u
tlinp4:t38 38 0 39 0 model = "c_line" length=159.14u
tlinp4:t39 39 0 40 0 model = "c_line" length=151.52u
tlinp4:t40 40 0 41 0 model = "c_line" length=144.26u
tlinp4:t41 41 0 42 0 model = "c_line" length=137.35u
tlinp4:t42 42 0 43 0 model = "c_line" length=130.77u
tlinp4:t43 43 0 44 0 model = "c_line" length=124.51u
tlinp4:t44 44 0 45 0 model = "c_line" length=118.54u
tlinp4:t45 45 0 46 0 model = "c_line" length=112.86u
tlinp4:t46 46 0 47 0 model = "c_line" length=107.46u
tlinp4:t47 47 0 48 0 model = "c_line" length=52.41u
resistor:rl 48 0 r=50.
***Outputs***
.out plot element "diode:d47" 0 ut in "sol.wavetran"
.end
```

E.5 Summing Amplifier

```
***Input***
vsource:v2 153 0 f=1e5 vac=1 delay=10e-6 phase=0
resistor:r2s 153 154 r=10.
vsource:v1 151 0 f=1e4 vac=0.1 delay=10e-6 phase=0
resistor:r1s 151 152 r=10.
resistor:r1ext 152 202 r=5e3
resistor:r2ext 154 202 r=20e3
resistor:r4ext 202 23 r=20e3
resistor:r5ext 201 0 r=3.33e3
***Power Supply****
***VCC PIN 1
***VEE PIN 2
vsource:vcc 100 0 vdc=7.5 tr=10e-6
```

```
resistor:rcc 1 100 r=20
vsource:vee 0 101 vdc=7.5 tr=10e-6
resistor:ree 101 2 r=20
***Widlar Current Source***
bjtnpn:q10 5 4 3 2 model="MYNPN" area=1
bjtnpn:q11 4 4 2 2 model="MYNPN" area=1
resistor:r4 3 2 r=5e3
resistor:r5 6 4 r=39e3
bjtnpn:q9 5 7 1 1 model="MYPNP" area=1
bjtnpn:q12 6 6 1 1 model="MYPNP" area=1
bjtnpn:q13b 15 6 1 1 model="MYPNP" area=3
bjtnpn:q13a 16 6 1 1 model="MYPNP" area=1
bjtnpn:q8 7 7 1 1 model="MYPNP" area=1
*Vplus → PIN201
*Vminus → PIN202
*Offset null1 → PIN14
*Offset null2 → PIN17
***Differential Amplifier***
bjtnpn:q1 7 201 9 2 model="MYNPN" area=1
bjtnpn:q2 7 202 10 2 model="MYNPN" area=1
bjtnpn:q3 11 5 9 1 model="MYPNP" area=1
bjtnpn:q4 12 5 10 1 model="MYPNP" area=1
bjtnpn:q5 11 13 14 2 model="MYNPN" area=1
bjtnpn:q6 12 13 17 2 model="MYNPN" area=1
bjtnpn:q7 1 11 13 2 model="MYNPN" area=1
resistor:r1 14 2 r=1e3
resistor:r2 17 2 r=1e3
resistor:r3 13 2 r=50e3
bjtnpn:q19 16 16 20 2 model="MYNPN" area=1
bjtnpn:q18 16 20 21 2 model="MYNPN" area=1
```

```
resistor:r10 20 21 r=40e3
capacitor:cc 15 12 c=30e-12
bjtnpn:q16 1 12 18 2 model = "MYNPN" area=1
bjtnpn:q17 15 18 19 2 model = "MYNPN" area=1
bjtnpn:q22 12 25 2 2 model = "MYNPN" area=1
resistor:r9 18 2 r=50e3
resistor:r8 19 2 r=100
bjtpnp:q23b 2 15 12 1 model = "MYPNP" area=1
bjtpnp:q23a 2 15 21 1 model = "MYPNP" area=1
***Output***
*Output → PIN23
bjtnpn:q14 1 16 22 2 model = "MYNPN" area=3
bjtnpn:q15 16 22 23 2 model = "MYNPN" area=1
resistor:r6 22 23 r=27
bjtpnp:q21 25 24 23 1 model = "MYPNP" area=1
resistor:r7 23 24 r=22
bjtpnp:q20 2 21 24 1 area=1 IS=1E-14 RB=150 RC=50 RE=2
+BR=4 BF=50 VAF=50 TF=20E-9 CJE=0.5E-12 CJC=2E-12
bjtnpn:q24 25 25 2 2 model = "MYNPN" area=1
resistor:r12 25 2 r=50e3
***Dummy Resistors
resistor:rdummy1 4 0 r=20e6
resistor:rdummy2 5 0 r=20e6
resistor:rdummy3 6 0 r=20e6
resistor:rdummy4 7 0 r=20e6
resistor:rdummy5 9 0 r=20e6
resistor:rdummy6 10 0 r=20e6
resistor:rdummy7 11 0 r=20e6
resistor:rdummy8 3 0 r=20e6
resistor:rdummy9 12 0 r=20e6
```

```
resistor:rdummy13 15 0 r=20e6
resistor:rdummy14 16 0 r=20e6
resistor:rdummy18 21 0 r=20e6
resistor:rdummy21 24 0 r=20e6
***Model*** .model MYNPN bjtnpn(IS=5E-15 RB=200 RC=200 BF=200
+BR=2 RE=2 VAF=130 TF=0.35E-9 CJE=1E-12 CJC=0.3E-12
+NE=0.33 NC=0.5 TR=400E-9 VJE=0.7 VJC=0.55)
.model MYPNP bjtpnp(IS=2E-15 RB=300 RC=100 RE=10
+BF=50 BR=4 VAF=50 TF=30E-9 CJE=0.3E-12 CJC=1E-12
+NE=0.5 NC=0.5 TR=3000E-9 VJE=0.55 VJC=0.55)
***Analysis***
.wavetran tstop=.4e-4 tstep=.005e-6 zref=2e3 tol=1e-8 out_steps=1 im=0
***Output plot*** .out plot term 23 vt in "opamp.wavetran"
.end
```

Bibliography

- [1] A. Fettweis, "Wave Digital Filters: Theory and Practice," *IEEE Proc.*, vol. 74, pp. 270–327, 1986.
- [2] A. Fettweis, "The role of passivity and losslessness in multidimensional digital signal processing - new challenges," *IEEE Proc. ISCAS '91*, vol. 1 pp. 112–115, 1991.
- [3] A. R. Newton and A. L. Sangiovanni-Vincentelli, "Relaxation-based electrical simulation," *IEEE Trans. on Electron Devices*, vol. 30, issue-9, pp. 1184–1207, 1983.
- [4] A. Fiedler and H. Grotstollen, "Simulation of power electronic circuits with principles used in wave digital filters," *IEEE Trans. on Industry Applications*, vol. 33, issue-1, pp. 49-57, 1997.
- [5] A. Sarti and G. De Poli, "Towards nonlinear wave digital filters," *IEEE Trans. on Signal Processing*, vol. 47, issue-6, pp. 1654–1668, 1999.
- [6] Alan J. Laub, *Matrix analysis for scientists and engineers*, 2005, The Society for Industrial and Applied Mathematics.
- [7] C. Fröberg, *Numerical Mathematics, Theory and Computer Applications*, 1985, The Benjamin/Cummings Publishing Company.
- [8] C. E. Christoffersen, "Global Modeling of Nonlinear Microwave Circuits," Ph.D. Dissertation, Dept. of EE, North Carolina State University, 2000.
- [9] C. Haiping, H. Taiping, "The Auto-adjustable Damping Method For Solving Nonlinear Equations," *Applied Mathematics and Mechanics, English Edition*, vol. 19, issue-2, 1998.

-
- [10] C. E. Christoffersen, M. Ozkar, M. B. Steer, M. G. Case and M. Rodwell, "State variable-based transient analysis using convolution," *IEEE Trans. on Microwave Theory and Tech.*, vol. 47, issue-6, pp. 882–889, 1999.
- [11] C. S. Koh, J. S. Ryu, K. Fujiwara, "Convergence acceleration of the Newton-Raphson method using successive quadratic function approximation of residual," *IEEE Trans. on Magnetics*, vol. 42, issue-4, pp. 611-614, 2006.
- [12] C. E. Christoffersen, "Transient Analysis of Nonlinear Circuits Based on Waves," *Proc. of the 7th Int. Conf. on Scientific Computing in Electrical Engineering, SCEE '08*, Helsinki Institute of Technology, Finland, 2008.
- [13] Compact Software, *Microwave Harmonica Elements Library*, 1994.
- [14] Calyampudi Radhakrishna Rao, Sujit Kumar Mitra, *Generalized inverse of matrices and its applications*, 1971, Wiley.
- [15] D. Hill, "On the Stability of Nonlinear Networks" *IEEE Trans. on Circuits and Systems*, vol. 25, issue-11, pp. 941-943, 1978.
- [16] D. A. Smith, W. F. Ford, A. Sidi, "Extrapolation methods for vector sequences," *SIAM Review*, vol. 29, issue-2, pp. 199-233, 1987.
- [17] E. I. Jury, *Theory and Application of the z-Transform Method*, 1964, Wiley.
- [18] E. Lelarasme, "The waveform relaxation method for the time-domain analysis of large scale integrated circuits: Theory and applications", Ph.D. dissertation, Dept. of EECS, Univ. of California, Berkeley, 1982.
- [19] E. Lelarasme, A. E. Ruehli and A. L. Sangiovanni-Vincentelli, "The waveform relaxation method for time domain analysis of large scale integrated circuits," *IEEE Trans. on CAD of Integrated Circuits and Systems.*, vol. 1, issue-3, pp. 131-145, 1992.
- [20] F. P. Hart, N. Kriplani, S. Luniya, C. E. Christoffersen and M. B. Steer, "Streamlined Circuit Device Model Development with Freeda and ADOL-C," *4th Int. Conf. on Automatic Differentiation*, Chicago, USA, 2004.

- [21] F. R. Gantmacher, *The theory of matrices, Volume 1*, 2000, American Mathematical Society.
- [22] G. Borin, G. De Poli, D. Rocchesso, "Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems," *IEEE Trans. on Speech And Audio Processing*, vol. 8, issue-5, pp. 597-605, 2000.
- [23] H. Shi, C. W. Domier and N. C. Luhmann, "A monolithic nonlinear transmission line system for the experimental study of lattice solutions," *Jou. of Applied Physics*, vol. 4, pp. 2558-2564, 1995.
- [24] H. Nagashima and Y. Amagish, "Experiment on the Toda lattice using nonlinear transmission lines," *Jou. of the Physical Society*, vol. 45, pp. 680-688, 1978.
- [25] I. Gohberg, P. Lancaster, L. Rodman, *Matrix Polynomials*, 2009, The Society for Industrial and Applied Mathematics.
- [26] IBM., "Advanced Statistical Analysis Program (ASTAP)," Program Reference Manual, Technical Report SH20-1118-0, IBM, 1973.
- [27] J. M. Bahi, S. Contassot-Vivier, R. Couturier, *Parallel iterative algorithms: from sequential to grid computing*, 2008, Chapman & Hall/ CRC.
- [28] J. K. White, *Relaxation techniques for the simulation of VLSI circuits*, 1987, Kluwer Academic Publishers.
- [29] J. Wyatt, L. Chua, J. Gannett, I. Goknar, D. Green, "Energy concepts in the state-space theory of nonlinear n-ports: Part I-Passivity," *IEEE Trans. on Circuits and Systems*, vol. 28, issue-1, pp. 48-61, 1981.
- [30] K. Meerkötter and R. Scholz, "Digital simulation of nonlinear circuits by wave digital filter principles," *IEEE Proc. ISCAS '89*, pp. 720-723, 1989.
- [31] K. Meerkötter and T. Felderhoff, "Simulation of nonlinear transmission lines by wave digital filter principles," *IEEE Proc. ISCAS '92*, vol. 2, pp. 875-878, 1992.
- [32] K. Muroya and S. Watanabe, "Experiment on lattice soliton by nonlinear LC circuit - Energy of soliton" *Jou. of the Physical Society*, vol. 50, pp. 2762-2769, 1981.

- [33] K. Gulati, J.F. Croix, S.P. Khatri, R. Shastry, "Fast circuit simulation on graphics processing units," *Design Automation Conf. Asia and South Pacific*, pp. 403-408, 2009.
- [34] L. Zhu and C. E. Christoffersen, "Transient and Steady-State Analysis of Nonlinear RF and Microwave Circuits," *EURASIP Jou. on Wireless Communications and Networking*, vol. 2006, issue-CMOS RF Circuits for Wireless Applications, pp. 1-11, 2006.
- [35] LAMMDA, "Notes on MMIC design, LAMMDA", University of Massachusetts Amherst, MA, USA.
- [36] Louis A. Hageman, David M. Young, *Applied Iterative Methods*, 1981, New York: Academic Press.
- [37] M. G. Case, "Nonlinear transmission lines for picosecond pulse, impulse and millimeter-wave harmonic generation", Ph.D Dissertation, Dept. of ECE, University of California, Santa Barbara, 1993.
- [38] M. B. Steer and C. E. Christoffersen, "Generalized circuit formulation for the transient simulation of circuits using wavelet, convolution and time-marching techniques," *Proc. of the 15th European Conf. on Circuit Theory and Design*, pp. 205-208, 2001.
- [39] M. Toda, "Nonlinear lattice and soliton theory" *IEEE Trans. on Circuits and Systems*, vol. 30, issue-8, pp. 542-554, 1983.
- [40] 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 Tech.*, vol. 39, issue-7, pp. 1194-1204, 1991.
- [41] N. Balabanian and T. Bickart, *Linear Network Theory: Analysis, Properties, Design and Synthesis*, 1981, Matrix Publishers.
- [42] Oapos, J. Dwyer, T. Donnell, "Choosing the relaxation parameter for the solution of nonlinear magnetic field problems by the Newton-Raphson method," *IEEE Trans. on Magnetics*, vol. 31, issue-3, pp. 1484-1487, 1995.
- [43] P. J. C. Rodrigues, *Computer-Aided Analysis of Nonlinear Microwave Circuits*, 1998, Artech House Pub.

- [44] P. Moylan, "Implications of Passivity in a Class of Nonlinear Systems" *IEEE Trans. on Automatic Control*, vol. 19, issue-4, pp. 373-381, 1974.
- [45] R. E. Poore, Agilent Technol., "GPU-accelerated time-domain circuit simulation," *IEEE Custom Integrated Circuits Conf. CICC '09*, pp. 629-632, 2009.
- [46] Richard S. Varga, *Matrix Iterative Analysis*, 2009, Springer Series in Computational Mathematics.
- [47] R. R. Spencer and M. S. Ghausi, *Introduction to Electronic Circuit Design*, 2003, Prentice Hall.
- [48] R. Hirota and K. Suzuki, "Theoretical and experimental studies of lattice solitons in nonlinear lumped networks," *IEEE Proc.*, vol. 61, pp. 1483-1491, 1973.
- [49] R. Hirota and K. Suzuki, "Studies on lattice solitons by using electrical networks," *Jou. of the Physical Society*, vol. 28, pp. 1366-1367, 1970.
- [50] S. A. Maas, *Nonlinear Microwave and RF Circuits, Second edition*, 2003, Artech House Pub.
- [51] S. Luniya, W. Batty, V. Caccamesi, M. Garcia, C. E. Christoffersen, S. Melamed, W. R. Davis and M. Steer, "Compact Electrothermal Modeling of an X-band MMIC," *IEEE Int. Microwave Symp. Digest*, pp. 651-654, 2006.
- [52] Stefan Bilbao and Julius O. Smith, "Lecture on Wave Digital Filter," *Center for Computer Research in Music and Acoustics, CCRMA*, Department of Music, Stanford University, 2007.
- [53] S. Petrausch and R. Rabenstein, "Wave digital Filters with multiple nonlinearities," *Proc. of 12th European Signal Processing Conf. '04*, 2004.
- [54] T. Grasser, "Mixed-Mode Device Simulation," PhD Dissertation, Ins. for Microelec., Vienna University of Technology, 1999.
- [55] T. Felderhoff, "Jacobi's method for massive parallel wave digital filter algorithm," *IEEE Proc. Conf. on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1621-1624, 1996.

- [56] U. Miekkala and O. Nevanlinna, "Convergence of dynamic iteration methods for initial value problems," *SIAM Jou. on Scientific and Statistical Computing.*, vol. 8, issue-4, pp. 459-482, 1987.
- [57] 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, issue-1, pp. 12-28, 1992.
- [58] W Batty, C. E. Christoffersen, A. J. Panks, S. David, C. M. Snowden and M. B. Steer, "Electro-Thermal CAD of Power Devices and Circuits with Fully Physical Time-Dependent Compact Thermal Modelling of Complex Non Linear 3-D Systems," *IEEE Trans. on Components and Packaging Technology*, vol. 34, issue-4, pp. 566-590, 2001.
- [59] W. Weeks, A. Jimenez, G. Mahoney, D. Mehta, H. Qassemzadeh, T. Scott, "Algorithms for ASTAPA network-analysis program," *IEEE Trans. on Circuit Theory*, vol 20, issue-6, pp. 628-634, 1973.
- [60] <http://en.wikipedia.org/wiki/FLOPS>