

ABSTRACT

Senthil N Velu. Charge Based Modeling in State Variable Based Simulator. (Under the direction of Dr. Michael B. Steer.)

A new parameterized nonlinear device model formulation is described that enables the same computer code to be used in any circuit analysis (ie. Harmonic Balance, Transient and D.C) routine with no charge conservation issues. The parametric description provides great flexibility for the design of nonlinear device models. The number of parameters or state variables required is the minimum number necessary and can be chosen to achieve robust numerical characteristics. An example illustrates charge conservation problems that can occur in the transient simulation of microwave circuits if the models are not correctly formulated. Implementation of the BJT and MESFET in *fREEDA*TM using both the Universal Modeling algorithm (Charge as state variable) and the conventional algorithm (Voltage as state variable) is described. The above concept allows the modeling of thermal effects to be included in the simulations of electronic circuits, by viewing thermal sub-systems as sub-circuits. All these developments are implemented in a circuit simulator program, called *fREEDA*TM. This program provides unprecedented flexibility for the addition of new device models or circuit analysis algorithms. *fREEDA*TM was applied to the two tone harmonic balance simulation of a MESFET amplifier. Distortion in MESFET models was studied and the importance of the time delay parameter τ is discussed with the aid of IM3 plots.

CHARGE BASED MODELING IN STATE VARIABLE BASED SIMULATOR

by

Senthil N. Velu

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Electrical Engineering

Raleigh

October 2002

APPROVED BY:

Chair of Advisory Committee

BIOGRAPHY

Senthil N. Velu was born in Chennai, India in 1976. He received a degree in Electrical and Electronics engineering in 1998 from Madras University, India. From November 1997 to April 1999 he worked as a research student at the Suzuki manufacturing plant in Chennai, India. His work at Suzuki included design and programming of Computer Numerically Controlled (CNC) machines. From May 1998 to January 2002 he worked as a team leader at Complete Business Solutions Inc. in Chennai, India. From February 2002 to July 2002 he worked as a CNC maintenance engineer at P.T.Warewell in Phnom Penh, Cambodia. Since August 2000, he has been with the Department of Electrical and Computer Engineering at North Carolina State University as a research assistant. His research interests include computer aided analysis of circuits and analog, RF and microwave circuit design.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor Dr. Michael Steer for his support and guidance during my graduate studies. It was a privilege to be a member of his simulator research group. I wish to express my sincere gratitude to Dr. Griff Bilbro and Dr. Mo-Yuen Chow for their interest in my research and for serving on my thesis committee.

I would also like to thank my past and present graduate student colleagues. Special thanks to Dr. Carlos Christoffersen. On a personal note I would like to thank my family who have been my never-ending source of inspiration and support.

Contents

List of Figures	vi
List of Tables	viii
1 Introduction	1
1.1 Motivation	1
1.2 Thesis Overview	3
1.3 Original Contributions	4
2 Literature Review	5
2.1 Introduction	5
2.2 Parameterized Device Modeling	5
2.3 Universal Device Modeling	9
2.4 Object Oriented Circuit Simulators	10
2.5 Freeda	11
2.5.1 Introduction	11
2.5.2 Support Libraries	12
2.5.3 Automatic Differentiation	13
2.6 Electro-Thermal Modeling	14
3 Universal Modeling	16
3.1 Introduction	16
3.2 Nonlinear Device Models and Circuits	18
3.3 Parameterized Device Models	19
3.4 Universal Model Formulation	22
3.4.1 Example: Bipolar Junction Transistor	23
3.4.2 Bipolar transistor Equations	24
3.5 Bipolar Transistor: Voltage Based State Variable	26
3.6 Bipolar Transistor: Universal Model Formulation	30
3.7 Mesfet Model : Curtice-Ettenberg	32
3.8 Universal Nonlinear Error Function Formulation	35
3.8.1 Linear Network	35

3.8.2	Nonlinear Network	36
3.8.3	Error Function Formulation	36
3.8.4	Time Domain Jacobian	37
3.8.5	Harmonic Balance Equations	38
3.9	Electro-Thermal Modeling	39
4	Results	42
4.1	Introduction	42
4.2	Varactor Non-Conservation Of Charge	43
4.3	Universal Modeling	44
4.3.1	Bipolar Junction Transistor	44
4.3.2	MESFET	49
4.4	Two Tone Analysis	49
4.5	Electrothermal Simulation	54
5	Conclusions and Future Research	58
5.1	Summary Of Research	58
5.2	Future Research	59
	Bibliography	61
	A Bipolar Junction Transistor Model Parameters	65
	B MESFET Model Parameters	69
	C Freeda Circuit Netlist	71
C.1	Bjt Universal Modeling	71
C.1.1	Transient Analysis	71
C.1.2	Harmonic Balance Analysis	72
C.1.3	Bjt DC Analysis	73
C.2	Bjt Thermal Netlist	73
C.3	BJT Wide Band Amplifier	74
C.4	MESFET Universal Modeling	75
C.4.1	Harmonic Balance Analysis	75
C.4.2	Convolution Transient Analysis	76
C.4.3	Transient Analysis	77
C.5	Mesfet Multi-Tone Test	78

List of Figures

1.1	Topologies that may present charge conservation problems in microwave circuits.	2
2.1	Relation between v and i in a diode.	7
2.2	Relation between x and i in a diode.	7
2.3	Relation between x and v in a diode.	8
2.4	Equivalent Diode model in Spice.	9
3.1	Example: partition representation of the time-invariant harmonic balance method.	19
3.2	Example: schematic of the simplified Diode Model.	20
3.3	Bipolar junction transistor model	24
3.4	Mesfet Curtice Ettenberg Model.	32
3.5	Network with nonlinear elements	35
3.6	Nonlinear electro-thermal device	40
4.1	Varactor Circuit	43
4.2	Comparison of the simulation using the capacitor and charge based models.	44
4.3	Input waveform.	45
4.4	Wideband amplifier output in <i>fREEDA</i> TM	45
4.5	Wideband amplifier output in Spice.	46
4.6	BJT amplifier test circuit.	47
4.7	BJT universal model output.	47
4.8	BJT common-emitter circuit to test the DC characteristics.	48
4.9	BJT I-V curves generated at different values of bias current.	48
4.10	MESFET circuit used for Universal Simulation.	50
4.11	Universal Modeling output.	50
4.12	MESFET amplifier circuit used to simulate the two-tone test.	51
4.13	Results of the two-tone test. A plot of output power at the fundamental frequency as a function of input power.	52

4.14	A plot of the output power at the third harmonic as a function of the input power.	53
4.15	A plot of the output power at the third harmonic as a function of the input power with $\tau = 6.56$ ps and $\tau = 0$	54
4.16	Temperature variation in the thermal BJT.	55
4.17	Power dissipation in the thermal BJT.	55
4.18	Electro-thermal BJT amplifier circuit.	56
4.19	Power dissipation in the thermal BJT amplifier.	56
4.20	Temperature variation in the amplifier circuit.	57
5.1	Layout of the <i>LMA411</i> MMIC amplifier.	60
5.2	Layout of the multi layer spiral inductor.	60

List of Tables

3.1	MESFET state variables	34
3.2	MESFET model output variables	34
4.1	Comparison of the numerical error due to non-conservation of charge as a function of the time step.	43
A.1	Parameters of the Bipolar Junction Transistor.	65
A.2	Continued.	66
A.3	Continued.	67
A.4	Continued.	68
B.1	Parameters of the MESFET model.	69
B.2	Continued.	70

Chapter 1

Introduction

1.1 Motivation

Traditionally device model formulation and circuit simulator technology are intricately related with the consequence that for each circuit analysis a particular device formulation is required. Thus, one formulation of a transistor model is required for transient analysis and another for harmonic balance. This is required in part because of the need for derivatives in the analysis iteration algorithm but also because of peculiarities related to the choice of state variables and local convergence control. The major contribution of this thesis is to present a universal model technology that enables the same model (*i.e.* computer code) to be used with any analysis type; has global convergence properties; thus avoiding the need for local convergence control; and enables physically realistic choice of state variables so that model development can proceed smoothly without the need to use what can be construed as artificial voltage-like or current-like quantities. The use of automatic differentiation also avoids the need to perform derivative evaluations with the device model code dramatically reducing the amount of code needed (typically a factor of 10 reduction is achieved compared to the normal modeling procedure). Object-oriented design practices further extend the functionality of device models [9, 3].

Parameterized models [22] can be used to allow the modeling of nonlinear devices in different analysis types using the same implementation of the device's equations

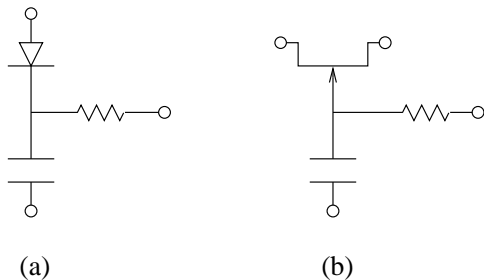


Figure 1.1: Topologies that may present charge conservation problems in microwave circuits.

(generic evaluation, [4]). In this way the number of nonlinear state variables is kept reduced to the minimum number necessary, the models can be formulated to avoid positive exponential dependencies, and the resulting code can be developed faster and maintained more easily because the equations must be coded only once.

Modern transient circuit simulators use charge or flux as the state variables of capacitors or inductors to avoid stability and accuracy problems in transient analysis [5]. This type of problems have rarely been reported for microwave circuits. When these state variables are not used, accuracy problems occur when there is a series connection of capacitors and at least one of them is nonlinear [5]. For example, some designs of distributed amplifiers, voltage-controlled oscillators and phase shifters [21] present a series connection of a Schottky junction with a linear capacitor as shown in Fig. 1.1.

It is shown that in the original parametric formulation it is not always possible to write a charge-conserving model with the minimum number of state variables [4]. Further, the necessary modifications to the formulation to obtain a charge-conserving model with the minimum number of state variables and flexible parameterization is presented. The derivation of the Jacobian of the element in the time domain is shown. The formulation is developed for a bipolar junction transistor (BJT) using the Gummel-Poon model and a MESFET using both the Curtice-Ettenberg model and Materka-Kacrapaz model [24]. The numerical error in transient analysis that results due to a model not based on charge is illustrated with a microwave circuit example. Joint computer simulation of circuits and thermal interactions in high performance

integrated circuits is of particular importance for the modeling of electronic packaging since these are complex coupled systems where all these aspects interact in a dynamic sense.

The implementation of thermal modeling in *fREEDA*TM and the simultaneous thermal-electrical simulation of transistor amplifier circuits as an example are shown. One way of incorporating thermal effects in a circuit simulator is to make the thermal model look like an electrical circuit. The thermal and electrical circuits are then solved simultaneously as if they were one large electrical circuit. Power dissipated in the active devices is represented as a heat current source referenced to thermal ground. One problem with this strategy is provision of separate circuits for the electrical and thermal parts. This has been addressed by the concept of local reference nodes. The use of local reference nodes guarantees that there is no mixing of electric and thermal currents.

1.2 Thesis Overview

Chapter 2 describes the existing trends in device modeling and simulation incorporated by widely used circuit simulator (*i.e. Spice2, AplaC, ... , etc.*). Chapter 2 is concluded with an introduction to *fREEDA*TM, a circuit simulator developed at North Carolina State University. A study of the various off-the-shelf libraries (*i.e. ADOL-C and NNES*) used to implement a universal modeling algorithm.

Chapter 3 describes the universal modeling algorithm. Implementation of the algorithm in a state-variable-based simulator (*fREEDA*TM) is described with the assistance of two transistors (BJT and MESFET). Chapter 3 concludes with an introduction to the implementation of electro-thermal models in *fREEDA*TM.

Chapter 4 presents results obtained by using the same model computer code for different analysis types (*i.e. D.C., Transient and Harmonic Balance*) are shown. The results obtained by simulating an electro-thermal bipolar junction transistor model are explained and the results are compared with the static temperature models used in Spice. Results obtained for a two-tone analysis of microwave devices is described with the aid of two tone measurements and simulations.

Chapter 5 summarizes the work presented in this thesis. It outlines requirements for future research involving extension of the universal algorithm to model electro-thermal elements and EM structures. Future plans of modeling Monolithic Microwave Integrated Circuits (MMIC) and model validation is outlined.

1.3 Original Contributions

The implementation of the BJT in *fREEDA*TM required calculation of second order derivative of state variables. This was accomplished by changing the source code of ADOL-C. The algorithm to calculate the second order derivative is shown in Section 3.5. The original ADOL-C program was modified to facilitate the implementation of the universal model formulation algorithm. The modifications involve evaluation of state variable functions at various stages of the model formulation as shown in Section 3.6.

Another original contribution was the implementation of the BJT (Gummel Poon) device in *fREEDA*TM using both the voltage based state variable routine and the charge based universal formulation method, shown in Sections 3.5 and 3.6. The MESFET device (Cutice Ettenberg and Materka Kacprazak) models were implemented and validated using the charge based algorithm. Two tone simulations using the MESFET model were validated against measure data [29] as shown in Section 4.4. Another contribution was the implementation of the electro-thermal BJT device in *fREEDA*TM, shown in Section 4.5.

Chapter 2

Literature Review

2.1 Introduction

This chapter outlines the work related to universal modeling and charge conservation. This presents a base for the original contributions presented in this thesis. Section 2.2 describes the efforts taken in the past to develop parameterized device models for circuit simulators. Section 2.3 details the need for a universal modeling algorithm and the drawbacks in some simulators. Section 2.4 describes circuit simulators written in an object-oriented circuit simulator. Section 2.5 gives an outline of *fREEDA*TM and the external packages used in the simulator. Section 2.6 extends the goal of universal modeling to include the simultaneous simulation of electrical and thermal characteristics in a circuit simulator.

2.2 Parameterized Device Modeling

The current circuit simulation architecture was largely established in the late 1960's leading up to the development of ASTAP at IBM and the Spice program at the University of California Berkeley. A prime aspect of this architecture is the integrated time-discretization of element characteristics along with a Newton iteration rendering circuit modeling as the iterative solution of a resistive companion model of the circuit. A key feature of this approach is the implementation of heuristically-based local con-

vergence control for elements with strongly nonlinear characteristics. This leads to a generally robust circuit simulation strategy, but one based only on Newton's method and not amenable to incorporating advances in numerical analysis techniques. An example of local convergence control is the way in which the exponential current-voltage characteristic of a diode is handled. Within the diode model code, the voltage and current updates are limited to relatively small changes. As mentioned, the requirements of local convergence control renders it not possible to use globally convergent schemes. Furthermore the use of voltage and current controlled primitives imposes severe restrictions on the types of devices that can be modeled. In *fREEDA*TM, the embodiment of a global simulation architecture, parameterized device models are used as shown in Fig. 2.1 for a diode. By converting the strong exponential current-voltage nonlinearity to two smoother functions of current and voltage as functions of a state-variable x the nonlinear problem becomes much better behaved [3]. Thus the unknowns in a simulation become the state-variables x and advanced off-the-shelf numerical methods can be used as network equation formulation is done at the top circuit level. Parameterization and the use of state variables increases the model development flexibility for modeling electro-opto-mechanical systems where the state variables can be chosen to achieve robust numerical characteristics. Furthermore *fREEDA*TM uses an object-oriented design paradigm, which dramatically reduces the amount of code required to implement a model.

The modeling of a mixed system is facilitated by using an error function concept that applies to all physical systems. This is a combination of a sameness error condition (for a circuit this becomes equivalent to requiring that at each terminal in the circuit the voltage calculated for each element should be the same), and a sum-to-zero condition (which, for a circuit, becomes the sum of currents entering a terminal is zero). In an electrical circuit this increases robustness in simulating situations of near zero current as occurs frequently in CMOS circuits. Thus we have a general unifying concept of fluxes and potentials that leads to the universal error formulation: the sum of fluxes at a terminal must be zero and all potentials calculated for a particular terminal must be the same. This analogy is used to model the thermal and mechanical systems where the potentials are temperature and position, respectively [18].

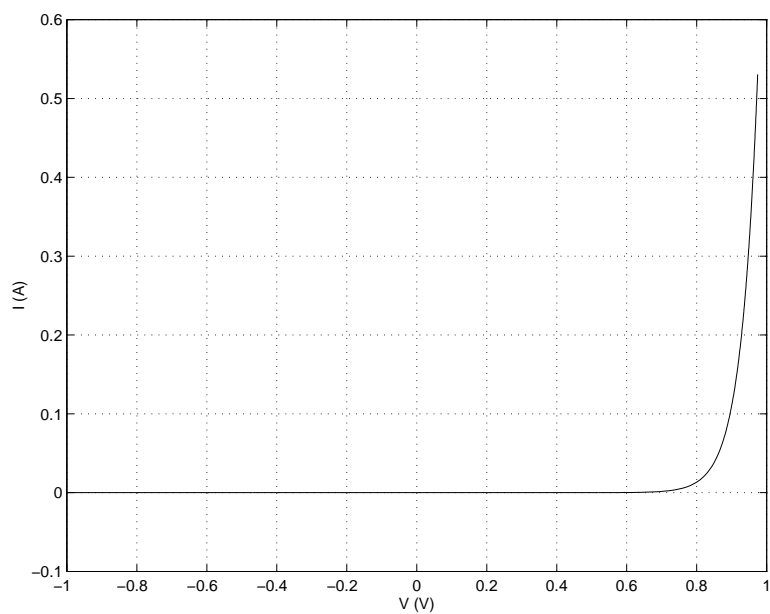


Figure 2.1: Relation between v and i in a diode.

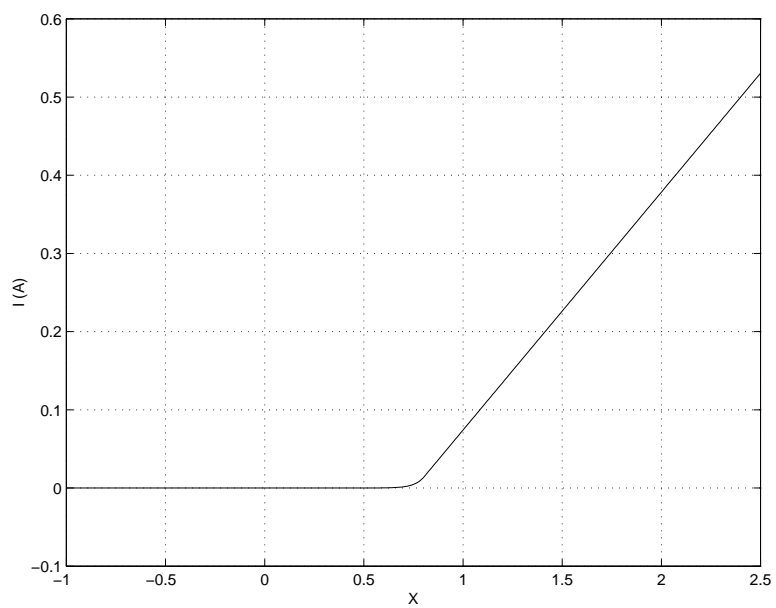


Figure 2.2: Relation between x and i in a diode.

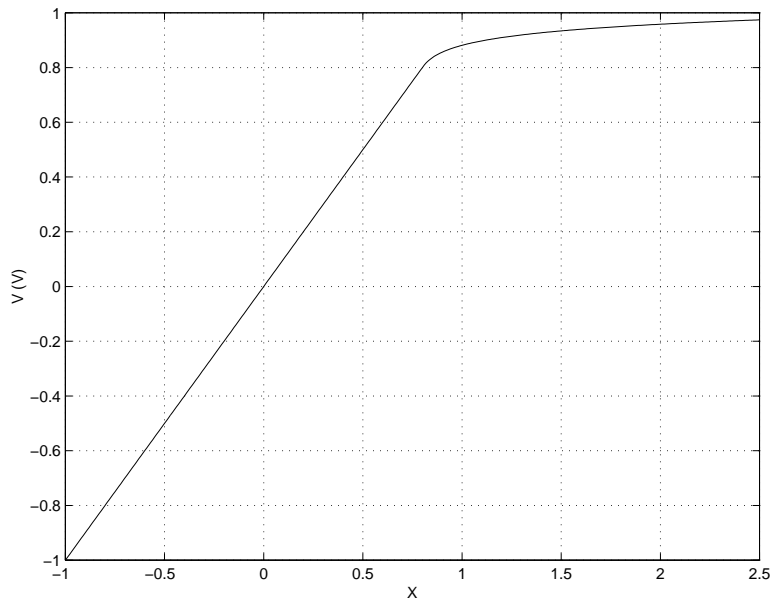


Figure 2.3: Relation between x and v in a diode.

In the future on-chip signal integrity analysis will require distributed electromagnetic analysis to provide the rigor needed for multi-gigahertz signal integrity-driven designs. This contrasts with the current quasi-static approach that leads to independent inductance, capacitance and resistance extraction. For example, effects such as current crowding and common-impedance coupling, which have been shown to be of critical importance to the accurate prediction of interconnect-induced delay and crosstalk, cannot be quantified properly without the simultaneous modeling of inductive and capacitive coupling. Distributed electromagnetic modeling renders the concept of a global reference node inadequate as it implies instantaneous redistribution of charge among spatially separated components. Instead the concept of local reference node is used to properly accommodate distributed models. The local reference node concept facilitates the modeling of mixed systems where, for example, the local reference node is absolute zero kelvin for a thermal system and the inertial reference frames for mechanical systems.

*fREEDA*TM implements several types of analysis including a time marching analysis (with several different integration methods available), a unique wavelet transient

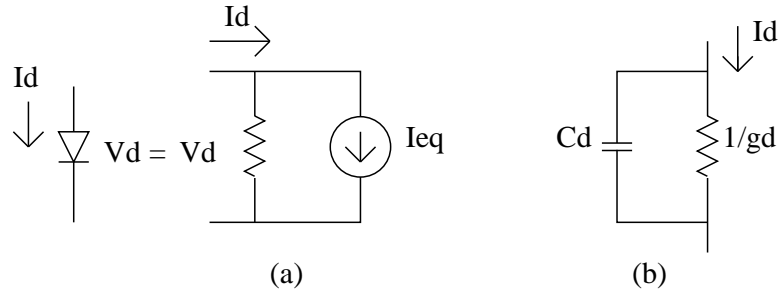


Figure 2.4: Equivalent Diode model in Spice.

analysis, and harmonic balance analysis. All of the analyses described use the same element model code. This is made possible by using generic evaluation mechanism combined with automatic differentiation technology. Consequently the element code does not include code to calculate derivatives (or equivalently the development of companion models) so model development time is dramatically simplified.

2.3 Universal Device Modeling

The nonlinear diode equation in SPICE is expressed by

$$I_d = I_s \left[\exp\left(\frac{qV_d}{NKT}\right) - 1 \right]. \quad (2.1)$$

Fig. 2.4 illustrates an equivalent circuit representation of the linearized diode equation:

$$I_d = G_d * V_d + I_{eq}. \quad (2.2)$$

The circuit in Fig. 2.4 is the equivalent circuit SPICE [2] uses to represent diodes in the system equations. This equivalent circuit is known as the linear diode model. In SPICE, diodes are modeled by a conductance in parallel with a current source. During a simulation, the conductance value G_d will be stored in the conductance array. In turn these values will be used to compute a new set of solution voltages. At each solution point, the diode current and conductance values are computed and stored within the system equations.

When discussing the linear diode model, the distinction between the linear model and the small-signal diode model should be made. Although the linear diode model

is similar to the small-signal diode model, the two are not the same. The linear model of the diode contains two elements, the dynamic conductance (G_d) of the diode and the equivalent current (I_{eq}). The small signal model also contains two elements, the dynamic resistance ($1/g_d$) and the small signal capacitance (c_d). The linear diode model may or may not contain the small signal capacitance (c_d) depending on whether a transient analysis is being performed and the small signal diode model does not contain the equivalent current (I_{eq}). Transistors are represented by extending the linear diode model. Hence it is obvious that the same device model cannot be used for both transient and small signal analysis.

In this thesis a new parameterized nonlinear device model formulation is presented. The new formulation enables a unique description of the model in any circuit analysis type and avoids charge conservation problems that may arise in transient analysis [4]. The parametric description provides great flexibility for the design of nonlinear device models. The number of parameters or state variables required is the minimum necessary and they can be chosen to achieve robust numerical characteristics.

2.4 Object Oriented Circuit Simulators

APLAC [9] is a significant achievement in the development of object-oriented circuit simulators with the object orientation implemented in the standard C language using macros. The most important feature in *APLAC* is that every circuit element is modelled internally using independent and voltage controlled current sources. Since all models in *APLAC* are eventually mapped to current sources, the simple nodal linear DC analysis, $Gu = j$, is all that is required to realize nonlinear DC, AC, transient and harmonic balance analysis. Here G , u and j denote conductance matrix, nodal voltages and the independent source current, respectively. The cost of this approach is reduced speed. In part this is because the C language is not optimal for OO applications but also because high level of abstraction introduces overhead. However the objective of providing great functionality to enable experimentation with new element types and analysis techniques was achieved.

Another object-oriented circuit simulator *Sframe* adopts a common interface for

all the circuit elements. In this way, all the code related to one element is separated from the rest of the program. The main program does not have dependencies on the individual elements. The result is that the programming effort required to add new elements and algorithms is greatly reduced. Sframe is written in C++ and allows one element to be composed of other basic elements. Sframe incorporates several novel features including automatic differentiation. In this simulator C++ is used as the circuit description language rather than a SPICE netlist.

2.5 Freeda

2.5.1 Introduction

The rapid rate of innovation of microwave and millimeter wave systems requires the development of an easily extensible and modifiable computer aided engineering (CAE) environment. While great strides have been made in the flexibility of commercial CAE tools, these sometimes prove inadequate in modeling advanced systems. As with virtually all aspects of electronic engineering the abstraction level of RF and microwave theory and techniques has increased dramatically. In particular, large systems are being designed with attention given to the interaction of components at many levels. One of the most significant developments relevant to computer aided engineering is the rise of object-oriented (OO) design practice [11, 12]. While it is normal to think of OO-specific programming languages as being the main technology for implementing OO design, good OO practice can be implemented in more conventional programming languages such as C. However OO-specific languages foster code reuse and have constructs that facilitate object manipulation. The OO abstraction is well suited to modeling electronic systems, for example, circuit elements are already viewed as discrete objects and at the same time as an integral part of a (circuit) continuum. The OO view is a unifying concept that maps extremely well onto the way humans perceive the world around them.

Non-OO circuit simulators always become complicated with many layers of special cases. Referring to circuit elements again, traditional simulation implementa-

tions have many if-then like statements and individually identify every element in many places for special handling. An integral part of the various high performance computing initiatives is the separation of the core components embodying numerical methods from the modeling and solver formulation process with the result that numerical techniques developed by computer scientists and mathematicians can be formulated using formal correctness procedures [26]. Thus, what is adopted, is that the circuit abstraction is adapted so that highly reliable and efficient pre-developed libraries can be used. C++ was once considered slow for scientific applications. Advances in compilers and programming techniques, however, have made this language attractive and in some benchmarks C++ outperforms Fortran. Several OO numerical libraries have been developed. Of great importance to the work described here is the incorporation of the standard template library (STL). The STL is a C++ library of container classes, algorithms, and iterators; it provides many of the basic algorithms and data structures of computer science. The STL is a generic library, meaning that its components are heavily parameterized: almost every component in the STL is a template. The current ISO/ANSI C++ standard has not been fully implemented and C++ compilers support a variable subset of the standard. The biggest areas of noncompliance being the templates and the standard library. The goal in design was to obtain speed in development, to use off the shelf advanced numerical techniques, and to allow easy expansion and testing of new models and numerical methods. The circuit simulator implementing these ideas is *fREEDA*TM. *fREEDA*TM is the first circuit simulator to use recent OO techniques. The design intent was to combine the advantages of previous OO circuit simulators with these new developments as well as expanding capability. *fREEDA*TM uses C++ libraries written in C or Fortran.

2.5.2 Support Libraries

*fREEDA*TM makes intensive use of support libraries and these have been reviewed recently: see [3].

2.5.3 Automatic Differentiation

The Automatic Differentiation library will be reviewed here as the capabilities of this library are critical to the universal device modeling presented in this thesis. Automatic differentiation is a technology for evaluating the derivative of a function with respect to a variable without explicit expressions for the derivative. Most nonlinear computations require the evaluation of first and higher derivatives of vector functions with m components in n real or complex variables. Often these functions are defined by sequential evaluation procedures involving many intermediate variables. By eliminating the intermediate variables symbolically, it is theoretically always possible to express the m dependent variables directly in terms of the n independent variables. Typically, however, the attempt results in unwieldy algebraic formulae, if it can be completed at all. Symbolic differentiation of the resulting formulae will usually exacerbate this problem of expression swell and often entails the repeated evaluation of common expressions.

An obvious way to avoid such redundant calculations is to apply an optimizing compiler to the source code that can be generated from the symbolic representation of the derivatives in question. Given a code for a function $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$, automatic differentiation (AD) uses the chain rule successively to compute the derivative matrix. AD has two basic modes, forward mode and reverse mode. The difference between these two is the way the chain rule is used to propagate the derivatives [27].

A versatile implementation of the AD technique is Adol-C, a software package written in C and C++. The numerical values of derivative vectors (required to fill a Jacobian for solving non-linear elements using Newton's method) are obtained free of truncation errors at a small multiple of the run time required to evaluate the original function with little additional memory required. It is important to note that AD is not numerical differentiation and the same accuracy achieved by evaluating analytically developed derivatives is obtained. The `eval()` method of the nonlinear element class (a `fREEDA™` class) is executed at initialization time and so the operations to calculate the currents and voltages of each element are recorded by Adol-C in a tape which is actually an internal buffer. After that, each time that the values or the

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

2.6 Electro-Thermal Modeling

The performance and characteristics of semiconductor components in electronic packages can be considerably affected by temperature variations, and for this reason accurate circuit simulation requires that the dynamic temperature effects induced by the heat dissipated in the circuit be taken into account [18]. Modeling electro-thermal interactions in integrated circuits has been addressed in a variety of ways. Existing methods can be broadly classified into two groups: "relaxation" methods, which sim-

ulate the thermal and electrical problems separately and periodically exchange temperature and power information until thermal and electrical convergence is reached. The second method, direct or fully coupled methods, where a single circuit simultaneously handles both electrical and thermal states. In the later, the thermal model is integrated into the circuit simulator, thus the presentation of the thermal environments is rather simple. Direct coupled methods have better convergence properties than relaxation methods.

The electro-thermal problem has been a major concern in analog circuit design because the bipolar circuits consume a large amount of power and have potential thermal runaway problem. An overview of the generic electrothermal analysis flow is presented.

Electrothermal simulation consists of electrical and thermal simulations. The purpose of electrical simulation is to obtain the information on power dissipation and the performance of devices or circuits. On the other hand, the thermal simulation is used to find the temperature profile and to update all the temperature-dependent physical parameters of the device or circuit model.

A coupled set of nonlinear electrothermal equations is first generated. The equations are represented by a matrix form and then linearized and solved using the Newton-Raphson method. The linearized circuit matrix contains three parts.

1. Elements corresponding to the electrical circuit (Y_V).
2. Elements corresponding to the thermal circuit (Y_{th}).
3. Elements corresponding to the coupling between the two circuits.

Once the matrix is solved and the dc solution is found at time t the transient solution of the temperature and the node voltages can be found using the preferred integration formula. The above procedure are similar to those of SPICE.

Chapter 3

Universal Modeling

3.1 Introduction

This chapter describes in detail the Universal Modeling algorithm implemented in *fREEDA*TM.

Section 3.2 is an introduction to nonlinear device models. Several issues which complicate the task of modeling a nonlinear device are explained.

Section 3.3 describes the use of parameterized device models [22]. The nonlinear diode model (Fig. 3.2) example is used to emphasize the need for device parameterization in circuit simulators.

Section 3.4 describes the universal device model implemented in *fREEDA*TM. The universal device model formulation was designed to solve two issues,

1. Nonlinear device charge conservation.
2. Analysis independent device model.

The Gummel–Poon BJT was initially implemented in *fREEDA*TM using the conventional voltage based state variable algorithm. The equations and the implementation of the BJT in *fREEDA*TM, using the voltage state variable routine, are described in Section 3.4.1 and Section 3.5 respectively. The observations made during the modeling

procedure were,

1. The model did not conserve charge.
2. Implementation of the BJT device equations in *fREEDA*TM were time consuming.

The algorithm adopted to obtain the final terminal currents and voltages, involved calculation of second order derivatives of the state variables, (i.e. voltage). This is was time consuming and computationally expensive.

The universal algorithm was hence designed to solve the charge non-conservation problem observed in the BJT and reduce the time taken to implement non-linear devices in *fREEDA*TM. The universal algorithm makes use of the automatic differentiation feature present in ADOL-C. Section 3.6 describes the universal model formulation with the assistance of the BJT non-linear device model. Charge is chosen as the state variable in this algorithm as opposed to voltage. The final terminal current is obtained as a time derivative of the state variable, (i.e. charge). It was observed that the maximum order of the state variable derivative can never exceed 1 using the universal model formulation. This reduces the time required to implement non-linear device models in *fREEDA*TM.

Section 3.7 describes the steps involved in implementing the Curtise–Ettenberg MESFET in *fREEDA*TM.

The need for a universal model was discussed in Chapter 2.3. By using the universal formulation algorithm proposed in this chapter, we were able to use the same device model code for different analysis routines. This mechanism is termed as generic evaluation. Both time domain and frequency domain simulations were used to validate the universal BJT and MESFET models. The results obtained using different analysis routines and the same device model code are shown in Chapter 4.

Section 3.9 extends the concept of generic evaluation to include electro-thermal simulations. The algorithm used to implement electro-thermal models in *fREEDA*TM is described. The simulation results obtained for a electro-thermal BJT are shown in Section 4.5.

3.2 Nonlinear Device Models and Circuits

Techniques for the analysis of linear systems are powerful and relatively simple at the same time. In circuits the inputs are provided by independent sources while the outputs can be voltages or currents anywhere in the circuit. A circuit is said to be linear if and only if it satisfies the principle of superposition, that is

$$F[\alpha_1(t) + \beta_2(t)] = [x_1(t)] + [x_2(t)]. \quad (3.1)$$

where α and β are real numbers; otherwise, the circuit is said to be nonlinear. Any two signals applied to a linear circuit are processed independently (i.e. different frequencies applied to the circuit do not interact). The only frequencies present in a linear circuit are those of the independent sources. Linear circuits must be necessarily made of linear circuit elements, that is, elements that are defined by the equations that satisfy the principle of superposition. For example, there is a linear relationship between voltage and current in a linear resistor. Conversely, equations of definition of nonlinear circuit elements do not satisfy (3.1).

A resistor defined by the voltage-current characteristic

$$i = a_1v + a_2v^2 + a_3v^3. \quad (3.2)$$

is clearly nonlinear. Suppose a voltage $V_1 \cos \omega_1 t + V_2 \cos \omega_2 t$ is applied to the resistor. Then

$$i = a_1(V_1 \cos \omega_1 t + V_2 \cos \omega_2 t) + a_2(V_1 \cos \omega_1 t \quad (3.3)$$

$$+ V_2 \cos \omega_2 t)^2 + a_3(V_1 \cos \omega_1 t \quad (3.4)$$

$$+ V_2 \cos \omega_2 t)^3. \quad (3.5)$$

It is obvious that several new frequencies will be generated for the resistor, by expanding terms $a_2(V_1 \cos \omega_1 t + V_2 \cos \omega_2 t)^2$ and $a_3(V_1 \cos \omega_1 t + V_2 \cos \omega_2 t)^3$ in (3.5). A closer examination reveals that the term v^n generates frequencies having the form $p\omega_1 + q\omega_2$, where p and q are integers such that $|p| + |q| \leq n$. For analog systems the generation of new frequencies is the most relevant effect of nonlinearities and is what differentiates linear and nonlinear circuits for practical purposes. The total current at the

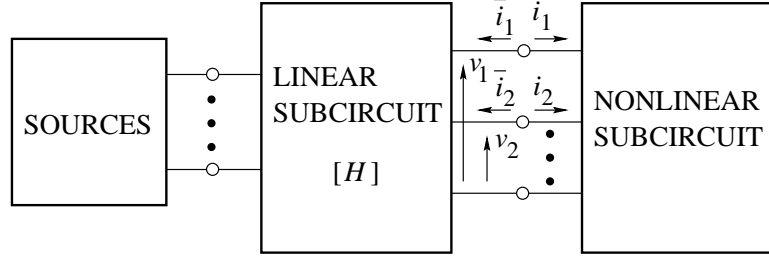


Figure 3.1: Example: partition representation of the time-invariant harmonic balance method.

frequency ω_1 in (3.2) is given by

$$a_1 V_1 + \frac{3}{2} a_3 \frac{V_1^3}{2} + V_1 V_2^2 \cos \omega_1 t. \quad (3.6)$$

The amplitude of this current is a nonlinear function of both V_1 and V_2 . Hence, even the relationship between quantities at the same frequency is nonlinear in a nonlinear device. The situation becomes more complex when the resistor defined by (3.2) is connected in a circuit.

3.3 Parameterized Device Models

This section presents a way to formulate the equations of the nonlinear device inside a circuit. This concept was originally applied to the piecewise harmonic balance circuit analysis proposed by Nakhla and Vlach [23]. This approach is based on partitioning the linear and nonlinear portions of the circuit as shown in Fig. 3.1.

The nonlinear device model can be described with the following set of equations [22]:

$$\mathbf{v}(t) = \mathbf{v}(\mathbf{x}(t), d\mathbf{x}/dt, \dots, d^m \mathbf{x}/dt^m, \mathbf{x}_D(t)) \quad (3.7)$$

$$\mathbf{i}(t) = \mathbf{i}(\mathbf{x}(t), d\mathbf{x}/dt, \dots, d^m \mathbf{x}/dt^m, \mathbf{x}_D(t)). \quad (3.8)$$

where $\mathbf{v}(t)$ and $\mathbf{i}(t)$ are vectors of voltages and currents at the ports of the nonlinear device, $\mathbf{x}(t)$ is a vector of parameters or state variables and $\mathbf{x}_D(t)$ a vector of time-delayed state variables, i.e., $[\mathbf{x}_D(t)]_i = x_i(t - \tau_i)$. All vectors in (3.7) and (3.8) have the

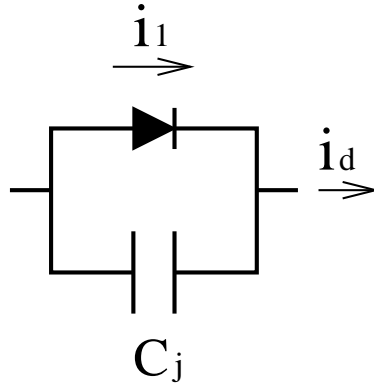


Figure 3.2: Example: schematic of the simplified Diode Model.

same size equal to the number of ports of the nonlinear device being modeled. This kind of representation is convenient from the physical viewpoint, as it is equivalent to a set of implicit integro-differential equations in the port currents and voltages. This results in complete generality in device modeling. For example, it is no longer necessary to express nonlinear elements as voltage controlled current sources.

Consider the simplified microwave diode model of Fig. 3.2. The corresponding equations are the following:

$$i_1(v) = I_s(\exp(\alpha v) - 1).$$

$$c_j(v) = \begin{cases} C_{t0}(1 - v/\phi)^{-\gamma} + C_{d0} \exp(\alpha'v) & \text{if } v \leq .8\phi \\ C_{t0}(.2)^{-\gamma} + C_{d0} \exp(\alpha'v) & \text{if } v \geq .8\phi \end{cases},$$

where v is the junction voltage. The capacitor charge q_j can be evaluated as

$$q_j(v) = \int_0^v c_j(u) du.$$

Accurate transient analysis modeling requires q_j to be chosen as the state variable. We need v to calculate $i_1(v)$. Since it is not possible to analytically solve for $v(q)$, the only alternative is to model the diode with two state variables, namely q and v or x (x for the diode model is defined in [22]). The addition of extra state variables is not desirable because it increases the dimension of the nonlinear system of equations in the circuit analysis algorithm [26].

Note also that even if we could analytically solve for $q(v)$, the advantageous parametric formulation demonstrated in [22] is no longer possible.

We propose to formulate the diode equations in different stages. First, the voltage (v) and the current (i_1) through the ideal diode in Fig. 3.2 are calculated from a parameter x . The capacitor charge $q(v)$ can also be calculated at this stage. Then the current through the capacitor i_c is evaluated at a second stage with

$$i_c = \frac{dq}{dt}.$$

The actual code that performs the derivation is outside the nonlinear device model, and so the model itself is independent of the type of circuit analysis. This procedure provides for good numerical properties of parameterization, charge conservation in transient analysis and immunity to discontinuities in the first derivative of the capacitances. The general formulation is described in the next section.

To obtain better numerical properties [22], this model can be parameterized as follows

$$v = \begin{cases} x & \text{if } x \leq V_1 \\ V_1 + 1/\alpha \ln(1 + \alpha(x - V_1)) & \text{if } x > V_1 \end{cases} \quad (3.9)$$

$$i_1 = \begin{cases} I_s(\exp(\alpha x) - 1) & \text{if } x \leq V_1 \\ I_s \exp(\alpha V_1)(1 + \alpha(x - V_1)) - I_s & \text{if } x > V_1 \end{cases} \quad (3.10)$$

where V_1 is a threshold value [3]. The junction capacitance, C_j can then be calculated from the value of v in (3.9).

The total diode current is equal to the sum of i_1 in (3.10) and the current through the capacitance:

$$i_d = i_1 + C_j(v) \frac{dv}{dx} \frac{dx}{dt} \quad (3.11)$$

As noted before, this formulation results in charge conservation problems when the model is used for transient analysis. Also, if the first derivative of $C_j(v)$ is not a

continuous function, (3.11) is likely to produce non-convergence even in a Harmonic Balance analysis.

3.4 Universal Model Formulation

*fREEDA*TM offers a refined way to implement nonlinear elements [4], provided the element equations can be represented in a parametric form.

The nonlinear device models are described by the following set of equations:

$$\text{stage 1} : \begin{cases} \mathbf{f}_1(\mathbf{x}, \mathbf{x}_D) \\ \mathbf{g}_1(\mathbf{x}, \mathbf{x}_D) \end{cases} \quad (3.12)$$

$$\text{stage 2} : \begin{cases} \mathbf{f}_2(\mathbf{f}_1, d\mathbf{g}_1/dt) \\ \mathbf{g}_2(\mathbf{f}_1, d\mathbf{g}_1/dt) \end{cases} \quad (3.13)$$

⋮

$$\text{stage } n - 1 : \begin{cases} \mathbf{f}_{n-1}(\mathbf{f}_{n-2}, d\mathbf{g}_{n-2}/dt) \\ \mathbf{g}_{n-1}(\mathbf{f}_{n-2}, d\mathbf{g}_{n-2}/dt) \end{cases} \quad (3.14)$$

$$\text{stage } n : \begin{cases} \mathbf{v}(\mathbf{f}_{n-1}, d\mathbf{g}_{n-1}/dt) \\ \mathbf{i}(\mathbf{f}_{n-1}, d\mathbf{g}_{n-1}/dt) \end{cases} . \quad (3.15)$$

Here \mathbf{x} and \mathbf{x}_D are the state variable vectors defined in (3.7) and (3.8). The vector functions \mathbf{f}_j and \mathbf{g}_j are evaluated in order of increasing j . The dimension of these vector functions depends on the type of model being implemented. The vectors of voltages and currents, v and i respectively, are evaluated at the end.

This set of equations retain the generality of (3.7) and (3.8). Elements that originally required only first order derivatives of the state variables now require two function stages. If higher order derivatives were necessary in the formulations (3.7) and (3.8), this would translate in up to several function stages in (3.12)–(3.15).

The new parametric model formulation shares the advantages of the original one and solves the charge conservation problem in transient analysis. It is also compatible with the generic evaluation technique described in Ref. [26]. This means the nonlinear

device models can be described by a unique set of routines used for different circuit analysis types, such as HB or transient analysis. The automatic differentiation technique [27] can be applied to calculate the Jacobian of the nonlinear model because the time differentiation operation is performed outside the model description. The division of the calculation in different stages in (3.12)–(3.15) also simplifies the implementation of complex device models because it is no longer necessary to express the external currents and voltages from the original state variables and derivatives. Instead, intermediate variables can be calculated and time derivation is applied to some of them.

The derivation of the equations and the corresponding Jacobian for different analysis types other than transient starting from the set of (3.12)–(3.15) will be shown in Section 3.8.5.

To illustrate how this set of equations is used with different circuit analysis types, we will show the derivation of the Jacobian of the nonlinear model in time domain and the formulation of the model equations in harmonic balance.

3.4.1 Example: Bipolar Junction Transistor

The Gummel-Poon charge conserving Bipolar Junction Transistor model was implemented in *fREEDA*TM. In the following pages a detailed model development procedure is described for both the conventional voltage based algorithm and the universal charge based algorithm. The Gummel-Poon model was implemented using 50 model parameters. A list of the parameters and their default values are shown in Appendix A. The equivalent circuit of the Gummel-Poon transistor implemented in *fREEDA*TM is shown in Fig. 3.3.

In the NPN and PNP models with substrate effects included, terminal N_X is connected to node C. The NPN and PNP BJT models are identical but with positive sense of currents and voltages opposite so that the model parameters are always positive. The bipolar junction transistor model in Spice is based on the charge control model of Gummel and Poon. Extensions in the SPICE implementation deal with effects at high bias levels. The model reduces to the simpler Ebers-Moll model with

The non-ideal base-emitter current,

$$I_{LE} = I_{SE} \left(e^{V_{BE}/(N_E V_{TH})} - 1 \right) \quad (3.20)$$

The reverse diffusion current,

$$I_{BR} = I_S \left(e^{V_{BC}/(N_R V_{TH})} - 1 \right) \quad (3.21)$$

The non-ideal base-collector current,

$$I_{LC} = I_{SC} \left(e^{V_{BC}/(N_C V_{TH})} - 1 \right) \quad (3.22)$$

and the base charge factor,

$$K_{QB} = \frac{1}{2} \left[1 - \frac{V_{BC}}{V_{AF}} - \frac{V_{BE}}{V_{AR}} \right]^{-1} \left(1 + \sqrt{1 + 4 \left(\frac{I_{BF}}{I_{KF}} + \frac{I_{BR}}{I_{KR}} \right)} \right) \quad (3.23)$$

The base-emitter capacitance

$$C_{BE} = \text{Area} (C_{BE\tau} + C_{BEJ}). \quad (3.24)$$

where the base-emitter transit time or diffusion capacitance

$$C_{BE\tau} = \tau_{F,EFF} \frac{\partial I_{BF}}{\partial V_{BE}}. \quad (3.25)$$

The effective base transit time

$$\tau_{F,EFF} = \tau_F \left[1 + X_{TF} (3x^3 - 2x^2 e^{(V_{BC}/1.44V_{TF})}) \right]. \quad (3.26)$$

Where x is calculated as

$$x = I_{BF}/(I_{BF} + \text{Area}I_{TF}). \quad (3.27)$$

The base-emitter junction (depletion) capacitance

$$C_{BEJ} = \begin{cases} C_{JE} \left(1 - \frac{V_{BE}}{V_{JE}} \right)^{-M_{JE}} & V_{BE} \leq F_C V_{JE} \\ C_{JE} (1 - F_C)^{1+M_{JE}} \left(1 - F_C (1 + M_{JE}) + M_{JE} \frac{V_{BE}}{V_{JE}} \right) & V_{BE} > F_C V_{JE} \end{cases} \quad (3.28)$$

The base-collector capacitance

$$C_{BC} = \text{Area} (C_{BC\tau} + X_{CJC}C_{BCJ}). \quad (3.29)$$

where the base-collector transit time or diffusion capacitance

$$C_{BC\tau} = \tau_R \frac{\partial I_{BR}}{\partial V_{BC}}. \quad (3.30)$$

The base-collector junction (depletion)capacitance

$$C_{BCJ} = \begin{cases} C_{JC} \left(1 - \frac{V_{BC}}{V_{JC}}\right)^{-M_{JC}} & V_{BC} \leq F_C V_{JC} \\ C_{JC} (1 - F_C)^{1+M_{JC}} \left(1 - F_C (1 + M_{JC}) + M_{JC} \frac{V_{BC}}{V_{JC}}\right) & V_{BC} > F_C V_{JC} \end{cases} \quad (3.31)$$

The capacitance between the extrinsic base and the intrinsic collector

$$C_{BX} = \begin{cases} \text{Area} (1 - X_{CJC}) C_{CJC} \left(1 - \frac{V_{BX}}{V_{JC}}\right)^{-M_{JC}} & V_{BX} \leq F_C V_{JC} \\ (1 - X_{CJC}) C_{CJC} (1 - F_C)^{-(1+M_{JC})} \\ \quad \times (1 - F_C (1 + M_{JC}) + M_{JC} \frac{V_{BX}}{V_{JC}}) & V_{BX} > F_C V_{JC} \end{cases} \quad (3.32)$$

The substrate junction capacitance

$$C_{JS} = \begin{cases} \text{Area} C_{JS} \left(1 - \frac{V_{CJS}}{V_{JS}}\right)^{-M_{JS}} & V_{CJS} \leq 0 \\ \text{Area} C_{JS} \left(1 + M_{JS} \frac{V_{CJS}}{V_{JS}}\right) & V_{CJS} > 0 \end{cases} \quad (3.33)$$

In Section 3.5 the algorithm to model the above equations using voltage as a state variable is described.

3.5 Bipolar Transistor: Voltage Based State Variable

The conventional voltage based modeling algorithm used to implement the Bipolar Transistor makes use of the ADOL-C class [27]. To implement the transistor the state

variables are

1. Voltage, (V_{BC}), across the base-collector capacitor, C_{BC} .
2. Voltage, (V_{BE}), across the base-emitter capacitor, C_{BE} .
3. Voltage, (V_{CJS}), across the substrate capacitor, C_{JS} .

To find the voltage and current components at the terminals in Fig. 3.3 we need to solve the Kirchoff's current and voltage laws at every node of the equivalent circuit and around every loop. The state variables are assigned as follows

1. $x[0] : V_{BC}$
2. $x[1] : V_{BE}$
3. $x[2] : V_{CJS}$

where $x[]$ is the vector of state variables.

The DC equations are converted to replace the respective voltage components with their equivalent state variables (i.e.)

$$I_{BF} = I_S \left(e^{x[1]/(N_F V_{TH})} - 1 \right)$$

The non-ideal base-emitter current,

$$I_{LE} = I_{SE} \left(e^{x[1]/(N_E V_{TH})} - 1 \right)$$

The reverse diffusion current,

$$I_{BR} = I_S \left(e^{x[0]/(N_R V_{TH})} - 1 \right)$$

The non-ideal base-collector current,

$$I_{LC} = I_{SC} \left(e^{x[0]/(N_C V_{TH})} - 1 \right)$$

and the base charge factor,

$$K_{QB} = \frac{1}{2} \left[1 - \frac{x[0]}{V_{AF}} - \frac{x[1]}{V_{AR}} \right]^{-1} \left(1 + \sqrt{1 + 4 \left(\frac{I_{BF}}{I_{KF}} + \frac{I_{BR}}{I_{KR}} \right)} \right)$$

The terminal voltages at the base, collector and emitter are computed by solving a set of Kirchoff's voltage equations. Applying Kirchoff's voltage law in loop 1 we obtain

$$V_{BX} + I_B R_B = V_{BC}. \quad (3.34)$$

The base current in (3.34)

$$I_B = I_{BC} + I_{BE} + I_{C_{BC}} + I_{C_{BE}} + I_{C_{BX}}. \quad (3.35)$$

The current through the capacitors are calculated as

$$I = C \frac{\partial v}{\partial t}. \quad (3.36)$$

and the current through the base-collector capacitance C_{BC} is calculated as

$$I_{C_{BC}} = C_{BC} \frac{\partial V_{BC}}{\partial t}. \quad (3.37)$$

The time derivative of V_{BC} in (3.37) is calculated by ADOL-C. Using the same method the current through capacitor C_{BE} is calculated.

$$I_{C_{BE}} = C_{BE} \frac{\partial V_{BE}}{\partial t} \quad (3.38)$$

The current components through the base-emitter and base-collector capacitance involve calculation of the first time derivative of their respective voltages. The final current component in equation (3.35) is calculated as

$$I_{C_{BX}} = C_{BX} \frac{\partial V_{BX}}{\partial t}. \quad (3.39)$$

In (3.39) $V_{BX} = V_{BC} + \text{Potential drop across } R_B$. The potential drop across R_B is calculated from:

$$R_B \times \text{Current through } R_B \quad (3.40)$$

The total current through R_B ,

$$R_B = I_{BC} + I_{BE} + C_{BC} \frac{V_{BC}}{\partial t}. \quad (3.41)$$

Hence the voltage drop across V_{BX} is,

$$V_{BX} = V_{BC} + R_B \left(C_{BC} \frac{\partial V_{BC}}{\partial t} \times \partial V_{BE} \partial t \right). \quad (3.42)$$

From (3.39)

$$I_{CBX} = C_{BX} \left(\partial V_{BC} \partial t + R_B \left(C_{BC} \frac{\partial^2 V_{BC}}{\partial t^2} \times \frac{\partial^2 V_{BE}}{\partial t^2} \right) \right). \quad (3.43)$$

In (3.43) the second derivative of V_{BE} and V_{BC} are calculated to solve I_{BX} . The Bipolar Transistor is the only device in *fREEDA*TM to make use of the second time derivative of the state variables. The substrate current has a DC and capacitive current component. The DC component is

$$I_{JS} = Area I_{SS} \left(\exp\left(\frac{V_{CJS}}{NS V_{TH}}\right) - 1 \right). \quad (3.44)$$

The capacitive current component is

$$I_{CJS} = C_{JS} \times \frac{\partial V_{CJS}}{\partial t} \quad (3.45)$$

The final terminal voltages and currents are calculated by using Kirchoff's law.

Collector Current

$$I_C = I_{CE} - (I_{BC} + I_{C_{BC}}) - I_{BX} - (I_{JS} + I_{C_{JS}}) \quad (3.46)$$

Base Current

$$I_B = I_{BX} + (I_{BC} + I_{C_{BC}}) + (I_{BE} + I_{C_{BE}}) \quad (3.47)$$

Emitter Current

$$I_E = -(I_{CE} + (I_{BE} + I_{C_{BE}}) - (I_{JS} + I_{C_{JS}})) \quad (3.48)$$

Collector Substrate Voltage

$$V_{C_{JS}} = V_{C_{JS}} + R_C \times I_C \quad (3.49)$$

Base Substrate Voltage

$$V_{B_{JS}} = V_{C_{JS}} + V_{BE} + I_B \times R_B \quad (3.50)$$

Emitter Substrate Voltage

$$V_{E_{JS}} = V_{BE} - V_{BC} + V_{C_{JS}} + I_E \times R_E \quad (3.51)$$

3.6 Bipolar Transistor: Universal Model Formulation

To illustrate the universal model formulation consider the simplified NPN-type BJT model of Fig. 3.3. The voltages across the base–collector capacitor V_{BC} , the base–emitter capacitor V_{BE} and the substrate capacitor $V_{C_{JS}}$ are chosen as state variables (a similar parameterization as explained in Ref. [22] can also be done). Note that we only need three state variables. If the charge at the capacitors were chosen, this number would be four and would thus increase the dimension of the nonlinear system of equations in the simulation algorithm [26]. With the proposed approach we need three functional stages to model the transistor. Each stage has two input and two output parameters. In stage 1 the dc current components I_{bc} and I_{be} are computed as

$$\begin{aligned} I_{be} &= \frac{I_{bf}}{\beta_F} + I_{le} \\ I_{bc} &= \frac{I_{br}}{\beta_R} + I_{lc}, \end{aligned}$$

where Ibf and Ile are the components of the currents through the base-emitter diode and Ibr and Ilc are components of the currents through the base-collector diode (3.19), (3.20), (3.21), (3.22). The first output vector from stage 1 stores the charge across the base collector and base emitter capacitors which can be evaluated as

$$\begin{aligned} q_{bc}(v) &= \int_0^v c_{bc}(v_{bc})dv_{bc} \\ q_{be}(v) &= \int_0^v c_{be}(v_{be})dv_{be} \\ q_{cjs}(v) &= \int_0^v c_{cjs}(v_{bx})dv_{cjs}. \end{aligned}$$

The second output vector stores the diode current components and junction voltages. In the second stage the derivatives of the charges across the capacitors are determined.

$$\begin{aligned} I_{Cbc} &= \frac{dq_{bc}}{dt} \\ I_{Cbe} &= \frac{dq_{be}}{dt} \\ I_{C_{cjs}} &= \frac{dq_{cjs}}{dt} \end{aligned}$$

Hence the capacitive current contribution is obtained. The above result is used to calculate the charge across the distributed base collector capacitor C_{bx} .

Inputs to stage 3 contain the corresponding current through C_{bx} and the junction voltages. In stage 3 the final external voltages and currents are calculated using the intermediate variables generated at the previous stages. Only the first time derivative of the state variable is used at any stage in the universal modeling algorithm. By

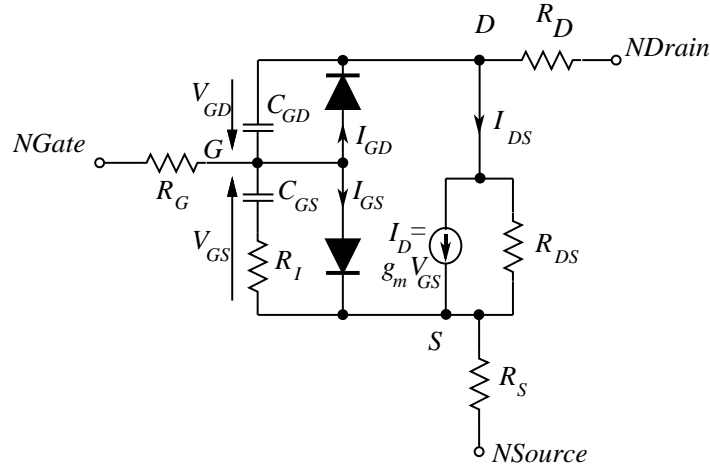


Figure 3.4: Mesfet Curtice Ettenberg Model.

using the three stage algorithm for the Bipolar Transistor we can avoid the second time derivative of state variables. The use of intermediate data in the universal model reduces both the memory used to store the state variables and the overall simulation time. To our knowledge this is the only microwave circuit simulation algorithm which can handle different analysis routines using a single device model.

3.7 Mesfet Model : Curtice-Ettenberg

The Curtice Ettenberg MESFET model shown in Fig. 3.4 was implemented in *fREEDA*TM using the parameters listed in Appendix B.

The equations used to model the MESFET (Curtice Ettenberg Model) are:

Drain Source Channel Current

$$I_{DS} = \begin{cases} (A0 + A1V_x + A2V_x^2 + A3V_x^3) \tanh(GAMA V_{DSI}), & V_{GSI} > VT0 \\ 0, & \text{otherwise} \end{cases} \quad (3.52)$$

In Fig. 3.4 the gate-source diode current

$$I_{GS} = IS \left(\exp\left(\frac{V_{GSI}}{NRV_T}\right) - 1 \right) - IB0 \exp\left(\frac{-(V_{GSI} + VBD)}{NRV_T}\right) \quad (3.53)$$

The gate–drain diode current

$$I_{GD} = I_{GDC} - IB0 \exp\left(\frac{-(V_{GDI} + VBD)}{NRV_T}\right) \quad (3.54)$$

where

$$I_{GDC} = IS \left(\exp\left(\frac{V_{GDI}}{NV_T}\right) - 1 \right) \quad (3.55)$$

The channel resistance

$$R_i = \begin{cases} R10(1 - KRV_{GSI}) & KRV_{GSI} < 1.0 \\ 0 & KRV_{GSI} \geq 1.0 \end{cases} \quad (3.56)$$

The gate–source capacitance in Fig. 3.4

$$C_{GS} = \begin{cases} CGS0 \left(1 - \frac{V_{GSI}}{VBI}\right)^{-MGS} & V_{GSI} < FCC \\ CGS0(1 - FCC)^{-MGS} \left(1 + MGS \frac{V_{GSI} - FCC}{VBI(1 - FCC)}\right) & V_{GSI} \geq FCC \end{cases} \quad (3.57)$$

The gate–drain capacitance

$$C_{GD} = \begin{cases} CGD0 \left(1 - \frac{V_{GDI}}{VBI}\right)^{-MGD} & V_{GDI} < FCC \\ CGD0(1 - FCC)^{-MGD} \left(1 + MGD \frac{V_{GDI} - FCC}{VBI(1 - FCC)}\right) & V_{GDI} \geq FCC \end{cases} \quad (3.58)$$

Implementation of the MESFET model in *fREEDA*TM is similar to that used for the Bipolar Junction Transistor. Step 1 is to initialize the state variables. For the MESFET model in Fig. 3.4 two state variables are chosen.

1. $x[0]$ Gate Source Voltage : V_{GS}
2. $x[1]$ Gate Drain Voltage : V_{GD}

To model the MESFET we need the first time derivative of the state variables. A list of the state variables and their equivalent representation in *fREEDA*TM is shown in Table 3.1.

Table 3.1: MESFET state variables

Mesfet	
State Variable	<i>f</i> REEDA™ Representation
V_{GS}	$x[0]$
V_{GD}	$x[1]$
$\partial V_{GS}/\partial t$	$x[2]$
$\partial V_{GD}/\partial t$	$x[3]$
$V_{GS}(t - \tau)$	$x[4]$

Table 3.2: MESFET model output variables

Mesfet	
Output Variable	<i>f</i> REEDA™ Representation
Gate Source Voltage	$vp[0]$
Drain Source Voltage	$vp[1]$
Gate Current	$ip[0]$
Drain Current	$ip[1]$

$x[4]$ is a time delayed state variable generated by ADOL-C [27]. In (3.7) the drain source channel current is proportional to $V_{GS}(t - \tau)$. We substitute the value of $x[4]$ in place of $V_{GS}(t - \tau)$. To our knowledge *f*REEDA™ is the only microwave circuit simulator to physically model the time delayed gate source voltage. Several attempts have been made in the past to model the time delay in Mesfets [28]. All the efforts have concentrated on addition of a active circuit to reciprocate time delay. In *f*REEDA™ we have the luxury of using ADOL-C to generate a time delay. This method has been found to generate results that agree with measured data.

final terminal voltages and currents are computed in the same manner as the bipolar transistor. The final voltages and currents are shown in Table 3.2

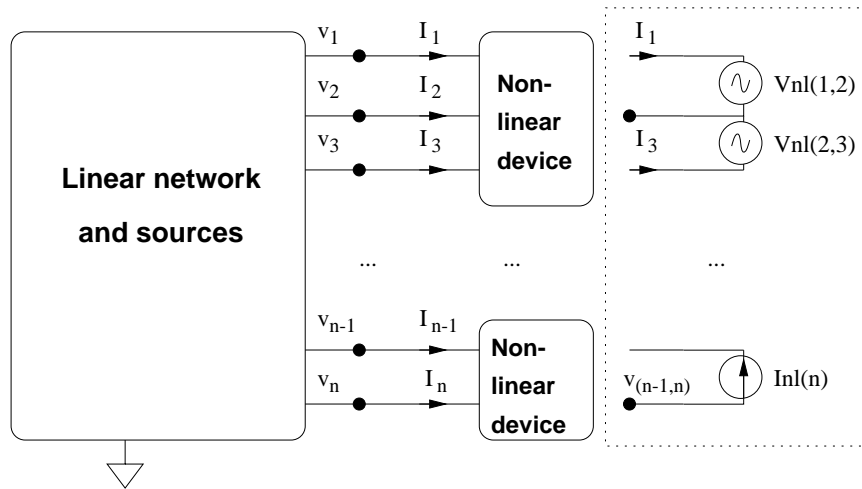


Figure 3.5: Network with nonlinear elements

3.8 Universal Nonlinear Error Function Formulation

The formulation of the system equations begins with the partitioned network of Fig. 3.5 with the nonlinear elements replaced by variable voltage or current sources. For each nonlinear element one terminal is taken as the reference and the element is replaced by a set of sources connected to the reference terminal. Both voltage and current sources are valid replacements for the nonlinear elements, but current sources are more convenient because they yield a smaller modified nodal admittance matrix. (MNAM) [17].

3.8.1 Linear Network

The MNAM of the linear subcircuit is formulated as follows. Define two matrices \mathbf{G} and \mathbf{C} of equal size n_m , where n_m is equal to the number of non-reference terminals i the circuit plus the number of additional required variables. Define a vector \mathbf{s} of size n_m for the right hand side of the system. The contributions of the fixed sources and the non-linear elements (which depend on the time t) will be entered in this vector. All conductors and frequency-independent MNAM stamps arising in the formulation

will be entered in \mathbf{G} , whereas capacitor and inductor values and other values that are associated with the dynamic elements will be stored in matrix \mathbf{C} . The linear system obtained is the following,

$$Gu(t) + C \frac{du(t)}{dt} = s(t). \quad (3.59)$$

where u is the vector of the terminal voltages and required currents. The vector is composed of an independent component s_f and a component s_v that depends on the state variables as in the HB case [6]:

$$s(t) = s_f(t) + s_v(t). \quad (3.60)$$

The s_f vector is due to the independent sources in the circuit. The s_v vector is the contribution of the currents injected into the linear circuit by the nonlinear network.

3.8.2 Nonlinear Network

The equations that describe the behavior of the nonlinear elements in the circuit are shown in Section 3.4. The error function of an arbitrary circuit is developed using connectivity information. The incidence matrix \mathbf{T} is built as follows. The number of columns is n_m , and the number of rows is equal to the number of state variables, n_s . In each row, enter +1 in the column corresponding to the positive terminal of the row nonlinear element port and -1 in the column corresponding to the negative terminal. Then, each row of \mathbf{T} has at most 2 nonzero elements and the number of nonzero elements is at most $2n_2$. The following equations are true for all t :

$$v_L(t) = Tu(t)s_v = T^T i_{NL}(t). \quad (3.61)$$

where V_L is the vector of the port voltages of the nonlinear elements calculated from the nodal voltages of the linear network.

3.8.3 Error Function Formulation

A general equation for the linear network is obtained from (3.59), (3.60), (3.61).

$$Gu(t) + C \frac{\partial u(t)}{t} = s_f(t) + T^T i_{NL}(t). \quad (3.62)$$

The reduced error function $f(t)$ is defined as

$$f(t) = v_L(t) - v_{NL}(t) = 0. \quad (3.63)$$

Replacing $v_L(t)$ from (3.63),

$$f(t) = Tu(t) - v_{NL}(t) = 0. \quad (3.64)$$

(3.59), (3.61) and (3.63) conform the generalized state variable reduction formulation. The error function in (3.62) only depends on the state variables and time,

$$f \left[x(t), \frac{d}{dt}, \dots, \frac{d^n x}{dt^n}, x_D(t), t \right] = 0. \quad (3.65)$$

The dimension of the error function and the number of unknowns are equal to n_s , and this number is the minimum necessary to solve the equations of a circuit without any loss of information. This information is very general and can be applied to derive several types of analysis. The method to approximate $x(t)$ and $u(t)$ and their derivatives will determine the type of analysis.

3.8.4 Time Domain Jacobian

In the time domain, the time derivatives are approximated by a function $h(x_i)$ that depends on the current and previous history of the variable to be derived. Similarly the elements of the time-delayed state variable vector \mathbf{x}_D is calculated by a function $k(x_i)$:

$$\begin{aligned} \frac{dx_i}{dt} &\approx h(x_i) \\ x_i(t - \tau) &\approx k(x_i). \end{aligned}$$

The Jacobians for \mathbf{f}_1 and \mathbf{g}_1 are then

$$\begin{aligned} \mathbf{J}_{f_1} &= \mathbf{J}_{f_1,x} + \mathbf{J}_{f_1,x_D} d\mathbf{k}(\mathbf{x}) \\ \mathbf{J}_{g_1} &= \mathbf{J}_{g_1,x} + \mathbf{J}_{g_1,x_D} d\mathbf{k}(\mathbf{x}), \end{aligned}$$

where $d\mathbf{k}(\mathbf{x})$ is a diagonal matrix where each diagonal element is calculated as $dk(x_i)/dx_i$.

$$d\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \frac{dh}{dx_1} & & & \\ & \frac{dh}{dx_2} & & \\ & & \ddots & \\ & & & \frac{dh}{dx_m} \end{bmatrix},$$

The diagonal matrix $d\mathbf{h}(\mathbf{x})$ is similarly defined. The Jacobian matrices of the form $\mathbf{J}_{y,z}$ are obtained directly from the automatic differentiation routines, *i.e.* they do not need to be coded explicitly.

The Jacobians for \mathbf{f}_n and \mathbf{g}_n are calculated by

$$\begin{aligned} \mathbf{J}_{f_n} &= \mathbf{J}_{f_n, f_{n-1}} \mathbf{J}_{f_{n-1}} + \mathbf{J}_{f_n, g_{n-1}} d\mathbf{h}(\mathbf{g}_{n-1}) \mathbf{J}_{g_{n-1}} \\ \mathbf{J}_{g_n} &= \mathbf{J}_{g_n, f_{n-1}} \mathbf{J}_{f_{n-1}} + \mathbf{J}_{g_n, g_{n-1}} d\mathbf{h}(\mathbf{g}_{n-1}) \mathbf{J}_{g_{n-1}}. \end{aligned}$$

The final Jacobian matrices \mathbf{J}_u and \mathbf{J}_i are obtained in a similar way. It is important to remark that this calculation of the Jacobian is the same for any element and thus it can be implemented outside the actual element routine.

3.8.5 Harmonic Balance Equations

Lets represent the Discrete Fourier Transformation (DFT) with a pre-multiplication by a matrix Γ . Similarly, time delay in the frequency domain is represented by a matrix Δ and time derivation by a matrix Ω . We define $\tilde{\Gamma}$ as

$$\tilde{\Gamma} = \mathbf{I}_{m \times m} \otimes \Gamma,$$

where $\mathbf{I}_{m \times m}$ is an identity matrix of dimension m . A similar definition is used for $\tilde{\Delta}$ and $\tilde{\Omega}$. The vector function $\bar{\mathbf{f}}_1$ is defined as \mathbf{f}_1 evaluated at all the sample points necessary for the DFT. The elements of $\bar{\mathbf{f}}_1$ are arranged to be coherent with the definition of $\tilde{\Gamma}$.

The harmonic balance equations are then written as:

$$\text{stage 1} : \begin{cases} \bar{\mathbf{f}}_1(\tilde{\Gamma}^{-1}\bar{\mathbf{x}}, \tilde{\Gamma}^{-1}\tilde{\Delta}\bar{\mathbf{x}}) \\ \bar{\mathbf{g}}_1(\tilde{\Gamma}^{-1}\bar{\mathbf{x}}, \tilde{\Gamma}^{-1}\tilde{\Delta}\bar{\mathbf{x}}) \end{cases} \quad (3.66)$$

$$\text{stage } 2 : \begin{cases} \bar{\mathbf{f}}_2(\bar{\mathbf{f}}_1, \tilde{\Gamma}^{-1}\tilde{\Omega}\tilde{\Gamma}\bar{\mathbf{g}}_1) \\ \bar{\mathbf{g}}_2(\bar{\mathbf{f}}_1, \tilde{\Gamma}^{-1}\tilde{\Omega}\tilde{\Gamma}\bar{\mathbf{g}}_1) \end{cases} \quad (3.67)$$

$$\vdots$$

$$\text{stage } n - 1 : \begin{cases} \bar{\mathbf{f}}_{n-1}(\bar{\mathbf{f}}_{n-2}, \tilde{\Gamma}^{-1}\tilde{\Omega}\tilde{\Gamma}\bar{\mathbf{g}}_{n-2}) \\ \bar{\mathbf{g}}_{n-1}(\bar{\mathbf{f}}_{n-2}, \tilde{\Gamma}^{-1}\tilde{\Omega}\tilde{\Gamma}\bar{\mathbf{g}}_{n-2}) \end{cases} \quad (3.68)$$

$$\text{stage } n : \begin{cases} \Gamma\bar{\mathbf{v}}(\bar{\mathbf{f}}_{n-1}, \tilde{\Gamma}^{-1}\tilde{\Omega}\tilde{\Gamma}\bar{\mathbf{g}}_{n-1}) \\ \Gamma\bar{\mathbf{i}}(\bar{\mathbf{f}}_{n-1}, \tilde{\Gamma}^{-1}\tilde{\Omega}\tilde{\Gamma}\bar{\mathbf{g}}_{n-1}) \end{cases} . \quad (3.69)$$

As in the transient analysis case, the Jacobian for the above equations can be derived as a function of the primitive Jacobians of the original functions in time domain that can be obtained through automatic differentiation.

Note that this approach allows a separation between the time derivation of variables from the rest of the operations. This separation makes the description of the model equations independent of the type of discretization method used in the circuit analysis. It also enables the efficient use of automatic differentiation because the same set primitive Jacobians are use to derive the specific analysis Jacobian.

3.9 Electro-Thermal Modeling

This section describes a computer environment that supports the simultaneous simulation of thermal and circuit interactions in microwave circuits. The method is based on coupling electrical and thermal environments using a lumped-parameter model of heat dissipation dynamics. Through this technique, simultaneous simulation of electrical and thermal interactions have been achieved [18].

One way of incorporating thermal effects in a circuit simulator is to make the thermal model look like an electrical circuit. The thermal and electrical problems are then solved simultaneously as if they were one large electrical problem. This strategy is based on transforming the thermal problem into an equivalent electrical problem. This concept can be adapted to represent electro-thermal interactions by using thermal terminals, including a local reference node for thermal ground. This is depicted in Fig. 3.6 where a nonlinear electro-thermal device including both electrical

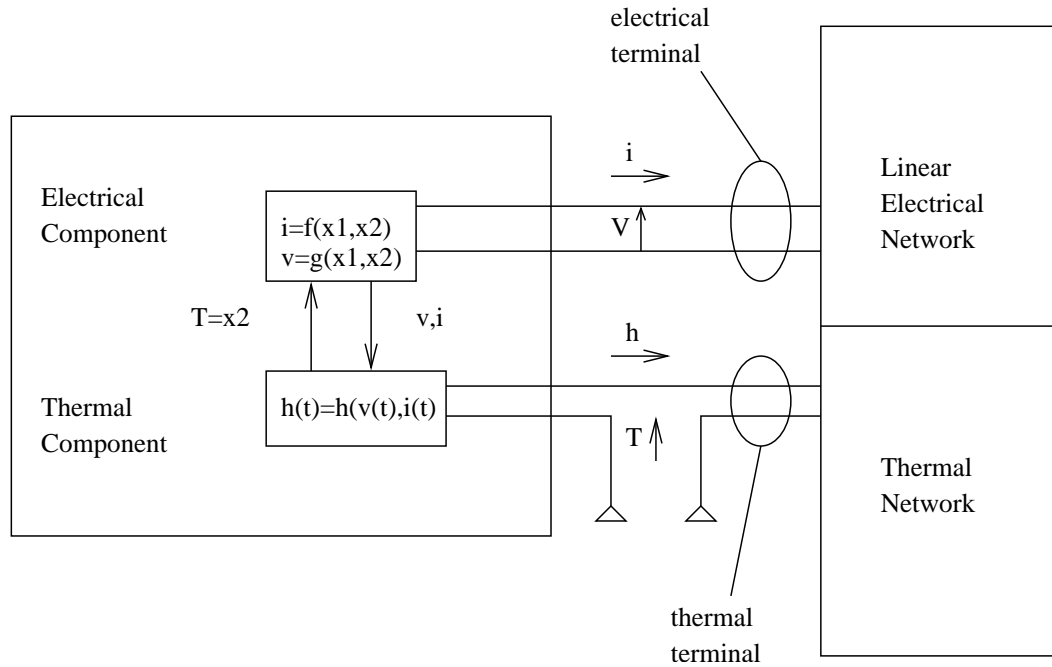


Figure 3.6: Nonlinear electro-thermal device

and thermal terminals is shown. Power dissipated in the active device is represented as heat current source referenced to thermal ground.

The state variable at the thermal terminals is device temperature and the associated error function is derived from heat current. The network representation allows efficient simulation of electro-thermal interactions in steady-state. The temperature is calculated simultaneously with the electrical quantities since it becomes an additional element of the system state vector. The procedure begins by identifying thermal terminals that interface heat dissipating electrical components with the model of the thermal load. In circuits where the thermal behavior is linear the thermal load can be represented by a thermal admittance matrix, Y_{TH} , so that

$$\begin{bmatrix} Y_E & 0 \\ 0 & Y_{TH} \end{bmatrix} \begin{bmatrix} X \\ T \end{bmatrix} = \begin{bmatrix} i \\ h \end{bmatrix}$$

where Y_E corresponds to the electrical modified nodal admittance matrix, X is the vector of electrical state variables, T the vector of unknown temperature, and i, h are the corresponding electrical and thermal current vectors. Thus the thermal behavior

of the system is represented by a thermal admittance matrix.

The method described above was used to model a BJT electro-thermal device in *fREEDA*[™]. The results obtained by simulating the electro-thermal BJT are shown in Section 4.5.

Chapter 4

Results

4.1 Introduction

Chapter 4 describes the various test netlists used to validate the universal modeling claim. The results shown in this chapter were obtained from *fREEDA*TM. Section 4.2 shows an example of non-conservation of charge in a varactor diode simulation. A comparison is made between the convergence values obtained by using the universal model formulation technique and the voltage based simulation method. Section 4.3 describes the amplifier netlists used to validate the universal parameterized device model. The results obtained for the BJT and MESFET amplifier circuits made use of the same device model code for different analysis routines. Section 4.4 describes the set up of the amplifier circuit used for the two tone harmonic balance analysis. The results obtained in *fREEDA*TM were identical to those obtained from measurements. The effect of modeling time delay in estimating intermodulation is discussed. Section 4.5 shows the results obtained for electro-thermal simulation of transistor circuits. The outputs of temperature against time and power dissipation are used to distinguish *fREEDA*TM from other circuit simulators.

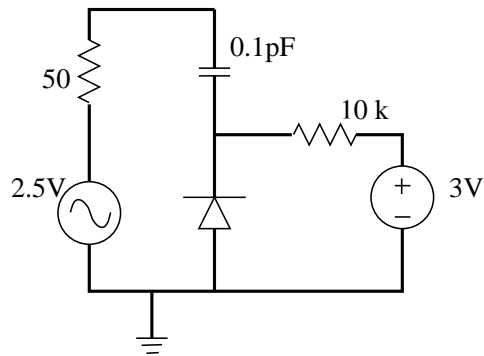


Figure 4.1: Varactor Circuit

4.2 Varactor Non-Conservation Of Charge

We present an illustration of the dramatic loss of accuracy that can result from non conservation of charge taking the circuit of Fig. 4.1 as an example. The simulations were performed in *fREEDA*[™].

Fig. 4.2 compares two identical transient simulation results using the diode model with a capacitor-based model and with a charge-based model. Table 4.1 shows that reducing the time step tends to reduce the error due to nonconservation of charge.

Table 4.1: Comparison of the numerical error due to non-conservation of charge as a function of the time step.

Time Step (ps)	Numerical Error (V)
20	.917
10	.533
5	.295
2.5	.161

This type of accumulation of numerical error is not present in other analysis types. For example the harmonic balance simulation produces exactly the same result with either diode model.

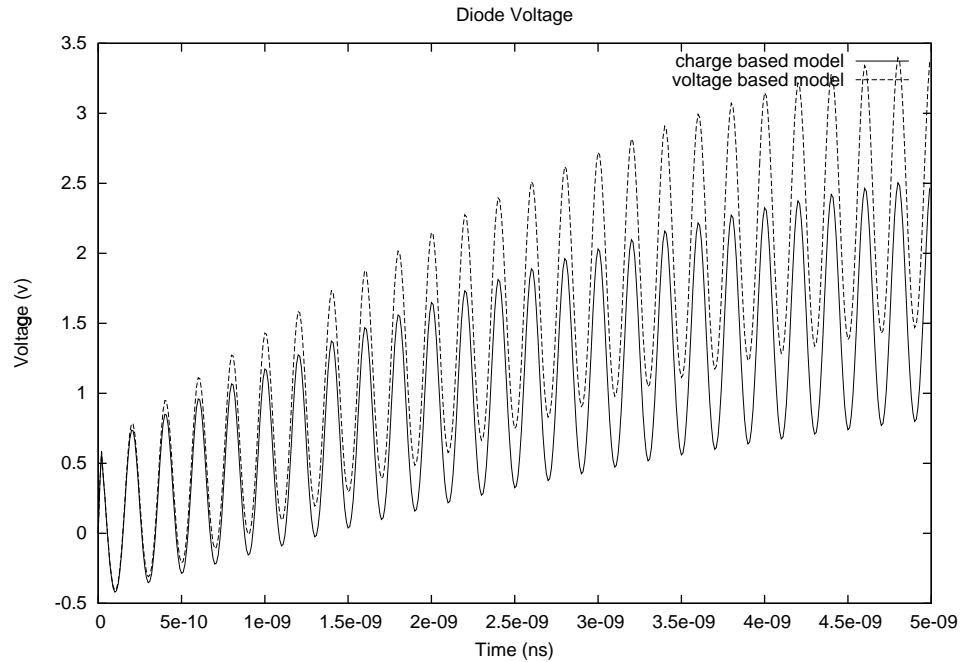


Figure 4.2: Comparison of the simulation using the capacitor and charge based models.

4.3 Universal Modeling

4.3.1 Bipolar Junction Transistor

The circuits used to validate the Universal Modeling algorithm are *MCNC* benchmarks. The Wide Band Amplifier simulation for performed to test the validity of the *BJT* model implemented in *fREEDA*TM. The circuit was simulated in *fREEDA*TM and *Spice*. The output waveforms and voltage amplitudes were found to match, hence validating the Gummel-Poon model implemented in *fREEDA*TM. The netlist for the Wide Band Amplifier circuit is provided in Appendix B.

The amplifier circuit consists of 14 Gummel-Poon Bipolar Transistors. The amplifier circuit was simulated for a period of 250 μ s. The output waveforms obtained in *fREEDA*TM Fig. 4.6 and *Spice* Fig. 4.5 are in excellent agreement.

In Section 3.4 the idea of universal modeling was described in detail using the example of a Gummel-Poon bipolar transistor. The theory was to model a device in a

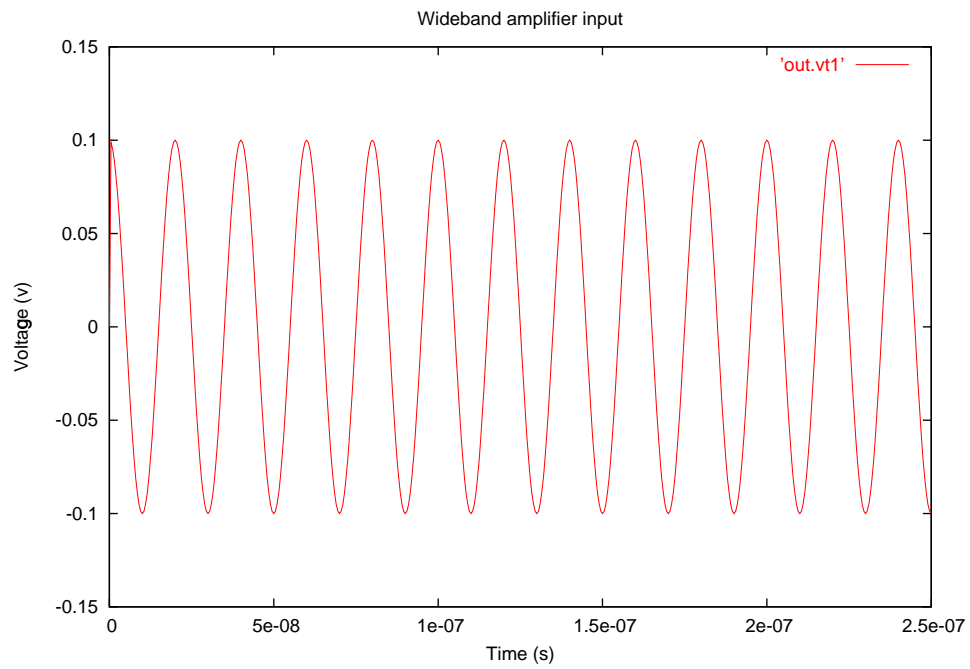
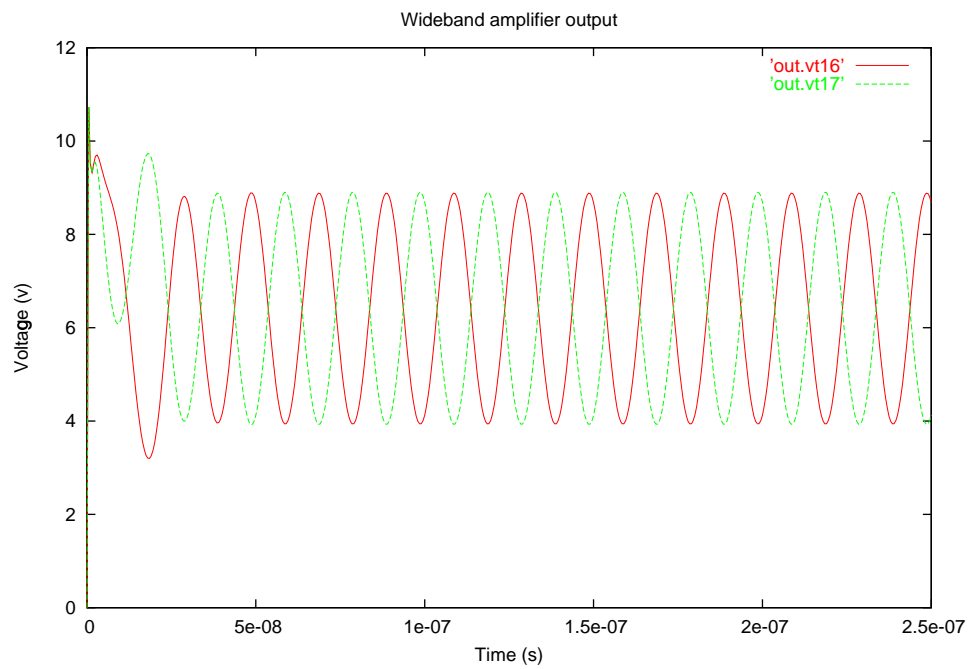


Figure 4.3: Input waveform.

Figure 4.4: Wideband amplifier output in *fREEDA*[™].

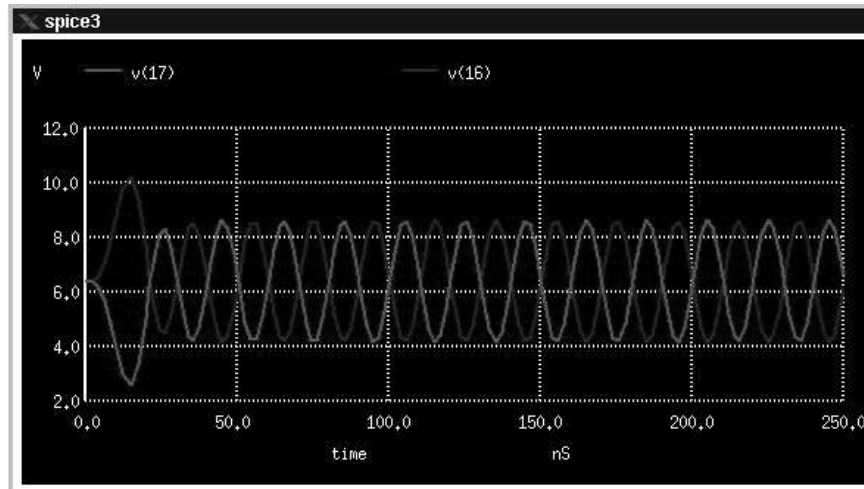


Figure 4.5: Wideband amplifier output in Spice.

circuit simulator independent of analysis routines (i.e. Transient, Harmonic Balance). The theory was validated by simulating a common-emitter bipolar transistor amplifier circuit shown in Fig. 4.7. Both transient and harmonic balance routines used to simulate the amplifier circuit make use of the same transistor device code. This was made possible due to the universal model proposed in this thesis.

The final result shows the I - V characteristics of the BJT. The circuit shown in Fig. 4.8 was used to test the DC characteristics using the universal Gummel-Poon model.

In Fig. 4.8 the collector terminal is connected to varying voltage source, V_C , the base of the transistor is biased by a current source, I_B . The collector current, I_C , is plotted as a function of V_{CE} . The collector current is plotted for various values of I_B to generate an I - V curve shown in Fig. 4.9.

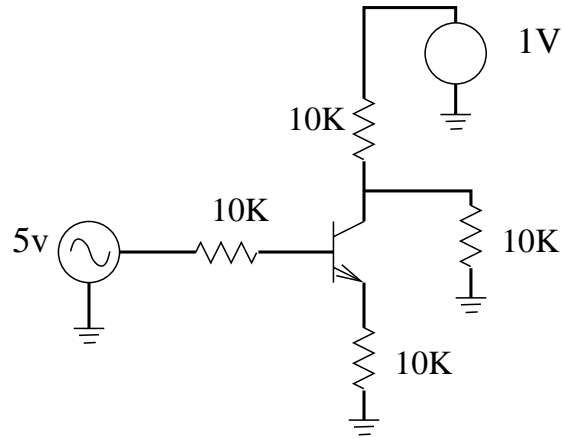


Figure 4.6: BJT amplifier test circuit.

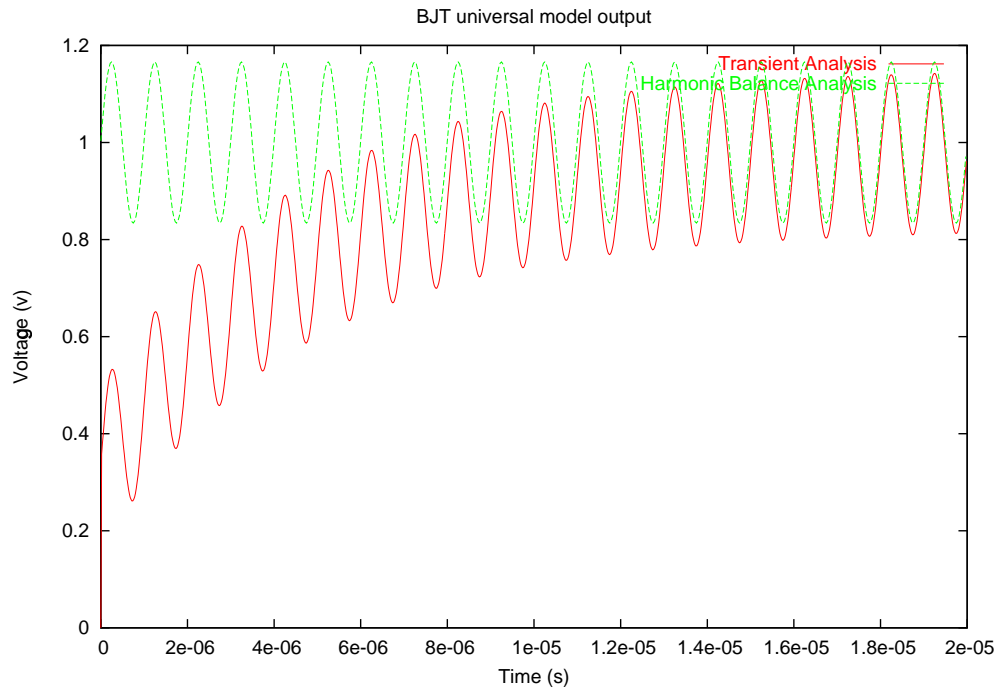


Figure 4.7: BJT universal model output.

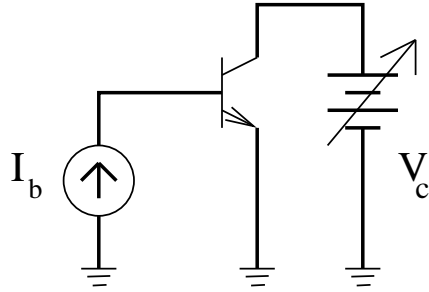


Figure 4.8: BJT common-emitter circuit to test the DC characteristics.

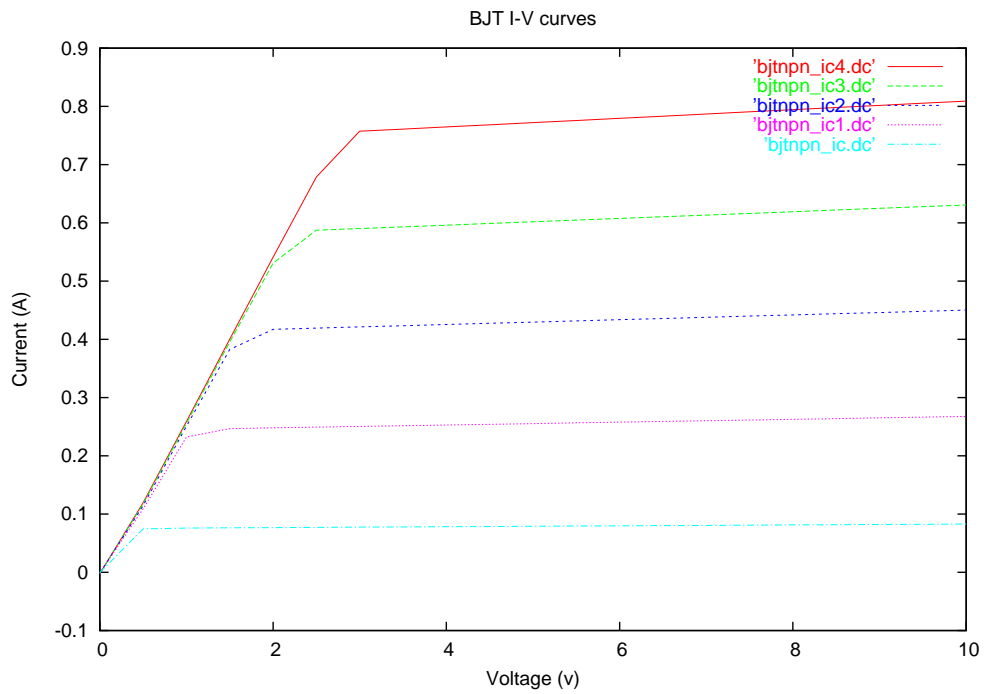


Figure 4.9: BJT I-V curves generated at different values of bias current.

4.3.2 MESFET

The MESFET (Materka Kacprzak) model [24] was used to illustrate universal modeling. The common source Mesfet circuit shown in Fig. 4.10 was simulated for three different analysis routines.

1. Harmonic Balance
2. Transient
3. Convolution Transient

The output waveforms are exactly the same using all three analysis routines. This validates the claim that at single device model can be used for different analysis routines. The output in Fig. 4.11 is a plot of voltage versus time for three different analysis. The results obtained for the three analysis routines were identical.

4.4 Two Tone Analysis

The Curtice-Ettenberg Mesfet model was modified to fit the data obtained from [25]. The simulation results for a two-tone input of the MESFET Amplifier are compared with the experimental data. Results of the *fREEDA*TM simulation using the universal mesfet model were compared to measurements for two tone excitation. For two tone excitation two equal amplitude signals input at 2.35GHz and 2.4 GHz were used. The output power at the fundamental (2.35 GHz) and the third harmonic (2.3 GHz) as a function of input power are shown in Fig. 4.13, Fig. 4.14. The results show that the predicted gain compression of the fundamental power in Fig. 4.13 can closely match the measured gain compression. Comparisons were made by sweeping the input power from -10 dBm to $+10$ dBm. The netlist used to generate the test data is shown in Appendix B.

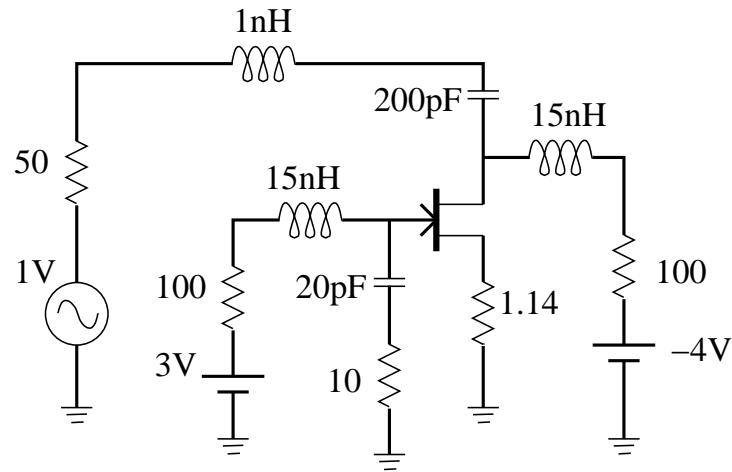


Figure 4.10: MESFET circuit used for Universal Simulation.

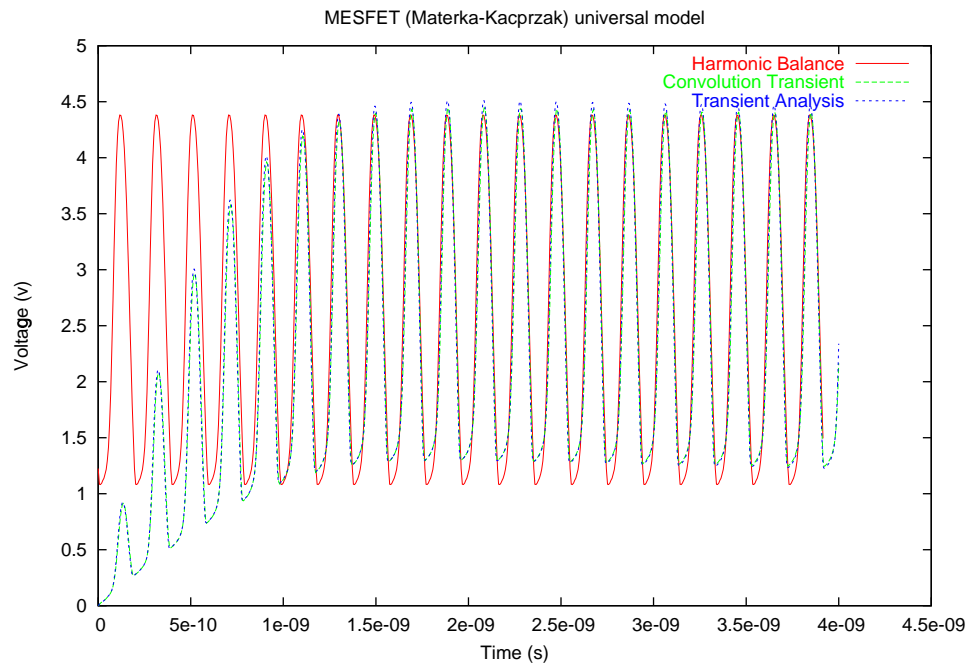


Figure 4.11: Universal Modeling output.

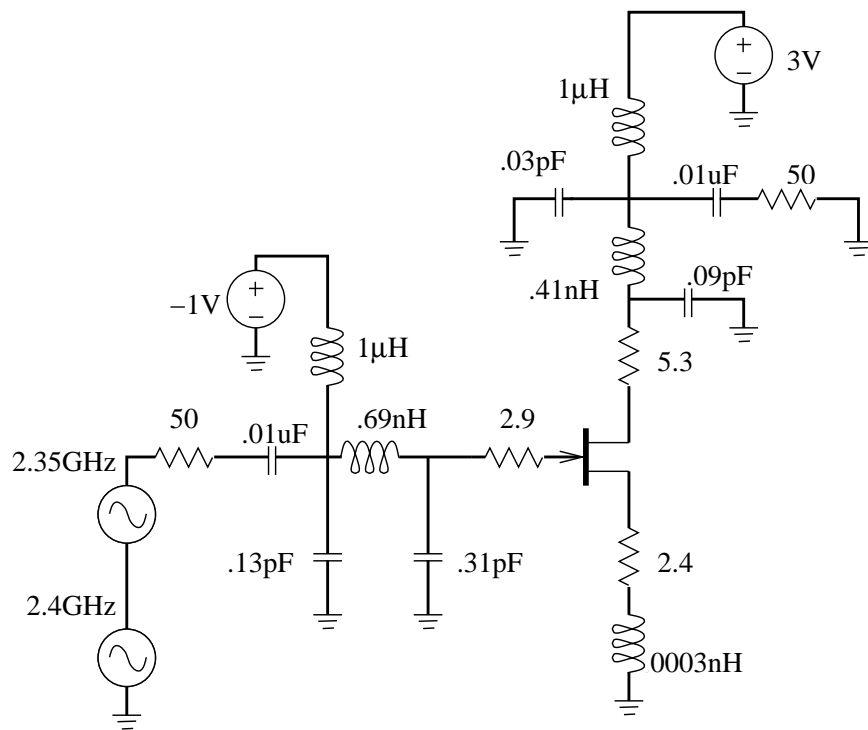


Figure 4.12: MESFET amplifier circuit used to simulate the two-tone test.

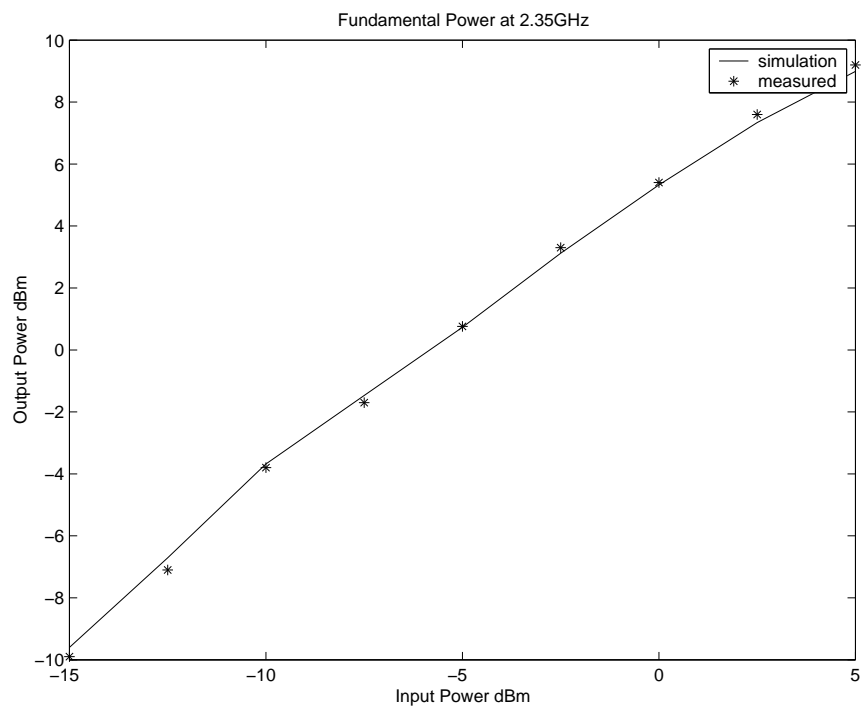


Figure 4.13: Results of the two-tone test. A plot of output power at the fundamental frequency as a function of input power.

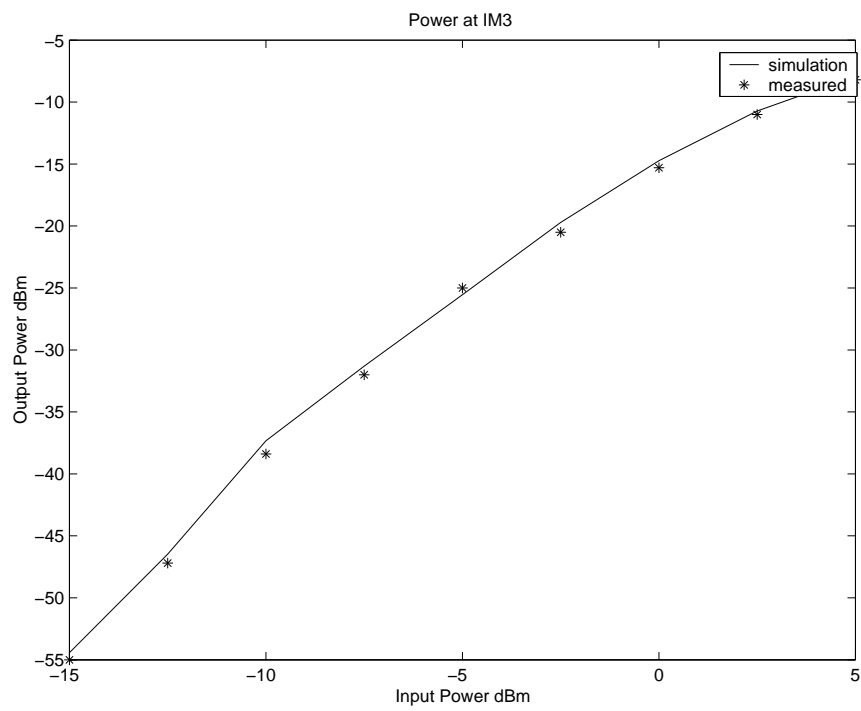


Figure 4.14: A plot of the output power at the third harmonic as a function of the input power.

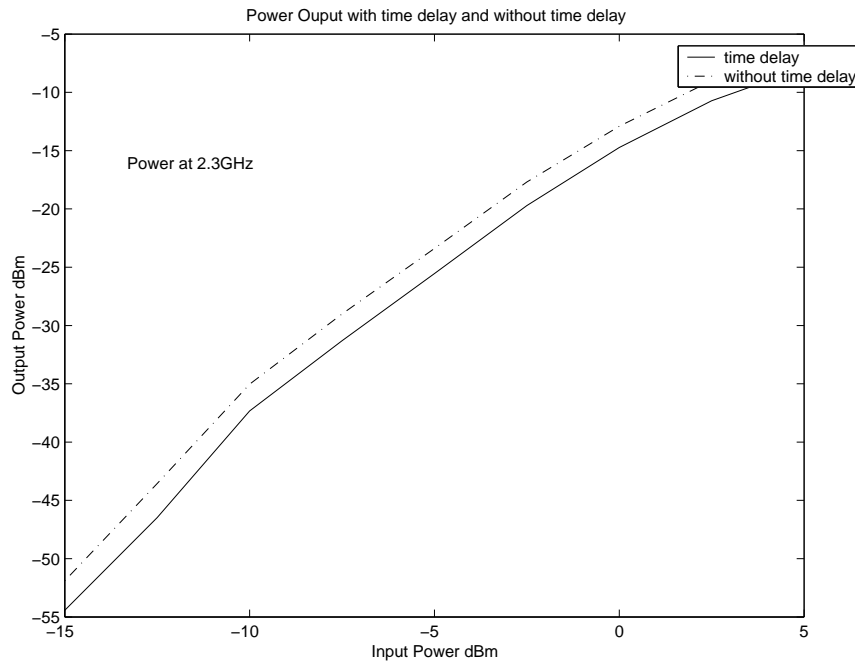


Figure 4.15: A plot of the output power at the third harmonic as a function of the input power with $\tau = 6.56$ ps and $\tau = 0$.

4.5 Electrothermal Simulation

The techniques outlined in Section. 3.9 were validated by evaluating thermal dissipation in a common emitter thermal BJT amplifier. The input to the amplifier is an a.c voltage source. The results shown in Fig. 4.16 illustrate an important point. The sources of heat current in a microwave circuit have a period corresponding to the input signal, thus a self-consistent electro-thermal simulation must permit tightly integrated modeling of the thermal network with the electrical network. The temperature variations shown indicate the effect of thermal damping as there is very little variation of temperature at the input frequency. Nevertheless, the variation is still present.

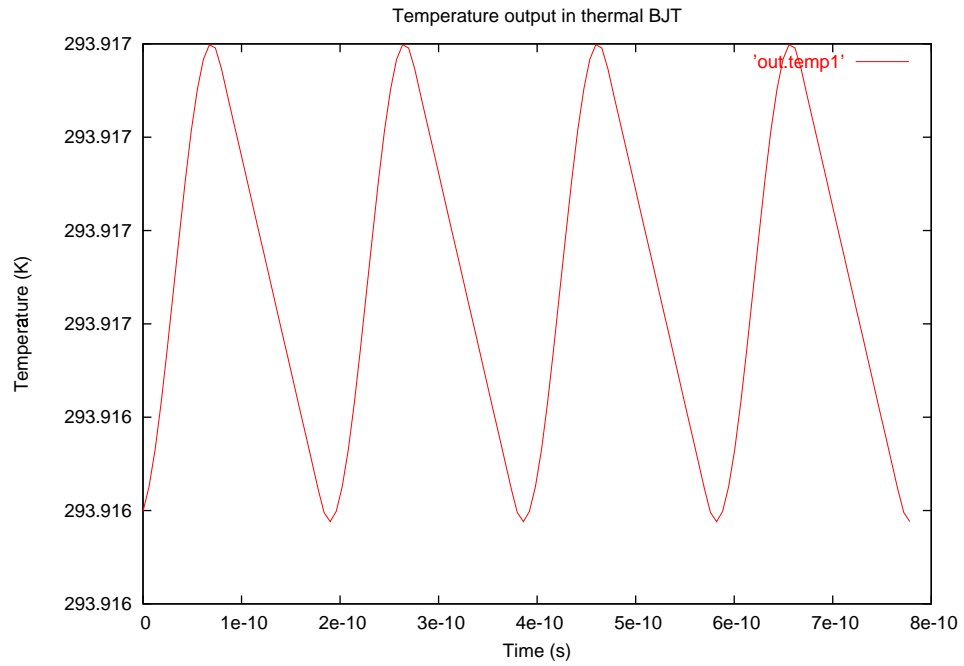


Figure 4.16: Temperature variation in the thermal BJT.

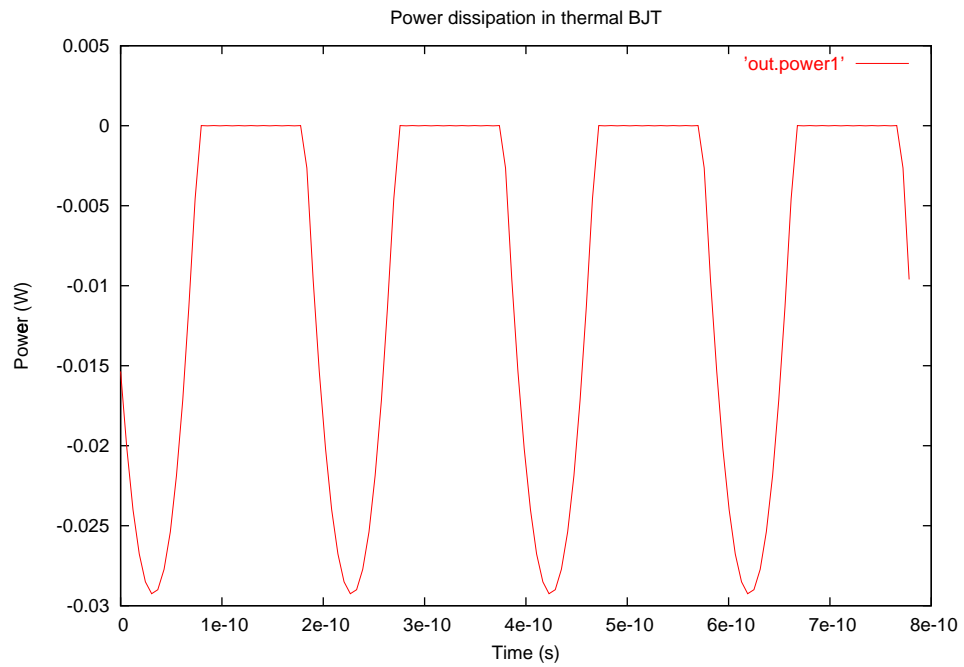


Figure 4.17: Power dissipation in the thermal BJT.

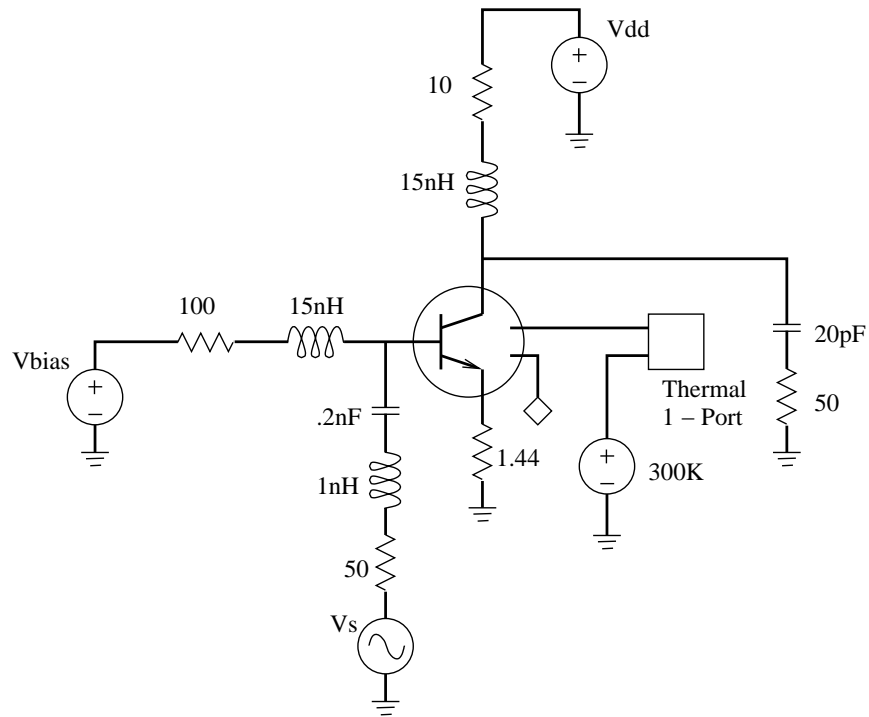


Figure 4.18: Electro-thermal BJT amplifier circuit.

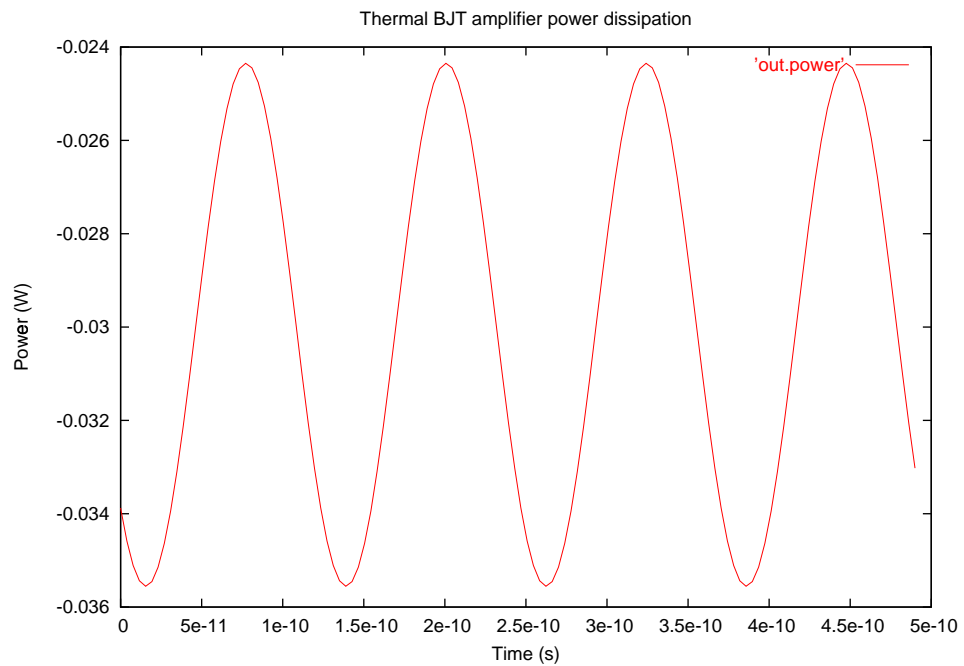


Figure 4.19: Power dissipation in the thermal BJT amplifier.

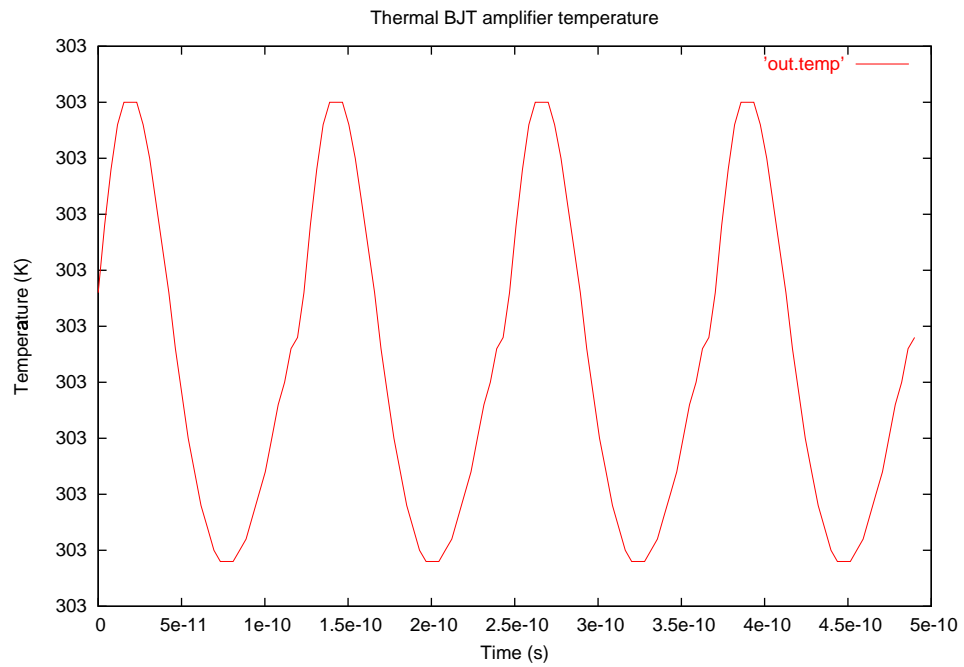


Figure 4.20: Temperature variation in the amplifier circuit.

Chapter 5

Conclusions and Future Research

5.1 Summary Of Research

The important contribution of this work is the universal parameterized nonlinear device model formulation. This formulation enables a unique description of the model in any circuit analysis type and provides a mechanism to describe complex charge-based models without adding extra state variables to the nonlinear system of equations in the main simulation algorithm. In some cases it also simplifies the equations that must be coded in the model implementation. The parametric description provides great flexibility for the design of nonlinear device models. The number of parameters or state variables required is the minimum necessary and they can be chosen to achieve robust numerical characteristics. We have demonstrated the numerical errors that may arise from a capacitance-based nonlinear device model in the transient simulation of microwave circuits. The necessary support for the description of nonlinear elements using the new formulation has been implemented in the *fREEDA*TM circuit simulator. This capacity is fundamental to reducing the development time of new system component models. To this date no other circuit simulator provides the same level of flexibility for the addition of new nonlinear device models and circuit analysis algorithms.

Other contributions include modeling the Avantek MESFET described in [25].

Harmonic balance analysis routines were used to conduct a two tone analysis and hence characterize the importance of time delay (τ) parameter in the MESFET model. The results obtained using simulations in *fREEDA*TM were in excellent agreement with the measured data [29].

Other contributions include the implementation of the charge based BJT and MESFET models. The electro-thermal model of the bipolar transistor was implemented in *fREEDA*TM. The simulation results in Section 4.5 show that thermal effects cannot be neglected.

5.2 Future Research

An issue at microwave and high frequencies is the development of high power solid state sources. This is an enabling technology for a variety of military and commercial systems. Given the current state of maturity of Monolithic Microwave Integrated Circuits (*MMIC*) technology, the quest to produce high powers from solid state devices requires the investigation of novel semiconductor materials and power combining architectures.

The power amplifier under scrutiny is the low noise *PHEMT LMA411* MMIC. The Filtronic Solid State LMA411 is a high dynamic range low noise PHEMT amplifier that operates from 8.5 to 14 GHz. Reactively matched 2-stage amplifier provides 18 dB nominal gain with 2 dB typical noise figure and 1-dB gain compression power output of +17 dBm that can be used as a pre-driver amplifier for phased array radar as well as commercial communications applications. Ground is provided to the circuitry through vias to the backside metallization. The layout shown in Fig. 5.1 contains several microwave elements, (i.e.) spiral inductors, transmission lines, etc. The main focus of future research is to self-consistently model and simulate the interrelated effects of the EM field, the linear and nonlinear circuit elements and the thermal subsystems present in quasi-optical systems.

Bibliography

- [1] R. Kielkowski, *Inside Spice*, McGrawHill Publications, 1998.
- [2] W. Liu, *Mosfet Models for Spice Simulation including BSIM3 and BSIM4*, Wiley Interscience Publications, 2001.
- [3] C. E. Christoffersen, *Global Modeling of Nonlinear Microwave Circuits*, Ph.D. Dissertation, Dept. Of North Carolina State University, North Carolina State University, 2001.
- [4] C.E.Christoffersen, S. Velu and M. B. Steer, "Universal Parameterized Nonlinear Device Model Formulation for Microwave Circuit Simulation," *2002 IEEE MTT-S Int Microwave Symposium*, June 2002.
- [5] J.Ogrodzky, *Circuit Simulation Methods and Algorithms*, CRC Press, 1994.
- [6] C. E. Christoffersen, M. B. Steer and M. A. Summers, "Harmonic balance analysis for systems with circuit-field interactions," *IEEE MTT-S Int. Microwave Symp. Dig.*, June 1998, pp. 1131–1134.
- [7] 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, Jan 1992, pp. 12–28.
- [8] M. Valtonen and T. Veijola, "A microcomputer tool especially suited for microwave circuit design in frequency and time domain," *Proc. URSI/IEEE National Convention on Radio Science*, Espoo, Finland, 1986, p. 20.

- [9] M. Valtonen, P. Heikkilä, A. Kankkunen, K. Mannersalo, R. Niutanen, P. Stenius, T. Veijola and J. Virtanen, "APLAC - A new approach to circuit simulation by object orientation," *10th European Conference on Circuit Theory and Design Dig.*, 1991.
- [10] K. Mayaram and D. O. Pederson, "CODECS: an object-oriented mixed-level circuit and device simulator," *1987 IEEE Int. Symp. on Circuits and Systems Digest*, 1987, pp. 604–607.
- [11] A. Davis, "An object-oriented approach to circuit simulation," *1996 IEEE Midwest Symp. on Circuits and Systems Dig.*, 1996, pp. 313–316.
- [12] B. Melville, P. Feldmann and S. Moinian, "A C++ environment for analog circuit simulation," *1992 IEEE Int. Conf. on Computer Design: VLSI in Computers and Processors.*, 1992, pp. 516–519.
- [13] P. Carvalho, E. Ngoya, J. Rousset and J. Obregon, "Object-oriented design of microwave circuit simulators," *1993 IEEE MTT-S Int. Microwave Symp. Digest*, June 1993, pp. 1491–1494.
- [14] M. B. Steer, C. Chang and G. W. Rhyne, "Computer-Aided Analysis of Nonlinear Microwave Circuits Using Frequency-Domain Nonlinear Analysis Tech.: The State of the Art," *Int. Journal of Microwave and Millimeter-Wave Computer-Aided Engineering*, Vol. 1, No. 2, 1991, pp. 181–200.
- [15] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*, Van Nostrand Reinhold, 1994.
- [16] C. E. Christoffersen, S. Nakazawa, M. A. Summers, and M. B. Steer, "Transient analysis of a spatial power combining amplifier", *1999 IEEE MTT-S Int. Microwave Symp. Dig.*, June 1999, pp. 791–794.
- [17] R. S. Bain, *NNES User's Manual*, 1993.
- [18] H. Gutierrez, C. E. Christoffersen and M. B. Steer, "An integrated environment for the simulation of electrical, thermal and electromagnetic interactions in

- high-performance integrated circuits,” *Proc. IEEE 6 th Topical Meeting on Electrical Performance of Electronic Packaging*, Sept. 1999, pp. 217–220.
- [19] C. E. Christoffersen and M. B. Steer “Implementation of the local reference concept for spatially distributed circuits,” *Int. J. of RF and Microwave Computer-Aided Eng.*, Vol. 9, No. 5, 1999.
- [20] V. Rizzoli, F. Mastri, F. Sgallari and G. Spaletta, “Harmonic-Balance Simulation of Strongly Nonlinear very Large-Size Microwave Circuits by Inexact Newton Methods,” *IEEE MTT-S Int. Microwave Symp. Dig.*, Vol. 3 , 1996, pp. 1357–1360.
- [21] V. Rizzoli, A. Costanzo, and A. Lipparini, “An Electrothermal Functional Model of the Microwave FET Suitable for Nonlinear Simulation,” *Int. Journal of Microwave and Millimeter-Wave Computer-Aided Engineering*, Vol. 5, No. 2, 1995, pp. 104-121.
- [22] V. Rizzoli, A. Lipparini, A. Costanzo, F. Mastri, C. Ceccetti, A. Neri and D. Masotti, “State-of-the-Art Harmonic-Balance Simulation of Forced Nonlinear Microwave Circuits by the Piecewise Technique,” *IEEE Trans. on Microwave Theory and Tech.*, Vol. 40, No. 1, Jan 1992, pp. 12–28.
- [23] M.S.Nakhla and J.Vlach, “A Piecewise Harmonic Balance Technique for Determination of Periodic Response of Nonlinear Systems,” *IEEE Trans. on Circuits and Systems*, Vol. 23, No. 2, Feb 1976.
- [24] A. Materka and T. Kacprzak, “Computer Calculation of Large-Signal GaAs FET Amplifier Characteristics,” *IEEE Trans. on Microwave Theory and Tech.*, Vol MTT-33, No. 2, Feb. 1985.
- [25] C. R. Chang, *Computer-Aided Analysis of Nonlinear Microwave Analog Circuits Using Frequency-Domain Spectral Balance*, Ph.D. Dissertation, Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, 1990.

- [26] C. E. Christoffersen, U. A. Mughal and M. B. Steer, "Objectoriented microwave circuit simulation," *Int. Journal of RF and Microwave Computer-Aided Engineering*, Vol. 10, No. 3, 2000, pp. 164–182.
- [27] A.Griewank, D.Juedes and J.Utke, "Adol-C A Package for the Automatic Differentiation of Algorithms Written in C/C++," *ACM TOMS*, Vol. 22(2), June 1996, pp. 131-167.
- [28] W.R.Curtice, "Microwave Theory and Techniques," *IEEE Transactions*, Vol. 36, Feb. 1988, pp.
- [29] George W. Rhyne, *Nonlinear Analysis of Microwave Circuits*, Ph.D. Thesis, Department of Electrical and Computer Engineering, North Carolina State University, 1988.

Appendix A

Bipolar Junction Transistor Model Parameters

Table A.1 lists the 50 parameters used to model the BJT in *fREEDA*TM. The model parameters that are scaled by the Area element parameter are designated in the Area column.

Table A.1: Parameters of the Bipolar Junction Transistor.

Bipolar Junction Transistor - Parameters				
Name	Description	Units	Default	Area
AF	flicker noise exponent	-	1	
BF	ideal forward maximum beta	-	1	
BR	ideal maximum reverse beta	-	100	
CJC	base collector zero bias depletion capacitance	F	0	★
CJE	base emitter zero bias depletion capacitance	F	0	★
CJS	zero bias collector substrate capacitance	F	0	★
EG	energy gap voltage	eV	1.11	
FC	co-efficient for forward bias depletion capacitance formula	-	0.5	

Table A.2: Continued.

Bipolar Junction Transistor - Parameters				
Name	Description	Units	Default	Area
ISC	base collector leakage saturation current	A	0	★
ISE	base emitter leakage saturation current	A	0	★
ISS	substrate p-n junction saturation current	A	★	
ITF	high current parameter for effect on TF	A	0	★
KF	flicker noise coefficient	-	0	
MJC	base collector junction exponential factor	-	0.33	
MJE	base emitter junction exponential factor	-	0.33	
MJS	substrate junction exponential factor	-	0.33	
NC	base collector leakage emission coefficient	-	2	
NE	base emitter leakage coefficient	-	1.5	
NF	forward current emission coefficient	-	1.0	
IKF	corner of high current beta roll-off	A	∞	★
IKR	corner of reverse beta high current roll-off	A	∞	★
IRB	current where base resistance falls halfway to its minimum value	-	∞	★
IS	transport saturation current	A	$1.0e^{-16}$	★

Table A.3: Continued.

Bipolar Junction Transistor - Parameters				
Name	Description	Units	Default	Area
RC	collector resistance		0	*
RE	emitter resistance		0	*
TF	ideal forward transit time	s	0	
TR	ideal reverse transit time	s	0	
TRB1	RB linear temperature coefficient	-	1	*
TRB2	RB quadratic temperature coefficient	-	1	*
TRM1	RBM linear temperature coefficient	-	1	*
TRM2	RBM quadratic temperature coefficient	-	1	*
VAF	forward early voltage	V	∞	
VAR	reverse early voltage	V	∞	
VJC	base collector built-in potential	V	0.75	
VJE	base emitter built-in potential	V	0.75	
VJS	substrate junction built-in potential	V		
VTF	voltage describing V_{BC} dependence of TF	V	∞	
NR	reverse current emission coefficient	-	1	
NS	substrate p-n emission coefficient	-	1	
PTF	excess phase at frequency= $1.0/(2\pi TF)$	degree	0	

Table A.4: Continued.

Bipolar Junction Transistor - Parameters				
Name	Description	Units	Default	Area
RB	zero bias base resistance		0	*
RBM	minimum base resistance at high current		0	*
XCJC	fraction of B-C depletion capacitance connected to internal base node	-	1	
XTB	forward and reverse beta temperature exponent	-		
XTI	temperature exponent for effect on IS	-	3	
XTF	coefficient for bias dependence of TF	-		

Appendix B

MESFET Model Parameters

Table B.1: Parameters of the MESFET model.

Curtice-Ettenberg Model Parameters			
Name	Description	Units	Default
A0	drain saturation current for $V_{GS} = 0$	amp	0.1
A1	coefficient of V_X	amp/volt	0.05
A2	coefficient of V_X^2	amp/volt ²	0
A3	coefficient of V_X^3	amp/volt ³	0
BETA	V_X dependance on V_{DS}	/volt	0
VDS0	V_{DS} at which β was measured	volt	4.0
GAMA	slope of drain characteristic in the linear region	/volt	1.5
VT	voltage at which the channel current is forced to be zero for $V_{GS} \leq VT0$	volt	$-\infty$
CGS0	gate-source schottky barrier capacitance for $v_{GS} = 0$	farad	0
CGD0	gate-drain schottky barrier capacitance for $V_{GD} = 0$	farad	0
MGS	gate-source grading coefficient		0.5

Table B.2: Continued.

Curtice-Ettenberg Model Parameters			
Name	Description	Units	Default
MGD	gate-drain grading coefficient		0.5
IS	diode saturation current	amp	0
N	diode ideality factor		1.0
IB0	breakdown saturation current	amp	0
NR	breakdown ideality factor		10
VBD	breakdown voltage	volt	∞
GMAX	breakdown conductance	amp/volt	0
K1D	fitting parameter	/volt	0
K2D	fitting parameter	volt	0
K3D	fitting parameter	volt ²	0
VBD	breakdown voltage	volt	∞

Appendix C

Freeda Circuit Netlist

C.1 Bjt Universal Modeling

C.1.1 Transient Analysis

The transient analysis netlist was used to generate results shown in Fig. 4.7.

```
* Transient Analysis of a C-E Amplifier
.options freq = 1e6

***** Voltage Source *****
vsource:v1 1 0 f= freq
vac=.5 phase=-90. vsource:vcc 5 0 vdc=1.
*****

rs 1 2 10k r1 4 5 10k *rb 4 0 10k r2 3 0 10k

***** Bjt Model Statement ***** .model qstd
bjtnpn(is=1e-16 cje=.4nf cjc=.4nf rb=50 vjc=.8 tf=.12ns tr=5ns
+mje=.8 mjc=.33 vaf=50 br=50 rc=10)

bjtnpn:qq1 4 2 3 0 model="qstd"
*****

***** Transient Analysis *****
.tran2 tstop=20us
tstep=.02us im=0 out_steps=100
```

```

*****

***** Transient Analysis Output *****

.out plot term 4vt in "out.vt4" .out plot term 5 vt in "out.vt5"
.out plot term 1 vt in "out.vt1" .out plot term 3 vt in "out.vt3"
*****

.end

```

C.1.2 Harmonic Balance Analysis

The harmonic balance netlist was used to generate results shown in Fig. 4.7.

```

*Harmonic Balance Analysis of a C-E Amplifier

.options freq = 10e9 output=0

***** Voltage Source *****
vsource:v1 1 0 f= freq
vac=.5 phase=-90. vsource:vcc 5 0 vdc=1.
*****

rs 1 2 10k r1 4 5 10k *rb 4 0 10k r2 3 0 10k

***** Bjt Model Statement *****

.model qstd bjtnpn(is=1e-16 cje=.4nf cjc=.4nf rb=50 vjc=.8
+tf=.12ns tr=5ns mje=.8 mjc=.33 vaf=50 br=50 rc=10)

bjtnpn:qq1 4 2 3 0 model="qstd"

*****

***** Harmonic Balance Analysis
***** .svhb n_freqs = 15 fundamental = freq
deriv=0 steps=0
*****

```

```
***** Harmonic Balance Output
***** .out plot term 4 vf invfft 20
repeat in "out.vt4hb" .out plot term 5 vf term 0 vf sub invfft in
"out.vt5" .out plot term 1 vf invfft 20 repeat in "out.vt1hb"
*****
.end
```

C.1.3 Bjt DC Analysis

The DC netlist was used to generate the results shown in Fig. 4.9.

```
*Generate dc sweep

.options ib = 400e-6

.dc sweep="vsource:vs1" start=0. stop=10. step=0.5 vsource:vs1 2 0
isource:ibias 1 0 idc= ib .model qst bjtnpn (is=5.25e-14 bf=190.
vaf=100. rc=3.4 rb=37.) bjto:qq1 2 1 0 0 model="qst" .out plot
+element "bjto:qq1" 0 it in "bjto_ic.dc"
.end
```

C.2 Bjt Thermal Netlist

The electro-thermal netlist was used to generate results shown in Fig. 4.19.

```
*** Sample netlist for a Bjt with power dissipation ***

.options f0 = 8.1e9 jupdm=4 output=0

.svhb n_freqs = 15 fundamental = f0 steps=0 deriv=1

ind:l1 1 3 l=1e-9 time_d=0

ind:l2 3 7 l=15e-9 time_d=0 res:r2 7 8 r=100k

ind:l3 4 5 l=15e-9 time_d=0 res:r3 5 6 r=10 cap:cloud 4 9 c=20e-12
```

```

time_d=0 res:rload 9 0 r=50.

vsource:vbias 8 0 vdc = .5 vsource:vdrain 6 0 vdc = .3 res:rin 11
1 r = 50 vsource:vs 11 0 f = f0 vac = .4

.model qstd tbjto (is=1e-16 cje=.4nf cjc=.4nf rb=50 vjc=.8
tf=.12ns tr=5ns + mje=.8 mjc=.33 vaf=50 br=50 rc=10)

*** Bjt with power dissipation tbjto:m1 3 4 0 0 1000 "tref"

*** Thermal circuit res:rtherm 1000 1001 r=2. cap:ctherm 1000 1001
c=.1e-9. vsource:tabs 1001 "tref" vdc=303. .ref "tref"

.out plot term 4 vf invfft 4 repeat in "out.vds" .out plot term
1000 vf invfft 4 repeat in "out.temp" .out plot element "tbjto:m1"
1 if invfft 4 repeat in "out.ids" .out plot element "tbjto:m1" 2
if invfft 4 repeat in "out.power"

.end

```

C.3 BJT Wide Band Amplifier

The wideband amplifier netlist was used to generate the results shown in Fig. 4.6.

```

*rca netlist
* Wide Band Amp.

rs1 30 1 1k rs2 31 0 1k r1 5 3 4.8k r2 6 3 4.8k r3 9 3 811 r4
8 3 2.17k r5 8 0 820 r6 2 14 1.32k r7 2 12 4.5k r8 2 15 1.32k
r9 16 0 5.25k r10 17 0 5.25k bjtnpn:q1 2 30 5 0 model="qnl"
bjtnpn:q2 2 31 6 0 model="qnl" bjtnpn:q3 10 5 7 0 model="qnl"
bjtnpn:q4 11 6 7 0 model="qnl" bjtnpn:q5 14 12 10 0 model="qnl"
bjtnpn:q6 15 12 11 0 model="qnl" bjtnpn:q7 12 12 13 0 model="qnl"
bjtnpn:q8 13 13 0 0 model="qnl" bjtnpn:q9 7 8 9 0 model="qnl"
bjtnpn:q10 2 15 16 0 model="qnl" bjtnpn:q11 2 14 17 0 model="qnl"
*****bjt model

```

```

statement***** .model qn1 bjtnpn(bf=80 rb=100
tf=.3ns tr=6ns rb=100 cje=3pf cjc=2pf vaf=50)
*****
vsource:vin 1 0 vdc=0. vac=.1 f=50e6 vsource:vcc 2 0 vdc=15
vsource:vee 3 0 vdc=-15
*****Transient
Analysis***** .tran2 tstop=250ns tstep=.5ns im=0
out_steps=50
*****
.out plot term 1 vt in "out.vt1" .out plot term 16 vt in
"out.vt16" .out plot term 17 vt in "out.vt17" .end

```

C.4 MESFET Universal Modeling

The MESFET netlist was used to generate results shown in Fig. 4.11.

C.4.1 Harmonic Balance Analysis

The harmonic balance netlist was used to generate results shown in Fig. 4.11.

```

**** Test netlist for mesfet models ****

.options f0 = 5.1e9 jupdm=4 output=0 nonlin=4

.svhb n_freqs = 15 fundamental = f0 steps=0 deriv=0

ind:l1 1 2 l=1e-9 time_d=0 cap:c1 2 3 c=20e-11 time_d=0 ind:l2 3 7
l=15e-9 time_d=0 res:r2 7 8 r=100

mesfetm:m1 3 4 123 idss = 0.06 vp0 = -1.906 gama = -0.015 e = 1.8
+ s1 = 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

res:rs 123 0 r=1.144 ind:l3 4 5 l=15e-9 time_d=0 res:r3 5 6 r=10
cap:cload 4 9 c=20e-12 time_d=0 res:rload 9 0 r=50.

vsource:vbias 8 0 vdc = -.4 vsource:vdrain 6 0 vdc = 3. res:rin
11 1 r = 50 vsource:vs 11 0 f = f0 vac = 1. .out plot term 11 vf

```

```

invfft 11 repeat in "out.vd11t" .out plot term 4 vf term 123 vf
sub invfft 20 repeat in "out.vdst" .out plot term 4 vf term 123 vf
sub mag in "out.vds.magt" .out plot element "mesfetm:m1" 1 if
invfft 4 repeat in "out.idst" .out plot element "mesfetm:m1" 1 if
term 4 vf mult invfft 10 repeat in "out.pdst" .out plot element
"vsource:vs" 0 if term 11 vf mult invfft 10 repeat in "in.pdst"

.end

```

C.4.2 Convolution Transient Analysis

The convolution netlist was used to generate the results seen in Fig. 4.11.

```

**** Test netlist for mesfet models ****

.svtr tstop = 1e-9 n_freqs = 8192 tstep = .002e-9 opt=0 adjust=0
imp_tol=30 + rcomp = 100 deriv=0

ind:l1 1 2 l=1e-9 time_d=0 *res:rfake3 1 2 r=1e6 cap:c1 2 3
c=20e-11 time_d=0
* res:fake1 2 3 r=1e6

ind:l2 3 7 l=15e-9 time_d=0 res:r2 7 8 r=100

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

res:rs 123 0 r=1.144

ind:l3 4 5 l=15e-9 time_d=0 res:r3 5 6 r=10 cap:cload 4 9 c=20e-12
time_d=0
* res:rfake2 4 9 r=1e6
res:rload 9 0 r=50.

vsource:vbias 8 0 vdc = -.4 vsource:vdrain 6 0 vdc = 3. res:rin
11 1 r = 50 vsource:vs 11 0 f = f0 vac = 1.

open:1 4 123

```

```
.out plot element "open:1" 0 ut in "out.v" .out plot element
"mesfetm:m1" 1 ut in "vds.tran" *.out plot term 4 vf term 123 vf
sub invfft 4 repeat in "out.vds" *.out write term 4 vt term 123 vt
sub in "out.dc"

.end
```

C.4.3 Transient Analysis

The transient analysis netlist was used to generate results shown in Fig. 4.11.

```
***** Test netlist for mesfet models *****

*.options f0 = \ 5e9 5.1e9 .1e9 \ .options f0 = 5.1e9 method = 2
jupdm = 1

.tran2 tstop = 4e-9 tstep = .005e-9 nst=0 msv=0 deriv=0 im=1

ind:l1 1 2 l=1e-9 *res:rfake3 1 2 r=1e6 cap:c1 2 3 c=20e-11
* res:fake1 2 3 r=1e6

ind:l2 3 7 l=15e-9 res:r2 7 8 r=100

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

res:rs 123 0 r=1.144

ind:l3 4 5 l=15e-9 res:r3 5 6 r=10 cap:cload 4 9 c=20e-12
* res:rfake2 4 9 r=1e6
res:rload 9 0 r=50.

vsource:vbias 8 0 vdc = -.4 vsource:vdrain 6 0 vdc = 3. res:rin
11 1 r = 50 vsource:vs 11 0 f = f0 vac = 1.

open:1 4 123
```

```
.out plot element "mesfetm:m1" 1 it in "id.tran2" .out plot
element "mesfetm:m1" 1 ut in "vds.tran2" .out plot term 4 vt term
123 vt sub in "vds.tran.node" *.out write term 4 vt term 123 vt
sub in "out.dc"

.end
```

C.5 Mesfet Multi-Tone Test

The two-tone netlist was used to generate results shown in Fig. 4.13, 4.14, 4.15.

```
***** Mesfet two tone analysis to calculate power at
fundamental .options f0=10.e9 f1=10.5e9 output=0 jupdm=4 .svhb
n_freqs=2 n_freqs2=1 fundamental=f0 fundamental2=f1 steps=1
deriv=0 *+regrowth=1 n_fund=3 f_step=10 vsource:vs1 22 0 f=f1
vac=.56 vsource:vs2 2 22 f=f0 vac=.56 res:rin 2 15 r=50. cap:cin
15 7 c=.01e-6 ind:l1 7 10 l=.69e-9 ind:l2 7 102 l=1.e-6 cap:c0 7 0
c=.13e-12 vsource:vd1 0 102 vdc=-1.V *vsource:vd1 102 0 vdc=-1.V
cap:c1 10 0 c=.31e-12 res:r1 10 3 r=2.9

mesfetm:m1 3 4 123 +a0=.08494 a1=.1116 a2=-.03371 a3=-.05953
gama=2.48 cgs0=.62e-12. cgd0=.34e-12. +t=0.0 +beta= .17 vds0=3. *+
vbd = 15 nr = 10 ib0 = 1e-9 *+cgs0 = .52785e-12 cgd0 = .087e-12
res:r2 123 8 r=2.4 ind:l3 8 0 l=.003e-9 res:r3 4 9 r=5.3 cap:c2 9
0 c=.9e-12 ind:l4 9 11 l=.41e-9 cap:c3 11 0 c=.03e-12 cap:cload 11
100 c=.01e-6 res:rload 100 0 r=50. ind:l5 11 101 l=1.e-6
vsource:vd2 101 0 vdc=3. *.out plot term 22 vf invfft 4 repeat in
"vin.source" *.out plot term 22 vf mag in "vin.mag" *.out plot
term 22 vf invfft 4 repeat fft in "vin.sourcet" *.out plot term 2
vf invfft 4 repeat fft in "vin.source" .out plot term 100 vf mag
in "mag.out" *.out plot term 100 vf in "out.vds" *.out plot term 4
vf in "out.vd1" .out plot term 100 vf invfft 4 repeat in
"out.vdst1" .out plot term 100 vf invfft 2 repeat fft in
"out.vdst2" .out plot term 100 vf invfft 4 repeat fft in
"out.vdst3" .out plot term 4 vf term 123 vf sub invfft 4 repeat in
"out.vds1" .end
```