



HAL
open science

Implementation of harmonic balance reduce model order equation

Abdallah Hijazi

► **To cite this version:**

Abdallah Hijazi. Implementation of harmonic balance reduce model order equation. Probability [math.PR]. Université de Limoges, 2015. English. NNT : 2015LIMO0139 . tel-01614852

HAL Id: tel-01614852

<https://theses.hal.science/tel-01614852>

Submitted on 11 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE DE LIMOGES

Science – Technologie – Santé

FACULTE DES SCIENCES ET TECHNIQUES

XLIM – Département C2S2

Thèse

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE LIMOGES

Discipline / Spécialité : Electronique des Hautes Fréquences et optoélectronique

présentée et soutenue par

Abdallah HIJAZI

le, 21 Décembre 2015

Implementation of Harmonic Balance Reduce Model Order Equation

Thèse dirigée par Edouard NGOYA

JURY :

M. Raymond QUERE	Professeur, Université de Limoges	Président
M. Juan-Mari COLLANTES	Professeur à l'Université de Bilbao – Espagne	Rapporteur
Mme. Almudena SUAREZ	Professeur à l'Université de Santander – Espagne	Rapporteur
M. Rémi LARCHEVEQUE	Ingénieur, Mentor Graphics, Montbonnot St-Martin (38)	Examineur
M. Sébastien MONS	Chargé de Recherches CNRS, Université de Limoges	Examineur
M. Edouard NGOYA	Directeur de Recherches CNRS, Université de Limoges	Examineur
M. Raphaël SOMMET	Chargé de Recherches CNRS, Université de Limoges	Examineur
M. Arnaud SOURY	Ingénieur, Keysight Technologies, Les Ulis (91)	Examineur

*To the memory of my grandparents
To my parents
To my beloved ones*

Acknowledgment

This work is realized at laboratory of (XLIM UMR CNRS n°6172), C2S2 departement (Composants, Circuits, Signaux et Systèmes Hautes Fréquences), University of Limoges, and I would like to thank M. Bernard Jarry, head of department, for receiving me in his team.

I would like to express my sincere gratitude to my advisor Prof. NGOYA Edouard for the continuous support of my Ph.D. study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I would like to thank the rest of my thesis committee: M. Juan-Mari COLLANTES, Mme Almudena SUAREZ, M. Rémi LARCHEVEQUE, M. Sébastien MONS, M. Raymond QUERE, M. Paphaël SOMMET, and M. Arnaud SOURY, for their encouragement, insightful comments.

I would like to express my deepest gratitude to both Dr. CAMPOVECCHIO Michel and Dr. MONS Sebastian once again for their fruitful help and support, without their help, it wouldn't be possible to reach this level.

I would like to thank my beloved family: to my father Dr. HIJAZI Ali who always had faith in me, to my mother RAMMAL Mariam who always enlightens my way with her love and kindness. My sisters Dr. HIJAZI Faten and Dr. HIJAZI Rola, and my brother Dr HIJAZI Firas, thank you all for your support and love you gave me, without your help and support, it wouldn't be possible to finish my thesis.

I deeply thank my friends Mr. FARHAT Abbas, Dr HIJAZI Alaa, Dr. HIJAZI Ragheb, Mr. SAFIELDINE Ali, Dr. EZZELDINE Hussein, and Dr. EL NASHEF Georges for their endless support, and motivation to finish my thesis, also, I would like to thank all my fellows lab-mates in XLIM Group for the lovely environment and the courage they offered, list is big, so I will not name them one by one so I don't forget any.

A big special thanks to my lovely nieces, Sirine, Lea, Mariam and Sarah, and to my lovely nephews Ali and Nader, you cherished my life.

A big thanks for my uncle Mr RAMMAL Abdallah, my uncle Dr HIJAZI Fahed and my cousin Dr RAMMAL Hassan for their fruitful support.

Last but not least, a very special thanks to WENTA Katarzyna and her family for their endless support.

Table of Contents

Chapitre I : Introduction.....	1
I. Overview	3
II. Organization of the thesis:	7
Chapitre II : Background and Overview	9
I. Generalized Formulation of Circuit Equations and numerical methods for finding Steady-State solution	11
I.1. Modified Nodal Analysis	11
I.2. Newton Raphson	16
I.3. Continuation methods.....	18
II. Background on steady-state analysis techniques	19
II.1. Time domain approaches	19
II.2. Frequency Domain approaches	23
III. Reference	35
Chapitre III : Model Order Reduction of circuit equations	39
I. Introduction	41
II. Model Order Reduction.....	43
II.1. MOR for Linear Systems	43
II.2. MOR for Nonlinear Systems.....	49
III. Discussion	54
Chapitre IV : Harmonic Balance Reduced Order Model Equation	59
I. Introduction	61
I.1. General formulation of Harmonic Balance reduced order model equation.....	64
I.2. General RF circuit equation.....	64

I.3. . Harmonic Balance equation.....	65
I.4. Reduced order equation.....	65
I.5. Padé approximation approach to HB equation model order reduction	66
I.6. dentification of moments of Padé (MP) approximation.....	68
I.7. . Iterative solution of the MP identification equation.....	70
I.8. . Polynomial factorization of the nonlinear device equation.....	74
I.9. Moments of Padé Identification algorithm.....	75
I.10. Formal factorization of the nonlinear device equation.....	76
II. Solving for Steady state solution using MOR.....	80
II.2. MOR HB implementation	82
II.3. Numerical examples:.....	84
II.4. Disscussion.....	94
II.5. Limitations of the method	96
II.6. Proposed Numerical Differentiator	99
III. Numerical appraoch.....	100
III.1. Variable α	100
III.2. EXPLENATION OF ALGORITHM.....	101
IV. Numerical results	117
IV.1. Example of half wave rectifier	117
IV.2. Example of a power amplifier	119
V. Discussion	124
VI. References:	126
Chapitre V : Conclusion and Future Scope.....	129
I. Summary.....	131
II. FutureWork	133

List of Figures

Figure II- 1: stamps of circuit components of MNA formulation	12
Figure II- 2 : shows a circuit of linear components	13
Figure II- 3 : Rectifier Circuit.....	14
Figure II- 4 Graphical Representation of Newton Formula.....	17
Figure II- 5 Flow chart of Harmonic balance method	25
Figure II- 6 : Nonlinear circuit modified to be formulated using MNA	26
Figure II- 7: simple circuit with single nonlinearity	31
Figure IV - 1 : Illustration of HB equation parameterization	69
Figure IV - 2 : Illustration of the Jacobian matrix	72
Figure IV - 3 : Illustration of a simple diode circuit.....	77
Figure IV - 4 : Rectifier circuit	85
Figure IV - 5 : HB vs Proposed output	86
Figure IV - 6 : Tuned amplifier circuit	87
Figure IV - 7 : Ebers-Moll model	88
Figure IV - 8 : HB vs Proposed output	89
Figure IV - 9 : Solution at the first harmonic vs input level of Tuned amplifier.....	90
Figure IV - 10 : Solution at the third harmonic vs input level of Tuned amplifier.....	90
Figure IV - 11 : Solution at the fifth harmonic vs input level of Tuned amplifier	91
Figure IV - 12 : Power amplifier circuit	92
Figure IV - 13 : HB vs Proposed output	93
Figure IV - 14 : Nonlinear element controlled by multiple variables.....	96
Figure IV - 15 :Percentage error difference of Analytical and Numerical derivatives of reduced system.....	112
Figure IV - 16 : Percentage of error difference of sixth derivative of the system	113
Figure IV - 17 : Percentage of error difference of sixth derivative of the system	114
Figure IV - 18 :Time required to analytically compute the Moments of the Power amplifier circuit	117
Figure IV - 19 :computational complexity of moment evaluation	118
Figure IV - 20 : Power Amplifier circuit	119

Figure IV - 21 :Time required to analytically compute the Moments of the Half-wave rectifier	120
Figure IV - 22 :Computational complexity of evaluation of the Moments for a half-wave rectifier .	121
Figure IV - 23 : Solution of the first harmonic vs input level of power amplifier	122
Figure IV - 24 : Solution of the third harmonic vs input level of power amplifier	123

List of Tables

Table IV - 1: Formulas for the derivatives of some functions.....	79
Table IV - 2 : shows a comparison of the conventional HB versus the proposed algorithm.....	89
Table IV - 3 : Shows a comparison of the conventional HB versus the proposed algorithm for example 3	93
Table IV - 4 Effect of variation of factor α	115
Table IV - 5 : Computational time required to solve HB using analytical and numerical proposed method.....	122

Abbreviations

ADS	Advanced Design System
CAD	Computer Added Design
CPU	Central Processing Unit
DAE	differential-algebraic equations
DC	Direct Current
DFT	Discret Fourier Transform
FET	Field-Effect Transistor
FFT	Fast Fourier transform
GMRES	Generalized Minimal Residual
HB	Harmonic balance
HB	Harmonic Balance
ICs	integrated circuits
IFFT	Inverse Fast Fourier transform
KCL	Kirchhoff's current law
LMS	linear multi-step
LU	LU factorization
MATLAB	Matrix Laboratory
MIMO	Multi-Input/Multi-Output System
MNA	Modified Nodal Analysis
MOR	Model order reduction
MP	Moments of Padé
NR	Newton Raphson

POD	Proper Orthogonal Decomposition
PRIMA	Passive Reduced-Order Interconnect Macro-modeling
PVL	Padé via Lanczos
RF	Radio Frequency
SPICE	Simulation Program with Integrated Circuit Emphasis
SVD	Singular Value Decomposition
TS	Time Shooting
VLSI	Very Large Scale Integration
TPWL	Trajectory PieceWise Linear

Chapitre I :Introduction

I. Overview

The increasing demand for very high performance mobile communication systems has pushed CAD tools to find more powerful simulation algorithms, that they can be used to efficiently and accurately analyze frequency response, distortion, and noise of Radio Frequency (RF) communication circuits, such as mixers, amplifiers and filters. Such algorithms are considered to reduce CPU cost, decrease memory usage, and reduce time to market for designers.

The principal methods currently used for the design of RF circuits are the Harmonic balance (HB) and the time shooting (TS) techniques. HB is preferred to TS for circuits containing multitoned signals. TS in turn is preferred for periodic excitation circuit with very strong nonlinearity.

While these techniques are used successfully in many designs, their computation cost and memory occupation grows rapidly with increasing circuit size, making large design simulation a difficult challenge.

It is therefore necessary to work out techniques that can reduce the computation cost

Model reduction is a technique where a large system of equations is reduced to a much smaller set of equations, and the obtained response from the reduced system is equivalent to the original one in terms of dimensionality. Model reduction as a research field is divided into two sub-domains, linear and nonlinear, in which linear model reduction is a mature field of research, where we have numerous established methods to reduce linear systems. However, nonlinear model reduction is still facing many challenges.

Consider a linear system, techniques such as moment matching where Padé approximation have been widely studied and used to reduce a linear system. Padé approximation however, turned out to exhibit an ill-conditioned problem. In order to overcome this problem, several techniques based on Krylov-subspace and congruent transformations have been developed, such as Padé via Lanczos (PVL) that was Developed by Feldmann and Freund in 1994 [1], alternatively Silveira, Kamon, White, Elfadel, et al [2] have developed an “Arnoldi” algorithm that’s based on krylov subspace, in order to ensure stability. PVL and Arnoldi methods have managed to overcome the Pade matix ill-conditioning problem, but passivity remained a problem until Odabasioglu, Celik and Pileggi [3] have developed the PRIMA (Passive

Reduced-Order Interconnect Macromodeling) method. PRIMA method combines moment matching with congruence transformation and preserves system passivity.

The various work carried out in this field have made linear Model Order reduction a mature domain with well-established concepts. The situation is however more stammering for nonlinear (MOR), where the most work published rely on such techniques like first order and quadratic approximation or piece-wise linear expansion [4]

The quadratic method expands the function to the first and second order Taylor approximation, then it applies one of the linear model reduction methods such as Arnoldi algorithm in order to reduce the system. Obviously its main disadvantage is that it depends on how the original function is similar to quadratic function [4].

The basic idea of Piece-wise-linear method [5] is to represent the non-linear system with a piecewise-linear system and then reduce each of the pieces with linear model reduction methods. This method has a better performance than linear and Quadratic method, however, its disadvantage is that the resultant piece-wise-linear system's accuracy depends on the training input; therefore it is not qualified as a general purpose approach for circuit simulation. Such methods can be applied when the circuit is weakly nonlinear, however, when the circuit is mildly or heavily nonlinear they cannot work; therefore finding more effective nonlinear MOR methods is necessary to tackle large RF circuit simulation problems.

From the literature, the most promising technique found so far is the nonlinear Padé approximation based method proposed by Gad et al in [6]. The method projects the original nonlinear large problem onto a smaller subspace, by constructing moments of Padé approximation from DC operating point, then the problem is simulated entirely on the reduced space, afterwards, the obtained solution is used to regenerate the original space of the problem. This method has proved a good performance in terms of system size reduction, as well as stability, passivity and accuracy.

The main challenge of the method resides in the necessity for an explicit factorization of the nonlinear devices equations in the form of a power series, which is seldom possible in a general purpose simulation

tool because of the high complexity of device models. Besides for multiple variables device equation, the number of power coefficients grows exponentially, making the method impractical.

This Thesis will review the above method and study solutions to overcome the mentioned limitations, through numerical differentiation approach.

II. Organization of the thesis:

Chapter II presents a brief background of steady state solution methods for both the time domain and frequency domain. Then it gives a short presentation of the Modified Nodal Analysis (MNA) method including linear and nonlinear elements. This formulation serves as the basis for mathematical development of the rest of the thesis.

Chapter III discusses the background of Model Order Reduction. It starts with a definition of model order reduction for linear systems, and then it follows with a classification of the main strategies available in the field. It then introduces the more challenging problem of nonlinear dynamic systems. The second part of the chapter presents a review of the ideas currently used for the reduction of large nonlinear system representations and the difficulties and limitations found in each one of these techniques.

Chapter IV consists of two parts, the first part reviews in details the model order reduction technique via krylov subspace proposed by Gad et al [6], It shows the application of the method to harmonic balance equation. Finally, this part spots the light on the limitation of the proposed method. The second part of the chapter introduces a numerical algorithm aiming to overcome the limitations pointed out, and shows numerical examples to evaluate the effectiveness

Finally, the last chapter presents a discussion about the work presented in this thesis and a future scope.

Chapitre II : Background and Overview

This chapter presents a general framework for the mathematical representation of electrical circuits.

The formulation used in this chapter is general and will be used throughout the thesis in explaining the main objectives of circuit simulation. Two main sections will take place in this chapter, Section I will discuss general numerical background of circuit solution techniques such as Modified Nodal analysis, Newton Raphson method and continuation methods, in which they are prerequisite for the second section which overviews the main techniques of steady state solution analysis.

I. Generalized Formulation of Circuit Equations and numerical methods for finding Steady-State solution

I.1. Modified Nodal Analysis

I.1.1. MNA Formulation of Linear circuits

Nodal equations are formulated applying Kirchhoff's current law (KCL) which states that the sum of all currents flowing into node i is equal to zero $\sum_j I_{i,j} = 0$

Consider a circuit consisting of only linear elements, in particular lumped RLC components and independent voltage source. Using the MNA formulation [7] and [8], the circuit equations can be presented in a matrix form as:

$$\mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{G}\mathbf{x}(t) = \mathbf{b}(t) \quad II-1$$

Where

$\mathbf{x}(t) \in \mathfrak{R}^N$ is a vector of node voltage waveforms appended by independent voltage source and linear inductors' currents.

$\mathbf{C} \in \mathfrak{R}^{N \times N}$ and $\mathbf{G} \in \mathfrak{R}^{N \times N}$ are matrices that contain the linear lumped memory and memoryless elements, respectively. For simplicity, they will be presented in one matrix called \mathbf{Y} , when possible, where:

$$Y \in \mathfrak{R}^{N \times N}, \text{ and } Y = C \frac{d}{dt} + G$$

$b(t)$ is a vector with entries determined by the independent voltage and current sources

Finally, N is the total number of variables, including node voltages and currents of voltages sources and inductor.

The Y matrix is typically referred to as the MNA matrix. The advantage of using the MNA formulation is that the construction of the different terms in Equation (2.1) can be easily automated through using predefined formulations for each component, called stamps. Figure II- 1 shows the stamps corresponding to the RLC lumped components in addition to independent voltage source.

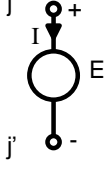
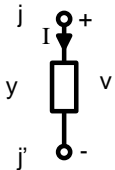
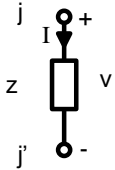
Element	Symbol	Matrix	Equations
VOLTAGE SOURCE		$ \begin{array}{c} V_j \quad V_{j'} \quad I \\ \text{Source} \\ \text{Vector} \\ j \quad \left[\begin{array}{cc c} & & 1 \\ & & -1 \\ \hline 1 & -1 & \end{array} \right] \begin{array}{c} \\ \\ E \end{array} \\ j' \\ m+1 \end{array} $	$ \begin{array}{l} V_j - V_{j'} = E \\ I_j = I \\ I_{j'} = -I \end{array} $
ADMITTANCE		$ \begin{array}{c} V_j \quad V_{j'} \\ j \quad \left[\begin{array}{cc} y & -y \\ -y & y \end{array} \right] \\ j' \end{array} $	$ \begin{array}{l} I_j = y(V_j - V_{j'}) \\ I_{j'} = -y(V_j - V_{j'}) \end{array} $
IMPEDANCE		$ \begin{array}{c} V_j \quad V_{j'} \quad I \\ j \quad \left[\begin{array}{cc c} & & 1 \\ & & -1 \\ \hline 1 & -1 & -z \end{array} \right] \\ j' \\ m+1 \end{array} $	$ \begin{array}{l} V_j - V_{j'} - zI = 0 \\ I_j = I_{j'} = I \end{array} $

Figure II- 1: stamps of circuit components of MNA formulation

As an example, for the circuit of Figure II- 2 we obtain the modified nodal formulation.

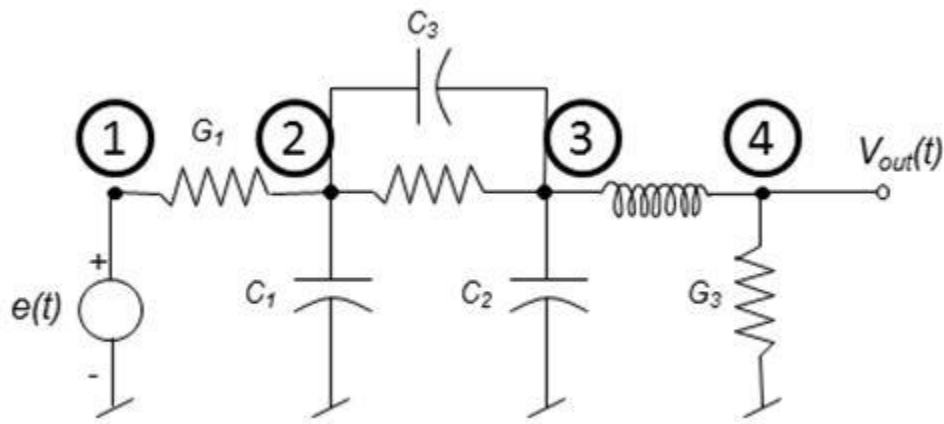


Figure II- 2 : shows a circuit of linear components

$$Y = C + G = \begin{bmatrix} G_1 & -G_1 & 0 & 0 & 0 & 0 & 1 \\ -G_1 & G_1 + G_2 + j\omega C_1 + j\omega C_3 & -G_2 - j\omega C_3 & 0 & 0 & 0 & 0 \\ 0 & -G_2 - j\omega C_3 & j\omega C_2 + j\omega C_3 + G_2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & G_3 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & j\omega L_1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad II-2$$

$$b(t) = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad E]^T \quad II-3$$

From equations (2.2) and (2.3) a general structure for G , C and $x(t)$ can be written as [3] :

$$C = \begin{bmatrix} Q & 0 \\ 0 & L \end{bmatrix}, G = \begin{bmatrix} R & E \\ -E^T & 0 \end{bmatrix}, x(t) = \begin{bmatrix} v(t) \\ i(t) \end{bmatrix} \quad II-4$$

In which Y could be presented as follows:

$$Y = \begin{bmatrix} Q + R & E \\ -E^T & L \end{bmatrix} \quad II-5$$

Where R , Q and L are matrices that contain the stamps for the resistors, capacitors, and inductors, respectively. The elements of E consists of positive and negative ones, and zeros only, such that E maps the current variables to the space of nodes in the network.

The matrices \mathbf{R} , \mathbf{Q} and \mathbf{L} are symmetric nonnegative definite provided that the original network is composed of passive linear elements, i.e. positive resistors, capacitors, and inductors only [3].

I.1.2. Nonlinear Elements

The inclusion of nonlinear elements in the circuit requires an additional term to be added to the equations in (2.1). A nonlinear element can be resistive, capacitive or inductive. A circuit with nonlinear elements can be described by [7]

$$\mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{G}\mathbf{x}(t) + \mathbf{i}_{NL}(\mathbf{x}(t)) = \mathbf{b}(t) \quad II-6$$

Where $\mathbf{i}_{NL}(\mathbf{x}(t))$ is a vector of nonlinear currents and voltages representing the contribution of nonlinear elements.

Circuit shown in Figure II- 3 will be used to illustrate the nonlinear element contributions. In this example, we have a single nonlinear element defined by the following nonlinear equation:

$$I = I_s \left(e^{\frac{(v_1-v_2)}{V_T}} - 1 \right) \quad II-7$$

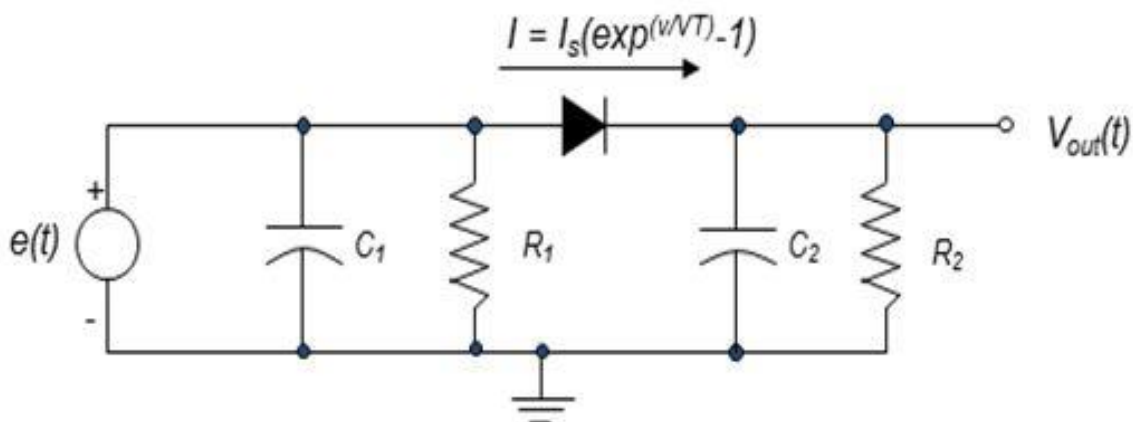


Figure II- 3 : Rectifier Circuit

We therefore have the following matrices and vectors composing the MNA equation system:

$$Y = \begin{bmatrix} j\omega C_1 + \frac{1}{R_1} & 0 & 1 \\ 0 & j\omega C_2 + \frac{1}{R_2} & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad II-8$$

$$i_{NL}(x(t)) = \begin{bmatrix} I_S \left(e^{\frac{(v_1-v_2)}{v_t}} - 1 \right) \\ -I_S \left(e^{\frac{(v_1-v_2)}{v_t}} - 1 \right) \\ 0 \\ 0 \end{bmatrix} \quad II-9$$

$$x(t) = \begin{bmatrix} v_1(t) \\ v_2(t) \\ I_{V_1} \end{bmatrix} \quad II-10$$

$$b(t) = \begin{bmatrix} 0 \\ 0 \\ E \end{bmatrix} \quad II-11$$

In general, after formulating a problem, we are end up with a set of nonlinear equations that need to be solved numerically by means of iterative methods to get the steady state solution. Newton's method and its variants are commonly considered for this problem.

I.2. Newton Raphson

I.2.1. Definition of Newton Raphson:

The Newton's method [7] (or also Newton-Raphson method) is the main technique used in electrical circuit simulator for its relative programming simplicity and quadratic convergence property

Newton's method solves iteratively the algebraic system of equations:

$$f(x) = 0, x \in \mathfrak{R}^N$$

By producing the recurrence:

$$x_{n+1} = x_n - \left[\frac{df(x_n)}{dx} \right]^{-1} f(x_n); \quad n = 0, 1, \dots$$

The matrix $J = \left[\frac{df(x_n)}{dx} \right]$ is called the Jacobian of the equation system. A graphical illustration of Newton's method is given in Figure II- 4 for a single variable equation.

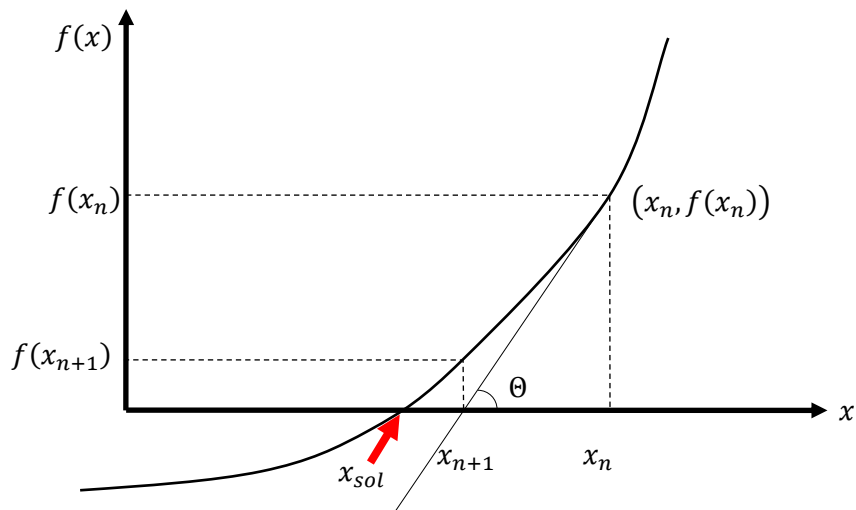


Figure II- 4 Graphical Representation of Newton Formula

Newton's method may not be convergent for an arbitrary initial guess x_0 , however, its convergence is quadratic when x_0 is close enough to the solution x_{sol} .

For this reason, Newton's method is usually coupled with homotopy and continuation methods to ensure an initial guess close to solution [9]

I.3. Continuation methods

I.3.1. Definition

The remedy found to solve the problem of convergence for NR is the continuation method.

Continuation methods consist of introducing a new factor into the system, then increasing this factor from 0 to 1 in some increment rate in order to find the steady state solution.

The main idea of the continuation method [9] is to augment the original system using the introduced factor, in order to obtain auxiliary system. Once the continuant factor is equal to zero, the auxiliary system has a trivial solution to obtain, and when the continuant factor reaches one, the auxiliary system is identical to the original one. The method of introducing the continuation factor will be introduced later in this

II. Background on steady-state analysis techniques

In the recent decades, the steady state analysis became an essential need for commercial circuit simulators, for the accuracy it provides in meeting complicated RF circuit designs' specifications and limitations of circuits such as power amplifiers, mixers, and filters [10]. The key to steady state analysis techniques is that the response of a stable circuit to a periodic input must be periodic in the long term. It is then possible to formulate the problem as a boundary-value problem [11]. Different approaches could be considered in solving such systems. Most known approaches proposed in the literature are the time domain and others in the frequency domain methods, where the latter methods are there whenever time domain methods become too expensive in computers' resource. Some other approaches are based on combining both time and frequency domain together, aiming to come up with a solution that benefits from the advantages of both approaches at once [12], [13].

This subsection presents a brief overview for the various methods that appeared in the literature for finding the steady-state solution of nonlinear circuits. Subsection 2.2.1 shows the basic idea of steady-state methods based on the time-domain approach. Subsection 2.2.2 illustrates the harmonic balance method as an example of frequency-domain approaches.

II.1. Time domain approaches

Time-domain approaches tackle the problem of computing the steady-state operating point as a boundary-value problem in the time-domain[10]. The aim of a time-domain based technique is to find a set of initial conditions for the circuit variables such that numerical solution of the MNA differential-algebraic equations (DAE) described in (2.6) results in the same initial starting point after integration for one oscillation period of the forcing input.

II.1.1. Conventional transient analysis

As a definition, transient analysis is a brute-force method to obtain the steady-state solution through approaching the problem as a classical initial-value problem. In initial-value problems, an initial condition for the circuit variables at $t = t_0$ is assumed to be known. The response is obtained by starting with the initial condition and then moving forward in time using numerical integration techniques until all transients have died out [7] [14]. Fast Fourier transform (FFT) [15] is then applied to the signal obtained when the steady-state has been reached in order to obtain the frequency components of the circuit. Typical integration methods are based on linear multi-step (LMS) methods such as the Trapezoidal and Backward Euler rules [16].

As an example, consider the Backward Euler technique, where the following formula is used to approximate the derivative of the circuit variables, $\mathbf{x}(t)$, at $t_n = t_{n-1} + h$,

$$\left. \frac{dx(t)}{dt} \right|_{t=t_n} = \frac{x(t_n) - x(t_{n-1})}{h} \quad II-12$$

Applying (2.17) to the MNA equations of (2.6) yields

$$\left(\mathbf{G} + \frac{1}{h} \mathbf{C} \right) \mathbf{x}(t_n) + \mathbf{i}_{NL}(\mathbf{x}(t_n)) = \frac{1}{h} \mathbf{C} \mathbf{x}(t_{n-1}) + \mathbf{b}(t_n) \quad II-13$$

Equation (2.18) can be written in the form

$$\mathbf{A} \mathbf{x}(t_n) + \mathbf{i}_{NL}(\mathbf{x}(t_n)) = \mathbf{u} \quad II-14$$

Where

$$\mathbf{A} = \mathbf{G} + \frac{1}{h} \mathbf{C}; \quad \mathbf{u} = \frac{1}{h} \mathbf{C} \mathbf{x}(t_{n-1}) + \mathbf{b}(t_n) \quad II-15$$

Thus, applying backward Euler rule to 2.6 converts the set of DAE into a set of nonlinear algebraic equations with $\mathbf{x}(t_n)$ as the set of unknowns. Solving the above system of nonlinear equations is typically executed through the Newton method [17] as indicated previously. A typical NR method works iteratively to find the solution of (3.19) as follows:

$$\mathbf{x}^{(i+1)}(t_n) = \mathbf{x}^{(i)}(t_n) - \left[\mathbf{J} \left(\mathbf{x}^{(i)}(t_n) \right) \right]^{-1} \boldsymbol{\Psi} \left(\mathbf{x}^{(i)}(t_n) \right) \quad II-16$$

Where $\mathbf{J}(\mathbf{x})$ is the Jacobian matrix of (2.21) and is given by

$$\mathbf{J}(\mathbf{x}) = \mathbf{A} + \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \quad II-17$$

and

$$\boldsymbol{\Psi}(\mathbf{x}) = \mathbf{A} \mathbf{x} + \mathbf{f}(\mathbf{x}) - \mathbf{u} \quad II-18$$

As shown from the above analysis, finding the solution at each time step requires several LU decompositions of the Jacobian matrix in order to solve the linear system of equations in (2.21) in which, a significant CPU cost will be taken into consideration, and for large circuits, the size of these matrices becomes typically in the order of tens of thousands, which is a cumbersome for the computer resources and time consuming for circuit simulation for medium and large size circuits [6]. The CPU cost becomes even more prohibitive in RF circuits since quality factor is higher (high Q), requiring the circuit to go through many cycles before reaching the steady-state.

II.1.2. Shooting Method

The goal of shooting methods is to accelerate convergence to steady state of the above conventional transient analysis, i.e., shooting methods looks for skipping long transient cycles [18], [19], [20]. The basic ideas in the shooting method stems from the fact that a periodic steady state response, $x(t)$, with period T satisfies condition , $x(t) = x(t+T)$ for any time $t > T_{\text{transient}}$. If an initial condition vector $x(0)$ can be found in a way that

$$x(T) - x(0) = 0 \quad \text{II-19}$$

Then the periodic solution can be reached in only one cycle of numerical integration. Actually, the condition stated by eq II-24 is a typical boundary-value problem and express through the state-transition formalism [14]

$$\Phi(x(t), \tau) = x(t + \tau) \quad \text{II-20}$$

$\Phi(x(t), \tau)$, the state transition function expresses the transformation from the initial state condition $x(t)$, at time t , to the final condition $x(t + \tau)$, at time $t + \tau$, due to the dynamics of the circuit. Thus we have the following equation where the unknowns are the initial state condition $x_0 = x(0)$:

$$\Phi(x_0, T) - x_0 = 0 \quad \text{II-21}$$

Eq II-27 is typically solved using Newton method and is referred to as Newton-Shooting technique [17] [18] [21].

$$x_0^{i+1} = x_0^i - [J(x_0^i)]^{-1} \Psi(x_0^i)$$

Where function $\Psi(x_0^i)$ is the nonlinear function which can be presented as follows:

$$\Psi(x_0^i) \equiv \Phi(x_0^i, T) - x_0^i = 0$$

Newton method necessitates the computation and LU factorization of the Jacobian matrix $\mathbf{J}(\mathbf{x}_0^i)$

$$\mathbf{J}_s(\mathbf{x}_0^i) = \left. \frac{\partial \Psi(\mathbf{x}_0)}{\partial \mathbf{x}_0} \right|_{\mathbf{x}_0=\mathbf{x}_0^i} = \left. \frac{\partial \Phi(\mathbf{x}_0, T)}{\partial \mathbf{x}_0} \right|_{\mathbf{x}_0=\mathbf{x}_0^i} - \mathbf{U} \quad \text{II- 22}$$

Where \mathbf{U} is an identity matrix in eq II-22

The main computational burden of the Newton-shooting method is caused by computing and factorizing the Jacobian matrix, which is a dense matrix, so that the cost tend to be $\mathcal{O}(N^3)$ [14] where N is the number of circuit nodes.

Nowadays however circuit simulators typically use iterative Krylov-subspace methods to form and factor the Jacobian matrix, hence drastically reducing the Newton iteration cost [22]. With use of the Krylov-subspace methods, Shooting-Newton is very powerful technique allowing the simulation of large scale circuits. The major limitation is that the technique is limited to periodic regime analysis, hence cannot efficiently handle RF circuits driven by multiple tone signals.

II.2. Frequency Domain approaches

II.2.1. General Theory of Harmonic Balance Approach

The Harmonic Balance (HB) approach is a frequency domain technique for finding the steady-state solution. In this section, HB theory will be introduced.

HB has been investigated widely for the last three decades. Rizzoli [23]-[24] has published several papers on harmonic balance, algorithms, and implementation techniques. Several authors, like Nakhla and Vlach [25], Maas [26], Kundert [10], Rizzoli et. Al [27][28][36][37], and A. Demir, J. Ryochowdhury et. Al [29][30][38], have explained HB approach in more details, and it's left to the reader to refer to their works in case it is needed.

The concept of HB will be reviewed in this section as follows. In the first section, the general formulation of the harmonic balance problem is reviewed. Section two introduces an example of harmonic balance. Section three will introduce the Newton iteration method to solve the HB equations, and it will investigate the evaluation of Jacobian matrix, section four will explained the sampling and the Fast Fourier Transom (FFT), where the fifth section will give an idea about how to solve the harmonic balance equations using the GMRES.

II.2.2. Harmonic Balance Equations for Nonlinear Circuits

We consider a case of a nonlinear circuit; the nodal equations in the time domain can be written as a differential equation

$$\frac{dq(v(t))}{dt} + i(v(t)) + \int_{-\infty}^t y(t - \tau)v(\tau)d\tau - b(t) = \mathbf{0} \quad II-23$$

Where $q(v)$, $i(v)$, y and $b(t)$ represent nonlinear charge, nonlinear current, admittance (impulse response), and excitation terms, respectively. The unknowns $v(t)$, in equation (2.48) are the node voltages of the circuit at some time point, t .

Going back to the general definition, the harmonic balance method looks for the steady state solution of the problem of equation above in the frequency domain. Assuming that the excitation $b(t)$ is a periodic or quasi-periodic signal, the waveform of each node voltage can be represented by a generalized Fourier series of the form

$$v(t) = \sum_{k=-H}^H V_K e^{jk\omega t} \quad II-24$$

Same as for $i(v(t))$, $q(v(t))$, and $i_s(t)$

$$i(t) = \sum_{k=-H}^H I_K e^{jk\omega t} \quad II-25$$

$$q(t) = \sum_{k=-H}^H Q_K e^{jk\omega t} \quad II-26$$

$$b(t) = \sum_{k=-H}^H (B)_K e^{jk\omega t} \quad II-27$$

Where V_k , I_k , Q_k , and $(I_s)_k$ represent the harmonics for the nodes voltages, currents of the nonlinear conductances, charge of the nonlinear capacitances, and sources currents, respectively.

Thus, the equation (2.37) can transformed from the time-domain into the frequency domain:

$$F(V) = YV + I(V) + j\Omega Q(V) - B = \mathbf{0} \quad II-28$$

II.2.3. Numerical Solution

The process of finding steady-state solution using harmonic balance is describe in the flow diagram in Figure II- 5

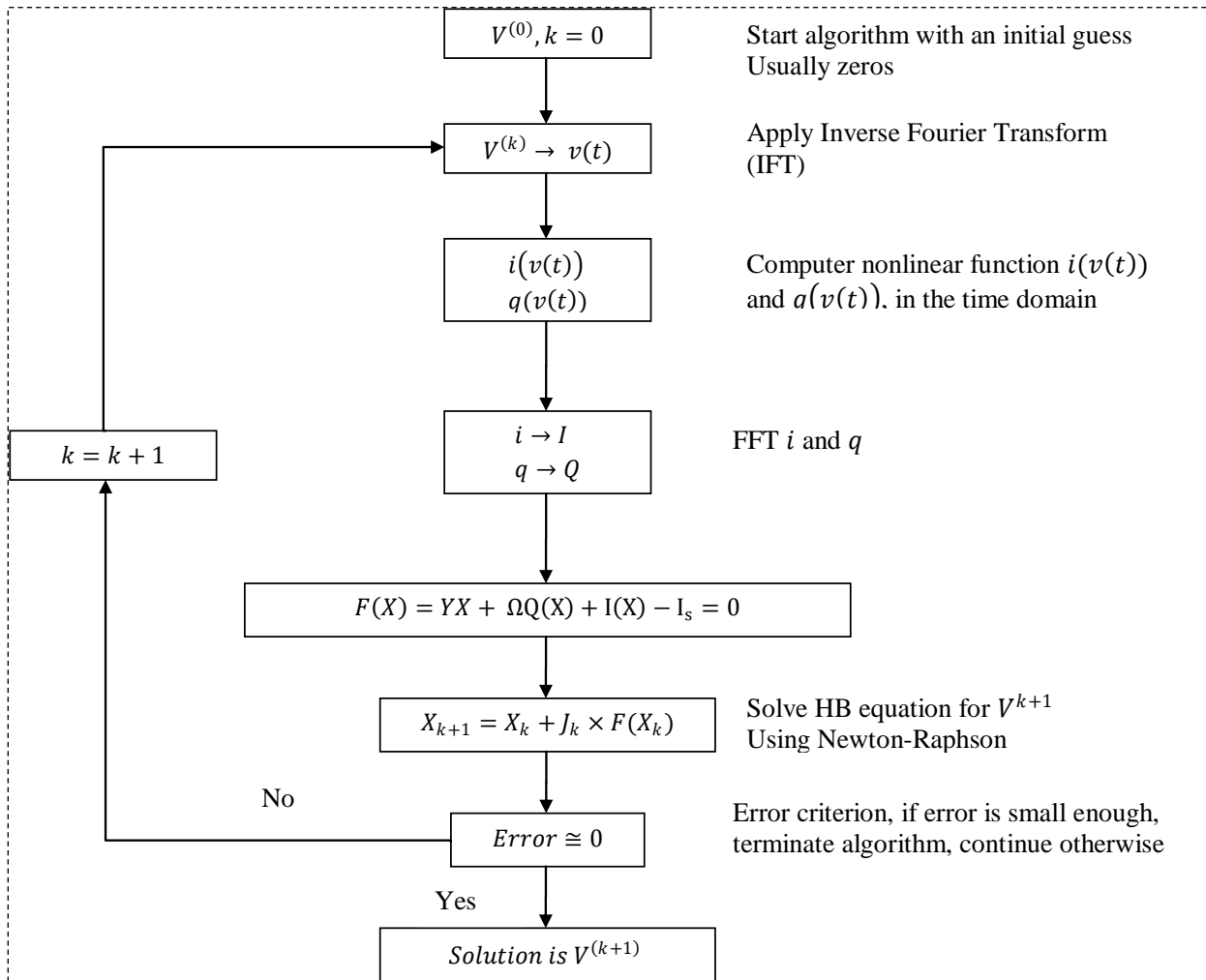


Figure II- 5 Flow chart of Harmonic balance method

II.2.4. Harmonic balance equation's formulation

In this section, we will illustrate the formulation of Harmonic Balance equation using MNA technique.

As stated before, the harmonic balance equation in the frequency domain is:

$$YV + I(V) + j\Omega Q(V) - B = 0 \quad \text{II- 29}$$

Where Y is the linear components matrix of size $N(2H + 1) \times N(2H + 1)$.

X is a vector contains the harmonics for the voltage waveforms at the N nodes, and it is of $N(2H + 1) \times 1$.

$I(X)$ is a vector contains the currents of the nonlinear conductances in the circuit, it's of size $N(2H + 1) \times 1$.

Q is a vector that presents the charge of the nonlinear capacitances in the circuit, it has $N(2H + 1) \times 1$ as size.

B represents the sources currents in the circuit and it is of size $N(2H + 1) \times 1$ as well.

Last but not least, Ω is a diagonal matrix of size $N(2H + 1) \times N(2H + 1)$ and it contains N cycles of $(-H\omega_0, \dots, 0, \dots, H\omega_0)$ along its main diagonal.

As an example, the HB equations are written for the circuit shown in Figure II- 6, however, as a first step, the circuit must be adapted to the MNA technique formulation that was discussed earlier in this chapter.

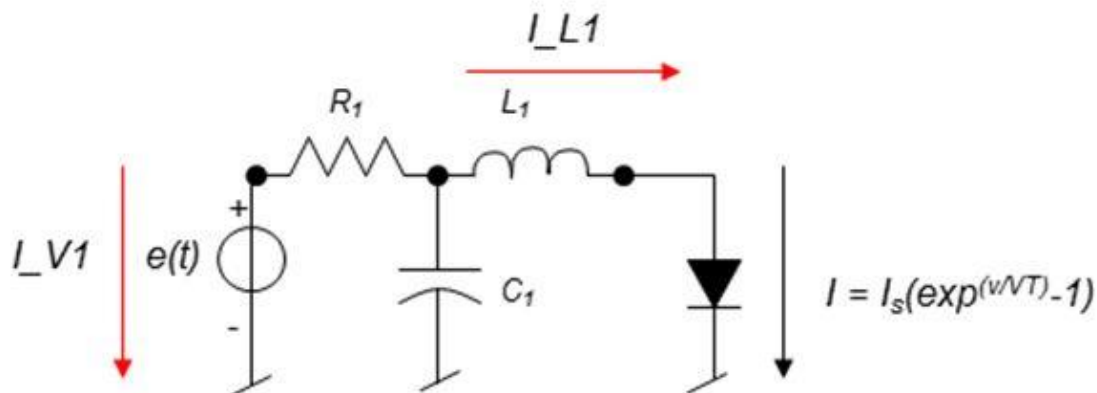


Figure II- 6 : Nonlinear circuit modified to be formulated using MNA

As we can see in figure above that two new unknowns were added up, I_{L1} and I_{V1} , but for simplicity, we will call them simply V_4 and V_5 , they are actually the currents through inductor L and voltage source $e(t)$, respectively.

The admittance matrix of linear part of the circuit at angular frequency ω is:

$$Y_{lin}(\omega) = \begin{bmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & 0 & 1 \\ -\frac{1}{R_1} & \frac{1}{R_1} + j\omega C_1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & -1 & j\omega L_1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad II-30$$

The source vector would be presented in such way:

$$B(\omega) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -E(\omega) \end{bmatrix} \quad II-31$$

The Y matrix in equation 2.54 would be presented as a whole as follow

$$Y = \begin{bmatrix} Y_{lin}(-H_2) & & & & \\ & Y_{lin}(-H_1) & & & \\ & & Y_{lin}(0) & & \\ & & & Y_{lin}(H_1) & \\ & & & & Y_{lin}(H_2) \end{bmatrix} \quad II-32$$

The I matrix would be:

$$I = \begin{bmatrix} 0 \\ 0 \\ I_s \left(e^{\frac{v_3}{V_T}} - 1 \right) \\ 0 \\ 0 \end{bmatrix} \quad II-33$$

The unknowns are:

$$V = \begin{bmatrix} V(-H) \\ V(0) \\ V(H) \end{bmatrix}, (-H) = \begin{bmatrix} V_1(-H) \\ V_2(-H) \\ V_3(-H) \\ V_4(-H) \\ V_5(-H) \end{bmatrix}, V(0) = \begin{bmatrix} V_1(0) \\ V_2(0) \\ V_3(0) \\ V_4(0) \\ V_5(0) \end{bmatrix}, \text{ and } V(H) = \begin{bmatrix} V_1(H) \\ V_2(H) \\ V_3(H) \\ V_4(H) \\ V_5(H) \end{bmatrix} \quad II-34$$

The time-derivative matrix Ω in this example is

$$\Omega = \begin{bmatrix} -jH_2\omega & & & & \\ & -jH_1\omega & & & \\ & & 0 & & \\ & & & jH_1\omega & \\ & & & & jH_2\omega \end{bmatrix} \quad \text{II- 35}$$

II.2.5. Solving the Harmonic Balance Equations

After formulating the equation of Harmonic Balance in the previous section, we end up with a set of vectors and matrices that represent a set of nonlinear equations with the harmonics of the circuit variables as the unknowns. Then the HB equations can be solved using an iterative method such as Newton-Raphson.

The equation of NR is presented as follow:

$$\mathbf{V}^{k+1} = \mathbf{V}^k - \mathbf{j}_k(\mathbf{V}^k) \times \mathbf{F}(\mathbf{V}^k) \quad \text{II- 36}$$

Where

$$\mathbf{F}(\mathbf{V}) = \mathbf{YV} + \mathbf{I} + \mathbf{j}\Omega\mathbf{Q} - \mathbf{B} \quad \text{II- 37}$$

And $\mathbf{j}_k(\mathbf{V}^k)$ is a matrix of the same size as matrix \mathbf{Y} , it is called Jacobian matrix and it is the derivative of equation 2.63 with respect to \mathbf{V} .

Before constructing Jacobian matrix, it is good to mention that NR has a problem of convergence. As stated earlier in this chapter, one remedy to the problem mentioned above is the continuation method.

The remedy found to solve the problem of convergence for NR is the continuation method. It consists of introducing a new factor to the system, let's call it continuant α , then α is increased from 0 to 1, in some increment rate.

The main idea of the continuation method is to augment the harmonic balance equation using factor α to an auxiliary system $\mathbf{H}(\mathbf{x}, \alpha)$, where α is chosen in a way that makes the solution $\mathbf{H}(\mathbf{x}, \mathbf{0})$ trivial to obtain, and

$H(\mathbf{x}, \mathbf{1}) = \mathbf{F}(\mathbf{x})$ identically in \mathbf{x} . Therefore, the choice for the parameter α would be the power of the time-varying forcing input, as shown in equation (below) .

$$H(\mathbf{X}, \alpha) = \mathbf{Y}\mathbf{X} + \mathbf{I}(\mathbf{X}) + \mathbf{j}\Omega\mathbf{C}\mathbf{X} + \mathbf{B} = \mathbf{0}$$

Where $\mathbf{B} = \mathbf{I}_{dc} + \alpha\mathbf{I}_t$, therefore the augmented equation becomes:

$$H(\mathbf{X}, \alpha) = \mathbf{Y}\mathbf{X} + \mathbf{I}(\mathbf{X}) + \mathbf{j}\Omega\mathbf{C}\mathbf{X} + \mathbf{I}_{dc} + \alpha\mathbf{I}_t = \mathbf{0}$$

In equation above, \mathbf{I}_{dc} is the biasing DC sources, and \mathbf{I}_t is the time-varying input in the circuit. Obviously, at $\alpha = \mathbf{0}$, the solution of the above equation is the solution obtained from DC analysis of the circuit, however, at $\alpha = \mathbf{1}$, the solution of the above equation is identical to the original Harmonic Balance equation $H(\mathbf{X}, \mathbf{1}) = \mathbf{F}(\mathbf{X})$

This type of Continuations methods is called “Internal Parameter Embedding”, because the factor α is embedded in the equation, as we can notice element \mathbf{B} has been replaced by $\mathbf{I}_{dc} + \alpha\mathbf{I}_t$.

However, it's essential to mention that there are different approaches for the continuation methods that might formulate the auxiliary system of HB equation differently. This type of continuation methods is called “External parameter Embedding”, because while constructing the auxiliary system $H(\mathbf{X}, \alpha)$, the parameter α remains external to the original equation $\mathbf{F}(\mathbf{X})$, the presentation of the auxiliary system could be formulated as follows:

$$H(\mathbf{X}, \alpha) \cong (\mathbf{1} - \alpha)(\mathbf{X} - \mathbf{X}_a) + \alpha\mathbf{F}(\mathbf{X})$$

When α is zero, then the solution of $H(\mathbf{X}, \mathbf{0})$ is trivial given as:

$$\mathbf{X} = \mathbf{X}_a$$

And when the parameter α is increased to one, then the auxiliary system $H(\mathbf{X}, \mathbf{1})$ has a solution given as:

$$H(\mathbf{X}, \mathbf{1}) = \mathbf{F}(\mathbf{X})$$

However, this formulation of the continuation methods is not considered in this thesis, and the “Internal Parameter Embedding” is the only formulation that we are concerned about.

Consider that N is the number of nodes and K is the number of harmonics in the circuit. Then Jacobian matrix could be constructed using the following equation

$$J(\mathbf{V}^k) = \left. \frac{dF}{dV} \right|_{V=V^k} = \left(\frac{dI}{dV} + j\Omega \frac{dQ}{dV} \right) \Big|_{V=V^k} + Y \quad II-38$$

In fact, the terms of Jacobian matrix J are:

$$\frac{\partial F_{n,k}}{\partial V_{m,l}} = Y_{n,m} + \frac{\partial I_{G;n,k}}{\partial V_{m,l}} + jk\omega \frac{\partial Q_{n,k}}{\partial V_{m,l}} \quad II-39$$

Where n and m refers to the nodes of the circuit, whereas k and l indicate the harmonic number.

$$Y_{n,m} = \begin{cases} Y_{n,m}(k\omega), & k = l \\ \text{zero otherwise} \end{cases} \quad II-40$$

$$\frac{\partial I_{G;n,k}}{\partial V_{m,l}} = \frac{1}{T} \int_{-\frac{1}{2}T}^{\frac{1}{2}T} \frac{\partial i_{g,n}(t)}{\partial V_m(t)} e^{-j(k-l)\omega t} dt \quad II-41$$

And

$$\frac{\partial Q_{n,k}}{\partial V_{m,l}} = \frac{1}{T} \int_{-\frac{1}{2}T}^{\frac{1}{2}T} \frac{\partial q_{g,n}(t)}{\partial V_m(t)} e^{-j(k-l)\omega t} dt \quad II-42$$

Where T is considered to be the period of the fundamental excitation frequency. These terms are the coefficients of the Fourier-series of the derivative waveforms of the nonlinear elements, since it's cheaper to apply derivative in time domain then transfer it to frequency domain rather than differentiating directly in the frequency domain.

The following section will show an example of how to construct the Jacobian matrix of a simple nonlinear circuit that contains one single nonlinear element.

Consider the simple circuit in Figure II- 7

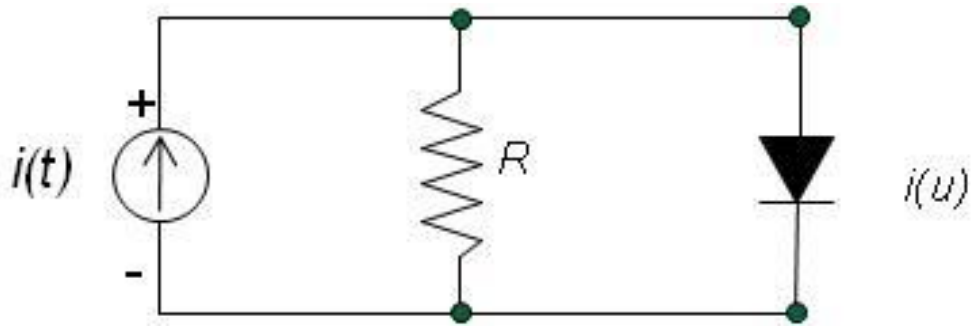


Figure II- 7: simple circuit with single nonlinearity

Where I_D is the nonlinear current and it is equal to

$$I_D = I_s \left(e^{\frac{v(t)}{V_T}} - 1 \right) \quad II- 43$$

For simplicity, the Jacobian equation will be modified a little, since there is no nonlinear charge in the circuit shown in figure 2.7

$$J = Y_{n,m} + \frac{\partial I_{n,k}}{\partial V_{m,l}} \quad II- 44$$

Consider that $g(t)$ is the derivative of I_D in the time domain, then

$$\begin{aligned} g(t) &= \left. \frac{\partial I_d}{\partial v(t)} \right|_{v=v_1(t)} \\ &= \frac{d}{dv} \left[I_s \left(e^{\frac{v(t)}{V_T}} - 1 \right) \right] \\ &= \frac{I_s}{V_T} e^{\frac{v(t)}{V_T}} \end{aligned}$$

Applying the Fourier transform to $g(t)$ gives the frequency components G_k .

The Jacobian will have the following form

$$J = \begin{bmatrix} \frac{\partial F_{1,-k}}{\partial V_{1,-k}} & \dots & \frac{\partial F_{1,-k}}{\partial V_{1,0}} & \dots & \frac{\partial F_{1,-k}}{\partial V_{1,k}} \\ \vdots & \ddots & \vdots & & \vdots \\ \frac{\partial F_{1,0}}{\partial V_{1,-k}} & \dots & \frac{\partial F_{1,0}}{\partial V_{1,0}} & \dots & \frac{\partial F_{1,0}}{\partial V_{1,k}} \\ \vdots & & \vdots & \ddots & \vdots \\ \frac{\partial F_{1,k}}{\partial V_{1,-k}} & \dots & \frac{\partial F_{1,k}}{\partial V_{1,0}} & \dots & \frac{\partial F_{1,k}}{\partial V_{1,k}} \end{bmatrix} \quad II- 45$$

Replacing real values into J would give

$$J = \begin{bmatrix} Y_{1,1}(-H\omega) + G_0 & \cdots & G_H & \cdots & G_{2H} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ G_{-H} & \cdots & Y_{1,1}(0) + G_0 & \cdots & G_H \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ G_{-2H} & \cdots & G_{-H} & \cdots & Y_{1,1}(k\omega) + G_0 \end{bmatrix} \quad II-46$$

The main computational complexity of NR methods comes from the need to calculate the inverse Jacobian, therefore applying LU factorization is equivalent to $\mathcal{O}(N^3)$ where N is the size of the circuit.

II.2.6. Iterative approaches

In order to overcome the expensive factorization caused by Jacobian, Iterative approaches based on Krylov-subspace techniques [31] have been introduced in circuit simulation, namely, the generalized minimal residual method (GMRES)

GMRES is a projection method that was published by Saad and Schultz in 1986 [31], it is based on solving a system of linear equations:

$$Ax = b$$

GMRES tries to find the best solution x_k from the Krylov space $\mathcal{K}_k(A, b)$, that is, to find the residual as small as possible over $\mathcal{K}_k(A, b)$, i.e, x_k solves the least squares problem

$$\min_{x \in \mathcal{K}_k(A, b)} \|b - Ax\| \quad II-47$$

GMRES solves this least squares problem by constructing an orthonormal basis $\{q_1, q_2, \dots, q_k\}$ for $\mathcal{K}_k(A, b)$ using Arnoldi's method, which is a version of the Gram-Schmidt procedure tailored to Krylov spaces. Starting with the normalized right-hand Side $q_1 = b/\|b\|$ as a basis for $\mathcal{K}_1(A, b)$, Arnoldi's method recursively constructs an orthonormal basis for $\mathcal{K}_{j+1}(A, b)$ from an orthonormal basis for $\mathcal{K}_j(A, b)$ by orthogonalizing the vector Aq_j from $\mathcal{K}_{j+1}(A, b)$ against the previous space $\mathcal{K}_j(A, b)$, that is:

$$\hat{q}_{j+1} = Aq_j - (h_{1j}q_1 + \cdots + h_{jj}q_j)$$

Where $h_{ij} = q_i^* Aq_j$ and $*$ denotes the conjugate transpose. The new basis vector becomes:

$$\mathbf{q}_{j+1} = \hat{\mathbf{q}}_{j+1} / \|\hat{\mathbf{q}}_{j+1}\|$$

Then, collecting the orthonormal basis vectors for $\mathcal{K}_j(\mathbf{A}, \mathbf{b})$ in a matrix, $\mathbf{Q}_j = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_j)$, allows to obtain the decomposition associated with Arnoldi's method:

$$\mathbf{A}\mathbf{Q}_j = \mathbf{Q}_{j+1}\mathbf{H}_j$$

Where \mathbf{H}_j is an upper Hessenberg matrix of size $(j+1) \times j$.

In the context of the least squares problem (equ. $\min_{\mathbf{x} \in \mathcal{K}_k(\mathbf{A}, \mathbf{b})} \|\mathbf{b} - \mathbf{A}\mathbf{x}\|$) this means if we consider $\mathbf{x} \in \mathcal{K}_1(\mathbf{A}, \mathbf{b})$, then $\mathbf{x} = \mathbf{Q}_k \hat{\mathbf{x}}$ for some $\hat{\mathbf{x}}$, so that:

$$\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{Q}_k \hat{\mathbf{x}} = \mathbf{Q}_{k+1} \mathbf{H}_k \hat{\mathbf{x}}$$

And

$$\mathbf{b} = \beta \mathbf{q}_1 = \beta \mathbf{Q}_{k+1} \mathbf{e}_1$$

Where $\beta = \|\mathbf{b}\|$ and \mathbf{e}_1 is the first column of the identity matrix. The least squares problem in iteration K of GMRES reduces to

$$\min_{\mathbf{x} \in \mathcal{K}_k(\mathbf{A}, \mathbf{b})} \|\mathbf{b} - \mathbf{A}\mathbf{x}\| = \min_{\hat{\mathbf{x}}} \|\beta \mathbf{e}_1 - \mathbf{H}_k \hat{\mathbf{x}}\|$$

GMRES algorithm could be written as follows:

Iteration 0:

Step 1: Initialize $\mathbf{x}_0 = 0$

Step 2: $\mathbf{q}_1 = \mathbf{b}/\beta$

Step 3: $\mathbf{Q}_1 = \mathbf{q}_1$

Iteration ≥ 1 :

Step 1: Orthogonalize: $\hat{\mathbf{q}}_{k+1} = \mathbf{A}\mathbf{q}_k - \mathbf{Q}_k \mathbf{h}_k$

$$\mathbf{h}_k = \mathbf{Q}_k^* \mathbf{q}_k$$

Step 2: Normalize $\mathbf{q}_{k+1} = \hat{\mathbf{q}}_{k+1} / \|\hat{\mathbf{q}}_{k+1}\|$

Step 3: Update $\mathbf{Q}_{k+1} = (\mathbf{Q}_k \ \mathbf{q}_{k+1})$, and $\mathbf{H}_k = \begin{pmatrix} \mathbf{H}_{k-1} & \mathbf{h}_k \\ 0 & \|\hat{\mathbf{q}}_{k+1}\| \end{pmatrix}$

When $K = 1$, the first column of \mathbf{H}_k is removed

Step 4: Solve the least squares problem $\min_{\hat{x}} \|\beta e_1 - H_k \hat{x}\|$, find a solution for \hat{x}

Step 5: Set the approximation of the solution to be $x_k = Q_k \hat{x}_k$

However, this method has several limitations such as it is not always guaranteed to converge; also it has to use a pre-conditioner in each iteration in order to improve the convergence.

III. Reference

- [1] P. Feldmann, R.W. Freund. “*Efficient linear circuit analysis by Padé approximation via the Lanczos process,*” Proceedings of EURO-DAC'94 with EURO-VHDL'94 (IEEE Computer Society Press, Los Alamitos, CA), pp. 170-175, 1994.
- [2] L. M. Silveira, M. Kamen, I. Elfadel and J. White, “*A coordinate transformed Arnoldi algorithm for generating guaranteed stable reduced-order models for RLC circuits,*” in Technical Digest ICCAD, pp. 2288-294, Nov. 1996.
- [3] A. Odabasioglu, M. Celik, and L.T. Pileggi, “*PRIMA: passive reduced-order interconnect macromodeling algorithm,*” IEEE Trans. Computer-Aided Design, vol. 17, pp, 645-654, Aug. 1998.
- [4] Y. Chen and J. White, “*A quadratic method for nonlinear model order reduction,*” Tech. Proc. of the 2000 International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, MSM 2000, San Diego, USA, pp. 477-480, March 2000.
- [5] M. Rewienski and J. White, “*A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micro machined Devices,*” In Proc. ICCAD, Nov 2001.
- [6] E. Gad, , R. Khazaka, M. Nakhla, R. Griffith, “*A circuit reduction technique for finding the steady-state solution of nonlinear circuits,*” IEEE Trans. Microwave Theory Tech, Vol 48, No 12, p 2389-2396. Dec, 2000
- [7] J. Vlach and K. Singhal, “*Computer Methods for Circuit Analysis and Design,*” New York: Van Norstand Reinolds, 1983.
- [8] C. W. Ho, A. E. Ruehli, and P. A. Brennan, “*The modified nodal approach to network analysis,* *IEEE Trans,*” Circuits Syst., vol. 22, pp. 504-509, June 1975.
- [9] E. L. Allgower and K. Georg, “*Numerical continuation methods: an introduction. Springer series in computational mathematics,*” Springer-Verlag, 1990.
- [10] Nastov et al, “*Fundamentals of Fast Simulation Algorithms for FR Circuits,*” Proceedings of the IEEE. Vol. 95, N0. 3, March 2007
- [11] S. M. Roberts and J. S. Shipman, “*Two-Point Boundary Value Problems,*” New York: Elsevier, 1972.
- [12] K. S. Kundert, J. White, and A. Sangiovanni-Vincentelli, “*A mixed frequency-time approach for distortion analysis of switching filter circuits,*” IEEE J. Solid-State Circuits, vol. 24, pp. 443–451, Apr. 1989.
- [13] J. Roychowdhury, “*Efficient methods for simulating highly nonlinear multi-rate circuits,*” in Proc. Design Automation Conf., pp. 269–274, 1997.
- [14] K. S. Kundert, J. K. White, and A. Sangiovanni-Vincentelli, “*Steady-State Methods for Simulating Analog and Microwave Circuits,*” Boston: Kluwer Academic, 1990.
- [15] J. W. Demmel, “*Applied Numerical Linear Algebra,*” Philadelphia, PA: SIAM Publishers, 1997.
- [16] J. D. Lambert, “*Computational methods in ordinary differential equations. Introductory mathematics for scientists and engineers,*” London Wiley, 1973.

- [17] J. M. Ortega, *“Iterative solution of nonlinear equations in several variables. Computer science and applied mathematics,”* New York: Academic Press, 1970.
- [18] T. J. Aprille, Jr. and T. N. Trick, *“A computer algorithm to determine the steady-state response of nonlinear oscillators,”* IEEE Trans. Circuit Theory, vol. 19, pp. 354–360, July 1972.
- [19] S. Skelboe, *“Conditions for quadratic convergence of quick periodic steady-state methods,”* IEEE Trans. Circuits Syst., vol. 29, pp. 234–239, Apr. 1982.
- [20] S. Skelboe, *“Computation of the periodic steady-state response of nonlinear networks by extrapolation methods,”* IEEE Trans. Circuits Syst., vol. 27, pp. 161–175, Mar. 1980.
- [21] T. J. Aprille, Jr. and T. N. Trick, *“Steady-state analysis of nonlinear circuits with periodic inputs,”* Proceedings IEEE, vol. 60, pp. 108–114, Jan. 1972.
- [22] R. Telichevesky, K.S. Kundert, J.K. White, *“Efficient Steady-State Analysis based on Matrix-Free Krylov-Subspace Methods,”* In: Proc. DAC'95, San Francisco, pp. 480-484, 1995
- [23] V. Rizzoli, C. Cecchetti, A. Lipparini, and F. Mastri, *“General-Purpose Harmonic Balance Analysis of Nonlinear Microwave Circuits Under Multitone Excitation, IEEE Transactions on Microwave Theory and Techniques,”* Vol. 36, No. 12, pp. 1650-1660, December 1988.
- [24] V. Rizzoli, A. Lipparini, A. Costanzo, F. Mastri, A. Neri, D. *“Masotti, State-of-the-Art Harmonic-Balance Simulation of Forced Nonlinear Microwave Circuits by the Piecewise Technique,”* IEEE Transactions on Microwave Theory and Techniques, Vol. 40, No. 1, January 1992.
- [25] M.S. Nakhla and J. Vlach, *“A piecewise harmonic-balance technique for determination of periodic response of nonlinear systems,”* IEEE Trans. Circuits Syst., vol. 23, pp. 85-91, Feb. 1976
- [26] S. A. Maas, *“Nonlinear Microwave Circuits,”* New York: IEEE Press, 1996.
- [27] V. Rizzoli, F. Mastri, and D. Masotti, *“General Noise Analysis of Nonlinear Microwave Circuits by Piecewise Harmonic-Balance Technique,”* IEEE Trans. on Microwave Theory and Techniques, Vol. 42, No. 5, May 1994.
- [28] V. Rizzoli, A. Costanzo, D. Masotti, F. Mastri, *“Computer-Aided Analysis of Near-Carrier Noise in RF-Microwave Frequency Converters,”* Int. Journal of RF and Microwave CAE, Vol. 9, 1999, pp. 449-467, John Wiley & Sons, Inc.
- [29] A. Demir, A. Mehrotra, and J. Roychowdhury, *“Phase Noise in Oscillators: A Unifying Theory and Numerical Methods for Characterization,”* IEEE Transactions on Circuits and Systems, Vol. 47, No. 5, May 2000.
- [30] A. Demir, D. Long, and J. Roychowdhury, *“Computing Phase Noise Eigen-functions Directly from Harmonic Balance/Shooting Matrices,”* IEEE Fourteenth International Conference on VLSI Design, 2001.
- [31] K. Kundert, *“The Designer’s Guide to SPICE and Spectre. Nowell,”* MA.:Kluwer Academic Publishers, 1995.
- [32] K. Kundert and Sangiovanni-Cincentelli, *“Simulation of nonlinear circuits in the frequency domain,”* IEEE Trans. Computer-Aided Design, vol. 5, pp. 521-535, Oct. 1986.
- [33] K. Mayaram, D. C. Lee, S. Moinian, D. Rich, and J. Roychowdhury, *“Overview of computer-aided analysis tools for RFIC simulation: Algorithms, features, and limitations,”* in Proc. IEEE Custom Integrated Circuits Conf., pp. 505–512, 1997.
- [34] K. S. Kundert, *“Accurate fourier analysis for circuit simulators,”* in Proc. IEEE Custom Integrated Circuits Conf., pp. 25–28, May 1994.

- [35] V. Rizzoli, F. Mastri, F. Sgallari, V. Frontini, “*The Exploitation of Sparse-Matrix Techniques in Conjunction with the Piecewise Harmonic-Balance Method for Nonlinear Microwave Circuit Analysis*,” IEEE MTT-S Digest, 1990
- [36] V. Rizzoli, D. Masotti, and F. Mastri, “*Full Nonlinear Analysis of Microwave Circuits*,” IEEE MTT-S Digest, 1994, pp 961-963.
- [37] V. Rizzoli, F. Mastri, and D. Masotti, “*A General-Purpose Harmonic Balance Approach to the Computation of Near-Carrier Noise in Free-Running Microwave Oscillators*,” IEEE MTT-S Digest, 1993, pp. 309-312.
- [38] A. Demir, A. Mehrotra, and J. Roychowdhury, “*Phase Noise and Timing Jitter in Oscillators*,” IEEE Custom Integrated Circuits Conference, 1998.
- [39] R. L. Burden, J. D. Faires, “*Numerical Analysis*,” 5th Edition, PWS Publishing Company, 20 Park Plaza, Boston, MA 02116-4324
- [40] E. Lahaye, “*Une méthode de resolution d’une categorie d’équation transcendantes*,” C. R. Acad. Sci. Paris 198, 1840-1842
- [41] Y. Saad, *Iterative Methods for Sparse Linear Systems*,” Boston, MA: ITP, 1996.
- [42] B. L. Ehle, “*On Padé Approximations to the Exponential Function and A-stable Methods for the Numerical Solution of Initial Value Problems*,” Report CSRR 2010, University of Waterloo, Department of Applied Analysis and Computer Science, March 1969

Chapitre III :Model Order Reduction of circuit equations

I. Introduction

The idea of reducing a system model to an equivalent lower dimensional system was introduced first in the area of control system. Genesio and Milanese in [1] mentioned that research about MOR was intensively considered during the 60s and 70s. Where, the main interest in this domain was to find a minimal size representation of a linear dynamic system, and to control complex system using smaller and simpler linear models, while preserving input/output of the original system. However, The rapid growth of the computer capabilities that was witnessed in the same era has added a special touch to MOR domain, in a sense that more powerful tools gave researchers and designers the opportunity to simulate very large models resulting from both very complex dynamic systems and distributed systems, Moreover, such tools encouraged researchers to think of expanding the applicability of these tools to a much larger set of problem using MOR.

In 1981 [2], Moore [3][4] published the method of a truncated balanced realization as a preliminary step before the reduction process. A balanced realization was shown as the linear transformation which removes any scaling effects over the internal representation of a state-space model.

Kabamba [5] presented an improvement over the balanced realization concept by introducing the balanced gains instead of the principal values that was used by Moore, for weighting the contribution of the states to the input-output behavior. Additional research into the balancing strategy followed as described for example in [6] - [9].

In 1987, Proper Orthogonal Decomposition method was proposed by Sirvovich [10], where the problem of truncating the state-space realization was also considered as an equivalent to the geometrical projection of the original formulation to a reduced state-space. This method gives the chance to transform the reduction process into the extraction of an effective projection base, in which, the approximation to the solution of a dynamic linear system is contained in a Krylov subspace[11] that can be defined from the original system. In the meanwhile, Krylov subspaces was very much considered as an alternative solution for the eigenvalue problem of linear systems, along with the iteration methods that are meant to derive the related orthogonal bases for the geometrical subspaces, such as as Lanczos [12], Arnoldi [13] and Padé [14] techniques.

The idea of MOR was already established at early stages, but it had a lack of effective techniques that could represent it. However, this soon changed especially when the demands for very large interconnection electrical networks in VLSI in the early 90s became the test-bench for MOR techniques, such as PVL [4], and PRIMA [15] proved to be very efficient methods for simulation of such very large systems. The importance of such techniques is that they allow the designers to reduce systems into a compact model so that very large interconnection problems could be simulated in a reasonable time. However, this was an important issue because traditional simulators such as SPICE [16] were not able to deal with such systems without an excessive memory use of the computer and without acquiring extremely long simulation time.

After the success of PRIMA by overpassing the passivity problem, efficient MOR algorithms have been developed and applied in many different areas to deal with the simulation of large dimensional problems [17][18][19].

Nowadays, MOR of linear systems is considered to be a mature area of research and it contains very efficient algorithms that are used by contemporary and advanced simulators. In this domain, one of the main focuses of research is to find algorithms that can offer error bounds for the reduced model while keeping the accuracy of the performance that was already obtained by the established techniques. Moreover, stability is also not guaranteed in any of the current Krylov based algorithms, consequently an additional research push is to merge the stable simple truncation of state technique with the efficient computational Krylov subspace based technique.

This chapter is intended to be an introductory chapter of Model Order Reduction techniques of both linear and nonlinear systems. It is divided into two parts, the first part of this chapter will review a classification of the main methods and algorithms used for model order reduction (MOR) of linear systems. The second part points on the current and more challenging problem of MOR of nonlinear systems. Then the review will be followed by the important ideas currently being used to deal with the general problem of MOR for nonlinear systems, where we show the difficulties found when trying to reduce the order of these types of problems.

II. Model Order Reduction

II.1. MOR for Linear Systems

Several definitions could take place once introducing MOR, due to the various areas of research that MOR fits in. Initially, MOR is a field that was developed in the area of systems and control theory that concentrates on the study of the properties of dynamical systems in application in order to reduce their complexity, while preserving the input-output port behavior as possible. MOR was not conserved for the domain of control systems, in reality it was taken up by numerical mathematicians, especially when methods such as PVL were published.

Currently, MOR is an enriching field of research for both control theory and numerical analysis. Its main idea is to develop methodologies that make it possible to the designers to deal with a simplified system that can imitate the behavior of the original large system, in order to perform simulations within an acceptable amount of time and a moderate storage capacity, while keeping a reliable outcome. Such simplifications make computational simulations of complex and large systems very efficient, however, MOR is enrolled by some limitations that should be respected:

- *Good Accuracy*: Every MOR method should attain a good accuracy in a way that a minimal error on the behavior of the input/output relationship of the original system against the reduced one should be achieved.
- *System properties preservation*: for many types of problems, it is preferred that some specific properties or parameters of the original system would be conserved, such as stability, passivity, etc.
- *Computational efficiency*: Two aspects should be studied in this part. The first is the computational efficiency of the generation of the compact realization, where the user should decide whether the cost of that generation is reasonable in comparison with the evaluation of the original system which is usually very large or not. The second is how computationally efficient is the evaluation of the compact realization once compared to the original.
- *User independency*: It is highly recommended that each and every MOR algorithm would offer a minimal user intervention while generating the compact realization of the original system. It is desirable to keep the algorithm independent of the specific characteristics of any system that is intended to work on as much as possible.

There are two well defined sub areas in this field, MOR of linear and MOR of nonlinear systems. As expected, linear MOR is a well-known and developed area of research, and that by itself it is used as the basis for research in the newer and still developing field of nonlinear MOR. We start with a description of the linear case and follow with an analysis of the current state of development for the nonlinear area

II.1.1. Polynomial approximations of the transfer function

Polynomial approximation of the transfer function is a MOR technique that is based on approximating the system using the truncated power series expansion of its transfer function. It is normally performed on the transfer function representation of the system in the frequency domain.

For a better understanding of this method, let's consider a linear state-space representation of a dynamic system:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\dot{\mathbf{x}} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}^T\mathbf{x} \end{aligned} \quad \text{III-1}$$

Where vector \mathbf{y} is the output of the system, vector \mathbf{u} is the input of the system, and vector \mathbf{x} is the state vector of the system, however, matrix \mathbf{A} is the transformation matrix of the system, \mathbf{B} is the input connectivity matrix, and \mathbf{C} is the output connectivity matrix for the system model.

Then the transfer function of the above system could be presented in the frequency domain as follows:

$$\mathbf{H}(s) = \mathbf{C}^T(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} \quad \text{III-2}$$

Where \mathbf{I} is the identity matrix, and the output of the system could be characterized as follows:

$$\mathbf{y}(s) = \mathbf{H}(s)\mathbf{u}(s)$$

Expanding the transfer function $\mathbf{H}(s)$ around $s = \mathbf{0}$ would give a power series of the transfer function as follows:

$$\mathbf{H}(s) = \mathbf{C}^T(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} = -\mathbf{C}^T\mathbf{A}^{-1}\mathbf{B} - \mathbf{C}^T\mathbf{A}^{-2}\mathbf{B} \times s - \dots, -\mathbf{C}^T\mathbf{A}^{-n}\mathbf{B} \times s^{n-1} - \dots$$

Consider $-\mathbf{C}^T\mathbf{A}^{-1}\mathbf{B}$ to be replaced by \mathbf{m}_0 and $-\mathbf{C}^T\mathbf{A}^{-2}\mathbf{B}$ by \mathbf{m}_1 , so on and so forth up till n-1. Those coefficients – also could be called moments of the system – are mainly all what the polynomial approximation techniques is in need for in order to find a reduced representation of the system, in a way

that a reduced system could contain several of these terms with an adequate approximation of the coefficients, e.g. Padé approximation [14].

Moreover, the expansion of the transfer functions is not necessary to happen only around $\mathbf{s} = \mathbf{0}$, because in some systems, it is better to use higher \mathbf{s} in order to achieve a better match.

The advantages of this method are that the resulting model has a good match for system behavior at the expansion point and that this approach is computationally inexpensive in comparison with other approaches. However, the disadvantages of this method are that it is difficult to maintain stability of the compact model, also, it is difficult to estimate the error bound on the compact model, and to apply this method to Multi-Input/Multi-Output (MIMO) systems. It doesn't guarantee passivity for the compact system, even when the original system is passive by itself, and it is well known that the lack of conserving passivity leads to instability in the system[15], in which the compact system will be instable. However, several modifications recently took place to preserve passivity of the compact model, also to make it possible to apply polynomial approximation of the transfer function to a MIMO systems [15][20]

II.1.2. Sub-space projection techniques.

Sub-space projection technique is a model order reduction technique that consists of projecting system under study into a subspace of a much smaller system in size [21]. Obviously, this technique was inspired from geometry theories, as its own name states. For better understanding, consider a system of $\mathbf{N} \in \mathfrak{R}^N$ size in its original state-space, sub-space projection method consists of projecting the original system of large size into a subspace $\mathbf{V} \in \mathfrak{R}^q$ of lower dimensionality where $q \ll N$, in a condition that the reduced system \mathbf{V} should keep its behavior from the point of view of the output nodes of interest.

However, the subspace \mathbf{V} is not unique, because as it is known from linear analysis, any base characterizing a subspace is not unique since there are an infinite number of equivalent bases that can define such a subspace.

Let us consider the general state representation for a linear system one more time:

$$\begin{cases} E\dot{\mathbf{x}} + A\mathbf{x} = B\mathbf{u} \\ \mathbf{y} = C^T\mathbf{x} \end{cases} \quad \text{III-3}$$

with initial condition $\mathbf{x}(0) = \mathbf{x}_0$. where, $\mathbf{x}(t) \in \mathfrak{R}^N$ is the state vector, $\mathbf{u}(t) \in \mathfrak{R}^m$ is the input excitation vector, $\mathbf{y}(t) \in \mathfrak{R}^p$ is the output vector, t is the time variable N is the state space dimension and m and p are the number of input / outputs, respectively. Matrices $E, A \in \mathfrak{R}^{N \times N}$ are system matrices, $B \in \mathfrak{R}^{N \times m}$ and $C \in \mathfrak{R}^{N \times p}$ are input and output distribution arrays, respectively.. In most practical cases, it's assumed that $(m, p) \ll N$ and $m \geq p$.

Assume that we have an adequate projection base $V \in \mathfrak{R}^{q \times N}$ that has a good approximation of the original system, in which it allows us to project the state-vector $\mathbf{x} \in \mathfrak{R}^n$ into the subspace defined by this base as:

$$\begin{cases} \mathbf{x} = V\hat{\mathbf{x}} \\ E_q = V^T E V \\ A_q = V^T A V \\ B_q = V^T B \\ C_q^T = C^T V \end{cases} \quad \text{III-4}$$

where $\hat{\mathbf{x}} \in \mathfrak{R}^q$ is the state-vector for the projected system and $q \ll N$. The reduced system using this transformation becomes as follows:

$$\begin{cases} E_q \hat{\mathbf{x}} + A_q \hat{\mathbf{x}} = B_q \mathbf{u} \\ \hat{\mathbf{y}} = C_q^T V \hat{\mathbf{x}} \end{cases} \quad \text{III-5}$$

Where $\hat{\mathbf{y}}$ is a close representation for \mathbf{y} , since we assumed that projection V is a good approximation.

However, one of the more important issues in this technique is how to define suitable projection bases of the original system that are computationally efficient and that preserve good accuracy of system output. Fortunately, the success of this technique is closely related to Krylov subspaces. The recent progress of projection based model order reduction for large linear systems came from the development of efficient projection algorithms based on Krylov subspaces methods. Krylov subspace theory states that the solution of a linear system of the form in III-1, equation III-4 might be approximated in a subspace of order q which also corresponds to the rank of A , in which, this subspace can be very small compared to the original size

of the problem and more importantly researchers have developed powerful iterative methods that allow us to find such subspace, such as the Arnoldi algorithm and Lanczos.

II.1.3. Padé Approximation

Padé approximation consists of finding a reduced model of order k , in a way that the first $2k$ moments of the transfer function of the reduced model constructed around a frequency s_0 would have the same values as the original system.

Consider a SISO system (Single Input, Single Output) where the input and the output are of dimension 1. Consider the equation (III-3), B and C become b and l . Also, E, A, b and l are presented as D, r and l , therefore, the system becomes:

$$\begin{cases} D = -(G + s_0 E)^{-1} E \\ r = -(G + s_0 E)^{-1} b \end{cases} \quad \text{III-6}$$

Therefore, the system is now characterized by matrix D only, instead of matrices E and A . The transfer function is expressed as a function of D, r and l . At this point, the transfer function of the reduced model of order k must be found where the first $2k$ moments are equivalent to the ones of the original system. The computation of the transfer function moments is generally (ill-conditioned) that cannot be solved unless the order is very small, this is why, we use (sub-space) Krylov of dimension k which is defined as follows:

$$\begin{aligned} K_r(D, r) &= \text{span}\{r, Dr, D^2r, \dots, D^{k-1}r\} \\ K_r(D^T, l) &= \text{span}\{l, D^T l, (D^T)^2 l, \dots, (D^T)^{k-1} l\} \end{aligned} \quad \text{III-7}$$

The vectors which forms the sub-space of Krylov contain the required information however, they are not practically of a good use to construct the vectors of base. Thus, the projection base is constructed using Lanczos process, this process is generally known as PVL (Pade via Lanczos) and the most known algorithms for PVL are PRIMA, MPVL [11].

Lanczos process generates two bases, W_k and V_k that are two Krylov bases (equation III-7).

This process tridiagonalizes the matrix \mathbf{D} , once the process reaches the k th order, then a tridiagonal matrix \mathbf{T}_k is obtained, this matrix is used to define the approximation of the transfer function $\mathbf{H}_k(\mathbf{s})$ which is an approximation of Padé of the original transfer function.

The reduced model is first obtained in the time domain, then it passes by the transformation which lead to a presentation of $\mathbf{D}, \mathbf{r}, \mathbf{l}$, afterwards, the change of variables $\mathbf{x} = \mathbf{V}_k \hat{\mathbf{x}}$ is made, then the projection of the dynamics equation can be applied using the base \mathbf{W}_k .

Another approach is followed by [22] and [23], which uses the Arnoldi process in order to determine the Krylov base for development around the frequency $\mathbf{s}_0 = \mathbf{0}$. The algorithm of Arnoldi generates a base which serves for both the change of variable and the projection of the unchanged equation with respect to the representation in equation (III-5).

This algorithm suffers from a disadvantage which is the problem of choosing the order k of the reduced model, however, a method of “error estimation” of Lanczos is proposed in [11] in order to specify the stopping point. Another approach is also considered to overcome the problem of choosing the order k , this approach is called “backward error analysis”, it considers the reduced model as a perturbed complete model.

II.2. MOR for Nonlinear Systems

In the previous section, several methods of linear MOR have been discussed, methods that are currently in action in contemporary CAD tools. However, RF circuits such as power amplifiers, and filters consist not only of linear elements, therefore they cannot be presented as linear system. In order to deal with this limitation, it was necessary to find a new area that aims to reduce nonlinear systems.

Nonlinear model order reduction area has not been successful or easy realm to deal with, due to the complexity and variety of nonlinear systems when compared to the clearly defined structure of linear systems. Research in this field is still ongoing in order to establish a formal base for nonlinear model order reduction. The nowadays nonlinear MOR techniques stem deeply from the linear MOR techniques, with a slight modification applied to them afterwards.

This section will present some of these approaches, namely, linearization methods, Quadratic methods and Orthogonal Decomposition techniques.

II.2.1. Linearization

Linearization methods are the most straightforward approach of model order reduction of nonlinear systems. The idea behind it is based on linearizing the system around a point in its state-space and then applying any of the efficient linear MOR techniques .

The procedure of linearization methods can be summarized in two steps, in the first step consists of expanding the nonlinear system around an operative state-space value using a multidimensional Taylor series expansion technique for example. Then, the second step consists of truncating the resulting sum of infinite terms to only the first order term of the expanded system. Therefore, the linearization procedure leads to obtaining a linear model approximation of the Jacobian matrix of the nonlinear multi-dimension function.

Consider the following nonlinear system:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}^T \mathbf{x}\end{aligned}\tag{III-8}$$

In this nonlinear system, we have $\mathbf{x} \in \mathfrak{R}^N$ is the state vector, $\mathbf{u} \in \mathfrak{R}^k$ is the input vector, $\mathbf{y} \in \mathfrak{R}^m$ is the output vector, $\mathbf{B} \in \mathfrak{R}^{N \times k}$ is the input connectivity matrix, $\mathbf{C}^T \in \mathfrak{R}^{m \times N}$ is the output scanning matrix, and $\mathbf{f}(\cdot) \in \mathfrak{R}^N$ represents a nonlinear set of functions. As a generalization of the linear case introduced above, we can apply a multidimensional Taylor series expansion around $\mathbf{x} = \mathbf{0}$ to Eq III-8, this would be presented as follows:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{0}) + \mathbf{A}_1 \mathbf{x} + \mathbf{A}_2(\mathbf{x} \otimes \mathbf{x}) + \mathbf{A}_3(\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}) + \dots, \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}^T \mathbf{x}\end{aligned}\tag{III-9}$$

It's obvious that the infinite series of terms in the above equation (III-9) matches the system in equation (III-8) around the expansion point. From the equation above, \mathbf{A}_1 represents the Jacobian matrix which is in fact the first order contributions of the state variable, where $\mathbf{A}_1 \in \mathfrak{R}^{N \times N}$, therefore the system is approximated as a truncated expansion series with only the first order contributions or linear effects as follows:

$$\begin{aligned}\dot{\mathbf{x}} &\cong \mathbf{f}(\mathbf{0}) + \mathbf{A}_1 \mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}^T \mathbf{x}\end{aligned}\tag{III-10}$$

Once the above system is obtained, any linear model order reduction techniques can be used in order to obtain the compact model of the system,

Consider Arnoldi process to find a projection base $\mathbf{V} \in \mathfrak{R}^{n \times q}$, via Krylov subspace, where $q \ll n$ and $\mathbf{x} = \mathbf{V}\hat{\mathbf{x}}$ is the well approximation of the original system.

As it was discussed in the previous section, the compact system would be presented differently, maintaining appropriate dimensions of the reduced system:

$$\begin{aligned}\hat{\mathbf{x}} &\cong \mathbf{V}^{-1} \mathbf{f}(\mathbf{0}) + \mathbf{V}^{-1} \mathbf{A}_1 \mathbf{V} \mathbf{x} + \mathbf{V}^{-1} \mathbf{B} \mathbf{u}, \\ \mathbf{y} &= \mathbf{C}^T \mathbf{V} \mathbf{x},\end{aligned}$$

II.2.2. Quadratic Methods

The Quadratic Methods are considered to be an improvement over the previous approach, in these methods, the second order term of expansion presented in equation (III-9) will be included for the state-space approximation of the nonlinear system. Therefore, the nonlinear system will be presented as follow:

$$\begin{aligned}\dot{\mathbf{x}} &\cong \mathbf{f}(\mathbf{0}) + \mathbf{A}_1\mathbf{x} + \mathbf{A}_2(\mathbf{x} \otimes \mathbf{x}) + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}^T\mathbf{x}\end{aligned}\tag{III- 11}$$

Where \mathbf{A}_1 is simply the Jacobian as stated before, and $\mathbf{A}_2 \in \mathfrak{R}^{N \times N^2}$ is the Hessian tensor [24] whose entrees are given by:

$$A_{i,j,k} = \frac{\partial^2 f_i}{\partial x_j \partial x_k}\tag{III- 12}$$

Equation (III-10) is also known as a second order or quadratic state-space representation. A trivial approach is to consider the Krylov sub-spaces defined by the linear part of eq (III-10), that is, to ignore the second order term in order to obtain the projection base $\mathbf{V} \in \mathfrak{R}^{N \times q}$. Therefore, the obtained projection base can be reduced as follows:

$$\begin{aligned}\dot{\mathbf{z}} &\cong \mathbf{V}^{-1}\mathbf{f}(\mathbf{0}) + \mathbf{V}^{-1}\mathbf{A}_1\mathbf{V}\mathbf{z} + \mathbf{V}^{-1}\mathbf{A}_2(\mathbf{V} \otimes \mathbf{V})(\mathbf{z} \otimes \mathbf{z}) + \mathbf{V}^{-1}\mathbf{B}\mathbf{u}, \\ \mathbf{y} &= \mathbf{C}^T\mathbf{V}\mathbf{z}\end{aligned}$$

Therefore:

$$\begin{aligned}\dot{\mathbf{z}} &\cong \hat{\mathbf{f}}(\mathbf{0}) + \hat{\mathbf{A}}_1\mathbf{z} + \hat{\mathbf{A}}_2(\mathbf{z} \otimes \mathbf{z}) + \hat{\mathbf{B}}\mathbf{u}, \\ \mathbf{y} &= \hat{\mathbf{C}}^T\mathbf{z}\end{aligned}$$

The key of success of Quadratic methods is how to find the orthogonal vector base \mathbf{V} , several methods are considered, such as quadratic projection based methods [24]. However, such methods are based on idea of considering the effect of linear and second order elements for the definition of the sub-space of Krylov and constructing the corresponding orthogonal vector base.

However, both methods are limited to a good match of the system only in the neighborhood of the expansion point. the success of these methods depend directly on the fact of how close the original system

to the linear or quadratic representation. This limitation makes Quadratic techniques very efficient for the weakly nonlinear system.

II.2.3. Trajectory Piecewise Linear

This method consists of linearizing the model around various points in the state-space that are not only equilibrium points. The linear model is presented here in pieces based on a learning path (TPWL: Trajectory Piecewise Linear) of [25] [26]. The linearized model pieces has the following form:

$$\dot{\mathbf{x}} = \sum_{i=0}^{s-1} \mathbf{p}_i(\mathbf{x}) [\mathbf{f}(\mathbf{x}_i) + \mathbf{J}_i(\mathbf{x} - \mathbf{x}_i)] + \mathbf{B}\mathbf{u}$$

where the weights \mathbf{p}_i of different linear models depend only on the distance of the current point \mathbf{x} from the linearization points \mathbf{x}_i and verify $\sum_{i=0}^{s-1} \mathbf{p}_i(\mathbf{x}) = \mathbf{1}$ for all \mathbf{x} , and \mathbf{J} is the Jacobian matrix.

Rewienski et al. then use the method of Galerkin. Where \mathbf{V} is the projection base, z becomes the state variable. consider $\mathbf{x} = \mathbf{V}\hat{\mathbf{x}}$, this base could be based on Krylov after concatenation bases determined from the linearized models around different points, or a balanced truncation base [27] or any other base. The final reduced order model has the form:

$$\begin{cases} \dot{\hat{\mathbf{x}}} = \sum_{i=0}^{s-1} \mathbf{p}_i(\hat{\mathbf{x}}) [\mathbf{V}^T \mathbf{f}(\mathbf{x}_i) + \mathbf{V}^T \mathbf{A}_i (\mathbf{V}\hat{\mathbf{x}} - \mathbf{x}_i)] + \mathbf{V}^T \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}^T \mathbf{V}\hat{\mathbf{x}} \end{cases}$$

Where $\sum_{i=0}^{s-1} \mathbf{p}_i(\mathbf{x}) = \mathbf{1}$ for all \mathbf{z}

A simple approach would be to set $\mathbf{p}_i(\mathbf{z})$ to 1 for the linearization point i closest to $\mathbf{V}\mathbf{z}$, where the others weights are set to zero. Here the approach used is more complex and ensures better continuity of weight in

time. The conditions applied on weights also allow flexibility on which it is possible to play in ensuring the stability and / or the passivity of the system.

In practice, the size of the reduced system plays a significant role for the linearization points to be chosen by simple discretization uniform of the space. So they are selected along the most relevant possible training trajectories. Two methods are used to determine the linearization points. One is to use the original model and set a new linearization point when the current state is too far from all linearization points already defined, and the other is to simulate the model starting from the linearized model at the origin and projected based on Krylov associated.

In Rewienski et al, when a state which is very distant from the initial state is reached, a new point of linearization must be defined, which leads to determine the basis of Krylov associated and represent the new projection base by concatenation. Afterwards, the linearized model around this new point is then used instead in order to continue the simulation and so on and so forth. Thus simulations of training trajectories require a reduced cost.

III. Discussion

In this chapter we have reviewed the field of model order reduction for very large dynamic systems for both Linear and Nonlinear systems. This field has reached a level of maturity suitable for its current use as a major tool in the design and analysis on such diverse areas such as VLSI and Micromechanical technology etc....

For the linear MOR area, we presented the problems in the field and described the most widely used and successful techniques that have been developed and implemented. Krylov based MOR techniques were discussed as offering several key advantages that have made them the preferred techniques for MOR

These methods could however be used in the case where the system is nonlinear as well, but it becomes necessary to consider the nonlinear elements while constructing the projection base. some methods require a training form, where the reduced model is generated through certain simulation (Décomposition propre orthogonale, troncature balancée [28]). In this chapter, we reviewed the main NMOR methods that are widely used in this domain, namely, the Quadratic methods, and TPWL.

In the following chapter, we will first review a NMOR method that was recently proposed by Gad et al in [29]. This method studies the construction of the projection base from a different perspective, the idea is that instead of applying MOR technique onto linear problem that is the Jacobian of the HB equation; find a way to apply it directly onto the original nonlinear equation. In other words, project the large size nonlinear system equation onto a lower dimensional space, and solve the lower dimensional equation using an iterative method such as Newton iteration. The difference is then that the Krylov subspace projection basis is constructed once at the beginning of the solution, therefore, this would overcome the disadvantage of Krylov which sometimes necessitate a posteriori action in order to ensure stability and passivity. However, this method is based on mathematically sound methodology to construct the projection base, which in fact encounters a number of important limitations in practical applications.

Then, in the next chapter we will present this methodology, study the limitations and propose some solutions to overcome these limitations.

IV. References:

- [1] R. Genesio, M. Milanese, "A note on the derivation and use of reduced-order models. *IEEE Trans. Automat. Contr.*," Vol. AC-21, pp.118-122, February 1976.
- [2] W.H.A. Schilders, H.A. Van der Vorst and J. Rommes, "Model order reduction: theory, research aspects and applications," Springer, 2008.
- [3] B.C. Moore, "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction," *IEEE Transactions on Automatic Control*, Vol. AC-26, No. 1, pp. 17-32, Feb. 1981.
- [4] B.C. Moore, "Singular value analysis of linear systems, parts I, II," Dep. Elec. Eng., Univ. Toronto, Toronto, Ont., Syst. Contr. Rep. 7801 and 7802, July 1978; also in Proc. IEEE Conf. Dec. Contr., pp. 66-73.
- [5] P.T. Kabamba, "Balanced gains and their significance for L2 model reduction," *IEEE Trans. Automat. Contr.*, Vol. AC-30, No, 7, pp. 690-693, 1985.
- [6] E.A. Jonckheere, L.M. Silverman. "A new set of invariants for linear systems – Applications to reduced-order compensator design," *IEEE Trans. Automat. Contr.*, Vol. AC-28, pp. 953-964, 1983.
- [7] R. Ober, D. McFarlane, "Balanced Canonical Forms for Minimal Systems: A normalized Comprime Factor Approach," *Linear Algebra and its Applications*, pp. 23-64, 1989.
- [8] C.D. Yang, F.B. Yeh, "One-Step Extension Approach to Optimal Hankel-Norm Approximation and H_{∞} - optimization Problems," *IEEE Trans. Automat. Contr.*, Vol. AC-38, No. 6, pp.674-688, 1993.
- [9] D.F. Enns, "Model Reduction with Balanced Realizations: An Error Bound and A Frequency weighted Generalization," Proc. 23rd IEEE CDC, New York, 1984, pp. 127- 132.
- [10] L. Sirovich, "Turbulence and the dynamics of coherent structures. *Quarterly of Applied Mathematics*," Vol. XLV, pp, 561-590, 1987.
- [11] Z. Bai, "Krylov Subspace Techniques for Reduced-Order Modeling of Large-Scale Dynamical Systems," *Applied Numerical Mathematics*, Vol. 43, No. 1, pp. 9-44, October 2002.
- [12] C. Lanczos, "An Iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *J. Res. Nat. Bureau Stan.*, Vol. 45, pp. 255-282, 1950.
- [13] W.E. Arnoldi, "The principle of minimized iterations in solution of the matrix eigenvalue problem," *Quarrt., Appl. Math.*, Vol. 9, pp. 17-29, 1951.
- [14] G.A. Baker, Jr. P. Graves-Morris, "Padé Approximants, Part I: Basic Theory," 1981.
- [15] A. Odabasioglu, M. Celik, L. T. Pileggi, "PRIMA: passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. on Computer-Aided Design*, Vol. 17, No. 8, pp. 645-654, Aug. 1998.
- [16] L. W. Nagel, "SPICE2, A computer program to simulate semiconductor circuits," Tech. Rep. ERL-M520, Univ. California, Berkeley, May 1975.
- [17] W. Hong, A.C. Cangellaris, "Model-order reduction of finite-element approximations of passive electromagnetic devices including lumped electrical-circuit models," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 52, No. 9, Sept. 2004, pp. 2305-2313, 2004.

- [18] K. Krohne, R. Vahldieck, "On the application of model-order reduction in the fast and reliable optimization of microwave filters and duplexers," IEEE Transactions on Microwave Theory and Techniques, Vol. 52, No. 9, Sept. 2004. pp. 2285-2291, 2004.
- [19] Y. Che-Chia, Y. Yao-Joe "Extraction of heat transfer macromodels for MEMS devices," 12th International Conference on Transducers, Solid-State Sensors, Actuators and Microsystems, 2003, Vol. 2, 8-12 June 2003, pp. 1852-1855, 2003.
- [20] R.W. Freund, P. Feldmann, "Efficient Small-signal Circuit Analysis and Sensitivity Computations with the PVL Algorithm," IEEE/ACM International Conference on Computer-Aided Design, 1994, November 6-10, 1994, pp. 404-411.
- [21] Y. Saad, "Numerical Methods for Large Eigenvalue Problems," Manchester, UK: Manchester University Press, 1992.
- [22] P. Feldmann and R. Freund. "Efficient linear circuit analysis by Padé approximation via Lanczos process," IEEE Trans. Computer-Aided Design, 14, 137-158, 1993.
- [23] J. Chen, S.M. Kang, J. Zhou, C. Liu, and J. E. Schutt-Aine, "Reduced-Order Modeling of Nonlinear MEMS Devices With Taylor-Series Expansion and Arnoldi Approach," Journal of Microelectromechanical Systems, 2004, vol. 13, no. 3, pp. 441-451.
- [24] Y. Chen and J. White, "A quadratic method for nonlinear model order reduction," Tech. Proc. of the 2000 International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators, MSM 2000, San Diego, USA, pp. 477-480, March 2000.
- [25] M. Rewienski, J. White. "Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations". Linear algebra and its applications, 2005.
- [26] M. Rewienski, J. White . "A Trajectory Piecewise-Linear Approach to Model order Reduction and Fast Simulation of Nonlinear Circuit and Micromachined Devices". IEEE Transactions on Computer Aided design of integrated circuit and systems, Février 2003, Vol. 22, N°2.
- [27] D. Vasilyev, M. Rewienski, J. White, "A TBR-based trajectory piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and MEMS," Proc. of ACM/IEEE DAC 2003 pp. 490-495, 2003.
- [28] S. Lall, J. E. Marsden and Sonja Glavaski, "A subspace approach to balanced truncation for model reduction of nonlinear control systems," International Journal of robust and nonlinear control, 2002.
- [29] E. Gad and M. Nakhla, "A Circuit Reduction Technique for Finding the Steady-State solution of Nonlinear Circuits," IEEE Transactions on Microwave theory and techniques, Vol, 48, N0. 12, Dec 2000
- [30] P. Holmes, J.L Lumley and G. Berkooz, "Turbulence, Coherent Structures, Dynamical Systems and Symmetry," Cambridge Monogr. Mech., Cambridge University Press, 1996.
- [31] B. Salimbahrami, B; Lohmann, T. Bectold, J. G. Korvink, "Two-sided Arnoldi Algorithm and Its Application in Order Reduction of MEMS," In Proceedings 4th MATHMOD, Vienna, pp. 1021-1028, Feb. 2003.
- [32] C. Lanczos, "An Iteration method for the solution of the eigenvalue problem of linear differential and integral operators," J. Res. Nat. Bureau Stan., Vol. 45, pp. 255-282, 1950.
- [33] A. Chatterjee, "An introduction to the proper orthogonal decomposition," Special section: Computational Science, Current Science, Vol. 78, No. 7, pp. 808 - 817, 10 April, 2000.
- [34] S. Ravindram, "Proper Orthogonal Decomposition in Optimal Control of Fluids," NASA/TM-1999-209113.

- [35] R.W. Freund, “*SPRIM: Structure-Preserving Reduced-Order Interconnect Macromodeling*,” In Technical Digest of the 2004 IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD-2004), pp. 80-87, San Jose, California, 2004. IEEE Computer Society Press.
- [36] J. Chen, S-M Kdng, “*Model-Order Reduction of Nonlinear MEMs Devices through Arclength-Based Karhunen-Loeve Decomposition*,” The 2001 IEEE International Symposium on Circuits and Systems, ISCAS 2001, Vol. 3, 6-9 May 2001, pp. 457-460, 2001.
- [37] N. Vora, P. Daoutidis, “*Nonlinear Model Reduction of Reaction Systems with Multiple Time Scale Dynamics*,” Proceedings of the American Control Conference, Arlington, VA June 25-27, pp. 4752-4757, 2001
- [38] D. Kosambi, “*Statistics in function space*,” Indian Mathematical Society , pp 76–88, 1943,
- [39] M. Rewienski, “*A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems*,” Thèse du Massachussets Institute of Technology, U.S., 2003

**Chapitre IV :Harmonic Balance Reduced Order
Model Equation**

I. Introduction

The RF silicon integrated circuits (ICs) for communication systems witnesses a very rapid grow, which makes it essential to find more powerful methods of circuit simulation, especially when it comes to finding steady state solution response for large RF and microwave nonlinear circuits, which is a real challenge for designers and engineers.

Time domain methods, such as shooting method [1], as we have seen are considered as good simulation tools, however, they have limitations especially when there are widely separated time constants in the circuit (e.g. mixer analyses) and distributed elements (e.g. transmission lines).

Harmonic Balance technique (HB) as introduced in the preceding chapter is better suited to RF circuits for its ability to handle mixer and distributed elements. The main idea behind HB is that for many applications, signal waveforms can be represented in the frequency domain as a finite Fourier series with a limited number of independent tones [1],[2],[3]. This way, the differential equations that describe the circuit will be transformed into a set of nonlinear algebraic equations that can be solved directly using iterative techniques such as Newton-Raphson.

However, the main problem of NR is that it becomes computationally expensive while solving the nonlinear equations, because of the need to store and factorize the Jacobian matrix. As the size of RF circuit increases, Jacobian matrix grows and becomes time and memory consuming, limiting the method to medium sized circuits [4]

As previously indicated, about three decades ago, “Matrix-implicit linear” GMRES [5] solvers have been introduced for circuit analysis purposes. These methods are used to reduce the high cost of factoring and storing the Jacobian matrix in a way that only a few components of the Jacobian matrix need to be stored and the number of arithmetic operations minimized. These methods actually correspond to applying MOR technique to the Jacobian of the HB problem. The methods are incorporated in modern simulators and

have lately given substantial performance improvement in the simulation of large scale circuits like single chip RF transmitters [1].

The challenging point in the above methods is the necessity for a preconditioner matrix to achieve convergence of the GMRES solver in a reasonably small amount of iterations, typically less than 20. Unfortunately there is no well-known methodology for constructing such a preconditioner, given an arbitrary circuit; hence the number of iterations to convergence is often much higher, in the order of hundreds. Besides the GMRES preconditioner has to be constructed and factored every Newton iteration of HB equation. The consequence is excessive consumption of both computer memory and CPU time as the circuit size and nonlinearity grow, occasionally making the method efficiency extremely poor.

Two decades ago, Gad *et al.* [6] have proposed an alternate method that looks at the problem from a different perspective. That is instead of applying MOR technique onto linear problem that is the Jacobian of the HB equation; find a way to apply it directly onto the original nonlinear HB equation. In other words, project the large size HB equation onto a lower dimensional space, and solve the lower dimensional equation using Newton iteration. The expectation is that the lower dimensional size is small enough so that we can efficiently apply LU factor to the resulting Jacobian matrix and no more resort to GMRES solver. As we will see this approach is in fact equivalent to using the traditional approach of HB equation solution with GMRES solver onto the Jacobian problem.

The difference is then that the Krylov subspace projection basis is constructed once at the beginning of the solution and not repeated every NR iteration as is the case with classical HB- Jacobian -GMRES approach. The challenge with the new approach is then how to construct this single projection basis in a numerically effective manner. In his thesis [7], Gad has proposed a mathematically sound methodology to construct the basis, which however encounters a number of important limitations in practical applications. In this chapter we will present this methodology, study the limitations and propose some solutions to overcome these limitations.

So the first section gives an overview of the nonlinear model order reduction proposed by Gad *et al.*, in [6]. The second section introduces the concept of this method and how the system is reduced using projection technique through a congruent transformation matrix. Section III shows how the reduced system is solved

for steady state response. Section IV is intended to explain in details the construction of the congruent matrix. Section V provides numerical examples to show validity and efficiency of the proposed method. Section VI discusses the limitations of the proposed method, as an entrance to propose a solution to these limitations. Section VII proposes a numerical differentiator that can overcome to some extent the limitations discussed in section VI. Section VIII shows the advantages and disadvantages of the proposed numerical differentiator. Section IX provides numerical examples to prove correctness of numerical differentiator

I.1. General formulation of Harmonic Balance reduced order model equation

In this section, the model order reduction technique proposed by Gad in [6] will be overviewed in details. The MOR technique is based on orthonormal projection methods through congruent transformation, the basic idea behind it is the assumption that the state variables in the circuit equation are linearly dependent, and can be composed from a few number of vectors (orthogonal basis) in some parameter space, therefore if the circuit equation is projected onto this space, the resulting equation dimension can be drastically reduced (congruence transformation). In the following, we will present the method in details

I.2. General RF circuit equation

For simplicity of presentation, we will re-introduce the HB equation.

First recall the N dimensional RF circuit time-domain equilibrium equation .

$$\int_{-\infty}^t \mathbf{y}(t - \tau) \mathbf{x}(\tau) d\tau + \dot{\mathbf{q}}(\mathbf{x}(t)) + \mathbf{i}(\mathbf{x}(t)) + \mathbf{u}(t) = \mathbf{0} \quad IV-1$$

$\mathbf{x}(t) \in \mathfrak{R}^N$: circuit state variables (node voltages and non-admittance element currents)

$\mathbf{u}(t) \in \mathfrak{R}^N$: circuit excitation (independent nodal current sources and voltage sources)

$\mathbf{i}(\mathbf{x}(t)) \in \mathfrak{R}^N, \mathbf{q}(\mathbf{x}(t)) \in \mathfrak{R}^N$: nodal nonlinear currents and charges.

$\mathbf{y}(t) \in \mathfrak{R}^{N \times N}$: nodal impulse response of the linear elements

I.3. Harmonic Balance equation

As introduced previously, the corresponding harmonic balance form to the above equation is a $\mathbf{D} = \mathbf{N}(2\mathbf{H} + 1)$ dimensional equation, where \mathbf{H} is the number of harmonics or intermodulation products, as below.

$$\mathbf{F}(\mathbf{X}) = \mathbf{Y}\mathbf{X} + \mathbf{j}\Omega\Gamma\mathbf{q}(\Gamma^{-1}\mathbf{X}) + \Gamma\mathbf{i}(\Gamma^{-1}\mathbf{X}) + \mathbf{U} = \mathbf{0} \quad \text{IV- 2}$$

$$\mathbf{X} = \{\mathbf{X}_k; k = -\mathbf{H}, \dots, \mathbf{H}\} \in \mathbb{C}^{\mathbf{D}}$$

$$\mathbf{U} = \{\mathbf{U}_k; k = -\mathbf{H}, \dots, \mathbf{H}\} \in \mathbb{C}^{\mathbf{D}}$$

$$\mathbf{Y} = \text{diag}\{\mathbf{Y}(k\omega_0); k = -\mathbf{H}, \dots, \mathbf{H}\} \in \mathbb{C}^{\mathbf{D} \times \mathbf{D}}$$

$$\Omega = \text{diag}\{k\omega_0 \times \mathbf{1}; k = -\mathbf{H}, \dots, \mathbf{H}\} \in \mathbb{C}^{\mathbf{D} \times \mathbf{D}}$$

Γ : *Fourier transform operator matrix*

For notational simplicity we define the nonlinear function

$$\Phi(\mathbf{X}) = \mathbf{j}\Omega\Gamma\mathbf{q}(\Gamma^{-1}\mathbf{X}) + \Gamma\mathbf{i}(\Gamma^{-1}\mathbf{X})$$

So that the above equation rewrites:

$$\left\{ \begin{array}{l} \mathbf{F}(\mathbf{X}) = \mathbf{Y}\mathbf{X} + \Phi(\mathbf{X}) + \mathbf{U} = \mathbf{0} \\ \mathbf{F}, \mathbf{X}, \Phi, \mathbf{U} \in \mathbb{C}^{\mathbf{D}} \end{array} \right. \quad \text{IV- 3}$$

I.4. Reduced order equation

Assume we can find a good projection subspace of dimension $\hat{\mathbf{D}}$ and orthogonal basis $\mathbf{V} \in \mathbb{C}^{\mathbf{D} \times \hat{\mathbf{D}}}$, such that $\hat{\mathbf{X}} \in \mathbb{C}^{\hat{\mathbf{D}}}$ is the projection of the solution in the subspace e.g:

$$\mathbf{X} = \mathbf{V}\hat{\mathbf{X}} \quad \text{IV- 4}$$

Then inserting (IV-4) in (IV-3) and left multiplying (IV-3) with the matrix transpose \mathbf{V}^T ; we obtain the reduced order equation in the parametric space.

$$\left\{ \begin{array}{l} \hat{\mathbf{F}}(\hat{\mathbf{X}}) = \hat{\mathbf{Y}}\hat{\mathbf{X}} + \hat{\Phi}(\hat{\mathbf{X}}) + \hat{\mathbf{U}} = \mathbf{0} \\ \hat{\mathbf{F}}, \hat{\mathbf{X}}, \hat{\Phi}, \hat{\mathbf{U}} \in \mathbb{C}^{\hat{\mathbf{D}}} \end{array} \right. \quad \text{IV- 5}$$

Where,

$$\hat{F}(\hat{X}) = V^T F(V\hat{X})$$

$$\hat{Y} = V^T Y V$$

$$\hat{\Phi}(\hat{X}) = V^T \Phi(V\hat{X})$$

$$\hat{U} = V^T U$$

The reduced dimension of the system is expected to be sufficiently small, $\hat{D} \ll D$, in the order of tens and hundred, so that we can efficiently perform LU factorization of the Jacobian and gain important computer memory and CPU savings. However, the challenge is how do we form the projection basis $V \in \mathbb{C}^{D \times \hat{D}}$, given an arbitrary circuit topology and equations. The approach suggested by E. Gad et al. is Padé approximation, which we will consider below.

I.5. Padé approximation approach to HB equation model order reduction

The idea of Padé approximation to HB equation model order reduction can be summarized as follows:

I.5.1. Parameter space definition

In order to apply Padé approximation theory, it is necessary to identify a convenient parametric observation space, on which we can carry a power expansion of the response. In this purpose, starting from the HB equation (IV-2), let us define an arbitrary real parameter $\alpha \in [0, 1]$, and make the excitation vector become a function of the parameter $\alpha: U \rightarrow U(\alpha)$

By doing so, we transform the harmonic balance equation (IV-2) into a parameterized equation as follows:

$$F(X(\alpha)) = YX(\alpha) + \Phi(X(\alpha)) + U(\alpha) = \mathbf{0} \quad \text{IV- 6}$$

I.5.2. . HB Equation power series expansion

Function $\mathbf{U}(\boldsymbol{\alpha})$ is arbitrary; however as we will see, a trivial and practical case is to consider a linear equation. It is obvious that, if excitation $\mathbf{U}(\boldsymbol{\alpha})$ is considered a power series of $\boldsymbol{\alpha}$, then for a stable circuit (i.e., non-autonomous), we may expect solution $\mathbf{X}(\boldsymbol{\alpha})$ to be also a power series of $\boldsymbol{\alpha}$. The same is true for the nonlinearity vector $\boldsymbol{\Phi}$ and other components of eq. (IV-6). Hence we may expand all state vectors in the form of a power series:

$$\mathbf{U}(\boldsymbol{\alpha}) = \sum_{p=0}^{\hat{D}-1} \bar{\mathbf{U}}_p \boldsymbol{\alpha}^p,$$

$$\mathbf{X}(\boldsymbol{\alpha}) = \sum_{p=0}^{\hat{D}-1} \bar{\mathbf{X}}_p \boldsymbol{\alpha}^p,$$

$$\boldsymbol{\Phi}(\mathbf{X}(\boldsymbol{\alpha})) = \sum_{p=0}^{\hat{D}-1} \bar{\boldsymbol{\Phi}}_p \boldsymbol{\alpha}^p \quad \text{IV- 7}$$

Similarly the Jacobian of the parameterized equation expands as a power series as well:

$$\mathbf{J}(\boldsymbol{\alpha}) = \frac{\partial \boldsymbol{\Phi}(\mathbf{X}(\boldsymbol{\alpha}))}{\partial \mathbf{X}(\boldsymbol{\alpha})} = \sum_{p=0}^{\hat{D}-1} \bar{\mathbf{J}}_p \boldsymbol{\alpha}^p \quad \text{IV- 8}$$

I.5.3. . Notational conventions

It is important to clarify the conventions used in the above equation, which can be a little ambiguous. The indexed variable with a bar, $\bar{\mathbf{X}}_p \in \mathbb{C}^D$ indicates the coefficient of the power series decomposition, not to confuse with Fourier series decomposition coefficients $\mathbf{X}_p \in \mathbb{C}^N$ of the variable $\mathbf{x}(t) \in \mathfrak{R}^N$, $D = N(2H + 1)$

Function $\mathbf{X}(\boldsymbol{\alpha})$ designates the parameterized equation solution, not to confuse with the scalar \mathbf{X} , solution of the original system.

I.5.4. Introducing the continuant α

Introducing a parameter and expanding the excitation as a power series of the parameter provides a natural path to Padé approximation theory. If for some parameter value $\alpha = \underline{\alpha}$ we have the identity of parameterized excitation and original circuit excitation, i.e: $\mathbf{U}(\underline{\alpha}) = \mathbf{U}$, then parameterized solution is identically the original system solution, i.e., $\mathbf{X}(\underline{\alpha}) = \mathbf{X}$.

Hence per virtue of the power expansion above, we have

$$\mathbf{X} = \sum_{p=0}^{\hat{D}-1} \bar{\mathbf{X}}_p \underline{\alpha}^p \quad \text{IV- 9}$$

We then see that the desired solution \mathbf{X} is a vector of the subspace spanned by the coefficients of the power expansion (known as moments of Padé approximation) of the parameterized problem solution : $\text{colspan}\{\bar{\mathbf{X}}_0, \bar{\mathbf{X}}_1, \dots, \bar{\mathbf{X}}_{D-1}\}$

Provided that \hat{D} , the order of power decomposition is much less than D , we have thus defined a convenient projection space and derived a procedure how to construct a congruence basis , i.e.

$$\mathbf{V} = \text{colspan}\{\bar{\mathbf{X}}_0, \bar{\mathbf{X}}_1, \dots, \bar{\mathbf{X}}_{D-1}\} \quad \text{IV- 10}$$

Note however that since the column vectors of \mathbf{V} are coefficients of a plain power series expansion, they tend to be linearly dependent, so that \mathbf{V} is not naturally an orthogonal basis. Once the basis is obtained, orthogonalization can be carried out a-posteriori, using e.g., modified Gram-Schmidt, QR or SVD methods. Now it remains how to select the convenient excitation parameterization function $\mathbf{U}(\alpha)$ and derive the identification algorithm for moments of Padé approximation composing the basis \mathbf{V} .

I.6. Identification of moments of Padé (MP) approximation

I.6.1. . HB Equation parameterization

So far, the excitation parameterization function $\mathbf{U}(\alpha)$ we have defined is an arbitrary polynomial. A good and practical selection is to consider a first order polynomial, i.e. $\mathbf{U}(\alpha) = \bar{\mathbf{U}}_0 + \bar{\mathbf{U}}_1 \alpha$.

Hence let us set:

$$\mathbf{U}(\alpha) = \bar{\mathbf{U}}_0 + \bar{\mathbf{U}}_1 \alpha \tag{IV- 11}$$

This excitation configuration is sketched in the figure below (Fig IV-1) for a simple diode circuit. Indeed, this parameterization selection corresponds to the classical continuation formula (known as source stepping) commonly adopted to help convergence of Newton iteration in HB simulators. The parameter α thus corresponds to the normalized input power, which we sweep from 0 to the final value, 1. Every harmonic component \mathbf{X}_k of the solution is then a smooth function of the parameter α : $\mathbf{X}_k(\alpha)$

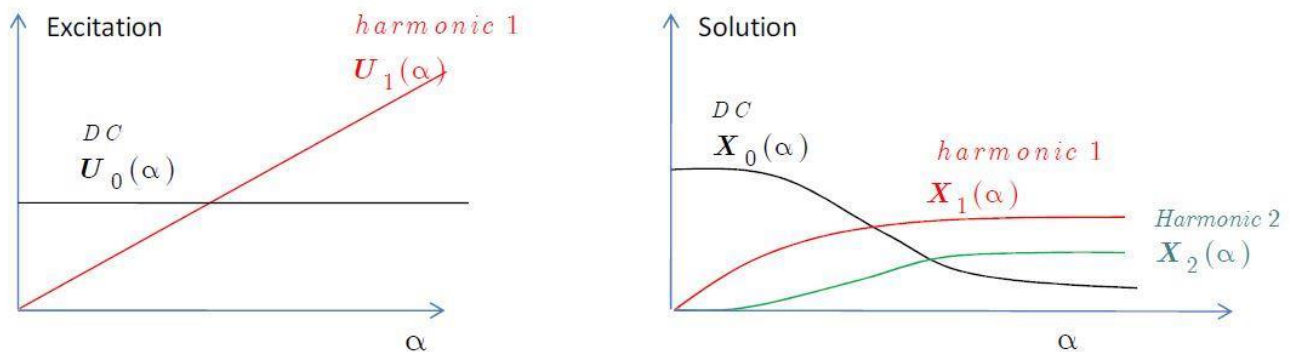
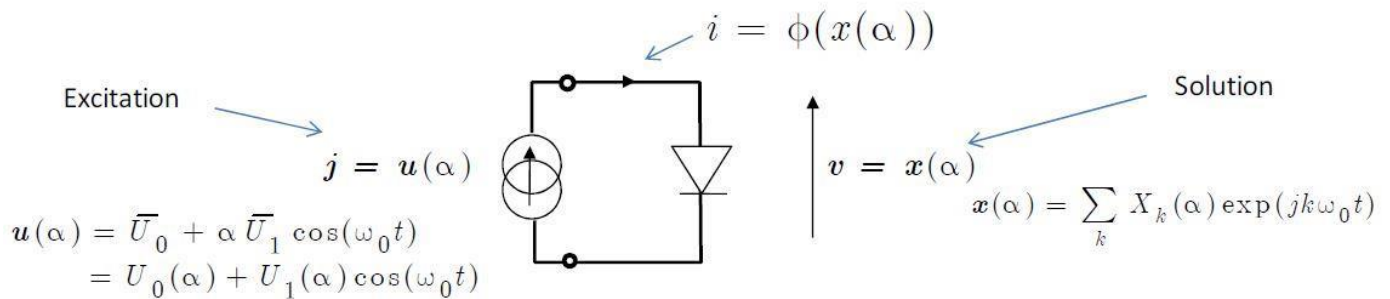


Figure IV - 1 : Illustration of HB equation parameterization

It is important to recall that the power coefficients $\bar{\mathbf{U}}_p$ are vectors of dimension $\mathbf{D} = \mathbf{N}(2\mathbf{H} + 1)$ and notice that the content of the two non-zero coefficients in eq. (IV-11) is as below:

$\bar{\mathbf{U}}_0 \equiv [\mathbf{0}, \dots, \mathbf{U}_0, \dots, \mathbf{0}]^T$: Zero order power coefficient is the DC excitation

$\bar{\mathbf{U}}_1 \equiv [\mathbf{0}, \dots, \mathbf{U}_{-1}, \mathbf{0}, \mathbf{U}_1, \dots, \mathbf{0}]^T$: First order coefficient is the Fundamental harmonic of excitation

$\bar{\mathbf{U}}_p \equiv \bar{\mathbf{0}}, p = 2, \dots, \hat{\mathbf{D}} - 1$: Higher order power coefficients are zero

I.6.2. Identification equation of moments of Padé

From the above parameterization strategy, we observe the following:

- when $\alpha = 0$, the parameterized problem corresponds to the simulation of DC conditions in the original problem
- when $\alpha = 1$, the parameterized problem corresponds to the simulation of the original problem
- Now inserting power expansion of all state variables, eq. (IV-7), into the reduced order equilibrium equation (IV-6), this finally expands as below:

$$\sum_{p=0}^{\hat{D}-1} (Y\bar{X}_p + \bar{\Phi}_p + \bar{U}_p)\alpha^p = \mathbf{0} \quad IV-12$$

- In order to satisfy eq. (IV-12) for any value of α , it is necessary that all power coefficients of the series be zero, i.e:

$$\left\{ \begin{array}{l} Y\bar{X}_0 + \bar{\Phi}_0 + \bar{U}_0 = \mathbf{0} \\ Y\bar{X}_1 + \bar{\Phi}_1 + \bar{U}_1 = \mathbf{0} \\ Y\bar{X}_p + \bar{\Phi}_p + \bar{U}_p = \mathbf{0}, p = 2, \dots, \hat{D} - 1 \end{array} \right. \quad IV-13$$

Equation (IV-13) is a set of algebraic equations having power coefficients $\bar{X}_0, \bar{X}_1, \dots, \bar{X}_{\hat{D}}$ as unknowns. We therefore have succeeded to obtain the equation to be solved in order to determine the moments of Padé approximation spanning the reduced order subspace.

I.7. Iterative solution of the MP identification equation

In order to solve the above Padé moments identification equation we resort to an interesting property of power expansion, that is below.

I.7.1. Property

If \mathbf{X} expands has a polynomial of α : $\mathbf{X} = \sum_p \bar{X}_p \alpha^p$, and

$$\Phi$$
 is a function of \mathbf{X} : $\Phi = \Psi(\mathbf{X}) = \sum_p \bar{\Phi}_p \alpha^p$

a coefficient of order p of the polynomial expansion of $\Phi(\mathbf{X})$ depend only on coefficients of order less than or equal to p of the variable \mathbf{X} ; i.e.: $\bar{\Phi}_p = f(\bar{X}_0, \bar{X}_1, \dots, \bar{X}_p)$. In other terms $\bar{\Phi}_p$ is independent of, $\bar{X}_i, i > p$.

I.7.2. Recursive algorithm

Thanks to the above property, we can perform the identification of the power coefficients on a one by one basis, starting with $p = 0$, and progressing to $p = \hat{D}$, i.e., a recursive procedure. This is briefly described in the following.

- DC state: $p = 0$

$$Y\bar{X}_0 + \bar{\Phi}_0 + \bar{U}_0 = 0$$

IV- 14

Where,

$$\bar{U}_0 \equiv [\mathbf{0}, \dots, U_0, \dots, \mathbf{0}]^T$$

$$\bar{X}_0 \equiv [\mathbf{0}, \dots, X_0, \dots, \mathbf{0}]^T$$

$$\bar{\Phi}_0 \equiv [\mathbf{0}, \dots, \Phi(X_0), \dots, \mathbf{0}]^T$$

We observe the following important features that are the main figures of the identification algorithm:

- ♦ solution \bar{X}_0 corresponds the DC solution of the circuit
- ♦ $\bar{J}_0 = \partial \bar{\Phi}_0 / \partial \bar{X}_0$ is the small signal conductance of the nonlinearities
- ♦ $J_{AC} = (Y + \bar{J}_0) \in \mathbb{C}^{D \times D}$, the jacobian of the above system, corresponds to the small signal admittance of the circuit. J_{AC} is a sparse and block diagonal matrix as sketched below.

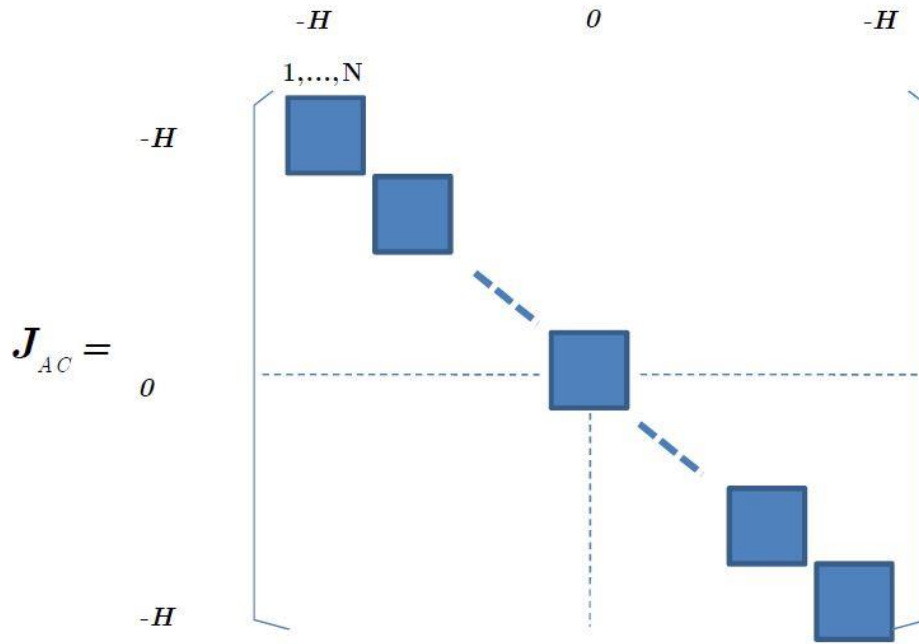


Figure IV - 2 : Illustration of the Jacobian matrix

- **First order coefficient: $p = 1$**

$$Y\bar{X}_1 + \bar{\Phi}_1 + \bar{U}_1 = \mathbf{0} \quad IV - 15$$

Where,

$$\bar{U}_1 \equiv [\mathbf{0}, \dots, U_{-1}, \mathbf{0}, U_1, \dots, \mathbf{0}]^T$$

Resorting to polynomial algebra from (IV-7) and (IV-8), we easily derive the derivatives

$$\Phi'(\bar{X}(\alpha)) = \frac{\partial \Phi(\bar{X}(\alpha))}{\partial \bar{X}(\alpha)} \bar{X}'(\alpha) = J(\alpha) \bar{X}'(\alpha)$$

$$\bar{\Phi}_1 + \sum_{p=2}^{\hat{D}-1} p \bar{\Phi}_p \alpha^{p-1} = \left(\bar{J}_0 + \sum_{p=1}^{\hat{D}-1} \bar{J}_p \alpha^p \right) \left(\bar{X}_1 + \sum_{p=2}^{\hat{D}-1} p \bar{x}_p \alpha^{p-1} \right)$$

So that setting $\alpha = \mathbf{0}$ both sides in the above, we isolate the expression, $\bar{\Phi}_1 = \bar{J}_0 \bar{X}_1$, which inserted in (IV-15) yields a linear equation below to be solved for \bar{X}_1 . We observe that the coefficient matrix in the linear equation is simply the small signal admittance matrix obtain from DC analysis.

$$(Y + \bar{J}_0) \bar{X}_1 = -U_1 \quad IV - 16$$

- **2nd order coefficient: $p = 2$**

$$Y\bar{X}_2 + \bar{\Phi}_2 + \bar{U}_2 = \mathbf{0} \quad IV - 17$$

Where,

$$\bar{U}_2 \equiv [\mathbf{0}]$$

Repeating the above polynomial algebra on second order derivative $\Phi''(\bar{X}(\alpha))$, we obtain the expression, $\bar{\Phi}_2 = \bar{J}_0\bar{X}_2 + \frac{1}{2}\bar{J}_1\bar{X}_1$, which inserted in (IV-15) yields again a linear system to be solved for \bar{X}_2 :

$$(Y + \bar{J}_0)\bar{X}_2 = -\frac{1}{2}\bar{J}_1\bar{X}_1 \quad IV - 18$$

- **Higher order coefficients: $p \geq 2$**

Repeating the process for higher order coefficients, we finally obtain the recurrence below, constituting the solution algorithm for all Padé moments of the HB equation.

$$Y\bar{X}_0 + \Phi(\bar{X}_0) + \bar{U}_0 = \mathbf{0} \quad IV - 19$$

$$(Y + \bar{J}_0)\bar{X}_1 = -U_1$$

$$(Y + \bar{J}_0)\bar{X}_p = -\frac{1}{p} \sum_{i=1}^{p-1} (p-1)\bar{J}_i \bar{X}_{p-i}, \quad p = 1, \dots, \hat{D} - 1$$

1.7.3. Computation complexity of the recursive algorithm

There are two major operations at each iteration p of the above algorithm, the solution the linear equation $(Y + \bar{J}_0)\bar{X}_p = \mathbf{b}_p$, and the computation of the nonlinear device Jacobian coefficient \bar{J}_{p-1} .

As shown above, $(Y + \bar{J}_0)$ the small signal admittance matrix is sparse and block diagonal ; besides since the matrix is unchanged from iteration to iteration, it is therefore LU factored once and reused for all iterations which constructing the matrix V . The cost of linear equation solution is therefore negligible.

The critical point remains therefore the computation of the nonlinear device Jacobian coefficient \bar{J}_{p-1} which we address in the next section.

I.8. Polynomial factorization of the nonlinear device equation

Computation of the power expansion coefficients of the nonlinear devices Jacobian, $\bar{J}_p, p = 1, \dots, \hat{D} - 1$ is the key and most critical point of the reduced order HB equation process.

Recall from (IV-8) that:

$$J(\alpha) = \frac{\partial \Phi(X(\alpha))}{\partial X(\alpha)} = \sum_{p=0}^{\hat{D}-1} \bar{J}_p \alpha^p \quad IV-20$$

We showed above, that \bar{J}_0 is the Jacobian of the DC analysis; we need to develop a procedure to derive the higher order terms. However, If the symbolic expression of the nonlinearity (X) is known, it is fortunately possible to derive the power expansion coefficients \bar{J}_p one at a time, starting from \bar{J}_0 , within the process described in the recursive algorithm derived above.

For the sake of derivation simplicity, we consider that nonlinear equation $\Phi(X)$ contains only nonlinear currents (extension to nonlinear charges and fluxes is straightforward).

Hence, referring to definition (IV-2), $(X) = \Gamma i(\Gamma^{-1}X)$, and inserting in the above, we have:

$$J(\alpha) = \sum_{p=0}^{\hat{D}-1} \bar{J}_p \alpha^p = \frac{\partial \Phi(X(\alpha))}{\partial X(\alpha)} = \Gamma \frac{\partial i(x(\alpha))}{\partial x(\alpha)} \Gamma^{-1} \quad IV-21$$

The nonlinear time domain Jacobian $\frac{\partial i(x(\alpha))}{\partial x(\alpha)}$ is a diagonal matrix of size $S \times N$, where S and N are the number of time samples and circuit nodes, respectively. Let us note it:

$$g(x) \triangleq \frac{\partial i(x)}{\partial x} = \text{diag} \left\{ \frac{\partial i(x_0)}{\partial x_0}, \dots, \frac{\partial i(x_{S-1})}{\partial x_{S-1}} \right\} \quad IV-22$$

According to the considerations above, both the time-domain state variables $x(\alpha)$ and Jacobian $g(x(\alpha))$ expands as a polynomial of the parameter α , i.e:

$$x(\alpha) = \sum_{p=0}^{\hat{D}-1} \bar{x}_p \alpha^p \quad IV-23$$

$$g(x(\alpha)) = \sum_{p=0}^{\hat{D}-1} \bar{g}_p \alpha^p$$

Finally a comparison of (eq-22) with (eq-21) and (eq-23) we find the important Fourier time to frequency domain map of the state variable and Jacobian.

$$\bar{x}_p = \Gamma^{-1} \bar{X}_p \quad IV-24$$

$$\bar{J}_p = \Gamma \bar{g}_p \Gamma^{-1}$$

Now we are in a position to establish the full algorithm for the identification of moments of Padé; it follows directly from the identification equation (eq-19)

I.9. Moments of Padé Identification algorithm

- Iteration: $p = 0$

1. Solve regular DC equation, i.e., the N dimensional equation: $Y \bar{X}_0 + \bar{\Phi}_0 + \bar{U}_0 = \mathbf{0}$
2. Compose the zero order polynomial coefficient of state variable: $\bar{X}_0 \equiv [\mathbf{0}, \dots, X_0, \dots \mathbf{0}]^T$
3. Map the state variable coefficient into time domain: $\bar{x}_0 = \Gamma^{-1} \bar{X}_0$
4. Compute time domain zero order coefficient of the Jacobian: $\bar{g}_0 = \partial i(\bar{x}_0) / \partial \bar{x}_0$
5. Map time domain Jacobian coefficient to frequency domain: $\bar{J}_0 = \Gamma \bar{g}_0 \Gamma^{-1}$
6. Form and LU factor the small signal admittance matrix: $(Y + \bar{J}_0) = (Y + \Gamma \bar{g}_0 \Gamma^{-1})$

- Iteration: $p = 1$

1. Solve 2nd equation of (eq-19) and get first order coefficient of state variable \bar{X}_1 :

$$(Y + \bar{J}_0) \bar{X}_0 = -U_1$$
2. Map the state variable coefficient into time domain: $\bar{x}_1 = \Gamma^{-1} \bar{X}_1$
3. Apply the formal vector of power series $x(\alpha) = X_0 + X_1 \alpha$ to the nonlinear equation (eq-22) and identify the Jacobian first order time-domain coefficient \bar{g}_1 as in (IV-23)
4. Map time domain Jacobian coefficient to frequency domain: $\bar{J}_1 = \Gamma \bar{g}_1 \Gamma^{-1}$

- Higher order iterations: $p \geq 2$

1. Solve 3rd equation of (eq-19) and get p^{th} order coefficient of state variable \bar{X}_p

$$(Y + \bar{J}_0)\bar{X}_p = -\frac{1}{p} \sum_{i=1}^{p-1} (p-1)\bar{J}_i \bar{X}_{p-i}$$

2. Map the state variable coefficient into time domain: $\bar{x}_p = \Gamma^{-1}\bar{X}_p$
3. Apply the formal vector of power series $x(\alpha) = \sum_{i=0}^p \bar{x}_i \alpha^i$ to the nonlinear equation (eq-22) and identify the Jacobian p th order time-domain coefficient \bar{g}_p as in (eq-23)
4. Map time domain Jacobian coefficient to frequency domain: $\bar{J}_p = \Gamma \bar{g}_p \Gamma^{-1}$
5. $p \leftarrow p + 1$
6. loop until $p = \hat{D} - 1$

I.10. Formal factorization of the nonlinear device equation

So far we have made important progress in establishing the algorithm for MP identification; however we haven't still established the process for identifying the time-domain polynomial coefficient \bar{g}_p of the Jacobian in (eq-23), within step 3) in the above algorithm. The ultimate complexity in the algorithm resides in the identification of these coefficients. This is fortunately possible in a formal way if all device equations can be expressed with continuous transcendental functions.

For illustration, consider a simple diode circuit equation as sketched in Figure below (Fig IV-3). It is driven by a parameterized excitation $\mathbf{u}(\alpha, t)$ so that the resulting diode voltage $x(\alpha, t)$, diode current $\Phi(\alpha, t)$ and conductance $\mathbf{g}(\alpha, t) = \frac{d\Phi(x(\alpha, t))}{dx(\alpha, t)}$ are also a function of the parameter α .

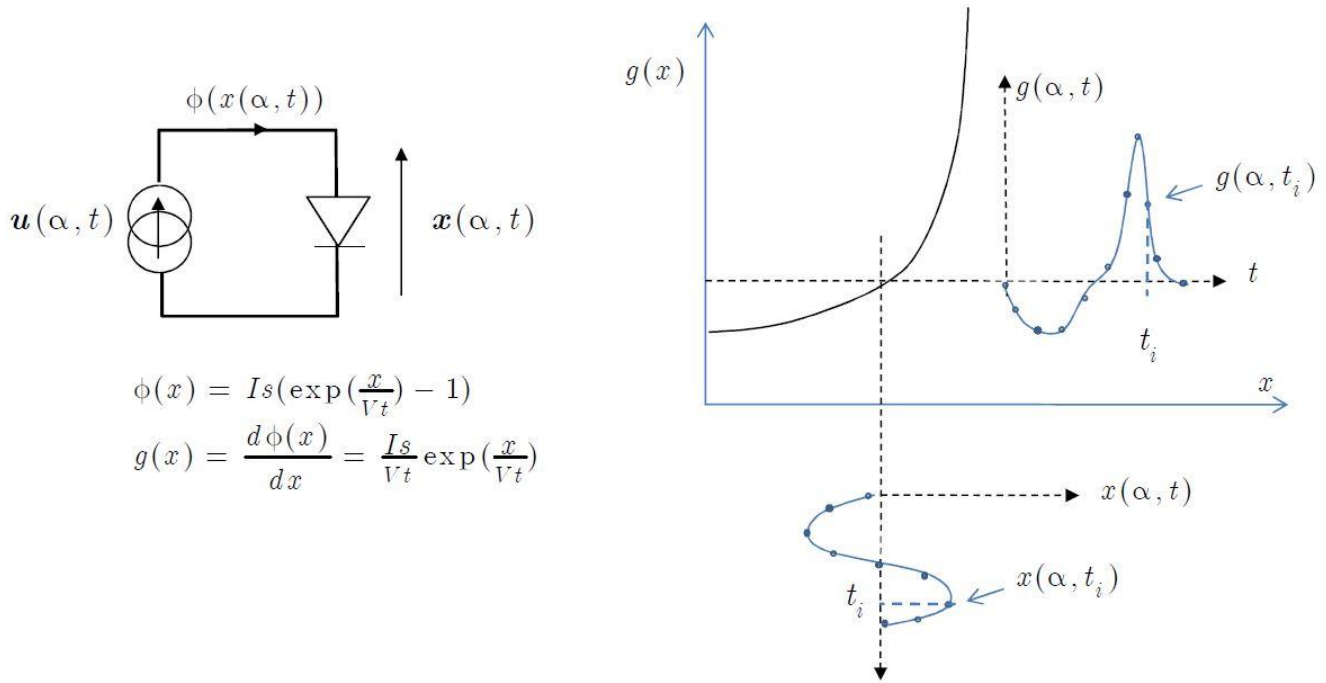


Figure IV - 3 : Illustration of a simple diode circuit

- Consider s time samples of the voltage and consequently conductance

$$\mathbf{x}(\alpha) = [x(\alpha, t_0), x(\alpha, t_1), \dots, x(\alpha, t_{s-1})]^T$$

$$\mathbf{g}(\alpha) = [g(\alpha, t_0), g(\alpha, t_1), \dots, g(\alpha, t_{s-1})]^T$$

- Consider power expansion of each time sample of voltage and consequently conductance

$$x(\alpha, t_i) = \sum_p \bar{x}_p(t_i) \alpha^p \Rightarrow g(\alpha, t_i) = \sum_p \bar{g}_p(t_i) \alpha^p, \quad i = 0, 1, \dots, S - 1$$

- Gathering all time samples $\bar{g}_p(t_i)$ of power coefficients of the same order for the conductance, we obtain the vector coefficient $\bar{\mathbf{g}}_p$ in (eq-23)

$$\bar{\mathbf{g}}_p = [\bar{g}_p(t_0), \bar{g}_p(t_1), \dots, \bar{g}_p(t_{s-1})]^T, \quad p = 0, \dots, \hat{D} - 1$$

- Now the question is, given a power expansion of a time sample of voltage $x(\alpha, t_i) = \sum_p \bar{x}_p(t_i) \alpha^p$, how do one identifies the power expansion coefficients of conductance $\bar{\mathbf{g}}_p(t_i)$
- Considering the property of power expansion already stated, we know that a coefficient of order p in the response depends only on coefficients of lower or equal order in the input signal. In other

terms, $\bar{g}_p(t_i)$ is only dependent on $\bar{x}_0(t_i), \bar{x}_1(t_i), \dots, \bar{x}_p(t_i)$ and not on the higher orders $\bar{x}_{p+1}(t_i), \bar{x}_{p+1}(t_i), \dots$

- Consequence of the above is that in the particular case where the device equations are relatively simple and expressed in terms transcendental functions, it is possible to derive a concise recursive formula of the conductance power coefficients as a function of voltage coefficients. For example in the common case of diode exponential current equation $\Phi(x) = I_s \left(\exp\left(\frac{x}{V_t}\right) - 1 \right)$, it can be shown that [8] the coefficients of the conductance $g(x) = \frac{d\Phi(x)}{dx} = \frac{I_s}{V_t} \exp\left(\frac{x}{V_t}\right)$ are given by the relation:

$$\begin{cases} \bar{g}_0(t_i) = \frac{I_s}{V_t} \exp\left(\frac{\bar{x}_0(t_i)}{V_t}\right) \\ \bar{g}_p(t_i) = \left(\frac{I_s}{V_t}\right) \frac{1}{p} \sum_{j=0}^{p-1} \bar{x}_{p-j}(t_i) \bar{g}_j(t_i) \quad , \quad p = 1, 2, \dots \end{cases} \quad IV-25$$

An important contribution, of E. Gad and R. Griffith [8] is to have considered the formal factorization of the commonly used transcendental functions in compact device modeling, and derived the table of polynomial decomposition factors given below. Note that the dependence on time of the factors has been removed for easier readability.


<p><i>Conductance equation</i></p>	<p><i>Factors :</i></p> <div style="text-align: center;"> $x = \sum_i x_i \alpha^i$ $y = \sum_i y_i \alpha^i$  $g = \sum_i \bar{g}_i \alpha^i$ </div>
<p>$g = \exp(x)$</p>	$\bar{g}_0 = \exp(\bar{x}_0)$ $\bar{g}_p = \left(\frac{1}{p}\right) \sum_{j=0}^{p-1} (p-j) \bar{x}_{p-j} \bar{g}_j$
<p>$g = \log(x)$</p>	$\bar{g}_0 = \log(\bar{x}_0)$ $\bar{g}_p = \left(\frac{1}{\bar{x}_0}\right) \left(\bar{x}_p - \left(\frac{1}{p}\right) \sum_{j=1}^{p-1} (p-j) \bar{x}_j \bar{g}_{p-j} \right)$
<p>$g = x^n$</p>	$\bar{g}_0 = \bar{x}_0^n$ $\bar{g}_p = \left(\frac{1}{p \bar{x}_0}\right) \left(n \cdot p \cdot \bar{g}_0 \bar{x}_p - \sum_{j=1}^{p-1} (pn - j(n+1)) \bar{x}_{p-j} \bar{g}_j \right)$
<p>$g = x + y$</p>	$\bar{g}_p = \bar{x}_p + \bar{y}_p$
<p>$g = x - y$</p>	$\bar{g}_p = \bar{x}_p - \bar{y}_p$
<p>$g = x \times y$</p>	$\bar{g}_p = \sum_{j=0}^p \bar{x}_j \bar{y}_{p-j}$
<p>$g = \frac{x}{y}$</p>	$\bar{g}_p = \left(\frac{1}{\bar{y}_0}\right) \left(\bar{x}_p - \sum_{j=0}^{p-1} \bar{x}_j \bar{y}_{p-j} \right)$

Table IV - 1: Recursive formulae for the derivatives of simple functions

II. Solving for Steady state solution using MOR

So far, MOR algorithm has been reviewed, and the process of constructing the matrix V has been shown in details. Once the matrix V is obtained, HB equation then could be solved using NR iterative technique as follows:

- Step 1:
Reduce the system as (eq -5)
- Step 2:
Construct the Jacobian matrix, then Reduce it as follows:
$$\hat{J} = V^T J V$$
- Step 3
Apply NR for the reduced system
$$\hat{X}_{n+1} = \hat{X}_n - \hat{J}^{-1}(\hat{X}_n) \hat{F}(\hat{X}_n)$$

Bring back the solution of NR iteration to the original system.
$$X \rightarrow V \hat{X}$$

Repeat NR till a solution is obtained.
- Step 4:
Once the final answer is obtained, the residual error should be calculated, if the residual error is greater than a pre-specified tolerance, the order of reduction is then increased. Then, the system is solved again starting at step 1.

It's good to mention that it's preferable to use the continuation method while solving for steady state solution for the same reasons as explained in chapter II, however, this is not mandatory, and it was removed in this section in order to eliminate any confusion due to the use of continuation method in the process of creating the projection base matrix V .

II.1.1. Computation Complexity

Two sources of computation complexity can be studied for the MOR technique, namely, the evaluation of moments, and construction of reduced Jacobian matrix and solving reduced system for steady state solution.

Constructing the moments of MOR technique requires only one LU decomposition, equation (IV-19) of the original size Jacobian matrix, and then each derivative is obtained by one forward/backward substitution. The computation complexity of LU is $\mathcal{O}(\widehat{D}N_h^3)$, where \widehat{D} is the number of orders, and $N_h = N(2H + 1)$, however, since \widehat{D} is too small in comparison to N_h , then complexity can be considered $\mathcal{O}(N^3)$

The second source of computation complexity of MOR technique comes from solving the reduced system to find the steady state solution, and constructing the reduced Jacobian for every iteration of Newton-Raphson recurrence.

Solving the reduced system requires computing the LU decompositions of the reduced Jacobian matrix, in particular, the nonlinear part of the reduced Jacobian matrix. The procedure of constructing the reduced Jacobian matrix for every NR iteration can be characterized as follows:

For every NR iteration, the solution must be brought back to the original size $X_{new} = V \times \widehat{X}_{old}$

Then the Jacobian should be constructed in the time domain $J = \left. \frac{\partial F(X)}{\partial X} \right|_{V \times \widehat{X}_{old}}$

Jacobian is then transformed to frequency domain by using FFT

Finally, J is reduced by matrix multiplication $\widehat{J} = V^T \times J \times V$

The computational cost of this procedure would require $\mathcal{O}(N_h S \log_2 N_h)$

However, this computation expense could be considered insignificant in comparison to the expense of computing the LU decomposition of the Jacobian matrix of the original system for every iteration of NR, especially that the size of the reduced Jacobian is much smaller than the size of the original Jacobian therefore, the factorization of reduced Jacobian is much less expensive. This difference in expense is an

advantage point of the MOR approach over the original HB, because it saves CPU time by reducing the time required to find the steady state solution. Also, in comparison to iterative methods such as GMRES, which in fact has a computational cost equivalent to $SN_h \text{Log} N_h$ [9], the MOR algorithm is almost similar in terms of computational cost.

As a conclusion, the computational cost of MOR technique is considered to save computational time of the CPU, because it solves the system of nonlinear equations in the reduced form, where the size of the reduced system is much smaller than the original one. Even though, constructing the congruent transformation matrix is considered to be expensive, however, in comparison with the number of LU decomposition of the original Jacobian matrix, then MOR technique has a great advantage over the original Harmonic Balance.

Also, in comparison to GMRES, MOR technique showed robustness in terms of constructing the reduction matrix, due to the fact that MOR technique requires the construction of the congruent transformation matrix only once, whereas, GMRES has to find an appropriate preconditioner during every iteration of NR.

The next section will provide some numerical examples to ensure that MOR HB has an advantage over regular HB in terms of saving CPU time, also, to show the correctness and efficiency of this method.

II.2. MOR HB implementation

II.2.1. Introduction

This section presents the development of the MOR Harmonic Balance simulator that was implemented using MATLAB for the simulation of RF circuits in the context of traditional SPICE circuit netlists [10] in order to provide a stable environment to study in depth the Model Order Reduction Technique [6]

We first present the motivation and strategy for the development of the simulator, then we show the performance of the simulation tool compared to a well-known circuit simulator “ADS” We show that simulation results are almost indistinguishable in terms of accuracy when the mathematical models chosen

to represent the physical components in both tools are the same. However, it is understood that the computational speed of a simulation platform developed on top of a MATLAB framework is not comparable in any manner to an optimized commercial tool.

II.2.2. Motivation

The idea of implementing a harmonic balance simulator using MATLAB stems from the need to simulate MOR method in a flexible and easy to use environment.

Some existing simulators already have HB implemented, it would have been easier to test MOR if it can be easily integrated into an environment where standard circuit netlists can be parsed and simulation results can be compared to existing tools. However, integrating MOR into an open source version of harmonic balance simulators (e.g. Qucs, harmbal and others) is an error prone task. On the other hand, MATLAB is a powerful and flexible programming environment with a handful of integrated mathematic libraries for equation solutions and data modeling. In particular MATLAB is constructed in a way that makes data structures look a lot easier than other lower level languages, also manipulating matrices is more visible and user friendly.

Out of these two aspects, the ease of use of MATLAB, and the lack of ability of connecting MOR to an already existing HB simulator, the implementing of HB simulator has been accomplished

In the following sections we describe in details the simulation strategy used in the development of this program. Also we present several tests where we compare the performance of this program to ADS for validity of our work.

The construction of HB simulator follows a simulation strategy that consists of several steps that should be done in order.

Circuits in HB simulator are introduced in a netlist form, a program that uses modified nodal analysis method (MNA) is to transform the circuits of SPICE netlist form into MATLAB data structures [11]. A DC

analysis is to be done and presented as an initial guess to HB, in which HB implements both continuation method and Newton-Raphson that come together to achieve the steady-state solution of the circuit if there is any. Newton-Raphson is applied on harmonic balance equation, in parallel to continuation method which helps to solve the problem of convergence faced by Newton-Raphson recurrence.

Once the solution of the original Harmonic balance is achieved, the simulator will save the solution, and start to perform MOR technique. The equations of MOR technique are taken from [6]

II.3. Numerical examples:

So far in this chapter, the model order reduction technique proposed in [6] was discussed, and the construction of congruent matrix was explained in details. This section is intended to show the effectiveness of proposed method through a comparison of the ordinary harmonic balance method versus the reduced one, in terms of rapidity, precision, and size of the circuit.

II.3.1. Example one: Half-wave rectifier circuit

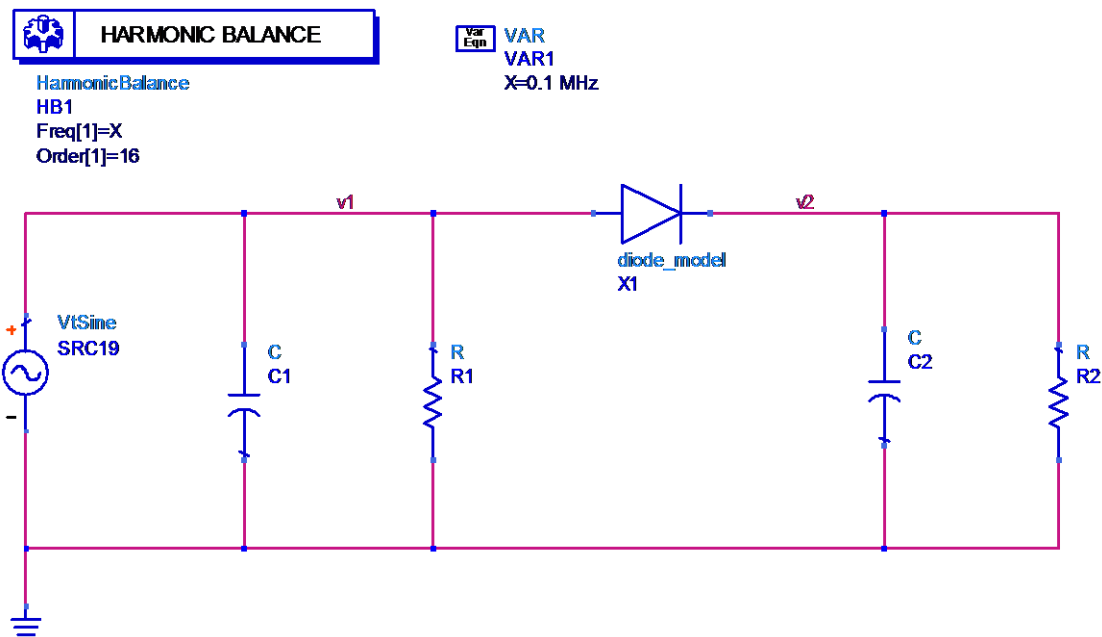


Figure IV - 4 : Rectifier circuit

The circuit shown in the figure above (Fig IV-4) is a rectifier circuit that contains a single diode, and it is meant to demonstrate the validity of the proposed algorithm in [6].

This rectifier takes a sinusoidal input $v(t) = 0.6 + 0.15\cos(\omega_0 t)$, where $f_0 = 0.1 \text{ MHz}$. It was first simulated using the HB technique, the HB approach required $K = 16$ harmonics to approximate the periodic steady-state response. The size of the original system is $n(2K + 1) = 3 \times (2 \times 16 + 1) = 99$ equations and 99 unknowns, where the reduced system required only 16 equations to reach the same steady state solution accuracy. There is no criteria to define \hat{D} , it is chosen by trial and error, starting from a very small \hat{D} , then steady-state solution analysis starts, if the error is not small enough, then the order is increased, and new steady-state solution is obtained.

Fig IV- 5 shows the output voltage waveform $v_{out}(t)$ using both HB approach and reduced order approach explained in this chapter. The maximum relative error between the waveforms obtained from the two methods is $3.1295\text{e-}006$.

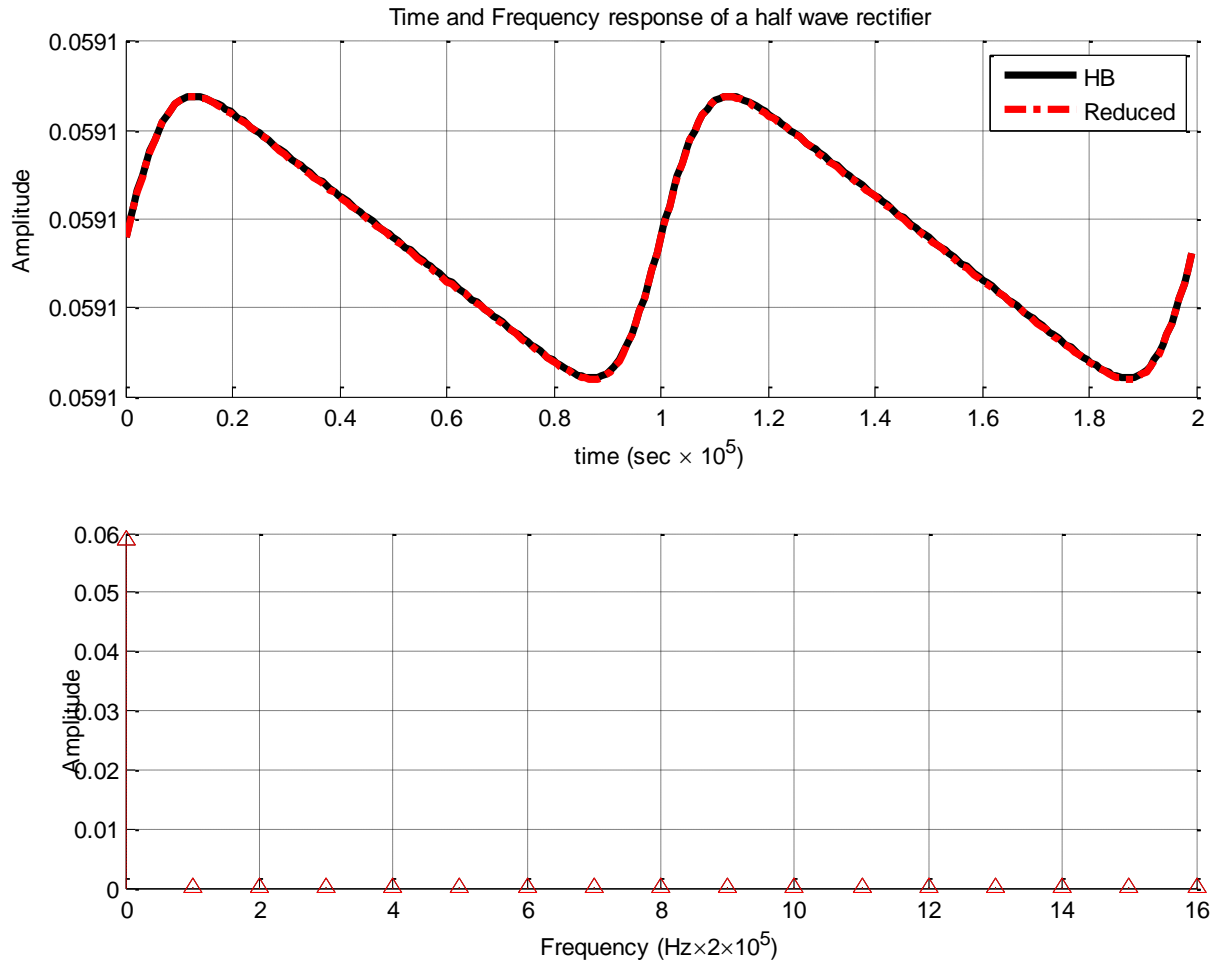


Figure IV - 5 : HB vs Proposed output

Note: $R_1 = 10K\Omega$, $R_2 = 1K\Omega$, $C_1 = C_2 = 2\mu F$, $I_d = I_{sat} e^{\frac{v(t)}{V_T}}$, $I_{sat} = 1e - 15$, and $V_T = 0.026$

II.3.2. Example two Tuned Amplifier

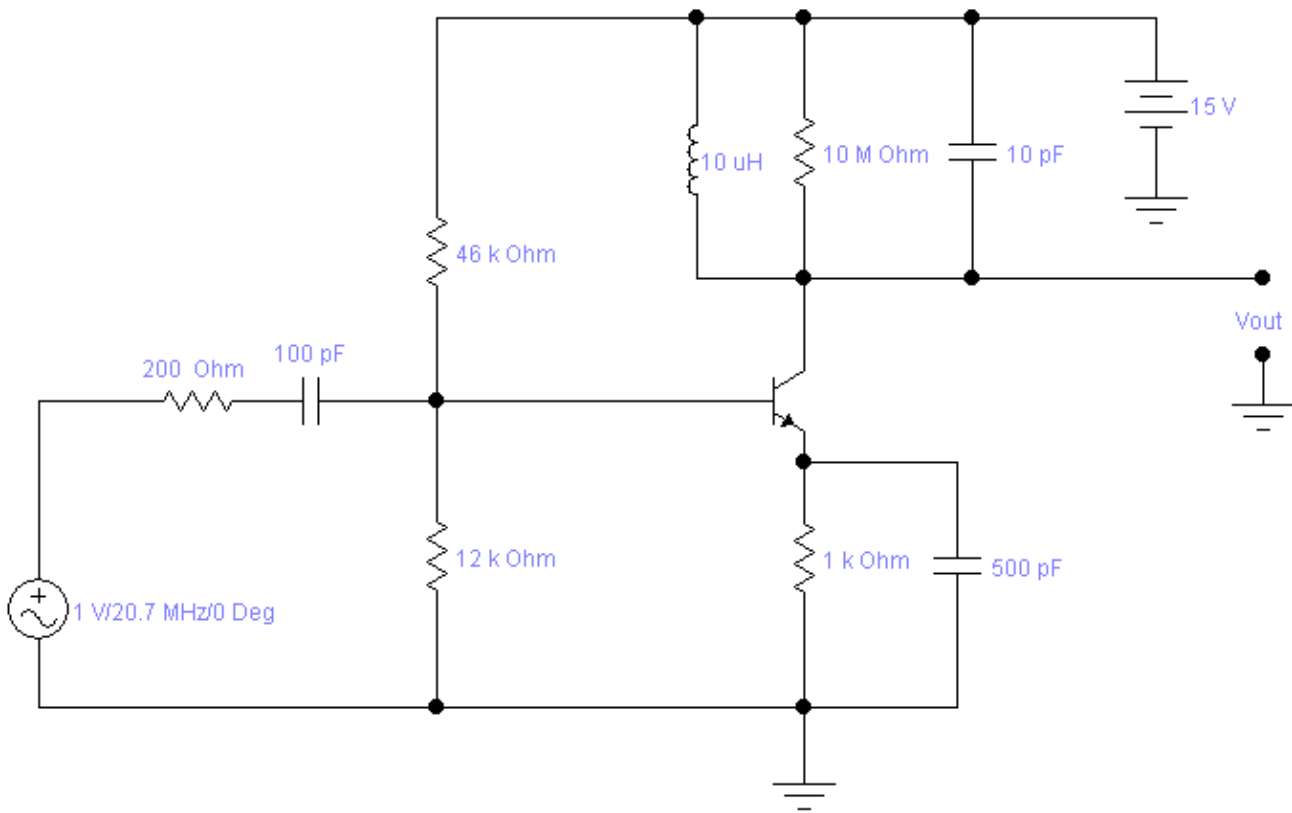


Figure IV - 6 : Tuned amplifier circuit

Figure IV-6 shows a circuit of a tuned amplifier that contains a single transistor, the transistor used in this example is a simple Ebers-Moll injection model [10], which is presented in Figure IV-7 along with the equations and values of the components.

This amplifier takes input of $v(t) = \cos(\omega_0 t)$, where $\omega_0 = 2\pi(2.07 \times 10^7)$. the HB approach required $K = 16$ harmonics to approximate the steady-state response, and the size of the original system is $n(2K + 1) = 13 \times (2 \times 16 + 1) = 429$ nonlinear equations, as for the reduced system, it was enough to solve 48 nonlinear equations to get the steady-state solution with a maximum relative error between the waveforms from the two approaches is $1.7453e-006$.

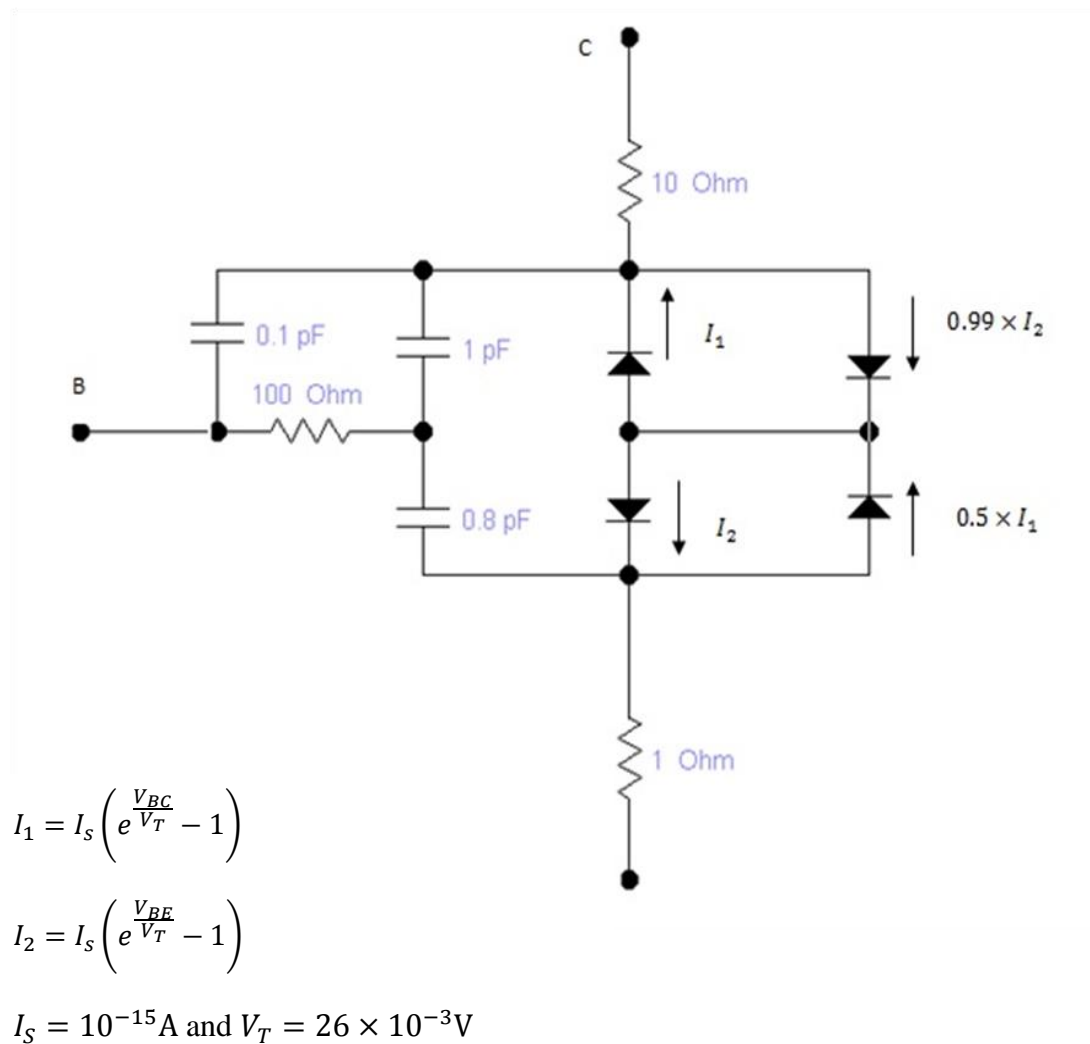


Figure IV - 7 : Ebers-Moll model

The Figure below (Fig IV- 8) shows the comparison between the time response $v_{out}(t)$ of HB original approach versus MOR algorithm, which proves the correctness of the MOR technique. Figures (Fig IV-8, 9, 10, 11) present the output response of the system as a function of the amplitude of the input waveform, this shows that the congruent transformation of order 48 is enough to match the response of the original circuit.

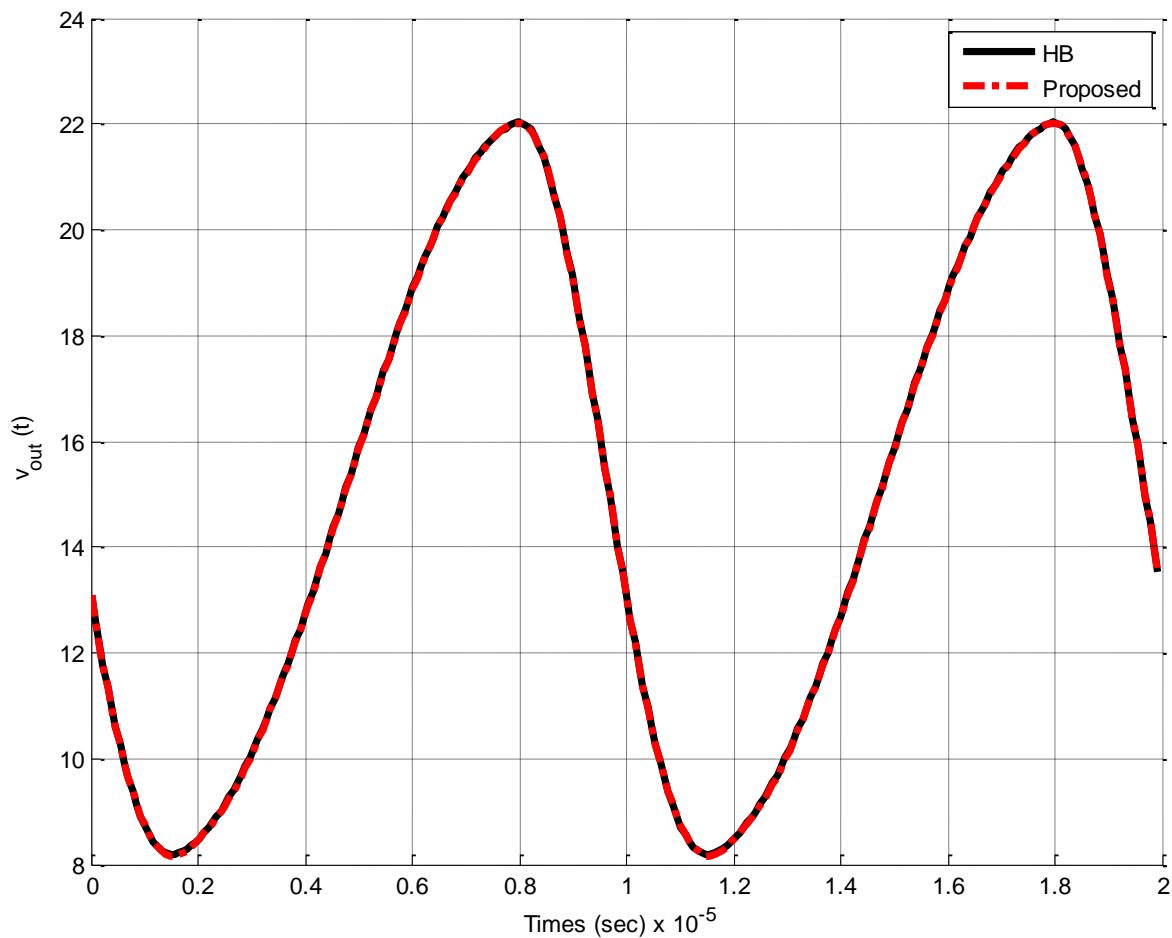


Figure IV - 8 : HB vs Proposed output

Table IV - 2 shows that original HB approach required 75 iterations, whereas, the proposed algorithm required only 46 iterations to reach steady state solution, keeping in mind that the proposed algorithm requires only one LU decomposition of the full Jacobian matrix at the time when the original harmonic balance required 75 LU decomposition of original size Jacobian.

Table IV - 2 shows that the proposed algorithm have managed to achieve a speedup factor of 5.4, while preserving accuracy of the solution.

	<i>Size of the System</i>	<i>CPU time (sec)</i>	<i>iterations</i>	<i>Construction of matrix V</i>	<i>Total required time</i>
<i>Original circuit</i>	429	12.52 (sec)	75	-	12.52 (sec)
<i>Reduced circuit</i>	48	1.45 (sec)	46	0.85 (sec)	2.3 (sec)

Table IV - 2 : shows a comparison of the conventional HB versus the proposed algorithm

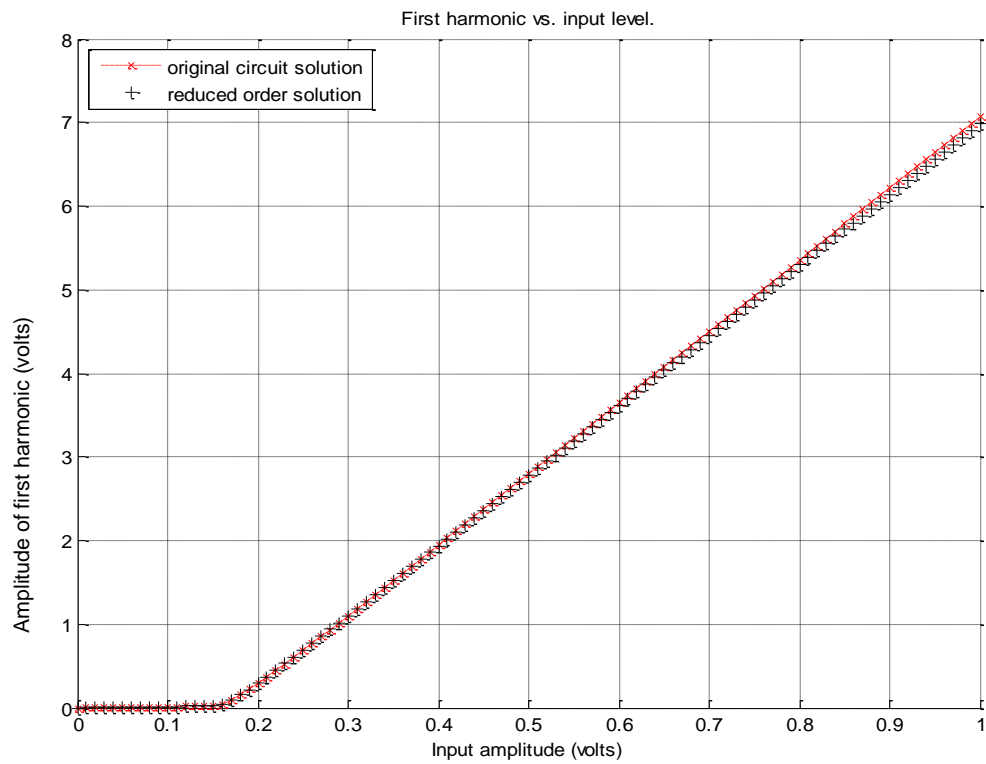


Figure IV - 9 : Solution at the first harmonic vs input level of Tuned amplifier

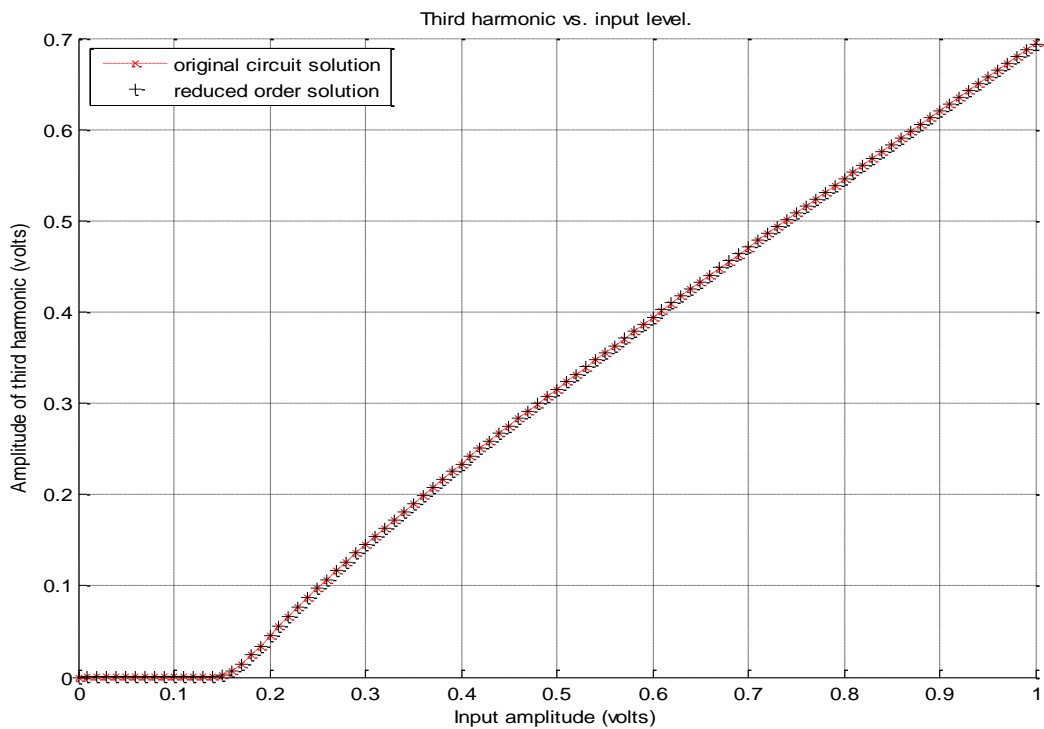


Figure IV - 10 : Solution at the third harmonic vs input level of Tuned amplifier

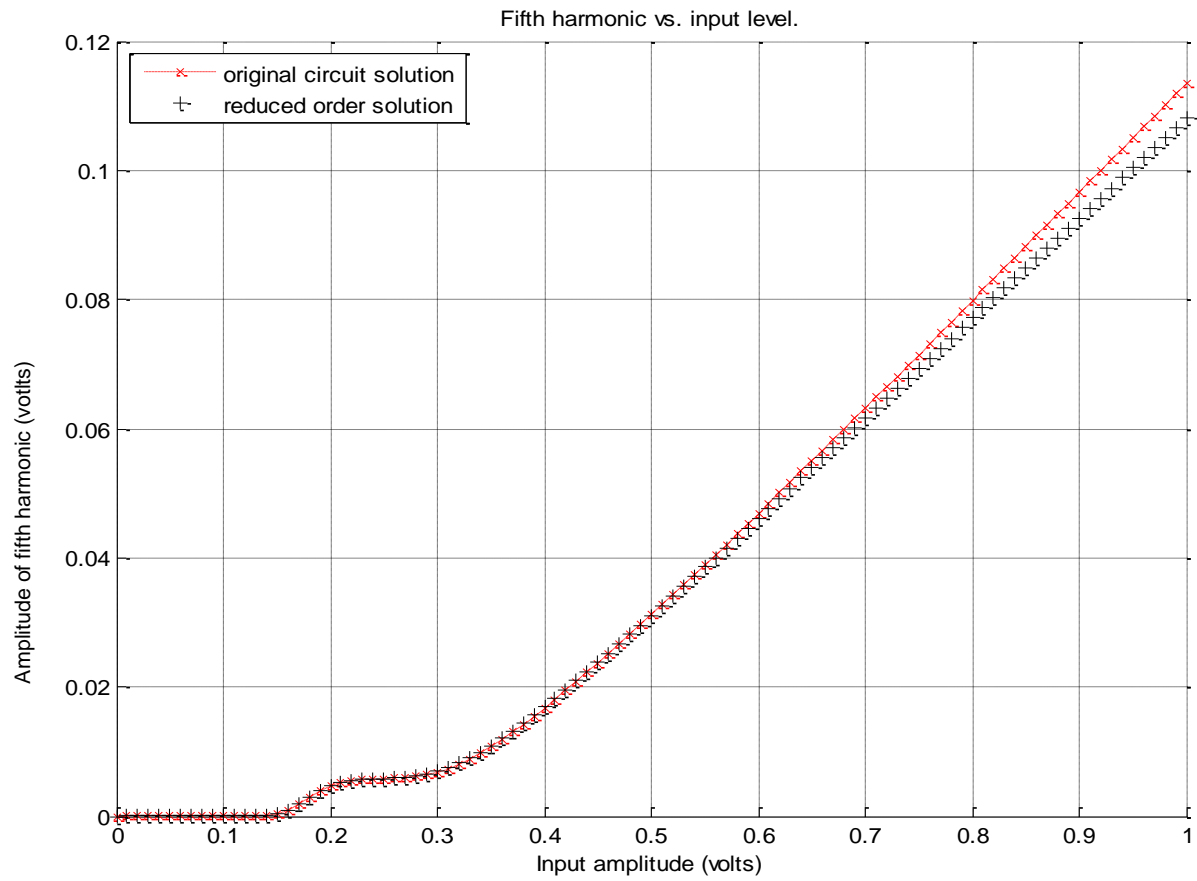


Figure IV - 11 : Solution at the fifth harmonic vs input level of Tuned amplifier

II.3.3. Example three : Power amplifier

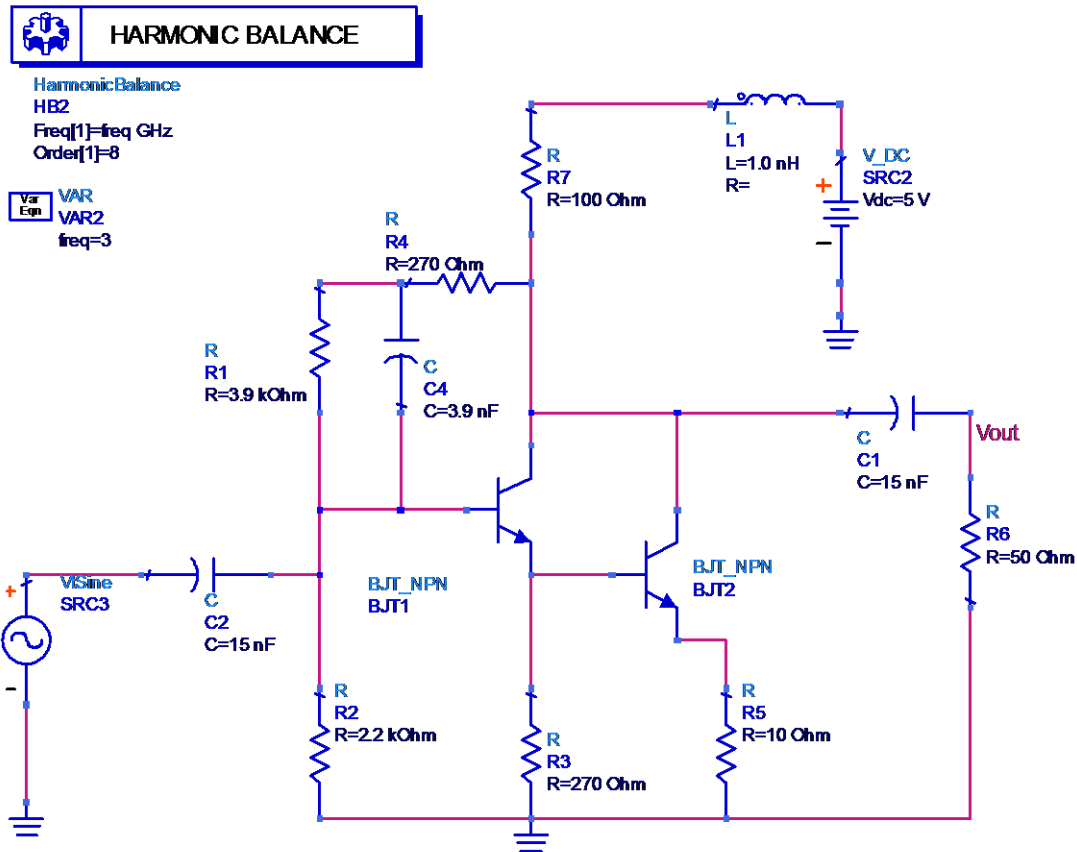


Figure IV - 12 : Power amplifier circuit

Figure IV-12 illustrates the setup for simulating a power amplifier circuit that contains two transistors. The amplifier takes input of $v_{in} = V_{in} \cos(\omega_0 t)$, where $\omega_0 = 2\pi(4.7746 \times 10^8)$. The HB approach required $K = 8$ harmonics to approximate the steady-state response, and the size of the original system is $n(2K + 1) = 21 \times (2 \times 8 + 1) = 357$ nonlinear equations, as for the reduced system, it was enough to solve 35 nonlinear equations to get the steady-state solution with a maximum relative error between the waveforms from the two approaches is $7.4581e-005$.

The Figure below (Fig IV-13) shows the comparison between $v_{out}(t)$ of HB approach versus the proposed algorithm

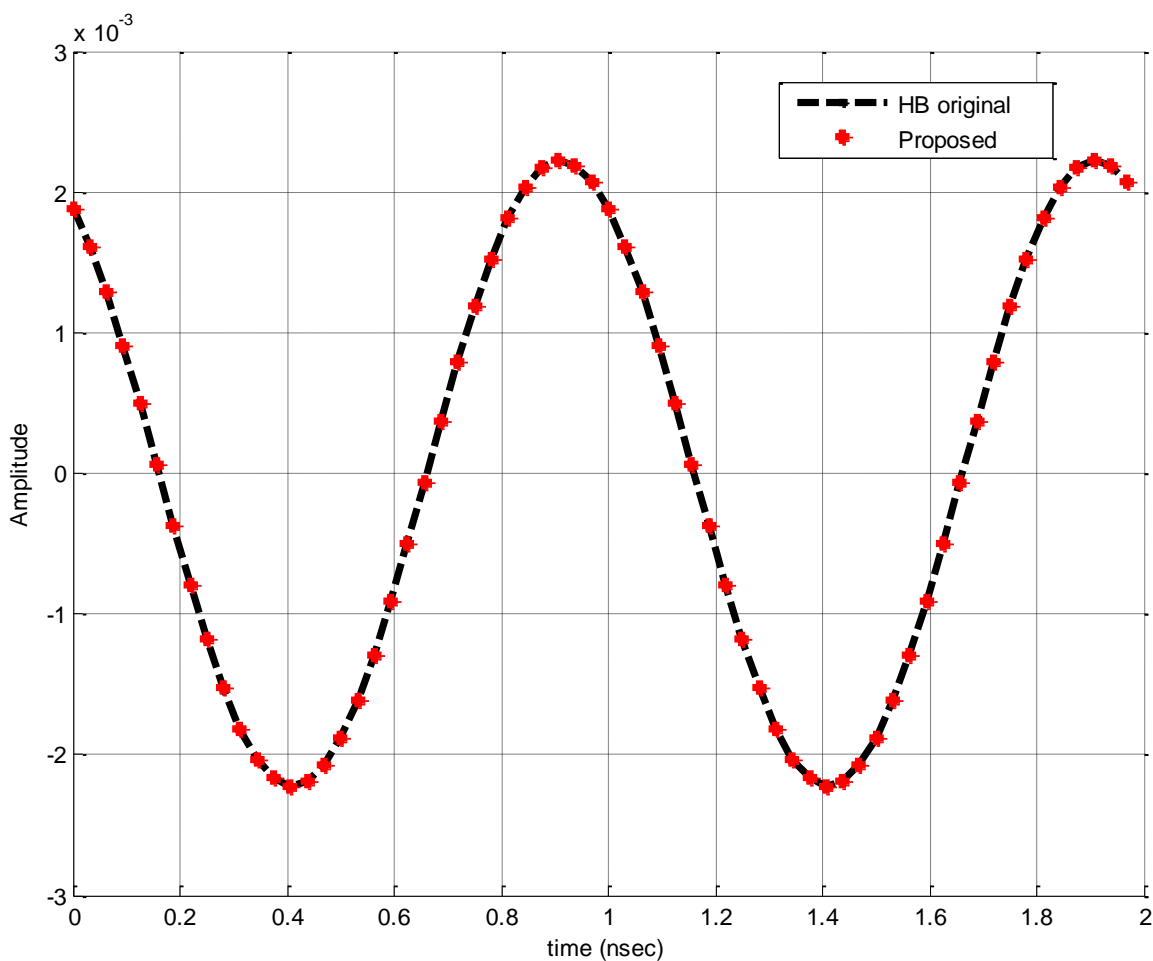


Figure IV - 13 : HB vs Proposed output

Table IV - 3 shows that HB approach required 70 LU decomposition of the Jacobian matrix that has a size of 357×357 , whereas, the proposed algorithm required only 38 iterations to reach steady state solution. Table IV - 3 shows that the proposed algorithm have managed to achieve a speedup factor of 12.3, while preserving accuracy of the solution.

	<i>Size of system</i>	<i>CPU time</i>	<i>iterations</i>	<i>Construction of V</i>	<i>Total required time</i>
<i>Original circuit</i>	<i>357</i>	<i>5.94 (sec)</i>	<i>70</i>	<i>-</i>	<i>12.52 (sec)</i>
<i>Reduced circuit</i>	<i>35</i>	<i>0.4 (sec)</i>	<i>38</i>	<i>0.98(sec)</i>	<i>2.3 (sec)</i>

Table IV - 3 : Shows a comparison of the conventional HB versus the proposed algorithm for example 3

II.4. Discussion

II.4.1. Etat de l'art

In this section, an overview of the development and improvement of the model order reduction technique that was proposed initially by Gad et al [6] will be presented.

First, in 1999, Nakhla and Gad developed a new model order reduction technique for DC solution of Large Nonlinear Circuits, using congruent transformation [12].

This approach consists of reducing the original system to a smaller set of unknowns through a congruent transformation, which is obtained using Krylov subspace.

The algorithm consists of:

- Formulating the nonlinear equation using MNA
- Computing the first few derivatives w.r.t the sweeping parameter of continuation method.
- Formulating the Krylov subspace using the derivatives obtained in the previous step
- Constructing the congruent transformation matrix V
- Applying QR decomposition method to matrix V in order to obtain an orthonormal matrix
- Reducing the system using matrix V
- Solving for steady state
- Bringing back the solution to the original size using matrix V

Afterwards, in 2000, Gad et al [6] applied the previously invented “model order reduction technique using congruent transformation” to harmonic Balance technique in order to reduce the time (CPU cost) while finding steady state solution.

The algorithm – which was reviewed in details previously in this chapter – is a continuation of the previous DC MOR technique, with an additional part added to it, which in fact describes the aspects of harmonics balance in the MOR.

In 2001, Gunupudi and nakhla [13] applied the model order reduction using congruent transformation techniques to nonlinear High-Speed Interconnect Networks.

In 2006, Praveen , Gad, et al [14] introduces a new algorithm for computing Sensitivity of Steady-State Response of Nonlinear Circuits using the Model order reduction.

the main idea, in [14], is to apply the model order reduction, proposed by Gad et al [6], to a nonlinear system in order to find the steady state solution, then, once the solution is obtained, the solution is used to compute sensitivity using few number of forward/backward substitutions, by introducing the solution into sensitivity equation system, then projecting-based reduction is preformed through a change of variable similar to the one used in the first projection.

The work done in [6][12][13][14] proves the correctness and efficiency of the MOR technique in terms of robustness, saving CPU cost, and in terms of accuracy. However, as far as we know, the limitation of this method have not yet been treated.

II.5. Limitations of the method

II.5.1. black box models

The aim of proposed algorithm is to evaluate Padé moments in order to create congruent transformation matrix V , however, computing Jacobian matrices is essential in order to evaluate these moments recursively using equation IV-23. As mentioned in section IV-II-5, the proposed algorithm depends directly on the equivalent derivatives formulas, such as the expressions in Table IV - 1, in order to compute the nonlinear part of the Jacobian matrices. This is however doable as long as the mathematical expression of the device is known by the simulator kernel, which is not the case for most user defined models where the simulators have no symbolic reach to the mathematical expression.

II.5.2. Elements of multiple variables

The most important limitation of the proposed nonlinear MOR stems from the fact that most nonlinear device equations have more than one voltage or current control variable [3].

Fig IV-11 illustrates an example of FET equivalent circuit that characterizes such situation, in this figure, we notice that the drain current I is a function of both the gate voltage V_1 and the drain voltage V_2 , therefore, $I = f(V_1, V_2)$.

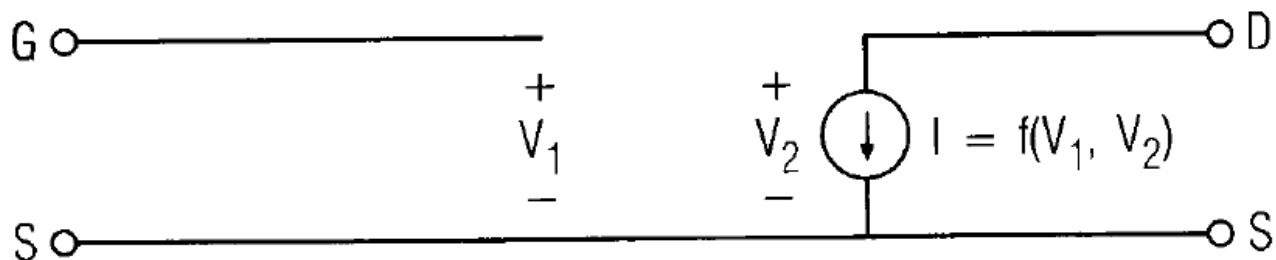


Figure IV - 14 : Nonlinear element controlled by multiple variables

The function $f(V_1, V_2)$ Taylor expansion, and for a function of only two variable, [3] describe such expansion as being “sticky,”

$$\begin{aligned} i &= \frac{\partial f}{\partial V_1} v_1 + \frac{\partial f}{\partial V_2} v_2 \\ &+ \frac{1}{2} \left(\frac{\partial^2 f}{\partial V_1^2} v_1^2 + \frac{\partial^2 f}{\partial V_1 \partial V_2} v_1 v_2 + \frac{\partial^2 f}{\partial V_2^2} v_2^2 \right) \\ &+ \frac{1}{6} \left(\frac{\partial^3 f}{\partial V_1^3} v_1^3 + 3 \frac{\partial^3 f}{\partial V_1 \partial V_2^2} v_1 v_2^2 + 3 \frac{\partial^3 f}{\partial V_1^2 \partial V_2} v_1^2 v_2 + \frac{\partial^3 f}{\partial V_2^3} v_2^3 \right) + \dots \end{aligned}$$

The equation above shows that as the order increases, it becomes more difficult and complicated to evaluate the derivatives of a function that depends on only two variables, due to the exponential growth of the derivatives combinations, that is, as the order increases; the combination of derivatives will become more difficult. Also, when the number of variables increases, evaluating required derivatives becomes a real cumbersome.

As for the proposed algorithm in [6], dealing with multi-variable nonlinear functions and finding an equivalent series expansion such as the ones in Table IV - 1 seem to be very difficult, if not impossible task to apply. As far as we know, there is no automatic tool that can find such equivalent expressions, therefore, the proposed method in [6] would find it impossible to carry on with multi-variable nonlinear elements, and the proposed algorithm comes to an end in such cases.

II.5.3. Piecewise nonlinear equation

A third limitation that the proposed method in [6] faces is dealing with “piecewise equation” nonlinear elements. A piecewise function is generally presented as follows:

$$f(x) = \begin{cases} f_1(x), & x < 0 \\ f_2(x), & x = 0 \\ f_3(x), & x > 0 \end{cases}$$

Finding an equivalent recursive mathematical expression for such equations is probably not reachable because we cannot predict all the combinations of the sub-equations that form the piecewise equation. And as mentioned above, there is no automatic way to construct such recursive expressions.

This section showed that the proposed method faces some limitations with its actual implementation, because it counts directly on equivalent series expansion expressions, and in some cases, such as black box device or multiple variable device, the simulator has no way to find those equivalent expressions. Next section will attempt to overcome this limitation by constructing a numerical differentiator that can replace the analytical mathematical expressions such as ones in Table IV - 1.

II.6. Proposed Numerical Differentiator

II.6.1. Introduction

So far in this chapter, we discussed in details the HB nonlinear model order reduction technique, which looked at the problem of reducing models from a different perspective. In other words, the algorithm makes use, first, of projecting the original system into smaller space using a congruent transformation matrix, then solves the system in the reduced space so that it decreases the numerical cost of factorizing Jacobian matrix at each NR iteration, afterwards, the steady-state response, once found, is brought back to the original system using the congruent transformation matrix.

Also, in this chapter we explained in details how to construct the congruent transformation matrix, and some numerical examples were provided in order to prove validity, correctness, and performance of the proposed algorithm.

We also presented some limitations of this algorithm, especially when the nonlinear functions in the RF circuits are considered to be black box devices; therefore, the mathematical forms of nonlinear equations in the system are not reachable, or when the mathematical form is modified by the designer in which the simulator has no equivalent power series form of the new equation. The other limitation presented is the multivariable nonlinear functions, where the evaluation of the n th order derivative becomes impossible as the order of derivative increases, because the complexity increases exponentially.

This section will study solutions to overcome the limitations of the model order reduction proposed in [6], by introducing a numerical differentiator.

III. Numerical approach

As stated before, the main complexity of the MOR technique is due to the need to obtain the time-domain polynomial coefficient \bar{g}_p of the Jacobian matrix. The numerical differentiator which will be presented in the following section aims to evaluate these coefficients numerically. once the required derivatives are constructed using the proposed numerical differentiator, the matrix \mathbf{V} can then be formed using the numerically obtained moments, and the algorithm of MOR technique listed above can proceed in the same way.

In other words, the proposed numerical differentiator will calculate the moments differently; however, the procedure of reducing and solving the system would remain the same as before.

III.1. Variable α

In the process of calculating Padé moments, it was shown that the continuant α was always replaced by zero, however, in the numerical approach, α cannot be zero, rather, it should be a small number which is close to zero, but not zero. This limitation to α has to be respected in order to avoid division by zero.

Therefore, α should be replaced by a different variables that can characterize a vector of numbers that have nonzero values

III.2. EXPLENATION OF ALGORITHM

Before introducing the numerical algorithm, it's essential to mention that the procedure of identifying MP that was explained in IV-9 would be applied similarly in this section, the only change is how to compute the Jacobian coefficients.

. The numerical algorithm is based on least square approach, it consists of

- Iteration $p = 0$

- 1) Obtain \bar{X}_0 by solving regular DC equation: $Y\bar{X}_0 + \bar{\Phi}_0 + \bar{U}_0 = \mathbf{0}$
- 2) Then compose the zero order polynomial coefficient of state variable: $\bar{X}_0 \equiv [\mathbf{0}, \dots, X_0, \dots \mathbf{0}]^T$
- 3) Apply IFFT to map the state variable coefficient into time domain: $\bar{x}_0 = \Gamma^{-1}\bar{X}_0$
- 4) Compute time domain zero order coefficient of the Jacobian with respect to \bar{x}_0 : $\bar{g}_0 = \partial i(\bar{x}_0)/\partial \bar{x}_0$
- 5) Map time domain Jacobian coefficient to frequency domain: $\bar{J}_0 = \Gamma \bar{g}_0 \Gamma^{-1}$
- 6) Form and LU factor the small signal admittance matrix: $(Y + \bar{J}_0) = (Y + \Gamma \bar{g}_0 \Gamma^{-1})$

- Iteration: $p = 1$

- 1) Solve 2nd equation of (IV-19) and get first order coefficient of state variable \bar{X}_1 :

$$(Y + \bar{J}_0)\bar{X}_0 = -U_1$$
- 2) Map the state variable coefficient into time domain: $\bar{x}_1 = \Gamma^{-1}\bar{X}_1$
- 3) Apply the formal vector of power series $x(\alpha) = X_0 + X_1\alpha$ to the nonlinear equation (eq-22) and identify the Jacobian first order time-domain coefficient as follows:
 - i. Calculate $g(\alpha)$ in terms of \bar{x}_0 and \bar{x}_1 : $g(\alpha) = \left. \frac{\partial f(x_n(\alpha))}{\partial \alpha} \right|_{n=0,1}$
 - ii. Expand $g(\alpha)$: $g(\alpha) = \bar{g}_0 + \bar{g}_1\alpha + \bar{g}_2\alpha^2 + \dots + \bar{g}_p\alpha^p$
 - iii. Substutue $g(\alpha)$ from g_0 and divide by α : $\frac{g(\alpha) - \bar{g}_0}{\alpha} = \bar{g}_1 + \bar{g}_2\alpha + \dots + \bar{g}_{\hat{D}-1}\alpha^{\hat{D}-2}$

- iv. Replace $\frac{g(\alpha) - \bar{g}_0}{\alpha}$ by $\hat{g}_1(\alpha)$
 - v. Apply the fitting algorithm to \hat{g}_1 w.r.t. α , in order to obtain a numerical approximation for \bar{g}_1
- 4) Map time domain Jacobian coefficient to frequency domain: $\bar{J}_1 = \Gamma \bar{g}_1 \Gamma^{-1}$
- Higher order iterations: $p \geq 2$
 - 1) Solve 3rd equation of (eq-19) and get p th order coefficient of state variable \bar{X}_p

$$(Y + \bar{J}_0)\bar{X}_p = -\frac{1}{p} \sum_{i=1}^{p-1} (p-1)\bar{J}_i \bar{X}_{p-i}$$
 - 2) Map the state variable coefficient into time domain: $\bar{x}_p = \Gamma^{-1} \bar{X}_p$
 - 3) Apply the formal vector of power series $x(\alpha) = \sum_{i=0}^p \bar{x}_i \alpha^i$ to the nonlinear equation (eq-22) and identify the Jacobian p th order time-domain coefficient \bar{g}_p as follows:
 - i. Calculate $g(\alpha)$ in terms of $x(\alpha) = \bar{x}_0 + \bar{x}_1 \alpha + \dots + \bar{x}_p \alpha^p$:
$$g(\alpha) = \left. \frac{\partial f(x_n(\alpha))}{\partial \alpha} \right|_{n=0,1,\dots,p} \quad \text{IV - 26}$$
 - ii. Calculate $\hat{g}_p(\alpha) = \frac{g(\alpha)}{\alpha^i} - \sum_{k=0}^{p-1} \frac{\bar{g}_k(\alpha)}{\alpha^{p-k-1}}$
 - iii. Apply the fitting algorithm to $\hat{g}_p(\alpha)$ w.r.t. α , in order to obtain a numerical approximation for \bar{g}_p
 - 4) Map time domain Jacobian coefficient to frequency domain: $\bar{J}_p = \Gamma \bar{g}_p \Gamma^{-1}$
 - 5) $p \leftarrow p + 1$
 - 6) loop until $p = \hat{D} - 1$

III.2.1. Fitting algorithm

Among least square methods, the Chebyshev polynomials are chosen because they are considered to be the most important family of orthogonal polynomials [15]. Their properties can be derived by rather simple methods (Annex A)

The fitting of vector $\hat{\mathbf{g}}_i(\boldsymbol{\alpha})$ w.r.t. α . is carried in the following way:

Define the vector of variable $\boldsymbol{\alpha}$ which should vary between a small negative value which is close to zero and a positive small value different than zero. Also, the vector should not contain zero, otherwise, we end up with a division by zero error. The size of the vector is constant which should be specified by user:

$$\boldsymbol{\alpha} \rightarrow [-\alpha_n: \Delta: \alpha_p] \quad IV-27$$

Chebyshev polynomials algorithm takes the following variables as input:

- The nonlinear function $\hat{\mathbf{g}}(\boldsymbol{\alpha}, t_i)$
- The vector $\boldsymbol{\alpha}$
- The Curve fitting order which is a constant defined by user

And as output, it gives a set of coefficients as a polynomials

It was shown before:

$$\mathbf{x}(\boldsymbol{\alpha}) = [\mathbf{x}(\boldsymbol{\alpha}, t_0), \mathbf{x}(\boldsymbol{\alpha}, t_1), \dots, \mathbf{x}(\boldsymbol{\alpha}, t_{S-1})]^T \quad IV-28$$

$$\mathbf{g}(\boldsymbol{\alpha}) = [\mathbf{g}(\boldsymbol{\alpha}, t_0), \mathbf{g}(\boldsymbol{\alpha}, t_1), \dots, \mathbf{g}(\boldsymbol{\alpha}, t_{S-1})]^T$$

$$\bar{\mathbf{g}}_p(\boldsymbol{\alpha}) = [\bar{\mathbf{g}}_p(t_0), \bar{\mathbf{g}}_p(t_1), \dots, \bar{\mathbf{g}}_p(t_{S-1})]^T, \quad p = 0, \dots, \hat{D} - 1$$

The key point in this algorithm is that $\mathbf{g}(\boldsymbol{\alpha}, t_i)$ varies with each iteration, that means, for every iteration $\mathbf{g}(\boldsymbol{\alpha}, t_i)$ is calculated according to the new coefficient of $\mathbf{x}(\boldsymbol{\alpha}, t_i)$ that was obtained in the previous iterations. Therefore, we can define a new variable $\hat{\mathbf{g}}_p(\boldsymbol{\alpha}, t_i)$:

$$\hat{\mathbf{g}}_p(\boldsymbol{\alpha}, t_i) = \frac{g(\boldsymbol{\alpha}, t_i)}{\alpha^i} - \sum_{k=0}^{p-1} \frac{\bar{\mathbf{g}}_k(\boldsymbol{\alpha}, t_i)}{\alpha^{p-k-1}} \quad IV-29$$

Gathering all time samples $\hat{\mathbf{g}}_p(\boldsymbol{\alpha}, t_i)$ of power coefficients of the same order for the conductance, we have the vector coefficient $\hat{\mathbf{g}}_p(\boldsymbol{\alpha})$:

$$\hat{\mathbf{g}}_p(\boldsymbol{\alpha}) = [\hat{\mathbf{g}}_p(t_0), \hat{\mathbf{g}}_p(t_1), \dots, \hat{\mathbf{g}}_p(t_{S-1})]^T, \quad p = 0, \dots, \hat{D} - 1 \quad IV - 30$$

- Step 1: calculate $\mathbf{x}(\boldsymbol{\alpha})$

As a first step, we have to calculate $\mathbf{x}(\boldsymbol{\alpha})$ (eq -28) in terms of $\boldsymbol{\alpha}$. In section IV-9, $\boldsymbol{\alpha}$ was set to zero, however, Chebyshev polynomials algorithm requires that $\boldsymbol{\alpha}$ has to be a vector that varies between two small values, nearly zero, as defined above.

$$\mathbf{x}(\boldsymbol{\alpha}, t_0) = [x(\alpha_n, t_0), \dots, x(\alpha_p, t_0)] \quad IV - 31$$

⋮

$$\mathbf{x}(\boldsymbol{\alpha}, t_{S-1}) = [x(\alpha_n, t_S), \dots, x(\alpha_p, t_{S-1})]$$

Where the size of $\mathbf{x}(\boldsymbol{\alpha}, t_S)$ is the same size as the vector $\boldsymbol{\alpha}$

- Step 2: calculate $\mathbf{g}(\boldsymbol{\alpha})$

Calculate $\mathbf{g}(\boldsymbol{\alpha})$ in terms of $\mathbf{x}(\boldsymbol{\alpha})$ using (eq-26).

- Step 3: calculate $\hat{\mathbf{g}}(\boldsymbol{\alpha})$

Form the vector $\hat{\mathbf{g}}_p(\boldsymbol{\alpha})$ using (eq- 29)

$$\hat{\mathbf{g}}_p(\boldsymbol{\alpha}, t_0) = [\hat{\mathbf{g}}_p(\alpha_n, t_0), \dots, \hat{\mathbf{g}}_p(\alpha_p, t_0)] \quad IV - 32$$

⋮

$$\hat{\mathbf{g}}_p(\boldsymbol{\alpha}, t_{S-1}) = [\hat{\mathbf{g}}_p(\alpha_n, t_S), \dots, \hat{\mathbf{g}}_p(\alpha_p, t_{S-1})]$$

- Step 4: evaluate the Chebyshev polynomials

for every sample of time, apply the fitting algorithm for $\hat{\mathbf{g}}_p(\boldsymbol{\alpha}, t_i)$.

$$\bar{\mathbf{g}}_p(\boldsymbol{\alpha}, t_0) = \mathbf{ChebPoly}(\hat{\mathbf{g}}_p(\boldsymbol{\alpha}, t_0), \boldsymbol{\alpha}) \quad IV - 33$$

⋮

$$\bar{\mathbf{g}}_p(\boldsymbol{\alpha}, t_{S-1}) = \mathbf{ChebPoly}(\hat{\mathbf{g}}_p(\boldsymbol{\alpha}, t_{S-1}), \boldsymbol{\alpha})$$

Chebyshev polynomials algorithm takes as input the vector $\hat{\mathbf{g}}_p(\boldsymbol{\alpha}, t_i)$, the vector $\boldsymbol{\alpha}$, and the number of coefficients, and gives as output the calculated polynomials

.

- Step $p = 5$ obtain $\bar{g}_p(\alpha)$

After fitting vector $\hat{g}_p(\alpha, t_i)$, the Chebychev polynomials are collected for every instance of time. Consider the first Chebyshev Polynomial coefficient of every $\hat{g}_p(\alpha, t_i)$, this would approximate the vector $\bar{g}_p(\alpha)$

As an example, consider again a circuit with a single node and a single nonlinear function $f(\alpha)$:

$$f(x(\alpha)) = I_{sat} \left(e^{\frac{v(x(\alpha))}{V_T}} - 1 \right)$$

Therefore $g(\alpha)$ would be :

$$g(\alpha) = \frac{I_{sat}}{V_T} \left(e^{\frac{1}{V_T}(\bar{x}_0 + \bar{x}_1\alpha + \bar{x}_2\alpha^2 + \dots + \bar{x}_{\hat{D}-1}\alpha^{\hat{D}-1})} \right) = \bar{g}_0 + \bar{g}_1\alpha + \dots + \bar{g}_{\hat{D}-1}\alpha^{\hat{D}-1}$$

\bar{x}_0 is the dc solution, \bar{g}_0 is obtained using equation (IV-16), then \bar{x}_1 is computed using equation (IV-19).

Having \bar{x}_0, \bar{g}_0 , and \bar{x}_1 , then, \bar{g}_1 is computed as follows:

$$g(\alpha) = \frac{I_{sat}}{V_T} \left(e^{\frac{1}{V_T}(\bar{x}_0 + \bar{x}_1\alpha)} \right) = \bar{g}_0 + \bar{g}_1\alpha + \dots + \bar{g}_{\hat{D}-2}\alpha^{\hat{D}-2}$$

$$\frac{g(\alpha) - \bar{g}_0}{\alpha} = \bar{g}_1 + \bar{g}_2\alpha + \dots + \bar{g}_{\hat{D}-2}\alpha^{\hat{D}-2}$$

For simplicity of presentation, consider $\hat{g}_1(\alpha) \equiv \frac{g(\alpha) - \bar{g}_0}{\alpha}$.

The next step is to compute the numerical value of $\hat{g}_1(\alpha)$ by replacing $g(\alpha)$, g_0 , and vector α by their real values, then fitting $\hat{g}_1(\alpha)$ w.r.t. α for every instance of time in vector $\hat{g}_1(\alpha)$, then \bar{g}_1 is formed by considering the first coefficient of chebyshev polynomials

Once \bar{g}_1 is approximated, then α_2 can be obtained using equation (IV-19)

So far, we obtained the values of α_0, α_1 and α_2 , therefore $g(\alpha)$ can be computed as follows:

$$g(\alpha) = \frac{I_{sat}}{V_T} e^{\frac{a}{V_T}} = \frac{I_{sat}}{V_T} e^{\frac{1}{V_T}(\bar{x}_0 + \bar{x}_1\alpha + \bar{x}_2\alpha^2)}$$

The approximation of \bar{g}_2 is evaluated in the same way as \bar{g}_1 . Consider equation IV 29 again:

$$\frac{g(\alpha) - \bar{g}_0}{\alpha} = \bar{g}_1 + \bar{g}_2\alpha + \bar{g}_3\alpha^2 + \dots + \bar{g}_n\alpha^{n-1}$$

We substitute \bar{g}_1 from $\frac{g(\alpha) - \bar{g}_0}{\alpha}$ and divide it all by α , therefore the equation will be reorganized as follows:

$$\frac{\left(\frac{g(\alpha) - \bar{g}_0}{\alpha}\right) - \bar{g}_1}{\alpha} = \bar{g}_2 + \bar{g}_3\alpha + \bar{g}_4\alpha^2 + \dots + \bar{g}_n\alpha^{n-2}$$

The above equation can be represented differently:

$$\hat{g}_2(\alpha) = \bar{g}_2 + \bar{g}_3\alpha + \bar{g}_4\alpha^2 + \dots + \bar{g}_{n-2}\alpha^{n-2} \quad IV-35$$

Fitting $\hat{g}_2(\alpha)$ in equation (eq-35) w.r.t to α the same way, which will end up with an approximate to the second derivative of the system. Obtaining \bar{g}_2 would allow us to compute J_2 , therefore, \bar{X}_2

This process is then repeated until $p = \hat{D} - 1$

III.2.2. Computational Complexity:

The proposed algorithm has two main sources of computational complexity. The first CPU computational effort is characterized by to the moments' evaluation of the transformation matrix, and the second CPU computational cost is due to solving the reduced system and finding the steady-state solution.

In this section, we will study the computational cost of both sources, and we will show a comparison between the analytical algorithm and the numerical one.

III.2.2.1. Evaluation of Moments:

The first computational cost of the numerical differentiator comes from the process of evaluating moments. The evaluation of moments by itself could be divided into categories, the first one is computing vectors \mathbf{g}_n , and the second one is the vector fitting of functions with respect to α , using Chebyshev Polynomials.

Actually, the first category could not be considered as a heavy load on the CPU, because it mainly consists of vector subtraction and division by a constant, which are considered to be linear behavior, $\mathbf{O}_1 = \mathbf{N}_h \times \mathbf{S}$.

The Chebyshev polynomials factorization is similar to FFT in terms of computation complexity [16], it requires in general $\mathbf{O} = \mathbf{S} \mathbf{N}_h \log \mathbf{S} \mathbf{N}_h$, where S is the time sampling, and $\mathbf{N}_h = \mathbf{N}(\mathbf{2H} + \mathbf{1})$, the number of nodes and harmonics in the system. However, the numerical differentiator algorithm requires to factorize the ‘‘Chebyshev polynomials’’ on all nodes of the systems, and for each time sample, moreover, this task is repeated $\widehat{\mathbf{D}}$ times. Therefore, the computational complexity will end up as $\mathbf{O} = \widehat{\mathbf{D}}(\mathbf{S} \mathbf{N}_h \times \mathbf{S} \mathbf{N}_h \log(\mathbf{S} \mathbf{N}_h))$, and the total cost would be $\mathbf{O}_2 = \widehat{\mathbf{D}} \times \mathbf{S}^2 \mathbf{N}_h^2 \log(\mathbf{S} \mathbf{N}_h)$

Where $\widehat{\mathbf{D}}$ is the number of derivatives required to construct the transformation matrix, \mathbf{N} is the number of nodes in the system, \mathbf{N}_h is the time samples in the system and \mathbf{n} is the polynomials coefficient. The upper bound of algorithm's running time is taken into consideration, that means

$$\mathbf{O} = (\mathbf{O}_1 + \mathbf{O}_2) = (\mathbf{SN}_h) + \hat{\mathbf{D}} \times \mathbf{S}^2 \mathbf{N}_h^2 \mathbf{Log}(\mathbf{SN}_h)$$

Removing the constants for simplicity, therefore, the computational complexity that comes from evaluating the moments could be characterized as follows:

$$\mathbf{O} \approx \mathbf{S}^2 \mathbf{N}_h^2 \mathbf{Log}(\mathbf{SN}_h)$$

III.2.2.2. Computation complexity of Construction and Solution of the Reduced system

Once the transformation matrix has been constructed, the following task is to apply reduction method in order to solve the system for steady state solution. This task is actually considered to be the second source of CPU cost which must be similar for both analytical and numerical approaches.

As explained in section IV-3, NR is used to solve the system of nonlinear equations of reduced harmonic balance. This technique requires reducing the Jacobian matrix before computing its inverse, which in fact reduces the calculation of solving the system. However, the source of computation complexity stems from the need to apply FFT and IFFT for the original size Jacobian matrix, at each iteration.

It is proved in [6] that the computational cost, of reducing Jacobian matrix using transformation matrix then applying Fourier transform, would be presented as follows:

$$O(SN_h \text{Log}(N_h))$$

Where $N_h = N(2H + 1)$ and S is the sampling time points in the system.

The advantage and drawback of the proposed algorithm will be discussed in the next section

III.2.3. Advantages and drawback of the numerical algorithm:

In this section, we will discuss the advantages of the proposed numerical algorithm and the drawback as well.

III.2.3.1. Advantages

It was shown earlier in this chapter that the analytical algorithm has several limitations, one of them is that when the analytical mathematical power series expression is not given for a specific nonlinear device in the system, the whole method of constructing an orthogonal base comes to an end.

However, the mathematical expression could not be available for two main reasons, the first is that the nonlinear equations in nature are unlimited, therefore they cannot be totally covered, and the second reason is that when multiple variables are introduced, the calculation of n th derivative becomes exhaustive because the complexity increases exponentially.

The main advantage of the numerical differentiator is that it overcomes this limitation, because it requires the first derivative only, therefore, it'd be a lot easier to find the first derivative of multi-variable function and for any other nonlinear equation in the system as well.

The second advantage of the numerical differentiator is that once the Projection base V is constructed, the time required to reduce the system and solve it is almost identical to the analytical method.

III.2.3.2. Disadvantages:

a. Significant amount of time needed for evaluation of the moments

The numerical differentiator faces a first disadvantage that it requires much more time to approximate the moments than the numerical algorithm, as it is shown in Table IV - 3, even though Chebyshev polynomials method is considered to be one of the best least square methods, however, the computation cost is significant once this method is applied to each instance of time of the nonlinear functions.

b. Cascaded error

In the numerical differentiator, we notice that \mathbf{g}_1 in the algorithm depends directly on \mathbf{g}_0 :

$$\mathbf{g}_1 = \mathit{fitting} \left(\frac{g(\alpha) - g_0}{\alpha}, \alpha \right), \text{ where } \alpha \in \mathbb{Z}^*$$

And \mathbf{g}_2 , depends on both \mathbf{g}_0 and \mathbf{g}_1 as well:

$$\mathbf{g}_2 = \mathit{fitting} \left(\frac{g(\alpha) - \frac{g_1}{\alpha} - g_0}{\alpha^2} \right)$$

In other words, every derivative depends directly on all the previous ones, this applies to \mathbf{g}_n , which depends on $\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_{n-1}$. Therefore, we can conclude that the numerical algorithm is actually as cascaded algorithm, hence the error in this algorithm

Usually, cascaded error is a type of error that's very hard to control, the reason is that the new error (e_1) is a sum of the error generated while computing vector \mathbf{g}_1 and the previous error (e_0) which is generated while computing \mathbf{g}_0 , therefore, e_0 is amplified in e_1 , and therefore, both e_0 and e_1 are amplified in e_2 , as \hat{D} increases, the error would increase, and becomes more incontrollable.

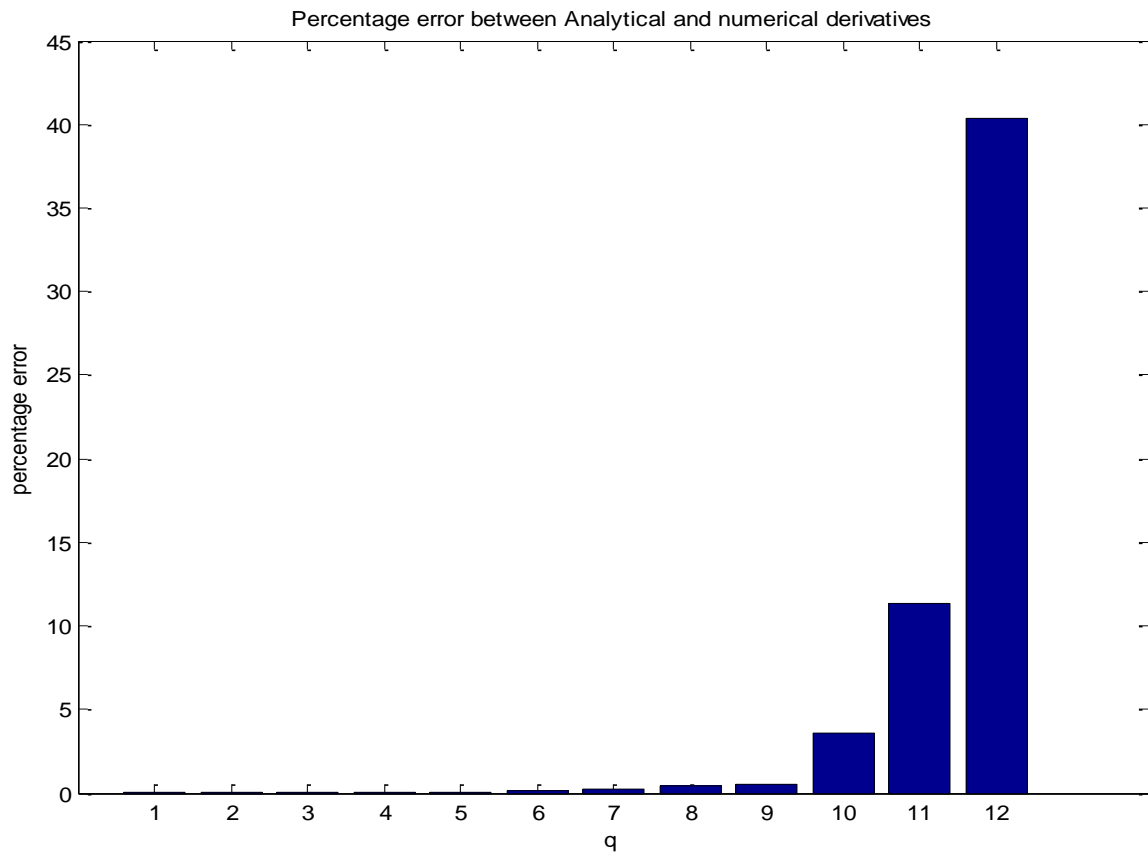


Figure IV - 15 ::Percentage error difference of Analytical and Numerical derivatives of reduced system

Figure above (Fig IV-15) shows how the percentage error between analytical and numerical error increases as \hat{D} increases, this dramatic increase is due to the fact of cascaded error, which means that the error in the last derivative always considers all the previous errors, at one moment, the error becomes enormous.

Let's consider the same graph, but for $\hat{D} = 6$:

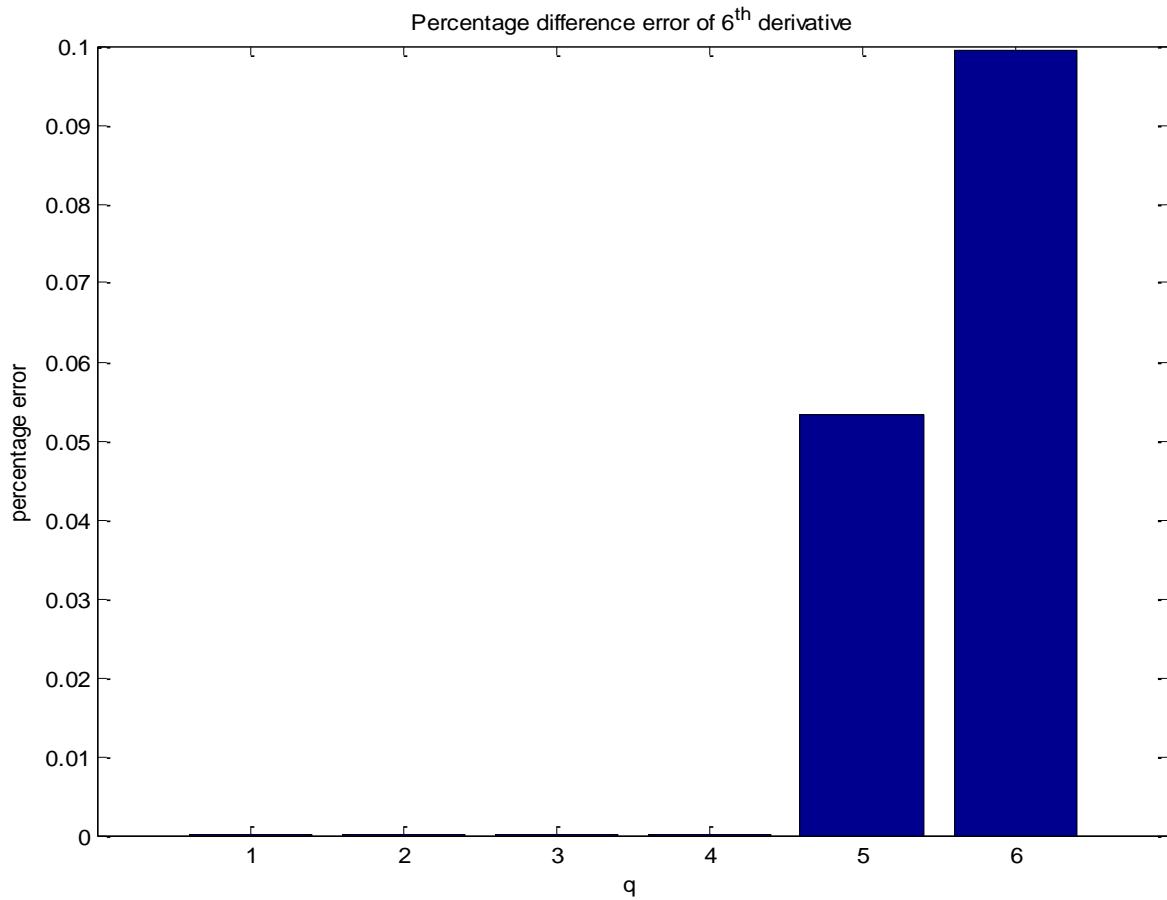


Figure IV - 16 :: Percentage of error difference of sixth derivative of the system

The figure above proves that the behavior of error is unique, that means, it increases in the same way, and as the order \hat{D} increases, the error is multiplied.

c. Dependency on the congruent transformation order \hat{D}

It was stated in [6] that “The order of the reduced model \hat{D} is increased if the residual error is greater than a pre-specified tolerance.” That is, once the simulation is terminated, and the steady-state solution is not yet reached, q must be increased to some extent and the simulator should run again. However, there is no criteria of the incremental rate required in order to increase \hat{D} , and how many times the simulator would run before the optimal V matrix is achieved, in other words, steady-state solution is reached.

The figure below (Fig IV- 17) shows that as the order \hat{D} increases, the resolution of the solution becomes closer to the original solution.

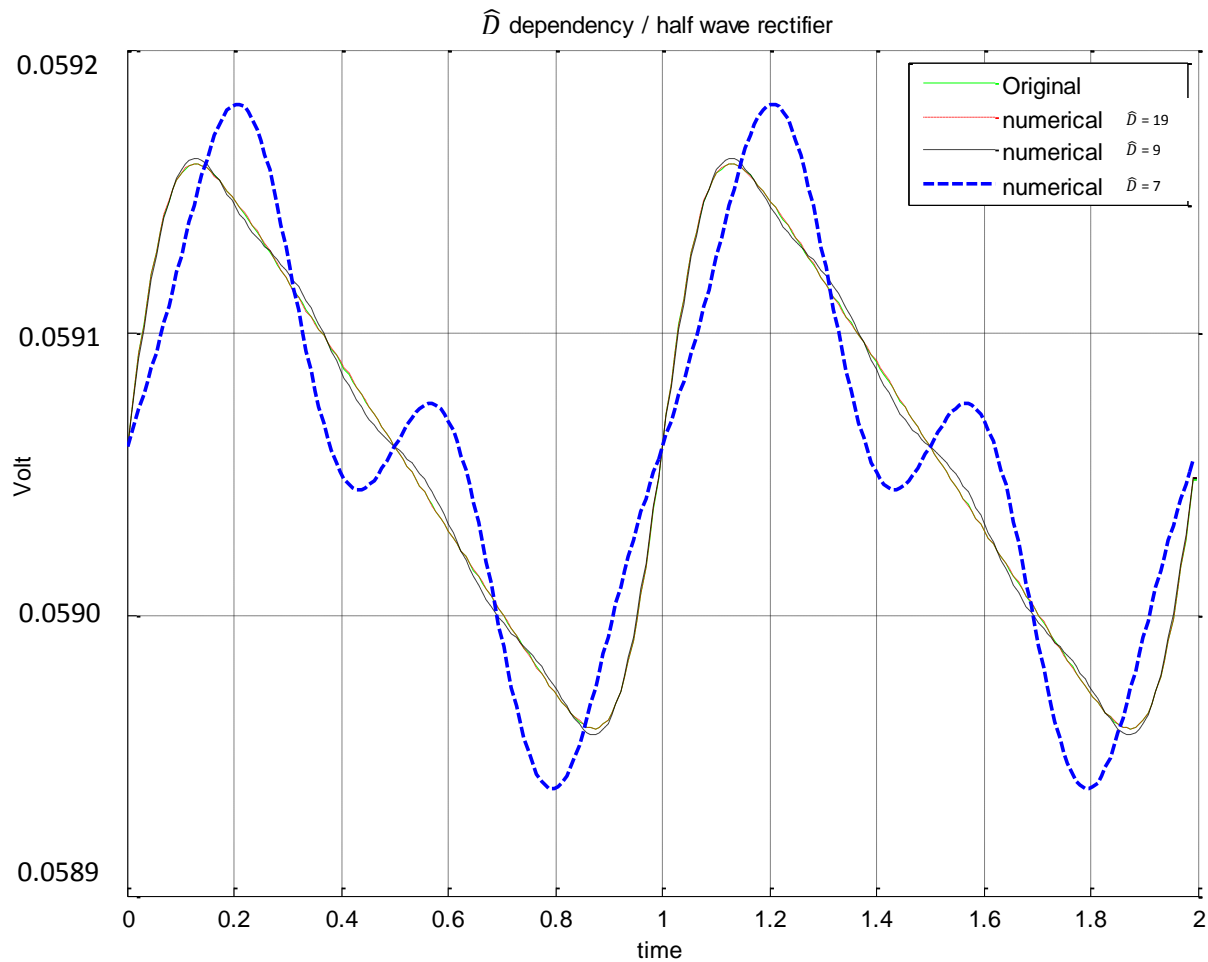


Figure IV - 17 : Percentage of error difference of sixth derivative of the system

d. Dependency on the variation of vector α

The Chebyshev polynomials properties requires the variation of parameter α . from a small negative number to a small positive number which are both close zero, but not null

$$\cdot \alpha_{chebyshev} = [-\alpha: \frac{1}{\Delta}: + \alpha]$$

This variable wasn't considered to be a drawback in the analytical method, due to the fact that it was always set to zero after every derivative operation, recall equation IV- 22, because this fact helps simplifying and simplifies the mathematical deduction.

However, in the numerical algorithm, it's necessary to replace the variable α by real values, because the fitting algorithm is based on fitting the equation w.r.t variable $\alpha_{chebyshev}$

The main drawback is caused by the necessity to find the optimal α which must be close enough to zero. For this reason, we will sweep α from a very small value to one, and we observed the outcome by terms of comparing the error of original system against the reduced one, and then calculated the percentage error between the two systems in order to find the optimal value for α .

Precision	Percentage error
0.01	significant
0.05	0.01364%
0.09	0.03787%
0.1	0.00100%
0.2	significant
0.3	Numerical error

Table IV - 4 Effect of variation of factor α

Table IV - 4 shows that α needs to be swept in order to have the optimal solution, however, the drawback is that criteria of sweeping α is based on trial and error, and it cannot be fixed to a standard value, because it differs from a circuit to another.

IV. Numerical results

IV.1. Example of half wave rectifier

Consider the example of half wave rectifier, figure IV-4, this circuit requires an order \hat{D} equal to 20 to achieve a good steady state accuracy.

The figures below (Fig IV- 18, 19) illustrate the behavior of the numerical algorithm for different Sampling rates. as we can see, the time required to evaluate the first derivative is too little in comparison to the other derivatives because there is no fitting applied in this iteration. However, at order 2, we notice that the curve increases drastically for both trajectory in the same behavior, which is in fact due to the fitting algorithm which requires to compute the Chebyshev polynomials for every time sampling of the nonlinear device. Also, we notice that from order 2 and up the increase of time is almost linear.

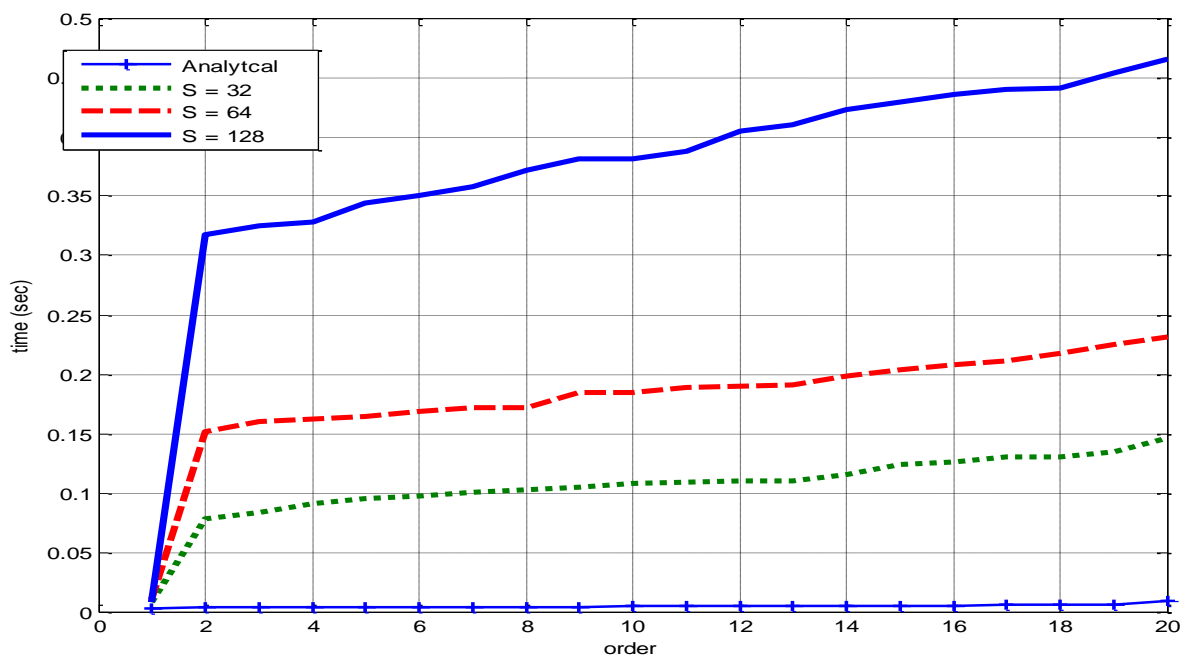


Figure IV - 18 :Time required to analytically compute the Moments of the Power amplifier circuit

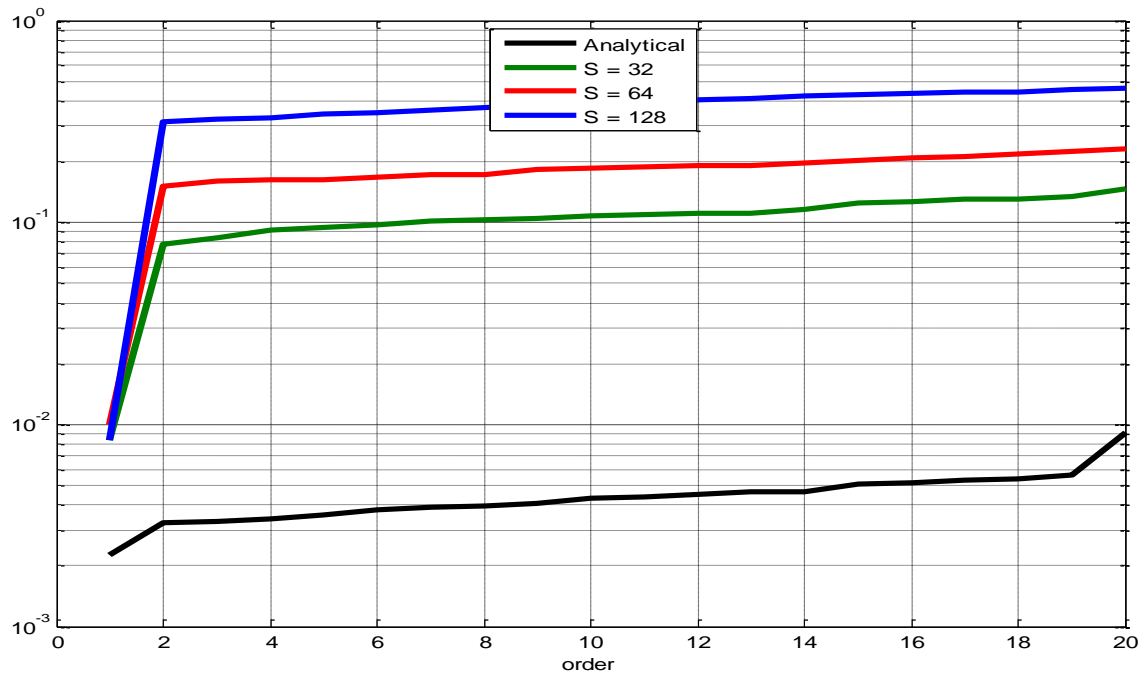


Figure IV - 19 :computational complexity of moment evaluation

IV.2. Example of a power amplifier

Consider an example of a power amplifier that contains one transistor, $v_{in} = 0.01$, fundamental frequency is 20 MHz, order 7, and it required \hat{D} (reduce order) equal to 16.

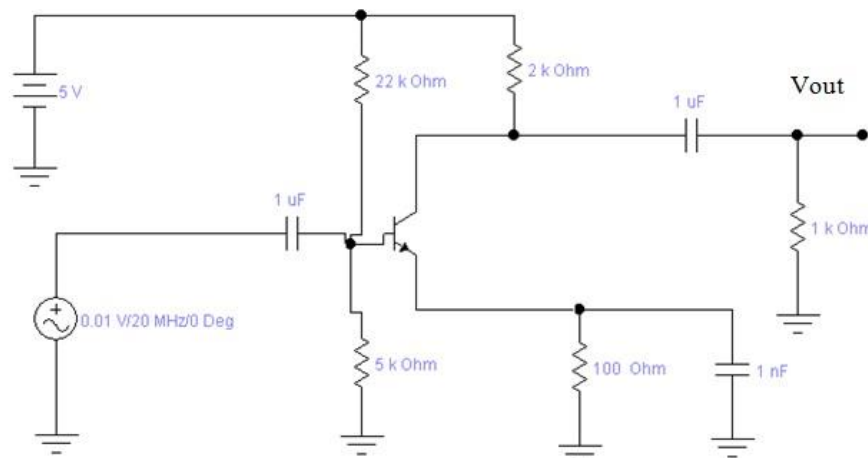


Figure IV - 20 : Power Amplifier circuit

The figures below (Fig: IV- 21, 22) illustrate the time required for computing the moments in order to construct the orthogonal base V

The first figure shows the numerical time complexity for computing the moments for different Sampling rates, and the second figure shows difference between the analytical and the numerical one.

As we can see that the behavior is more or less equivalent, except that the scale of the numerical time is greater than the analytical, due to the fitting algorithm explained earlier in this section. Since the fitting algorithm computes the Chebyshev polynomials for every sample of time in the vector \hat{g}_i . then the time required to approximate the moments numerically depends directly on the number of the Sampling in the circuit, as illustrated in figure IV-21

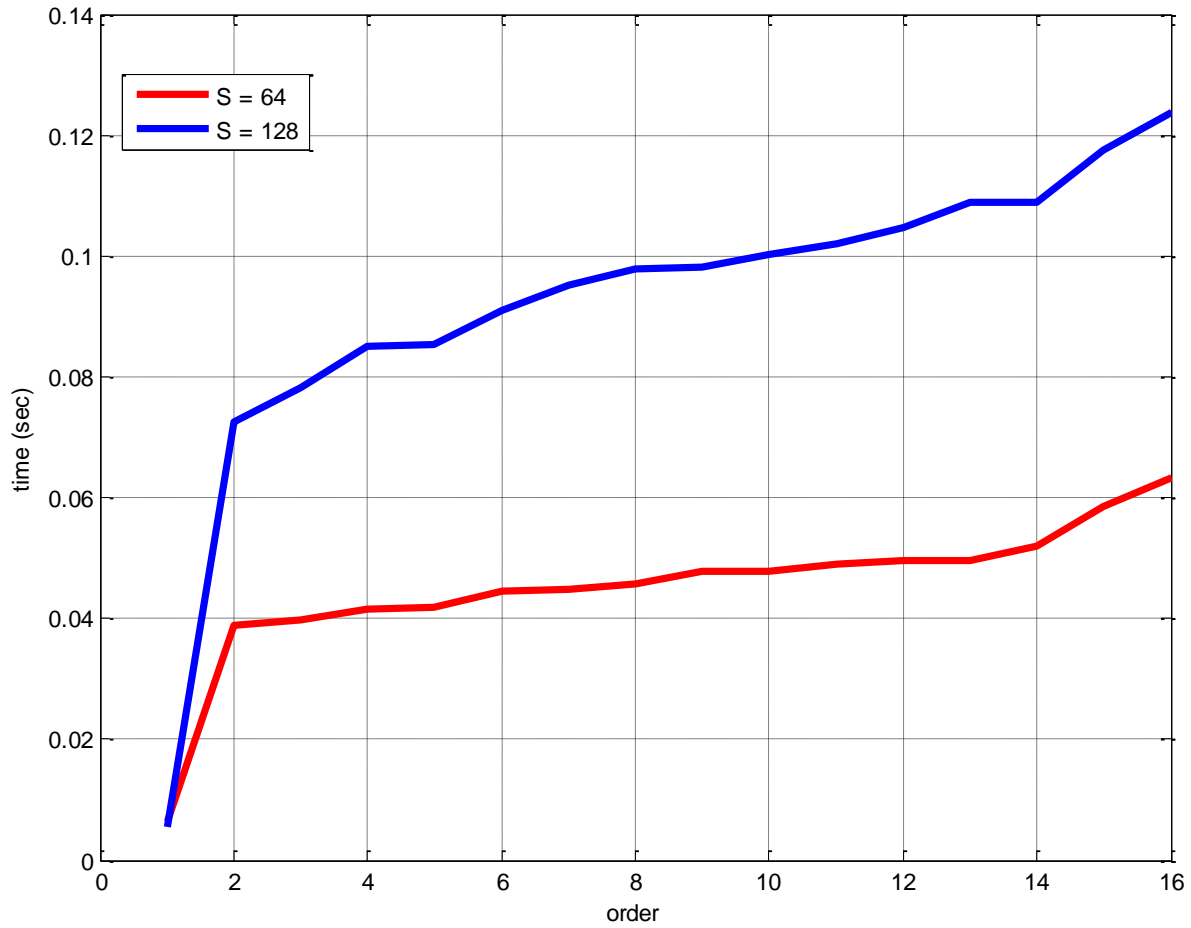


Figure IV - 21 :Time required to analytically compute the Moments of the Power Amplifier circuit

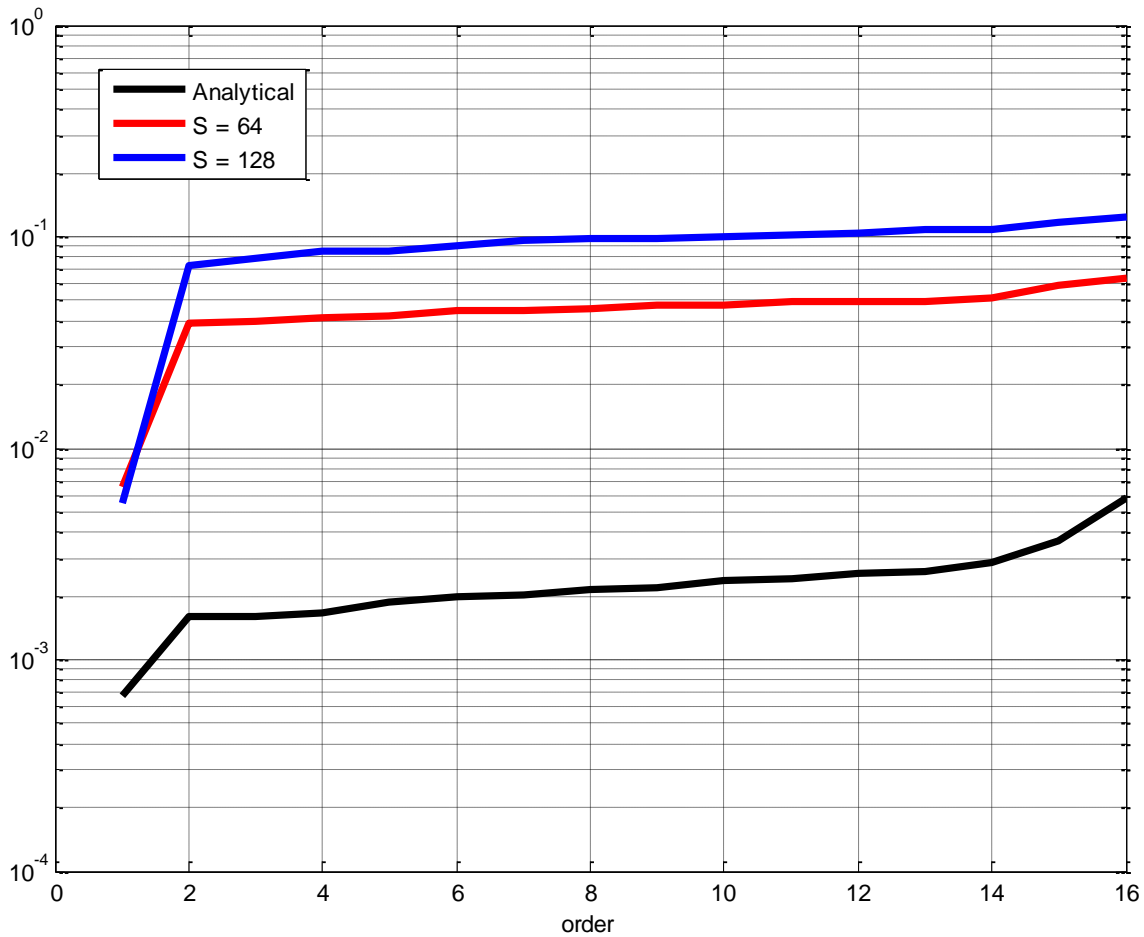


Figure IV - 22 :Computational complexity of evaluation of the Moments for a Power Amplifier circuit

Table IV - 5 shows that the numerical differentiator reached good results, because it managed to help reducing the system and solving it almost same time as the analytical algorithm, however, the time required to construct the orthonormal base, V , using the numerical differential is greater than the analytical method.

		<i>Number of iterations</i>	<i>Time required to solve for steady state solution (sec)</i>	<i>times required to construct the congruent transformation matrix (sec)</i>	<i>Total time (sec)</i>
<i>original HB</i>	<i>S = 32</i>	<i>99</i>	<i>3.25</i>	<i>0</i>	<i>3.25</i>
	<i>S = 64</i>	<i>99</i>	<i>3.28</i>	<i>0</i>	<i>3.28</i>
	<i>S =</i>	<i>99</i>	<i>3.69</i>	<i>0</i>	<i>3.69</i>

	<i>128</i>				
<i>Analytical MOR HB</i>	<i>S=32</i>	<i>50</i>	<i>0.51</i>	<i>0.154186</i>	<i>0.664186</i>
	<i>S=64</i>	<i>50</i>	<i>0.51</i>	<i>0.190224</i>	<i>0.700224</i>
	<i>S=128</i>	<i>50</i>	<i>0.52</i>	<i>0.204469</i>	<i>0.724469</i>
<i>numerical MOR HB</i>	<i>S=32</i>	<i>51</i>	<i>0.51</i>	<i>2.108918</i>	<i>2.618918</i>
	<i>S=64</i>	<i>51</i>	<i>0.52</i>	<i>3.595621</i>	<i>4.115621</i>
	<i>S=128</i>	<i>51</i>	<i>0.525</i>	<i>7.457201</i>	<i>7.982201</i>

Table IV - 5 : Computational time required to solve HB using analytical and numerical proposed method

Figures below (Fig IV – 23, 24) illustrate the correctness of the numerical differentiator by presenting the solution of the second and third harmonic to be matching the response of the original and the MOR technique .

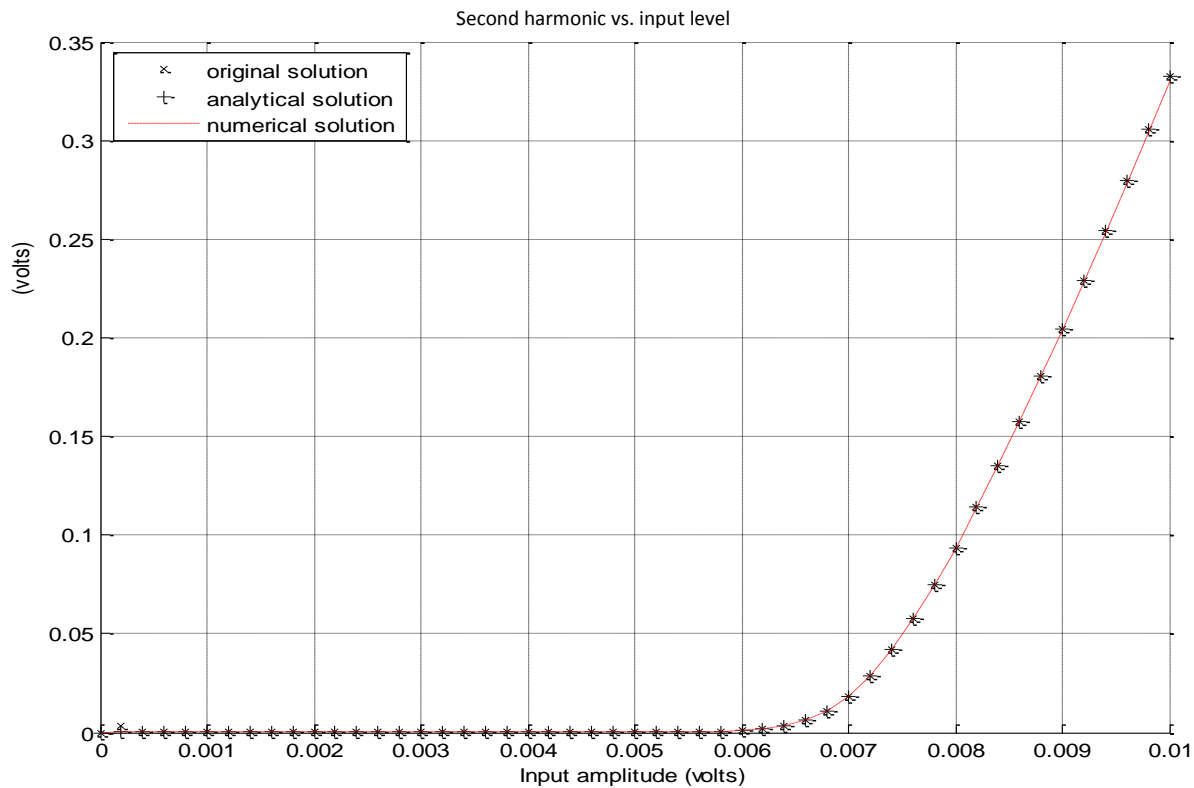


Figure IV - 23 : Solution of the second harmonic vs input level of power amplifier

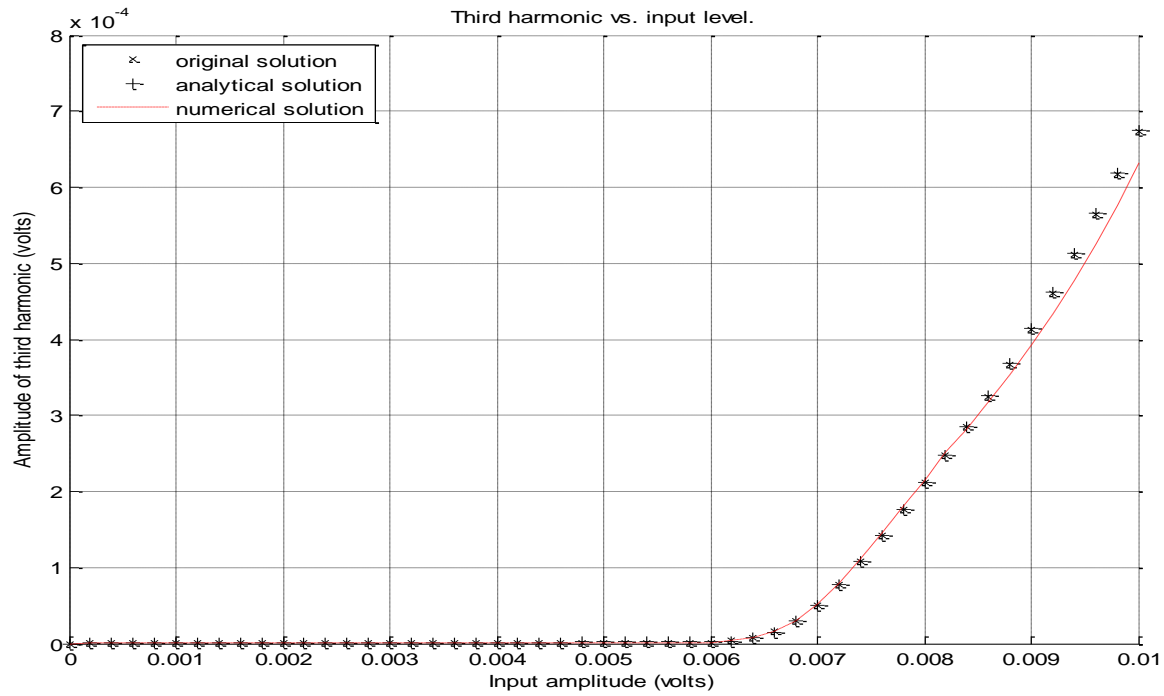


Figure IV - 24 : Solution of the third harmonic vs input level of power amplifier

V. Discussion

In this chapter we first introduced the model order reduction method that was proposed by Gad for solving harmonic balance nonlinear equation via Krylov sub-space. This method showed that once the nonlinear equations in the system are explicitly defined, and there exist a mathematical equivalent form characterized in a power series, it manages to heavily reduce the system while keeping a very good precision of the output in comparison to the original system. However, once the mathematical equivalent is not predefined, this method can no longer proceed with constructing the transformation matrix \mathbf{V} , therefore, this limitation makes the proposed MOR method inapplicable in real life.

In the second part of this chapter, we proposed a numerical differentiator algorithm that can overcome the limitation stated above, therefore, if the mathematical form of nonlinear equations in the system is not predefined, we still can construct the orthonormal base, \mathbf{V} , the numerical algorithm is based on least square method, Chebyshev Polynomials precisely. The main advantage of the numerical algorithm is that it overcomes the limitation that the analytical algorithm faced, while keeping a good precision of the output. However, this method faces some disadvantages it terms of computational cost, because it requires more time to construct matrix \mathbf{V} in comparison with the analytical algorithm, also, it has dependencies on the factor $\hat{\mathbf{D}}$ which is the order of the derivatives required to construct the orthonormal base \mathbf{V} .

However, criteria of finding an optimal $\hat{\mathbf{D}}$ for certain system is not obvious, therefore only trial and error works in this matter. Also, the proposed numerical algorithm faces another dependency, in fact it depends on the variation of the artificial factor α introduced in the nonlinear equations, this factor was always set to zero in the analytical algorithm, that's why it didn't affect the final precision, however, in the numerical differentiator algorithm, this factor has to have a real value very close to zero, therefore, it is not obvious as well which value could be optimal for a system being simulated. Therefore, we have two factors to manipulate before we come to the optimal solution.

Another disadvantage is that as $\hat{\mathbf{D}}$ increases, the numerical error increases too, therefore, the numerical method is limited to a certain margin of variation of $\hat{\mathbf{D}}$ before the error becomes significant.

The important contribution of this work was to show that the HB orthonormal projection base can be constructed using finite difference differentiation, ensure convergence and preserve accuracy the final solution. This then can overcome the crucial limitation of the analytical method, that is the exponential growth of polynomial coefficients when factoring multiple-variable device equations. Further work must be carried on the algorithm in order to reduce error as the order of derivatives increase, and computation cost. In particular the numerical differentiation algorithm developed inherently a parallel task that can efficiently benefit multithreading to speed up the computation.

VI. References:

- [1] K. Kundert, J. White, and A. Sanjiovanni-Vincentilli, “*Steady State Methods for simulating Analog and Microwave Circuits,*” Morwell, MA: Kluwer, 1990.
- [2] M.S. Nakhla and J. Vlach, “*A piecewise harmonic balance technique for the determination of the periodic response of nonlinear systems,*” IEEE Trans. Circuits Syst., vol. CAS-23, pp. 85-91, Feb. 1976.
- [3] S. Maas, “*Nonlinear Microwave Circuits,*” Norwood, MA: Artech House, 1988.
- [4] P. Feldmann, B. Melville, D. Long, “*Efficient Frequency Domain Analysis of Large Nonlinear Analog Circuits,*” IEEE Custom Integrated Circuits Conference, 1996, P. 461-464
- [5] Y. Saad, “*Iterative Methods for Sparse Linear Systems,*” Boston, MA: ITP, 1996.
- [6] E. Gad and M. Nakhla, “*A Circuit Reduction Technique for Finding the Steady-State solution of Nonlinear Circuits,*” IEEE Transactions on Microwave theory and techniques, Vol, 48, NO. 12, Dec 2000
- [7] E. Gad, “*Circuit Simulation techniques via generalized krylov subspaces methods,*” 2003.
- [8] E. Gad, , R. Khazaka, M. Nakhla, R. Griffith, “*A circuit reduction technique for finding the steady-state solution of nonlinear circuits,*” IEEE Trans. Microwave Theory Tech, Vol 48, No 12, p 2389-2396. Dec, 2000
- [9] R. Telichevesky, K. Kundert, I. Elfadel, J. White, “*Fast Simulation Algorithm for RF Circuits,*” CICC’96, 1996.
- [10] J. Vlach and K. Singhal, “*Computer Methods for Circuit Analysis and Design,*” New York: Van Nostrand Reinolds, 1983.
- [11] www.mathworks.com
- [12] E. Gad and M. Nakhla, “*Model reduction for the DC solution of large nonlinear circuits,*” in Proc. ICCAD’99, pp. 376-379.
- [13] P. Gunupudi, M. Nakhla, “*Nonlinear Circuit-Reduction of High-Speed Interconnect Networks Using Congruent Transformation Techniques,*” IEEE Transactions on advanced packaging, vol. 24, No. 3, Aug 2001
- [14] P. Praveen, E. Gad, R. Achar, M. S. Nakhla, & R. Khazaka, “*A projection based reduction approach to computing sensitivity of steady-state response of nonlinear circuits,*” Informs journal on Computing, vol 18, No-2 pp 173-185, 2006
- [15] A. Bjorck, “*Numerical Methods for Least Squares Problems,*” SIAM, 1996

- [16] B. N. Ryland, H. Z. Munthe-Kaas, “*On Multivariate Chebyshev Polynomials and Spectral Approximations on Triangles*,” Spectral and High Order Methods for Partial Differential Equations, Lecture Notes in Computational Science and Engineering, 76, Springer-Verlag, Berlin (2011), 19-41.
- [17] E.L. Allgower, K. Gerog “*Introduction to numerical continuation methods*,” Colorado, 1997
- [18] G. Harvey, “*Computer Architecture and Implementation*,” Cabridge University Press 2000, 2000
- [19] P. Gunupudi and M. Nakhla, “*Model-Reduction of Nonliear Circuits Using Krylov-Spral Techniques*,” DAC pp 13-16, 1999
- [20] G. H. Golub and C. Van Loan, “*Matrix Computations*,” Baltimore, MD: John Hopkins Univ. Press, 1989
- [21] Vladimir Ceperic, “*Black-Box Modelling of Mixed-Signal Integrated Circuits*,” Faculty of Electrical Engineering and Computing, University of Zagreb.
- [22] A.A. Puntambekar, “*Advanced Data Structures and Algorithms*,” technical publications Pune, 2008

Chapitre V :Conclusion and Future Scope

I. Summary

Recently, the model order reduction domain has been widely introduced to circuit simulation due to the achievement and effectiveness it showed, in terms of decreasing the computational cost of CAD tools, by saving time and resources of designers while keeping good precision of the output.

Model order reduction has two main fields, linear and nonlinear MOR. Linear MOR is a well-established domain already that contains a numerous techniques that proved correctness and effectiveness, however, when it comes to nonlinear techniques, nonlinear MOR is still considered to be immature field because of the difficulties and limitations it faces, moreover it was limited in applications where there are large linear blocks with few nonlinear terminations.

Gad has recently tackled this problem by proposing an algorithm of nonlinear model order reduction for large nonlinear systems, using the projection techniques based on Krylov-subspace methods. This technique aims to create an orthogonal base using the first few derivatives of the nonlinear equations in the system. This technique was applied to harmonic balance simulation method to find the steady-state solution of large system; this is done by projecting the large set of nonlinear equations of HB into a much smaller subspace, then solving for the steady state solution using Newton Raphson recurrence numerical method, and once the solution is found, it is mapped back to its original size.

The gist of this technique is that it avoids computing the expensive inverse of Jacobian matrix of large size at every iteration for NR, therefore the orthonormal base is ought to reduce the system, and solve NR in a much smaller size. However, this technique faced a limitation in constructing such orthonormal base, basically, this technique proposes to find an equivalent mathematical form of the n th order derivative of nonlinear equations in the system, and to present them in a power series in order to evaluate the moments that build up the orthogonal base. However, when the mathematical equations in the system are difficult to derive, or not explicitly pre-defined, this method cannot proceed to construct the orthogonal base.

This thesis proposes a numerical differentiator algorithm that helps to overcome the limitation stated above using least square fitting through Chebyshev Polynomials.

The numerical differentiator algorithm does not change the process of how the proposed MOR works; in fact, it is just an alternative method of how to create the orthonormal base when the proposed method fails to do so.

Even though the computational cost of the numerical differentiator is high in comparison to the analytical method, however, the advantage is that it can overcome a crucial limitation while preserving correctness of the output behavior.

II. FutureWork

Although the numerical algorithm managed to compute the nonlinear equation derivatives, however, we see that as the order to derivative increases, the error increases as well, and become rather uncontrollable.

Several points can be investigated in the future:

Finding algorithm that can reduce the numerical error while approximating the Padé moments numerically

To some extent, the proposed numerical differentiator proved the ability of approximating the required orders which are needed to construct the congruent matrix . A recent algorithm based a traditional second order sliding mode controller has been implemented, namely, Super *twisting algorithm* , this approach can be the base of a different approach to build a numerical differentiator for the MOR technique proposed by Gad.

Applying the method of fitting through Chebyshev polynomials to every sample in the nonlinear equations vectors was shown to be very expensive in terms of computation complexity, this cost could be reduced through the use of multi-processing algorithm that can significantly reduce the time spent by the numerical differentiator in order to approximate the Padé moments.

Annex

Chebyshev polynomials :

Among least square methods, the Chebyshev polynomials are chosen because they are considered to be the most important family of orthogonal polynomials [15]. Their properties can be derived by rather simple methods.

Consider the easily verified formula

$$\cos(n+1)\theta + \cos(n-1)\theta = 2\cos\theta \cos(n\theta), \quad n \geq 1$$

The above formula can be used recursively in order to express $\cos(n\theta)$ as a polynomial in $\cos(\theta)$

Consider that

$$\beta = \cos\theta$$

Where $\theta = \arccos\beta$, then we obtain a triangle family of polynomials that are called Chebyshev polynomials, defined for $-1 \leq \beta \leq 1$, by the formula $T_n(\beta) = \cos(n \arccos\beta)$, $n \geq 0$. Where the Chebyshev polynomials satisfy the recursion formula,

$$T_0(\beta) = 1$$

$$T_1(\beta) = \beta$$

$$T_{n+1}(\beta) = 2\beta T_n(\beta) - T_{n-1}(\beta), \quad n \geq 0$$

The leading coefficient of $T_n(\beta)$ is 2^{n-1} for $n \geq 1$ and 1 for $n = 0$. The symmetry property $T_n(-\beta) = (-1)^n T_n(\beta)$ also follows from the recurrence formula.

$T_n(\beta)$ has n zeros in $[-1, 1]$ given by the Chebyshev abscissa

$$\beta_k = \cos\left(\frac{2k+1}{n} \times \frac{\pi}{2}\right), \quad k = 0, 1, \dots, n-1$$

and $n+1$ extrema

$$\beta'_k = \cos\frac{k}{n}\pi, \quad T_n(\beta'_k) = (-1)^k, \quad k = 0, 1, \dots, n.$$

These results follow directly from the fact that $|\cos(n\theta)|$ has maxima for $\theta'_k = \frac{k\pi}{n}$ and $\cos(n\theta_k) = 0$ for

$$\theta_k = \frac{(2k+1)\pi}{2n}$$

The Chebyshev polynomials T_0, T_1, \dots, T_{n-1} are orthogonal with respect to the inner product:

$$(f, g) = \sum_{k=0}^{n-1} f(\beta_k)g(\beta_k)$$

Where $\{\beta_k\}$ are the Chebyshev abscissa for T_n .

$$(T_i, T_j) = \begin{cases} 0, & i \neq j \\ \frac{1}{2}n, & i = j \neq 0 \\ n, & i = j = 0 \end{cases}$$

Finding chebyshev polynomials is done using the following algorithm:

Let $p(\beta)$ denote the interpolation polynomial in the Chebyshev points β_k , then we have

$$p(\beta) = \sum_{j=0}^{n-1} c_j T_j(\beta)$$

$$c_i = \frac{1}{\|T_i\|^2} \sum_{k=0}^{n-1} f(\beta_k) T_i(\beta_k)$$

Where $\|T_i\|^2$ is given in the equation above

Definition I: Chebyshev polynomials are usually used in an interval that goes from -1 to 1. However, If we are working in terms of a parameter t which varies in the interval $[a, b]$, then we should make the substitution, $t = \frac{1}{2}(a + b) + \frac{1}{2}(a - b)x$ where $t \in [a, b] \Leftrightarrow x \in [-1, 1]$ and work with the Chebyshev points

$$t_k = \frac{1}{2}(a + b) + \frac{1}{2}(a - b)x_k, \quad k = 0, 1, \dots, n - 1$$

Resumé

MOR (Model Order Reduction) est devenu un domaine très répondeu dans la recherche grâce à l'intérêt qu'il peut apporter dans la réduction des systèmes, ce qui permet d'économiser du temps, de la mémoire et le coût de CPU pour les outils de CAO. Ce domaine contient principalement deux branches: linéaires et non linéaires. MOR linéaire est un domaine mature avec des techniques numériques bien établie et bien connus dans la domaine de la recherche, par contre le domaine non linéaire reste vague, et jusqu'à présent il n'a pas montré des bons résultats dans la simulation des circuits électriques. La recherche est toujours en cours dans ce domaine, en raison de l'intérêt qu'il peut fournir aux simulateurs contemporains, surtout avec la croissance des puces électroniques en termes de taille et de complexité, et les exigences industrielles vers l'intégration des systèmes sur la même puce.

Une contribution significative, pour résoudre le problème de Harmonic Balance (Equilibrage Harmonique) en utilisant la technique MOR, a été proposé en 2002 par E. Gad et M. Nakhla. La technique a montré une réduction substantielle de la dimension du système, tout en préservant, en sortie, la précision de l'analyse en régime permanent. Cette méthode de MOR utilise la technique de projection par l'intermédiaire de Krylov, et il préserve la passivité du système. Cependant, il souffre de quelques limitations importantes dans la construction de la matrice "pre-conditioner" qui permettrait de réduire le système. La limitation principale est la nécessité d'une factorisation explicite comme une suite numérique de l'équation des dispositifs non linéaires. cette limitation rend la technique difficile à appliquer dans les conditions générales d'un simulateur.

Cette thèse examinera les aspects non linéaires du modèle de réduction pour les équations de bilan harmoniques, et il étudiera les solutions pour surmonter les limitations mentionnées ci-dessus, en particulier en utilisant des approches de dérivateur numériques.

Implementation of Harmonic Balance Reduce Model Order Equation

Abstract

MOR recently became a well-known research field, due to the interest that it shows in reducing the system, which saves time, memory, and CPU cost for CAD tools. This field contains two branches, linear and nonlinear MOR, the linear MOR is a mature domain with well-established theory and numerical techniques. Meanwhile, nonlinear MOR domain is still stammering, and so far it didn't show good and successful results in electrical circuit simulation. Some improvements however started to pop-up recently, and research is still going on this field because of the help that it can give to the contemporary simulators, especially with the growth of the electronic chips in terms of size and complexity due to industrial demands towards integrating systems on the same chip.

A significant contribution in the MOR technique of HB solution has been proposed a decade ago by E. Gad and M. Nakhla. The technique has shown to provide a substantial system dimension reduction while preserving the precision of the output in steady state analysis. This MOR method uses the technique of projection via Krylov, and it preserves the passivity of the system. However, it suffers a number of important limitations in the construction of the pre-conditioner matrix which is ought to reduce the system. The main limitation is the necessity for explicit factorization as a power series of the equation of the nonlinear devices. This makes the technique difficult to apply in general purpose simulator conditions.

This thesis will review the aspects of the nonlinear model order reduction technique for harmonic balance equations, and it will study solutions to overcome the above mentioned limitations, in particular using numerical differentiation approaches