TIME-DOMAIN STEADY-STATE ANALYSIS OF CIRCUITS WITH MULTIPLE ADAPTIVE GRIDS

BY

ZHOU WANG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

CONTROL ENGINEERING

LAKEHEAD UNIVERSITY

2005

Your file Votre référence
ISBN: 978-0-494-15646-9
Our file Notre référence
ISBN: 978-0-494-15646-9

Abstract

An effective adaptive algorithm to calculate the steady-state response of circuits is developed. This algorithm offers a powerful alternative to traditional steady-state simulation techniques, such as the shooting method or harmonic balance (HB). A favourable feature of the algorithm is that it obtains the unknown circuit solutions by using adaptive basis functions (ABF). The circuit equations are formulated by transformation matrices. One of the contributions of this thesis is to use the least squares method instead of Galerkin method to solve ordinary differential equations with ABF. Another contribution is that the proposed algorithm uses different grid resolutions to represent each state variable simultaneously. The position of the grid points for each state variable is adaptively controlled by an algorithm that attempts to minimize artifact oscillations in the solutions. The algorithm is demonstrated by simulating two circuits and comparing the results with Spice and Aplac.

Acknowledgements

I would like to take this opportunity to extend my deep gratitude to my supervisor Dr. Carlos E. Christoffersen, I have had the pleasure of learning from him over the past two years and appreciate for awarding me the chance to pursue my research interest. His superb guidance and support have encouraged me a lot and incite my great passion in the research.

I would like to thank my co-supervisor Professor K. Natarajan for his suggestions and profound ideas. I am also grateful to other professors in Engineering Department for their assistance throughout the thesis work.

I want to thank all graduate students for their companionship, co-operation and for being good friends. Special thanks to all former classmates who have been very supportive of my decision to continue my academics. Last but not least, I am greatly indebted to my family for their understanding, patience and support during the entire period of my study.

Zhou Wang

zwang1@lakeheadu.ca

Contents

List of Figures

List of Tables

List of Symbols

t – Time
N – An integer
$v(t)$ – A state variable
$\bar{v}(t)$ – The approximation of a state variable
$\hat{V}$ – An coefficient vector for approximation of a state variable
$\hat{v}_i$ – An individual coefficient in the vector $\hat{V}$
U – A vector of sample points for many state variables
U_i – A vector of sample points for a state variable
$\hat{U}$ – A coefficient vector for many state variables
$\hat{U}_i$ – A coefficient vector for a state variable
u_{span} – The total span of sample points
$\mathbf{u}$ – A vector of nodal voltages
$f()$ – A given function
$g()$ – Another given function
$b(t)$ – Another given function
$X_1(t)$ – Another given function
$X_2(t)$ – Another given function
$\psi_t()$ – Weighting function
$\gamma()$ – Residual function
A – A matrix
$\mathbf{b}$ – A vector
ϕ – A nonlinear function
$v_n(t)$ – Voltage in the time domain
V_m – The m^{th} harmonic of voltage
ω – Angular frequency
ϕ_m – Phase angle
$\psi_{j,l}(x)$ – Wavelet function
$\phi_{j,l}(x)$ – Scaling function

j – The index of points
t_j – A discrete point
Δt_j – An interval of two adjacent discrete points
$\varphi_i(t)$ – Basis function
$\varphi_{2j}(t)$ – One form of basis function
$\varphi_{2j+1}(t)$ – Another form of basis function
n – The number of intervals in one period
m – The number of state variables in the circuit
B – A transformation matrix for a state variable
B_p – Another transformation matrix for a state variable
k – The index of grid points
t_k – A grid point
$\triangle t_k$ – An interval of two adjacent grid points
G – A matrix representing the linear conductance terms
C – A matrix representing the linear charge terms
$I(t)$ – A function representing the nonlinear algebraic terms
$\dot{Q}(t)$ – A function representing the nonlinear charge terms
$S(t)$ – An independent source vector
I_{NL} – Current vector in the frequency domain
$I_(u^j)$ – A nonlinear algebraic term at a sample point
$I(u^{j+1})$ – A nonlinear algebraic term at the next sample point
J_I – The Jacobian of the function $I(u^j)$
J_Q – The Jacobian of the function $Q(u^j)$
q – Nonlinear charge
M – An identity matrix
$\bar{M}$ – A rectangular matrix
M_q – An identity matrix associated with the dimension of q
P_j – Area residual
l_k – The length of a linear interpolation
$\bar{P}$ – The mean of irregular area residuals
$\bar{l}$ – The mean of linear interpolations
s – Scaling factor
F – The termination number
$\mathscr{W}$ – An integration form
$\Re$ – Real part
$\mathscr{F}$ – Fourier transformation operator
$\mathscr{F}^{-1}$ – Inverse Fourier transformation operator

HB - Harmonic balance
FFT - Fast Fourier transform
CAD - Computer aided design
R - Resistor
C - Capacitor
L - Inductor
Z - Impedance
Y - Y parameter of a network
S - Scattering parameter
f - Frequency
h_{FE} - Typical DC current gain

Chapter 1

Introduction

Circuit analysis can be divided in steady state or transient analysis. This thesis focus in the steady-state analysis driven by periodic inputs because the steady-state is the main concern in many practical applications. Examples are the design of power supplies, frequency multipliers, and amplifiers.

As the literature suggests [5, 31, 16], much previous work has been done to determine efficient algorithms for computing steady-state. Some examples are the harmonic balance (HB) method [23, 25] and wavelet-based method [35, 29].

In this thesis, we investigate a multiple adaptive algorithm working in time domain, that has significant advantages with respect to conventional methods. This algorithm potentially has a lot of extended applications to handle linear or nonlinear circuits which are driven by periodic excitations.

1.1 Objective

The main objective of this thesis is to find a fast steady-state analysis algorithm that allows an independent adaptive grid for each state variable in the circuit. The grids can be adapted to the shape of the waveforms in the circuit. Thus, fewer grid points are needed and that leads to computational saving. To the author's best knowledge, an algorithm with similar capabilities has not been demonstrated in the literature.

A major challenge is how to best balance the demands of accuracy and speed requirements especially when dealing with large nonlinearity circuits.

1.2 Scope and Structure of the Thesis

The main topics that describe the proposed algorithm are as follows:
(1) Approximation with cubic splines and transformation matrices
(2) Formulation of circuit equations
(3) Least squares method to acquire numerical solutions
(4) Grids adaptation

The first part intend[illegible]e the fundamental elements involved in the approximation of functions usi[illegible] cubic splines. Part 2 and 3 show the complete derivation of equations for a gener[illegible] circuit and a numerical approach is used to solve the differential equations. The fourth part explores a grid position control strategy.

As shown in Fig. 1.1, the algorithm starts by building matrices for every state variable with uniform grids , which are discussed in Section 3.2.1.

Secondly, reduce algebraic differential equations to algebraic equations and solve them by least squares method. As a result, the unknown coefficients are determined and the circuit solutions are obtained.

Finally, to achieve high accuracy with less unknowns and without increasing the computational effort, the design of grids adaptation is necessary as discussed in Section 3.3.

Two concrete examples, a RC circuit and a feedback amplifier circuit with periodic excitations, are provided in Chapter 4 to evaluate the algorithm effectiveness. The results are compared to PSpice or APLAC simulations.

The conclusion is drawn in Chapter 5 followed by the direction of future work.

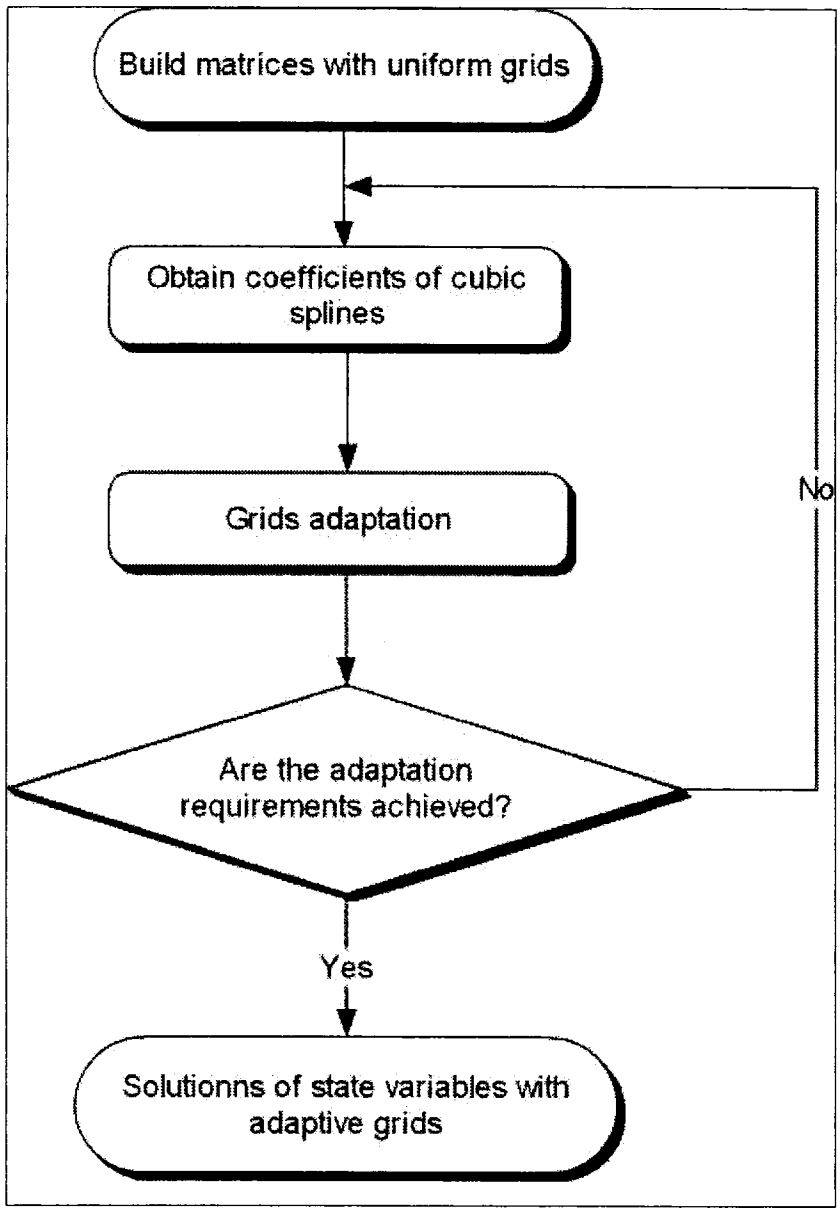

Figure 1.1: Flowchart of designing multiple adaptive algorithm

Chapter 2

Literature Review

The standard description of electrical circuits is in terms of differential equations. For solving the differential equations of a circuit, analytical solutions are impossible in most practical cases. Therefore, it is wise to apply some numerical techniques to acquire the circuit solutions (voltages and currents) as described in the first part in this chapter.

The related simulation algorithms are addressed in Section 2.2. The simulation procedure contains two principle steps: In the first step, the unknown functions in differential equations are replaced by finite element approximations according to selected basis functions. In the second step, a set of formulated algebraic equations are solved by the least squares method. Section 2.3 reviews the basis functions representation followed by a description of the grids adaptation.

2.1 Numerical Solutions of Differential Equations

Some commonly applied techniques include Galerkin method [13], collocation method [15] and least square fitting [26]. Each regression method serves the same purpose of minimizing the residual function γ shown as Eq. (2.5) either at a set of points or by means of integration.

Suppose a state variable $v(t)$ is approximated by a basis function series

$$v(t) \simeq \bar{v}(t) = \sum_{i=0}^{N} \varphi_i(t) \cdot \hat{v}_i \tag{2.1}$$

$$\dot{v}(t) \simeq \dot{\bar{v}}(t) = \sum_{i=0}^{N} \dot{\varphi}_i(t) \cdot \hat{v}_i \tag{2.2}$$

where, $v(t)$: a state variable

$\bar{v}(t)$: the approximation form of a state variable

$\varphi_i(t)$: a basis function

$\hat{v}_i$: one element in a coefficient vector of $\hat{V}$ and $\hat{V} = [\hat{v}_1, \hat{v}_2, ...\hat{v}_i, ...\hat{v}_N]^T$

N: an integer

A state variable $v(t)$ from a given ordinary differential equation is then approximated by the summation of a set of products of basis functions $\varphi_i(t)$ and their coefficients $\hat{v}_i$. The derivative $\dot{v}(t)$ is no longer solved by differential equation but obtained by a numerical summation of basis function series.

Let a circuit be described by the following state equation

$$\dot{v}(t) = f(v(t), t) \tag{2.3}$$

It follows that

$$f(v, t) - \dot{v}(t) = 0 \tag{2.4}$$

Define a residual function γ as

$$\gamma(\hat{V}, t) = f(\bar{v}, t) - \dot{v}(t) \simeq 0 \tag{2.5}$$

2.1.1 Galerkin Method

Galerkin method can be used to analyze the steady-state of a nonlinear circuit. The residual γ is orthogonal to the space spanned by the basis functions.

Two functions $X_1(t)$ and $X_2(t)$ are orthogonal implying that they carry independent information, that is,

$$\int_{-\infty}^{+\infty} X_1(t) \cdot X_2(t)dt = 0 \tag{2.6}$$

Choose N linearly independent weighting functions ψ_t as

$$\psi_t = \frac{\partial \bar{v}(t, \hat{v}_i)}{\partial \hat{v}_i} = \varphi_i(t) \tag{2.7}$$

Hence, in Galerkin method, the weighting functions are selected in the form of basis functions.

Next, make $X_1(t)$ refer to ψ_t, and $X_2(t)$ refer to residual funtion γ. Then by Eq. (2.6), we define a integral form $\mathscr{W}$ and minimize it to be zero as Eq. (2.8), leading to a set of nonlinear equations for solving unknown coefficients $\hat{V}$. Integrals can be calculated numerically.

$$\mathscr{W} = \int_0^T \varphi_i(t) \cdot \gamma(\hat{V}, t) = 0 \tag{2.8}$$

Galerkin method has many applications in circuit simulation algorithms. For example, adopting the Galerkin approach, the HB system of algebraic equations can be obtained by using the harmonic basis functions.

2.1.2 Collocation Method

In this method, the differential equations are satisfied exactly at a set of collocation points. The residual function $\gamma(t_j)$ is made to zero at a set of points t_j. And this method requires to establish a system of equations with as many unknowns as equations.

$$\gamma(\bar{v}, t_j) = 0 \tag{2.9}$$

2.1.3 Least Squares Method

In contrast to collocation method, the least squares method requires greater number of equations than the number of unknown parameters. Those equations should be formed at a set of sample points t_j. The target of this method is to find $\hat{V}$ such that $\sum_{i=0}^{N}(\gamma(\hat{V}, t_j))^2$ is minimum.

In particular, we analyze the linear case. Suppose $v(t)$ satisfies the following state equation:

$$\dot{v}(t) = g(t)v(t) + b(t) \tag{2.10}$$

where $g(t)$ and $b(t)$ are two given functions. The residual function is then given by

$$\gamma(t) = g(t)\bar{v}(t) + b(t) - \dot{\bar{v}}(t) \tag{2.11}$$

Then,

$$\gamma(\hat{V}, t) = \sum_{i=1}^{N}[g(t)\varphi_i(t)\hat{v}_i - \dot{\varphi}_i(t)\hat{v}_i] + b(t) \tag{2.12}$$

Consider now a set of sample points $[t_1, t_2, ..., t_j, ...]$. If we apply Eq. (2.12) for each point

in the set and $\gamma = 0$, we obtain the following system of linear equations:

$$A\hat{V} + \mathbf{b} = 0 \tag{2.13}$$

where A is a known matrix, $\hat{V}$ is a vector with unknown parameters, $\hat{V} = [\hat{v}_1, \hat{v}_2, ...\hat{v}_i, ...\hat{v}_N]^T$, and $\mathbf{b}$ is a known vector.

Eq. (2.13) is over determined, because there are more sample points than unknowns. If the least squares approach is used, the coefficients can be found using the following equation [17],

$$\underbrace{A^T A}_{nonsingular} \hat{V} = -A^T \mathbf{b} \tag{2.14}$$

2.2 Steady-state Simulation Algorithms

The most widely used methods for steady-state analysis of circuits [15, 19, 22] are the shooting method [30, 18], harmonic balance [23, 4, 12, 36] and more recently developed wavelet-based methods [8, 25].

Suppose that a circuit is modeled by an ordinary differential equation:

$$\dot{v}(t) = f(v(t), t) \tag{2.15}$$

where, the input period T is embedded into the function $f(v(t), t)$.

2.2.1 Shooting Method

Introduce a nonlinear function ϕ [1] such that

$$\phi = v(t_0) + \int_{t_0}^{t_0+T} f(v(\tau), \tau) d\tau \tag{2.16}$$

where, t_0 is an initial time instant

The idea of shooting method is straightforward. For a given initial value $v(t_0)$, determine the value of ϕ in the subsequent time instant $t_0 + T$. Trying to find the specific initial value $v(t_0')$ which satisfies

$$v(t_0') = \phi \tag{2.17}$$

Unfortunately, one of shortcomings of this method is the sensitivity of selecting t_0. If it is chosen away from t_0', the accumulated difference between ϕ and $v(t_0)$ is saved to correct t_0, thus t_0 must be re-chosen as another starting guess value for ϕ calculation at next time until the Eq. (2.17) is satisfied.

2.2.2 Wavelets

Another available approach is wavelets, which draw the great attention in recent years. The wavelet method can be refined to obtain the steady-state circuit solutions [29].

The idea of wavelets is to expand the unknown function $f(t)$ into a set of equations by wavelet series and determine the corresponding expansion coefficients. The construction of wavelet series starts from the wavelet function (mother function) $\psi_{j,l}(x)$, where j the wavelet level ($j = 0, 1, ...$), l is the position index ($l = 0, 1, ..., 2^j$). The wavelet level controls the degree of localization while the position index affects the position of the center of the wavelet.

Depending on the choice of the wavelet function, there exists a large selection of wavelet families, such as Daubechies family. The optimal choice of the wavelet basis will depend on the application at hand. Wavelet function should satisfy $\int_{-\infty}^{+\infty} \psi_{j,l}(x)dt = 0$ and it can be manipulated by dilation and translation as Fig. 2.1 and 2.2 in form of $\psi_{j,l}(at - b)$ in order to satisfy the requirement of different resolutions, where a is dilation parameter and b is translation parameter. Another necessary function to construct wavelets is the scaling function (father function) $\phi_{j,l}(x)$. By introducing the scaling function we can limit wavelets from the infinite to finite numbers [6].

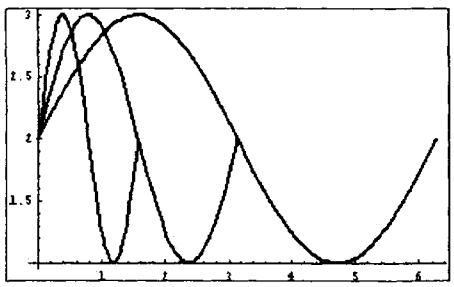

Figure 2.1: Dilation of wavelet function

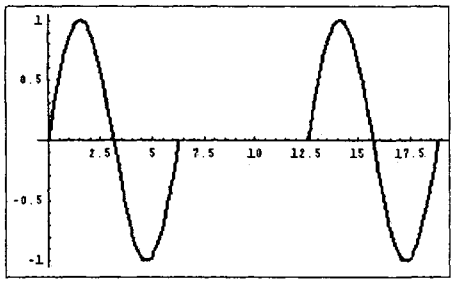

Figure 2.2: Translation of wavelet function

One of the main properties in the wavelet approach is to represent the circuit solutions by multi-resolution. When using wavelets as the basis functions to approximate a given function, we can increase the number of wavelets at the 'needed location' and go to any desired high resolution . Theoretically , we can use wavlets to achieve any high resolutions at any physical location.

Concisely, by means of the wavelet and scaling function, we build up circuit equations and solve the unknown wavelet coefficients. Then the circuit solutions are obtained by the series of wavelets and their coefficients.

Recently, a wavelet collocation method is proposed in the publication [11] for solving the circuit equations. The basic concept is to obtain the unknown variables $v(t)$ by manipulating the certain wavelet basis functions, and in terms of preset collocation points to determine the wavelet coefficients. Another improved wavelet method works in both frequency and time domain is presented in the reference [35].

The wavelet method is potentially able to simultaneously acquire all state variables with independent resolutions, but there are no publications to demonstrate that so far.

2.2.3 Harmonic Balance

The widespread HB [28] method is a mixed-domain, steady-state simulation technique. Suppose that a result at the n^{th} node with s harmonics [23] is

$$v_n(t) = \Re\{\sum_{m=0}^{s} V_m exp(j\omega mt + \varphi_m)\} \tag{2.18}$$

This represents a function in terms of sine waves with frequencies which are multiples (harmonics) of the basic frequency, with different phases. The sine waves used are orthogonal to each other. Thus Garlerkin method can be used to discretize the governing equations in HB method.

In this approach, the circuit first must be separated into a linear and a non-linear part, as shown in Fig. 2.3 and Flowchart 2.4 , resulting in corresponding frequency-domain and time-domain descriptions.

The linear part is modeled by two trans-admittance matrices: the first one $\tilde{Y}_{N\times M}$ relates the source voltages $v_{s1}, ..., v_{sm}$ to the interconnection currents $i_1, ..., i_N$ and the second one $\hat{Y}_{N\times N}$ relates the interconnection voltages $v_1, ..., v_N$ to the interconnection currents $i_1, ..., i_N$. Z_i are internal impedances of the voltage sources in the linear part. Since V_s is a known and

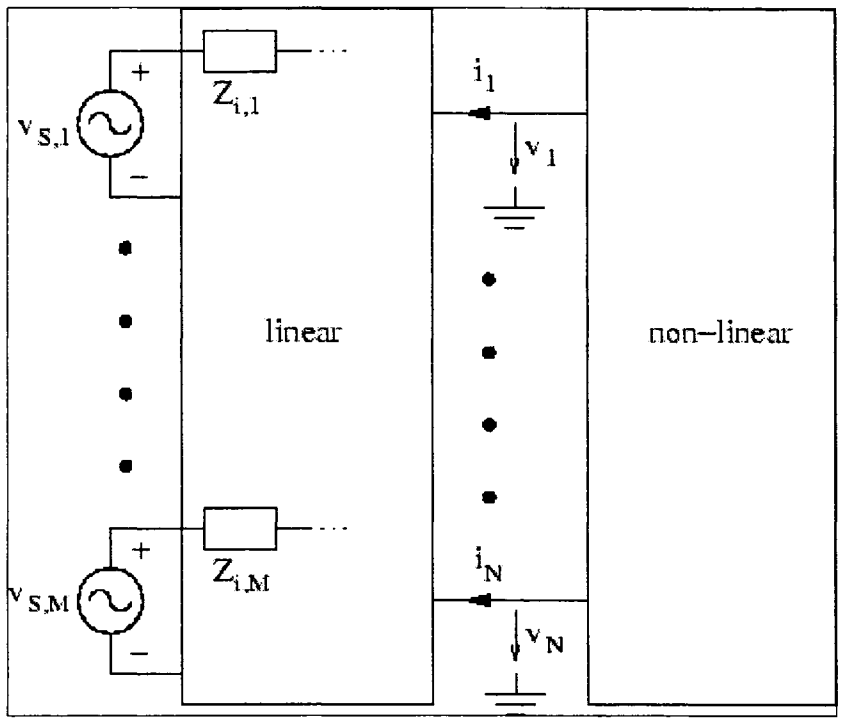

Figure 2.3: Circuit partitioning in harmonic balance

constant vector, the first item is written as I_s.

$$I = \tilde{Y}_{N\times M} \cdot V_s + \hat{Y}_{N\times N} \cdot V = I_s + \hat{Y}_{N\times N} \cdot V \tag{2.19}$$

On the other hand, the nonlinear part can only be modeled in the time domain. Assume the non-linear element is modeled by its current function $i(t) = f_{NL}(v_1, ..., v_p)$ and by the charge of its capacitances $q(t) = f_{NL}(v_1, ..., v_Q)$. The corresponding frequency-domain vectors Q and I_{NL}, respectively are obtained by means of the FFT.

Thus, the non-linear equation system is written as

$$F(V) = \underbrace{I_s + \hat{Y}_{N\times N} \cdot V}_{linear} \underbrace{- j \cdot \Omega \cdot Q - I_{NL}}_{nonlinear} = 0 \tag{2.20}$$

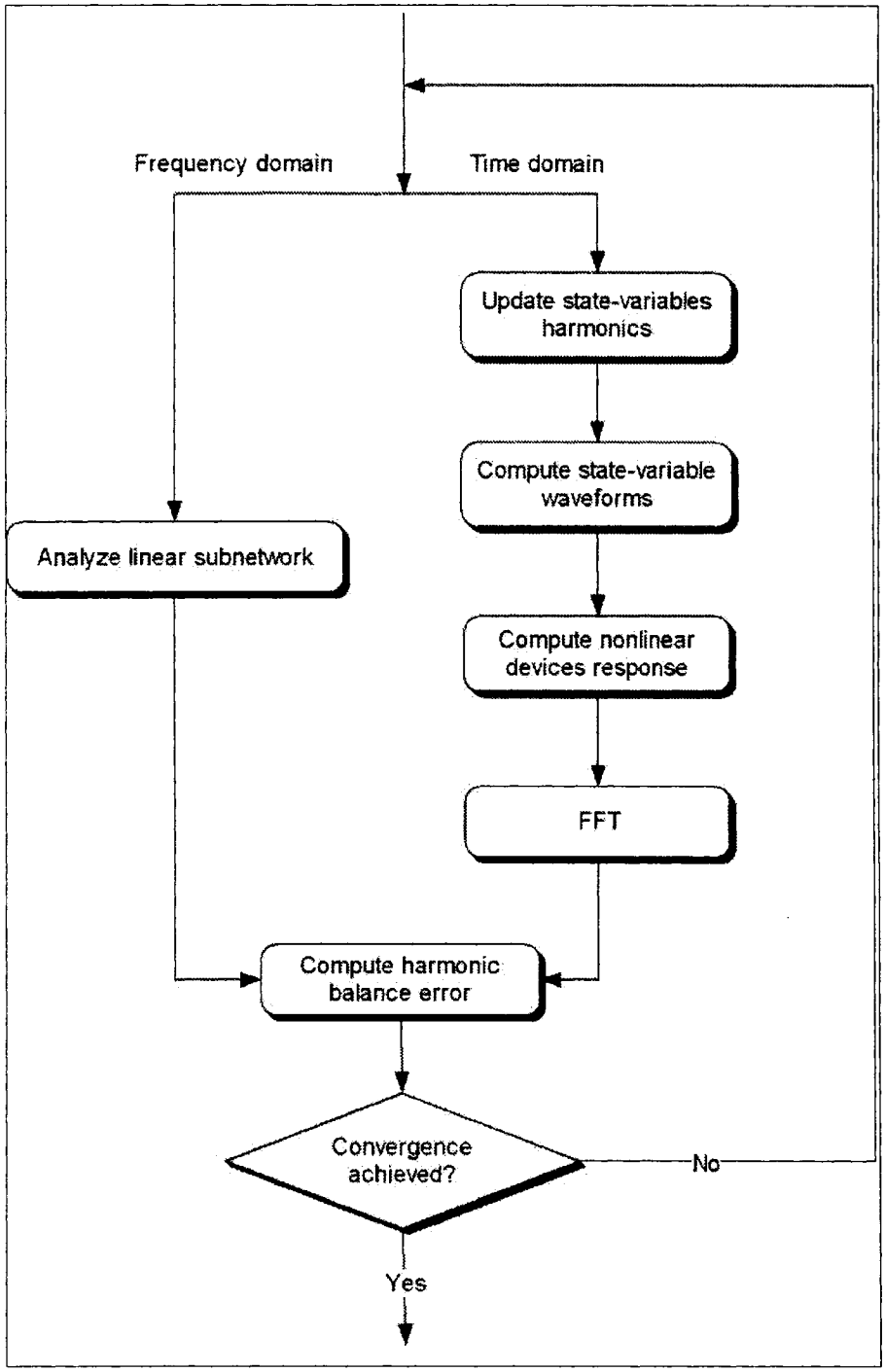

Figure 2.4: Flowchart of harmonic balance analysis

where Ω is a diagonal matrix containing the angular frequencies on the main diagonal. 0 represents zero matrix.

After each Newton-Raphson iteration step, the inverse fast Fourier transformation (IFFT) must be applied to the voltage vector V, that is

$$I_{NL}(V) = \Omega \cdot \mathscr{F}\{q[\mathscr{F}^{-1}(V)]\} + \mathscr{F}\{i[\mathscr{F}(V)]\} \tag{2.21}$$

Once the currents through the interconnections are the same for the linear and the non-linear subcircuits, the simulation results are obtained.

The obvious merit of this approach is its efficiency to directly solve linear components circuits excited with multi-tone sources especially at high frequency. This method is appropriate for most microwave circuits excited with sinusoidal signals (e.g. mixers, power amplifiers). However, HB suffers from limitations with strong nonlinear components, such as the microwave diode, because a large number of variables must be optimized along with forward or reverse FFT, resulting in a dense Jacobian matrix and expensive time costs. To deal with this problem, various time-domain methods are usually employed to treat the non-linear attributes of circuits.

2.3 Basis Functions Representation

Using time-domain basis functions to simulate circuits is a relatively new technique. Only a few methods have been proposed to analyze circuits with basis functions other than sinusoids (i.e., HB). In references [10, 7] pseudo-wavelet basis functions were used for the time domain analysis of circuits. The methods in publications [10] allow variable resolutions. Publications [25, 35, 29] are intended for steady-state analysis.

By using basis functions, the unknowns in circuit equations are no longer voltages and currents but are now the coefficients of the basis functions.

$$v(t) \simeq \sum_{i=1}^{N} \varphi_i(t) \cdot \hat{v}_i \tag{2.22}$$

where, $\varphi_i(t)$: a basis function
$\hat{v}_i$: one element in the coefficient vector of $\hat{V}$ and $\hat{V} = [\hat{v}_1, \hat{v}_2, ...\hat{v}_i, ...\hat{v}_N]^T$
N: the integer

The objective of introducing the basis functions is to convert the differential equations that model a circuit into algebraic equations.

The basis functions are not unique. Examples are *sinusoids* (*i.e.*, HB), the *Daubechies* and *symlet* family of wavelet, and they can be used in many application fields. A few methods have been proposed to analyze circuits relying on basis functions [10, 7, 34].

Spline functions can also be used as basis functions. In this thesis we will use a set of spline functions referred as adaptive basis function (ABF) because the locations of the splines are adapted according to the shape of the waveforms in the circuit. The ABF were first proposed for circuit analysis by Wenzler and Lueder [32]. The approach consists in dividing a period into n intervals where the given function is piecewise-approximated by third-degree polynomials (i.e. ABF) satisfying boundary conditions. This approach shares most of the

merits of wavelets: varying resolution and local support that originates sparse matrices.

2.3.1 Adaptive Basis Functions (ABF)

For the same accuracy, less points are required to evaluate the shape of circuit solutions if we apply a non-uniform grid instead of a uniform grid. Therefore, it is better to apply non-uniform grids in order to reduce the number of parameters.

More basis functions are necessary in regions of waveforms with 'drastic activity' but less in 'low activity' regions. ABF can be dilated and shrunk along with grids. When sudden variations in time domain take place (such as sudden varying amplitude or chaotic circuit behavior), more ABF will be accumulated.

For further performance, a novel idea is to make each state variable grasp its own adaptive grid without interfering with the rest of the grids — multiple adaptive grids. In Publication [32], one common grid is used for all state variables. However, the method presented here would support multiple grids.

Formulation of ABF

The functions displayed in Fig.2.5 and 2.6 are so-called ABF. ABF has two forms $\varphi_{2j}(t)$ as Eq. (2.23) and $\varphi_{2j+1}(t)$ as Eq. (2.24). They are created such that the coefficient parameters $\hat{v}_i$ represent function values and its derivative at every point t_j. In fact, some desirable properties [33] for regular basis function are satisfied by ABF. They are suitable to speed up superposition and good spatial localization and also able to represent the each state variable with independent resolution.

$$\varphi_{2j}(t) = \begin{cases} 1 - 3y^2 + 2y^3 & t_{j-1} \leq t \leq t_{j+1} \\ 0 & \text{elsewhere} \end{cases} \tag{2.23}$$

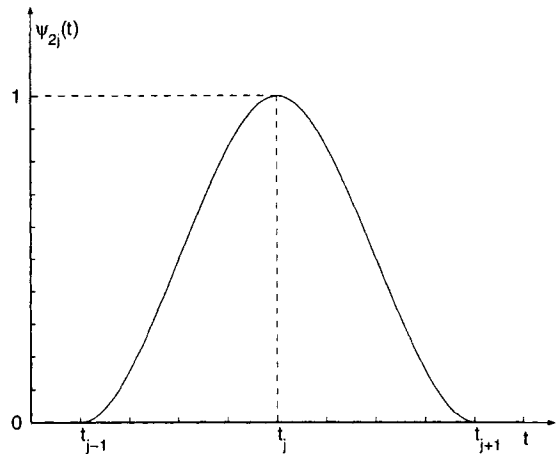

Figure 2.5: Adaptive basis function $\varphi_{2j}(t)$

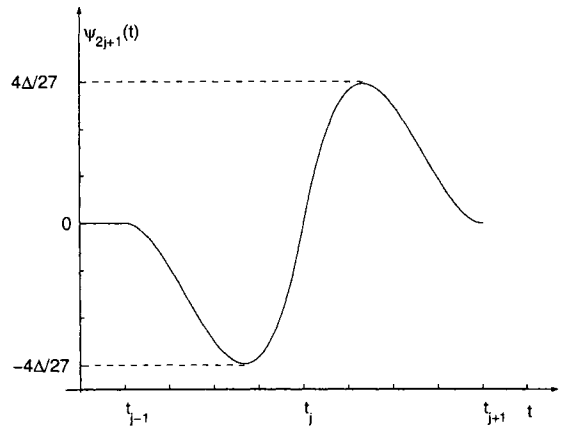

Figure 2.6: Adaptive basis function $\varphi_{2j+1}(t)$

$$\varphi_{2j+1}(t) = \begin{cases} (t - t_j)/(1 - y)^2 & t_{j-1} \leq t \leq t_{j+1} \\ 0 & \text{elsewhere} \end{cases} \tag{2.24}$$

where

$$y = \begin{cases} (t_j - t)/(t_j - t_{j-1}) & t \leq t_j \\ (t - t_j)/(t_{j+1} - t_j) & t \geq t_j \end{cases} \tag{2.25}$$

t_j represents discrete point , where j is the index of points. The interval $\triangle t_j$ of two adjacent

discrete points equals to $period/n$ for uniform grids.

Any arbitrary given function may be projected onto ABF and expressed as a linear combination of ABF and corresponding coefficients. In this manner, a single point $\bar{v}(t_a)$ on the waveform $\bar{v}(t)$, where t_a is an arbitrary point in the time domain, is approximated by

$$\bar{v}(t_a) = \varphi_{2(j-1)}(t_a)\cdot\hat{v}_{2(j-1)}+\varphi_{2(j-1)+1}(t_a)\cdot\hat{v}_{2(j-1)+1}+\varphi_{2j}(t_a)\cdot\hat{v}_{2j}+\varphi_{2j+1}(t_a)\cdot\hat{v}_{2j+1} \quad (2.26)$$

where $\varphi_{2(j-1)}(t)$ and $\varphi_{2(j-1)+1}(t)$ are the basis functions located at the previous $(j-1)$ interval, and $\varphi_{2j}(t)$ and $\varphi_{2j+1}(t)$ are located at the current j interval.

Note, since the steady-state response is periodic and the period is known, the last interval of the period will share two coefficients with the first interval.

2.4 Grids Adaptation

Cook and Duncan in the reference [2] explore a grid adaptation technique for one state variable when the circuit waveform show steep regions. For a set of elements of size $\triangle t_1$, $\triangle t_2$, ... $\triangle t_j$... , spanning the domain of the function f . Define a set of area residuals P_j as

$$P_j = \int_{\triangle t_j} (f - f^*)^2 dt \quad (2.27)$$

where, f is a function and f^* is the linear interpolation to f.

In Fig. 2.7, suppose that the solid curve represents an approximated waveform , and the dashed straight line represents a piecewise linear interpolation which is a segment connecting two adjacent grids denoted by t_{j-1} and t_j. P_j is the shaded area.

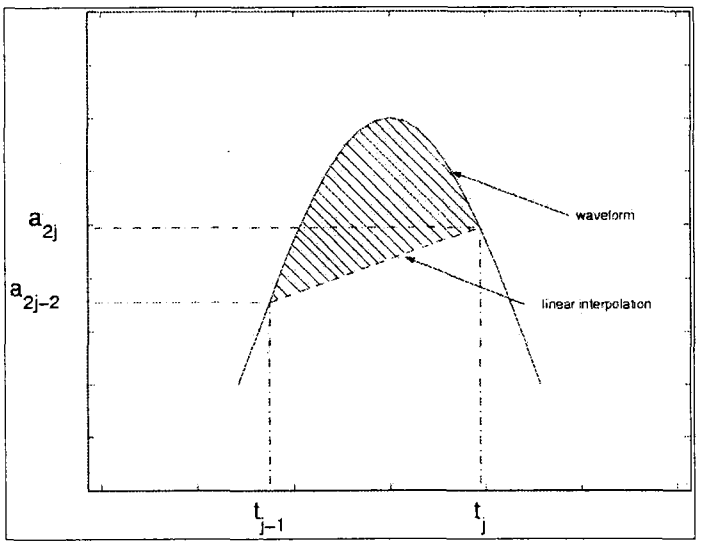

Figure 2.7: Illustration of error measurement in one interval

The objective of the grid adaptation technique is to distribute a set of P_j uniformly by moving the grids. The resulting interval sizes $\triangle t_j$ are changed.

The mean of irregular area residuals P_j is given by

$$\bar{P} = \frac{1}{N}\sum_{k=1}^{N} P_j \tag{2.28}$$

The new grid elements $\triangle \tilde{t_j}$ are calculated by

$$\triangle \tilde{t_j} = \triangle t_j (\bar{P}/P_j)^{0.2} \tag{2.29}$$

$\triangle \tilde{t_j}$ is corrected by a scaling factor s [2] to re-scale the mesh in order to reduce unexpected oscillation possibility and enhance the convergence rate.

$$s = \frac{\sum_{j=1}^{N} \triangle t_j}{\sum_{j=1}^{N} \triangle \tilde{t_j}} \tag{2.30}$$

to yield

$$\triangle \tilde{\tilde{t}}_j = s \cdot \triangle \tilde{t}_j \tag{2.31}$$

Finally, the new set of grids are obtained,

$$\triangle t_j(new) = 0.5 \triangle \tilde{\tilde{t}}_j + 0.5 \triangle t_j \tag{2.32}$$

Chapter 3

Equation Formulation

In this chapter, we formulate equations for finding steady-state solutions of the general non-linear circuit. The first section introduces the transformation matrices, which are the key of formulating the subsequent circuit equations. In Section 2, the completed formulation is described in two cases of uniform and non-uniform. The least square method for solving the formulated equations are also addressed in this section. In Section 3, we develop a procedure for grids adaptation, it provides the effective way to adapt the grids such that the circuit solutions achieve high accuracy using less unknowns.

3.1 Transformation Matrices B and B_p

Define U as a vector of sample points with the nodal voltages,

$$U = \left[\, U_1, U_2, ..., U_i, ..., U_m \, \right]^T \tag{3.1}$$

where U_1 represents a vector of voltages at node 1, U_2 is a vector of voltages at node 2 and so on. m is the number of state variables in the circuit. Particularly, for a certain variable,

the vector of sample points U_1 is composed by the values of $U_1(t)$ at a sequence of points t_k as shown in Eq. (3.2). The selection of t_k will be introduced later in this section.

$$U_1 = \left[U_1(t_0), U_1(t_1), ..., U_1(t_k), ... \right]^T \tag{3.2}$$

It is a possibility to define two transformation matrices B and B_p associated with $\varphi_i(t)$ such that

$$U_i = B \cdot \hat{U}_i \tag{3.3}$$

$$\dot{U}_i = B_p \cdot \hat{U}_i \tag{3.4}$$

where $\hat{U}_i$ is a element in vector $\hat{U}$, $\hat{U} = \left[\hat{U}_1, \hat{U}_2, ..., \hat{U}_i, ..., \hat{U}_m \right]^T$. $\hat{U}_i$ represents the unknown coefficient vector for all state variables. B and B_p are constant matrices. The purpose of using B is to convert coefficient vector $\hat{U}_i$ to a vector of time samples U_i for single state variable. Similarly, B_p converts $\hat{U}_i$ to the derivative of time samples $\dot{U}_i$. For example,$U_1 = B\hat{U}_1$, here $\hat{U}_1$ represents the unknown coefficient vector for the first state variable and U_1 represents the corresponding vector of time samples.

For illustration, a typical structure of B is shown as

$$B = \begin{bmatrix} \varphi_0(t_1) & \varphi_1(t_1) & \varphi_2(t_1) & \varphi_3(t_1) & 0 & ... & 0 \\ \varphi_2(t_2) & \varphi_3(t_2) & \varphi_4(t_2) & \varphi_5(t_2) & 0 & ... & 0 \\ 0 & ... & 0 & \varphi_{2(j-1)}(t_3) & \varphi_{2(j-1)+1}(t_3) & \varphi_{2j}(t_3) & \varphi_{2j+1}(t_3) \\ & & ... & & & & \\ & & & ... & & & \\ \varphi_0(t_k) & \varphi_1(t_k) & 0 & ... & 0 & \varphi_{2n-2}(t_k) & \varphi_{2n-1}(t_k) \end{bmatrix} \tag{3.5}$$

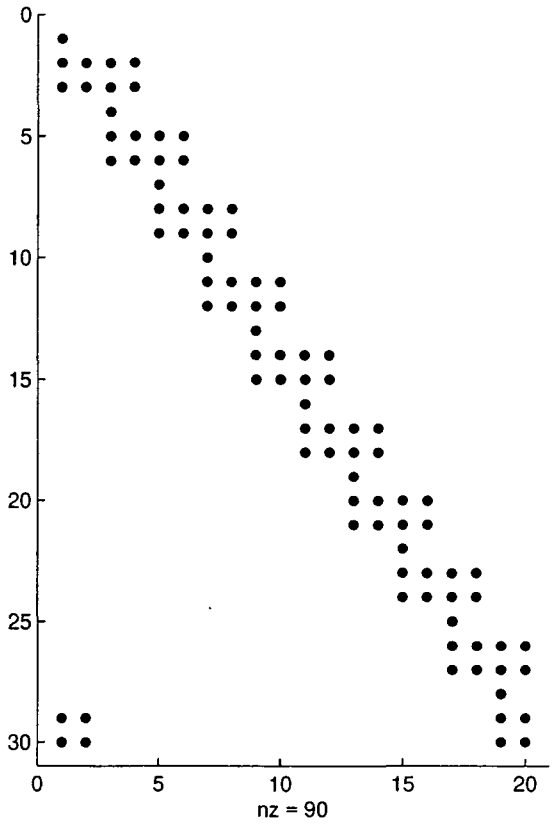

Figure 3.1: Structure of transformation matrix B

where $j = 1, 2, ..., n$ and n is the number of intervals.

The nonzero elements in each row in B are expressed by four consecutive ABF, denoted $\varphi_{2(j-1)}(t)$, $\varphi_{2(j-1)+1}(t)$, $\varphi_{2j}(t)$ and $\varphi_{2j+1}(t)$, the rest of elements in B are all zeros, thus evidently matrix B is sparse, Fig. 3.1 displays the sparsity of an transformation matrix B.

Sample point selection:

Now considering an interval $\triangle t_j$ of two discrete points with $j = 1, 2, ..., n$, we introduce and insert two extra intermediate points t'_j and t''_j in each interval as the Fig. 3.2 shows. For simplicity, we denote a new parameter named as the grid point t_k with $k = 1, 2, ..., 3n$ (with uniform grids) to represent those three points in one interval. For example,

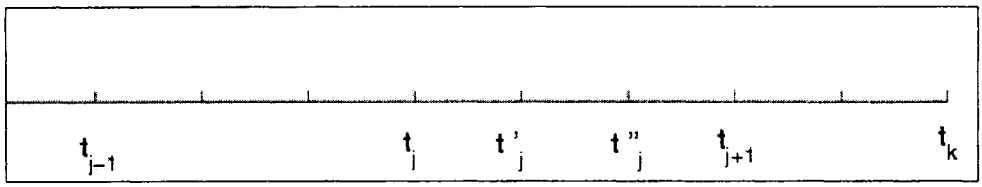

Figure 3.2: Illustration of t_j and t_k

$$t_k = \begin{cases} t_j & k = 1, j = 1 \\ t'_j & k = 2, j = 1 \\ t''_j & k = 3, j = 1 \end{cases} \tag{3.6}$$

The number of rows of B and B_p equals to the size of t_k, that is $3n$ in case of uniform grids. The number of columns of B and B_p equals to the double number of intervals in the circuit equations, because two basis functions exist in each interval corresponding to occupy two columns. For instance, assume $n = 10$ and uniform grids are used, the size of B goes to 30×20. The nonzero elements is 15 percent of the total elements as Fig. 3.1 shown .

3.2 Expansion of Circuit Equations

In this section nonlinear circuit equations are formulated and solved by least squares method. This derivation is based on nodal analysis and is valid for all circuits.

Consider the nodal equation formulation of a circuit,

$$\underbrace{Gu(t) + C\dot{u}(t)}_{linear} + \underbrace{I(u(t)) + \dot{Q}(u(t))}_{nonlinear} = S(t), \tag{3.7}$$

where, $u(t)$ is a vector of nodal voltages, G and C are square matrices that represent the linear conductance and charge terms, respectively. The vector functions $I(u(t))$ and $\dot{Q}(u(t))$ stand for the contributions of the nonlinear algebraic and charge terms, respectively. The

independent source vector $S(t)$ has the form of

$$S = \left[S_1, S_2, ..., S_m \right]^T \tag{3.8}$$

$u(t)$ and $S(t)$ satisfy the following periodical boundary conditions:

$$u(t+T) = u(t)\ ,\ S(t+T) = S(t) \tag{3.9}$$

It is important to separate linear and nonlinear terms in Eq. 3.7 because that results in a more efficient numerical solution. For demonstrating the primary idea in formulation procedure, we firstly formulate the equations in the uniform case followed by the nonuniform situation.

3.2.1 Equations with Uniform Grid

For the nonlinear terms $I(u)$ and $Q(u)$, we manipulate Eq. (3.7) as

$$\begin{cases} Gu(t) + C\dot{u}(t) + \bar{M}\dot{q} + I(u(t)) = S \\ -Mq + Q(u(t)) = 0 \end{cases} \tag{3.10}$$

$$\begin{bmatrix} G & 0 \\ 0 & -M \end{bmatrix} \begin{bmatrix} u \\ q \end{bmatrix} + \begin{bmatrix} C & \bar{M} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{q} \end{bmatrix} + \begin{bmatrix} I(u) \\ Q(u) \end{bmatrix} = \begin{bmatrix} S \\ 0 \end{bmatrix} \tag{3.11}$$

where M, $\bar{M}$ denote the identity matrix and the mapping (or incidence) matrix respectively. Eq. (3.11) is solved using Newton-Raphson method [15, 20]. The nonlinear terms at the next iteration $I(u^{j+1})$, $Q(u^{j+1})$ can be calculated as follows,

$$I(u^{j+1}) \simeq I(u^j) + J_I(u^{j+1} - u^j) \tag{3.12}$$

$$Q(u^{j+1}) \simeq Q(u^j) + J_Q(u^{j+1} - u^j) \tag{3.13}$$

$$\left(\begin{bmatrix} G & 0 \\ 0 & -M_q \end{bmatrix} + \begin{bmatrix} J_I^j & 0 \\ J_Q^j & 0 \end{bmatrix}\right)\begin{bmatrix} u^{j+1} \\ q^{j+1} \end{bmatrix} + \begin{bmatrix} C & \bar{M} \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \dot{u}^{j+1} \\ \dot{q}^{j+1} \end{bmatrix} = \begin{bmatrix} S \\ 0 \end{bmatrix} - \begin{bmatrix} I(u^j) \\ Q(u^j) \end{bmatrix} + \begin{bmatrix} J_I^j & 0 \\ J_Q^j & 0 \end{bmatrix}\begin{bmatrix} u^j \\ q^j \end{bmatrix} \tag{3.14}$$

where M_q is an identity matrix associated with the dimension of q.

Rearrange Eq. (3.14) to obtain,

$$\underbrace{\begin{bmatrix} G + J_I^j & 0 \\ J_Q^j & -M_q \end{bmatrix}}_{G'}\underbrace{\begin{bmatrix} u^{j+1} \\ q^{j+1} \end{bmatrix}}_{u'} + \underbrace{\begin{bmatrix} C & \bar{M} \\ 0 & 0 \end{bmatrix}}_{C'}\underbrace{\begin{bmatrix} \dot{u}^{j+1} \\ \dot{q}^{j+1} \end{bmatrix}}_{\dot{u}'} = \underbrace{\begin{bmatrix} S - I(u^j) + J_I^j u^j \\ -Q(u^j) + J_Q^j q^j \end{bmatrix}}_{\mathbf{S}'} \tag{3.15}$$

$$G'u' + C'\dot{u}' = \mathbf{S}' \tag{3.16}$$

Therefore, by means of Eq. (3.16), the set of nonlinear equations that can be solved as a sequence of linear solutions.

We will now derive the equations for a linear circuit with uniform grid. Generality is not lost because if the circuit is nonlinear, an equivalent system of equations (Eq. 3.16) can be obtained at each Newton iteration. Consider the equation describing a linear circuit,

$$Gu(t) + C\dot{u}(t) = S(t) \tag{3.17}$$

Recall Eq.(3.1) that U is a vector of sample points for many state variables and the $\dot{U}$ is the corresponding derivative vector. They satisfy the following equation,

$$(G \otimes M)U + (C \otimes M)\dot{U} = \mathbf{S} \tag{3.18}$$

where $\otimes$ is kronecker product. Rewrite Eq. (3.18) as a function of $\hat{U}$, where $\hat{U}$ is the coeffient vector, then

$$(G \otimes B)\hat{U} + (C \otimes B_p)\hat{U} = \mathbf{S} \tag{3.19}$$

By rearraging, we get

$$\underbrace{(G \otimes B + C \otimes B_p)}_{A}\hat{U} = \underbrace{\mathbf{S}}_{\mathbf{b}} \tag{3.20}$$

$$\begin{bmatrix} g_{11}B + c_{11}B_p & g_{12}B + c_{12}B_p & \cdots & g_{1m}B + c_{1m}B_p \\ g_{21}B + c_{21}B_p & g_{22}B + c_{22}B_p & \cdots & g_{2m}B + c_{2m}B_p \\ \vdots & & & \\ g_{m1}B + c_{m1}B_p & g_{m2}B + c_{m2}B_p & \cdots & g_{mm}B + c_{mm}B_p \end{bmatrix} \begin{bmatrix} \hat{I}_1 \\ \hat{U}_2 \\ \cdots \\ \hat{U}_m \end{bmatrix} = \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \\ \cdots \\ \mathbf{S}_m \end{bmatrix} \tag{3.21}$$

The total number of equations in Eq. (3.20) is larger than the unknowns. Therefore, Eq.(3.20) can be solved by the least squares method. The only unknowns $\hat{U}$ can be computed as below using Eq. (2.12),

$$A^T A \cdot \hat{U} = A^T \mathbf{b} \tag{3.22}$$

3.2.2 Equations with Non-uniform Grid

Now considering the grids points are not equal distances for all nodal voltages in a linear circuit, Eq. (3.20) has to be modified to use different transformation matrices (B and B_p) for each combination of state variables.

The circuit system is solved and the solutions are used to compute the next positions of the grids (i.e. non-uniform grids) using the strategy described in Section 3.3. Beware of B and B_p have different values with respect to different state variables. The items in Eq. (3.21) should be modified as the following,

$$\begin{bmatrix} g_{11}B_1 + c_{11}B_{p1} & g_{12}B_2 + c_{12}B_{p2} & \cdots & g_{1m}B_m + c_{1m}B_{pm} \\ g_{21}B_1 + c_{21}B_{p1} & g_{22}B_2 + c_{22}B_{p2} & \cdots & g_{2m}B_m + c_{2m}B_{pm} \\ \vdots & & & \\ g_{m1}B_1 + c_{m1}B_{p1} & g_{m2}B_2 + c_{m2}B_{p2} & \cdots & g_{mm}B_m + c_{mm}B_{pm} \end{bmatrix} \begin{bmatrix} \hat{U}_1 \\ \hat{U}_2 \\ \cdots \\ \hat{U}_m \end{bmatrix} = \mathbf{S} \quad (3.23)$$

The rest solving procedures with uniform grids are essentially the same to non-uniform grids.

Initially, ABF are assigned in uniform grids. Because we need higher level of accuracy and efficiency, it is a better choice to start with a unform grid followed by non-uniform grids adapted to the shape of any $u(t)$. In fact, non-uniform grids imply unequal-sized intervals but the total number of intervals n in a period still remains the same.

Now, we discuss the reselection of sample points in the nonuniform case. By definition of t_k in section 3.1, we know that the postions of t_k are determined by two adjacent node points t_{j-1} and t_j. When constructing a non-uniform grid, the new t_j grid is the union of all t_j grids counting from every state variable. Thus the size of new grid (or new t_k) is generally larger or at least the same as the size of any old grid. Once the t_j are settled, the immediate points t_k therefore can be obtained.

In addition, it is difficult to know in advance how many equations will be considered with

nonuniform grids. But in geneal, the number of equations will be greater than the one with uniform grids.

3.3 Measurement and Grid Adaptation

The main objective of this chapter is to individually adapt the nonuniform grid-positions of different state variables in the circuit such that we can achieve the desired resolutions by using relatively less coefficients. As the flowchart 3.3 shows, the prerequisite knowledge to adapt the grid is the area-residuals measurement determining the extent of grids adaptation, starting as the first step of the procedure. Subsequently, the grid adaptation technique is explored.

3.3.1 Stages of the Residual Measurement

The measurement of area-residuals sounds like a kind of 'sensor' used to monitor residuals between any two adjacent grids. The first stage, called a coarse measurement, attempts to roughly capture grid-positions, and the most immediate benefit is to run faster and be closer to actual shape of the curve. The second stage offers more precise residuals measurement and needs longer computation.

Stage 1

To accelerate the evaluation, only the linear interpolation length l_k will be taken into account in stage 1. Firstly, we take the normalization to ensure that l_k are distinct enough to be measured. Define $\triangle t_k = t_k - t_{k-1}$, $\triangle u(t_k) = u(t_k) - u(t_{k-1})$.

Normalize every grid $\triangle t_k$ by the input signal period T,

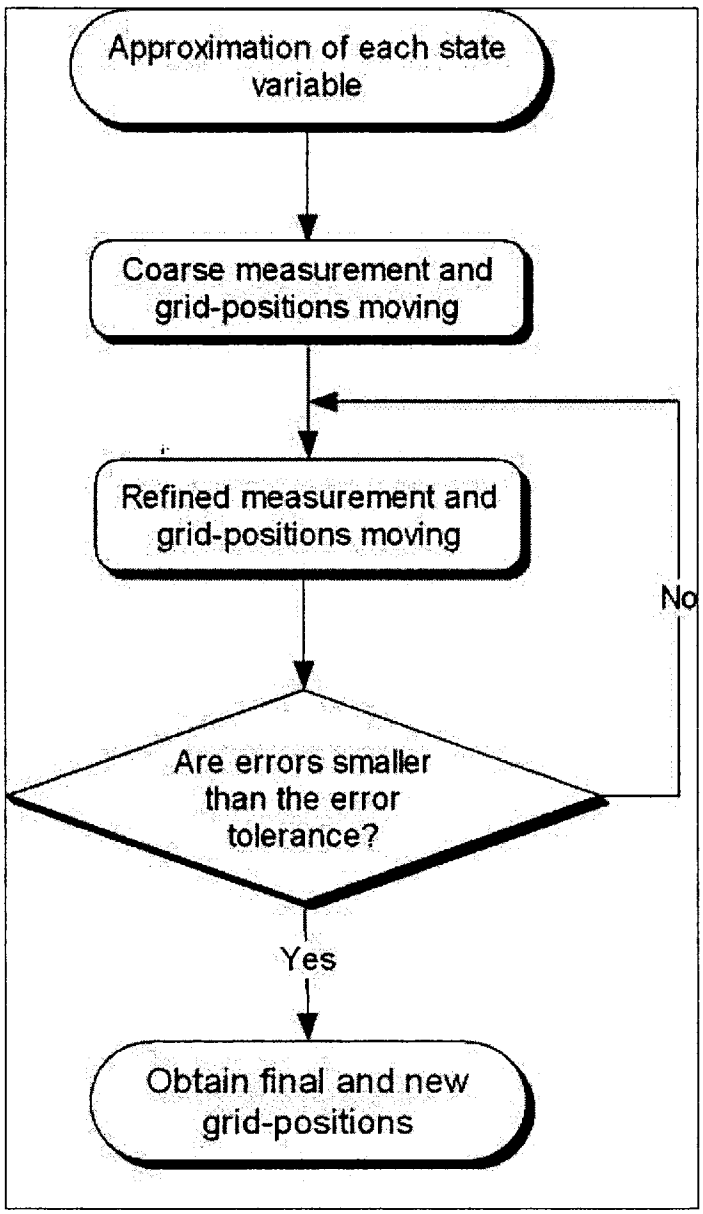

Figure 3.3: Flowchart of adaptive grid position control

$$\tilde{\triangle t_k} = \frac{\triangle t_k}{T}$$

Normalize the $\triangle u(t_k)$ by the total span of $u(t_k)$,

$$u_{span} = max[u(t_k)] - min[u(t_k)]$$

$$\triangle \tilde{u}(t_k) = \frac{\triangle u(t_k)}{u_{span}}$$

Then every length of linear interpolations l_k can be calculated by

$$l_k = \sqrt{[\triangle \tilde{u}(t_k)]^2 + (\triangle \tilde{t_k})^2} \tag{3.24}$$

Secondly, the average value $\bar{l}$ of linear interpolations is given by

$$\bar{l} = \frac{1}{N} \sum_{k=1}^{N} l_k \tag{3.25}$$

where, k is the index of linear interpolation, N is the total number of l_k.

When every l_k equals to $\bar{l}$, the stage 1 is accomplished and stage 2 will be followed.

Stage 2

For better accuracy, the impact of the arch existing between two intermediate grids is considered. Evaluate the residual accumulation by measuring P_k , which is the shaded region illustrated by Fig. 2.7. In other words, we measure irregular areas in replacement of solely considering lengths of the straight lines l_k. Apparently, the extra time is demanded for this delicate residual evaluation.

The calculation of P_k is shown in Section 2.4. P_k can be calculated analytically or numerically using the Simpson Rule [15] or any other numerical integration method.

3.3.2 Implementation of Grids Adaptation

The objective of the grid adaptation is to provide identical l_k spanning at every intervals followed by identical P_k as well. That is the reason that we manipulate the non-uniform grids.

Uniform grids imply unbalanced distribution of P_k that are an important indicators of the accuracy. The grid-position adaptation attempts to share P_k equally as much as possible by moving the grids. Every time the grids are moved the equations must be solved again. Moreover, grids are processed one by one, finally leading to m sets of new non-uniform grids.

The algorithm distributes P_k evenly by changing interval-sizes of each state variable. The interval-sizes then affects the collection extent of ABF along the time dimensions. 'Drastic activity' of a waveform must draw a dense collection of ABF, whereas 'low activity' region corresponds to a small gathering of ABF.

If any new grid point is created, the corresponding transformation matrices B and B_p must be recalculated. The adaption is to improve the accuracy of the analysis. Moreover, the

coefficient vector $\hat{U}$ must also be updated and saved as the starting guess for next iteration.

Chapter 4

Simulations

In this chapter we simulate two circuits, a linear RC circuit and a microwave feedback amplifier. The second circuit is more complicated than the first one when doing the simulation, because several tens of state variables are included in this circuit and we must linearize the nonlinear component into linear ones.

The simulations of the algorithm are accomplished by using Matlab 7.0. We test both circuits in terms of uniform and non-uniform grids and provide the results graphically. By testing two circuits, we not only investigate most of the attributes of the developed multiple adaptive algorithm, but also attempt to expand the application of the algorithm to the general and large circuit.

This chapter firstly exploit an RC linear system with square wave periodic input in order to test the algorithm followed by some evaluation and discussion in Section 4.1. Then we examine a high frequency feedback amplifier. The design procedure is given in the Appendix. The designed circuit is tested and discussed in the end of the chapter.

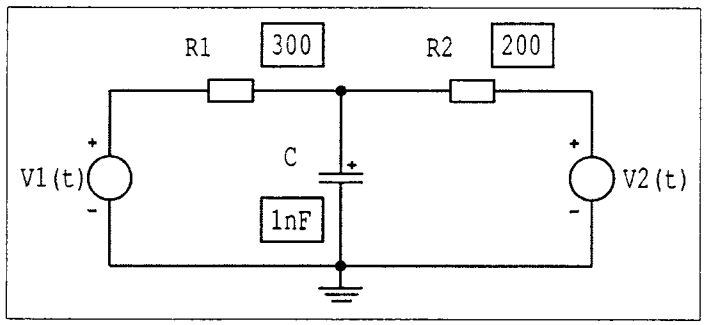

Figure 4.1: Topology of the RC circuit

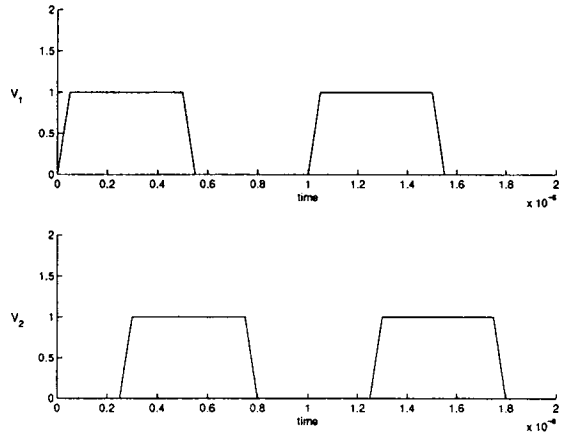

Figure 4.2: Voltage sources $V_1(t)$ and $V_2(t)$

4.1 RC Circuit with Square Wave Inputs

The RC circuit layout is illustrated in Figure 4.1. The circuit parameters are: $R_1 = 300\Omega$, $R_2 = 200\Omega$, $C_1 = 1nF$, and two square wave inputs $V_1(t)$ and $V_2(t)$ are displayed in Figure 4.2 , the amplitude is $1V$ and the period is $1\mu s$. Because the square wave implies many harmonics and sharp variations both at input and output, the time domain method is better suited to handle those harmonics as we mentioned early.

Describe the circuit by Eq. (3.7). The corresponding values of matrices G, C and S are shown below. The circuit is linear therefore there are no nonlinear $I(u)$ and $\dot{Q}(u)$ terms.

$$G=\begin{bmatrix} \frac{1}{R_2} & -\frac{1}{R_2} & 0 & 1 & 0 \\ -\frac{1}{R_2} & \frac{R_1+R_2}{R_1R_2} & -\frac{1}{R_1} & 0 & 0 \\ 0 & -\frac{1}{R_1} & \frac{1}{R_1} & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$C=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & c & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$S=\begin{bmatrix} 0 & 0 & 0 & V_1(t) & V_2(t) \end{bmatrix}^T$$

The circuit analysis becomes to seek the values of the unknown coefficient vector $\hat{U}$ which satisfies Eq. (3.7). Obtain the vector U at sample points t_k and least square fitting is used to acquire $\hat{U}$.

4.1.1 Simulation 1

Spice stands for simulation program with integrated-circuit emphasis, originally developed by the University of California, Berkeley in 1975 [21, 3]. In this example, the Spice solution is considered as the almost accurate solution calculated by taking 50 time steps with maximum sampling time $0.02\mu s$ which implies at least 50 intervals in a period.

10 intervals are chosen to test the performance of the developed algorithm with uniform grids. Results are compared with Spice in Fig. 4.3. The dashed line represents Spice results as the reference, whereas the solid line shows the waveform of the algorithm. In the Figures,

the grid positions are indicated as 'a set of points' listing along a line corresponding the marked points on every waveform for comparison purpose.

We can observe little unwanted oscillations of waveforms. When solving the equations of this RC circuit, the least square method tries to capture the information of all grid points in the intervals such that the squared residuals are minimized in order to form a smooth approximation curve. Because we chose a small number of intervals (10 intervals) in the uniform manner, the least square method can only solve the circuit based on limited grid points of 10 intervals in a period. Therefore, it leads to little oscillations on resulting approximation curve. Fortunately, the results can be improved by implementing the non-uniform grids or add more uniform grid points such as Fig. 4.4 (20 intervals).

4.1.2 Simulation 2

Contrast to Fig. 4.3, Fig. 4.6 emphasize on showing steady-state response comparisons of non-uniform case and Spice in one period, where the number of non-uniform grids iterations is limited at 15. The reason of selecting this value is discussed later in this section. Again, the grids adaptation strategy make it possible to achieve high accuracy using as little sample points as possible, but it pays an expense of more evaluation of circuit equations. The grid adaptation generates several sets of non-uniform grids.

For 10 intervals, the total element number in the vector $\hat{U}_i$ is 20. And for the 20 intervals, the total element number in the vector $\hat{U}_i$ is 40.

When 20 intervals and 5 state variable are included and 2 basis functions exist in each interval, the column number of B and B_p equals to 200 and the row number of them varies after every adaptation.

In addition, the nonzero map of matrix $A^T A$ in non-uniform case with 10 intervals is shown in Fig. 4.5. The number of nonzero elements is 5752, which is 8.22 percent of the number

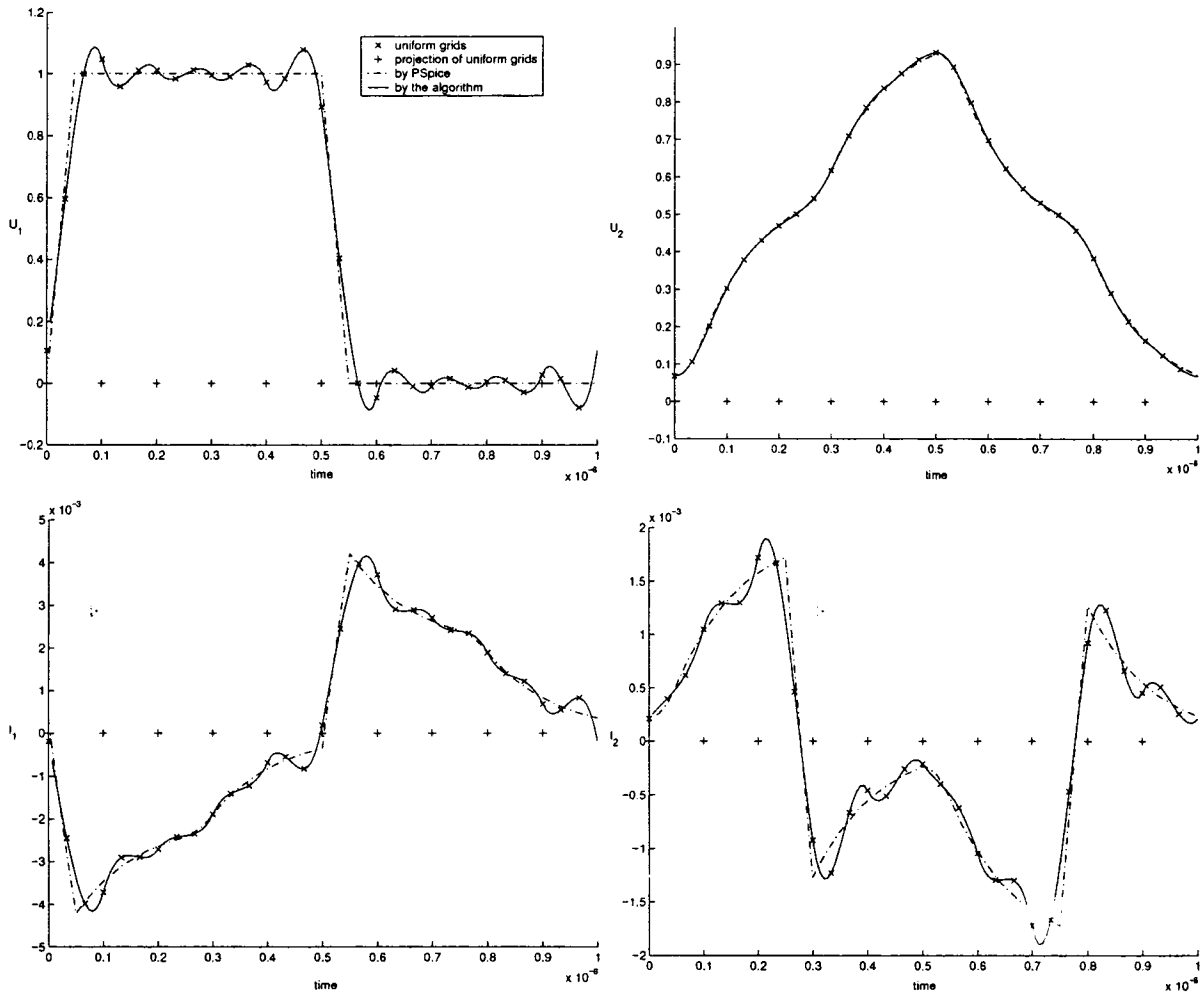

Figure 4.3: Example 1: Results with uniform grids compared to Spice (10 intervals)

of total elements. For this RC small circuit, the matrix $A^T A$ is not expected very sparse. But for a large circuit, the density will be decreased significantly because the matrices G and C are sparser (refer to Eq.(3.7)). Hence, the resulting $A^T A$ matrix are sparser than the small circuit.

By Fig. 4.6, the following observations can be made:

(1) Any set of grids is distinct from other set of grids corresponding to a different variable, such as the input voltage (U_1) and the voltage at node 2 (U_2) .

(2) More grids are collected in a region of 'drastic activity', however, few grids are assigned

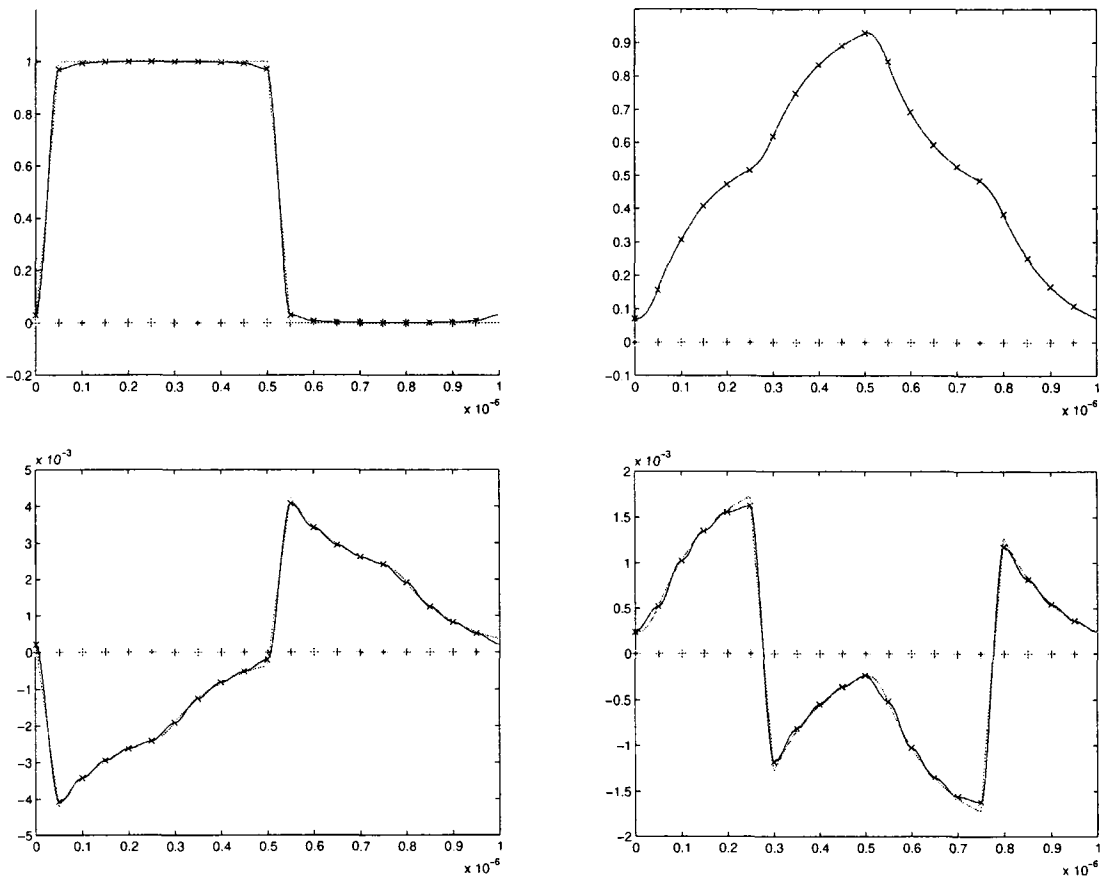

Figure 4.4: Example 1: Results with uniform grids compared to Spice (20 intervals)

in a mild changing region.

(3) By comparing of Figure 4.6 and Figure 4.3, we can observe that non-uniform grids can provide more flexible and accurate results than uniform grids using same number of intervals although some unwanted oscillations still exist.

(4) Even for simulating the sharp angular waveform especially at the spike and bottom, the outcomes still follow the actual solutions as Spice reference .

Another important observation can be made by comparing Fig. 4.8 and Fig. 4.9. They are plotting of the same state variable with 10 non-uniform intervals and 20 uniform intervals, respectively. We observe that they are almost equivalent to each other. Therefore, in terms

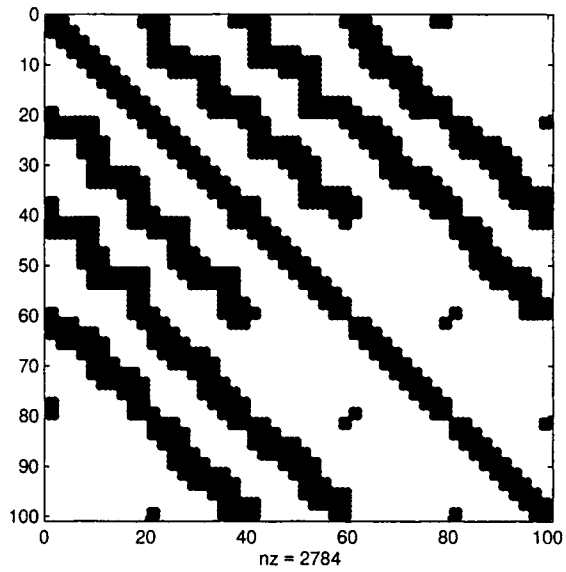

Figure 4.5: Example1: The nonzero map of matrix $A^T A$ by least squares method with the density of 27.84 percent(non-uniform, 10 intervals)

of non-uniform grids, high accuracy of results can be achieved by using less grids than the uniform grids case.

Evaluating the effectiveness of the grids adaptation:

For purpose of evaluating the grids adaptation, we define two parameters: a termination value and a guide number. The termination value denoted by F is defined as

$$F = \left(\frac{max(P_k)}{\bar{P}} - 1\right) + \left(1 - \frac{min(P_k)}{\bar{P}}\right) = 0 \tag{4.1}$$

where P_k is any of area-residuals in $[P_1, P_2, ..., P_k, ...]$ and $\bar{P}$ is the average value of all area-residuals. $max(P_k)$ represents the maximum value of P_k and $min(P_k)$ represents the minimum. For distributing P_k evenly, the perfect ratio of $P_k/\bar{P}$ is 1. However, we can not directly define F as $P_k/\bar{P} = 1$ because P_k is either larger or smaller than $\bar{P}$. If the ration of $P_k/\bar{P}$ is smaller than 1, the iterations will be terminated but in fact the best circuit solutions

have not been achieved. Therefore, we reasonably define the F as Eq. (4.1) whatever the value of P_k will be. The termination value should be set at 0. On the other hand, for each cycle of grids adaptation before reaching F, the value of $\left(\frac{max(P_k)}{P} - 1\right) + \left(1 - \frac{min(P_k)}{P}\right)$ is called as the guide number.

One interesting investigation is the relationship of iteration number and the guide number. Every iteration means a cycle of grid adaptation, and each grid adaptation leads to area-residuals are distributed uniformly. That is to say, the guide number becomes close to 0. The ideal situation is when the iteration number goes to increase, the guided number keeps to decrease until it reaches the preset termination value at 0.

Considering two following experiments:

(1) If we release the iteration number, let the iterations terminate until the termination value 0 is achieved.

Figure 4.10 shows the variation of the guide number when releasing the iteration number. In the figure, the guide number is smallest when the iteration number equals to 15 in this particular case. 10 intervals are chosen in this case.

We can observe that the guided number does not always decrease, but varies up and down. This situation still exists when we choose more or less interval numbers. The reason is that we evaluate the area residuals P_k based on a set of grid points. In fact,the small deviation of points from the exact waveform affects the value of P_k resulting in the errors in calculation of P_k. Because P_k must be remeasured by the new set of grid points when the grid-positions changes, based on the definition of the guided number, we can know that the guided number may goes up and down slightly. Therefore, it is reasonable to limit the iterations number at 15 to run the adaptation procedure.

(2)If we fix the iteration number at 15 and increase the intervals in a period, compute the CPU running time.

Table 4.1 lists the CPU time when fixing the iteration number at 15 and using more intervals. A Pentium 4 microprocessor is used to test the CPU time. Comparing Fig. 4.6 with Fig. 4.7, we can observe that

(i.) A larger interval number (such as, double intervals) results in a higher guided number, but the performance is significantly improved.

(ii.) Although the guided number does not equal to 0, the adaptation still conducts the results in the acceptable accuracy as Fig. 4.6 and Fig. 4.7 compared to uniform case in Fig. 4.3.

(iii.) With the increasing interval numbers, the overall CPU running time goes longer according to the specific cases.

The number of Intervals	CPU time	Guide number
10	2.81 s	1.736
20	10.33 s	3.349
30	24.88 s	3.160

Table 4.1: Example 1: Evaluation of CPU time when limiting the iteration number at 15 and using more intervals

4.1.3 Simulation 3

In this simulation test, the same RC circuit is used to compare effectiveness of two methods: collocation method and least square method.

At the previous simulation 1 and 2, we have tested the least square method. Fig. 4.11 shows the numerical result using a collocation method (I_2 is shown). We can observe that the artifact oscillation problems are encountered, this is because the A matrix turns out to be singular. However, the oscillation may be removed a lot by least square method, such as Fig. 4.4 and 4.7.

Next, we evaluate the computational effort of two methods by testing the density of the formulation matrix. By the collocation method, Fig. 4.12 shows the uniform case with 10 intervals, we observe that the number of nonzero elements in matrix A is 550, which is 5.5 percent of the total elements in the matrix. Note in the collocation method, because A is

always a square matrix, it is not necessary to compute $A^T A$ to see the density. Whereas, by the least squares method, A is rectangular, therefore, we need to evaluate $A^T A$ for comparison purpose. But $A^T A$ is more expensive to build than A alone. The number of nonzero elements is 2268 with the same conditions as Fig. 4.13 shown. The corresponding density is 22.68 percent.

Therefore, although the matrix $A^T A$ obtained by the least squares method is a little denser than matrix A by the collocation method, the unexpected oscillations can be significantly reduced by the least squares method. Furthermore, it offers the robustness and desirable accuracy. By comparison, we conclude that the least square method is better or at least equivalent to the collocation method for solving the formulated circuit equations.

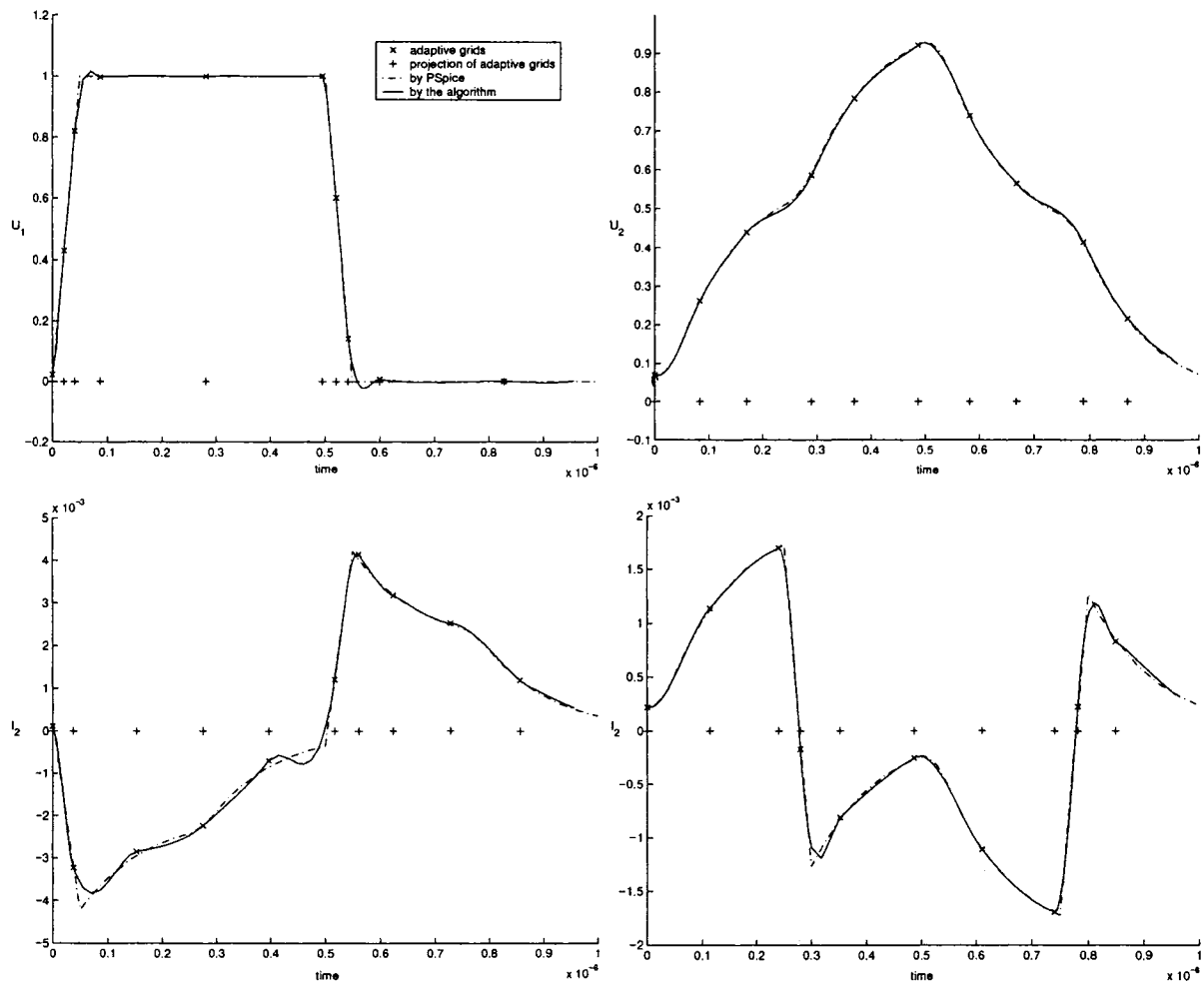

Figure 4.6: Example 1: Multiple adaptive algorithm in non-uniform grids compared to Spice (10 intervals)

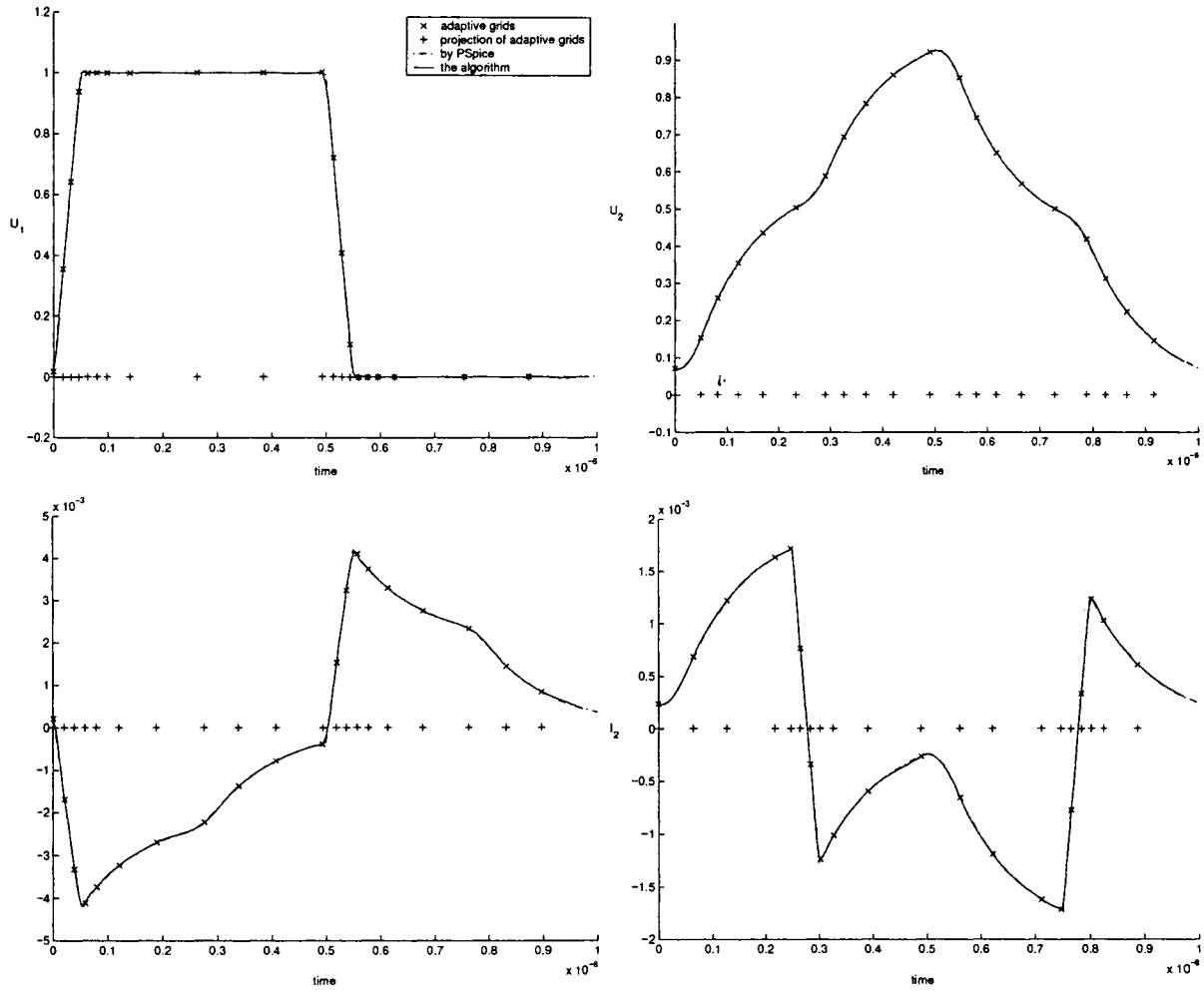

Figure 4.7: Example 1: Multiple adaptive algorithm in non-uniform grids compared to PSpice (20 intervals)

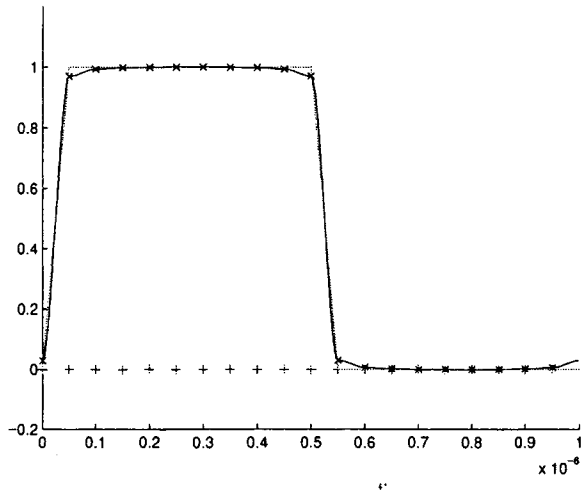

Figure 4.8: Example1: 20 intervals with uniform grids representing the input voltage

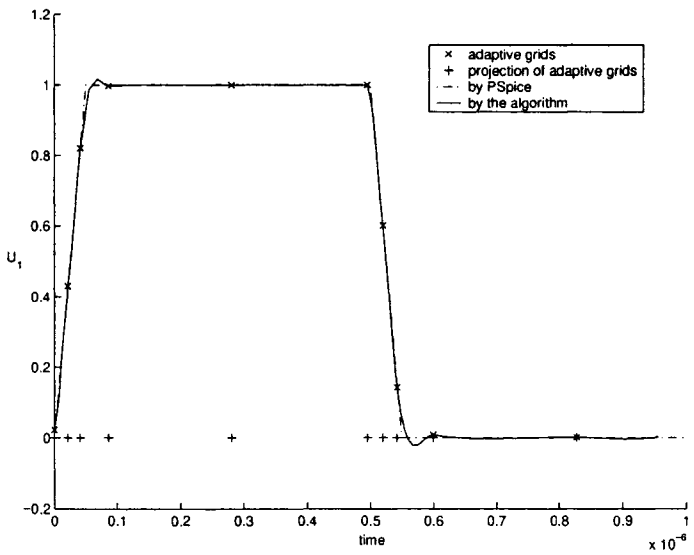

Figure 4.9: Example 1: 10 intervals with non-uniform grids representing the input voltage

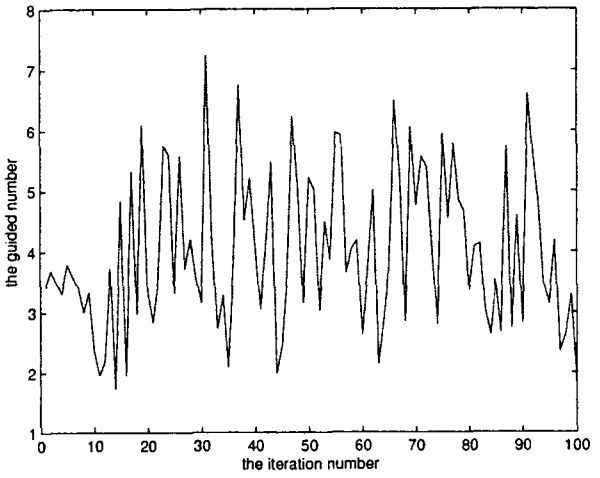

Figure 4.10: Example 1 :The guided number with respect to the number of iterations

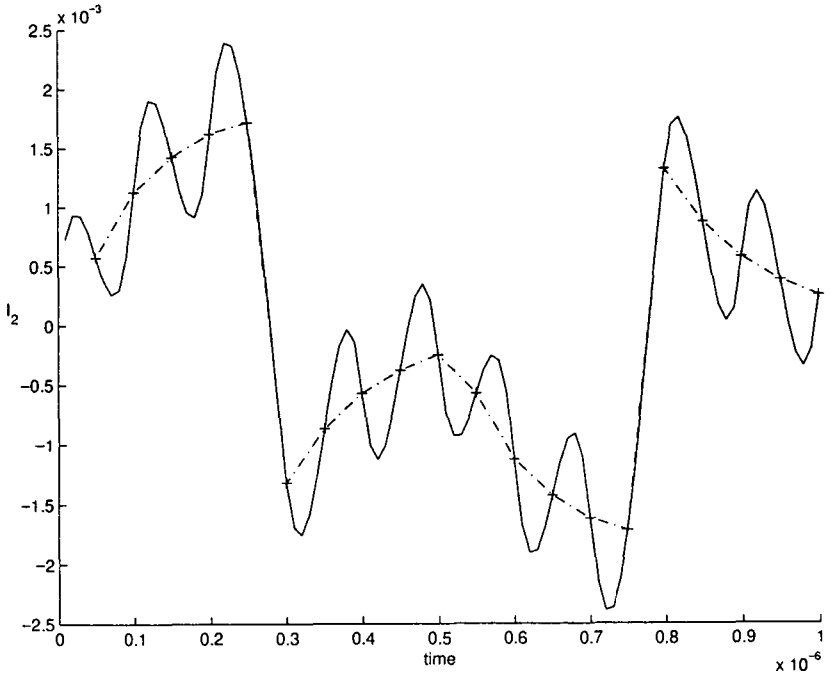

Figure 4.11: One example of unexpected oscillation

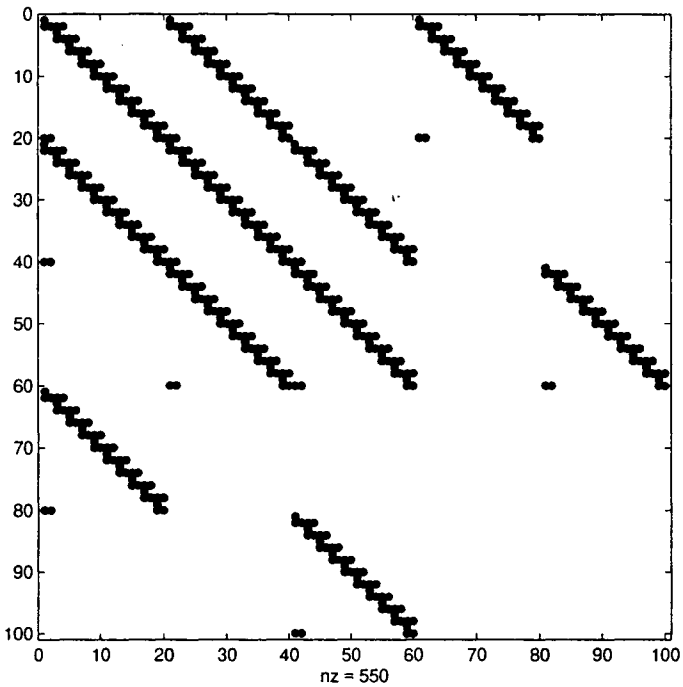

Figure 4.12: Example 1: The nonzero map of matrix A by the collocation method with the density of 5.5 percent(uniform, 10 intervals)

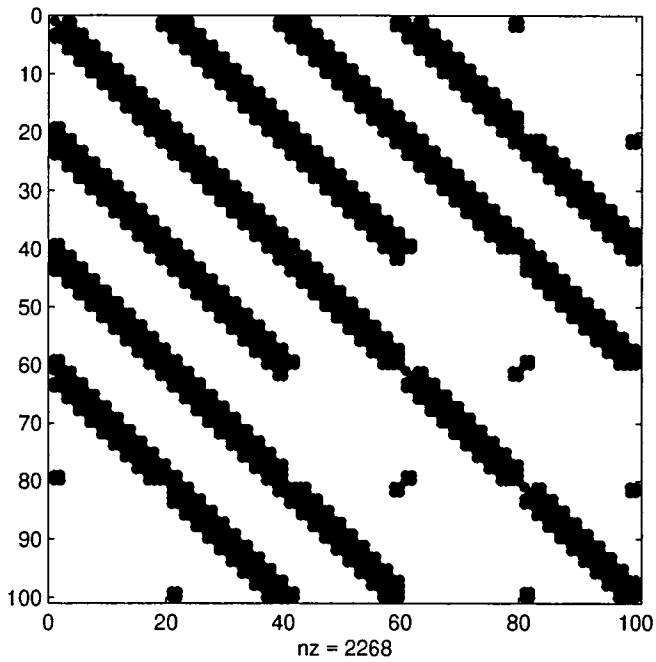

Figure 4.13: Example 1: The nonzero map of matrix $A^T A$ by the least squares method with the density of 22.68 percent (uniform, 10 intervals)

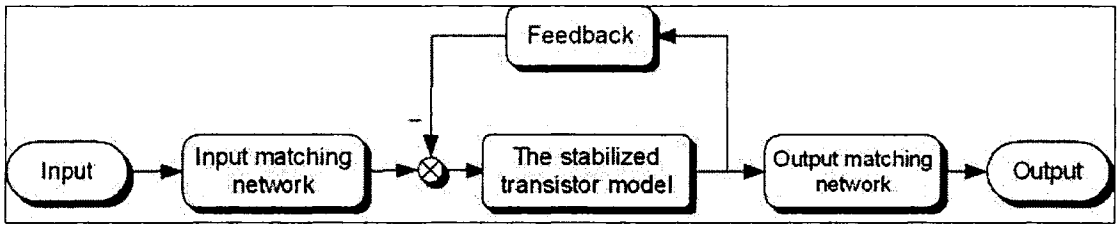

Figure 4.14: Diagram of the feedback amplifier design

4.2 Microwave Feedback Amplifier

In the preceding section, the detailed treatment has been given to simulate reaction of only the resistor and capacitor. In the following two sections, we will extend the adaptive algorithm to a larger circuit starting with design of a RF amplifier circuit, using a single transistor within single stage. The following important relevant design parts are attached in the appendix, and the related diagram is illustrated in Fig. 4.14.

- Intrinsic and extrinsic parameters of the small-signal equivalent transistor model [27]
- Design of feedback loop for flat gain $8dB$ from $10MHz$ to $0.8GHz$ [14]
- Biasing circuit
- Matching synthesis for lossless maximum power gain at frequency $0.7GHz$ [24]
- Optimization of Matching network over broadband frequency range

4.2.1 Design Objective

A design usually starts with the selection of the proper transistor in order to obtain the desired AC performance. We selected a RF transistor, Philips $BFS17W$, as the basic component in amplifier design. It is manufactured for design of broadband amplifier circuits working up to $1GHz$ at collector currents from $1mA$ to $20mA$, typical transducer gain is $12.7dB$.

A major design objective is to maintain constant power gain across a frequency range as wide as possible. Thus power gain and the frequency range will be compromised. We

reduced the gain ($|S_{21}|$) from $12.7dB$ to $8dB$ in order to obtain wide frequency bandwidth of amplification.

4.2.2 Final Design

The final circuit shown in Fig. 4.15 is the simulation object, where L_1, C_1 and L_3, C_2 form the input and output matching networks respectively. To evaluate the proposed algorithm capability of simulating circuit with multi-harmonic excitation, the circuit will be deliberately driven by triangular wave input. Because the triangular wave contains many harmonics, it is proper and reasonable to be used to examine the tracing property of the time-domain algorithm.

4.3 Linearized Feedback Amplifier Simulations

In contrast to the first RC circuit example, this is a larger circuit which contains the transistor model, peripheral feedback, stability and matching components. Due to the increasing size and complexity of the circuit, the need for simulation is twofold: validate the proposed algorithm and solve the circuit successfully.

The component values of the circuit are given in Fig. 4.15. To test the algorithm in the capability of following the steep waveform, the triangular input is used and composed by odd harmonics up to 19^{th} in Aplac. Obviously, the time-domain algorithm takes advantages to handle the waveforms with many higher order harmonics. The linearized circuit has 21 connection nodes and the nodal-based matrix G, C can be written with the dimension of 420×420 for each, and S with the dimension of 420×1. Then we rely on the Eq. 3.7 and least square method to acquire the circuit solutions.

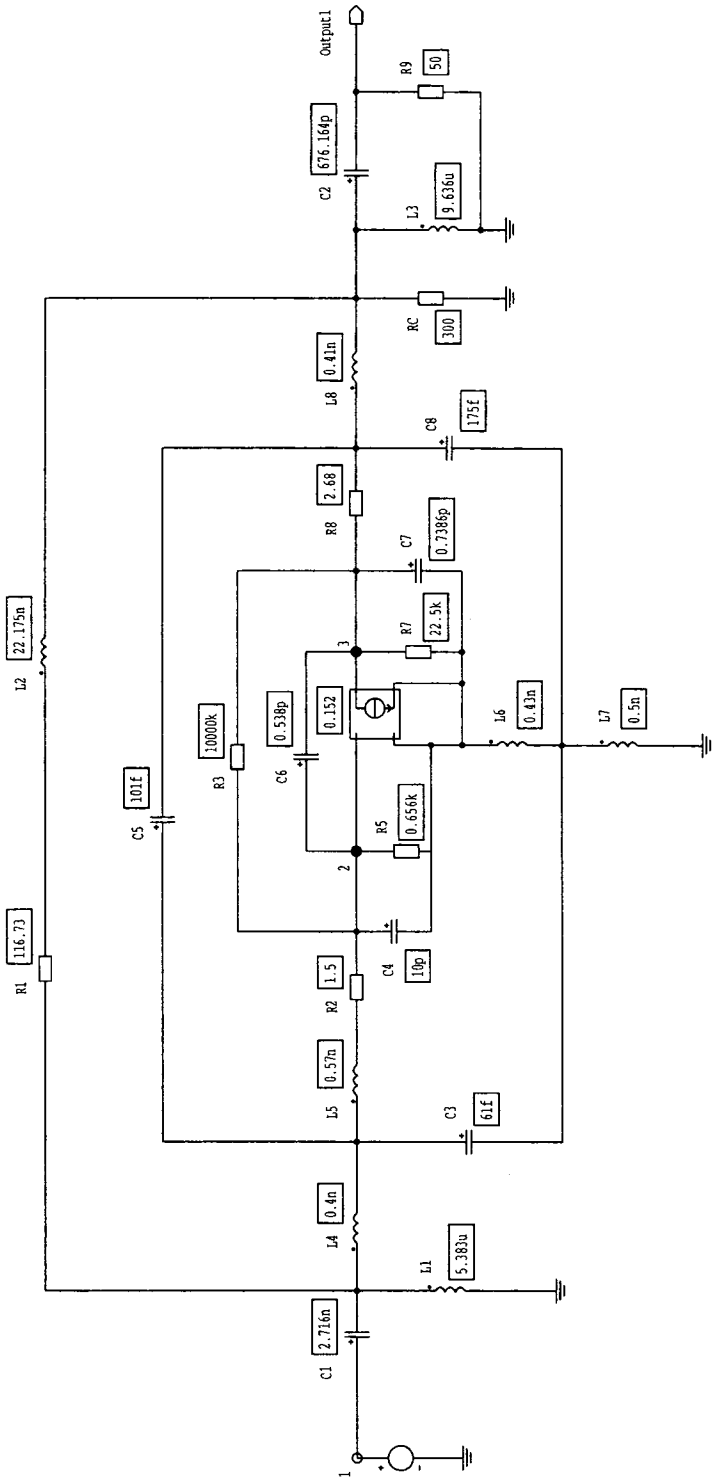

Figure 4.15: Final designed circuit

4.3.1 Simulation 4

Aplac 7.92 [9] simulation will be firstly shown and will be treated as the reference results to evaluate the adaptive algorithm, where Aplac is a RF analog simulation and design tool with rich analysis and optimization features. Aplac's capabilities exceed PSpice especially when dealing with RF and microwave frequencies. Fig. 4.7 shows the Aplac simulation results of the amplifier circuit for the period $T = 5ns$.

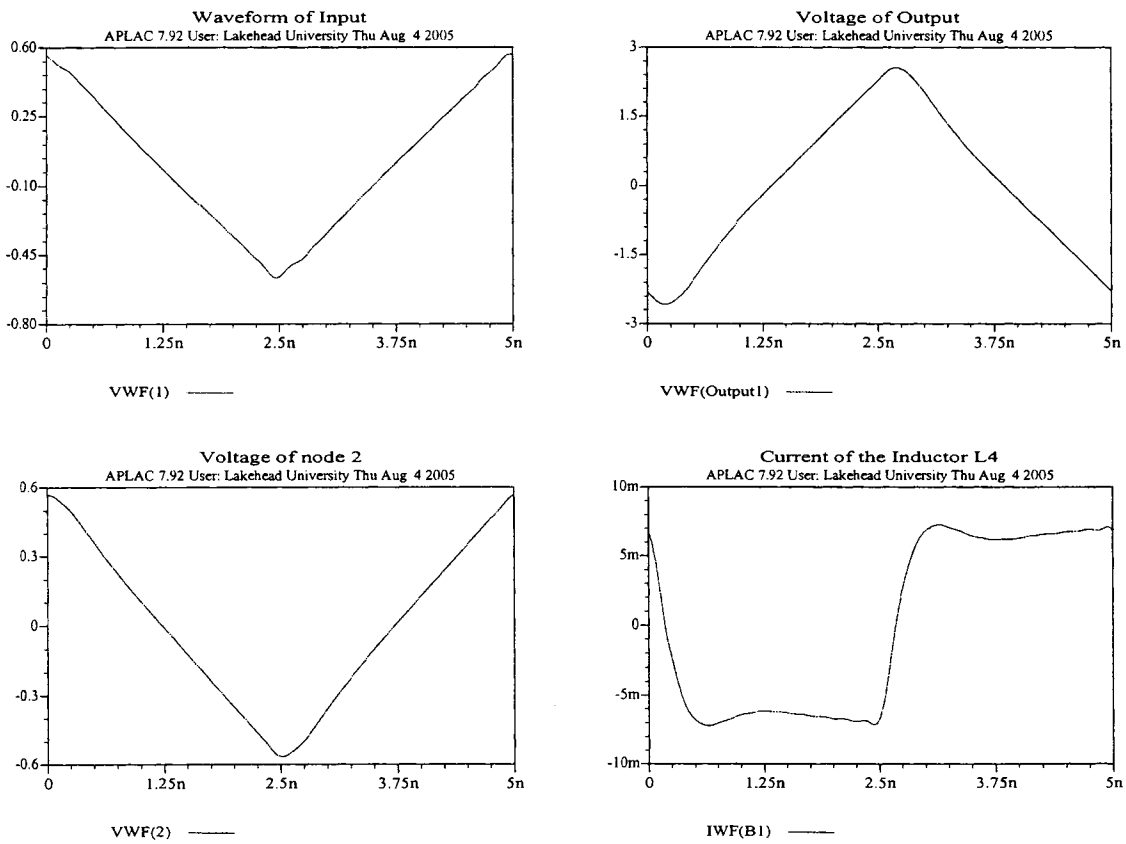

Figure 4.16: Example 2: APLAC results

4.3.2 Simulation 5

As the theoretical prediction, a large number of uniform grids are needed to yield the accurate results without oscillaltion in the algorithm. Considering 30 intervals in one period are applied as Fig. 4.17 shown. The four plottings are input, output voltages, the voltage at node 2 and the branch current of inductor L_4. The input and output period equal to $5ns$. The execution time of the algorithm in this uniform case is $26.12s$.

Particularly, For the waveform of current in inductor L_4, the resulted waveforms do not match well to the results of Aplac. The reason for this discrepancy is unknown at this time.

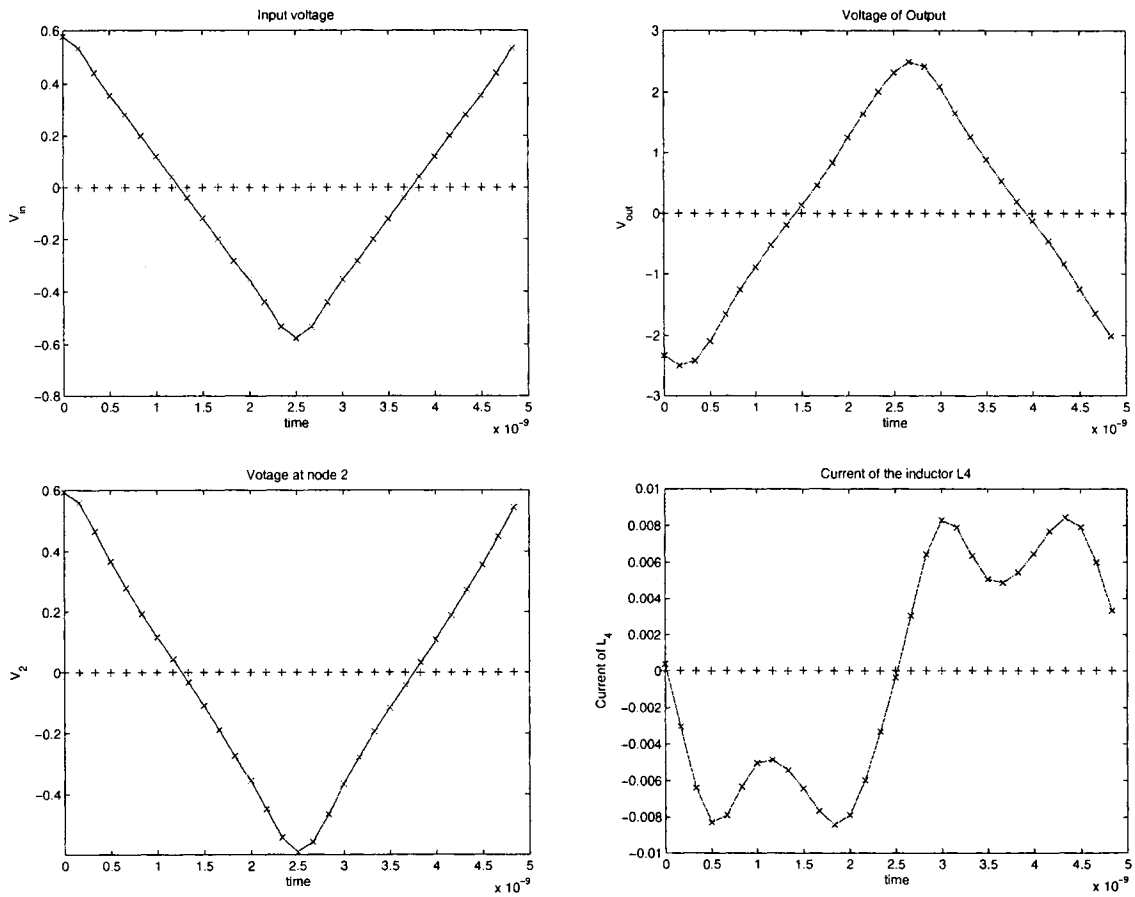

Figure 4.17: Example 2: Multiple adaptive algorithm in uniform grids (30 intervals)

4.3.3 Simulation 6

This simulation aims to evaluate the algorithm by non-uniform grids. In Fig. 4.20, the grid adaptation adjusts the distributed grid points efficiently. The results are closer to the reference data (Aplac) in Fig. 4.16, by using less grid points with non-uniform grids.

Considering 10 intervals are used in each period, Figure 4.19 shows the variation of the guide number when releasing the iteration number. In the figure, the guide numbers are small at 17, 34, 45, 73, 95 iterations. And the guide number is smallest when the iteration number equals to 73 in this particular case. As we discussed in the Simulation 2, we expect the guide number is as small as possible to obtain the higher accurate results. However, due to the expense of long time (73 iterations), we limit the iteration number at 45 to obtain the result as Figure 4.20 displays. The execution time of the algorithm in this non-uniform case is 306.41s.

Furthermore, the nonzero mapping of the matrix $A^T A$ is shown in Fig. 4.18 (10 non-uniform intervals). The number of nonzero elements is 22176, which is 12.28 percent of the number of total elements. Compared to the Fig.4.5 for the small size circuit (density: 27.84 percent), the density of this large circuit is decreased significantly. Because the sparsity of the matrices G and C in Eq.(3.7), results in the sparser matrix $A^T A$. The advantage is that the CPU storage and running time will be saved a lot when solving especially for large size circuits.

The adaptive strategy has the following advantages:
(1) After the grid moving, the collection extent of grids corresponding to each state variables is distinct.
(2) The grid positions move toward the region of 'high activity'.
(3) The waveforms can be followed even in the steep regions, such as for the input voltage, the bottom of triangular input can be achieved.

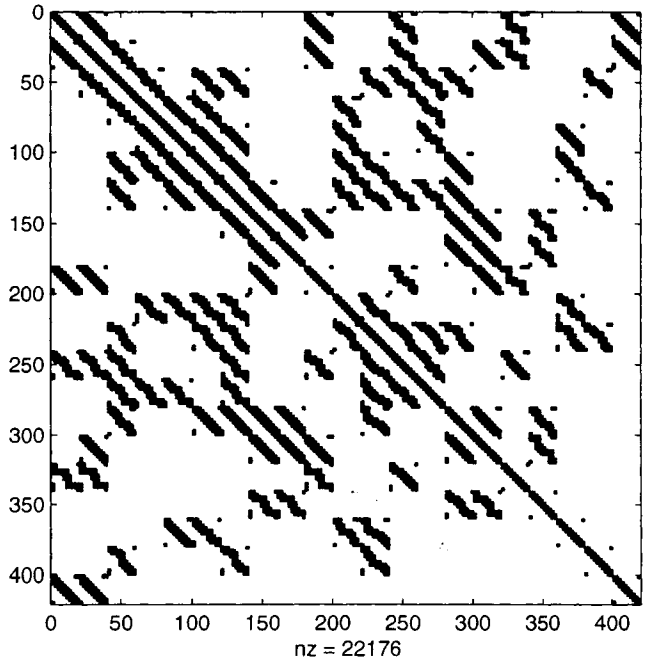

Figure 4.18: Example 2: The nonzero map of matrix $A^T A$ of 10 non-uniform intervals with the density of 12.28 percent

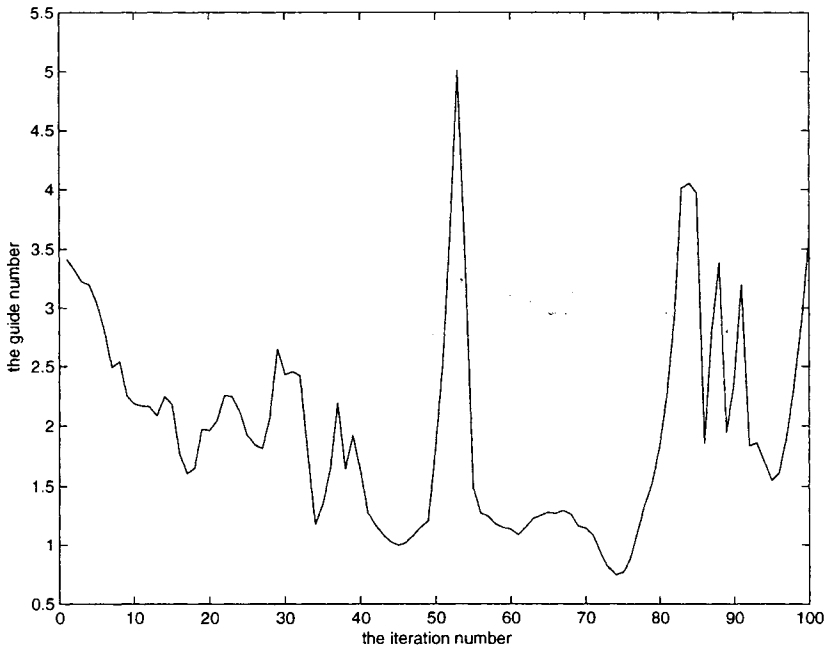

Figure 4.19: Example 2: The guided number with respect to the iteration number

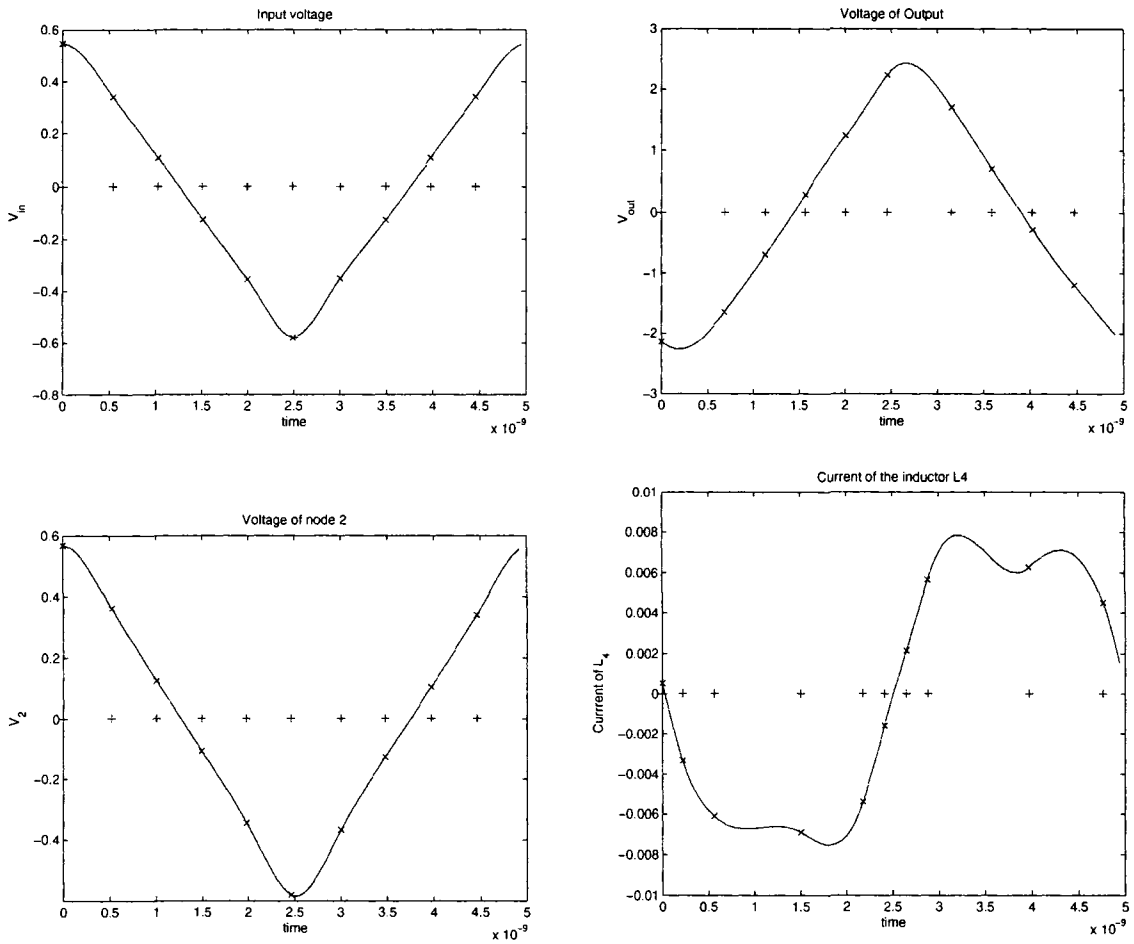

Figure 4.20: Example 2: Multiple adaptive algorithm in non-uniform grids (10 intervals)

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis documents the development and tests of a multiple adaptive algorithm to analyze the steady-state of circuits. Remarkably, this is the first time that an approach to obtain different adaptive grids simultaneously for each state variable is demonstrated.

In this research, we apply ABF effectively as fundamental elements to approximate the waveform of each state variable and then formulate state equations of the circuits by the transformation matrices B and B_p. Transformation matrices B and B_p are sparse and this facilitates the obtainment of numerical solutions.

The least squares method is used to solve the circuit equations. Although the matrices obtained by the least squares method are little denser than the ones by the collocation method, the unexpected oscillations can be significantly reduced by the least squares method.

By applying the non-uniform grids, we can achieve high accuracy with less grid points than the uniform grids case. From the analysis of the relationship of the iteration number and the guided number, we can limit the iteration number to speed up the algorithm computation

time.

Common applications of the algorithm cover the steady-state analysis and design of electronic circuit with periodic excitations, even for highly nonlinear circuits.

5.2 Future Work

In future undertaking, efforts can be concentrated on the following items:

Improve the simulation results and resolve discrepancies in the the feedback amplifier simulations. For example, for certain circuit solutions, the resulted waveforms are not matched perfectly to the results of Aplac, thus more work should be made to improve the matching.

A further study involves in testing the approach with nonlinear circuits and implementing the Newton method to acquire state variable solutions.

For larger circuits, since the size of equations becomes considerable, we could make use of sparse-matrix techniques to maintain the entire matrix in sparse form when formulating the circuit equations.

Another improvement would be to seek a method to reduce the number of interactions to achieve a set of non-uniform grids.

Finally, it is possible to use different number of intervals to simulate each state variable.

Bibliography

[1] P. Maffezzoni and A. Brambilla, "Envelope-following method to compute steady-state solution of electrical circuits", *IEEE Trans. on circuits and systems I Fundermental theory and applications*, vol. 50, pp. 407-417, March 2003.

[2] A. E. Cook and R. A. Duncan, "An adaptive node-moving algorithm for evolution type problems" *Appl.Math.Modeling*, vol. 16, pp. 431–438, 1992.

[3] J. O. Attia, "PSPICE and MATLAB for Electronics : An integrated Approach", CRC Press, 2002.

[4] A. Ushida and L. O. Ogborn, "Frequency-domain analysis of nonlinear circuits driven by multi-tone signals". *IEEE Trans. on Circuits and Systems*, vol. 31, pp. 766–779, September 1984.

[5] T. J. Brazil, "Simulation circuits and devices", *IEEE microwave magazine*, pp. 42–50, March 2003.

[6] C. S. Burrus, H. Guo, R. A. Gopinath, "Introduction to wavelets and wavelet transforms", Prentice Hall, 1998.

[7] C. E. Christoffersen , M. B. Steer, "State-variable-based transient circuit simulation using wavelets", IEEE Microwave and Guided Waves Letters, pp. 161–163, April 2001.

[8] C. K. Chui, "An introduction of Wavelets", Academic Press, 1992.

[9] Aplac Solutions Corp, "Manual of APLAC 7.91 - Circuit Simulation and Design Tool", vol. 1 and 2, Finland, 2003.

[10] W. Cai, D. Zhou ,W. Zhang, "An adaptive wavelet method for nonlinear circuit simulation". *IEEE Trans. on Circuits and SystemsII: Fundamental Theory and Application*, vol. 46, pp. 931–938, August 1999.

[11] W. Cai, D. Zhou, N. Chen, "A fast wavelet collocation method for high-speed vlsi circuit simulation", *IEEE Trans. on Circuits and Systems I*, vol. 46, pp. 920–930, August 1999.

[12] D. R. Frey, "A class of relaxation algorithms for finding the periodic steady-state solution in nonlinear system", *IEEE Trans. on Circuits and Systems I* , vol. 45, pp. 659–663, June 1998.

[13] C. A, J. Fletcher, "Computational Galerkin Method. Springer-Verlag, New York, 1964.

[14] G. Gonzalez, "Microwave Transistor Amplifiers Analysis and design", Prentice Hall, Upper Saddle River, NJ, 1997.

[15] D. V. Griffiths, I. M. Smith, "Numerical method for Engineers", Blackwell Scientific Publications, 2002.

[16] G.Welsch, H. Brachtendorf, "Numerical steady state analysis of electronic circuits driven by multi-tone signals", *In Electrical Engineering*, New York, Springer-Verlag, vol. 79, pp. 103–112, 1996.

[17] M. T. Heath, "Scientific Computing", McGraw-Hill, 1996.

[18] L. L. Ogborn, J. R. Parkuhurst, "Determining the steady-state output of nonlinear oscillatory circuits using multiople shooting", *IEEE Trans. Computer-Aided Design*, vol. 14 pp. 882–889, July 1995.

[19] J. Vlach, K. Singhal, "Computer methods for circuit analysis and design", Van Nostrand Reinhold, New York, 1983.

[20] M. Kakizaki, T. Sugawara, "A modified newton method for the steady-state analysis", *IEEE Transaction on Computer-aided Design*, pp. 662–667, October 1985.

[21] L. W. Nagel, " Spice2: A computer program to simulate semiconductor circuits". *PH.D. dissertation, Electr. Electr. Electron. Comp. Sci. dept., University of California, Berkeley, Electrical Research Lab.*, 1975.

[22] C. M. Snowden, M. B. Steer, J. Bandler, "Computer-aided design of rf and microwave circuit and systems", *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, No. 3, pp. 996–1005, March 2002.

[23] M. Nakhla, J. Vlach, "A piecewise harmonic balance technique for determination of periodic response of nonlinear circuit", *IEEE Trans. on Circuits and Systems*, CAS-23, No. 2, February 1976.

[24] D. M. Pozar, "Microwave Engineering", 3nd Edition, John Wiley and Sons Inc., January 2004.

[25] M. Nakhla, N. Soveiko, "Wavelet harmonic balance", *IEEE Microwave and Wireless components Letters I: Fundamental Theory and Application*, vol. 13, pp. 232–234, June 2003.

[26] H. Smith, N. R. Draper, "Applied Regression Analysis", 3rd edition, Wiley, 1998.

[27] P. R. Gray, P. J. Hurst, S. H. Lewis and R. G. Meyer, "Analysis and Design of Analog Integrated Circuits", 4th Edition, Wiley, 2001.

[28] P. J. C. Rodrigues, "Computer-Aided Analysis of Nonlinear Microwave Circuit", Artech House, Norwood, MA, 1998.

[29] N. Soveiko, M. Nakhla. "Steady state analysis of multitone nonlinear periodic circuits in wavelet domain", *2003 IEEE Microwave Symposium Digest*, pp. 2121–2124, 2003.

[30] S. Skelboe, "Computation of the periodic steady-state response of nonlinear circuits by extrapllation methods", *IEEE Trans. Circuits Syst.*, vol. 27 pp. 167–175, March 1980.

[31] T. J. Aprille, T. N. Trick, "Steady-state analysis of nonlinear circuits with periodic inputs", *proceedings of the IEEE*, vol. 60, No. 1, pp. 108–114, January 1972.

[32] A. Wenzler , E. Lueder, "Analysis of the periodic steady-state in nonlinear circuits using an adaptive function base", *1999 IEEE Int. Symp. on Circuits and Systems Digest*, pp. 1–4, June 1999.

[33] V. Wickerhauser, "Adapted Wavelet Analysis from Theory to Software", AK Peters, Boston, 1994.

[34] Y. Su, X. Zeng, S. Huang and D. Zhou. "An efficient sylvester equation solver for time domain circuit simulation by wavelet collocation method", *2003 IEEE Symp. on Circuits and Systems*, vol. 4, pp. 664–667, 2003.

[35] B. Hu, X. Li, "A wavelet-balance approach for steady-state analysis of nonlinear circuits", *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, No. 3, pp. 89–94, May 2002.

[36] A. Ushida, M. Takahashi, Y. Yamagami, Y. Nishio and K. Ogawa, "Analysis of communication circuits based on multidimensional fourier transformation", *IEEE Transaction on Computer-aided Design*, vol. 18, pp. 1165–1177, August 1999.

Appendix A

Design of the Microwave Feedback Amplifier

Based on Scattering parameters (S parameters) of the transistor and certain performance requirements, a systemic procedure is developed for the amplifier design. Some important relevant design parts will be addressed:

- Intrinsic and extrinsic parameters of the small-signal equivalent transistor model
- Design of feedback loop for flat gain $8dB$ from $10MHz$ to $0.8GHz$
- Biasing circuit
- Matching synthesis for lossless maximum power gain at frequency $0.7GHz$
- Optimization of Matching network over broadband frequency range

1. Linearized Transistor Model

The starting point of the work is to use a typical small-signal equivalent transistor model. Figure 1 depicts the distributed schematic of the transistor parameters. $BFS17W$'s plastic $SOT323$ (S-mini) package parameters and S parameters of the transistor (in a 50 Ω system) from $10MHz$ to $1GHz$ are given in the appendix B.

The next step, we need evaluate the S parameters to validate the working properties of the model, where S parameters are viewed as functions of frequency and characteristic impedance. The S parameters of the model in Fig.1 are compared with the measured S parameters in Fig.2 and Fig.4. Because the main reason is that we want to compare the simulations of the same object. Because S parameter of the model is close to the measured S parameter, then we can regard the model as our simulation object.

Quality	Formula	Value
Transconductance	$g_m = \frac{qI_c}{kT} = \frac{I_c}{V_T}$	0.152S
Input resistance	$r_{pi} = \frac{\beta_0}{g_m}$	656Ω
Output resistance	$r_0 = \frac{V_A}{I_c} = \frac{1}{\eta g_m}$	$22.5k\Omega$
Base-charging capacitance	$C_b = \tau_F g_m$	8.656pF
Base-emitter capacitance	$C_{pi} = C_b + C_{je}$	10pF
Collector-base junction capacitance	$C_\mu = \frac{C_{\mu o}}{(1-\frac{V_{BC}}{\psi_{oc}})^{n_c}}$	0.538pF
Collector-substrate junction capacitance	$C_{cs} = \frac{C_{cs0}}{(1-\frac{V_{sc}}{\psi_{oc}})^{n_s}}$	0.738pF

Table 1: Small-signal Forward-active parameters of $BFS17W$ model

Frequency (GHz)	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
K	0.061	0.091	0.4	0.585	0.731	0.860	0.961	1.02	1.095	1.148	1.168

Table 2: $BFS17W$ stability coefficient K at various frequencies

Since we want to compare simulation results from the same object, as long as all S parameters are turned out to be identical, it means the parameters in the model are valid and can be used in the following design.

As expected, Fig.2 and Fig.4 proved that the resulting S parameters from calculated small-signal transistor model are close to the measured S parameters which are seen as reference. thus we regard the small-signal model as our simulation object.

The stability of an amplifier is an important consideration in amplifier design. If the the amplifier is unstable, oscillations are possible when either the input or output port has a negative resistance. Check stability coefficient K by Eq.2 and found that the $BFS17W$ transistor is potentially unstable because K is smaller than 1 when the frequency is below $0.7MHz$ as Table 2 shown. One choice is to add a high value shunt resistor $R_2 = 300\Omega$ across the output of transistor to provide stability. The resistor R_2 dissipates some of the output power.

$$\triangle = S_{11} \cdot S_{22} - S_{21} \cdot S_{12}$$
$$K = \frac{1 + |\triangle|^2 - |S_{11}|^2 - |S_{22}|^2}{2 \cdot S_{21} \cdot S_{12}} \quad (1)$$

2. Negative Feedback Loop

Use of feedback is common since very early days in amplifier design. It is the primary mean of maintaining a circuit's constant power gain over a wide range of input signal's frequency range.

Negative Feedback is the main constituent to provide a flat gain. One way is to use a shunt resistor-inductor combination. The values are selected from Eq.2 and 3. The feedback resistor R_f is related to the scattering parameter S_{21} of the transistor:

$$R_f = Z_0(1 + |S_{21}|) = 175.5\Omega \quad (2)$$

On the other hand, L_2 is designed to improve the magnitude response above $0.6GHz$ such that provide the negative feedback. The L_2 is selected as

$$L_2 = \frac{R_f}{\omega}|_{f=600MHz} = 46.6nH \tag{3}$$

The resulting gain response with and without feedback loop are plotted in Figure 7 and 6 over the frequency from $10MHz$ to $0.8GHz$, it can be observed that the feedback significantly hold the gain in a certain value. In fact, to improve the performance, the feedback network elements (i.e. R_f, L_2) can be varied using trial and error, and a CAD method is necessary to optimize the feedback network design values. The counterpart be will discussed in section . The optimized $R_f = 116.73\Omega$, $L_2 = 22.175H$ help the gain flatness to $8dB$.

3. Biasing Network

Because a bias condition must be satisfied for our amplifier application, a biasing network is designed and supply the biasing voltage to a transistor over the broad range frequency. The purpose of biasing network in the AC signal amplifier is to provide enough quiescent current through the base to keep the transistor between the extremes of cutoff and saturation throughout the input signal's cycle.

Suppose biasing voltage $V_{CC} = 15V$, we may choose a typical biasing network structure and compute the component values. Next, force the $BFS17W$ DC collector-emitter voltage to the desired values $V_{CE} = 10V$and $I_C = 4mA$.

Fig.8 displays the circuit with the biasing network, stability and feedback components, where the biasing network is composed by R, R_b, R_{b1} and R_{b2}. The stability resistor is R_2, the feedback network includes R_f and L_2. C_{p1}, C_{p2}, C_{p3} and C_{p4} are coupling capacitors with values of $300pF$. L_1 and L_3 are choke inductors with values of $1000nH$.

For the biasing network, the component values $R = 1k\Omega$, $R_b = 14k\Omega$, $R_{b1} = 8.5k\Omega$, $R_{b2} = 1.5k\Omega$ are calculated as follows:

Look up $BFS17W$'s DC characteristics's table and get the typical DC current gain $h_{FE} =$

70 at $T_A = 25°C$, thus base current is given by

$$I_B = \frac{I_C}{h_{FE}} = \frac{4mA}{70} = 57\mu A \tag{4}$$

Assume the current flowing through R equals to $5mA$, then

$$R = \frac{V_{CC} - V_{CE}}{5mA} = \frac{(15-10)V}{5mA} = 1k\Omega \tag{5}$$

Because the voltage at node 1 V_1 is larger than conductance voltage of the transistor V_{BE} and V_{BE} has fixed value $0.7V$, assume V_1 = 1.5 V, then

$$R_b = \frac{V_1 - V_{BE}}{I_B} = \frac{1.5V - 0.7V}{57\mu A} = 14k\Omega \tag{6}$$

$$\frac{V_{CE}}{R_{b1} + R_{b2}} \simeq (5-4)mA \Rightarrow R_{b1} + R_{b2} = \frac{10V}{1mA} = 10k\Omega \tag{7}$$

$$R_{b2} = \frac{V_1}{(5-4)mA} \simeq \frac{1.5V}{1mA} = 1.5k\Omega \tag{8}$$

$$R_{b1} = 10 - R_{b2} = 8.5k\Omega \tag{9}$$

The biasing network keeps the transistor would operate in the chosen operating point over variations in transistor parameters and moderate temperature.

4. Input and Output Matching Network at a Certain Frequency

Besides the previous design, external impedance matching networks at the input and output stages are also required when we intend to deliver the maximum power gain and obtain the reasonable VSWR (voltage standing wave ratio) as well. Therefore, the impedance of the BJT should be matched to the input and load impedance in terms of lossless matching

S parameter at $0.7GHz$	Magnitude	Phase
S_{11}	0.4597	-154.1
S_{21}	3.041	83.2
S_{12}	0.0678	53.6
S_{22}	0.6445	-21.8

Table 3: $BFS17W$ $0.6GHz$ Scattering parameters at biasing point *10V* and *4mA*)

network over wide bandwidth of interest. Choose the LC (inductance-capacitance) lumped elements to establish matching network as Fig.9 shows.

Because the maximum applicable bandwidth of the transistor is up to $1GHz$, design L and C values at the specific frequency at $0.7GHz$ (or any other frequency below $1GHz$), followed by an optimization process from $10MHz$ to $0.8GHz$ to achieve a broad bandwidth. However, the lower gain is must be comprised. L and C values can be determined using the following procedure :

Firstly, obtain the S parameter of the transistor with feedback and stability networks. Calculate the admittance parameter Y_1 for the transistor model at $0.7GHz$, Y_2 for the stability resistor and Y_3 for feedback network. The S parameter of the transistor at operation point *10V 4mA* is given by Table 3.

$$Y_1 = Y_0(I + S)^{-1}(I - S)$$

Consider the definition of Y matrix

$$Y_{11} = \frac{I_1}{V_1}|_{V_2=0};\ Y_{12} = \frac{I_1}{V_2}|_{V_1=0}$$

$$Y_{21} = \frac{I_2}{V_1}|_{V_2=0};\ Y_{22} = \frac{I_2}{V_2}|_{V_1=0}$$

By Y parameter rule:

$$Y = Y_1 + Y_2 + Y_3$$

Therefore, the resulting S parameter of the entire circuit is

$$S = (Y_0 \cdot I + Y)^{-1}(Y_0 \cdot I - Y)$$

Calculate Γ_s and Γ_L by,

$$\triangle = S_{11}S_{22} - S_{12}S_{21}$$

$$B_1 = 1 + |S_{11}|^2 - |S_{22}|^2 - |\triangle|^2$$

$$B_2 = 1 + |S_{22}|^2 - |S_{11}|^2 - |\triangle|^2$$

$$C_1 = S_{11} - \triangle \cdot S_{22}^*$$

$$C_2 = S_{22} - \triangle \cdot S_{11}^*$$

$$\Gamma_s = \frac{B_1 \pm \sqrt{B_1^2 - 4|C_1|^2}}{2C_1}$$

$$\Gamma_L = \frac{B_2 \pm \sqrt{B_2^2 - 4|C_2|^2}}{2C_2}$$

Locate Γ_s and Γ_L on Smith Chart. Respecting to input and output matching, obtain the imaginary part jB and jX by using Smith Chart, then based on the prescribed topology of the matching network as Fig.9 to calculate the values of lumped elements L and C.

$$Z_0 = 50\Omega; f = 0.7GHz;$$

$$C = \frac{-1}{2\pi f X Z_0}$$

$$L = \frac{-Z_0}{2\pi f B}$$

Finally, the topology is shown in Fig. 9

$$C_1 = 3.5nF; L_1 = 4.2nH$$

$$C_2 = 912pF; L_3 = 20\mu H$$

4. Optimization

Better performance can be accomplished by optimizing the overall amplifier design with good VSWR and also achieve the flat gain over the wide bandwidth. This analysis is impractical in analytical form. A gradient optimization method in APLAC is used to deal with optimization of the matching circuits. Additionally, an analysis tool — Smith Chart, provides a very useful graphical aid to do the analysis. The Smith Chart is a plot of all passive impedances by means of reflection coefficient chart of unit radius.

Considering the frequency range of the amplifier, we optimize from $10MHz$ to $0.8GHz$. The final optimized gain is determined by S_{21} and is shown in Fig.10. The optimized parameter values are listed in Table 4.

Component	Before Optimization	After Optimization	Unit
C_1	3.5n	2.716n	F
L_1	4.2n	0.4n	H
R_f	175.5	116.73	Ω
L_2	46.6n	22.175n	H
C_2	912p	676.164p	F
L_3	20μ	9.636μ	H

Table 4: Circuit optimized component values

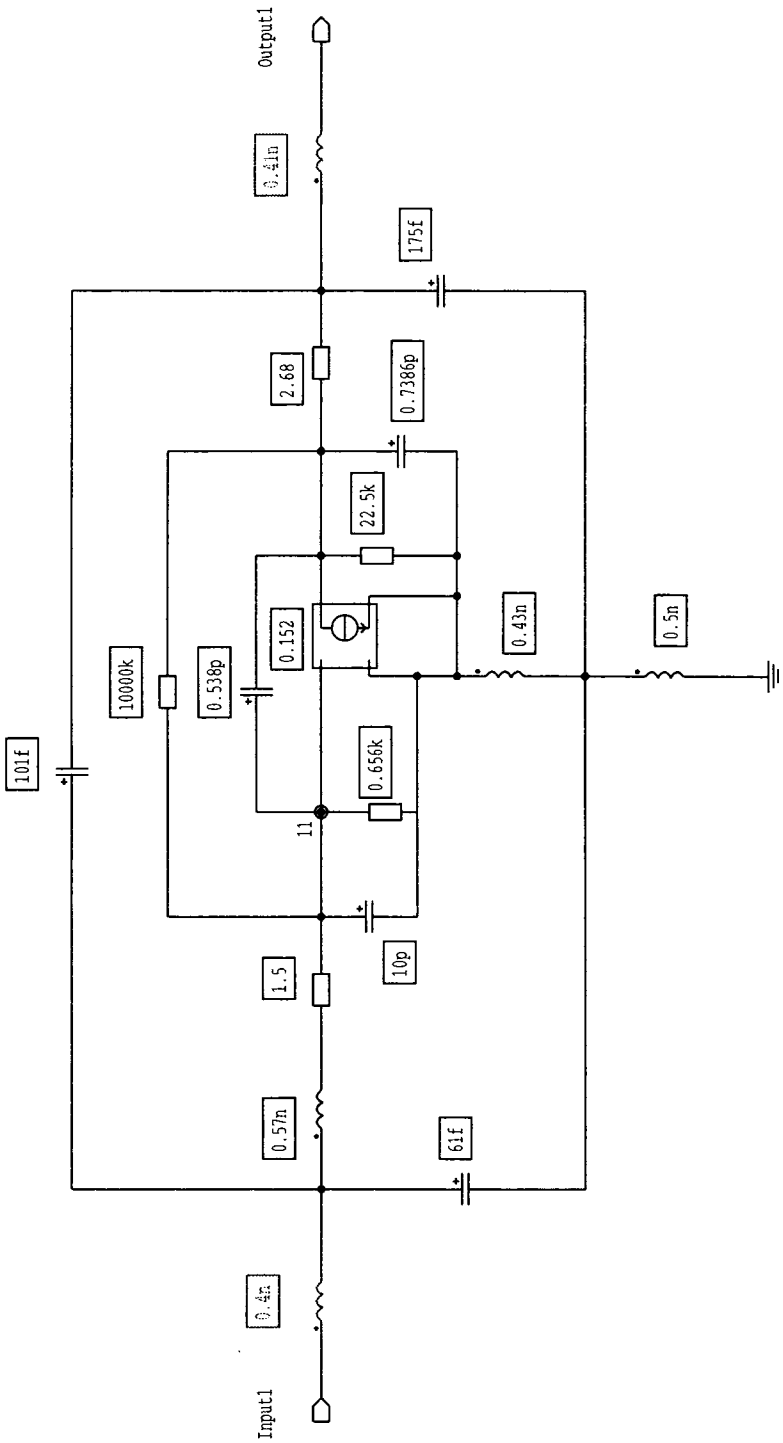

Figure 1: Small-signal equivalent model of BJT

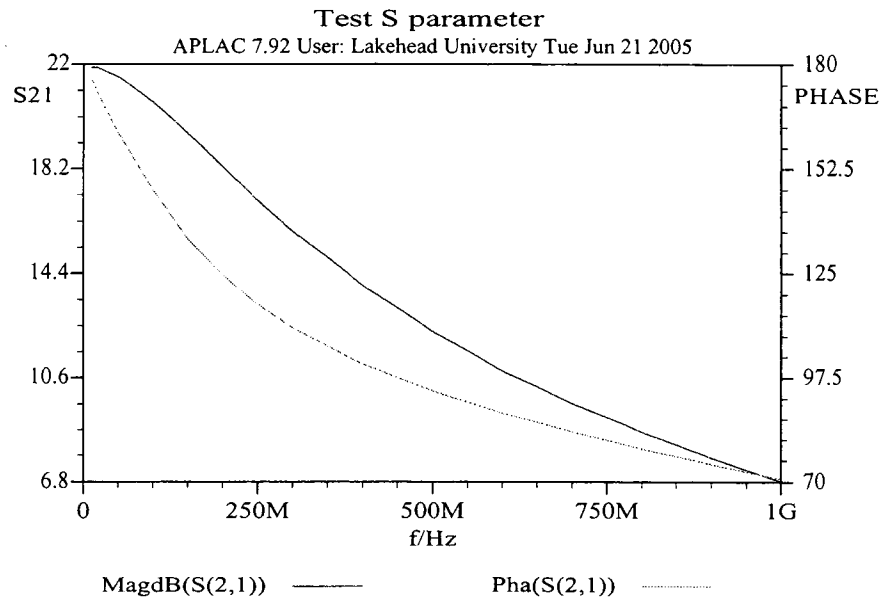

Figure 2: Power gain S_{21} of Aplac package model

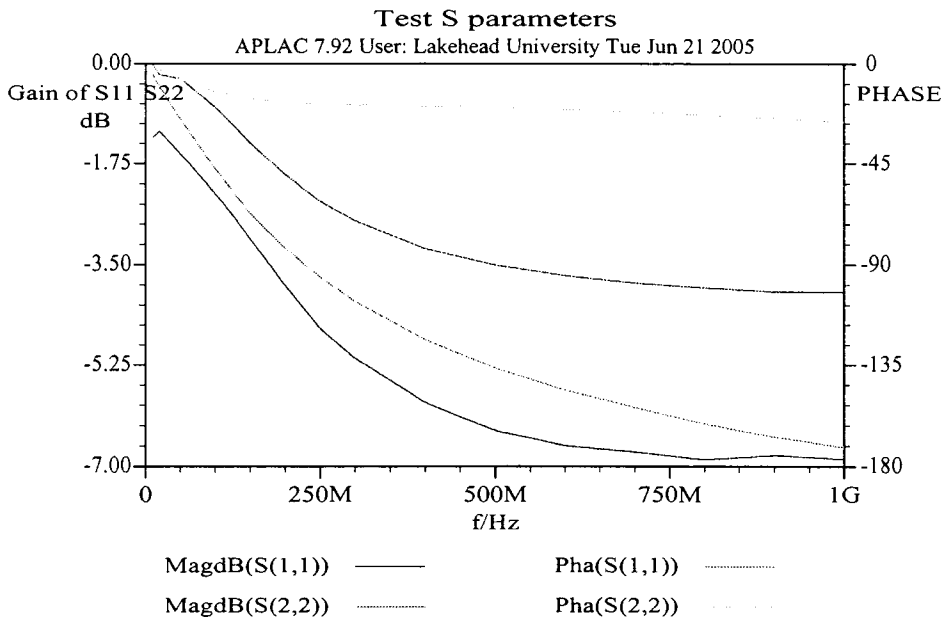

Figure 3: S_{11}, S_{22} of Aplac package model

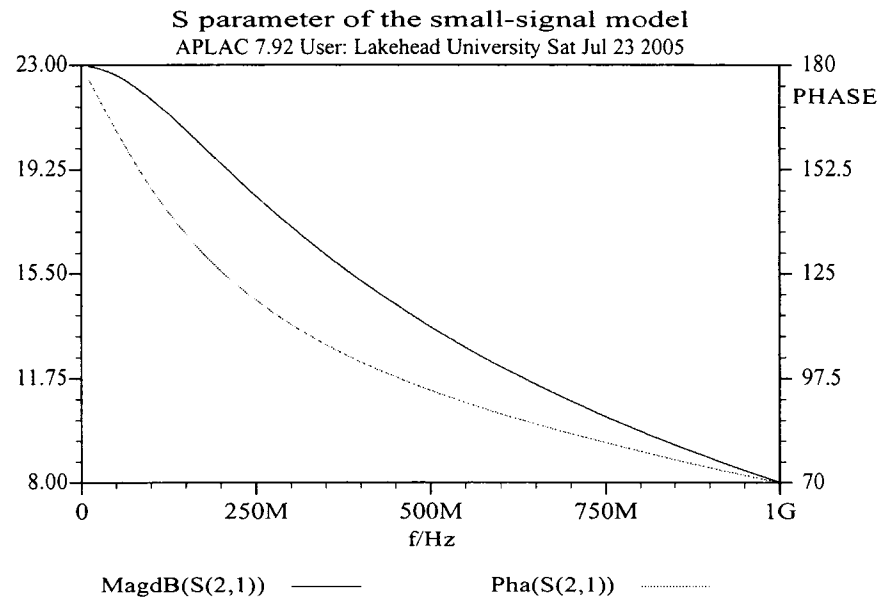

Figure 4: Power gain S_{21} of the small-signal model

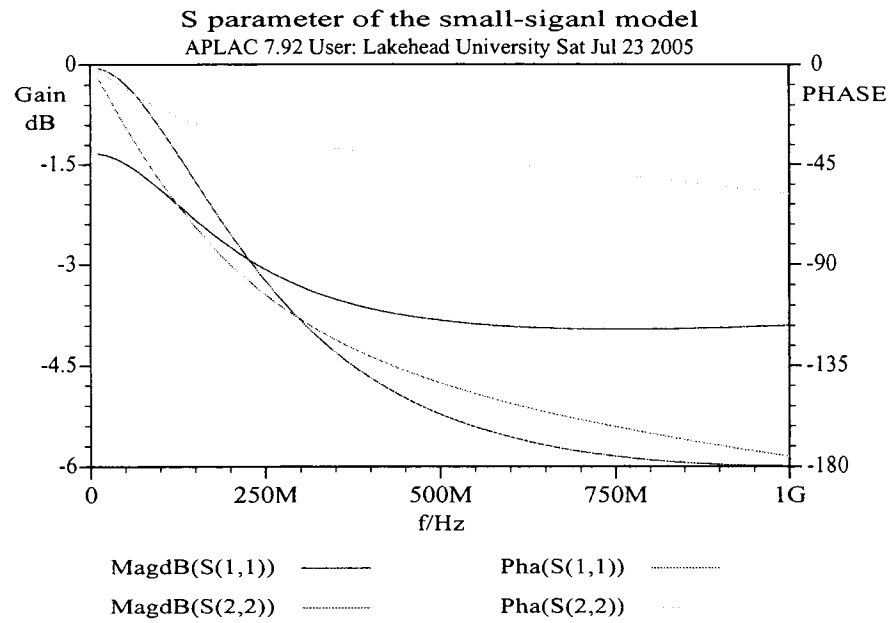

Figure 5: S_{11}, S_{22} of the small-signal model

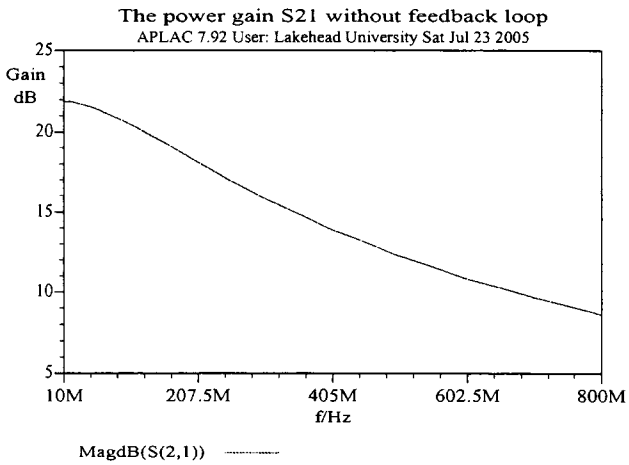

Figure 6: Power gain without the feedback

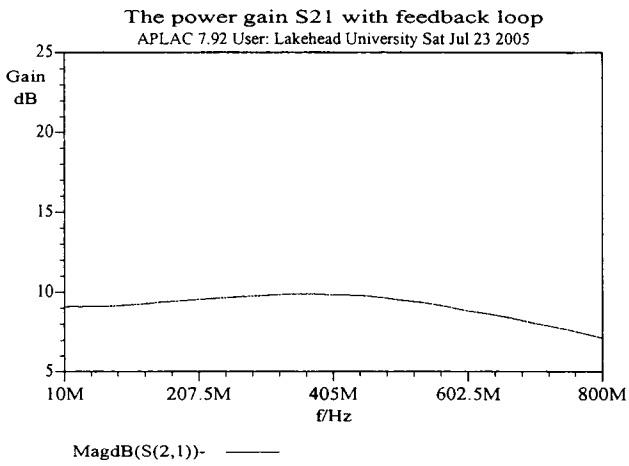

Figure 7: Power gain with the feedback

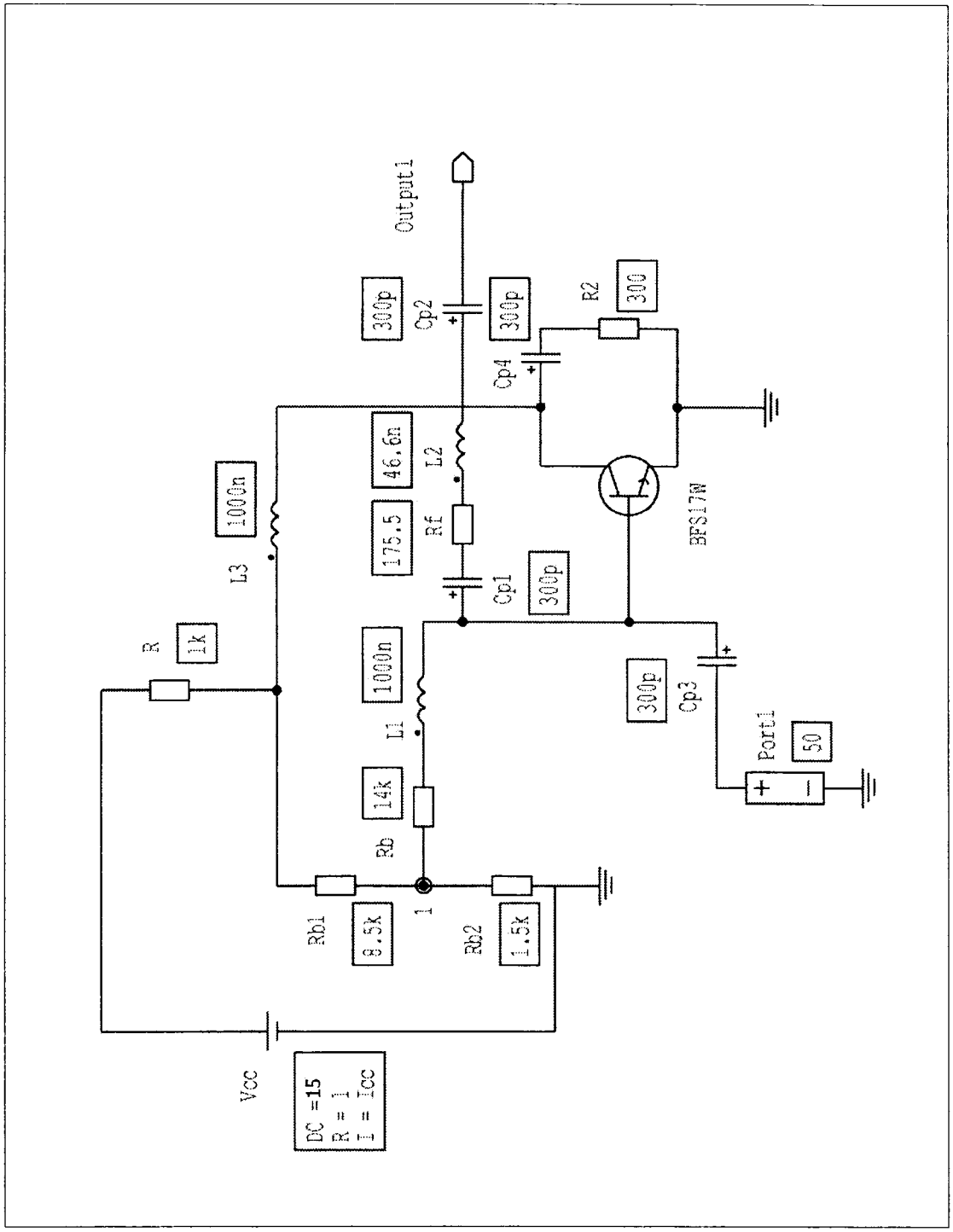

Figure 8: The amplifier circuit with the bias network, stability and feedback components

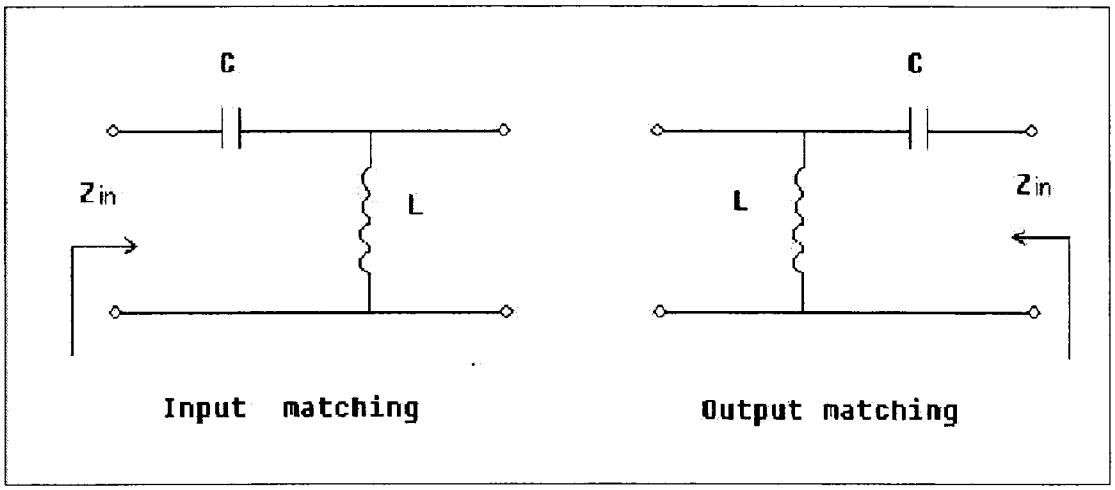

Figure 9: Topology of input and output matching network

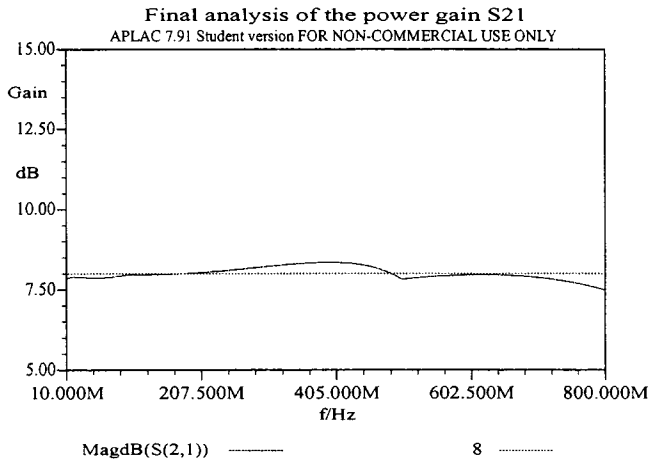

Figure 10: Final analysis of the power gain S_{21}

Appendix B

Table 1: Common Emitter S parameters of $BFS17W$ at $V_{CE} = 10V$ $I_C = 4mA$

Table 2: PSpice model of $BFS17W$

Table 3: Package parameters of the $BFS17W$ transistor

$frequency$	S_{11}	S_{21}	S_{12}	S_{22}

GHz	MAG	ANG	MAG	ANG	MAG	ANG	MAG	ANG
0.010	0.8619	-5.0	12.419	176.7	0.0028	85.5	0.9977	-1.0
0.020	0.8724	-10.4	12.419	172.6	0.0055	85.1	0.9779	-2.8
0.050	0.8333	-25.1	11.949	162.1	0.0141	79.1	0.9701	-7.0
0.100	0.7705	-47.4	10.758	147.0	0.0253	67.7	0.9159	-12.5
0.150	0.7031	-67.3	9.440	134.1	0.0339	58.5	0.8510	-15.7
0.200	0.6408	-82.7	8.186	124.5	0.0402	55.3	0.7994	-17.4
0.250	0.5874	-95.6	7.109	116.9	0.0437	51.7	0.7578	-18.3
0.300	0.5534	-106.5	6.247	110.7	0.0472	50.5	0.7288	-18.7
0.400	0.5061	-123.7	4.979	101.1	0.0530	49.6	0.6894	-19.3
0.500	0.4793	-136.4	4.113	94.0	0.0574	50.3	0.6677	-19.8
0.600	0.4652	-146.3	3.482	88.2	0.0622	51.8	0.6537	-20.7
0.700	0.4597	-154.1	3.041	83.2	0.0678	53.6	0.6445	-21.8
0.800	0.4526	-161.2	2.695	78.6	0.0721	55.2	0.6380	-23.3
0.900	0.4566	-167.0	2.413	74.6	0.0764	56.9	0.6329	-24.6
1.000	0.4533	-171.6	2.191	70.8	0.0825	58.6	0.6326	-26.2

Table 5: Common Emitter S parameters of $BFS17W$ at $V_{CE} = 10V$ $I_C = 4mA$

Quality	Value	Quality	Value
IS	0.480F	NF	1.008
BF	99.655	VAF	90.000
IKF	0.190	ISE	7.490F
NE	1.762	NR	1.010
BR	38.400	VAR	7.000
IKR	93.200M	ISC	0.200F
NC	1.042	RB	1.500
IRB	0.100M	RBM	1.200
RE	0.500	RC	2.680
CJE	1.325P	VJE	0.700
MJE	0.220	FC	0.890
CJC	1.050P	VJC	0.610
MJC	0.240	XCJC	0.400
TF	56.940P	TR	1.000N
PTF	21.000	XTF	68.398
VTF	0.600	ITF	0.700
XTB	1.600	EG	1.110
XTI	3.000	KF	1.000F
AF	1.000		

Table 6: PSpice model of $BFS17W$

Quality	Value
LBI	0.57nH
LCI	0nH
LEI	0.43nH
CCB	101fF
CCE	175fF
CBE	61fF
LBO	0.4nH
LCO	0.41nH
LEO	0.5nH

Table 7: Package parameters of the $BFS17W$ transistor