Southern Methodist University

SMU Scholar

Electrical Engineering Theses and Dissertations

Electrical Engineering

Fall 2022

An Efficient Integrated Circuit Simulator And Time Domain Adjoint Sensitivity Analysis

Jiahua Li
Southern Methodist University, li_jiahua_java@163.com

Follow this and additional works at: https://scholar.smu.edu/engineering_electrical_etds

Part of the Computational Engineering Commons, and the Electrical and Electronics Commons

Recommended Citation

Li, Jiahua, "An Efficient Integrated Circuit Simulator And Time Domain Adjoint Sensitivity Analysis" (2022). *Electrical Engineering Theses and Dissertations*. 62.
https://scholar.smu.edu/engineering_electrical_etds/62

This Dissertation is brought to you for free and open access by the Electrical Engineering at SMU Scholar. It has been accepted for inclusion in Electrical Engineering Theses and Dissertations by an authorized administrator of SMU Scholar. For more information, please visit http://digitalrepository.smu.edu.

AN EFFICIENT INTEGRATED CIRCUIT SIMULATOR AND TIME DOMAIN ADJOINT SENSITIVITY ANALYSIS

Approved by:

____
Dr. Ronald Rohrer – Committee Chairman
Professor of ECE department

____
Dr. Duncan Macfarlane
Professor of ECE department

____
Dr. Jennifer Dworak
Professor of ECE department

____
Dr. Peter Raad
Professor of ME department

____
Dr. Prasanna Rangarajan
Assistant Professor of ECE department

AN EFFICIENT INTEGRATED CIRCUIT SIMULATOR AND TIME DOMAIN ADJOINT SENSITIVITY ANALYSIS

A Dissertation Presented to the Graduate Faculty of the

Lyle School of Engineering

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

with a

Major in Electrical and Computer Engineering

by

Jiahua Li

B.S., ME, Harbin Institute of Technology, 2012
M.S., ECE, Southern Methodist University, 2016

December 17, 2022

Copyright (2022)

Jiahua Li

All Rights Reserved

ACKNOWLEDGMENTS

During the past five years, a lot of things changed my life. On the way of proceeding the research, my doubt, hesitation, anxiety, and uncertainty became confidence, creativity, and hope. Now, I know I made the best decision to be a PhD student at SMU and I know I could not make it without a great deal of help.

First of all, I would like to express my sincere appreciation to my advisor Dr. Ron Rohrer for his advice, support and confident attitude that guided me through the development of the research work presented here. His insight in discovering the topic of research that not only triggers the interest of academics but also the industries, and his connections secured my intellectual curiosity so that I always had access to the correct source of knowledge.

I would like to thank Kiran Gullapalli and Brian Mulvaney for providing the internship opportunity at NXP and leading me to appreciate the industrial approach to the production of EDA tools.

Finally, I would like to thank all my family members for their encouragement and patience. Their support that crossed the Pacific Ocean means a lot to my abroad life.

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of the requirements for an advanced degree at Southern Methodist University and is deposited in the university library to be made available for borrowers under the rules of the library.

Brief quotations from this dissertation are allowable without special permission, provided that an accurate acknowledgement of the source is made. Request for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

Li, Jiahua

B.S., Harbin Institute of Technology, 2016
M.S., Southern Methodist University, 2018

An Efficient Integrated Circuit Simulator and Time Domain Adjoint Sensitivity Analysis

Advisor: Ron Rohrer

Doctor of Philosophy conferred: December 17, 2022

Dissertation completed: December 17, 2022

Electronic Design Automation (EDA) algorithms provide irreplaceable support for the definition, design, implementation, and verification of semiconductor integrated circuits because modern designs require novel analysis tools. The complexity of state-of-the-art devices makes the run times unaffordable unless accurate and efficient simulations are provided.

This dissertation proposes a static-driven integration method which enhances the efficiency and accuracy of transient analysis and a time domain adjoint sensitivity method which in only a single simulation calculates the sensitivities of all the node voltages with respect to all the designated parameters. The applications of both methods include pseudo transient direct current biasing convergence and time domain noise analysis are explored. The error analysis of the time domain adjoint sensitivity is thoroughly studied to ensure the accuracy of the transient sensitivity responses.

The proposed method has been developed as an extension of XYCE, which is an open-source derivative of Simulation Program with Integrated Circuit Emphasis (SPICE). Various circuit examples which are constructed with industry standard models are tested on the platform.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Electronic Design Automation.	EDA
Simulation Program with Integrated Circuit Emphasis.	SPICE
Direct Current.	DC
Ordinary Differential Equation.	ODE
metal-oxide-semiconductor field-effect transistor.	MOSFET
Kirchhoff’s Voltage Law.	KVL
Kirchhoff’s Current Law	KCL
Forward Euler.	FE
Backward Euler	BE
Voltage Source.	vsrc
Current Source.	isrc
Newton-Raphson	NR
Integrated Circuit.	IC
Very Large-scale Integration.	VLSI
Truncation Error	TE

This is dedicated to my parents.

CHAPTER 1

1.1 Research Motivation and Contribution

The techniques of the integrated circuit (IC) simulation were introduced in the 1960s to supplement the circuit design engineer's hand calculations. Back then, the design of an integrated circuit with tens of transistors was too cumbersome to be affordable because of the complicated nonlinearity and each new generation of circuit has become ever more complex. Based on that, a computer-aided pre-manufacturing circuit simulator had been developed to keep up with the blooming demands. Such a simulator solves a set of nonlinear ordinary differential equations (ODE's) that describe a circuit's behavior. With the availability of parallel processors [1], hierarchical modeling [2], fast storage element [3] and the advent of nano scale transistors [4], circuit simulators run much faster than they did then. In the final analysis, circuit simulators become an indispensable aid to the design process, and they have even become the only thing to be trusted.

However, circuit simulators still barely keep pace with the increasing number of transistors and the complexity of modern IC. As the semiconductor processes are upgraded, the modelling of a device's physical properties has become more sophisticated than ever [5-7] which requires considerable run time in model evaluation. Take a single metal-oxide-semiconductor field-effect transistor (MOSFET) for example. The basic physical variables of its characteristic modelling are not enough to describe the short-channel effect [8] as it enters nano scale and supplemental parameters must be introduced to fit the experimental data [9]. Also, the parasitic elements [10] brought by MOSFET can easily boost the computation and memory cost in post-layout simulation which may slow down or lead to failure of simulations.

As the circuits become more vulnerable to process variations, multiple verification tools have been developed to ensure the reliability of designs [11-13]. Circuit designers need to statistically simulate the probability that a circuit does not meet the performance metric. Circuit testers run simulations over all potential faulty/nominal responses to improve the fault detection coverage. Some of the testers proceed to diagnose the physical location of those potential faults. Circuit manufacturers allocate enormous computational resources to estimate/improve the yield. Circuit machine learning players [14-15] also abuse the simulator to build their databases. It is not rare to experience simulations that take a few days or even weeks to finish.

In the meanwhile, more Electronic Design Automation (EDA) tools have been applied to help efficiently deliver a circuit design from idea to product, such as, auto-layout tool, design transfer tool, design rule check tool, etc. Unlike other general tools, they are facing challenges every time when a new process is released which means more ancillary patches are added.

Yet, the numerical technique scheme has not evolved with the scaling of circuits and the same ODE solver has been used to resolve a matrix with size of one hundred by one hundred in the past and now one million by one million [16]. Necessary statistical simulation directly deploys a few thousands of independent threads and the whole statistical simulation fails if one of the threads does not achieve convergence. Thus, a competitive integrated circuit simulator should enable more practical functionalities and upgrade the existing algorithms. This dissertation shows a novel way of running DC analysis, transient analysis, time domain sensitivity analysis and related applications. The content includes the capability of applying step functions as input excitations instead of steep ramp functions. The remaining transient simulation is supported by the static-driven integration method [17] after the step response as initial state is obtained; a time domain adjoint sensitivity analysis is performed along with the transient progression and all the node voltages' sensitivity with respect to all the parameters of interest are available in one simulation. Insight into the proposed algorithms, their efficiency improvement, numerical error analysis [18] and potential applications are provided. Simulations accommodate industry standard devices, and even customized devices in terms of the SPICE [19-21] format.

1.2 Dissertation Organization

This chapter provides an overview of IC simulation including the motivations, challenges, and a brief introduction to the proposed simulator. More details will be elaborated in the following chapters. In chapter II, the technical background of an industry standard simulator is introduced in the sequence of device modelling, netlist formation, system matrix construction and fundamental DC, AC, Transient algorithms for various analyses. An efficient transient methodology is proposed with a step function as input excitation in chapter III which uses the static driven method as the time integration approximation and the application of using the proposed static driven method to calculate the DC convergence point is discussed. In chapter IV, the proposed time domain adjoint sensitivity analysis shows the possibility of attaining all parameters' time domain sensitivities to be available withing one simulation. Related discussions of its complexity and error are compared with the incremental sensitivity, and a novel time domain noise analysis based on adjoint sensitivity is explored. Chapter V concludes the dissertation.

CHAPTER 2

Traditional integrated circuit simulation starts with the modeling of devices which is abstractly represented by the current as a function of voltage $i=f(v)$ or the voltage as a function of current $v=f(i)$. After all devices are declared in the circuit, an abstracted topology relationship of these devices called a netlist is formed. As the availability of each device's branch function and each branch's connectivity, the next step is combining all the independent branch functions to form the corresponding system matrix in terms of nodal analysis. Different analyses (time domain/frequency domain) employ different representations of the system matrix. During this process, nonlinear elements become linearized, and a great deal of matrix manipulations are applied to the linearized system until the result converges. Normally, most of the computational power entailed is spent on matrix manipulations.

This chapter will focus on introducing how the conventional analysis being generated in industrial standard IC simulator. The process starts with the definition of devices, and the potential method of improvement will be touched upon during the introduction.

2.1 Device Modelling

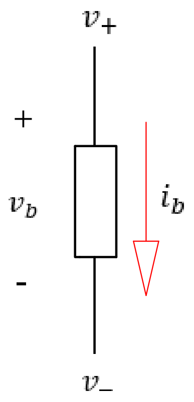


Figure 2.1 Branch information for a generic two-terminal device with associated reference directions.

In general, there are two types of devices: memory devices (capacitor/inductor) and memoryless devices (resistor/transistor), we use *q* and *f* to represent their voltage-current functions correspondingly. The voltage-current relations for any two-terminal devices are shown in Fig. 2.1 and the positive (+) reference for the branch voltage (v_b)/branch current (i_b) is with the tail of the reference arrow, the negative (-) reference is with the head of the reference arrow. As the direction being defined, branch current i_b can be represented by the function of branch voltage v_b as $i_b=f(v_b)$ or $i_b=q(v_b)$. The voltage-current relations of typical two-terminal devices are summarized as follows.

Table 2.1 Characteristic function of two-terminal devices

Types of devices	Resistor	Capacitor	Inductor	Diode
Characteristic function	$i_b = \frac{v_b}{R}$	$i_b = C \cdot \frac{d(v_b)}{dt}$	$v_b = L \cdot \frac{d(i_b)}{dt}$	$i_b = I_s \cdot [e^{\frac{v_b}{\eta v_T}} - 1]$

It should be noted that with the associated reference directions power. $p_b=v_b\cdot i_b$, is delivered to the element from the rest of the circuit if $p_b>0$; if $p_b<0$ power is being delivered to the rest of the circuit from the element.

For multiple-terminal devices, such as the N-channel MOSFET shown in Fig.2.2(a), the abstracted voltage-current branch function is shown in Fig.2.2(b) with the source node being the reference node. The equation for the branch current from the gate node to the source node is zero and the branch current from the drain node to the source node i_{ds} is described as follows according to the Schichman and Hodges model [22].

$$i_{ds}=0 \qquad \text{cutoff region } (v_{gs}<v_{th}) \text{ (1a)}$$

$$i_{ds}=\beta\cdot[(v_{gs}-v_{th})\cdot v_{ds}-\frac{v_{ds}^{\ 2}}{2}]\cdot(1+\lambda\cdot|v_{ds}|) \qquad \text{Triode region } (0<v_{ds}<v_{gs}\text{-}v_{th}) \text{ (1b)}$$

$$i_{ds}=\frac{\beta}{2}\cdot[v_{gs}-v_{th}]^2\cdot(1+\lambda\cdot|v_{ds}|) \qquad \text{saturation region } (0<v_{gs}\text{-}v_{th}<v_{ds}) \text{ (1c)}$$

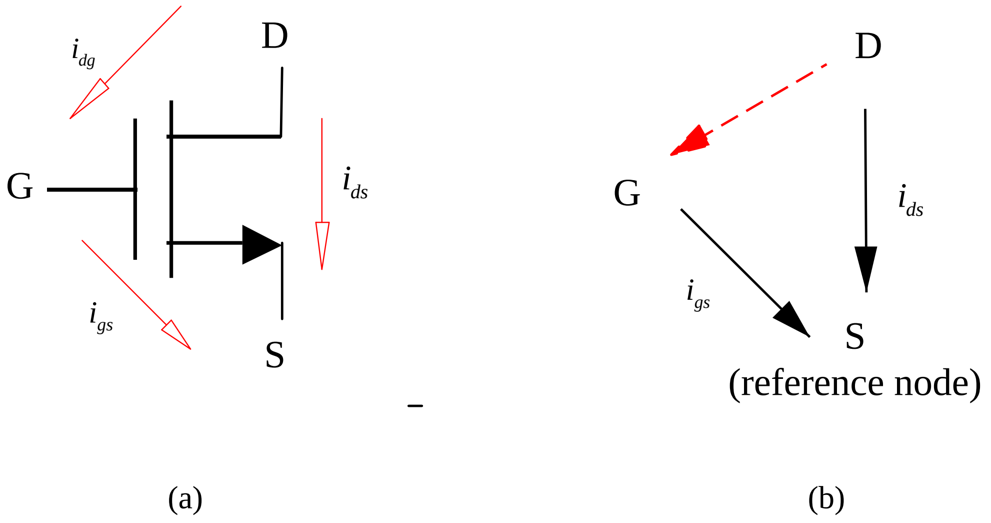


Figure 2.2 N-channel MOSFET and its abstracted branch information.

As a reference node of any multi-terminal devices is defined, a *n*-terminal device can be described in terms of (*n*-1) branches with the currents of such branches from every node to the defined reference node. For example, if the source node in Fig. 2.2(a) is selected to be the reference node, then the branch currents i_{gs} and i_{ds} are sufficient to describe this three-terminal device and the branch current i_{dg} becomes redundant.

2.2 Nodal Analysis

Now, any *n*-terminal devices can be described by (*n*-1) branch functions, and nodal analysis is used to represent the connectivity of those branches. Any device can be simplified in terms of multiple branches, in general, these branches have a positive (+) reference node called the 'from node' and a negative (-) reference node called the 'to node'.

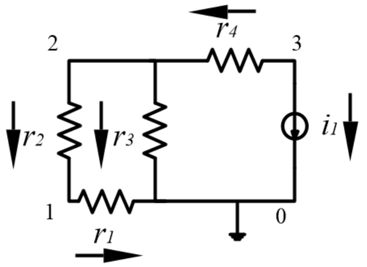


Figure 2.3 A simple circuit example.

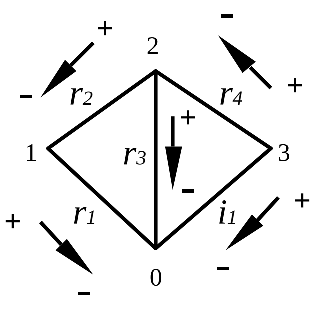


Figure 2.4 The topology abstraction of Fig.2.3.

Consider a simple circuit shown in Fig. 2.3, its connections of all the branches can be abstracted as in Fig. 2.4 after the global/local reference node is determined; the ground node is always the global reference node. Based on the pre-defined ‘from node’ and ‘to node’ of every branch, the equivalent netlist description is shown in Table 2.2. Now, all the connectivity of all of branches is available.

Table 2.2 The netlist representation of circuit shown in Fig.2.3

Branch type/name	From node	To node	Value
r_1	1	0	1.0
r_2	2	1	1.0
r_3	2	0	1.0
r_4	3	2	1.0
i_1	3	0	1.0

2.3 System Matrix Forming

Now that we have all necessary information to solve the circuit in Fig. 2.3, including the voltage-current function of each device and their connections. Before attaining the algebraic matrix and solving it, we first need each individual algebraic expression which are derived from Kirchhoff’s Voltage Law (KVL) and Kirchhoff’s Current Law (KCL). If a branch current leaves a node is positive and vice versa, then, based on KCL, each node’s expression of the circuit in Fig. 2.3 can be represented in (2). But the global reference node’s expression in (2) is a linear combination of the rest, thus, after taking out the ground node, the rest (2b), (2c)&(2d) can be expressed in matrix form in (3) where ***f*** is a vector which stores each nodal expression [*f₀*, *f₁*, *f₂*, … , *fₙ*], ***v*** and ***e*** are the vectors of node voltages and input excitations.

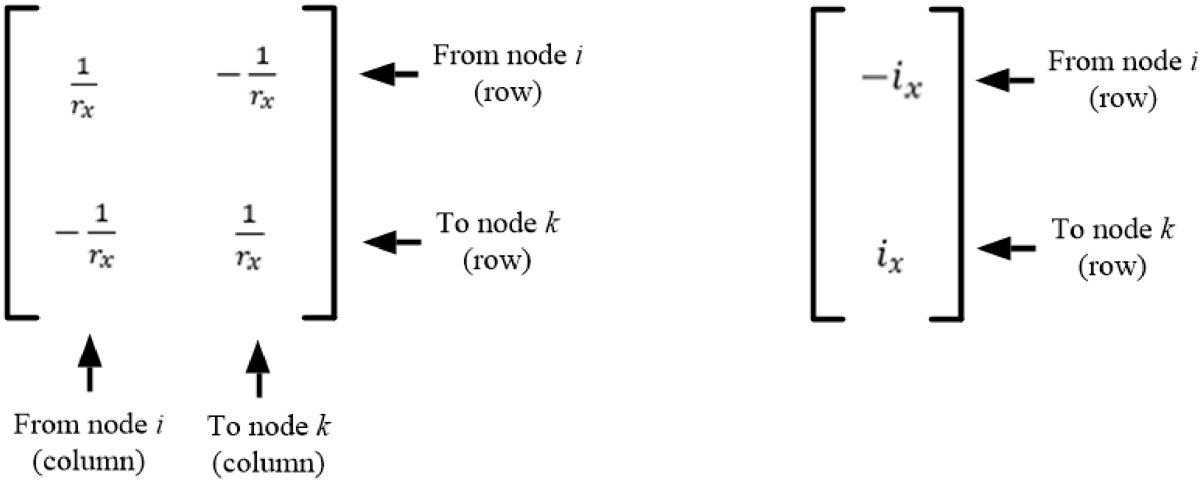
$$\text{node0:}\ \ f_0 = -i_{r1} - i_{r3} - i_1 = -\frac{v_1 - v_0}{r_1} - \frac{v_2 - v_0}{r_3} - i_1 \tag{2a}$$

$$\text{node1:}\ \ f_1 = i_{r1} - i_{r2} = \frac{v_1 - v_0}{r_1} - \frac{v_2 - v_1}{r_2} \tag{2b}$$

$$\text{node2:}\ \ f_2 = i_{r2} + i_{r3} - i_{r4} = \frac{v_2 - v_1}{r_2} + \frac{v_2 - v_0}{r_3} - \frac{v_3 - v_2}{r_4} \tag{2c}$$

$$\text{node3:}\ \ f_3 = i_{r4} + i_1 = \frac{v_3 - v_2}{r_4} + i_1 \tag{2d}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}} \cdot \boldsymbol{v} = \boldsymbol{e} = \begin{bmatrix} \frac{1}{r_1} + \frac{1}{r_2} & -\frac{1}{r_2} & 0 \\ -\frac{1}{r_2} & \frac{1}{r_2} + \frac{1}{r_3} + \frac{1}{r_4} & -\frac{1}{r_4} \\ 0 & -\frac{1}{r_4} & \frac{1}{r_4} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -i_1 \end{bmatrix} \tag{3}$$



(a) (b)

Figure 2.5 Element stamps: (a) for *∂**f**/∂**v*** matrix; (b) for ***e*** vector.

For a resistor of value r_x from node i to node k as shown in Fig. 2.3, a positive conductance value $1/r_x$ is added to matrix locations (i,i) and (k,k), respectively, $-1/r_x$ is added to matrix locations (i,k) and (k,i). For an independent current source with from node i and to node k, a negative current source value $-i_x$ is added to the i_{th} element of the $\boldsymbol{e}$ vector and a positive value i_x is added to the k_{th} element.

Only this simple circuit example is provided from schematic netlist to the algebraic matrix form. Nonlinear device modelling and memoryless elements will be reviewed when DC analysis and transient analysis is introduced.

2.4 Steady State Analysis

DC analysis is also known as steady state analysis. Steady state analysis calculates the stabilized state vector of a circuit when constant voltage sources and/or current source input excitations are applied to it. In most cases, the results of the DC analysis are intermediate values for further analysis, thus, its accuracy is essential to the next analysis. A simple DC analysis is shown in section 2.3, and after inverting the $\partial \boldsymbol{f}/\partial \boldsymbol{v}$ matrix in (3), the DC operating point vector $\boldsymbol{v}$ is available. But the example in section 2.3 only has resistors and a current sour.

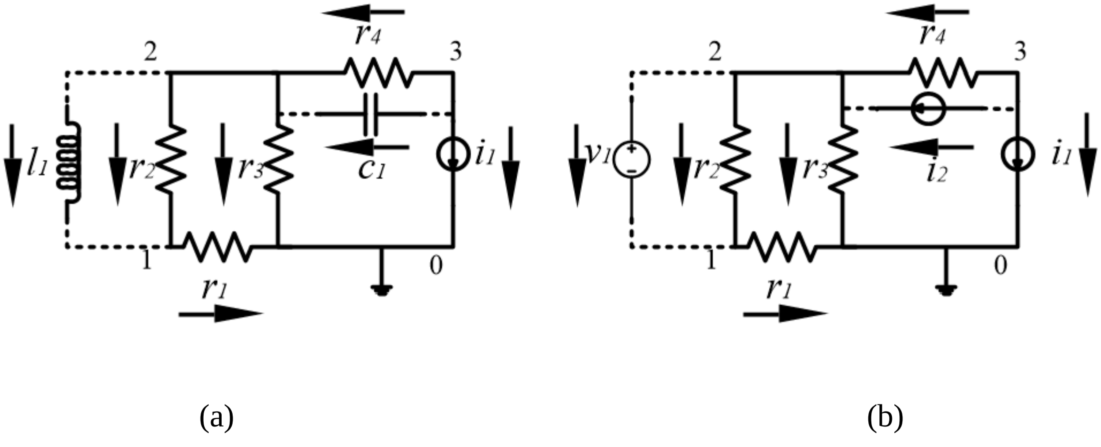


Figure 2.6 Circuit example for explaining steady state analysis: (a) capacitor/inductor is added to circuit shown in Fig. 2.3 and its equivalent replacement (b).

Since DC operating point analysis does not consider any time dependence on any of the devices, the capacitor becomes an open circuit and inductor becomes a short circuit in steady state. If one capacitor and inductor is added to circuit of Fig. 2.3, the new schematic is shown in Fig. 2.6 and corresponding KCL&KVL equations are

$$\begin{bmatrix} \frac{1}{r_1}+\frac{1}{r_2} & -\frac{1}{r_2} & 0 & -1 \\ -\frac{1}{r_2} & \frac{1}{r_2}+\frac{1}{r_3}+\frac{1}{r_4} & -\frac{1}{r_4} & 1 \\ 0 & -\frac{1}{r_4} & \frac{1}{r_4} & 0 \\ -1 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ i_{vsrc} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -i_1 \\ v_1 \end{bmatrix} \tag{4}$$

where i_{vsrc} represents the current goes through voltage source (vsrc).

As one voltage source is added to the original circuit shown in Fig. 2.3, one more unknown enters the algebraic equations because the branch current of voltage source is balanced by the known value of the voltage source, and the number of current unknowns equals the number of voltage source branches. Thus, one more column and row are needed in matrix $\partial \boldsymbol{f}/\partial \boldsymbol{v}$ which is also known as modified nodal analysis. For a circuit with multiple independent voltage sources and the m_{th} voltage source is the one with ‘from node’ *i* and ‘to node’ *k*, a positive unit is added to modified matrix locations $(n+m,k)$&$(i,n+m)$ and a negative unit is added to $(n+m,i)$&$(k,n+m)$.

Devices in the aforementioned circuit are linear and the steady state can be obtained after inverting the $\partial \boldsymbol{f}/\partial \boldsymbol{v}$ matrix once. But it is not practical to attain nonlinear device’s steady state analytically and Newton-Raphson method (NR) is the most straight forward and well-known solution [23-24] for it.

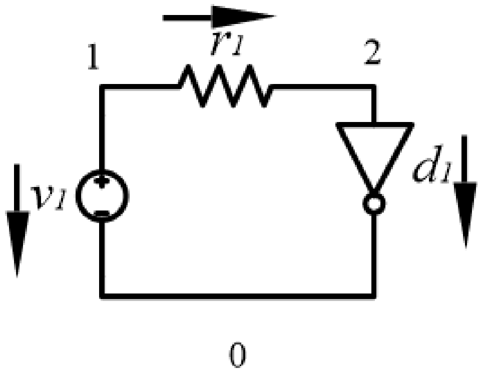


Figure 2.7 Simple nonlinear circuit (v_1=0.7*volt*; r_1=1*ohm*).

Consider the problem of determining the DC operating point for the simple nonlinear circuit shown in Fig. 2.7. The steady state of this single loop circuit can be easily formulated in terms of a loop equation as

$$f: \frac{0.7 - v_2}{r_1} - I_s \cdot [e^{\frac{v_2 - v_0}{\eta v_T}} - 1] = 0 \tag{5}$$

and the DC operating point can be easily found if a load line is drawn on the plot of the characteristic curve as shown in Fig. 2.8.

But unfortunately, not all circuits can be formulated in a single loop equation and a universal load line is seldom available. Thus, KCL&KVL equations for each node are necessary and listed as

$$f_1 = \frac{v_1 - v_2}{r_1} + i_{vsrc} \tag{6a}$$

$$f_2 = -\frac{v_1 - v_2}{r_1} + I_s \cdot [e^{\frac{v_2 - v_0}{\eta v_T}} - 1] \tag{6b}$$

$$f_{vsrc} = v_1 - v_0 - v_{vsrc} \tag{6c}$$

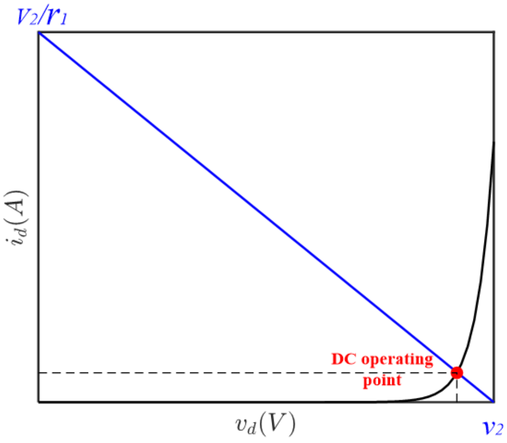


Figure 2.8 Load line of simple nonlinear circuit shown in Fig. 2.7.

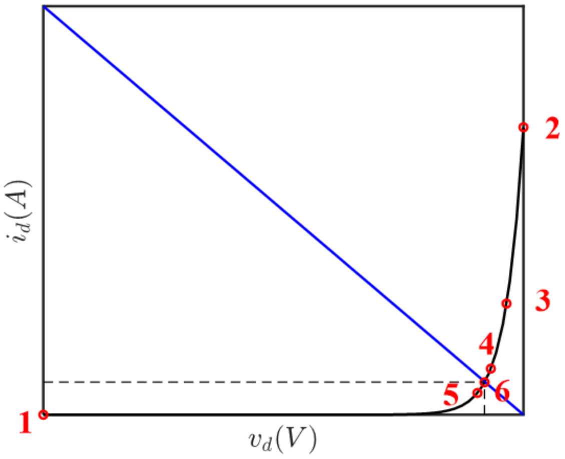


Figure 2.9 NR iteration process of simple nonlinear circuit shown in Fig. 2.7.

In steady state, those equations should have $\boldsymbol{f}$=0. Now, if NR is applied to solve the DC operating point problem shown in Fig. 2.7, an initial state vector $\boldsymbol{v}_{1\mathrm{st}}$ as the starting point of the iteration is necessary to kick start the NR, and the initial state vector have $\boldsymbol{f}(\boldsymbol{v}_{1\mathrm{st}})\neq 0$. At each iteration of NR, the partial derivative of function $\partial\boldsymbol{f}/\partial\boldsymbol{v}$ becomes a gradient and the next state $\boldsymbol{v}_{(\mathrm{n}+1)\mathrm{th}}$ is updated by following the local minimum of this gradient as

$$\boldsymbol{v}_{(n+1)th} = \boldsymbol{v}_{nth} - [\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}}]^{-1} \boldsymbol{f}(v_{nth}) \tag{7}$$

The full NR iteration process is shown in Fig. 2.9 and the initial state vector is assumed to be zero, which is the default setting of the NR method.

Based on the above, the initial guess is critical to the NR-derived algorithms because fewer iterations are needed if the initial guess is close enough to the steady state. NR does not guarantee convergence because it can be easily trapped in the local minimum of the function [24], thus, more backups are needed if NR fails steady state analysis [25-26].

2.5 Forward/Backward Euler Transient Analysis

A transient analysis first needs a point of departure for simulating time-varying behavior and mostly the point of departure is selected to be the result of DC analysis. Once the initial state $\boldsymbol{v}(t=0)$ is attained, the time-dependent aspects of circuit are reevaluated, and the simulation solves for the state vector at the next time point. Forward/Backward Euler are the numerical integration approximation algorithms used to approximate the analytical response of the transient. More advanced transient algorithms are available, but we confine attention to these two to avoid uselessly cluttering the ensuing exposition.

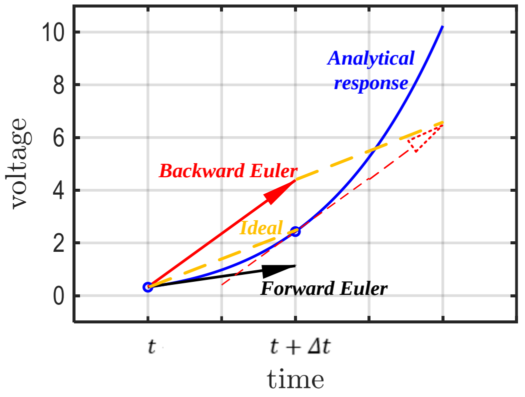


Figure 2.10 Difference between Forward/Backward Euler integrations from t to $t+\Delta t$.

Suppose both Forward-Euler and Backward-Euler start at the same point of departure, the analytical time varying behavior of capacitor voltage can be represented as

$$\boldsymbol{v}_c(t+\Delta t)=\boldsymbol{v}_c(t)+\int_t^{t+\Delta t}\frac{i_c(\tau)}{c}d\tau \tag{7}$$

Here we consider a numerical approximation of the integration from t to t+Δt in (7) as

$$\boldsymbol{v}_c(t+\Delta t)=\boldsymbol{v}_c(t)+\begin{cases}\dfrac{i_c(t)}{c}\cdot\Delta t & \mathit{Forward-Euler}\\ \dfrac{i_c(t+\Delta t)}{c}\cdot\Delta t & \mathit{Backward-Euler}\end{cases} \tag{8}$$

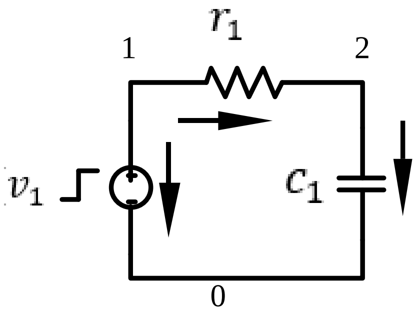


Figure 2.11 RC circuit with step function as excitation (v_1=1*volt*; r_1=1*ohm*; c_1=1*F*).

Forward Euler integration method applies the time derivative at time t, Backward Euler integration method selects the time derivative at time (t+Δt) as shown in Fig. 2.10. Note that the slope of the BE integration approximation is that of the tangent to the nonlinear characteristic curve at time (t+Δt). More details about Forward/Backward-Euler integration methods are discussed in terms of an RC circuit shown in Fig. 2.11. In Fig. 2.11, if a step function is applied as the input excitation, the analytical transient response can be represented as

$$\boldsymbol{v}_c(t)=1-e^{\frac{-t}{r_1c_1}} \tag{9}$$

If the transient analysis is in terms of Forward Euler (FE), then its excursion from time *t* to *t*+Δ*t* is

$$\Delta v_c = c_1^{-1}\cdot\frac{v_1(t)-v_2(t)}{r_1}\cdot\Delta t \tag{10}$$

If the transient analysis is in terms of Backward Euler (BE), then its same excursion is

$$\Delta \boldsymbol{v}_c = (\frac{\Delta t}{r_1}+c_1)^{-1}\cdot\frac{v_1(t)-v_2(t)}{r_1}\cdot\Delta t \tag{11}$$

The circuit in Fig. 2.11 is simulated in terms of both Forward-Euler and Backward-Euler with different time steps, and the simulation results are shown in Fig. 2.12 and Fig. 2.13.

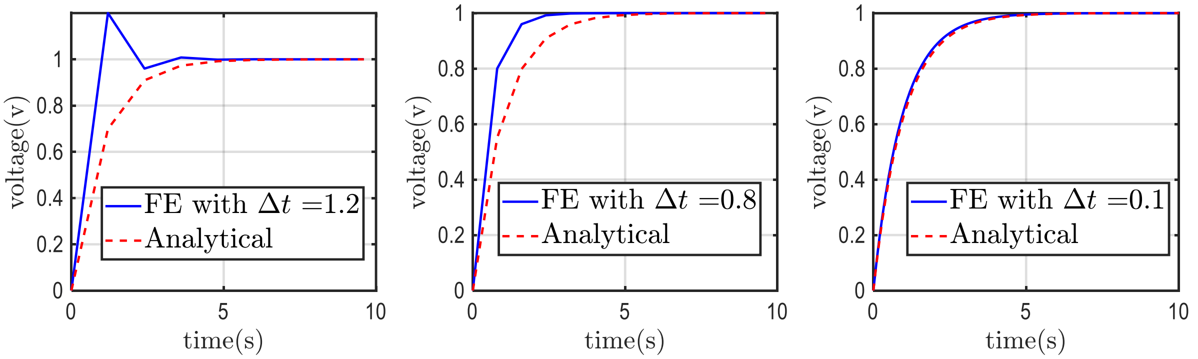


Figure 2.12 Transient analysis in terms of the Forward-Euler integration approximation with different time steps Δ*t*.

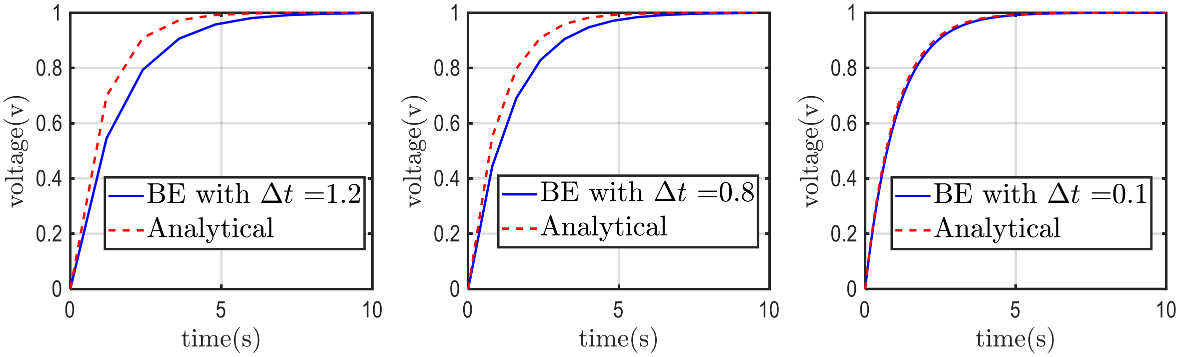


Figure 2.13 Transient analysis in terms of the Backward-Euler integration approximation with different time steps Δ*t*.

Based on (10), (11) and the simulations result above, Forward-Euler is a fast explicit integration approximation and Backward-Euler is a stable implicit integration approximation. If a very small time step is applied, then both FE and BE provide essentially the same result.

2.6 Background Summary

The above briefly explains how steady state analysis and Forward/Backward-Euler transient analysis works but with simple examples. This section will focus on revisiting DC and transient analysis again in a general circuit example and the form of vectors/matrices will be carried over to the entire dissertation.

Suppose a general integrated circuit netlist contains b branches and n nodes, the KCL&KVL functions for all memoryless elements at each node form a vector as

$$\boldsymbol{f} = \begin{bmatrix} f_1 \\ f_2 \\ \cdots \\ f_n \end{bmatrix} \tag{12}$$

where f_{n} is the KCL function at node n. In steady state, circuit has $\boldsymbol{f}(\boldsymbol{v})=0$ where $\boldsymbol{v}$ is the state vector of the circuit. If the NR method is applied to solve for the steady state, the state vector of the next iteration $\boldsymbol{v}_{(n+1)th}$ is

$$\boldsymbol{v}_{(n+1)th} = \boldsymbol{v}_{nth} - [\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}}]^{-1} \boldsymbol{f}(v_{nth}) \tag{13}$$

Before the discussion of FE and BE in terms of matrices. It should be noted that there is a vector $\boldsymbol{q}$ being defined in the same manner as $\boldsymbol{f}$ shown in (12), but it only includes the memory elements, capacitors and inductors. Vector $\boldsymbol{f}$ contains the sums of the currents leaving each node and vector ***q*** contains similar sums of charge/flux. The matrix ∂***f/***∂***v*** is abbreviated as *G* and the matrix ∂***q/***∂***v*** is abbreviated as *C* in the following chapters.

In Forward-Euler transient analysis, the integration from time *t* to *t*+Δ*t* is

$$\boldsymbol{v}(t+\Delta t)=\boldsymbol{v}(t)-[\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{v}}]^{-1}\boldsymbol{f}(\boldsymbol{v}(t))\cdot\Delta t \tag{14}$$

In Backward-Euler transient analysis, the integration from time *t* to *t*+Δ*t* is

$$\boldsymbol{v}(t+\Delta t)=\boldsymbol{v}(t)-[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{v}}+\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{v}}/\Delta \mathrm{t}]^{-1}\boldsymbol{f}(\boldsymbol{v}(t+\Delta t)) \tag{15}$$

As shown above, all circuit analysis entails the inversion of a Jacobian matrix which takes more than 90% of the simulation run time and the rest of 10% is mostly spent on device evaluation.

CHAPTER 3

The Backward Euler integration method and its variants are widely employed in SPICE [27] which is a well-known, general-purpose, open-source electronic circuit simulator. Those integration methods are able to provide utmost accuracy and stability, but they are a primary cause of slowing simulation time. To improve efficiency, various approaches have been developed either to incorporate multi-rate estimation [28] or to apply model simplification such as piecewise linear [29]. In any case a great deal of computational power is consumed to iterate and invert (LU-factor) a Jacobian matrix at each time step. This chapter shows the possibility of applying Forward-Euler integration approximation to underline transient analysis with an adaptively controlled time step, algorithmic stability, and user-defined accuracy control.

Also, SPICE was not initially designed to handle step function excitations directly. To avoid discontinuity, SPICE-derived transient simulators substitute inefficient steep ramps for step function excitations. As the rise time of a steep ramp is reduced to better approximate a step function, more iterations are required to maintain accuracy and stability, thereby consuming substantial computation time. To account for digitally dominated circuits with rapid turn on/off or the abstraction of the simulation environment to switch level [30], applying step functions as input excitations and generalizing any input functions to step functions are discussed in this chapter.

3.1 Step Response

As mentioned earlier, DC analysis requires an initial state vector to kick start the NR and a step response is a suitable candidate to attain the initial state vector because it represents the natural starting point after the input excitations are switched on. Also, the step response, once it has settled, can provide the departure point for simulating time-varying behavior. Usually, the point of departure is selected to be the result of DC operating point NR iteration if it converges. But NR convergence to the steady state is not guaranteed.

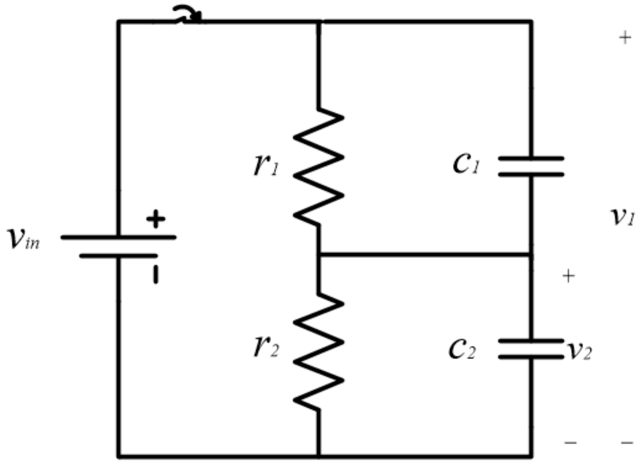


Figure 3.1 RC example for explanation of step response calculation (v_1=0.8V; r1=100ohm; r_2=400ohm; c_1=c_2=10^{-3}F).

The initial state $\boldsymbol{v}(t=0_+)$ of a capacitor array excited by a step function is its charge-voltage equilibrium point.

$$\boldsymbol{q_c} = c \cdot \boldsymbol{v_c}(t = 0_+) \tag{16}$$

To illustrate its computation, we consider a simple series connection of two resistors and capacitors in parallel shown in Fig. 3.1. The input excitation function to the circuit is a step function

$$\begin{cases} v_{in}(t = 0\_) = 0 \\ \int_{0_-}^{0_+} v_{in}(t)dt = 0.8 \\ v_{in}(t = 0_+) = 0.8 \end{cases} \tag{17}$$

Initially at the exact time of the step function excitation, t=0, the voltage source provides an impulse of current and the capacitors in the circuit instantaneously equilibrate their charges and voltages. The remaining elements are indifferent to this impulse of current, so it can be ignored for such analysis. In practical cases, because of highly elaborated transistor models, capacitors are everywhere. Thus, we assume that capacitors probably connect to all nodes, and the general case will be discussed later. The capacitor responses to the impulses of current which determines the state at t=0_+, and the calculation of the impulse response and the effective charge is equivalent to DC analysis with charge equilibrium replacing current equilibrium. To represent (16) in terms of matrices, we use $\boldsymbol{v_n}(t=0_+)$ as the node voltage vector representing the voltage source specifying the initial step response and C as the capacitance matrix. For the circuit of Fig. 3.1, the matrix equation is

$$\begin{bmatrix} c_1 & -c_1 & 1 \\ -c1 & c_1+c_2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} v_1(t=0+) \\ v_2(t=0+) \\ q_{vsrc}(t=0+) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ v_{in}(0_+) \end{bmatrix} \tag{18}$$

where q_{vsrc} is the electric charge flowing through the voltage source that initializes the capacitor voltages.

In simulators employing the BE integration method, if a circuit contains capacitor loops, then a great deal of computational resource must be devoted to climbing an approximating steep ramp to determine the peak current that charges the capacitors. Instead, the proposed simulator computes the peak directly by instantaneously charging all capacitors to their initial equilibria. This is a one-time DC-like computation that inverts (LU-factors) the C matrix, an operation that also needs be performed anyway to compute the remainder of the transient.

Table 3.1 Peak comparisons between ideal step & steep ramp functions

- ***Part A: Comparisons among $\boldsymbol{v_2(t_{rise})}$ and $\boldsymbol{v_2}(0_+)$***

t_{rise}/*ms*	$\boldsymbol{v_2(t_{rise})}$**/v or** $\boldsymbol{v_2}(0_+)$**/v**		
	Backward Euler	**Trapezoidal**	**GEAR2**
200	*0.6295*	*0.6319*	*0.6273*
100	*0.5769*	*0.5880*	*0.5648*
50	*0.5375*	*0.5493*	*0.5213*
10	*0.5092*	*0.5149*	*0.5064*
5	*0.5046*	*0.5088*	*0.5027*
Ideal step	*0.5*		

- ***Part B: Comparisons among q_{eff} and q_{vsrc}***

t_{rise}/*ms*	q_{eff}/C **or** q_{vsrc}/C		
	Backward Euler	**Trapezoidal**	**GEAR2**
200	*0.77543**10^{-3}	*0.77989**10^{-3}	*0.73981**10^{-3}
100	*0.64408**10^{-3}	*0.64591**10^{-3}	*0.64087**10^{-3}
50	*0.57584**10^{-3}	*0.59013**10^{-3}	*0.55698**10^{-3}
10	*0.51622**10^{-3}	*0.52319**10^{-3}	*0.51326**10^{-3}
5	*0.50883**10^{-3}	*0.51011**10^{-3}	*0.50491**10^{-3}
Ideal step	*0.5**10^{-3}		

- ***Part C: Error from climbing samples with $\boldsymbol{t_{rise} = 5ms}$***

Number of samples in climbing	*Absolute integration error*		
	Backward Euler	**Trapezoidal**	**GEAR2**
100	*1.101**10^{-7}	*5.802**10^{-9}	*3.353**10^{-9}
50	*2.281**10^{-7}	*5.877**10^{-9}	*7.756**10^{-9}
20	*5.956**10^{-7}	*6.179**10^{-9}	*7.77**10^{-8}
10	*1.252**10^{-6}	*6.955**10^{-9}	*3.198**10^{-7}
5	*2.731**10^{-6}	*9.735**10^{-9}	*1.227**10^{-6}

A comparison has been made between step function excitation and steep ramps with different rise times in the Model and Algorithm Prototyping Platform (MAPP) [31], a MATLAB™

based open-source simulation framework which is able to execute SPICE-derived transient analysis. The comparison results based on the above circuit example are shown in Table. 3.1 & Fig. 3.2. The effective electric charge flowing through voltage source q_{eff} from t=0 to t=t_{rise} with the input excited by a steep ramp is

$$q_{\boldsymbol{eff}} = \int_{0}^{t_{rise}} i_{vsrc}(t)dt \tag{19}$$

and its corresponding state vector at t=t_{rise} is defined as $\boldsymbol{v_n}(t=t_{rise})$.

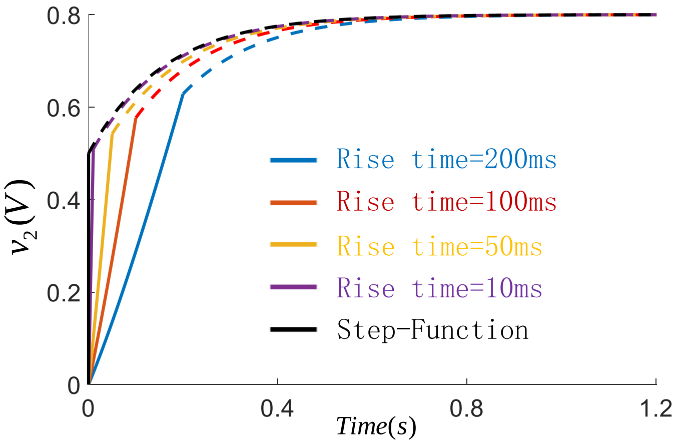


Figure 3.2 Step function response versus steep ramps with various rise time.

The end point of climbing $\boldsymbol{v_n}(t=t_{rise})$ is the initial state for the remainder of the transient. The overall accuracy compared with the step response is determined by both rise times and time samples. No matter what integration method is employed, as the steep ramp becomes more step-like, $\boldsymbol{v_n}(t=t_{rise})$ and the effective charge q_{eff} become closer to $\boldsymbol{v_n}(t=0_+)$ and q_{vsrc}, respectively, and the response becomes more accurate. The same experiment is to be performed on the upcoming nonlinear circuits examples, and the same conclusion obtains. For the above, increasing computational power is consumed to maintain accuracy when the input is excited by steeper and steeper ramps, but the step response solver does not climb such a hill to calculate $\boldsymbol{v_n}(t=0_+)$; hence, no extra effort is entailed.

The circuit in Fig. 3.1 shows how the step response is calculated for a simple example. The solution of the step response for a general circuit can be represented by

$$C \cdot \boldsymbol{v_n}(t=0+) = \boldsymbol{q}(\boldsymbol{v_n}(t=0)) \tag{20}$$

It should be noted, (20) is the same as (15) after the time step of (15) is set to be infinitesimal. Practically, a circuit may have nodes which not tied directly to a capacitor or are excited with a step function current source. To address such issues a two-step process is required, and modification of matrix in (20) is necessary. Charge-voltage equilibrium (flux-current equilibrium for inductors if it applies) of memory elements (capacitors/inductors) is the solution with priority and should be achieved at $t=0_+$, then, for those internal nodes which are not tied directly to memory elements, current equilibrium (KCL) must be obtained for them as a secondary solution. To clear the residue of charge and current for both memory elements and memoryless elements, modification of the matrix is established as

$$\begin{bmatrix} G & 0 \\ 0 & C \end{bmatrix} \cdot \boldsymbol{v_n}(t=0+) = \begin{bmatrix} \boldsymbol{f}(\boldsymbol{v_n}(t=0)) \\ \boldsymbol{q}(\boldsymbol{v_n}(t=0)) \end{bmatrix} \tag{21}$$

In (21), G and $\boldsymbol{f}$ stands for the linearization of memoryless elements which are not tied directly to memory elements, and their unbalanced current. C and $\boldsymbol{q}$ stand respectively for the linearization of memory elements and their unbalanced charge or flux.

In real integrated circuit, characteristics of memory elements could also be nonlinear, and the linearization of memory elements C which is the foundation of charge-voltage equilibrium also depends on the state of the circuit. The solution for such a situation is by splitting one step function into multiple step functions and acquiring charge-voltage equilibrium at each end of the step function to smooth the effect of nonlinearity as is shown in Fig. 3.3.

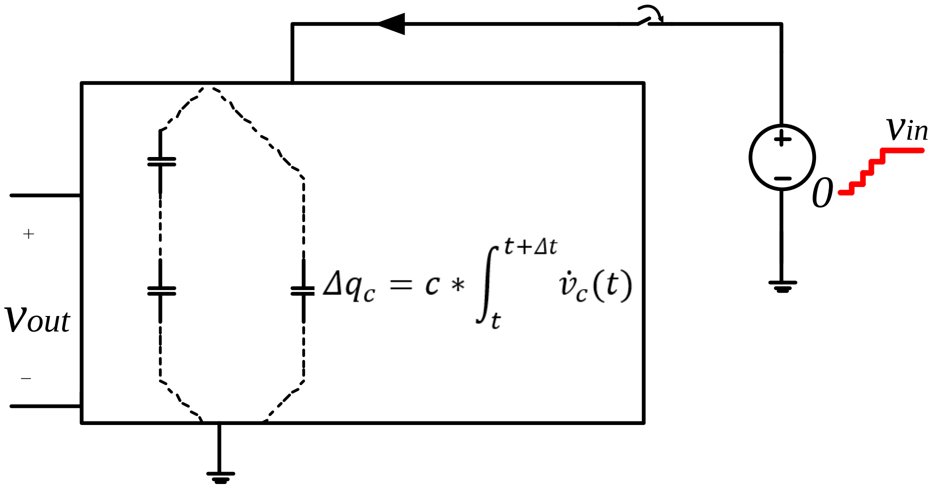


Figure 3.3 Step function excitation for general circuit schematic.

3.2 Static-driven Algorithm

After the initial state is attained, the remainder of the transient is taken over by the proposed explicit static-driven method which is inspired by Adaptively Controlled Explicit Simulation (ACES) [32]. State variables approach their static equilibria with different time constants. By partitioning state variables into a quiescent group and a dynamic group, the simulator naturally and adaptively differentiates the fast/slow state variables in the circuit. This static-driven method is developed to drive the dynamic state variables to be quiescent in order, and those quiescent state variables to remain quiescent until all state variables become quiescent when the system is in steady state. A state variable is set to quiescence by equating its second order time derivative to zero [32].

With Backward-Euler integration approximation for large stiff systems, a Jacobian matrix must be inverted at each iteration and ill-conditioning can occur. Quiescence automatically differentiates the state variables into a fast group and a slow group which enables it to iterate adaptively with larger time steps and no ill-conditioning.

At first, all state variables are dynamic. The explicit first and second order time derivative of a general state space system [33] can be described in terms of

$$\dot{\boldsymbol{x}} = A \cdot \boldsymbol{x} + B \cdot \boldsymbol{u} \tag{22a}$$

$$\ddot{\boldsymbol{x}} = A \cdot \dot{\boldsymbol{x}} + B \cdot \dot{\boldsymbol{u}} \tag{22b}$$

where $\boldsymbol{u}$ represents the vector of input functions. For simplicity, if the circuit system is excited by step functions, then at any time point the time derivatives of input functions are zero, so

$$\dot{\boldsymbol{x}} = A \cdot \boldsymbol{x} + B \cdot \boldsymbol{u} \tag{23a}$$

$$\ddot{\boldsymbol{x}} = A \cdot \dot{\boldsymbol{x}} \tag{23b}$$

On the way of driving dynamic state variables into quiescence, for the k^{th} state variable, an adaptive step size controller is defined to be

$$\Delta t_k = -\dot{x}_k(t) / \ddot{x}_k(t) \tag{24}$$

It provides an estimation of the real parts of time constants for those dynamic state variables. At each iteration, the smallest Δt_k is chosen to be the step size in order to ensure stability. At the end of the estimated time step, the k^{th} dynamic state variable with the smallest Δt_k then is set to be quiescent. For practicality, the Δt_k of those state variables which are close to the smallest one should be set to quiescence concurrently, because they display close time constants especially for symmetrical nodes which exists in mirror topology and perform the same response.

The second order time derivative of quiescent terms should be zero. Thus, for the quiescence terms in (23b), we have

$$\ddot{\boldsymbol{x}}_{\boldsymbol{q}} = A_{11}\dot{\boldsymbol{x}}_{\boldsymbol{q}} + A_{12}\dot{\boldsymbol{x}}_{\boldsymbol{nq}} = 0 \tag{25}$$

where the state vector $\boldsymbol{x}$ includes a q-dimensional quiescent state sub-vector $\boldsymbol{x_q}$ and an (n-q)-dimensional dynamic state sub-vector $\boldsymbol{x_{nq}}$, A_{11} and A_{12} are the sub-matrices of state space matrix A. x_i represents the i^{th} element of the quiescent state variables where i=1,2,...,q; x_j represents the j^{th} element of dynamic state variables where j=q+1,q+2,...n. Based on (25) the time derivative for the quiescent state variables can be derived to be

$$\dot{\boldsymbol{x}}_{\boldsymbol{q}} = -A_{11}^{-1}A_{12}\dot{\boldsymbol{x}}_{\boldsymbol{nq}} \tag{26}$$

Essentially, the time derivatives of quiescent state variables are controlled by the time derivatives of dynamic state variables. From the application of such control, the second order time derivatives of those quiescent state variables with small time constants are forced to be zero. Thus, the potential instability problem associated with explicit integration is properly addressed. Such control is applied for all subsequent iterations, so those quiescent state variables remain quiescent.

Based on (26) the modified state space equation can be written as

$$\begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{q}}(t) \\ \dot{\boldsymbol{x}}_{\boldsymbol{nq}}(t) \end{bmatrix} = \begin{bmatrix} -A_{11}^{-1}A_{12}A_{21} & -A_{11}^{-1}A_{12}A_{22} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{q}}(t) \\ \boldsymbol{x}_{\boldsymbol{nq}}(t) \end{bmatrix} + \begin{bmatrix} -A_{11}^{-1}A_{12}(B\boldsymbol{u})_{nq} \\ (B\boldsymbol{u})_{nq} \end{bmatrix} \tag{27}$$

Fast state variables enter quiescence earlier than slow state variables and eventually all state variables enter quiescence. During the integration process, the fastest state variable leads the rest to update before it enters quiescence. After they become quiescent, those quiescent state variables essentially are followers of the next fastest state variable. When there is only one dynamic state variable waiting to join the quiescent group at the last iteration, the second order time derivative of the last dynamic state variable can be evaluated according to (26), thus, according to (24) the final time step size is

$$\Delta t = -\dot{\boldsymbol{x}}_{\boldsymbol{nq}} \,/\, \ddot{\boldsymbol{x}}_{\boldsymbol{nq}} = \frac{-1}{A_{22} - A_{21}A_{11}{}^{-1}A_{12}} \tag{28}$$

where the size of the submatrix A_{22} is one by one. After this final time step Δ*t*, the former quiescent state variables and dynamic state variable at *t*+Δ*t* can be described independently as

$$\boldsymbol{x}_{\boldsymbol{q}}(t+\Delta t) = \boldsymbol{x}_{\boldsymbol{q}}(t) - A_{11}{}^{-1}A_{12}\dot{\boldsymbol{x}}_{\boldsymbol{nq}}(t)\cdot\Delta t \tag{29a}$$

$$\boldsymbol{x}_{\boldsymbol{nq}}(t+\Delta t) = \boldsymbol{x}_{\boldsymbol{nq}}(t) + \dot{\boldsymbol{x}}_{\boldsymbol{nq}}(t)\cdot\Delta t \tag{29b}$$

From (29), the time derivatives of the dynamic state variable at *t*+Δ*t* can be calculated based on (23), and its value will be equal to zero

$$\begin{aligned}\dot{\boldsymbol{x}}_{\boldsymbol{nq}}(t+\Delta t) &= A_{21}\boldsymbol{x}_{\boldsymbol{q}}(t+\Delta t) + A_{22}\boldsymbol{x}_{\boldsymbol{nq}}(t+\Delta t) + (B\boldsymbol{u})_{nq} \\ &= A_{21}\boldsymbol{x}_{\boldsymbol{q}}(t) - A_{21}A_{11}{}^{-1}A_{12}\dot{\boldsymbol{x}}_{\boldsymbol{nq}}(t)\cdot\Delta t + \\ &\quad A_{22}\boldsymbol{x}_{\boldsymbol{nq}}(t) + A_{22}\dot{\boldsymbol{x}}_{\boldsymbol{nq}}(t)\cdot\Delta t + (B\boldsymbol{u})_{nq} \\ &= 0\end{aligned} \tag{30}$$

The time derivative of quiescent state variables which are controlled by the dynamic state variable is equal to zero as well

$$\dot{\boldsymbol{x}}_{\boldsymbol{q}}(t+\Delta t) = A_{11}{}^{-1}A_{12}\dot{\boldsymbol{x}}_{\boldsymbol{nq}}(t+\Delta t) = 0 \tag{31}$$

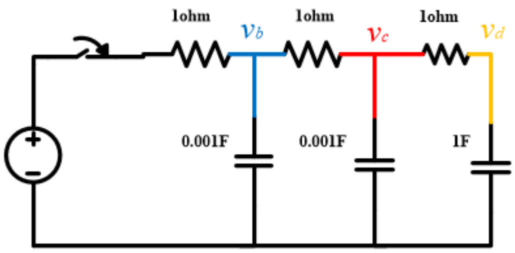


Figure 3.4 Example stiff system.

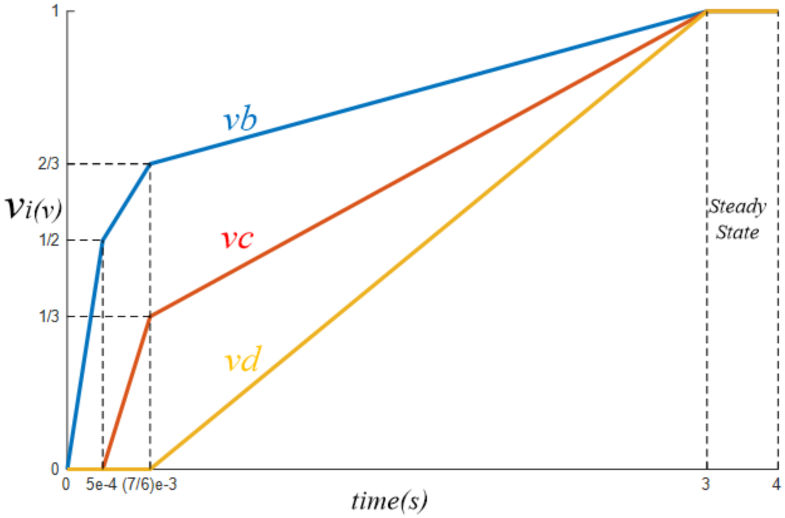


Figure 3.5 The gross response of a stiff system in Fig. 3.4.

So far, all state variables become quiescent, and the system is in steady state at $t+\Delta t$. To better explain the proposed method above, an inevitable linear stiff system and an underdamped RLC system are employed as examples. A classic stiff system and its transient response simulated based on proposed simulator are shown in Figs. 3.4&3.5, respectively. For the stiff circuit in Fig. 3.4, its state equation can be derived according to (22a) as follows

$$\begin{cases} \dot{v}_b = (u - 2\cdot v_b + v_c)\cdot 1000 \\ \dot{v}_c = (v_b - 2\cdot v_c + v_d)\cdot 1000 \\ \dot{v}_d = (v_c - v_d) \end{cases} \tag{32}$$

At the initial state, none of these state variables is quiescent. Thus, for those dynamic state variables

$$\dot{v}_i^{\,+}(t) = \dot{v}_i(t) \tag{33}$$

where left hand side of (33) represents the time derivative from (32) and the right hand side represents the time derivative from (27). The state equation and modified state equation deliver the same result as

$$\begin{cases} \dot{v}_b^{\,+}(0) = 1000 \\ \dot{v}_c^{\,+}(0) = 0 \\ \dot{v}_d^{\,+}(0) = 0 \end{cases} \tag{34}$$

from (23b) we obtain

$$\begin{cases} \ddot{v}_b^{\,+}(0) = [-2\cdot \dot{v}_b^{\,+}(0) + \dot{v}_c^{\,+}(0)]\cdot 1000 \\ \ddot{v}_c^{\,+}(0) = [\dot{v}_b^{\,+}(0) - 2\cdot \dot{v}_c^{\,+}(0) + \dot{v}_d^{\,+}(0)]\cdot 1000 \\ \ddot{v}_d^{\,+}(0) = [\dot{v}_c^{\,+}(0) - \dot{v}_d^{\,+}(0)] \end{cases} \tag{35}$$

After the calculation of the time derivative and the second order time derivative, the safe time step is calculated according to (24)

$$\Delta t = \Delta t_b = 5e^{-4} \tag{36}$$

thus, the simulation proceeds with

$$\boldsymbol{v}(5e^{-4}) = \boldsymbol{v}(0) + \dot{\boldsymbol{v}}^{+}(0)\cdot \Delta t_b \tag{37}$$

In the second iteration, *v_b* is set to be quiescent and the time derivative and the second order time derivative can be calculated based on (23b)&(27) to be

$$\begin{cases} \dot{v}_b{}^+(5e^{-4}) = 250 \\ \dot{v}_c{}^+(5e^{-4}) = 500 \\ \dot{v}_d{}^+(5e^{-4}) = 0 \end{cases} \tag{38a}$$

$$\begin{cases} \ddot{v}_b{}^+(5e^{-4}) = 0 \\ \ddot{v}_c{}^+(5e^{-4}) = -7.5e^{5} \\ \ddot{v}_d{}^+(5e^{-4}) = 500 \end{cases} \tag{38b}$$

and the safe time step is

$$\Delta t = \Delta t_c = \frac{2}{3}e^{-3} \tag{39}$$

For the third iteration, v_b and v_c are quiescent and the time derivative and the second order time derivative can be calculated based on (23b)&(27) to be

$$\begin{cases} \dot{v}_b{}^+(\frac{7}{6}e^{-3}) = \frac{1}{9} \\ \dot{v}_c{}^+(\frac{7}{6}e^{-3}) = \frac{2}{9} \\ \dot{v}_d{}^+(\frac{7}{6}e^{-3}) = \frac{1}{3} \end{cases} \tag{40a}$$

$$\begin{cases} \ddot{v}_b{}^+(\frac{7}{6}e^{-3}) = 0 \\ \ddot{v}_c{}^+(\frac{7}{6}e^{-3}) = 0 \\ \ddot{v}_d{}^+(\frac{7}{6}e^{-3}) = \frac{1}{9} \end{cases} \tag{40b}$$

and the safe time step is

$$\Delta t = \Delta t_d = 3 \tag{41}$$

After the third iteration, all state variables have become quiescent, and circuit has attained the steady state with all capacitor state variables equal to one volt. Notice that the time step size increased by one order of magnitude between the first and second iteration and by three orders of magnitude between the second and third. Classical FE approximation would have been stuck at a step size roughly equal to the firs throughout the transient simulation, so instead of three time steps there would have been thousands for such a stiff system.

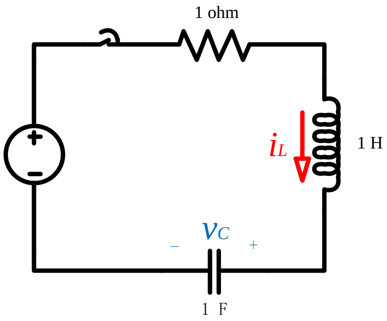


Figure 3.6 Example underdamped system.

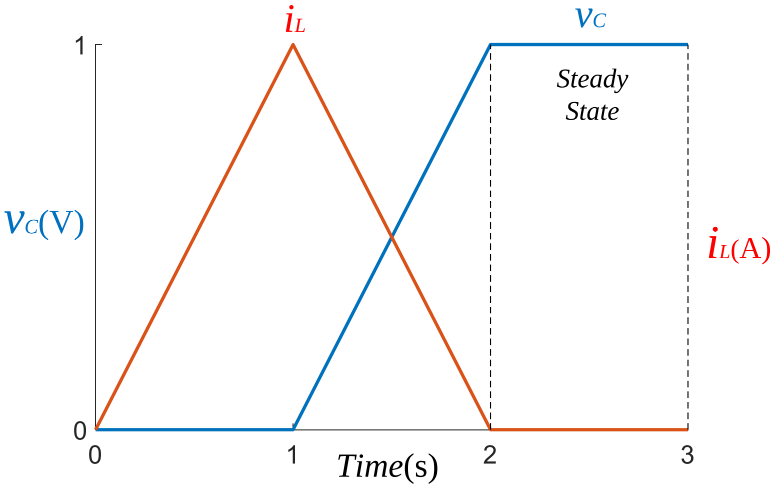


Figure 3.7 The gross response of the underdamped system in Fig. 3.6.

A classic underdamped system and its transient response simulated based on the proposed simulator are shown in Fig. 3.6&3.7, respectively. The state equation of the underdamped system can be represented as

$$\begin{cases} \dot{v}_c = i_L \\ \dot{i}_L = -v_c - i_L + u \end{cases} \tag{42}$$

At the initial state, none of these state variables is quiescent, thus the time derivatives and second order time derivatives can be calculated to be

$$\begin{cases} \dot{v}_c{}^{+}(0) = 0 \\ \dot{i}_L{}^{+}(0) = 1 \end{cases} \tag{43a}$$

$$\begin{cases} \ddot{v}_c{}^{+}(0) = 1 \\ \ddot{i}_L{}^{+}(0) = -1 \end{cases} \tag{43b}$$

and the safe time step based on (24) is Δ*t*=1.

In the second iteration, state variable *i*$_L$ becomes quiescent. Thus, the time derivative and second order time derivative can be calculated based on (23b) & (27)

$$\begin{cases} \dot{v}_c{}^{+}(1) = 1 \\ \dot{i}_L{}^{+}(1) = -1 \end{cases} \tag{44a}$$

$$\begin{cases} \ddot{v}_c{}^{+}(1) = -1 \\ \ddot{i}_L{}^{+}(1) = 0 \end{cases} \tag{44b}$$

The safe time step for the second iteration is Δ*t*=1 and after the second iteration, all state variables have become quiescent with the circuit in steady state. The steady state is correct, but the ringing of this underdamped system is greatly suppressed. The property of ringing suppression is because the quiescent state variables are forced to have zero second order time derivative and is applicable for finding steady state which is discussed in following chapter.

The whole of this section has focused on illustrating how the proposed simulator works for linear circuits and drives them all the way to their steady states. The drawbacks are obvious since accuracy control has yet to be applied. But there are advantages over the BE, or any other implicit integration method as follow. Applying the proposed static-driven algorithm, we need not perform an n by n matrix inversion at each iteration as with the BE integration method. As the number of quiescent state variables increases, the size of submatrix A_{11} gradually expands to become the entire state space matrix A. (Of course, this is not an efficient mechanism to invert a matrix A if one does not want to approximate the trajectories of the state variables.) Thus after n iterations, the simulation ceases at the steady state. Furthermore, the time constant approximation in the proposed algorithm groups all the close time constants in the matrix which needs to be inverted, thus, no ill-conditioned matrix inversion is entailed.

3.3 Integration Error Control

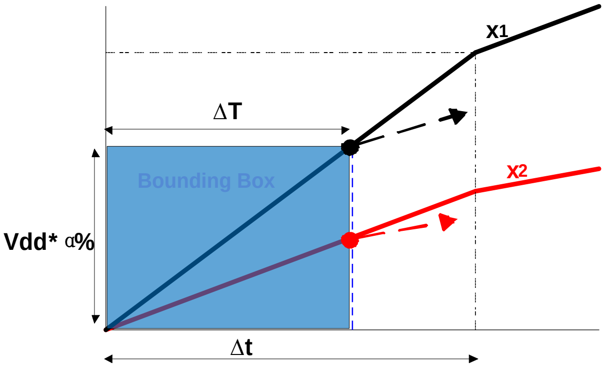


Figure 3.8 The depiction of the Excursion-limit accuracy control mechanism.

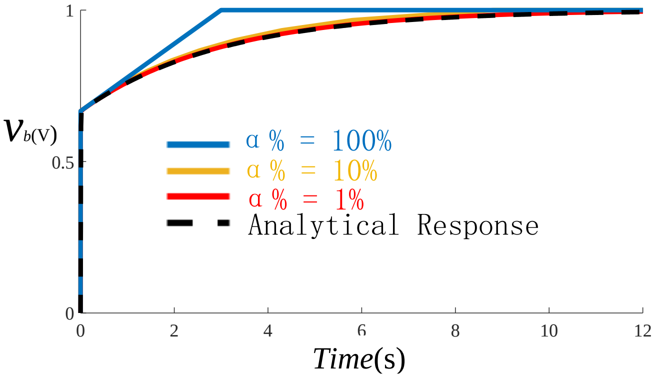


Figure 3.9 The refined response of v_b for the stiff system shown in Fig. 3.4.

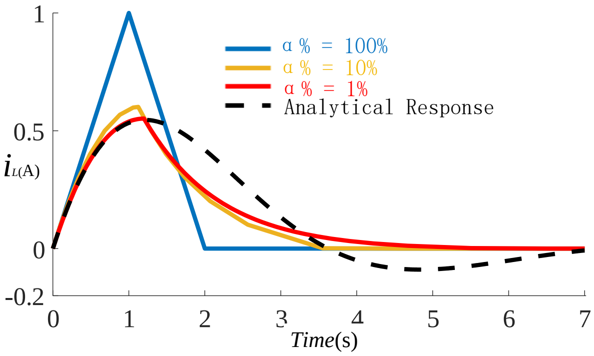


Figure 3.10 The refined response of i_L for the underdamped system in Fig. 3.6.

From the gross response of the stiff system and the underdamped system in Fig. 3.4&3.6, respectively, the proposed algorithm without error control drives the state variables to steady state, but very coarsely and therefore inaccurately. The time intervals calculated from (23) are not small enough to smooth the transient response. Thus, a refined step size is required. To address this issue, an excursion-limited accuracy control mechanism is applied. This mechanism employs a user defined accuracy control parameter α% introduced below to restrain the maximum excursion $\Delta \boldsymbol{x}$ at every iteration

$$\Delta T = \frac{Vdd \cdot \alpha\%}{\max(\dot{\boldsymbol{x}}^{+})} \tag{45}$$

When ΔT>min(Δt_{nq}), a direct transition towards quiescence following min(Δt_{nq}) is employed. When ΔT<=min(Δt_{nq}), then ΔT should be considered to be the safe time step for the present iteration. With ΔT<=min(Δt_{nq}) as illustrated in Fig. 3.8, the accuracy control technique in (45) creates a bounding box for the excursions of all state variables. All state variables which are programed to follow the solid line should be reevaluated at the edge of the bounding box depicted by the dashed arrows. At each iteration, this technique limits the maximum change for the leading state variable. The refined responses for the stiff system and the underdamped system with various accuracies α% are shown in Fig. 3.9&3.10, respectively.

Even with the availability of excursion-limited accuracy control mechanism, the property of ringing suppression for the RLC circuit is still there as shown in Fig. 3.10. To attain utmost accuracy, underdamping must be captured by the proposed algorithm, thus, extra tolerance control for these quiescent state variables is needed when the excursion-limited accuracy control mechanism is only required by these dynamic state variables. The tolerance control function can be expressed as

$$\boldsymbol{err_q}(\boldsymbol{x}) = \frac{\boldsymbol{q_q}(\boldsymbol{x}(t+\Delta t)) - \boldsymbol{q_q}(\boldsymbol{x}(t))}{\Delta t} + \boldsymbol{f_q}(\boldsymbol{x}(t+\Delta t)) \tag{46}$$

where $\boldsymbol{err_q}$ represents the error of quiescent state variables, if quiescent state variables with errors larger than the default tolerance value *tol*=1e^{-6}, then quiescent state variables rejoin the dynamic group and the circuit gets reevaluated. If we set the accuracy control parameter α% to be the coefficient of the default tolerance value as (*tol*·α%) and re-simulate the circuit in Fig. 3.6, we have the transient response as shown in Fig. 3.11.

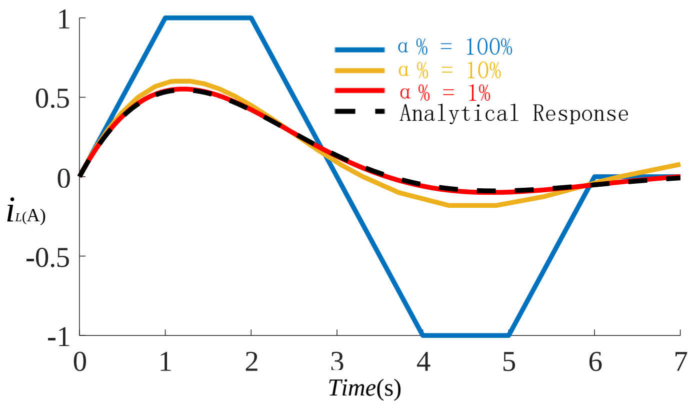


Figure 3.11 The refined response of i_L for the underdamped system in Fig. 3.6.

3.4 Performance Evaluation

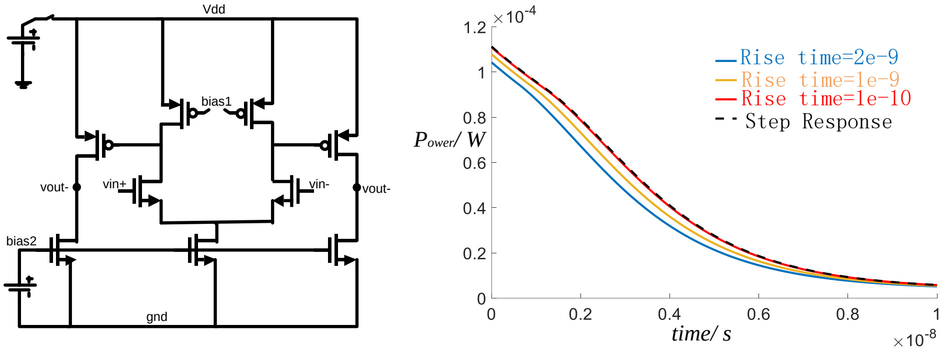


Figure 3.12 Operational amplifier circuit and its transient response.

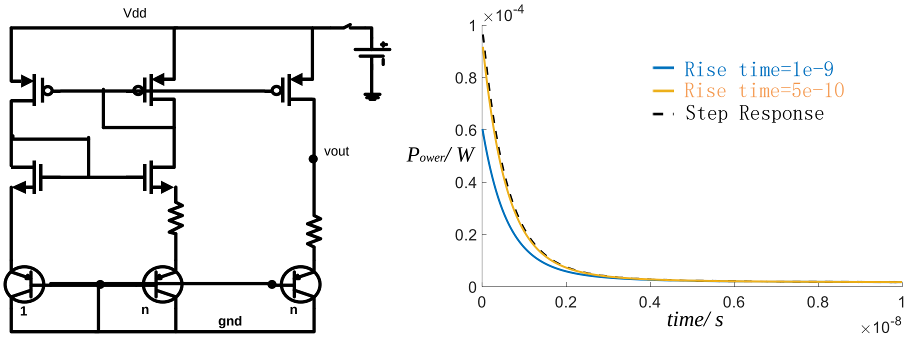


Figure 3.13 Bandgap circuit and its transient response.

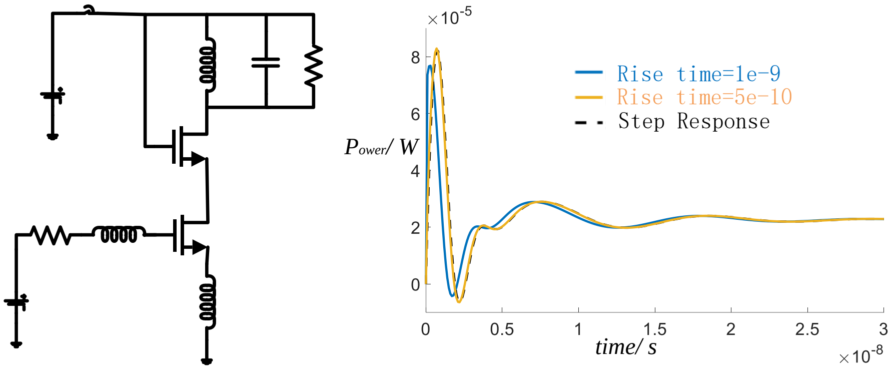


Figure 3.14 An underdamped nonlinear circuit and its response.

An operational amplifier, a bandgap circuit and an underdamped nonlinear circuit, are shown, respectively, in Figs. 3.12 to 3.14. These classic circuit examples are chosen because they are, respectively, sensitive to the input signal, insensitive to the input signal and underdamped. They are all simulated with BE integration method and with the proposed algorithm.

The transient response differences between the two integration methods, with $\alpha\%=0.1\%$ for the proposed algorithm, and multiple rise times to maximally mimic step-like functions are performed with the BE integration approximation method in MAPP. In MAPP, simulation starts when the input has already climbed to VDD. Thus, the latter suppresses stiff system information. The power flowing into that node VDD versus time is shown in Figs. 3.12 to 3.14, respectively.

As the rise time of the steep ramp becomes faster, the power curve becomes closer to the response of a step function. When high accuracy $\alpha\%$ is specified, the proposed algorithm generates simulation results that are essentially identical to those of BE integration methods.

In order to illustrate the computational effort of static driven method, a large-scale linear circuit is presented in Fig. 3.15 where $R1=R2=\cdots=1$ and $C1=C\text{n}/n$; $C2=2\cdot C\text{n}/n$; $C\text{n}=1$. To render this example a stiff system, *n* is selected to be 1000. The run time is defined as the CPU time spent on matrix manipulation for both algorithms and the run time ratio between the proposed method and BE with the same accuracy is shown in Fig. 3.16.

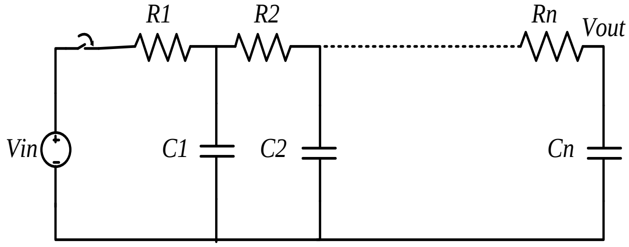


Figure 3.15 Schematic of an RC ladder circuit.

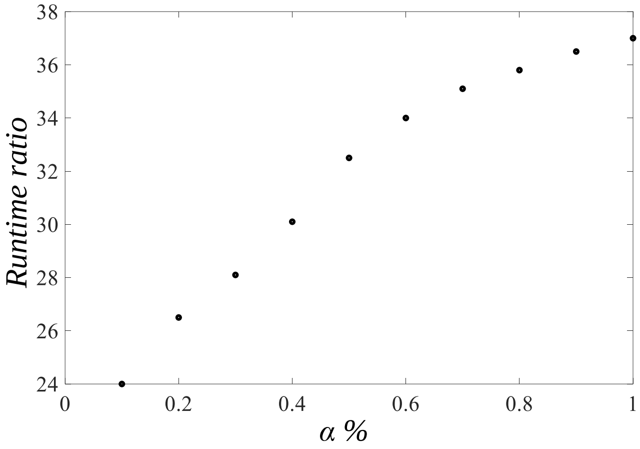


Figure 3.16 Run-time ratio versus accuracy control parameter with 1000 node RC ladder.

3.5 Pseudo transient DC convergence applying static driven method

Pseudo transient continuation is the most reliable method for computing the DC convergence point when other presumably faster DC convergence methods fail. Unlike the NR method, which ignores the effects of capacitance and inductance memory elements, the pseudo transient method approaches the steady state by roughly following the transient trajectory.

Moreover, following the exact physical transient may not be necessary as the intermediate states are not of interest.

Underdamped/oscillating and stiff systems are the main cause of slowing the transient progress [34-35]. Overcoming the underdamped/oscillating and stiff system aspects of circuit behavior, while retaining the stability of the transient process, is the key to improving the efficiency of pseudo transient simulation. Based on above, applying the static-driven method not only shortens the run time, but also suppresses the oscillating as shown in Fig. 3.7 when the second order time derivatives are controlled. The proposed pseudo transient method and its performance are discussed as follows.

The general form of the first and second order time derivatives of the system can be represented as

$$\dot{\boldsymbol{v}}_{\boldsymbol{n}} = \boldsymbol{g}(\boldsymbol{v}_{\boldsymbol{n}}, \boldsymbol{u}) \tag{47a}$$

$$\ddot{\boldsymbol{v}}_{\boldsymbol{n}} = \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{v}_{\boldsymbol{n}}} \cdot \dot{\boldsymbol{v}}_{\boldsymbol{n}} + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}} \cdot \dot{\boldsymbol{u}} \tag{47b}$$

where ***u*** stands for the input functions, and for simplicity step function excitations are presumed. If the node voltage vector $\boldsymbol{v_n}$ is decomposed into two groups $\boldsymbol{v_{st}}$ and $\boldsymbol{v_{dy}}$, then (5b) can be rewritten as

$$\begin{bmatrix} \ddot{\mathbf{v}}_{st} \\ \ddot{\mathbf{v}}_{dy} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \cdot \begin{bmatrix} \dot{\mathbf{v}}_{st} \\ \dot{\mathbf{v}}_{dy} \end{bmatrix} \tag{47c}$$

and according to the indices of the two groups, the matrix is decomposed into submatrices. To force the quiescent group to have zero second order time derivatives, the first order time derivative needs to be modified accordingly as

$$\ddot{\boldsymbol{v}}_{st} = J_{11} \cdot \dot{\boldsymbol{v}}_{st}{}^{+} + J_{12} \cdot \dot{\boldsymbol{v}}_{dy} = 0 \tag{48a}$$

$$\dot{\boldsymbol{v}}_{st}{}^{+} = -J_{11}{}^{-1} \cdot J_{12} \cdot \dot{\boldsymbol{v}}_{dy} \tag{48b}$$

and the dynamic group retains the same time derivative.

As the node voltages step to the next time point at different rates, the node voltage with the smallest approximated time constant enters the quiescent group earlier than other node voltages. Sequentially, all node voltages join the quiescent group and approach the universal steady state.

The static-driven method provides accurate transient analysis with proper error control [36]. But in the pseudo transient method, intermediate accuracy is not of concern, thus, sending the node voltages with the same order of magnitude approximated time constants to the quiescent group at the same time is acceptable. Based on the above, the new goal here is to set the quiescent group to have zero time derivative at the next time point, thus, the time derivative of the next time point can be represented as

$$\begin{aligned} \dot{\boldsymbol{v}}_{\boldsymbol{n}}(t+\Delta t) &= \boldsymbol{g}(\boldsymbol{v}_{\boldsymbol{n}}(t) + \dot{\boldsymbol{v}}_{\boldsymbol{n}}{}^{+} \cdot \Delta t, \boldsymbol{u}) = 0 \\ &= \boldsymbol{g}(\boldsymbol{v}_{\boldsymbol{n}}(t), \boldsymbol{u}) + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{v}_{\boldsymbol{n}}} \cdot \dot{\boldsymbol{v}}_{\boldsymbol{n}}{}^{+} \cdot \Delta t \\ &= \dot{\boldsymbol{v}}_{\boldsymbol{n}}(t) + \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{v}_{\boldsymbol{n}}} \cdot \dot{\boldsymbol{v}}_{\boldsymbol{n}}{}^{+} \cdot \Delta t \end{aligned} \tag{49a}$$

and the modified time derivative of the quiescent group can be calculated as

$$\dot{\boldsymbol{v}}_{st}{}^{+} = -[J_{11}]^{-1} \cdot [\frac{\dot{\boldsymbol{v}}_{st}(t)}{\Delta t} + J_{12} \cdot \dot{\boldsymbol{v}}_{dy}] \tag{49b}$$

Unlike (48a) which forces the second order time derivative to be zero to constrain the quiescent group at the next time point, (49b) minimizes the time derivatives of the quiescent group at the next time point. If the time derivatives of quiescent elements have already been successfully minimized, then (49b) becomes essentially the same as (48b) as

$$
\begin{aligned}
\dot{\boldsymbol{v}}_{st}{}^{+} &= -[J_{11}]^{-1}\cdot[\frac{\dot{\boldsymbol{v}}_{st}(t)\approx 0}{\Delta t}+J_{12}\cdot\dot{\boldsymbol{v}}_{dy}] \\
&= -[J_{11}]^{-1}\cdot[J_{12}\cdot\dot{\boldsymbol{v}}_{dy}]
\end{aligned} \tag{50}
$$

Finally, all node voltages are occupied in the quiescent group and the time derivative calculated from (49a) remains zero.

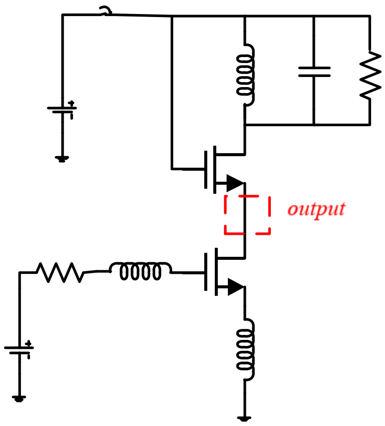


Figure 3.17 Circuit schematic for an LNA.

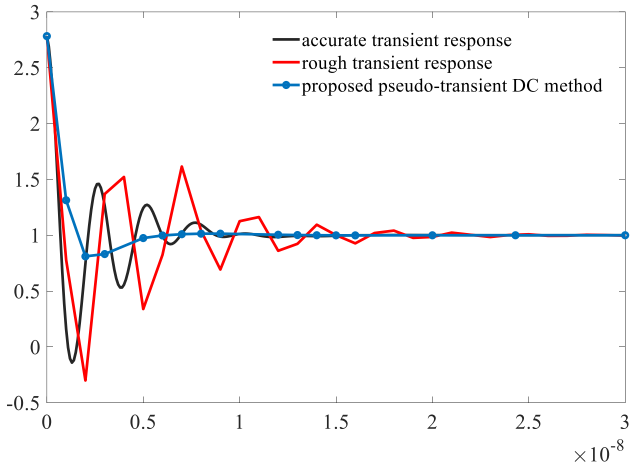


Figure 3.18 Accurate, rough, and pseudo transient response of the LNA of Fig. 3.17.

TABLE 3.2 Static-driven Pseudo Transient Method

```
Algorithm Flow:

Input:      Circuit netlist S; Excitation function u;
            Accuracy parameter α%;
Output      Time points vector tpts;
:           Converged node vector $\boldsymbol{v}_n$;
Find the first and second order time derivative of node vector based on chosen integration method;
Add the node with opposite sign of first and second order time derivative to the quiescent group and the rest to dynamic group;
$\boldsymbol{v}_n = [\boldsymbol{v}_{st} ; \boldsymbol{v}_{dy}]$;
While       $\boldsymbol{F}(\boldsymbol{v}_n(t)) >= 1e\text{-}6$
    Calculate $\dot{\boldsymbol{v}}_{st}^{+}$ and $\dot{\boldsymbol{v}}_{dy}$ based on (49b) and (47a);
    Remove $\dot{\boldsymbol{v}}_{st}^{+}$ and $\dot{\boldsymbol{v}}_{dy}$ that equal zero to avoid being divided by zero in the following calculations;
    Get $\Delta t_{st}$ as $\frac{Vdd \cdot \alpha\%}{\dot{\boldsymbol{v}}_{st}^{+}}$ for every quiescent element;
    Get $\Delta t_{dy}$ as $\frac{Vdd \cdot \alpha\%}{\dot{\boldsymbol{v}}_{dy}}$ for every dynamic element;
    $\Delta t = \min(\Delta t_{st}, \Delta t_{dy})$ ;
    $\boldsymbol{v}_{dy}(t+\Delta t) = \boldsymbol{v}_{dy}(t) + \dot{\boldsymbol{v}}_{dy} \cdot \Delta t \;\; \boldsymbol{v}_{st}(t+\Delta t) = \boldsymbol{v}_{st}(t) + \dot{\boldsymbol{v}}_{st}^{+} \cdot \Delta t$ ;
End
Return      tpts, $\boldsymbol{v}_n$;
```

Incidentally, no matter whether (48a) or (49a) is applied, the control technique of the second order time derivative suppresses ringing [36] which is beneficial for dealing with the underdamped region of the pseudo transient response. An illustrative underdamped circuit example is shown in Fig. 3.17 and its pseudo transient response applying (49a) is shown in Fig. 3.18.

For summary purposes, the general flow is presented as in Table 3.2 above. It is general and admits any pseudo transient integration approximation method.

Based on the algorithm flow, two more circuit examples including a comparator and a bandgap are tested in terms of both the proposed pseudo transient method and the conventional pseudo transient method. Fig. 3.19 shows the schematic of the comparator and Fig. 3.20 shows the response of the proposed pseudo transient response compared against the accurate and rough transient response. The time step size of the entire pseudo transient process is monitored and shown in Fig. 3.21.

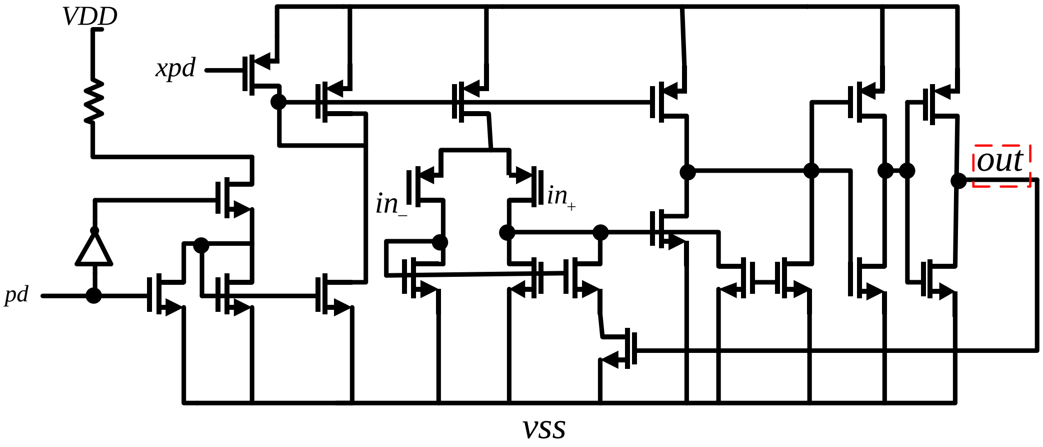


Figure 3.19 Schematic of the comparator circuit.

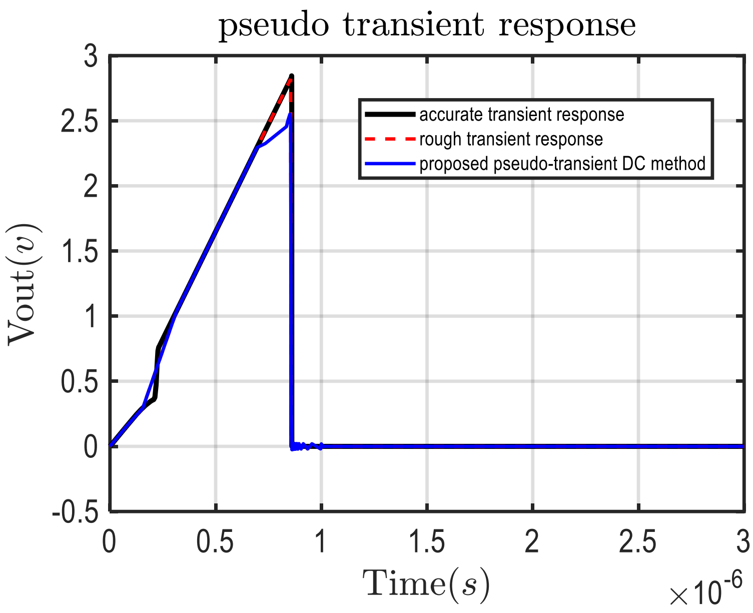


Figure 3.20 Accurate, rough, and pseudo transient response of comparator.

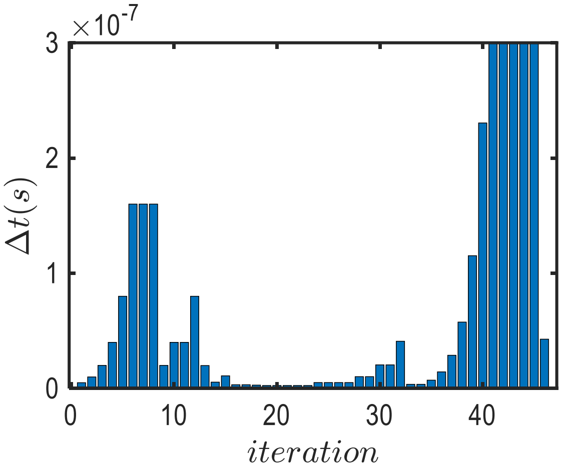


Figure 3.21 Time step size history of the comparator circuit.

The same experiment is deployed on the bandgap circuit as shown in Fig. 3.22. The pseudo transient response and its benchmark are shown in Fig. 3.23. The time step size is recorded as in Fig. 3.24. According to the time step size control technique, small time derivatives dictate a large time step which eventually becomes infinity in the universal steady state. When new node voltages with the same order of magnitude approximate time constants join the quiescent group and thereby reorganize the submatrix J11, the time derivatives of the node voltages that remain in the dynamic group surge. They reduce eventually until the next group joins quiescence as is shown in Fig. 3.21 and Fig. 3.24.

A performance summary table which includes the CPU time, number of iterations and DC convergence methods for the above three examples is shown in Table 3.3.

Table 3.3 Pseudo transient performance summary

Name of circuit	# Of iterations	CPU time (s)	DC convergence method
LNA	18	0.0010	NR
	31	0.0018	Conventional Pseudo Tran
	16	0.0012	Proposed Pseudo Tran
Comparator	-	-	NR
	181	0.01	Conventional Pseudo Tran
	49	0.0023	Proposed Pseudo Tran
Bandgap	-	-	NR
	215	0.0083	Conventional Pseudo Tran
	61	0.0029	Proposed Pseudo Tran

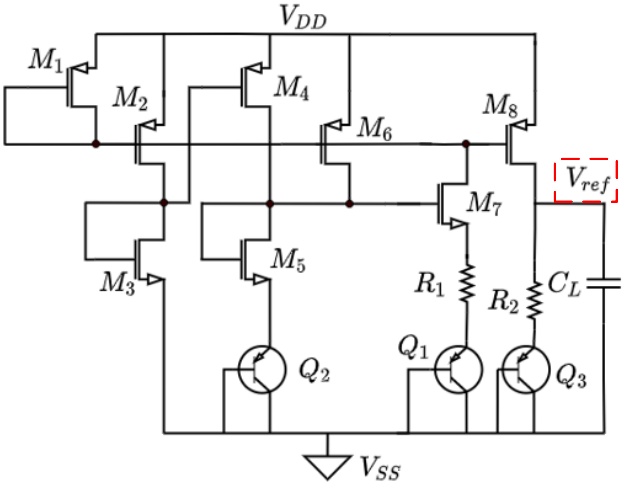


Figure 3.22 Schematic of the bandgap circuit.

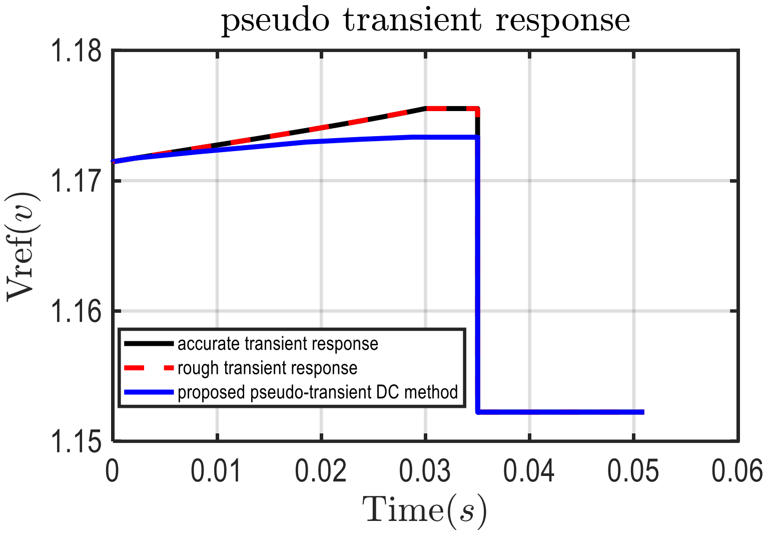


Figure 3.23 Accurate, rough, and pseudo transient response of the bandgap circuit.

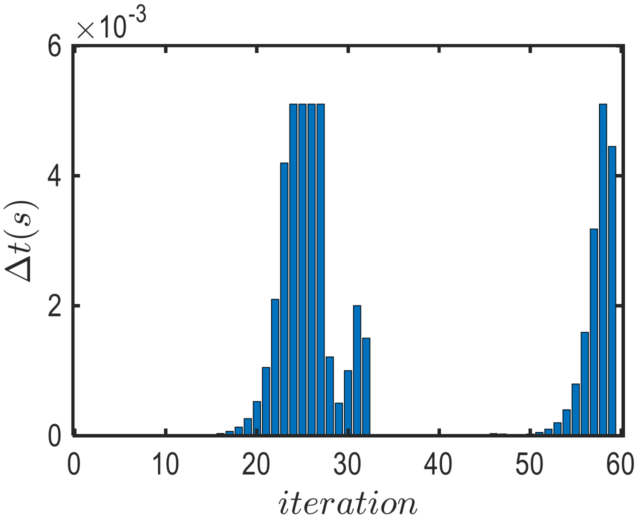


Figure 3.24 Time step size history of the bandgap circuit.

CHAPTER 4

Sensitivity analysis is crucial in both Integrated Circuit (IC) synthesis [37]-[38] and verification [39]-[43] tasks. The goal of sensitivity analysis is to calculate the partial derivatives of circuit performances (e.g., node voltages, branch currents, circuit specifications) with respect to circuit design parameters (e.g., transistor width/length) or process parameters (e.g., oxide thickness, threshold voltage). These obtained derivatives can be used in a variety of tasks including gradient based circuit design and optimization [37]-[38], noise analysis [39], fault detection [40]-[41], or yield estimation [42]-[43].

In general, the nominal response of any circuit can be represented by $\boldsymbol{v}_{\mathbf{n}}(t/w,p)$ for both frequency and time domain, and the response with perturbed parameter is $\boldsymbol{v}_{\mathbf{n}}(t/w,p+\delta p)$. The time domain illustrative example is shown in Fig. 4.1. The difference between parameters ($p+\delta p$) and (p) is an infinitesimal perturbation, and a finite parameter change is defined as Δp which will be discussed later. Any parameter's sensitivity can be expressed as

$$\frac{\partial \boldsymbol{v}_{\boldsymbol{n}}}{\partial p} = \frac{\boldsymbol{v}_{\boldsymbol{n}}(t/w, p+\delta p) - \boldsymbol{v}_{\boldsymbol{n}}(t/w, p)}{\delta p} \tag{51}$$

and such sensitivity is a first order partial derivative with respect to the specified parameter. If the sensitivity of any node voltage with respect to all parameters is available, then, any perturbed response can be estimated to be

$$\boldsymbol{v}_{\boldsymbol{n}}(t/w, p_1+\Delta p_1, \cdots, p_n+\Delta p_n) = \boldsymbol{v}_{\boldsymbol{n}}(t/w, p_1, \cdots, p_n) + \frac{d\boldsymbol{v}_{\boldsymbol{n}}}{dp_1}\cdot \Delta p_1 + \cdots + \frac{d\boldsymbol{v}_{\boldsymbol{n}}}{dp_n}\cdot \Delta p_n \tag{52}$$

For modern ICs millions of parameters are inevitable, and the proposed adjoint sensitivity analysis provides a solution to attain all sensitivities with respect to all possible parameters in one simulation.

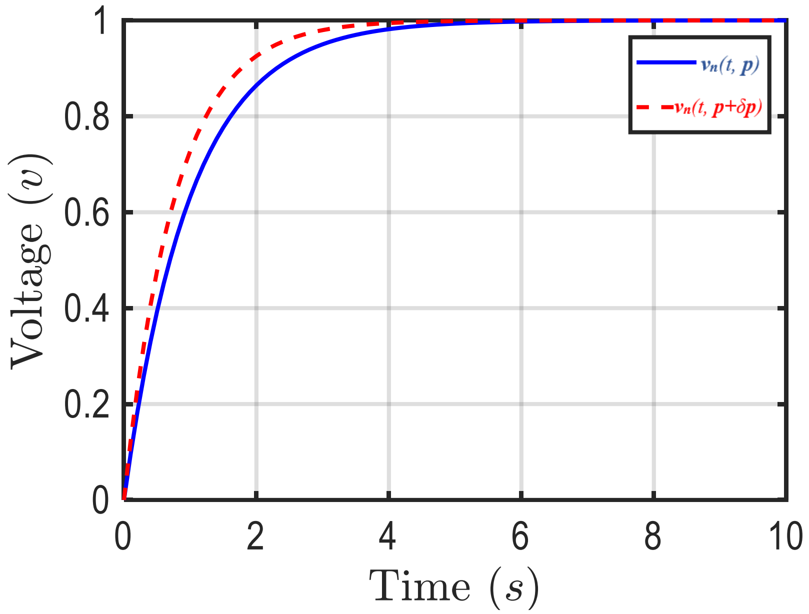


Figure 4.1 Transient response with parameter *p* and (*p*+δ*p*).

4.1 DC/AC Adjoint Sensitivity Analysis

Here we briefly review the adjoint circuit analysis for a circuit containing only independent sources, resistances and conductances as follows.

Given a circuit, we short-connect all independent voltage sources (if any), open all independent current sources (if any), and finally add a unit current source ϕ_{ad}=1A between the node of interest and ground, yielding the adjoint circuit as shown in Fig. 4.2. In other words, the adjoint circuit has the same topological connection as original circuit but with an independent unit source connected to the node of interest.

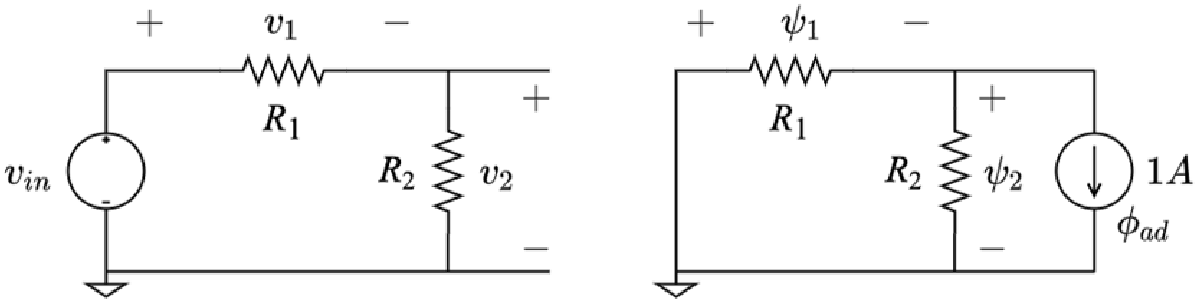


Figure 4.2 Illustration of original circuit and its corresponding adjoint version.

Tellegen’s theorem [44] tells us that:

$$\sum_B \psi_B(t) \cdot i_B(t) = 0 \tag{53a}$$

$$\sum_B \phi_B(t) \cdot v_B(t) = 0 \tag{53b}$$

where the branch voltage and current are denoted by $v_B(t)$ and $i_B(t)$, respectively, while in the adjoint circuit, they are denoted by $\psi_B(t)$ and $\phi_B(t)$. As we are now considering the steady state case, $\{v_B, i_B, \psi_B, \phi_B\}$ are all constant in time, hence, their dependences on time is omitted for brevity.

Consider a perturbation δ in the original circuit that causes a shift of branch currents by δi_B and a shift of branch voltages by δv_B. Re-applying Tellegen’s theorem gives us:

$$\sum_B \psi_B \cdot (i_B + \delta i_B) = 0 \tag{54a}$$

$$\sum_B \phi_B \cdot (v_B + \delta v_B) = 0 \tag{54b}$$

Subtracting (54) from (53), we have:

$$\sum_B \psi_B \cdot \delta i_B = 0 \tag{55a}$$

$$\sum_B \phi_B \cdot \delta v_B = 0 \tag{55b}$$

$$\sum_B \left(\psi_B \cdot \delta i_B - \phi_B \cdot \delta v_B\right) = 0 \tag{55c}$$

The above equation can be further simplified based on the categories of branches as:

$$\begin{aligned}\sum_{src}\left(\phi_{src}\delta v_{src} - \psi_{src}\delta i_{src}\right) &= \sum_R \left(\psi_R \delta i_R - \phi_R \delta v_R\right) \\ &\quad + \sum_G \left(\psi_G \delta i_G - \phi_G \delta v_G\right)\end{aligned} \tag{56}$$

In the adjoint circuit, all devices keep the same characteristic as in original circuit. For a resistive branch as example, we have the branch constituent relation as $\psi_R = R \cdot \phi_R$. Based on that, a first-order perturbative expansion of the differential component of the corresponding branch constituent relation yields:

$$\delta v_R = \delta R \cdot i_R + R \cdot \delta i_R \tag{57a}$$

$$\delta \psi_R = \delta R \cdot \phi_R + R \cdot \delta \phi_R \tag{57b}$$

Now the left hand side and right hand side of (56) can be simplified as

$$\begin{aligned}&\left(\phi_{in}\delta v_{in} - \psi_{in}\delta i_{in}\right) + \left(\phi_{ad}\delta v_{ad} - \psi_{ad}\delta i_{ad}\right) \\ &= \left(\phi_{in} \cdot 0 - 0 \cdot \delta i_{in}\right) + \left(1 \cdot \delta v_{ad} - \psi_{ad} \cdot 0\right) \\ &= \delta v_{ad} = \delta v_2\end{aligned} \tag{58a}$$

$$\begin{aligned}&\left(\psi_{R1}\delta i_{R1} - \phi_{R1}\delta v_{R1}\right) + \left(\psi_{R2}\delta i_{R2} - \phi_{R2}\delta v_{R2}\right) \\ &= -i_1\phi_1\delta R_1 - i_2\phi_2\delta R_2\end{aligned} \tag{58b}$$

After solving both original and adjoint circuits, we substitute the values of {v_B, i_B, ψ_B, ϕ_B } back into (58), yielding:

$$\delta v_2 = \frac{R_2 \cdot v_{in}}{\left(R_1 + R_2\right)^2} \cdot \delta R_1 - \frac{R_1 \cdot v_{in}}{\left(R_1 + R_2\right)^2} \cdot \delta R_2 \tag{59}$$

In the general case the sensitivity of any node voltage can be summarized as

$$\delta v_{ad} = \sum_R -i_R \phi_R \delta R + \sum_G v_G \psi_G \delta G \tag{60}$$

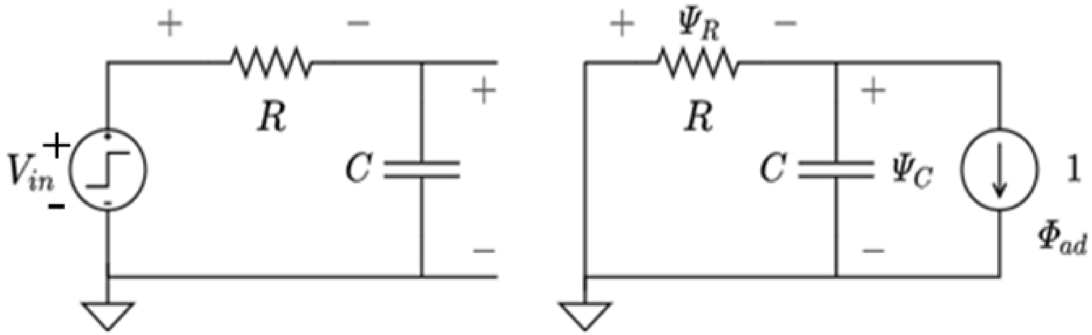


Figure 4.3 Illustration of the original RC circuit and its adjoint circuit in frequency domain.

The sensitivity in frequency domain is a function with extension to (60) as

$$\delta V_{ad} = \sum_R -I_R \Phi_R \delta R + s\sum_C V_C \Psi_C \delta C \tag{61}$$

Solving the original and adjoint circuits gives:

$$I_R(s) = \frac{C}{1+sRC} \tag{62a}$$

$$V_C(s) = \frac{1}{s}\frac{1}{1+sRC} \tag{62b}$$

$$\Phi_R(s) = \frac{1}{1+sRC} \tag{62c}$$

$$\Psi_C(s) = -\frac{R}{1+sRC} \tag{62d}$$

Substituting (62) into (61) gives:

$$\delta V_{ad} = -\frac{C}{\left(1+sRC\right)^2}\delta R - \frac{R}{\left(1+sRC\right)^2}\delta C \tag{63}$$

Which coincides with direct differentiation of V_c(s) given in the RC circuit. It is straightforward to analyze sensitivities in the frequency domain, while the same analysis is complicated to do so in the time domain since the convolution of the calculated circuit responses [38] is required and that convolution in the time domain is equivalent to multiplication in the frequency domain.

4.2 Forward-In-Time Adjoint Analysis

Instead of calculating the time domain sensitivity in the convolutional way, the difficulty of time domain adjoint analysis is that the time-varying behavior of the circuit needs to be taken into account. Time varying in the sense that new linearizations of nonlinear elements result in new linear circuits at new time points. Thus, the correct modeling of memory elements and the time dependence equations from time t to t+Δt is necessary.

The discretized equations for capacitances and inductances in terms of the BE integration approximation can be represented as

$$i_C\left(t+\Delta t\right) = C \cdot \frac{v_C\left(t+\Delta t\right) - v_C\left(t\right)}{\Delta t} \tag{64a}$$

$$v_L\left(t+\Delta t\right) = L \cdot \frac{i_L\left(t+\Delta t\right) - i_L\left(t\right)}{\Delta t} \tag{64b}$$

Based on the above equations, an equivalent companion model for capacitor, inductor or any general two terminal nonlinear device can be obtained as shown in Fig. 4.4, Fig. 4.5, and Fig. 4.6.

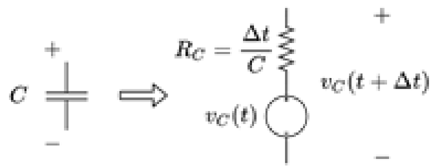


Figure 4.4 Illustration of the BE companion model for a capacitance.

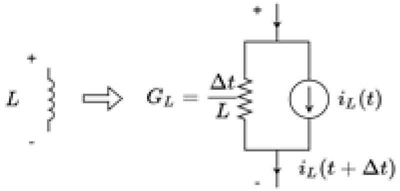


Figure 4.5 Illustration of the BE companion model for an inductance.

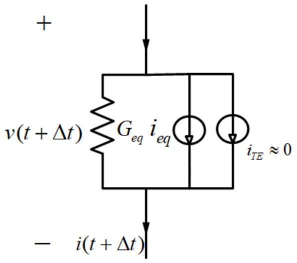


Figure 4.6 Illustration of the linearized BE companion model for nonlinear device.

If we restart deriving the Tellegen's theorem in (55c), then for a capacitor we have:

$$
\begin{aligned}
&\underbrace{-\left(\phi_{v_C}\left(t+\Delta t\right)\delta v_{v_C}(t)-\psi_{v_C}\left(t+\Delta t\right)\delta i_{v_C}(t)\right)}_{\text{related to } v_C(t)}+\\
&\underbrace{\left(\phi_{R_C}\left(t+\Delta t\right)\delta v_{R_C}(t)-\psi_{R_C}\left(t+\Delta t\right)\delta i_{R_C}(t)\right)}_{\text{related to R}_C}\\
&=-\left(\phi_{v_C}\left(t+\Delta t\right)\delta v_{v_C}(t)-0\cdot\delta i_{v_C}(t)\right)\\
&\quad -i_C\left(t+\Delta t\right)\phi_C\left(t+\Delta t\right)\delta R_C\\
&=-\phi_C\left(t+\Delta t\right)\delta v_C\left(t\right)\\
&\quad -i_C\left(t+\Delta t\right)\phi_C\left(t+\Delta t\right)\delta R_C
\end{aligned}
\tag{65}
$$

For an inductor:

$$\begin{aligned}
&\underbrace{-\left(\phi_{i_L}\left(t+\Delta t\right)\delta v_{i_L}\left(t+\Delta t\right)-\psi_{i_L}\left(t+\Delta t\right)\delta i_{i_L}\left(t+\Delta t\right)\right)}_{\text{related to } \mathrm{i}_L(t)}+\\
&\underbrace{\left(\phi_{R_C}\left(t+\Delta t\right)\delta v_{R_C}(t)-\psi_{R_C}\left(t+\Delta t\right)\delta i_{R_C}(t)\right)}_{\text{related to } \mathrm{G}_L}\\
&=-\left(0\cdot\delta v_{i_L}\left(t+\Delta t\right)-\psi_{i_L}\left(t+\Delta t\right)\delta i_{i_L}\left(t\right)\right)\\
&\quad -i_L\left(t+\Delta t\right)\phi_L\left(t+\Delta t\right)\delta R_L\\
&=\psi_L\left(t+\Delta t\right)\delta i_L\left(t\right)\\
&\quad -i_L\left(t+\Delta t\right)\phi_L\left(t+\Delta t\right)\delta R_L
\end{aligned} \tag{66}$$

For a nonlinear device:

$$\begin{aligned}
&\left(\psi_{G_{eq}}\delta i_{G_{eq}}-\phi_{G_{eq}}\delta v_{G_{eq}}\right)+\left(\psi_{G_{eq}}\delta i_{i_{eq}}-\phi_{i_{eq}}\delta v_{i_{eq}}\right)+\left(\psi_{G_{eq}}\delta i_{TE}-\phi_{TE}\delta v_{TE}\right)\\
&\overset{1}{=}\left(\psi_{G_{eq}}\delta i_{G_{eq}}-\phi_{G_{eq}}\delta v_{G_{eq}}\right)+\psi_{G_{eq}}\delta i_{i_{eq}}+\psi_{G_{eq}}\delta i_{TE}\\
&\overset{2}{=}\psi_{G_{eq}}\delta i_P-\phi_{G_{eq}}\delta v_P+\psi_{G_{eq}}\delta i_{TE}\\
&\overset{3}{=}\psi_{G_{eq}}\left[\frac{\partial i}{\partial v}\cdot\delta v_P+\left(\frac{\partial i}{\partial \mathbf{p}}\right)^T\cdot\delta\mathbf{p}\right]-\phi_{G_{eq}}\delta v_P+\psi_{G_{eq}}\delta i_{TE}
\end{aligned} \tag{67}$$

After all related functions have been prepared, RC-Diode circuit, a demonstrative example is shown in Fig. 4.7 along with its adjoint version shown in Fig. 4.8. And the sensitivity of the node of interest for this example can be derived as

$$\begin{aligned}
&\delta v\left(t+\Delta t\right)\\
&=-i_R\left(t+\Delta t\right)\phi_R\left(t+\Delta t\right)\delta R\\
&\quad-\left[i_C\left(t+\Delta t\right)\phi_C\left(t+\Delta t\right)\delta R_C+\phi_C\left(t+\Delta t\right)\delta v\left(t\right)\right]\\
&\quad+\psi_D\left(t+\Delta t\right)\cdot\left(\frac{\partial i}{\partial I_0}\cdot\delta I_0+\frac{\partial i}{\partial V_T}\cdot\delta V_T\right)\\
&\quad+\psi_D\left(t+\Delta t\right)\cdot\delta i_{TE}
\end{aligned} \tag{68}$$

Solving the linearized circuit shown in Fig. 4.7 yields:

$$\begin{aligned}
&i_R\left(t+\Delta t\right)=\left[v_{in}\left(t+\Delta t\right)-v\left(t+\Delta t\right)\right]/R\\
&i_C\left(t+\Delta t\right)=\left[v\left(t+\Delta t\right)-v\left(t\right)\right]/R_C
\end{aligned} \tag{69}$$

And solving the adjoint circuit shown in Fig. 4.5 yields:

$$
\begin{aligned}
&\psi_D\left(t+\Delta t\right) = -\left(\frac{1}{R}+\frac{1}{R_C}+G_{eq}\right)^{-1} \\
&\phi_R\left(t+\Delta t\right) = -\psi_D\left(t+\Delta t\right)/R \\
&\phi_C\left(t+\Delta t\right) = \psi_D\left(t+\Delta t\right)/R_C
\end{aligned}
\tag{70}
$$

Based on (68), we have the time domain sensitivity with respect to all parameters of RC Diode circuit plotted in Fig. 4.7.

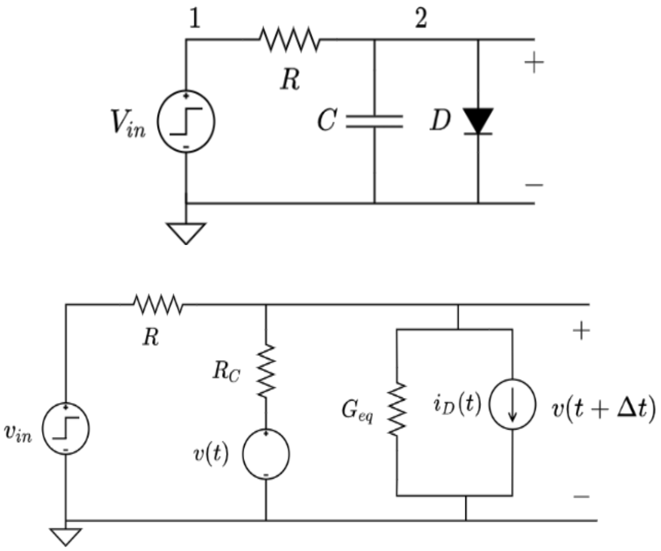


Figure 4.7 Illustration of an RC Diode circuit and its linearized version at *t*+Δ*t*.

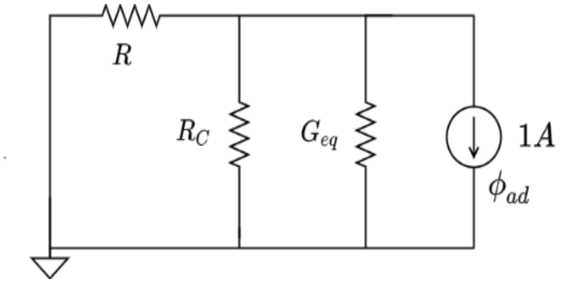


Figure 4.8 RC Diode circuit’s adjoint version at *t*+Δ*t*.

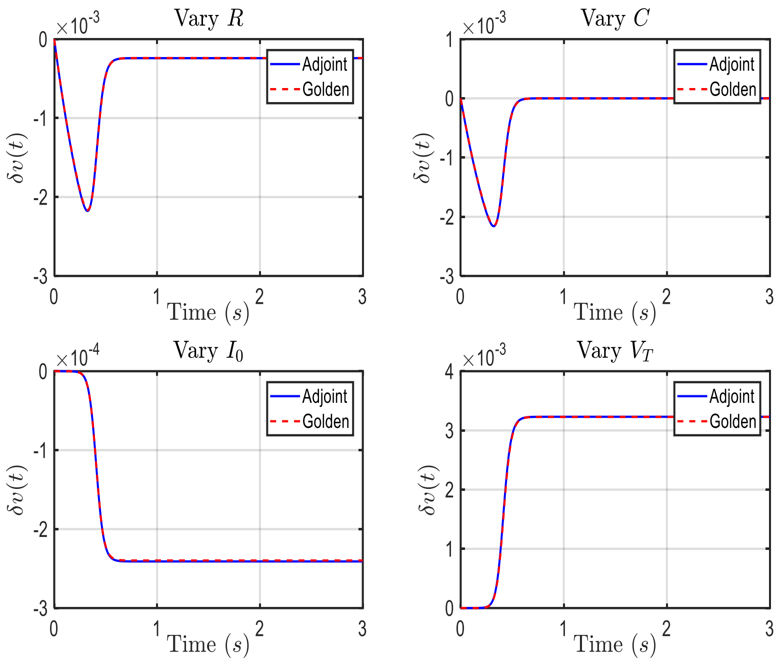


Figure 4.9 Sensitivities obtained by two methods, the adjoint curve is calculated from (68) and the dashed curve is calculated from the incremental sensitivity. (R=C=1; I_0=10^{-6}; V_T=0.25)

4.3 Examples and Error Accumulation Analysis

The above theoretical analysis assumes that both the time step and the variations of parameters are infinitesimal, but the use of an infinitesimal Δt is obviously not practical to implement and a numerical integration method is inevitable. As a nonlinear device traverses the i-v characteristic curve, the linearized terms G_{eq} and i_{eq} as shown in Fig. 4.6 will vary and the use of a finite Δt introduces a local truncation-derivative error over the integration interval. If one defines the truncation error (*TE*) of a device as the difference between the actual point and estimated point shown in Fig. 4.10, then such error becomes negligible as a proper numerical integration process is given to minimize such error. But for time domain sensitivity analysis, the derivative difference between the actual point and the estimated point with respect to the device parameters is critical and should not be ignored which is defined as the truncation-derivative error $\partial i_{TE}/\partial p$.

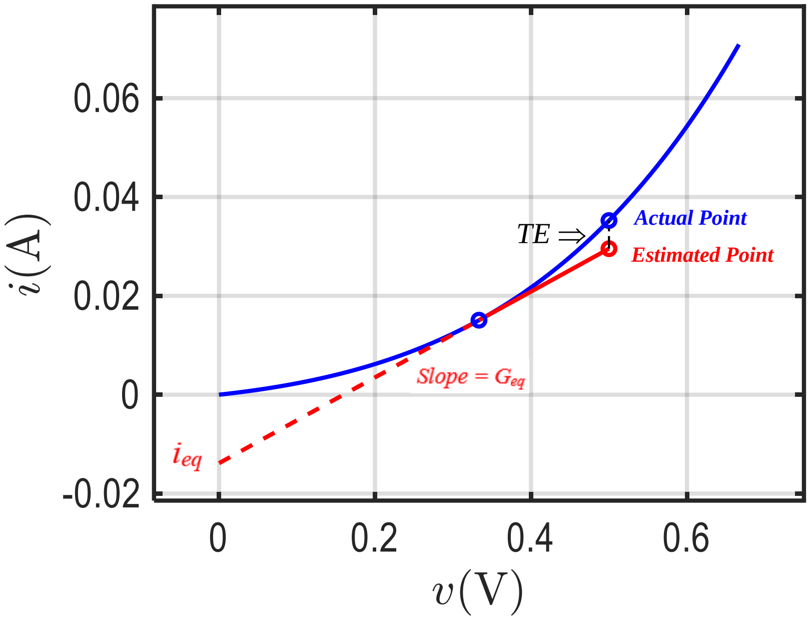


Figure 4.10 Hypothetical characteristic curve of nonlinear elements.

In the companion model of the nonlinear device the truncation error is treated as an additional branch current in the linearized element as shown in Fig. 4.6 as

$$\begin{aligned} i_{TE}(t+\Delta t) &= i_{actual}(t+\Delta t) - i_{estimated}(t+\Delta t) \\ &= i(v(t+\Delta t), p) - i(v(t), p) \\ &\quad - \frac{di}{dv}\cdot[v(t+\Delta t) - v(t)] \end{aligned} \tag{71}$$

As with many transient simulation methods, the truncation error is reasonably minimized [45]. But the mechanism of minimization does not ensure the same minimization of the truncation-derivative error which is critical in the nonlinear element's sensitivity. If the extra high order terms branch i_{TE} as shown in Fig. 4.6 are taken into calculation, then the sensitivity expression is updated to

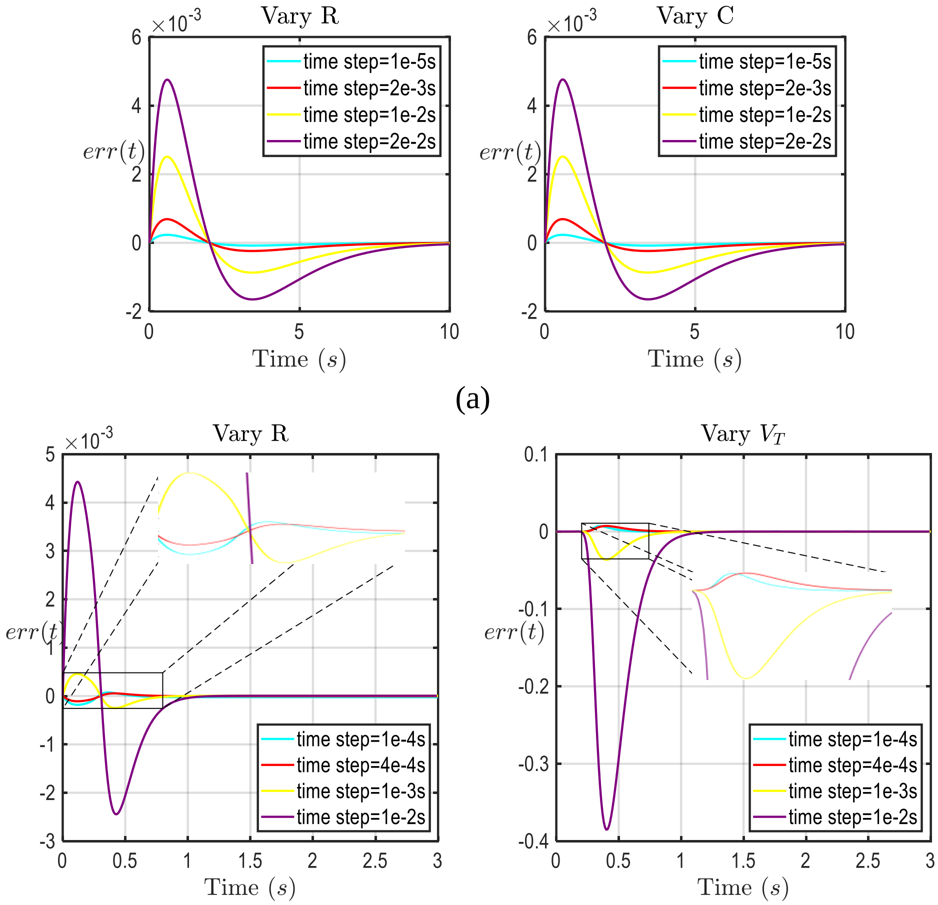
$$
\begin{aligned}
&\delta v_n\left(t+\Delta t\right)\\
&=-\sum_R i_R\left(t+\Delta t\right)\phi_R\left(t+\Delta t\right)\delta R\\
&\quad+\sum_G v_G\left(t+\Delta t\right)\psi_G\left(t+\Delta t\right)\delta G\\
&\quad-\sum_C\left[i_C\left(t+\Delta t\right)\phi_C\left(t+\Delta t\right)\delta R_C+\phi_C\left(t+\Delta t\right)\delta v_C\left(t\right)\right]\\
&\quad+\sum_L\left[v_L\left(t+\Delta t\right)\psi_L\left(t+\Delta t\right)\delta G_L+\psi_L\left(t+\Delta t\right)\delta i_L\left(t\right)\right]\\
&\quad+\sum_{nonlinear}\left[\psi_{G_{eq}}\left(t+\Delta t\right)\cdot\left(\frac{\partial i}{\partial \mathbf{p}}\Big|_{t+\Delta t}\right)^T\delta\mathbf{p}\right.\\
&\qquad\qquad\left.+\underbrace{\psi_{G_{eq}}\left(t+\Delta t\right)\delta i_{TE}\left(t+\Delta t\right)}_{\text{consider TE}}\right]
\end{aligned}
\tag{72}
$$

where δi_{TE} in (72) can be represented as

$$
\begin{aligned}
\delta i_{TE}(t+\Delta t)=&\frac{di(v(t+\Delta t),p)}{dp}\cdot\delta p-\frac{di(v(t),p)}{dp}\cdot\delta p\\
&-\frac{di}{dv}\cdot[\frac{dv(t+\Delta t)}{dp}-\frac{dv(t)}{dp}]\cdot\delta p\\
&-\frac{di}{dvdp}\cdot[v(t+\Delta t)-v(t)]\cdot\delta p
\end{aligned}
\tag{73}
$$

Based on (71) & (73), the high order term δi_{TE} will be zero if an analytical integration process is given. To view the significance of the sensitivity estimation with different time steps as shown in Fig. 4.11, both numerical solution without high order term and incremental sensitivity are applied to previously discussed examples, and for the RCD example as shown in Fig. 4.7, the error does not converge to zero even if a smaller time step is employed.

The result of Fig. 4.11 shows that a smaller time step does not improve the accuracy of adjoint sensitivity estimation if the high order term is not taken into consideration. To further illustrate the importance of corrections, the adjoint sensitivity calculation with and without taking the high order term into consideration are applied to the operational amplifier (opamp) circuit shown in Fig. 4.12. And Fig. 4.13 shows the capability of the truncation-derivative error minimization to allow accurate estimation of the parameter sensitivity.



(a)

(b)

Figure 4.11 sensitivity error curve for (a) an RC circuit, and (b) an RCD circuit with different time steps.

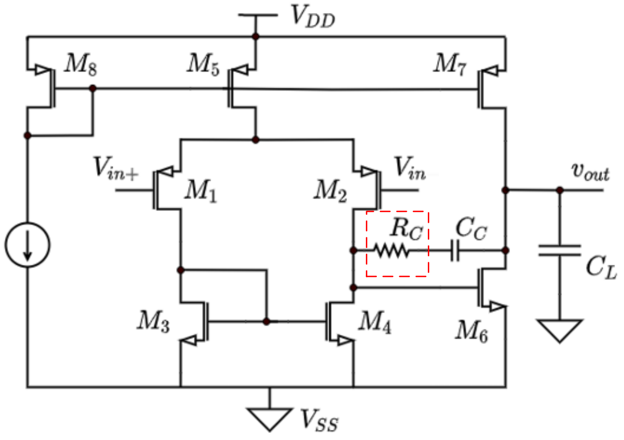


Figure 4.12 Schematic of an operational amplifier circuit.

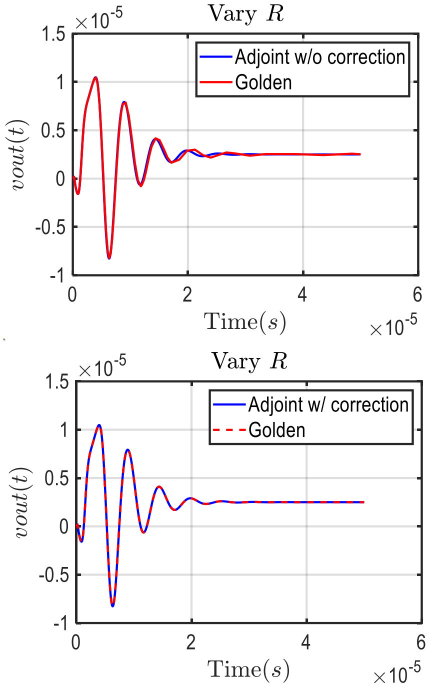


Figure 4.13 Output voltage sensitivities without and with correction excited by a step function input.

Fig. 4.11 and 4.13 straightforwardly shows the impact of the high order term, and the theoretical adjoint sensitivity error analysis is as follows. No matter what numerical integration method is applied, the general nominal excursion from t to t+Δt can be expressed as

$$J \cdot \Delta \boldsymbol{v_n} = J \cdot [\boldsymbol{v_n}(t+\Delta t) - \boldsymbol{v_n}(t)] = \boldsymbol{i_{TE}} \tag{74a}$$

Where J stands for the Jacobian matrix of the chosen integration method and the corresponding perturbed excursion expressed as

$$(J+\delta J)\cdot(\Delta \boldsymbol{v_n} + \delta \Delta \boldsymbol{v_n}) = \boldsymbol{i_{TE}} + \delta \boldsymbol{i_{TE}} \tag{74b}$$

After (74b)-(74a), we have

$$\boxed{\delta J \cdot \Delta \boldsymbol{v_n}} + J \cdot \delta \Delta \boldsymbol{v_n} + \left(\cancel{\delta J \cdot \delta \Delta \boldsymbol{v_n}}\right) = \delta \boldsymbol{i_{TE}} \tag{75}$$

Some people may ignore the boxed term $\delta J\cdot\Delta\boldsymbol{v_n}$ in (75) when calculating the desired sensitivity $\delta\Delta\boldsymbol{v_n}$, because $\delta J\cdot\Delta\boldsymbol{v_n}$ appears to be a high order term like the crossed out term in (75) when a hypothetical infinitesimal time step is applied. Suppose $\delta\boldsymbol{v_{n\_0}}$ represents the initial state of the time domain sensitivity, the sensitivity estimation for any next time point with and without taking the boxed term into consideration can be represented as

$$\begin{aligned}\delta\boldsymbol{v}_{\boldsymbol{n}\_i} &= \delta\boldsymbol{v}_{\boldsymbol{n}\_0} + \delta\Delta\boldsymbol{v_1} + \delta\Delta\boldsymbol{v}_2 + \cdots\cdots + \delta\Delta\boldsymbol{v_i} \\ &= \delta\boldsymbol{v}_{\boldsymbol{n}\_0} + J_1^{-1}\cdot\delta\boldsymbol{i_{TE\_1}} + J_2^{-1}\cdot\delta\boldsymbol{i_{TE\_2}} + \cdots\cdots + J_i^{-1}\cdot\delta\boldsymbol{i_{TE\_i}} + \\ &\boxed{J_1^{-1}\cdot(\delta J_1\cdot\Delta\boldsymbol{v_1}) + J_2^{-1}\cdot(\delta J_2\cdot\Delta\boldsymbol{v_2}) + \cdots\cdots + J_i^{-1}\cdot(\delta J_i\cdot\Delta\boldsymbol{v_i})}\end{aligned} \tag{76}$$

If $\delta\boldsymbol{v_{n\_i}}$ is used to denote the i_{th} sensitivity within the boxed term and $\delta\overline{\boldsymbol{v}}_{\boldsymbol{n\_i}}$ without the boxed term, the generalized difference between $\delta\boldsymbol{v_{n\_i}}$ and $\overline{\delta\boldsymbol{v_{n\_i}}}$ can be represented as

$$\begin{aligned}&\delta\boldsymbol{v_{n\_i}} - \overline{\delta\boldsymbol{v_{n\_i}}} = i\cdot J_1^{-1}\cdot(\delta J_1\cdot\Delta\boldsymbol{v_1}) + (i-1)\cdot J_2^{-1}\cdot(\delta J_2\cdot\Delta\boldsymbol{v_2}) + \\ &\cdots\cdots + J_i^{-1}\cdot(\delta J_i\cdot\Delta\boldsymbol{v_i})\end{aligned} \tag{77}$$

Based on (77), the difference of the sensitivity with and without taking into account the high order term into can be unignorable if enormous and potentially ill-conditioned matrix; unreasonable excursion Δv_i; and long simulation time are applied.

To verify the abovementioned correction, more nonlinear examples are tested against the incremental sensitivity result including a SRAM, a bandgap circuit, and an oscillator as shown in Fig. 4.14, Fig. 4.16 and Fig. 4.18. Their sensitivity responses against incremental sensitivities are shown in Fig. 4.15, Fig. 4.17 and Fig. 4.19 with the parameter of interest marked in the red dashed box.

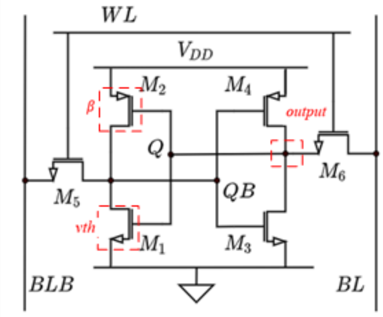


Figure 4.14 Circuit schematic for a SRAM cell.

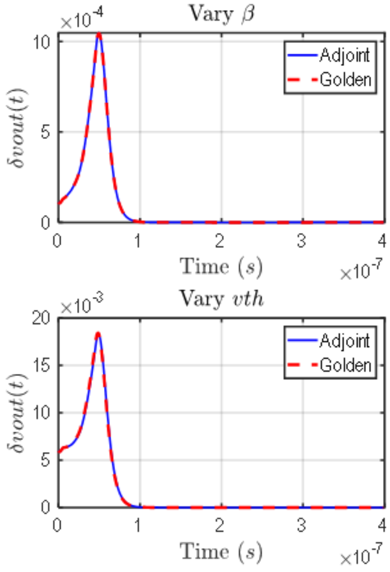


Figure 4.15 Sensitivity response for a SRAM cell.

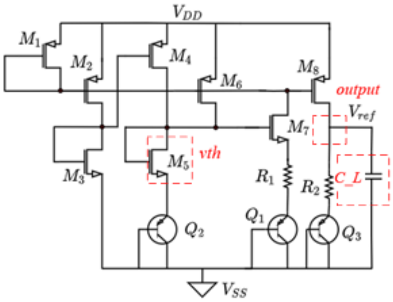


Figure 4.16 Circuit schematic for a bandgap circuit.

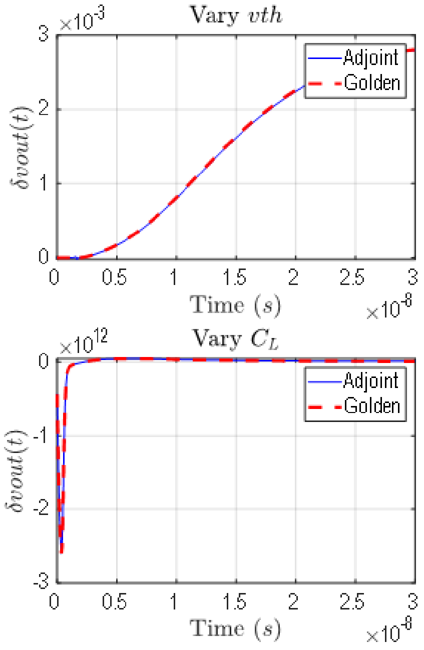


Figure 4.17 Sensitivity response for a bandgap circuit.

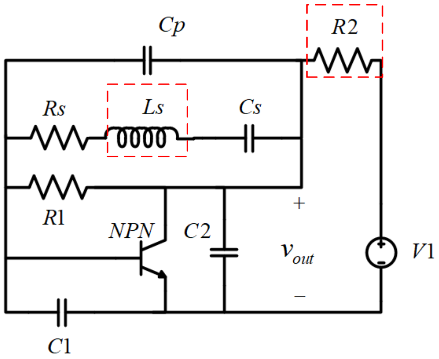


Figure 4.18 Circuit schematic for an oscillator.

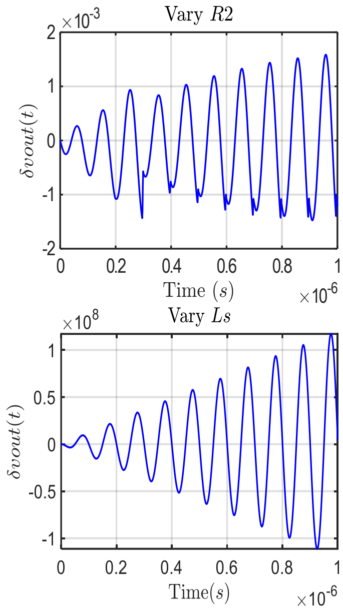


Figure 4.19 Sensitivity response for an oscillator.

4.4 Time domain noise analysis applying adjoint sensitivity

To acquire utmost accuracy, Monte-Carlo oriented time domain noise analysis [46-47] might be the only reliable solution for all general circuits. But as the name suggests, the computational power required is not affordable for the most advanced designs or processes. Moreover, with the above individual noise source contributions cannot be distinguished.

Derived from adjoint circuit theory, time domain adjoint sensitivity provides an elegant and efficient approach to compute the effects on the performance of a circuit with respect to all possible incremental parameter variations in a single simulation run and the time domain noise analysis applying adjoint sensitivity information is described as follows.

The general form of power spectral density can be represented as shown in Fig.1. The time domain noise series can be modelled as a linear combination of all of the frequency bands [48] as

$$\delta i = \sum_{x=1}^{x=M} A_x \cdot \cos(w_x \cdot t + \theta_x) \tag{78}$$

where Θ_x is randomly chosen to follow a uniform distribution over [0: 2*pi*]; A_x follows a Rayleigh distribution with variance $2S(w_x)\Delta f_x$ and $S(w)$ is defined as the spectral density function. The number of frequency bands determines the accuracy of this noise series.

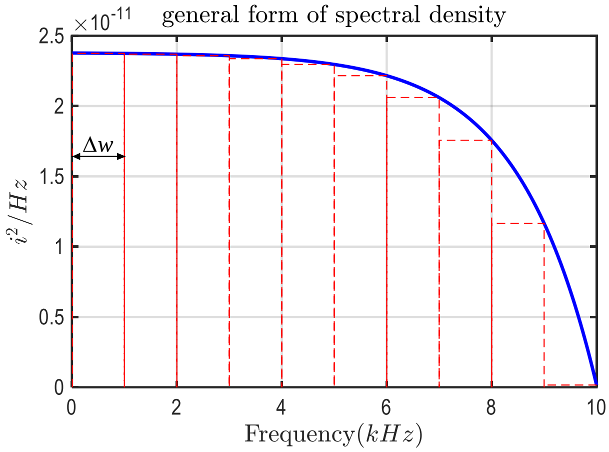


Fig. 4.20 General form of power spectral density (Note that this is a representative figure and not all power spectral densities take this form).

A noise free resistor has the characteristic function

$$i = \frac{v}{R} \tag{79a}$$

and after the noise series is added to (79a), it becomes

$$i + \delta i = \frac{v + \delta v}{R} \tag{79b}$$

Now suppose that a variation in *R* is applied to compensate for the noise series δi then

$$i = \frac{v + \delta v}{R + \delta R} \tag{79c}$$

It follows from (79b)-(79c), that the noise can be equivalently represented by a variation of the parameter *R* as

$$\begin{aligned}\delta i &= \frac{v+\delta v}{R} - \frac{v+\delta v}{R+\delta R} \\ &= \frac{v\cdot\delta R+\delta v\cdot\delta R}{R\cdot(R+\delta R)} \\ &= \frac{i}{R}\cdot\delta R\end{aligned} \tag{79d}$$

and the corresponding noise model in terms of the resistance branch can be represented as shown in Fig.2.

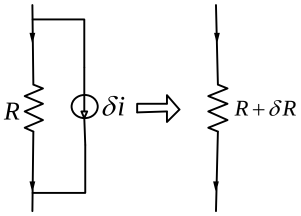


Fig. 4.21 Equivalent noise model representation.

A similar derivation can be implemented for a general device with characteristic function *i*=*f*(*v*, *p*) as follows

$$i = f(v,p) \tag{80a}$$

$$i+\delta i = f(v+\delta v, p) \tag{80b}$$

$$i = f(v+\delta v, p+\delta p) \tag{80c}$$

from (80b)-(80a), we have

$$\delta i = \frac{\partial f}{\partial v}\cdot\delta v \tag{80d}$$

and from (80c)-(80a), we have

$$0 = \frac{\partial f}{\partial v} \cdot \delta v + \frac{\partial f}{\partial p} \cdot \delta p \qquad (80e)$$

Thus, the noise of any general device can be equivalently transferred to the variation of its own parameters

$$\delta i = -\frac{\partial f}{\partial p} \cdot \delta P \qquad (80f)$$

Now Suppose Backward Euler is applied as the numerical integration method for transient analysis from t=0 to t=T. For every time point t, the noise spectral density function of the circuit is available as $\boldsymbol{s}(w, \boldsymbol{v_n}(t))$. As the function suggests, the spectral density function is both frequency dependent and circuit state dependent.

For simplicity, the implementation is only focused on the time interval Δt from t to t+Δt and such implementation can be identically applied to all the time intervals from t=0 to t=T. At t, the spectral density function is re-evaluated for each noise source as $\boldsymbol{s}(w, \boldsymbol{v_n}(t_1))$. Based on (79), randomized noise time series are generated with M frequency bands to approximate the noise input signal from t to t+Δt as

$$\delta \boldsymbol{i_n}(t) = \sum_{1_{th}\ node}^{n_{th}\ node} \sum_{x=1}^{x=M} A_x \cdot \cos(w_x \cdot t + \theta_x) \qquad (81)$$

Extra interpolated time points can be added to the time interval to increase the accuracy of the time domain adjoint sensitivity. The output noise response from t to t+Δt is available as

$$\delta \boldsymbol{v_n}(t) = \sum_{i=1}^{i=n} \frac{\partial \boldsymbol{v_n}}{\partial p_i}(t) \cdot [\delta i_i(t) / \frac{\partial f}{\partial p_i}] \qquad (82)$$

where n stands for the number of noise sources.

To verify the proposed time domain noise analysis applying sensitivities, two examples are tested with both the Monte-Carlo method and the proposed method. Fig. 4.22 shows an operational amplifier with sinusoidal signals applied to its input differential pair. The spectral density function of thermal noise of a MOSFET is

$$4KT \cdot \gamma \cdot g_{do} \quad (A^2 / Hz) \tag{83}$$

where Υ is 2/3 for long channel devices in saturation and 2 for sub-micron devices, g_{do} is the zero bias drain conductance. The spectral density function of flicker noise of a MOSFET is defined as

$$\frac{K_f}{f} \cdot \frac{g_m{}^2}{C_{ox} \cdot W \cdot L} \quad (A^2 / Hz) \tag{84}$$

where K_f is a constant, W&L are the effective width and length of the device. The spectral density function of the thermal noise of a resistor is

$$\frac{4KT}{R} \quad (A^2 / Hz) \tag{85}$$

where K is Boltzman constant, and T is absolute Kelvin temperature [49].

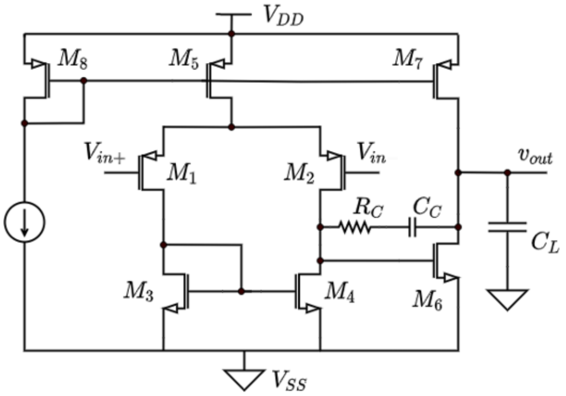


Fig. 4.22 Circuit schematic for a two stage OPAMP.

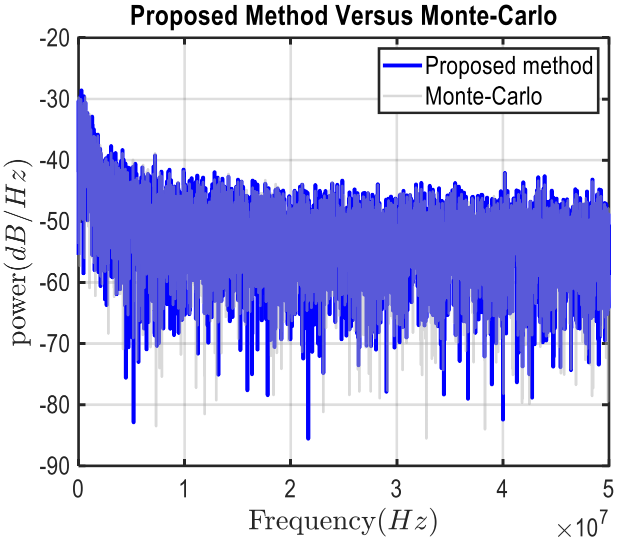


Fig. 4.23 Output noise response of a two stage OPAMP.

Both the tediously sampled Monte-Carlo noise analysis and the proposed time domain noise analysis are applied to this operational amplifier, and the results are compared in Fig. 4.23. The match at low frequencies is particularly better than other time domain transient noise analyses [50] that previously have been proposed which require a long transient simulation time to attain accuracy in the flicker noise region.

Fig. 4.24 shows an inverter. Unlike with the operational amplifier which works around a fixed bias operating point, the 0/1 transition of digital circuits poses challenges to time domain noise analysis [51]. The time domain sensitivity response of digital circuits with respect to any parameters will be an impulse-like curve and that impulse happens over the 0/1 transition time. Except for that impulse, the rest of the curve remains at zero. That is to say, the overall noise response only needs to be concerned with the transition time and more time samples are required in that region to secure the accuracy of the noise response. The comparison of this result with Monte-Carlo is shown in Fig. 4.25.

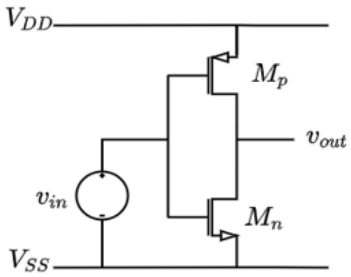


Fig. 4.24 Circuit schematic for an inverter.

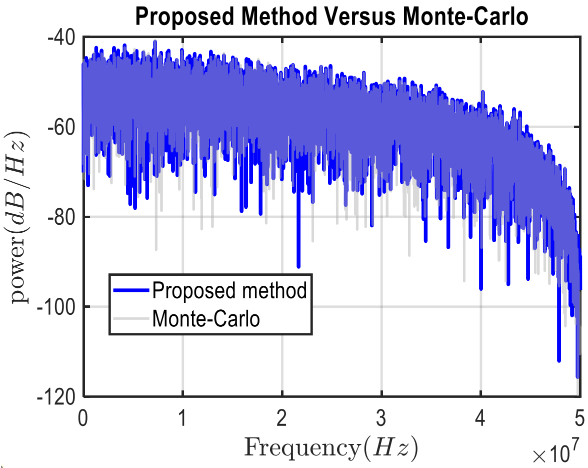


Fig. 4.25 Output noise response of an inverter.

CHAPTER 5

5.1 Summary

In this dissertation, multiple electronic design automation algorithms are proposed and developed in the XYCE platform for a fast and accurate simulation result. The step response solver does not need to climb an unstable steep ramp to approximate the real step response. The following static driven method with adaptively controlled step size naturally differentiates the state variables of the system into fast/slow group (quiescent/dynamic group). By managing the second order time derivative of the quiescent group, the dynamic group is leading the excursion of the quiescent group. A static driven based pseudo transient DC convergence is applied to difficult convergence examples and shows substantial speed up against the conventional pseudo transient method.

Time domain adjoint sensitivity is visited with a circuit theoretic approach. By applying the proposed time domain adjoint sensitivity estimation, all the sensitivities of node voltages with respect to all the potential parameters are available in one simulation run. When a numerical integration method is applied, the accuracy of the sensitivity is discussed with and without taking the high order term into account. The correction technique ensures the utmost accuracy. For verification purposes, the abovementioned time domain sensitivity is applied to simulate the time domain noise response after the noise model is equivalently transferred to be the variation of circuit parameters. The performance of the proposed method against Monte-Carlo based time domain noise response shows an excellent match.

APPENDIX

BIBLIOGRAPHY

[1] Keiter, Eric, Russo, Thomas, Schiek, Richard, Thornquist, Heidi, Mei, Ting, Verley, Jason, Sholander, Peter, Aadithya, Karthik, and Schickling, Joshua. *Xyce Parallel Electronic Simulator Reference Guide, Version 7.4.*. United States: N. p., 2021. Web. doi:10.2172/1826862.

[2] R. A. Rutenbar, G. G. E. Gielen and J. Roychowdhury, "Hierarchical Modeling, Optimization, and Synthesis for System-Level Analog and RF Designs," in *Proceedings of the IEEE*, vol. 95, no. 3, pp. 640-669, March 2007.

[3] K. Takeuchi, "Novel Co-Design of NAND Flash Memory and NAND Flash Controller Circuits for Sub-30 nm Low-Power High-Speed Solid-State Drives (SSD)," in *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1227-1234, April 2009.

[4] A. Datta, A. Goel, R. T. Cakici, H. Mahmoodi, D. Lekshmanan and K. Roy, "Modeling and Circuit Synthesis for Independently Controlled Double Gate FinFET Devices," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 11, pp. 1957-1966, Nov. 2007.

[5] L. Colalongo, S. Comensoli, A. Richelli and Z. M. K. Vajna, "A Nonlinearized Lumped-Charge Model for Power Semiconductor Devices," in *IEEE Transactions on Electron Devices*, vol. 68, no. 8, pp. 3974-3981, Aug. 2021.

[6] R. M. Kotecha, M. M. Hossain, A. U. Rashid, A. I. Emon, Y. Zhang and H. A. Mantooth, "Compact Modeling of High-Voltage Gallium Nitride Power Semiconductor Devices for Advanced Power Electronics Design," in *IEEE Open Journal of Power Electronics*, vol. 2, pp. 75-87, 2021.

[7] D. D. Mahajan, S. A. Albahrani, R. Sodhi, T. Eguchi and S. Khandelwal, "Physics-Oriented Device Model for Packaged GaN Devices," in *IEEE Transactions on Power Electronics*, vol. 35, no. 6, pp. 6332-6339, June 2020.

[8] Q. Xie, C. -J. Lee, J. Xu, C. Wann, J. Y. . -C. Sun and Y. Taur, "Comprehensive Analysis of Short-Channel Effects in Ultrathin SOI MOSFETs," in *IEEE Transactions on Electron Devices*, vol. 60, no. 6, pp. 1814-1819, June 2013.

[9] S. Mittal, Amita, A. S. Shekhawat, S. Ganguly and U. Ganguly, "Analytical Model to Estimate FinFET's I_{ON} , I_{OFF} , SS, and V_T Distribution Due to FER," in *IEEE Transactions on Electron Devices*, vol. 64, no. 8, pp. 3489-3493, Aug. 2017.

[10] K. Lee, T. An, S. Joo, K. Kwon and S. Kim, "Modeling of Parasitic Fringing Capacitance in Multifin Trigate FinFETs," in *IEEE Transactions on Electron Devices*, vol. 60, no. 5, pp. 1786-1789, May 2013, doi: 10.1109/TED.2013.

[11] P. Manfredi, D. Vande Ginste, D. De Zutter and F. G. Canavero, "Generalized Decoupled Polynomial Chaos for Nonlinear Circuits With Many Random Parameters," in *IEEE Microwave and Wireless Components Letters*, vol. 25, no. 8, pp. 505-507, Aug. 2015.

[12] Xiaojun Li, Jin Qin, Bing Huang, Xiaohu Zhang and J. B. Bernstein, "A new SPICE reliability simulation method for deep submicrometer CMOS VLSI circuits," in *IEEE Transactions on Device and Materials Reliability*, vol. 6, no. 2, pp. 247-257, June 2006.

[13] B. R. Epstein, M. Czigler and S. R. Miller, "Fault detection and classification in linear integrated circuits: an application of discrimination analysis and hypothesis testing," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 1, pp. 102-113, Jan. 1993.

[14] C. Chen, H. Wang, X. Song, F. Liang, K. Wu and T. Tao, "High Dimensional Bayesian Optimization for Analog Integrated Circuit Sizing Based on Dropout and gm/ID Methodology," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, doi: 10.1109/TCAD.2022.

[15] X. Li, "Post-silicon performance modeling and tuning of analog/mixed-signal circuits via Bayesian Model Fusion," *2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2012.

[16] J. M. Willenbring, M. A. Heroux and R. T. Heaphy, "The Trilinos Software Lifecycle Model," *Third International Workshop on Software Engineering for High Performance Computing Applications (SE-HPC '07)*, 2007.

[17] J. Li and R. Rohrer, "Efficient Static-driven Integration for Step-function Transient Simulation," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* doi: 10.1109/TCAD.2021.

[18] D. Ahsanullah, Z. Gao, J. Li and R. Rohrer, "Circuit Theory of Time Domain Adjoint Sensitivity," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*(submitted).

[19] L. W. Nagel, SPICE2: A Computer Program to Simulate Semiconductor Circuits, ERL Memo No. ERLM520, Electronics Research Laboratory, University of California, Berkeley, May 1975.

[20] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 4, pp. 352-366, April 1990.

[21] Eric R. Keiter, Ting Mei, Thomas V. Russo, Richard L. Schiek, Peter E. Sholander, Heidi K. Thornquist, Jason C. Verley, and David G. Baur. Xyce Parallel Electronic Simulator: User’s Guide, Version 6.4. Technical Report SAND2015-10527, Sandia National Laboratories, Albuquerque, NM, 2015.

[22] H. Shichman and D. A. Hodges, "Modeling and simulation of insulated-gate field-effect transistor switching circuits," in *IEEE Journal of Solid-State Circuits*, vol. 3, no. 3, pp. 285-289, Sept. 1968

[23] M. Kakizaki and T. Sugawara, "A Modified Newton Method for the Steady-State Analysis," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 4, no. 4, pp. 662-667, October 1985.

[24] O. Nastov, R. Telichevesky, K. Kundert and J. White, "Fundamentals of Fast Simulation Algorithms for RF Circuits," in *Proceedings of the IEEE*, vol. 95, no. 3, pp. 600-621, March 2007.

[25] J. Roychowdhury and R. Melville, "Delivering global DC convergence for large mixed-signal circuits via homotopy/continuation methods," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 1, pp. 66-78, Jan. 2006.

[26] J. S. Roychowdhury and R. C. Melville, "Homotopy techniques for obtaining a DC solution of large-scale MOS circuits," *33rd Design Automation Conference Proceedings, 1996*, 1996, pp. 286-291.

[27] L. W. Nagel, SPICE2: A Computer Program to Simulate Semiconductor Circuits, ERL Memo No. ERLM520, Electronics Research Laboratory, University of California, Berkeley, May 1975.

[28] J. Roychowdhury, "Efficient Methods For Simulating Highly Nonlinear Multi-rate Circuits," *Proceedings of the 34th Design Automation Conference*, 1997.

[29] A. R. Newton, "Techniques for the simulation of large-scale integrated circuits," in *IEEE Transactions on Circuits and Systems*, vol. 26, no. 9, pp. 741-749, September 1979.

[30] R. E. Bryant, "A Switch-Level Model and Simulator for MOS Digital Systems," in *IEEE Transactions on Computers*, vol. C-33, no. 2, pp. 160-177, Feb. 1984.

[31] T. Wang, K. Aadithya, B. Wu and J. Roychowdhury, "MAPP: A platform for prototyping algorithms and models quickly and easily," *2015 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, 2015.

[32] A. Devgan and R. A. Rohrer, "Event driven adaptively controlled explicit simulation of integrated circuits," *Proceedings of 1993 International Conference on Computer Aided Design (ICCAD)*, 1993.

[33] E. S. Kuh and R. A. Rohrer, "The state-variable approach to network analysis," in *Proceedings of the IEEE*, vol. 53, no. 7, pp. 672-686, July 1965.

[34] X. Wang and H. -D. Chiang, "Application of pseudo-transient continuation method in dynamic stability analysis," *2014 IEEE PES General Meeting | Conference & Exposition*, 2014.

[35] L. H. Wang, V. M. Yarushina, Y. Alkhimenkov and Y. Podladchikov, "Physics-inspired pseudo-transient method and its application in modelling focused fluid flow with geological complexity," in *Geophysical Journal International*, vol. 229, no. 1, pp. 1-20, Oct. 2021.

[36] J. Li and R. Rohrer, "Efficient Static-Driven Integration for Step-Function Transient Simulation," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 7, pp. 2213-2222, July 2022.

[37] M. Bakr, *Nonlinear Optimization in Electrical Engineering with Applications in MATLAB*. Institution of Engineering and Technology, 2013.

[38] S. Director and R. Rohrer, “The generalized adjoint network and network sensitivities,” *IEEE Transactions on Circuit Theory*, vol. 16, no. 3, pp. 318–323, 1969.

[39] R. Rohrer, L. Nagel, R. Meyer and L. Weber, “Computationally efficient electronic-circuit noise calculations,” in *IEEE Journal of Solid-State Circuits*, vol. 6, no. 4, pp. 204-213, 1971.

[40] B. Tasic, J. J. Dohmen, E. J. W. terMaten, T. G. Beelen, W. H. Schilders, A. de Vries, and M. van Beurden, “Robust dc and efficient time-domain fast fault simulation,” *COMPEL International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 2014.

[41] G. Fedi, S. Manetti, M. C. Piccirilli and J. Starzyk, “Determination of an optimum set of testable elements in the fault diagnosis of analog linear circuits,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 46, no. 7, pp. 779-787, 1999.

[42] Z. Gao, R. Rohrer, “Efficient non-Monte-Carlo yield estimation,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2021*

[43] I. Seghaier, M. H. Zaki and S. Tahar, "Mating sensitivity analysis and statistical verification for efficient yield estimation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 2, pp. 294-307, 2020.

[44] B. Tellegen, “A general network theorem, with applications,” *Philips Res Rep,* vol. 7, pp. 256–269, 1952.

[45] K. Kundert, *The Designer’s Guide to SPICE and SPECTRE®*. Springer Science & Business Media, 2006.

[46] M. Jeffery, P. Y. Xie, S. R. Whiteley and T. Van Duzer, "Monte Carlo and thermal noise analysis of ultra-high-speed high temperature superconductor digital circuits," in *IEEE Transactions on Applied Superconductivity*, vol. 9, no. 2, pp. 4095-4098, June 1999.

[47] R. H. Redus and A. C. Huber, "Monte Carlo time domain noise simulation in nuclear electronics," *2008 IEEE Nuclear Science Symposium Conference Record*, 2008, pp. 3421-3429.

[48] A. Mérigaud and J. V. Ringwood, "Free-Surface Time-Series Generation for Wave Energy Applications," in *IEEE Journal of Oceanic Engineering*, vol. 43, no. 1, pp. 19-35, Jan. 2018.

[49] S. M. Sze, “Physics of Semiconductor Devices”, WileyInterscience, John Wiley & Sons, 1969.

[50] Joon-Jea Sung, Guen-Soon Kang, Suki Kim, “A transient noise model for frequency-dependent noise sources”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, V.22, No.8, 2003, p.1097.

[51] C. E. Calosso and E. Rubiola, "Phase noise and jitter in digital electronics," *2014 European Frequency and Time Forum (EFTF)*, 2014, pp. 374-376.