MASTER

Transient sensitivity analysis in circuit simulation

Xu, H.

Award date:
2007

TECHNISCHE UNIVERSITEIT EINDHOVEN

Department of Mathematics and Computing Science

MASTER'S THESIS

Transient Sensitivity Analysis in Circuit Simulation

by
Hong Xu

Supervisor: Dr. Jan ter Maten

Eindhoven, November-1-2004

Acknowledgement

I would like to thank many people for their helps during my two-year master studying. With their helps, I got accustomed to the new language surrounding and the new culture of the Netherlands.

At first, I would thank my supervisor: Dr. Jan Ter Maten, for his guidance and support in my project. He gave me a lot of related literatures to help me understand this project well. His valuable suggestions and comments helped me to go through over the whole project.

I would say thanks to Prof. Mattheij for giving me the chance to study at TU/e. And I also thank him for his encouragement on my spoken English skill, which helped me to follow the courses quickly at the beginning.

I would like to thank Dr. Martijn Anthonissen, he was our student director, and gave me a lot of helps on my studying life. Dr. Paul de Haas was our numerical programming teacher, he taught us Latex, Matlab etc, which were important tools used in my thesis. And also I would like thank several teachers for their great teaching: Prof. Wil Schilders, he let me know about the Newton method in his lecture; Dr. Jos maubach and Dr. Thije Boonkkamp, they let me know about the knowledge of scientific computing; Dr. Fons Van de Ven, he was always very kind and patient in his teaching. At final, I would like to say thanks to Enna van Dijk, she was always friendly and willing to help me whenever I met the problems.

Here I would thank all my Chinese friends in Eindhoven for they giving my family a lot of helps. And also I would thank my Vietnamese classmate, Kim-Hoa Tran, for our nice corporation in the studying.

My final thanks would be given to my husband, my parents and my sister, for their encouragement and support all the time.

Contents

List of Figures

Notations and Conventions

s	Source
v	Voltage
i	Current
R	Resistance of Resistor
C	Capacitance of Capacitor
L	Inductance of Inductor
t	Time
T	End-time of Transient Analysis

nx	Number of Unknowns
nf	Number of Functions
np	Number of Parameters
DC	Direct Current
AC	Alternating Current
DAE	Differential Algebraic Equation
KCL	Kirchhoff's Current Law
KVL	Kirchhoff's Voltage Law
CDF	Centrical Differential Formula
FDF	Forward Differential Formula
BDF	Backward Differential Formula
NR	Newton-Raphson
HB	Harmonic Balance
SMW	Shermann-Morrison-Woodbury

Chapter 1

Introduction

This thesis introduces some approaches for the sensitivity analysis, especially in the circuit simulation. Sensitivity analysis is of great use for circuit, providing important clues while optimizing the circuit. If we are interested in the influences of some parameters on circuit elements, the circuit can be adjusted to the best according to the sensitivity curves.

Chapter 2 is to present how to work on the circuit simulation. Normally, we have four analyses: DC analysis, AC analysis, HB analysis, and transient analysis. Except for DC analysis, which is only for the DC condition, the left three analyses are all for AC condition.

Based on the simulation of Chapter 2, Chapter 3 aims to solve the related sensitivity problems. Typically, we have two optimization expressions: $\mathbf{F}$ and $\mathbf{f}$, which imply the time integral observation function and the simple observation function respectively. We have five approaches to perform sensitivity analysis. The basic two are: direct method and adjoint method. Direct method is used for the condition of many observation functions and one parameter. Adjoint method is used for the condition of many observation functions and many parameters. Apparently, adjoint method works on a wider range, but it is always based on the direct method. So to skip the efforts on direct method, two more methods are introduced for the condition of many observation functions and many parameters: integral method and extension integral method. But these two methods are only for the transient analysis.

Chapter 2

Fundamental Theory of Circuit Simulation

This chapter introduces the fundamental theory of circuit simulation. In section 2.1, the mathematical expression of circuit modelling is shown. Section 2.2 to section 2.5 discuss the three typical analyses for circuit equation.

2.1 Circuit Equation

2.1.1 Circuit Elements

We begin with introducing circuit elements, which are considered as ideal elements in this project. There are four main types of elements in the circuit: $Source$, $Resistor$, $Capacitor$, and $Inductor$.

Source

Sources are divided into two kinds: $Voltage\ Sources$ and $Current\ Sources$. Also each of these two kinds is divided into two parts: $Independent$ and $Dependent$.

A voltage source is a circuit element that will maintain a prescribed voltage across through its terminals regardless the current in it. An independent voltage source means that such a prescribed voltage is independent of the voltages or currents existing in other places of the circuit. But an independent voltage source may depend on its own currents, even in a nonlinear way. On the other side, a dependent voltage source depends on other voltages or currents in the circuit.

Contrary to a voltage source, a current source has a prescribed current regardless the voltage. An independent current source does not depend on other voltages or currents in the circuit. Inversely, dependent current source depends on it.

Resistor

Resistors are divided into two kinds: $Linear\ Resistors$ and $Nonlinear\ Resistors$. Following Ohm's law, a linear resistor is expressed as:

$$v = iR,$$

For a nonlinear resistor, the relation of the voltage and the current is expressed as a nonlinear function:

$$v = f(i), \quad \text{``}v - \textit{defined}\text{''}$$
$$i = g(v), \quad \text{``}i - \textit{defined}\text{''}$$

in which cases $R = \frac{\partial f}{\partial i}$, or $R = \frac{\partial v}{\partial g}$ respectively.

Capacitor

Capacitors are divided into two kinds: $Linear\ Capacitors$ and $Nonlinear Capacitors$. A linear capacitor has the mathematical expression:

$$i = C\frac{dv}{dt},$$

For a nonlinear capacitor, the current is the rate at which a nonlinear change function Q of the voltage across capacitor varies with time:

$$i = \frac{dQ(v)}{dt}.$$

Inductor

Inductors are divided into two kinds: $Linear\ Inductors$ and $Nonliear\ Inductors$. A linear inductor has the mathematical expression:

$$v = L\frac{di}{dt},$$

A nonlinear inductor is defined by the rate at which a nonlinear flux function Φ of the current through the inductor varies with time:

$$v = \frac{d\Phi(i)}{dt}.$$

2.1.2 Nodal Analysis

Two laws should be introduced as the fundament of nodal analysis:

$Kirchhoff's\ Current\ Law$(KCL): The algebraic sum of all the currents at any node in a circuit equals zero.

$Kirchhoff's\ Voltage\ Law$(KVL): The algebraic sum of all the voltages around any closed path in a circuit equals zero.

Nodal Analysis is a simple and efficient approach to construct circuit equation. One important step is to find the essential nodes in circuit. After applying KCL to each node, the circuit topology can be described as a matrix A which is named as $Adjacency\ Matrix$. Adjacency matrix is defined according to current direction:

$$A(i,j) := \begin{cases} 1 & \text{if node } i \text{ is the “from" node of branch } j \\ -1 & \text{if node } i \text{ is the “to" node of branch } j \\ 0 & \text{if node } i \text{ is not connected to branch } j \end{cases}$$

Therefore, KCL for the circuit equation is expressed as:

$$A\mathbf{i}_b = \mathbf{0}, \tag{2.1}$$

where $\mathbf{i}_b$ is the branch-current vector. Applying KVL to circuit, we have the relation between node-voltages and branch-voltages as:

$$A^T\mathbf{v}_n = \mathbf{v}_b. \tag{2.2}$$

where $\mathbf{v}_n$ is the node-voltage vector, and $\mathbf{v}_b$ is the branch-voltage vector.
Substituting the existing mathematical expression of elements shown in Section 2.1.1 into Eq. (2.1), and next replaces the branch-voltage by the node-voltage, we obtain the main circuit equation which is expressed as:

$$\mathbf{j}(\mathbf{x}(\mathbf{t})) + \frac{d\mathbf{q}(\mathbf{x}(\mathbf{t}))}{dt} = \mathbf{s}(t). \tag{2.3}$$

In this case $\mathbf{x} = \mathbf{v}_n$, Eq. (2.3) can also be extended to cover equations for current.

For multi-dimensional system of equation in which the term $\mathbf{j}$ and $\mathbf{q}$ are vector functions and $\mathbf{s}$ is the source vector of these equations, in which some are differential equations, some are algebraic equations, so the Eq. (2.3) is called a system of *Differential*
$Algebraic\ Equation$(DAE).

The terms $\mathbf{j}$ and $\mathbf{q}$ of DAE are functions of unknown $\mathbf{x}$, which may depends on the time t. So the current equation implicitly depends on time t.

2.1.3 Examples

Here a simple example is used to illustrate how to get the system of circuit equations. The circuit has four essential nodes: n_0, n_1, n_2, and n_3, and also has five branches(each with a component): a voltage source $s(t)$, linear resistors r_1(with resistance R_1) and r_2(with resistance R_2), a linear capacitor C_1(with capacitance C), and a linear inductor L_1(with inductance L).

Its topology is like below, shown in Fig. (2.1):

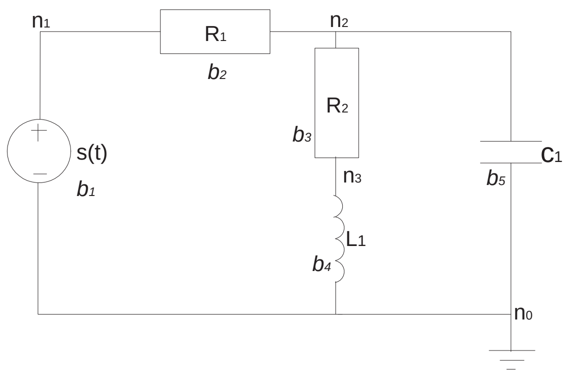


Figure 2.1: The topology of the example

Step 1:

According to nodal analysis, we have the adjacency matrix as:

$$A = \begin{array}{c} \\ n_0 \\ n_1 \\ n_2 \\ n_3 \end{array} \begin{array}{ccccc} b_1 & b_2 & b_3 & b_4 & b_5 \\ 1 & 0 & 0 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 \end{array}$$

Applying Eq. (2.1), we have:

$$\begin{pmatrix} 1 & 0 & 0 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 \end{pmatrix} \begin{pmatrix} i_{b1} \\ i_{b2} \\ i_{b3} \\ i_{b4} \\ i_{b5} \end{pmatrix} = \mathbf{0} \Rightarrow$$

$$\begin{aligned} i_{b1} - i_{b4} - i_{b5} = 0, \quad -i_{b1} + i_{b2} = 0 \\ -i_{b2} + i_{b3} + i_{b5} = 0, \quad -i_{b3} + i_{b4} = 0 \end{aligned} \tag{2.4}$$

The current i_{b2}, i_{b3}, and i_{b5} are the functions of the nodal voltages $v_0,\ v_1,\ v_2,\ v_3$ or the time derivatives of them. The currents i_{b1} and i_{b4} are special for they belongs to v-defined elements, so they should be included in the list of unknowns.

Step 2:

Using the mathematical expressions for elements shown in Section 2.1.1, we have:

$$\begin{aligned} i_{b2} = \frac{v_{b2}}{R_1}, \quad i_{b3} = \tfrac{v_{b3}}{R_2}, \quad i_{b5} = C\frac{dv_{b5}}{dt} \\ v_{b1} = -s(t), \quad v_{b4} = L\tfrac{di_{b4}}{dt} \end{aligned} \tag{2.5}$$

Applying the Eq. (2.2), we can express the branch-voltages in term of the node-voltages:

$$\begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \\ -1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} v_{b1} \\ v_{b2} \\ v_{b3} \\ v_{b4} \\ v_{b5} \end{pmatrix} \Rightarrow$$

$$\begin{aligned} v_{b1} = v_0 - v_1, \qquad v_{b2} = v_1 - v_2, \qquad v_{b3} = v_2 - v_3, \\ v_{b4} = -v_0 + v_3, \quad v_{b5} = -v_0 + v_2 = v_2. \end{aligned} \tag{2.6}$$

Substituting Eq. (2.6) into Eq. (2.5), we have a new equation:

$$\begin{aligned} i_{b2} = \frac{v_1 - v_2}{R_1}, \qquad i_{b3} = \tfrac{v_2 - v_3}{R_2}, \qquad i_{b5} = C\frac{dv_2}{dt}, \\ v_0 - v_1 = -s(t), \quad -v_0 + v_3 = L\tfrac{di_{b4}}{dt}. \end{aligned} \tag{2.7}$$

Step 3:

Combining Eq. (2.7) with Eq. (2.4), we have the circuit equation:

$$\underbrace{\begin{pmatrix} i_{b1} - i_{b4} \\ -i_{b1} + \frac{v_1 - v_2}{R_1} \\ -\frac{v_1 - v_2}{R_1} + \frac{v_2 - v_3}{R_2} \\ -\frac{v_2 - v_3}{R_2} + i_{b4} \\ v_0 - v_1 \\ v_3 \end{pmatrix}}_{\mathbf{j}(\mathbf{x})} + \frac{d}{dt}\underbrace{\begin{pmatrix} -C(v_2 - v_0) \\ 0 \\ C(v_2 - v_0) \\ 0 \\ 0 \\ -Li_{b4} \end{pmatrix}}_{\mathbf{q}(\mathbf{x})} = \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -s(t) \\ 0 \end{pmatrix}}_{\mathbf{s}(t)},$$

where the unknowns are $[v_1\ v_2\ v_3\ i_{b1}\ i_{b4}]^T$. Since the voltage of node 0 (which is the reference node) is always set as zero, normally we ignore the equation which detects the currents relation for node 0. So in this example, the equation ignored is: $i_{b1} - i_{b4} - i_{b5} = 0$. The final equation is:

$$\underbrace{\begin{pmatrix} -i_{b1} + \frac{v_1 - v_2}{R_1} \\ -\frac{v_1-v_2}{R_1} + \frac{v_2-v_3}{R_2} \\ -\frac{v_2-v_3}{R_2} + i_{b4} \\ v_1 \\ v_3 \end{pmatrix}}_{\mathbf{j}(\mathbf{x})} + \frac{d}{dt}\underbrace{\begin{pmatrix} 0 \\ Cv_2 \\ 0 \\ 0 \\ -Li_{b4} \end{pmatrix}}_{\mathbf{q}(\mathbf{x})} = \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ s(t) \\ 0 \end{pmatrix}}_{\mathbf{s}(t)},$$

Clearly the system is a DAE: its index=1.

2.2 DC analysis

If the source is invariant, we call this situation DC condition. DC condition plays a quite important role in the circuit equation. It is the base for the further analysis both AC analysis and transient analysis. In DC analysis, each capacitor is skipped in reason that it "opens" (that means the current across through the capacitor goes to zero ideally), and each inductor is replaced by a "short" (that means the voltage of the two terminals of inductor goes to zero ideally). So the term $\mathbf{q}(\mathbf{x}, t)$ in the circuit equation becomes zero, the circuit equation for DC condition is simplified as: $\mathbf{j}(\mathbf{x}(t)) = \mathbf{s}(\mathbf{t})$, also written as

$$\mathbf{j}(\mathbf{x}(0)) = \mathbf{s}_0 \Rightarrow \tilde{\mathbf{j}} = \mathbf{0}. \tag{2.8}$$

Where $\mathbf{s}_0$ is DC source, which is a constant vector. And we define $\tilde{\mathbf{j}} = \mathbf{j}(\mathbf{x}(0)) - \mathbf{s}_0$. Since in DC condition, the time is always considered as a simple constant: $t = 0$. The DC solution is the start point for AC analysis and set as the initial condition for transient simulation.

2.2.1 Newton-Raphson(NR) Method

NR approach is adopted to solve the Eq. (2.8).

We write Eq. (2.8) as $h(x) = 0$, and assume it has a unique exact solution x_*. Also this equation can be approximated by numerical method, the approximate solution is assumed as x_a for expected tolerance.

Using Taylor expansion, we have:

$$\begin{aligned} 0 = h(x_*) &= h(x_a) + h'(x_a)(x_* - x_a) + O(||x_* - x_a||^2) \\ \Rightarrow x_* &= x_a - \left[\frac{dh(x_a)}{dx}\right]^{-1} h(x_a) + O(||x_* - x_a||^2), \qquad (2.9) \\ \Rightarrow x_* &\simeq x_a - \left[\frac{dh(x_a)}{dx}\right]^{-1} h(x_a). \qquad (2.10) \end{aligned}$$

The error is proportional to $O(||x_* - x_a||^2)$ when $\left[\frac{dh(x_a)}{dx}\right]$ is non-singular. We assume that x_a is close to x_*. If we give an initial value as $x^{(1)}$, we will have:

$$x^{(2)} = x^{(1)} - \left[\frac{df(x^{(1)})}{dx}\right]^{-1} f(x^{(1)}), \tag{2.11}$$

Therefore if $x^{(1)}$ is not close enough to x_*, we may proceed as given:

$$x^{(r+1)} = x^{(r)} - \left[\frac{df(x^{(r)})}{dx}\right]^{-1} f(x^{(r)}). \tag{2.12}$$

Where r is a random positive integer.

The procedure of NR method is shown below:

$Input\ an\ initial\ value\ x^{(1)},\ an\ iteration\ number\ r\ (r\ is\ reasonably\ large),\ and\ a\ tolerance$
$Using\ NR\ method\ to\ get\ x^{(2)}:$
$If\ ||x^{(2)} - x^{(1)}|| < tolerance,$
$Output\ x_a = x^{(2)},$
$Else\ choose\ x^{(2)}\ as\ the\ initial\ value,$
$Using\ NR\ method\ to\ get\ x^{(3)}:$
$If\ ||x^{(3)} - x^{(2)}|| < tolerance,$
$Output\ x_a = x^{(3)},$
$Else\ choose\ x^{(3)}\ as\ the\ initial\ value,$
$\vdots$
$While\ step\ \ r,$
$If\ ||x^{(r)} - x^{(r-1)}|| < tolerance,$
$Output\ x_a = x^{(r)},$
$Else\ NR\ method\ is\ not\ convergent.$
$Output\ error\ message.$
$End.$

This procedure also can be well illustrated by the below figure:

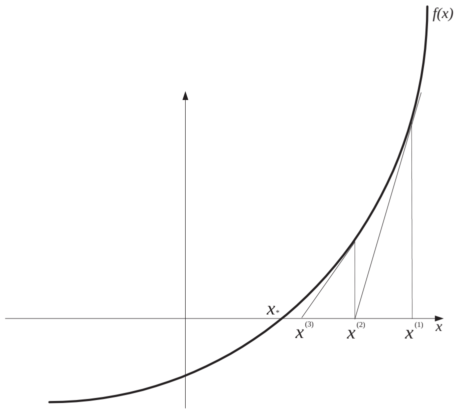


Figure 2.2: The illustration of NR method

From Fig. (2.2), the approximate solution is found to be closer and closer to the exact solution step by step. The iterations depend on the choice of initial value. From this figure, we can see that the closer the initial value is chosen to x_* , the faster the iteration speed.

2.2.2 Convergence of NR method

The solution of NR method is considered as an efficient approximation only on the assumption that the NR method is convergent. If NR method is not convergent, the solution will be divergent with terrible error.

How to ensure the convergence of NR method? Here we recall Eq. (2.9) and Eq. (2.12) respectively, writing into another expression:

$$f(x_a) = -\left[\frac{df(x_a)}{dx}\right](x_* - x_a) - O(||x_* - x_a||^2), \tag{2.13}$$

$$f(x^{(r)}) = -\left[\frac{df(x^{(r)})}{dx}\right](x^{(r+1)} - x^{(r)}). \tag{2.14}$$

We assume that the approximate solution $x^{(r+1)}$ is the numerical solution at iteration step $r+1$. We want to know about the difference between x_* and $x^{(r+1)}$, by way of x_a being replaced with $x^{(r)}$, the above Eqs.(2.13) and (2.14) turns into:

$$
\begin{aligned}
&-\left[\frac{df(x^{(r)})}{dx}\right](x_*-x^{(r)})-O(||x_*-x^{(r)}||^2)=-\left[\frac{df(x^{(r)})}{dx}\right](x^{(r+1)}-x^{(r)}),\\
\Rightarrow\quad & x^{(r+1)}-x_*=\left[\frac{df(x^{(r)})}{dx}\right]^{-1}O((x_*-x^{(r)})^2),\\
\Rightarrow\quad & ||x^{(r+1)}-x_*||=\left[\frac{df(x^{(r)})}{dx}\right]^{-1}\left|\left|O((x_*-x^{(r)})^2)\right|\right|.
\end{aligned}
$$

From the above equation, We can get below conclusions:

- For 1-dimensional problem, $\left[\frac{df(x^{(r)})}{dx}\right]$ shouldn't be zero; for multi-dimensional problem, the determinant of matrix $\left[\frac{df(x^{(r)})}{d\mathbf{x}}\right]$ shouldn't be zero.
- $\left[\frac{df(x^{(r)})}{dx}\right]^{-1}$ should be bounded to κ, we have:

  $$||x^{(r+1)}-x_*||\leq\kappa\left|\left|x_*-x^{(r)}\right|\right|^2$$

  which implies: the convergence of the $||x^{(r+1)}-x_*||$ is at least quadratic.

Also we can explain its convergence in another way. If we assume the error of $x^{(r)}$ is e_r, and the error of $x^{(r+1)}$ is e_{r+1}, they can be written as:

$$e_r=x^{(r)}-x_*,\ \ e_{r+1}=x^{(r+1)}-x_*.$$

From NR method, we have:

$$
\begin{aligned}
& x^{(r+1)}=x^{(r)}-\left[\frac{df(x^{(r)})}{dx}\right]^{-1}f(x^{(r)}),\\
\Rightarrow\quad & e_{r+1}=e_r-\left[\frac{df(x^{(r)})}{dx}\right]^{-1}f(x^{(r)}),\\
\Rightarrow\quad & e_{r+1}=\left[\frac{df(x^{(r)})}{dx}\right]^{-1}\left(\left[\frac{df(x^{(r)})}{dx}\right]e_r-f(x^{(r)})\right).
\end{aligned}
\tag{2.15}
$$

Since we have:

$$f(x_*)=f(x^{(r)})-\left[\frac{df(x^{(r)})}{dx}\right](x^{(r)}-x_*)+O((x^{(r)}-x_*)^2)\Rightarrow f(x^{(r)})=\left[\frac{df(x^{(r)})}{dx}\right]e_r-O(e_r^2).$$

After substituting the above equation into Eq. (2.15), we have:

$$e_{r+1}=\left[\frac{df(x^{(r)})}{dx}\right]^{-1}O(e_r^2)\Rightarrow||e_{r+1}||=\left[\frac{df(x^{(r)})}{dx}\right]^{-1}||e_r||^2\leq\kappa||e_r||^2.\tag{2.16}$$

which is also coincide with the conclusion what we discueesed, that means, the error of the solution will be convergent on the condition of $\left[\frac{df(x^{(r)})}{dx}\right]^{-1}$ is bounded.

2.2.3 NR method in DC analysis

We assume that the number of unknowns is: nx, so for DC analysis this is a nx-dimensional problem. The unknowns are written as a vector:

$$\mathbf{x} = [x_1, x_2, \cdots, x_{nx}]^T.$$

According to Eq. (2.11), we have:

$$\begin{pmatrix} x_1^{(r)} \\ x_2^{(r)} \\ \vdots \\ x_{nx}^{(r)} \end{pmatrix} = \begin{pmatrix} x_1^{(r-1)} \\ x_2^{(r-1)} \\ \vdots \\ x_{nx}^{(r-1)} \end{pmatrix} - \begin{pmatrix} \frac{\partial \tilde{j}_1(\mathbf{x}^{(r-1)})}{\partial x_1}, & \cdots, & \frac{\partial \tilde{j}_1(\mathbf{x}^{(r-1)})}{\partial x_{nx}} \\ \frac{\partial \tilde{j}_2(\mathbf{x}^{(r-1)})}{\partial x_1}, & \cdots, & \frac{\partial \tilde{j}_2(\mathbf{x}^{(r-1)})}{\partial x_{nx}} \\ \vdots & & \vdots \\ \frac{\partial \tilde{j}_{nx}(\mathbf{x}^{(r-1)})}{\partial x_1}, & \cdots, & \frac{\partial \tilde{j}_{nx}(\mathbf{x}^{(r-1)})}{\partial x_{nx}} \end{pmatrix}^{-1} \begin{pmatrix} \tilde{j}_1(\mathbf{x}^{(r-1)}) \\ \tilde{j}_2(\mathbf{x}^{(r-1)}) \\ \vdots \\ \tilde{j}_{nx}(\mathbf{x}^{(r-1)}) \end{pmatrix}, \quad (2.17)$$

This equation can be simplified as

$$\mathbf{x}^{(r)} = \mathbf{x}^{(r-1)} - \left[\frac{d\tilde{\mathbf{j}}(\mathbf{x}^{(r-1)})}{d\mathbf{x}}\right]^{-1} \tilde{\mathbf{j}}(\mathbf{x}^{(r-1)}). \quad (2.18)$$

Where the determinant of matrix $\left[\frac{d\tilde{\mathbf{j}}(\mathbf{x}^{(r-1)})}{d\mathbf{x}}\right]$ should not be zero.

2.2.4 Examples

We use the same circuit of the Fig. (2.1). For the special property of capacitor and inductor, the DC topology will be a little bit different, shown in Fig. (2.3).

Comparing with Fig. (2.1), we know that $v_3 = 0$ and $i_{b4} = i_{b1}$ in DC condition, so we only have three unknowns $[v_1,\ v_2,\ i_{b1}]^T$.

Linear example

If all elements are linear elements, it's easy to write out the circuit equation for this topology:

$$\begin{pmatrix} -i_{b1} + \frac{v_1 - v_2}{R_1} \\ -\frac{v_1 - v_2}{R_1} + \frac{v_2}{R_2} \\ v_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ s_0 \end{pmatrix} \Rightarrow \tilde{\mathbf{j}} = \begin{pmatrix} -i_{b1} + \frac{v_1 - v_2}{R_1} \\ -\frac{v_1 - v_2}{R_1} + \frac{v_2}{R_2} \\ v_1 - s_0 \end{pmatrix}. \quad (2.19)$$

Using NR method, the iteration equation is expressed as:

$$\begin{pmatrix} v_1^{(r)} \\ v_2^{(r)} \\ i_{b1}^{(r)} \end{pmatrix} = \begin{pmatrix} v_1^{(r-1)} \\ v_2^{(r-1)} \\ i_{b1}^{(r-1)} \end{pmatrix} - \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & -1 \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_2} & 0 \\ 1 & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} -i_{b1}^{(r-1)} + \frac{v_1^{(r-1)} - v_2^{(r-1)}}{R_1} \\ -\frac{v_1^{(r-1)} - v_2^{(r-1)}}{R_1} + \frac{v_2^{(r-1)}}{R_2} \\ v_1^{(r-1)} - s_0 \end{pmatrix} \quad (2.20)$$

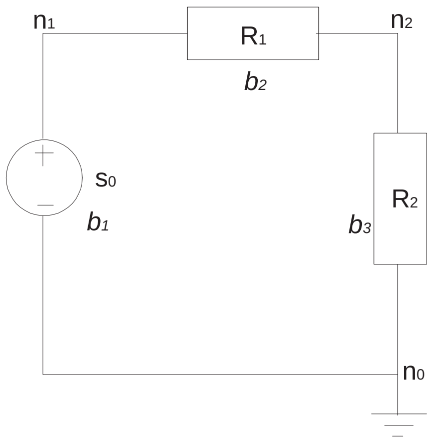


Figure 2.3: The DC topology

Nonlinear example

If there are some nonlinear elements in the circuit, the circuit equation will be a bit different. Here we assume that the R_2 is a nonlinear resistor, the relation of voltage and current is expressed as:

$$i_{b3} = v_{b3}^3 + 2v_{b3}^2 + v_{b3} = v_2^3 + 2v_2^2 + v_2.$$

So the circuit equation is:

$$\tilde{\mathbf{j}} = \begin{pmatrix} -i_{b1} + \frac{v_1 - v_2}{R_1} \\ -\frac{v_1 - v_2}{R_1} + v_2^3 + 2v_2^2 + v_2 \\ v_1 - s_0 \end{pmatrix}.$$

Using NR method, the iteration equation is expressed as:

$$\begin{pmatrix} v_1^{(r)} \\ v_2^{(r)} \\ i_{b1}^{(r)} \end{pmatrix} = \begin{pmatrix} v_1^{(r-1)} \\ v_2^{(r-1)} \\ i_{b1}^{(r-1)} \end{pmatrix} - \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & -1 \\ -\frac{1}{R_1} & \frac{1}{R_1} + 3\left(v_2^{(r-1)}\right)^2 + 4v_2^{(r-1)} + 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} -i_{b1}^{(r-1)} + \frac{v_1^{(r-1)} - v_2^{(r-1)}}{R_1} \\ -\frac{v_1^{(r-1)} - v_2^{(r-1)}}{R_1} + \left(v_2^{(r-1)}\right)^3 + 2\left(v_2^{(r-1)}\right)^2 + v_2^{(r-1)} \\ v_1^{(r-1)} - s_0 \end{pmatrix}$$

2.3 AC analysis

When performing AC analysis, we consider a small time-varying signal $\tilde{\mathbf{s}}(t)$ added to the constant source(DC condition), and write the response of the circuit(solution) as

$\mathbf{x}_0 + \tilde{\mathbf{x}}(t)$, which satisfies the DAE:

$$\mathbf{j}(\mathbf{x}_0 + \tilde{\mathbf{x}}(t)) + \frac{d\mathbf{q}(\mathbf{x}_0 + \tilde{\mathbf{x}}(t))}{dt} = \mathbf{s}_0 + \tilde{\mathbf{s}}(t),$$

Where $\mathbf{x}_0$ is the solution of DC condition.

Using Taylor expansion:

$$\begin{aligned}
\mathbf{j}(\mathbf{x}_0 + \tilde{\mathbf{x}}(t)) &= \mathbf{j}(\mathbf{x}_0) + \frac{d\mathbf{j}(\mathbf{x}_0)}{d\mathbf{x}}\tilde{\mathbf{x}}(t) + \frac{d^2\mathbf{j}(\mathbf{x}_0)}{d\mathbf{x}^2} O\left(\frac{\tilde{\mathbf{x}}^2(t)}{2}\right), \\
\mathbf{q}(\mathbf{x}_0 + \tilde{\mathbf{x}}(t)) &= \mathbf{q}(\mathbf{x}_0) + \frac{d\mathbf{q}(\mathbf{x}_0)}{d\mathbf{x}}\tilde{\mathbf{x}}(t) + \frac{d^2\mathbf{q}(\mathbf{x}_0)}{d\mathbf{x}^2} O\left(\frac{\tilde{\mathbf{x}}^2(t)}{2}\right).
\end{aligned}$$

Ignoring the 2nd and higher-order terms of the Taylor expansion, the DAE for AC analysis is rewritten as:

$$\mathbf{j}(\mathbf{x}_0) + \frac{d\mathbf{j}(\mathbf{x}_0)}{d\mathbf{x}}\tilde{\mathbf{x}}(t) + \frac{d}{dt}\left(\mathbf{q}(\mathbf{x}_0) + \frac{d\mathbf{q}(\mathbf{x}_0)}{d\mathbf{x}}\tilde{\mathbf{x}}(t)\right) = \mathbf{s}_0 + \tilde{\mathbf{s}}(t).$$

Based on DC analysis, $\mathbf{q}(\mathbf{x}_0)$ is known to be constant, and $\mathbf{j}(\mathbf{x}_0) = \mathbf{s}_0$ is known for DAE of DC condition. So this equation is simplified as:

$$\frac{d\mathbf{j}(\mathbf{x}_0)}{d\mathbf{x}}\tilde{\mathbf{x}}(t) + \frac{d\mathbf{q}(\mathbf{x}_0)}{d\mathbf{x}}\frac{d\tilde{\mathbf{x}}(t)}{dt} = \tilde{\mathbf{s}}(t).$$

If we define that:

$$\mathcal{G} = \frac{d\mathbf{j}(\mathbf{x})}{d\mathbf{x}}, \quad \mathcal{C} = \frac{d\mathbf{q}(\mathbf{x})}{d\mathbf{x}}.$$

Therefore:

$$\mathcal{G}_0 = \frac{d\mathbf{j}(\mathbf{x}_0)}{d\mathbf{x}}, \quad \mathcal{C}_0 = \frac{d\mathbf{q}(\mathbf{x}_0)}{d\mathbf{x}}.$$

The DAE for AC analysis is written as:

$$\mathcal{G}_0\tilde{\mathbf{x}}(t) + \mathcal{C}_0\frac{d\tilde{\mathbf{x}}(t)}{dt} = \tilde{\mathbf{s}}(t). \tag{2.21}$$

We assume that $\tilde{s}(t)$ can be written in a Fourier series:

$$\tilde{\mathbf{s}}(t) = \sum_{-\infty}^{+\infty} \tilde{S}_k e^{jk\omega t}, \quad \tilde{S}_k = \frac{1}{\mathcal{T}}\int_t^{t+\mathcal{T}} \tilde{\mathbf{s}}(t)e^{-jk\omega t}dt.^{1}$$

where ω is the basic frequency of the circuit.

So the solution is reasonably assumed as:

$$\tilde{\mathbf{x}}(t) = \sum_{-\infty}^{+\infty} \tilde{X}_k e^{jk\omega t}, \quad \tilde{X}_k = \frac{1}{\mathcal{T}}\int_t^{t+\mathcal{T}} \tilde{\mathbf{x}}(t)e^{-jk\omega t}dt. \tag{2.22}$$

[1] $j = \sqrt{-1}$

Now the Eq. (2.21) becomes:

$$\mathcal{G}_0 \sum_{-\infty}^{+\infty} \tilde{X}_k e^{jk\omega t} + \mathcal{C}_0 \sum_{-\infty}^{-\infty} jk\omega \tilde{X}_k e^{jk\omega t} = \sum_{-\infty}^{+\infty} \tilde{S}_k e^{jk\omega t},$$
$$\Rightarrow (\mathcal{G}_0 + jk\omega\mathcal{C}_0)\tilde{X}_k = \tilde{S}_k \Rightarrow \tilde{X}_k = (\mathcal{G}_0 + jk\omega\mathcal{C}_0)^{-1}\tilde{S}_k. \tag{2.23}$$

Clearly Eq. (2.23) requires a linear, complex system of equations. The algebraic Eq. (2.23) is cheaper to calculate than Eq. (2.21). Each $\tilde{X}_k$ is easy to be worked out using Eq. (2.23). The final solution should follow the Eq. (2.22) to be made:

$$\mathbf{x}_{AC} = \mathbf{x}_0 + \sum_{-\infty}^{+\infty} \tilde{X}_k e^{jk\omega t}. \tag{2.24}$$

Since source is always expressed as a function with sine and cosine by Fourier transformation, it is easy to deduce that the k is always appeared as some pairs of opposite numbers, like $k = \pm 1, \pm 2$ etc. This property can help us to decrease the cost of calculating. In the case we have worked out the X_k, we only need to do conjugating on it to get X_{-k} immediately, which rely on the below proof:

PROOF : From DAE of AC circuit:

$$(\mathcal{G}_0 + jk\omega\mathcal{C}_0)\tilde{X}_k = \tilde{S}_k,$$
$$\Rightarrow \overline{(\mathcal{G}_0 + jk\omega\mathcal{C}_0)\tilde{X}_k} = \overline{\tilde{S}_k},$$
$$\Rightarrow (\mathcal{G}_0 + j(-k)\omega\mathcal{C}_0)\overline{\tilde{X}_k} = \overline{\tilde{S}_k}.$$

So if $\overline{\tilde{S}_k}$ satisfy the condition that $\tilde{S}_{-k} = \overline{\tilde{S}_k}$, we have:

$$\tilde{X}_{-k} = \overline{\tilde{X}_k},$$

Which implies the equation:

$$(\mathcal{G}_0 + j(-k)\omega\mathcal{C}_0)\tilde{X}_{-k} = \tilde{S}_{-k}.$$

Q.E.D

2.3.1 Examples

Linear example

Using the same circuit of the Fig. (2.1), if the elements are all linear elements, the circuit equation is the same as well:

$$\begin{pmatrix} -i_{b1} + \frac{v_1 - v_2}{R_1} \\ -\frac{v_1 - v_2}{R_1} + \frac{v_2 - v_3}{R_2} \\ -\frac{v_2 - v_3}{R_2} + i_{b4} \\ v_1 \\ v_3 \end{pmatrix} + \frac{d}{dt}\begin{pmatrix} 0 \\ Cv_2 \\ 0 \\ 0 \\ -Li_{b4} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ s(t) \\ 0 \end{pmatrix},$$

It is easy to get $\mathcal{G}_0$ and $\mathcal{C}_0$:

$$\mathcal{G}_0 = \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & -1 & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1}+\frac{1}{R_2} & -\frac{1}{R_2} & 0 & 0 \\ 0 & -\frac{1}{R_2} & \frac{1}{R_2} & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \quad \mathcal{C}_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -L \end{pmatrix}$$

If we assume the voltage source is $\mathbf{s}(t) = \mathbf{s}_0 + \mathbf{s}_a \sin \omega t$, so we have:

$$\tilde{\mathbf{s}}(t) = \mathbf{s}_a \sin \omega t = -\frac{j\mathbf{s}_a}{2}(e^{j\omega t} - e^{-j\omega t}). \tag{2.25}$$

It is easy to determine that,

$$\tilde{S}_k = -\frac{j\mathbf{s}_a}{2}, \; \frac{j\mathbf{s}_a}{2}; \; \textit{the correponding } k = 1, \; -1.$$

Since $\overline{\tilde{S}_1} = \overline{-\frac{j\mathbf{s}_a}{2}} = \frac{j\mathbf{s}_a}{2} = \tilde{S}_{-1}$, we only need to calculate one equation following Eq. (2.23):

$$\tilde{X}_1 = (\mathcal{G}_0 + j\omega\mathcal{C}_0)^{-1}\tilde{S}_1.$$

Then we can get $\tilde{X}_{-1}$ directly by conjugating on $\tilde{X}_1$.

Nonlinear example

If the circuit has a nonlinear elements R_2, which has the relation:

$$i_{b_3} = v_{b3}^3 + 2v_{b3}^2 + v_{b3} = (v_2 - v_3)^3 + 2(v_2 - v_3)^2 + (v_2 - v_3).$$

The circuit equation is expressed as:

$$\begin{pmatrix} -i_{b1} + \frac{v_1 - v_2}{R_1} \\ -\frac{v_1 - v_2}{R_1} + (v_2 - v_3)^3 + 2(v_2 - v_3)^2 + (v_2 - v_3) \\ -(v_2 - v_3)^3 - 2(v_2 - v_3)^2 - (v_2 - v_3) + i_{b4} \\ v_1 \\ v_3 \end{pmatrix} + \frac{d}{dt}\begin{pmatrix} 0 \\ Cv_2 \\ 0 \\ 0 \\ -Li_{b4} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ s(t) \\ 0 \end{pmatrix},$$

It is easy to get $\mathcal{G}_0$ and $\mathcal{C}_0$:

$$\mathcal{G}_0 = \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & -1 & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} + (3v_2^2 - 6v_2v_3 + 3v_3^2 + 4v_2 - 4v_3 + 1) & -(3v_2^2 - 6v_2v_3 + 3v_3^2 + 4v_2 - 4v_3 + 1) & 0 & 0 \\ 0 & -(3v_2^2 - 6v_2v_3 + 3v_3^2 + 4v_2 - 4v_3 + 1) & (3v_2^2 - 6v_2v_3 + 3v_3^2 + 4v_2 - 4v_3 + 1) & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix},$$

$$\mathcal{C}_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -L \end{pmatrix}.$$

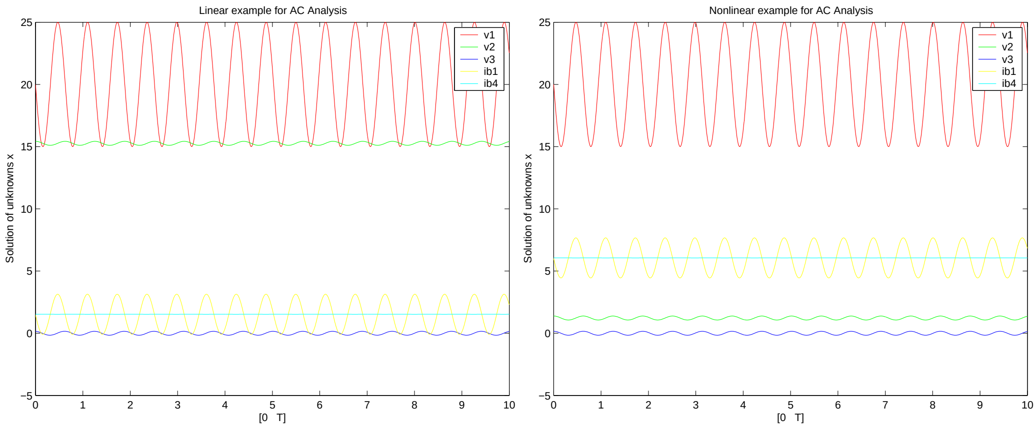


Figure 2.4: AC solution: $\mathbf{s}(t) = 20 + 5\sin 10t$

Where the unknowns $v_2,\ v_3$ is the corresponding solution of DC condition. Also we need to solve one equation:

$$\tilde{X}_1 = (\mathcal{G}_0 + j\omega\mathcal{C}_0)^{-1}\tilde{S}_1.$$

And conjugating on it to get $\tilde{X}_{-1}$.

Results

The results of Linear example and nonlinear example are illustrated in Fig. (2.4):

2.4 Harmonic Balance(HB) analysis

HB analysis is an alternative for AC condition. It no longer assumes liberalization.

The HB solution is a period function according to the period source. So its Fourier transformation is written here:

$$\mathbf{x}_{HB} = \sum_{-\infty}^{+\infty} X_k e^{jk\omega t}.$$

If we note the Fourier transforming as $\mathcal{F}$, and the inverse Fourier transforming as $\mathcal{F}^{-1}$, we have expressions as:

$$\mathbf{x}_{HB} = \mathcal{F}^{-1}(X) = \mathcal{F}^{-1}\begin{pmatrix} \vdots \\ X_k \\ \vdots \\ X_{-k} \\ \vdots \end{pmatrix}, \quad X = \mathcal{F}(\mathbf{x}_{HB}).$$

Recalling the general DAE:

$$\mathbf{j}(\mathbf{x}(t)) + \frac{d\mathbf{q}(\mathbf{x}(t))}{dt} = \mathbf{s}(t),$$

applying the Fourier transforming to this equation, which gives:

$$\begin{aligned} &\mathcal{F}\left(\mathbf{j}(\mathbf{x}(t))\right) + \mathcal{F}\left(\frac{d\mathbf{q}(\mathbf{x}(t))}{dt}\right) = \mathcal{F}\left(\mathbf{s}(t)\right), \\ \Rightarrow\ &\mathcal{F}\left(\mathbf{j}(\mathbf{x}(t))\right) + \mathcal{F}\left(\frac{d}{dt}\mathcal{F}^{-1}\mathcal{F}\left(\mathbf{q}(\mathbf{x}(t))\right)\right) = \mathcal{F}\left(\mathbf{s}(t)\right). \end{aligned} \tag{2.26}$$

If we write the related Fourier transformations as:

$$\begin{gathered} \mathbf{j}(\mathbf{x}(t)) = \sum_{-\infty}^{+\infty} J(X)e^{jk\omega t},\ \mathbf{q}(\mathbf{x}(t)) = \sum_{-\infty}^{+\infty} Q(X)e^{jk\omega t},\ \mathbf{s}(t) = \sum_{-\infty}^{+\infty} S(X)e^{jk\omega t}; \\ J(X) = \mathcal{F}\left(\mathbf{j}(\mathbf{x}(t))\right), \qquad Q(X) = \mathcal{F}\left(\mathbf{q}(\mathbf{x}(t))\right), \qquad S(X) = \mathcal{F}\left(\mathbf{s}(t)\right). \end{gathered}$$

The above Eq. (2.26) can be written as:

$$J(X) + \mathcal{F}\frac{d}{dt}\mathcal{F}^{-1}\left(Q(X)\right) = S(X). \tag{2.27}$$

Here define the operator:

$$\Omega = \mathcal{F}\frac{d}{dt}\mathcal{F}^{-1}.$$

The Eq. (2.27) becomes:

$$J(X) + \Omega Q(X) = S(X). \tag{2.28}$$

The analysis for the term $\Omega Q(X)$ works more:

$$\begin{aligned} \Omega Q(X) &= \mathcal{F}\frac{d}{dt}\mathcal{F}^{-1}\left(Q(X)\right) = \mathcal{F}\frac{d}{dt}\mathbf{q}(\mathbf{x},t), \\ &= \mathcal{F}\frac{d}{dt}\sum_{-\infty}^{+\infty} Q_k(X)e^{jk\omega t}, \\ &= \mathcal{F}\left(\sum_{-\infty}^{+\infty} jk\omega Q_k(X)e^{jk\omega t}\right). \end{aligned}$$

So the operator Ω can be written as:

$$\Omega = Block\ diag(\cdots, jkwI, \cdots, -jkwI, \cdots).$$

So Eq. (2.28) can be written into another expression:

$$\Omega \begin{pmatrix} \vdots \\ Q_k(X) \\ \vdots \\ Q_{-k}(X) \\ \vdots \end{pmatrix} = \begin{pmatrix} \ddots & & & & \\ & jk\omega I & & & \\ & & \ddots & & \\ & & & -jk\omega I & \\ & & & & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ Q_k(X) \\ \vdots \\ Q_{-k}(X) \\ \vdots \end{pmatrix}$$

The operator Ω can be considered as a block diagonal matrix. Since $Q_k(X)$ is a $nx \times 1$ vector, I is the identity matrix with size of $nx \times nx$.

So the Eq. (2.28) is an algebra equation indicating a series of equations:

$$\begin{gathered} \vdots \\ J_k(X) + jk\omega Q_k(X) = S_k(X), \\ \vdots \\ J_{-k}(X) - jk\omega Q_{-k}(X) = S_{-k}(X), \\ \vdots \end{gathered}$$

where $X = \begin{pmatrix} \vdots \\ X_k \\ \vdots \\ X_{-k} \\ \vdots \end{pmatrix}$. In general, this system is nonlinear. Restricting $|r| \leq K$, we find a completed system of equations in the X_k.

Therefore the HB solution by HB analysis is expressed as:

$$\mathbf{x}_{HB} = \cdots + X_{-k}e^{-jk\omega t} + \cdots + X_0 + \cdots + X_k e^{jk\omega t} + \cdots . \tag{2.29}$$

Actually we can find the Fourier transformation of DC solution is the DC solution itself, i. e., $\mathbf{x}_0 = X_0$.

2.4.1 Examples

Linear example

To compared with the AC analysis, we use the same linear example of AC analysis. Based what we have analyzed in the AC analysis, we know $k = 1,\ 0,\ -1$ in HB analysis. Therefore Ω can be expressed as a diagonal matrix with size 15×15:

$$\begin{pmatrix} \boxed{\begin{matrix} j\omega & & \\ & \ddots & \\ & & j\omega \end{matrix}}_{k=1} & & \\ & \boxed{\begin{matrix} 0 & & \\ & \ddots & \\ & & 0 \end{matrix}}_{k=0} & \\ & & \boxed{\begin{matrix} -j\omega & & \\ & \ddots & \\ & & -j\omega \end{matrix}}_{k=-1} \end{pmatrix}$$

The three blocked matrices are also diagonal matrix with size 5×5, related to $k = 1,\ 0,\ -1$ respectively.

Also we have $J(X)$ and $Q(X)$, both are 15×1 vectors:

$$J(X) = \begin{pmatrix} \boxed{\begin{matrix} -I_{b1} + \frac{V_1 - V_2}{R_1} \\ -\frac{V_1 - V_2}{R_1} + \frac{V_2 - V_3}{R_2} \\ -\frac{V_2 - V_3}{R_2} + I_{b4} \\ V_1 \\ V_3 \end{matrix}}_{X=X_1} \\ \boxed{\begin{matrix} -I_{b1} + \frac{V_1 - V_2}{R_1} \\ -\frac{V_1 - V_2}{R_1} + \frac{V_2 - V_3}{R_2} \\ -\frac{V_2 - V_3}{R_2} + I_{b4} \\ V_1 \\ V_3 \end{matrix}}_{X=X_0} \\ \boxed{\begin{matrix} -I_{b1} + \frac{V_1 - V_2}{R_1} \\ -\frac{V_1 - V_2}{R_1} + \frac{V_2 - V_3}{R_2} \\ -\frac{V_2 - V_3}{R_2} + I_{b4} \\ V_1 \\ V_3 \end{matrix}}_{X=X_{-1}} \end{pmatrix},\ Q(X) = \begin{pmatrix} \boxed{\begin{matrix} 0 \\ CV_2 \\ 0 \\ 0 \\ -LI_{b4} \end{matrix}}_{X=X_1} \\ \boxed{\begin{matrix} 0 \\ CV_2 \\ 0 \\ 0 \\ -LI_{b4} \end{matrix}}_{X=X_0} \\ \boxed{\begin{matrix} 0 \\ CV_2 \\ 0 \\ 0 \\ -LI_{b4} \end{matrix}}_{X=X_{-1}} \end{pmatrix},\ S(X) = \begin{pmatrix} \boxed{\begin{matrix} 0 \\ 0 \\ 0 \\ -j\frac{s_a}{2} \\ 0 \end{matrix}}_{k=1} \\ \boxed{\begin{matrix} 0 \\ 0 \\ 0 \\ s_0 \\ 0 \end{matrix}}_{k=0} \\ \boxed{\begin{matrix} 0 \\ 0 \\ 0 \\ j\frac{s_a}{2} \\ 0 \end{matrix}}_{k=-1} \end{pmatrix}$$

We get a equation as $h(X) = J(X) + \Omega Q(X) - S(X) = 0$, which is a complex function. So if we assume that the solution is $X = X_{re} + jX_{im}$, and substituting such solution into $h(X)$, we will get a new function as $h_{re}(X) + jh_{im}(X) = 0$, so actually we should solve two sub-equations: $h_{re}(X) = 0$ and $h_{im}(X) = 0$ to get X_{re} and X_{im}.

Nonlinear example

For the nonlinear example like before, the resistor R_2 is set as a nonlinear elements which has the relation: $i_{b3} = (v_2 - v_3)^3 + 2(v_2 - v_3)^2 + (v_2 - v_3)$.

Nonlinear problem is more complicated than linear example by using HB analysis. In above case, we can find that if the $\mathbf{x}$ is substituted into i_{b3}, the frequency domain will be added some higher frequency, that is, k will be enlarged by ± 2 and ± 3.

In practice, we always use Fourier transformation to transfer X of frequency domain into $\mathbf{x}$ of time domain, such that we substitute the $\mathbf{x} = \sum_{-\infty}^{+\infty} X_k e^{jk\omega t}$ into $\mathbf{j}$ and $\mathbf{q}$. Therefore we will have new $\mathbf{j}$ and new $\mathbf{q}$ with X_k and the expression of $e^{jk\omega t}$. For these new $\mathbf{j}$ and $\mathbf{q}$, it is easy to write out the Fourier transformation by $\mathbf{j} = \sum_{-\infty}^{+\infty} J(X_k)e^{jk\omega t}$ and $\mathbf{q} = \sum_{-\infty}^{+\infty} Q(X_k)e^{jk\omega t}$, whose frequencies are bounded to k and have same frequencies as HB solution. So that we can use these bounded $J(X_k)$ and $Q(X_k)$ into the Eq. (2.28) and do the similar precedures as linear example.

The Fourier transformation we mentioned is the discrete Fourier transformation(DFT). In this case, we obtain $\mathbf{x}, \mathbf{j}, \mathbf{q}$ in a finite number of time domain. We should always limit the number of $J_k = J(X_k)$ and $Q_k = Q(X_k)$.

2.5 Transient Analysis

The transient analysis determines the response of the circuit over a special time interval $[0\ \ T]$. We have the DAE of circuit in general:

$$\mathbf{j}(\mathbf{x}(t)) + \frac{d\mathbf{q}(\mathbf{x}(t))}{dt} = \mathbf{s}(t).$$

Using $Backward\ Differential\ Formula$ (BDF), the DAE turns into:

$$\begin{aligned} & \mathbf{j}(\mathbf{x}(\tilde{t}_{r+1})) + \frac{\mathbf{q}(\mathbf{x}(\tilde{t}_{r+1})) - \mathbf{q}(\mathbf{x}(\tilde{t}_r))}{\triangle t} = \mathbf{s}(\tilde{t}_{r+1}), \\ \Rightarrow\ & \mathbf{j}(\mathbf{x}(\tilde{t}_{r+1})) + \frac{\mathbf{q}(\mathbf{x}(\tilde{t}_{r+1}))}{\triangle t} = \mathbf{s}(\tilde{t}_{r+1}) + \frac{\mathbf{q}(\mathbf{x}(\tilde{t}_r))}{\triangle t}, \\ \Rightarrow\ & h(\mathbf{x}_{r+1}) = \mathbf{j}(\mathbf{x}(\tilde{t}_{r+1})) + \frac{\mathbf{q}(\mathbf{x}(\tilde{t}_{r+1}))}{\triangle t} - \mathbf{s}(\tilde{t}_{r+1}) - \frac{\mathbf{q}(\mathbf{x}(\tilde{t}_r))}{\triangle t}. \end{aligned} \tag{2.30}$$

The time interval is assumed to be divided into $(n+1)$ sub-intervals by points. r means a random point in such intervals, shown in the below Fig. (2.5),

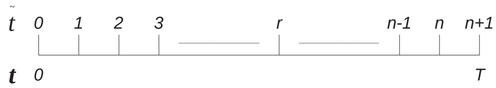


Figure 2.5: Division of time domain

Which implies the each time is: $t_r = \tilde{t} \cdot \frac{T}{n+1}$, $\tilde{t} = 0, 1, \cdots, n+1$, simply remarking corresponding time-step as $\tilde{t}_r$. Since the DC solution is known, we can choose the initial condition of Eq. (2.30) as DC condition, i. e.: $\mathbf{x} = \mathbf{x}_0,\ \tilde{t} = 0$. We solved Eq. (2.30) by NR method, assuming $\mathbf{x}_r$ to be known, and also assuming that $\mathbf{x}_r$ to be the initial guess for the $\mathbf{x}_{r+1}$, that is, $\mathbf{x}_{r+1}^{(1)} = \mathbf{x}_r$, we have:

$$\mathbf{x}_{r+1}^{(2)} = \mathbf{x}_r - \left[\frac{\partial h(\mathbf{x}_r)}{\partial \mathbf{x}}\right]^{-1} h(\mathbf{x}_r) = \mathbf{x}_r - \left[\mathcal{G}(\mathbf{x}_r) + \frac{1}{\triangle t}\mathcal{C}(\mathbf{x}_r)\right]^{-1} h(\mathbf{x}_r),$$

The above equation should do enough iteration steps to satisfy the tolerance requirement, therefore we get $\mathbf{x}_{r+1}$.

The calculating starts from DC condition, going ahead time-step by time-step. At each time-step, the solution can be got using NR method. At final, it will stop at the time-step $\tilde{t} = n+1$.

Here we discuess about the differential index of DAE. The index implies the complication of the DAE. The DAE with a high index is not easy to be solved. We call a DAE has index $\mathcal{V}$ if $\mathcal{V}$ is the mininal value for the DAE expression:

$$\frac{d^{\mathcal{V}}}{dt^{\mathcal{V}}} h(t, \mathbf{x}, \frac{d\mathbf{x}}{dt}) = \mathbf{0}.$$

In general, the index of circuit DAE is no more than 1.

2.5.1 Examples

Linear example

Using the same circuit as before, we have the circuit equation as:

$$\begin{pmatrix} -i_{b1} + \frac{v_1 - v_2}{R_1} \\ -\frac{v_1 - v_2}{R_1} + \frac{v_2 - v_3}{R_2} \\ -\frac{v_2 - v_3}{R_2} + i_{b4} \\ v_1 \\ v_3 \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} 0 \\ Cv_2 \\ 0 \\ 0 \\ -Li_{b4} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ s(t) \\ 0 \end{pmatrix},$$

applying the BDF to the circuit equation,

$$\begin{pmatrix} -i_{b1}(\tilde{t}_{r+1}) + \frac{v_1(\tilde{t}_{r+1}) - v_2(\tilde{t}_{r+1})}{R_1} \\ -\frac{v_1(\tilde{t}_{r+1}) - v_2(\tilde{t}_{r+1})}{R_1} + \frac{v_2(\tilde{t}_{r+1}) - v_3(\tilde{t}_{r+1})}{R_2} \\ -\frac{v_2(\tilde{t}_{r+1}) - v_3(\tilde{t}_{r+1})}{R_2} + i_{b4}(\tilde{t}_{r+1}) \\ v_1(\tilde{t}_{r+1}) \\ v_3(\tilde{t}_{r+1}) \end{pmatrix} + \frac{1}{\triangle t} \left[\begin{pmatrix} 0 \\ Cv_2(\tilde{t}_{r+1}) \\ 0 \\ 0 \\ -Li_{b4}(\tilde{t}_{r+1}) \end{pmatrix} - \begin{pmatrix} 0 \\ Cv_2(\tilde{t}_r) \\ 0 \\ 0 \\ -Li_{b4}(\tilde{t}_r) \end{pmatrix} \right] = \begin{pmatrix} 0 \\ 0 \\ 0 \\ s(\tilde{t}_{r+1}) \\ 0 \end{pmatrix},$$

We have:

$$\tilde{h}(\mathbf{x}) = \begin{pmatrix} -i_{b1}(\tilde{t}_{r+1}) + \frac{v_1(\tilde{t}_{r+1}) - v_2(\tilde{t}_{r+1})}{R_1} \\ -\frac{v_1(\tilde{t}_{r+1}) - v_2(\tilde{t}_{r+1})}{R_1} + \frac{v_2(\tilde{t}_{r+1}) - v_3(\tilde{t}_{r+1})}{R_2} + \frac{1}{\triangle t} Cv_2(\tilde{t}_{r+1}) - \frac{1}{\triangle t} Cv_2(\tilde{t}_r) \\ -\frac{v_2(\tilde{t}_{r+1}) - v_3(\tilde{t}_{r+1})}{R_2} + i_{b4}(\tilde{t}_{r+1}) \\ v_1(\tilde{t}_{r+1}) - s(\tilde{t}_{r+1}) \\ v_3(\tilde{t}_{r+1}) - \frac{1}{\triangle t} Li_{b4}(\tilde{t}_{r+1}) + \frac{1}{\triangle t} Li_{b4}(\tilde{t}_r) \end{pmatrix} = \mathbf{0}.$$

$\mathbf{x}(\tilde{t}_r)$ is known from the past for the time-step $\tilde{t}_{r+1}$, we have:

$$\frac{d\tilde{h}(\mathbf{x})}{d\mathbf{x}} = \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & -1 & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_2} + \frac{C}{\triangle t} & -\frac{1}{R_2} & 0 & 0 \\ 0 & -\frac{1}{R_2} & \frac{1}{R_2} & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -\frac{L}{\triangle t} \end{pmatrix}.$$

Nonlinear example

If the R_2 is the nonlinear element as described before: $i_{b3} = (v_2 - v_3)^3 + 2(v_2 - v_3)^2 + (v_2 - v_3)$. Applying the BDF to the DAE we have got,

$$\tilde{h}(\mathbf{x}) = \begin{pmatrix} -i_{b1}(\tilde{t}_{r+1}) + \frac{v_1(\tilde{t}_{r+1}) - v_2(\tilde{t}_{r+1})}{R_1} \\ -\frac{v_1(\tilde{t}_{r+1}) - v_2(\tilde{t}_{r+1})}{R_1} + \varrho + \frac{1}{\triangle t} Cv_2(\tilde{t}_{r+1}) - \frac{1}{\triangle t} Cv_2(\tilde{t}_r) \\ -\varrho + i_{b4}(\tilde{t}_{r+1}) \\ v_1(\tilde{t}_{r+1}) - s(\tilde{t}_{r+1}) \\ v_3(\tilde{t}_{r+1}) - \frac{1}{\triangle t} Li_{b4}(\tilde{t}_{r+1}) + \frac{1}{\triangle t} Li_{b4}(\tilde{t}_r) \end{pmatrix} = \mathbf{0}.$$

where we define

$$\varrho = (v_2(\tilde{t}_{r+1}) - v_3(\tilde{t}_{r+1}))^3 + 2(v_2(\tilde{t}_{r+1}) - v_3(\tilde{t}_{r+1}))^2 + (v_2(\tilde{t}_{r+1}) - v_3(\tilde{t}_{r+1})).$$

$\mathbf{x}(\tilde{t}_r)$ is known from the past for the time-step $\tilde{t}_{r+1}$, we have:

$$\frac{d\tilde{h}(\mathbf{x})}{d\mathbf{x}} = \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & -1 & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} + \tilde{\varrho} + \frac{C}{\triangle t} & -\tilde{\varrho} & 0 & 0 \\ 0 & -\tilde{\varrho} & \tilde{\varrho} & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -\frac{L}{\triangle t} \end{pmatrix}.$$

where we define

$$\tilde{\varrho} = 3v_2^2 - 6v_2v_3 + 3v_3^2 + 4v_2 - 4v_3 + 1.$$

The results of linear and nonlinear examples are illustrated below:

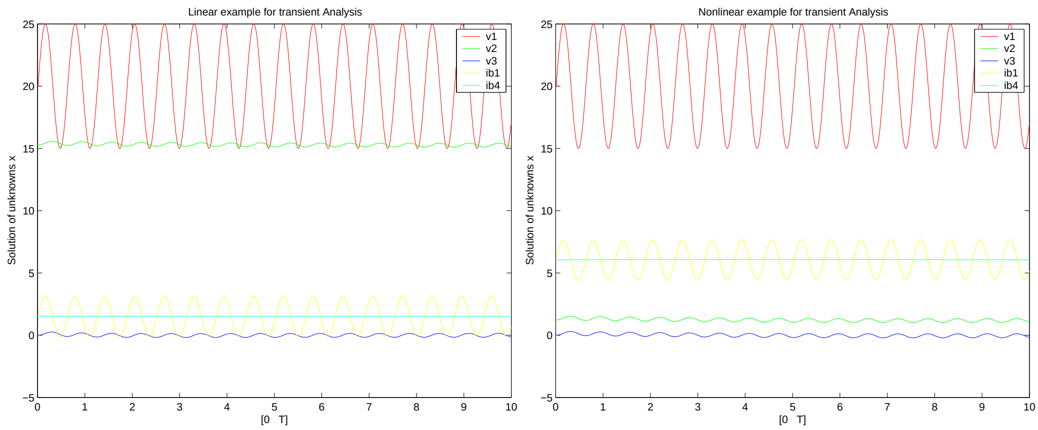


Figure 2.6: Transient solution: $\mathbf{s}(t) = 20 + 5\sin 10t$

Chapter 3

Sensitivity Analysis

The sensitivity analysis in this chapter is discussed on the assumption that the topology of the circuit is not changed in any way. Section 3.1 sketches the approaches for sensitivity analysis, some of them will be illustrated further in the later corresponding analysis. Section 3.2 presents sensitivity analysis for DC condition. The sensitivity analysis for AC condition is given in section 3.3. For HB, the sensitivity analysis is shown in the following section 3.4. In section 3.5, more emphases and approaches are dedicated to the transient analysis. At final, we will discuss the situation in case some elements are taken away from circuit, which is called *Fault Analysis* in section 3.6.

3.1 The approaches of sensitivity analysis

We are interested in the influence of some parameters on the solution of the circuit. The parameter is symbolled as: p_r with number np. We write it as:

$$\mathbf{p} = [p_1, p_2, \cdots, p_{np}]^T.$$

Either the sensitivity of the unknowns on parameters, or the sensitivity of some interesting functions(such as power, heat etc) on parameters is involved in this chapter. We write the interested functions as: $\mathbf{f}$ with number nf, which can be expressed as the below vector:

$$\mathbf{f} = [f_1, f_2, \cdots, f_{nf}]^T.$$

Hence the rate of the unknowns or of the functions varies with the parameter, like:

$$\frac{d\mathbf{x}(\mathbf{p})}{d\mathbf{p}}, \ \ or \ \ \frac{d\mathbf{f}}{d\mathbf{p}}.$$

Where $\frac{d\mathbf{x}(\mathbf{p})}{d\mathbf{p}}$ is symbolled as $\hat{\mathbf{x}}$.

The approaches in this section for sensitivity analysis is listed in the below table:

Analysis	*DiffM*	*DM*	*AdjM*	*IntM*	*Extension intM*
DC	+	+	+	-	-
AC	+	+	+	-	-
HB	+	+	+	-	-
Transient	+	+	+	+	+

Table 3.1: $+$ means 'Available'; $-$ means 'Not available'

In this table, the abbreviations shown above are:
DiffM is *Difference method*, *DM* is *Direct method*,*AdjM* is *Adjoint method*, *IntM* is *Integral method*, *Extension intM* is *Extension integral method*.

- **Difference method**
  We assume that $f(x)$ is known at $f(p_{r+1})$,$f(p_r)$,$f(p_{r-1})$ then derivative can be approximated by finite differences. We have three types of a difference method: $Forward\ Difference\ Formula$(FDF), $Backward\ Difference\ Formula$(BDF) and $Central\ Difference\ Formula$(CDF).

$$FDF: \quad f'(p_r) = \frac{f(p_{r+1}) - f(p_r)}{\triangle x}, \tag{3.1}$$

$$BDF: \quad f'(p_r) = \frac{f(p_r) - f(p_{r-1})}{\triangle x}, \tag{3.2}$$

$$CDF: \quad f'(p_r) = \frac{f(p_{r+1}) - f(p_{r-1})}{2\triangle x}. \tag{3.3}$$

- **Direct method**
  The direct method is based on the direct differentiation of the circuit equations to get $\hat{\mathbf{x}} = \frac{\partial \mathbf{x}}{\partial \mathbf{p}}$. The main idea is to calculate $\hat{\mathbf{x}}$ directly from the circuit solution what we have done in chapter 2.

$$\begin{aligned} \frac{d\mathbf{j}(\mathbf{x}(t,\mathbf{p})}{d\mathbf{p}} + \frac{d}{dt}\frac{d\mathbf{q}(\mathbf{x}(t,\mathbf{p})}{d\mathbf{p}} &= \frac{d\mathbf{s}(t)}{d\mathbf{p}}, \\ \frac{\partial \mathbf{j}}{\partial \mathbf{x}}\hat{\mathbf{x}} + \frac{\partial \mathbf{j}}{\partial \mathbf{p}} + \frac{d}{dt}\left(\frac{\partial \mathbf{q}}{\partial \mathbf{x}}\hat{\mathbf{x}} + \frac{\partial \mathbf{q}}{\partial \mathbf{p}}\right) &= \frac{\partial \mathbf{s}}{\partial \mathbf{p}}. \end{aligned} \tag{3.4}$$

  This equation implicitly demonstrates that the sensitivity circuit still satisfies the laws KCL and KVL.

- **Adjoint method**
  Base on the direct method, if we are interested in the sensitivity of some circuit functions $\mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})$, the adjoint method should be adopted to solve such problems. In general

$$\frac{d\mathbf{f}}{d\mathbf{p}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}}{\partial \mathbf{p}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\hat{\mathbf{x}} + \frac{\partial \mathbf{f}}{\partial \mathbf{p}}, \tag{3.5}$$

where $\hat{\mathbf{x}}$ can be get from direct method without more efforts. However, in general cases, the inner product $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\hat{\mathbf{x}}$ can be obtained more effeciectly, as explained in the next sections.

- **Integral method**
  The integral method is an approach for transient analysis. We assume that the circuit function is given as an integral function:

$$\mathbf{F}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) = \int_0^T \mathbf{f}(\mathbf{x}(\mathbf{t}, \mathbf{p}), \mathbf{p})dt, \tag{3.6}$$

  If the sensitivity of this integral function $\frac{d\mathbf{F}}{d\mathbf{p}}$ is interested, we try to avoid the calculation of $\hat{\mathbf{x}}$ because of its cost. So a luxury equation is introduced to avoid calculating $\hat{\mathbf{x}}$ here.

- **Extension integral method**
  As for the integral method, the so-called extension integral method is only used for transient analysis as well. If the circuit function is $\mathbf{f}(\mathbf{x}(\mathbf{t}, \mathbf{p}), \mathbf{p})$, the main idea of this approach is to use the result of integral method to minimize the cost. The relation of $\mathbf{f}$ and $\mathbf{F}$ is:

$$\mathbf{f}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) = \frac{d\mathbf{F}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})}{dT}. \tag{3.7}$$

  Also a luxury equation is needed in this approach, which is not a totally new equation, luckily having a tight relation to the luxury equation of integral method.

3.2 DC analysis

DC analysis is based on the circuit equation of DC condition. This analysis is most simple but most fundamental.

3.2.1 Direct method

This method is used to get sensitivity of DC solution. From Eq. (3.4), the sensitivity is easy to find:

$$\begin{aligned}\frac{d\mathbf{j}(\mathbf{x}_0)}{d\mathbf{p}} = \frac{d\mathbf{s}_0}{d\mathbf{p}} \Rightarrow \frac{d\mathbf{j}(\mathbf{x}_0)}{d\mathbf{p}} = \mathbf{0} \Rightarrow \frac{\partial\mathbf{j}(\mathbf{x}_0)}{\partial\mathbf{x}}\hat{\mathbf{x}}_0 + \frac{\partial\mathbf{j}(\mathbf{x}_0)}{\partial\mathbf{p}} = \mathbf{0},\\ \Rightarrow \hat{\mathbf{x}}_0 = -\left[\frac{\partial\mathbf{j}(\mathbf{x}_0)}{\partial\mathbf{x}}\right]^{-1}\frac{\partial\mathbf{j}(\mathbf{x}_0)}{\partial\mathbf{p}} = -\mathcal{G}_0^{-1}\frac{\partial\mathbf{j}(\mathbf{x}_0)}{\partial\mathbf{p}}.\end{aligned} \tag{3.8}$$

3.2.2 Adjoint method

The circuit function are assumed as $\mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})$, using what we have got in direct method for Eq. (3.5):

$$\begin{aligned}\frac{d\mathbf{f}}{d\mathbf{p}} &= \frac{\partial\mathbf{f}}{\partial\mathbf{x}}\hat{\mathbf{x}} + \frac{\partial\mathbf{f}}{\partial\mathbf{p}},\\ &= -\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\mathcal{G}_0^{-1}\frac{\partial\mathbf{j}(\mathbf{x}_0)}{\partial\mathbf{p}} + \frac{\partial\mathbf{f}}{\partial\mathbf{p}}.\end{aligned} \tag{3.9}$$

Such that adjoint method is a easy way to determine the sensitivity of many circuit functions on many parameters. In calculating , the first term of Eq. (3.9) has two ways to process:

- First calculating $\mathcal{G}_0^{-1}\frac{\partial\mathbf{j}(\mathbf{x}_0)}{\partial\mathbf{p}}$ named as ξ, then $\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\xi$, which is most efficient when $np < nf$.
- First calculating $\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\mathcal{G}_0^{-1}$ named as $\tilde{\xi}$, then $\tilde{\xi}\frac{\partial\mathbf{j}(\mathbf{x}_0)}{\partial\mathbf{p}}$, which is most efficient when $nf < np$.

3.2.3 Examples

The DC topology is given in Fig. (2.3). the parameter is given as $p = R_1$.

Linear example

1. Direct method:

The circuit equation is:

$$\begin{pmatrix} -i_{b1} + \frac{v_1 - v_2}{R_1} \\ -\frac{v_1 - v_2}{R_1} + \frac{v_2}{R_2} \\ v_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mathbf{s}(0) \end{pmatrix} \tag{3.10}$$

We have:

$$\frac{\partial \mathbf{j}}{\partial \mathbf{p}} = \frac{\partial \mathbf{j}}{\partial R_1} = \begin{pmatrix} -\frac{v_1 - v_2}{R_1^2} \\ \frac{v_1 - v_2}{R_1^2} \\ 0 \end{pmatrix}, \quad \mathcal{G}_0 = \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & -1 \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_2} & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

So we have the sensitivity for the DC solution is:

$$\hat{\mathbf{x}}_0 = -\begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & -1 \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_2} & 0 \\ 1 & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} -\frac{v_1 - v_2}{R_1^2} \\ \frac{v_1 - v_2}{R_1^2} \\ 0 \end{pmatrix}.$$

2. Adjoint method: the interested functions are assumed as: $f_1 = i_{b3} v_{b3} = \frac{v_2^2}{R_2}$, $f_2 = i_{b1} v_1$.

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{pmatrix} 0 & \frac{2v_2}{R_2} & 0 \\ i_{b1} & 0 & v_1 \end{pmatrix}, \quad \frac{\partial \mathbf{f}}{\partial R_1} = \mathbf{0}.$$

So we have:

$$\frac{d\mathbf{f}}{dR_1} = -\begin{pmatrix} 0 & \frac{2v_2}{R_2} & 0 \\ i_{b1} & 0 & v_1 \end{pmatrix} \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & -1 \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_2} & 0 \\ 1 & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} -\frac{v_1 - v_2}{R_1^2} \\ \frac{v_1 - v_2}{R_1^2} \\ 0 \end{pmatrix}.$$

Since $np < nf$, we can calculate the $\xi = \mathcal{G}_0^{-1} \frac{\partial \mathbf{j}}{\partial R_1}$, then calculate $\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \xi$.

Nonlinear example

1. Direct method:

The circuit equation is:

$$\begin{pmatrix} -i_{b1} + \frac{v_1 - v_2}{R_1} \\ -\frac{v_1 - v_2}{R_1} + v_2^3 + 2v_2^2 + v_2 \\ v_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mathbf{s}(0) \end{pmatrix} \tag{3.11}$$

We have:

$$\frac{\partial \mathbf{j}}{\partial \mathbf{p}} = \frac{\partial \mathbf{j}}{\partial R_1} = \begin{pmatrix} -\frac{v_1 - v_2}{R_1^2} \\ \frac{v_1 - v_2}{R_1^2} \\ 0 \end{pmatrix}, \quad \mathcal{G}_0 = \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & -1 \\ -\frac{1}{R_1} & \frac{1}{R_1} + 3v_2^2 + 4v_2 + 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

So we have:

$$\hat{\mathbf{x}}_0 = -\begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & -1 \\ -\frac{1}{R_1} & \frac{1}{R_1} + 3v_2^2 + 4v_2 + 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} -\frac{v_1 - v_2}{R_1^2} \\ \frac{v_1 - v_2}{R_1^2} \\ 0 \end{pmatrix}.$$

2. Adjoint method: the interested functions are assumed as: $f_1 = i_{b3}v_{b3} = (v_2^3 + 2v_2^2 + v_2)v_2 = v_2^4 + 2v_2^3 + v_2^2, \;\; f_2 = i_{b1}v_1$.

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{pmatrix} 0 & 3v_2^3 + 6v_2^2 + 2v_2 & 0 \\ i_{b1} & 0 & v_1 \end{pmatrix}, \;\; \frac{\partial \mathbf{f}}{\partial R_1} = \mathbf{0}.$$

So we have:

$$\frac{d\mathbf{f}}{dR_1} = -\begin{pmatrix} 0 & 3v_2^3 + 6v_2^2 + 2v_2 & 0 \\ i_{b1} & 0 & v_1 \end{pmatrix} \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & -1 \\ -\frac{1}{R_1} & \frac{1}{R_1} + 3v_2^2 + 4v_2 + 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} -\frac{v_1 - v_2}{R_1^2} \\ \frac{v_1 - v_2}{R_1^2} \\ 0 \end{pmatrix}.$$

Results

The results for linear and nonlinear examples are illustrated below:

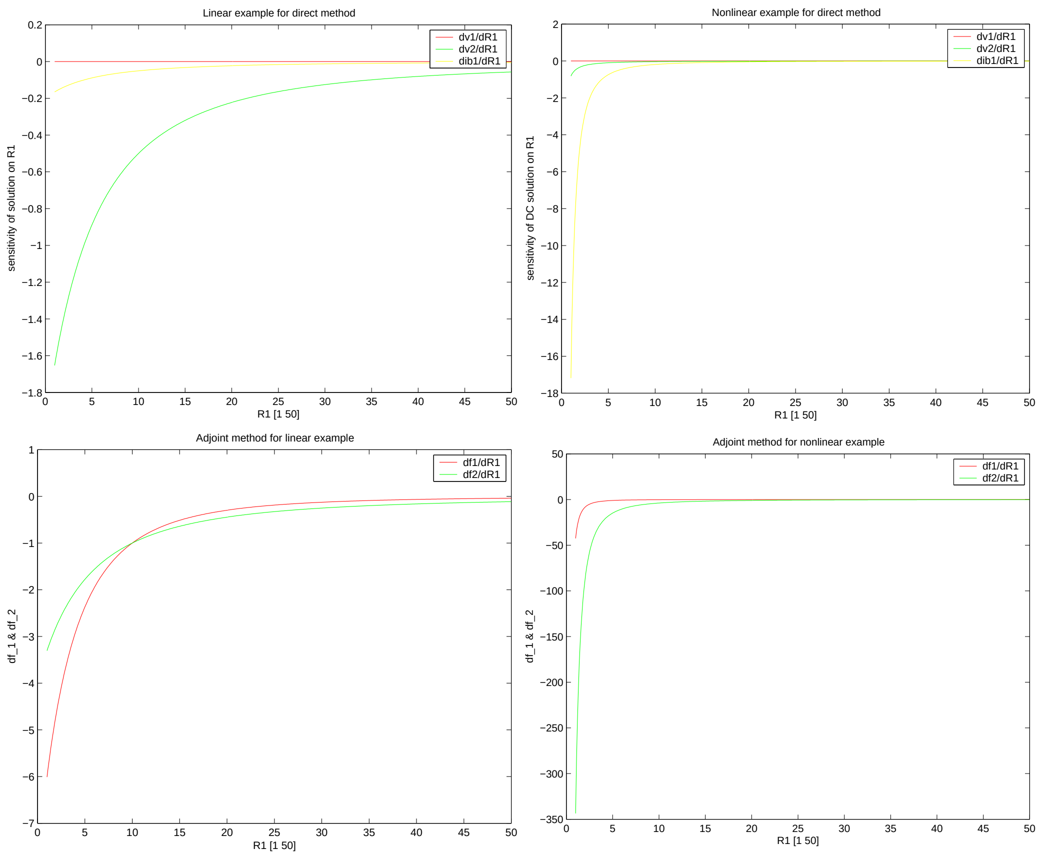


Figure 3.1: DC sensitivity solution: $s_0 = 20$

3.3 AC analysis

AC analysis is more complicated than DC analysis. The source is a varying signal with time.

3.3.1 Direct method

From circuit equation of AC analysis Eq. (2.23), we deduct the sensitivity of AC solutions:

$$\begin{aligned} (\mathcal{G}_0 + jk\omega\mathcal{C}_0)\tilde{X}_k &= \tilde{S}_k, \\ \Rightarrow \left[\frac{d}{d\mathbf{p}}(\mathcal{G}_0 + jk\omega\mathcal{C}_0)\right]\tilde{X}_k + (\mathcal{G}_0 + jk\omega\mathcal{C}_0)\frac{d\tilde{X}_k}{d\mathbf{p}} &= \frac{\partial \tilde{S}_k}{\partial \mathbf{p}}. \end{aligned} \tag{3.12}$$

where:

$$\mathcal{G}_0 = \frac{d\mathbf{j}(\mathbf{x}_0)}{d\mathbf{x}}, \quad \mathcal{C}_0 = \frac{d\mathbf{q}(\mathbf{x}_0)}{d\mathbf{x}}.$$

Therefore the Eq. (3.12) can be worked out:

$$\begin{aligned} \Rightarrow \left[\frac{d\mathcal{G}_0}{d\mathbf{p}} + jk\omega\frac{d\mathcal{C}_0}{d\mathbf{p}}\right]\tilde{X}_k + (\mathcal{G}_0 + jk\omega\mathcal{C}_0)\frac{d\tilde{X}_k}{d\mathbf{p}} &= \frac{\partial \tilde{S}_k}{\partial \mathbf{p}}, \\ \Rightarrow \left[\left(\frac{\partial\mathcal{G}_0}{\partial\mathbf{x}}\hat{\mathbf{x}}_0 + \frac{\partial\mathcal{G}_0}{\partial\mathbf{p}}\right) + jk\omega\left(\frac{\partial\mathcal{C}_0}{\partial\mathbf{x}}\hat{\mathbf{x}}_0 + \frac{\partial\mathcal{C}_0}{\partial\mathbf{p}}\right)\right]\tilde{X}_k + (\mathcal{G}_0 + jk\omega\mathcal{C}_0)\hat{\tilde{X}}_k &= \frac{\partial \tilde{S}_k}{\partial \mathbf{p}} \end{aligned}$$

$$\begin{aligned} \Rightarrow \hat{\tilde{X}}_k &= (\mathcal{G}_0 + jk\omega\mathcal{C}_0)^{-1}\left[\frac{\partial \tilde{S}_k}{\partial \mathbf{p}} - \left[\left(\frac{\partial\mathcal{G}_0}{\partial\mathbf{x}} + jk\omega\frac{\partial\mathcal{C}_0}{\partial\mathbf{x}}\right)\hat{\mathbf{x}}_0 + \left(\frac{\partial\mathcal{G}_0}{\partial\mathbf{p}} + jk\omega\frac{\partial\mathcal{C}_0}{\partial\mathbf{p}}\right)\right]\tilde{X}_k\right], \\ \Rightarrow \hat{\tilde{X}}_k &= (\mathcal{G}_0 + jk\omega\mathcal{C}_0)^{-1}\left[\frac{\partial \tilde{S}_k}{\partial \mathbf{p}} - \left[\frac{\partial(\mathcal{G}_0 + jk\omega\mathcal{C}_0)}{\partial\mathbf{x}}\hat{\mathbf{x}}_0 + \frac{\partial(\mathcal{G}_0 + jk\omega\mathcal{C}_0)}{\partial\mathbf{p}}\right]\tilde{X}_k\right]. \end{aligned} \tag{3.13}$$

If we define:

$$\zeta = \mathcal{G}_0 + jk\omega\mathcal{C}_0.$$

The Eq. (3.13) is simplified as:

$$\hat{\tilde{X}}_k = \zeta_k^{-1}\left[\frac{\partial \tilde{S}_k}{\partial \mathbf{p}} - \left(\frac{\partial\zeta_k}{\partial\mathbf{x}}\hat{\mathbf{x}}_0 + \frac{\partial\zeta_k}{\partial\mathbf{p}}\right)\tilde{X}_k\right]. \tag{3.14}$$

In Eq. (3.14), the product of $\frac{\partial\zeta_k}{\partial\mathbf{x}}\hat{\mathbf{x}}_0$ and $\frac{\partial\zeta_k}{\partial\mathbf{p}}\tilde{X}_k$ should be implemented as:

$$\left(\frac{\partial\zeta_k}{\partial\mathbf{x}_1}\hat{\mathbf{x}}_0, \frac{\partial\zeta_k}{\partial\mathbf{x}_2}\hat{\mathbf{x}}_0, \cdots, \frac{\partial\zeta_k}{\partial\mathbf{x}_{nx}}\hat{\mathbf{x}}_0\right), \quad \left(\frac{\partial\zeta_k}{\partial\mathbf{p}_1}\tilde{X}_k, \frac{\partial\zeta_k}{\partial\mathbf{p}_2}\tilde{X}_k, \cdots, \frac{\partial\zeta_k}{\partial\mathbf{p}_{np}}\tilde{X}_k\right),$$

respectively.

If $\frac{\partial \zeta_k}{\partial \mathbf{x}} \hat{\mathbf{x}}_0$ is symbolled as γ, the product of $\gamma \tilde{X}_k$ is more complicated. From the above expression, γ is known as a $(nx \times np) \times nx$ matrix, i. e.:

$$\gamma = (\gamma_1, \gamma_2, \cdots, \gamma_{nx}),$$

where each γ_r is a $nx \times np$ matrix. If we define a new matrix $\tilde{\gamma}$ as:

$$\tilde{\gamma} = (\tilde{\gamma}_1, \tilde{\gamma}_2, \cdots, \tilde{\gamma}_{np}),$$

where $\tilde{\gamma}_r$ is a $nx \times nx$ matrix:

$$\tilde{\gamma}_r = (\gamma_1(:, r), \gamma_2(:, r), \cdots, \gamma_{nx}(:, r)).$$

and where $\gamma_r(:, r)$ means the r-column of γ_r. Then the expression $\gamma \tilde{X}_k$ is processed as:

$$\left(\tilde{\gamma}_1 \tilde{X}_k, \tilde{\gamma}_2 \tilde{X}_k, \cdots, \tilde{\gamma}_{np} \tilde{X}_k \right).$$

Recalling Eq. (2.24) of AC simulation, we have the sensitivity of AC solution:

$$\mathbf{x}_{AC} = \mathbf{x}_0 + \sum_{-\infty}^{+\infty} \tilde{X}_k e^{jk\omega t},$$

$$\Rightarrow \hat{\mathbf{x}}_{AC} = \frac{d\mathbf{x}_0}{d\mathbf{p}} + \sum_{-\infty}^{+\infty} \frac{d\tilde{X}_k}{d\mathbf{p}} e^{jk\omega t} = \hat{\mathbf{x}}_0 + \sum_{-\infty}^{+\infty} \hat{\tilde{X}}_k e^{jk\omega t} \tag{3.15}$$

From above analysis, we find a essential operator ζ. This term can be acquired from AC simulation directly, which may reduce the waste of computing.

3.3.2 Adjoint method

Based the sensitivity of AC solution, the sensitivities of circuit functions can be easily acquired using Eq.(3.5).

$$\frac{d\mathbf{f}}{d\mathbf{p}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}}{\partial \mathbf{p}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \hat{\mathbf{x}} + \frac{\partial \mathbf{f}}{\partial \mathbf{p}}.$$

3.3.3 Examples

If we assume that the source is: $\mathbf{s} = \mathbf{s}_0 + \mathbf{s}_a \sin \omega t$, the parameter is $p = R_1$. The circuit topology is given as Fig. (2.1).

Linear example

1. Direct method:

The circuit equation is:

$$\begin{pmatrix} -i_{b1}+\frac{v_1-v_2}{R_1} \\ -\frac{v_1-v_2}{R_1}+\frac{v_2-v_3}{R_2} \\ -\frac{v_2-v_3}{R_2}+i_{b4} \\ v_1 \\ v_3 \end{pmatrix} + \frac{d}{dt}\begin{pmatrix} 0 \\ Cv_2 \\ 0 \\ 0 \\ -Li_{b4} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ s(t) \\ 0 \end{pmatrix},$$

Matrices $\mathcal{G}_0$ and $\mathcal{C}_0$ are the same as the linear example in AC simulation. Also we have got the solutions $\tilde{X}_k$ from AC simulation.

So following the Eq. (3.14), we have matrices:

$$\zeta = \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & -1 & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1}+\frac{1}{R_2}+jk\omega C & -\frac{1}{R_2} & 0 & 0 \\ 0 & -\frac{1}{R_2} & \frac{1}{R_2} & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -jk\omega L \end{pmatrix},$$

$$\frac{\partial \zeta}{\partial \mathbf{x}} = \mathbf{0}, \quad \frac{\partial \zeta}{\partial \mathbf{p}} = \frac{\partial \zeta}{\partial R_1} = \begin{pmatrix} -\frac{1}{R_1^2} & \frac{1}{R_1^2} & 0 & 0 & 0 \\ \frac{1}{R_1^2} & -\frac{1}{R_1^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \frac{\partial \tilde{S}}{\partial R_1} = \mathbf{0}.$$

Following the Eq. (3.14), we have:

$$\hat{\tilde{X}}_1 = -\zeta_1^{-1}\frac{\partial \zeta_1}{\partial R_1}\tilde{X}_1, \quad \hat{\tilde{X}}_{-1} = -\zeta_{-1}^{-1}\frac{\partial \zeta_{-1}}{\partial R_1}\tilde{X}_{-1},$$
$$\Rightarrow \quad \hat{\mathbf{x}}_{AC} = \hat{\mathbf{x}}_0 + \hat{\tilde{X}}_1 e^{j\omega t} + \hat{\tilde{X}}_{-1} e^{-j\omega t}.$$

2. Adjoint method: the interested functions are assumed as: $f_1 = i_{b3}v_{b3} = \frac{(v_2-v_3)^2}{R_2}$, $f_2 = i_{b1}v_1$.

We have:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{pmatrix} 0 & \frac{2(v_2-v_3)}{R_2} & -\frac{2(v_2-v_3)}{R_2} & 0 & 0 \\ i_{b1} & 0 & 0 & v_1 & 0 \end{pmatrix}, \quad \frac{\partial \mathbf{f}}{\partial R_1} = \mathbf{0}.$$

So the sensitivity equation for the two functions is:

$$\frac{d\mathbf{f}}{dR_1} = \begin{pmatrix} 0 & \frac{2(v_2-v_3)}{R_2} & -\frac{2(v_2-v_3)}{R_2} & 0 & 0 \\ i_{b1} & 0 & 0 & v_1 & 0 \end{pmatrix} \hat{\mathbf{x}}_{AC}.$$

Nonlinear example

1. Direct method:

   If the component R_2 is nonlinear element, the relation is like before:

$$i_{b3} = (v_2 - v_3)^3 + 2(v_2 - v_3)^2 + (v_2 - v_3).$$

   We have matrices,

$$\zeta = \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & -1 & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} + (3v_2^2 - 6v_2v_3 + 3v_3^2 + 4v_2 - 4v_3 + 1) + jk\omega C & -(3v_2^2 - 6v_2v_3 + 3v_3^2 + 4v_2 - 4v_3 + 1) & 0 & 0 \\ 0 & -(3v_2^2 - 6v_2v_3 + 3v_3^2 + 4v_2 - 4v_3 + 1) & (3v_2^2 - 6v_2v_3 + 3v_3^2 + 4v_2 - 4v_3 + 1) & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -jk\omega L \end{pmatrix},$$

$$\frac{\partial \zeta}{\partial \mathbf{x}} = \begin{pmatrix} \phi & \varpi & -\varpi & \phi & \phi \end{pmatrix}, \; where\ \phi\ is\ a\ 5 - by - 5\ zero\ matrix,$$

$$and\ \varpi = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 6v_2 - 6v_3 + 4 & -(6v_2 - 6v_3 + 4) & 0 & 0 \\ 0 & -(6v_2 - 6v_3 + 4) & 6v_2 - 6v_3 + 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\frac{\partial \zeta}{\partial \mathbf{p}} = \frac{\partial \zeta}{\partial R_1} = \begin{pmatrix} -\frac{1}{R_1^2} & \frac{1}{R_1^2} & 0 & 0 & 0 \\ \frac{1}{R_1^2} & -\frac{1}{R_1^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

   Following the Eq. (3.14), we have:

$$\hat{\tilde{X}}_1 = -\zeta_1^{-1}\left(\frac{\partial \zeta_1}{\partial \mathbf{x}}\hat{\mathbf{x}}_0 + \frac{\partial \zeta_1}{\partial \mathbf{p}}\right)\tilde{X}_1, \;\; \hat{\tilde{X}}_1 = -\zeta_{-1}^{-1}\left(\frac{\partial \zeta_{-1}}{\partial \mathbf{x}}\hat{\mathbf{x}}_0 + \frac{\partial \zeta_{-1}}{\partial \mathbf{p}}\right)\tilde{X}_{-1},$$

$$\Rightarrow \;\; \hat{\mathbf{x}}_{AC} = \hat{\mathbf{x}}_0 + \hat{\tilde{X}}_1 e^{j\omega t} + \hat{\tilde{X}}_{-1} e^{-j\omega t}.$$

2. Adjoint method: the interested functions are assumed as: $f_1 = i_{b3}v_{b3} = (v_2 - v_3)^4 + 2(v_2 - v_3)^3 + (v_2 - v_3)^2, \; f_2 = i_{b1}v_1$.

   We have:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{pmatrix} 0 & \alpha & -\alpha & 0 & 0 \\ i_{b1} & 0 & 0 & v_1 & 0 \end{pmatrix}, \;\; \frac{\partial \mathbf{f}}{\partial R_1} = \mathbf{0}.$$

   where $\alpha = 4(v_2 - v_3)^3 + 6(v_2 - v_3)^2 + 2(v_2 - v_3)$. So the sensitivity equation for the two functions is:

$$\frac{d\mathbf{f}}{dR_1} = \begin{pmatrix} 0 & \alpha & -\alpha & 0 & 0 \\ i_{b1} & 0 & 0 & v_1 & 0 \end{pmatrix}\hat{\mathbf{x}}_{AC}.$$

Results

The results for linear and nonlinear examples are illustrated below,

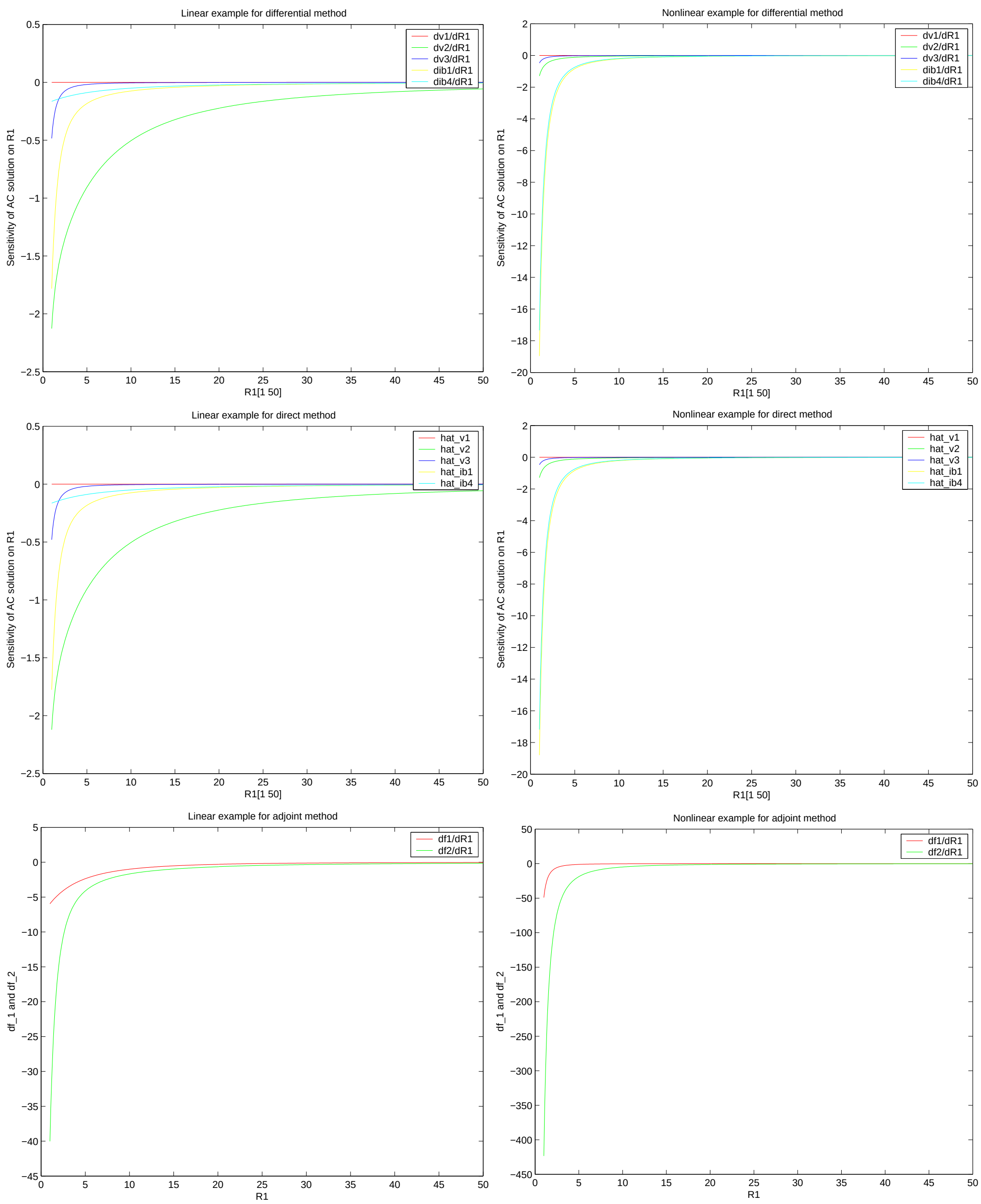


Figure 3.2: AC sensitivity solution: $s(t) = 20 + 5\sin 10t$

3.4 HB analysis

3.4.1 Direct method

From the HB simulation of Chapter 2, the circuit solution can be made out.

Now we investigate the sensitivity of HB analysis, starting from Eq. (2.28) of the DAE:

$$\begin{aligned} \frac{d\Omega Q(X)}{d\mathbf{p}} + \frac{dJ(X)}{d\mathbf{p}} &= \frac{dS(X)}{d\mathbf{p}}, \\ \Rightarrow \Omega\left(\frac{\partial Q(X)}{\partial X}\hat{X} + \frac{\partial Q(X)}{\partial \mathbf{p}}\right) + \frac{\partial J(X)}{\partial X}\hat{X} + \frac{\partial J(X)}{\partial \mathbf{p}} &= \frac{dS(X)}{d\mathbf{p}}, \\ \Rightarrow \hat{X} = \left(\Omega\frac{\partial Q(X)}{\partial X} + \frac{\partial J(X)}{\partial X}\right)^{-1}\left(\frac{dS(X)}{d\mathbf{p}} - \Omega\frac{\partial Q(X)}{\partial \mathbf{p}} - \frac{\partial J(X)}{\partial \mathbf{p}}\right)&. \end{aligned} \tag{3.16}$$

Here we assumed that Ω does not depend on $\mathbf{p}$. This excludes free oscillation problems, in which Ω depends on p. These problems require a more thorough analysis. As far as what we did, we get the sensitivity of X, which is the Fourier transformation of the HB solution. So for the sensitivity of HB condition, we have:

$$\begin{aligned} \hat{\mathbf{x}}_{HB} &= \sum_{-\infty}^{+\infty} \hat{X}_k e^{jk\omega t}, \\ &= \cdots + \hat{X}_{-k}e^{-jk\omega t} + \cdots + \hat{X}_0 + \cdots + \hat{X}_k e^{jk\omega t} + \cdots . \end{aligned}$$

In the above equation, actually the term $\hat{X}_0$ means the sensitivity of DC condition. But now it has no need to be calculated at first, it can be worked out with other k-conditions simultaneously by using Eq. (3.16).

3.4.2 Adjoint method

The adjoint method is straightforward since we have got the sensitivity of solution, using Eq. (3.5),

$$\frac{d\mathbf{f}}{d\mathbf{p}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}}{\partial \mathbf{p}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\hat{\mathbf{x}} + \frac{\partial \mathbf{f}}{\partial \mathbf{p}}.$$

3.5 Transient analysis

3.5.1 Direct method

Like all other analyses, we want to know about the sensitivity of transient solution at first. Recalling Eq. (2.30),

$$\begin{aligned}
&\mathbf{j}(\mathbf{x}(\tilde{t}_{r+1},\mathbf{p})) + \frac{\mathbf{q}(\mathbf{x}(\tilde{t}_{r+1},\mathbf{p})) - \mathbf{q}(\mathbf{x}(\tilde{t}_r,\mathbf{p}))}{\triangle t} = \mathbf{s}(\tilde{t}_{r+1},\mathbf{p}),\\
\Rightarrow\ &\frac{d\mathbf{j}(\mathbf{x}(\tilde{t}_{r+1},\mathbf{p}))}{d\mathbf{p}} + \frac{1}{\triangle t}\left(\frac{d\mathbf{q}(\mathbf{x}(\tilde{t}_{r+1}))}{d\mathbf{p}} - \frac{d\mathbf{q}(\mathbf{x}(\tilde{t}_r,\mathbf{p}))}{d\mathbf{p}}\right) = \frac{d\mathbf{s}(\tilde{t}_{r+1})}{d\mathbf{p}},\\
\Rightarrow \hat{\mathbf{x}}_{r+1} = &\left(\mathcal{G}_{r+1} + \frac{1}{\triangle t}\mathcal{C}_{r+1}\right)^{-1}\left(\frac{d\mathbf{s}(\tilde{t}_{r+1})}{d\mathbf{p}} + \frac{1}{\triangle t}\mathcal{C}_r\hat{\mathbf{x}}_r + \frac{1}{\triangle t}\frac{\partial\mathbf{q}(\mathbf{x}_r,\tilde{t}_r)}{\partial\mathbf{p}}\right.\\
&\left. - \frac{1}{\triangle t}\frac{\partial\mathbf{q}(\mathbf{x}(\tilde{t}_{r+1},\mathbf{p}))}{\partial\mathbf{p}} - \frac{\partial\mathbf{j}(\mathbf{x}(\tilde{t}_{r+1},\mathbf{p}))}{\partial\mathbf{p}}\right). \qquad (3.17)
\end{aligned}$$

where $\mathcal{G}_r = \frac{d\mathbf{j}(\mathbf{x}_r(\tilde{t}_r,\mathbf{p}))}{d\mathbf{x}}$ and $\mathcal{C}_r = \frac{d\mathbf{q}(\mathbf{x}_r(\tilde{t}_r,\mathbf{p}))}{d\mathbf{x}}$. From Eq. (3.17), we find that the matrices $\mathcal{G}$ and $\mathcal{C}$ vary with time and depend on the circuit solution at $t = t_r$. The all description of $\left(\mathcal{G}_{r+1} + \frac{1}{\triangle t}\mathcal{C}_{r+1}\right)$ may be re-used from the Newton process to determine the circuit solution $\mathbf{x}_{r+1}$.

We also start from the DC sensitivity, and then at each time-step, we use Eq. (3.17) to get the sensitivity of next time-step.

3.5.2 Adjoint method

Like what we have got in the direct method, the sensitivities of the circuit functions are:

$$\frac{d\mathbf{f}}{d\mathbf{p}} = \frac{\partial\mathbf{f}}{\partial\mathbf{x}}\frac{\partial\mathbf{x}}{\partial\mathbf{p}} + \frac{\partial\mathbf{f}}{\partial\mathbf{p}} = \frac{\partial\mathbf{f}}{\partial\mathbf{x}}\hat{\mathbf{x}} + \frac{\partial\mathbf{f}}{\partial\mathbf{p}}.$$

3.5.3 Integral method

Integral method is a special method for transient analysis. Since in circuit, we are sometimes interested in a function which is an integral expression on time, like:

$$\mathbf{F}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) = \int_0^T \mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})dt.$$

Therefore the sensitivity for such function is:

$$\begin{aligned}
\frac{d\mathbf{F}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{d\mathbf{p}} &= \int_0^T \frac{d\mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{d\mathbf{p}}dt,\\
\Rightarrow \frac{d\mathbf{F}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{d\mathbf{p}} &= \int_0^T \left(\frac{\partial\mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial\mathbf{x}}\hat{\mathbf{x}} + \frac{\partial\mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial\mathbf{p}}\right)dt, \qquad (3.18)
\end{aligned}$$

We don't like the term $\hat{\mathbf{x}}$ inside, which means we should solve the sensitivity problem by direct method firstly. To reduce this term, we recall the DAE here,

$$\begin{aligned}\frac{d}{dt}\frac{d\mathbf{q}(\mathbf{x},\mathbf{p})}{d\mathbf{p}}+\frac{d\mathbf{j}(\mathbf{x},\mathbf{p})}{d\mathbf{p}} &= \frac{d\mathbf{s}(t,\mathbf{p})}{d\mathbf{p}},\\ \Rightarrow \int_0^T \lambda(t)\left(\frac{d}{dt}\frac{d\mathbf{q}(\mathbf{x},\mathbf{p})}{d\mathbf{p}}+\frac{d\mathbf{j}(\mathbf{x},\mathbf{p})}{d\mathbf{p}}-\frac{d\mathbf{s}(t,\mathbf{p})}{d\mathbf{p}}\right)dt &= \mathbf{0}.\end{aligned} \tag{3.19}$$

for any λ of size $nx \times mx$. For the first term, we have:

$$\int_0^T \lambda(t)\frac{d}{dt}\frac{d\mathbf{q}(\mathbf{x},\mathbf{p})}{d\mathbf{p}}dt = \lambda(t)\left.\frac{d\mathbf{q}(\mathbf{x},\mathbf{p})}{d\mathbf{p}}\right|_0^T - \int_0^T \frac{d\lambda(t)}{dt}\frac{d\mathbf{q}(\mathbf{x},\mathbf{p})}{d\mathbf{p}}dt. \tag{3.20}$$

Substituting Eq. (3.20) into Eq. (3.19), we have:

$$\begin{aligned}\lambda(t)\left.\frac{d\mathbf{q}(\mathbf{x},\mathbf{p})}{d\mathbf{p}}\right|_0^T+\int_0^T\left(-\frac{d\lambda(t)}{dt}\frac{d\mathbf{q}(\mathbf{x},\mathbf{p})}{d\mathbf{p}}+\lambda(t)\frac{d\mathbf{j}(\mathbf{x},\mathbf{p})}{d\mathbf{p}}-\lambda(t)\frac{d\mathbf{s}(t,\mathbf{p})}{d\mathbf{p}}\right)dt &= \mathbf{0}.\\ \Rightarrow \lambda(t)\left.\frac{d\mathbf{q}(\mathbf{x},\mathbf{p})}{d\mathbf{p}}\right|_0^T+\int_0^T\left(\left(\lambda(t)\mathcal{G}-\frac{d\lambda(t)}{dt}\mathcal{C}\right)\hat{\mathbf{x}}+\left(\lambda(t)\frac{\partial\mathbf{j}(\mathbf{x},\mathbf{p})}{\partial\mathbf{p}}\right.\right.&\\ \left.\left.-\frac{d\lambda(t)}{dt}\frac{\partial\mathbf{q}(\mathbf{x},\mathbf{p})}{\partial\mathbf{p}}-\lambda(t)\frac{d\mathbf{s}(t,\mathbf{p})}{d\mathbf{p}}\right)\right)dt &= \mathbf{0}.\\ \Rightarrow \int_0^T\left(\lambda(t)\mathcal{G}-\frac{d\lambda(t)}{dt}\mathcal{C}\right)\hat{\mathbf{x}}dt = -\lambda(t)\left.\left(\mathcal{C}\hat{\mathbf{x}}+\frac{\partial\mathbf{q}(\mathbf{x},\mathbf{p})}{\partial\mathbf{p}}\right)\right|_0^T-\int_0^T\left(\lambda(t)\frac{\partial\mathbf{j}(\mathbf{x},\mathbf{p})}{\partial\mathbf{p}}\right.&\\ \left.-\frac{d\lambda(t)}{dt}\frac{\partial\mathbf{q}(\mathbf{x},\mathbf{p})}{\partial\mathbf{p}}-\lambda(t)\frac{d\mathbf{s}(t,\mathbf{p})}{d\mathbf{p}}\right)dt.&\end{aligned} \tag{3.21}$$

Comparing Eq. (3.18) and Eq. (3.21), we find that if we set the $\lambda(t)$ satisfying the below equation:

$$\frac{\partial\mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial\mathbf{x}} = \lambda(t)\mathcal{G}-\frac{d\lambda(t)}{dt}\mathcal{C}. \tag{3.22}$$

We can eliminate the $\hat{\mathbf{x}}$ in Eq. (3.18) for $0 < t < T$, only the boundary value $\hat{\mathbf{x}}(0)$ and $\hat{\mathbf{x}}(T)$ remain.

$$\begin{aligned}\frac{d\mathbf{F}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{d\mathbf{p}} = -\lambda(t)\left.\left(\mathcal{C}\hat{\mathbf{x}}+\frac{\partial\mathbf{q}(\mathbf{x},\mathbf{p})}{\partial\mathbf{p}}\right)\right|_0^T+\int_0^T\left(\frac{d\lambda(t)}{dt}\frac{\partial\mathbf{q}(\mathbf{x},\mathbf{p})}{\partial\mathbf{p}}\right.&\\ \left.+\frac{\partial\mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial\mathbf{p}}+\lambda(t)\left(\frac{d\mathbf{s}(t,\mathbf{p})}{d\mathbf{p}}-\frac{\partial\mathbf{j}(\mathbf{x},\mathbf{p})}{\partial\mathbf{p}}\right)\right)dt.&\end{aligned} \tag{3.23}$$

Looking at the first term at the right-hand side of the Eq. (3.23), we have:

$$-\lambda(t)\left.\left(\mathcal{C}\hat{\mathbf{x}}+\frac{\partial\mathbf{q}(\mathbf{x},\mathbf{p})}{\partial\mathbf{p}}\right)\right|_0^T = \lambda(0)\left(\mathcal{C}_0\hat{\mathbf{x}}_0+\frac{\partial\mathbf{q}(\mathbf{x}_0,\mathbf{p})}{\partial\mathbf{p}}\right)-\lambda(T)\left(\mathcal{C}_T\hat{\mathbf{x}}_T+\frac{\partial\mathbf{q}(\mathbf{x}_T,\mathbf{p})}{\partial\mathbf{p}}\right). \tag{3.24}$$

Analyzing the above Eq. (3.24), we should think about how to get the terms of λ, $\mathcal{C}$, $\hat{\mathbf{x}}$, $\frac{\partial \mathbf{q}}{\partial \mathbf{p}}$ at $t = T$ and $t = 0$ respectively. We already know that at $t = 0$ the terms of $\mathcal{C}$, $\hat{\mathbf{x}}$, $\frac{\partial \mathbf{q}}{\partial \mathbf{p}}$ are easy to be made out from DC condition(t=0). But for those terms of $t = T$, we may have to calculate them time-step by time-step, which will cost a lot of storage and time. So here we use a little trick to avoid calculating those terms of $t = T$: we always choose $\lambda(T) = 0$, which implies in Eq. (3.22), the start point is fixed to $\lambda(T) = 0$, calculating inverse time-step to get the DC condition(t=0). Problem of Eq. (3.20) together with $\lambda(T) = 0$, as a “final" value problem that can be integrated backward in time.

With this choice, the sensitivity equation for function $\mathbf{F}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})$ is expressed as:

$$\begin{aligned}\frac{d\mathbf{F}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{d\mathbf{p}} = \lambda(0)\left(\mathcal{C}_0\hat{\mathbf{x}}_0 + \frac{\partial \mathbf{q}(\mathbf{x}_0,\mathbf{p})}{\partial \mathbf{p}}\right) + \int_0^T \left(\frac{d\lambda(t)}{dt}\frac{\partial \mathbf{q}(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}}\right.\\ \left. + \frac{\partial \mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial \mathbf{p}} + \lambda(t)\left(\frac{d\mathbf{s}(t,\mathbf{p})}{d\mathbf{p}} - \frac{\partial \mathbf{j}(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}}\right)\right) dt.\end{aligned} \tag{3.25}$$

To calculate the integral part of Eq. (3.25), we use Trapezoidal rule:

$$\begin{aligned}\int_0^T \left(\frac{d\lambda(t)}{dt}\frac{\partial \mathbf{q}(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial \mathbf{p}} + \lambda(t)\left(\frac{d\mathbf{s}(t,\mathbf{p})}{d\mathbf{p}} - \frac{\partial \mathbf{j}(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}}\right)\right) dt =\\ \sum_{r=0}^{n+1}\frac{\tilde{t}_{r+1}-\tilde{t}_r}{2}\left[\left(\frac{d\lambda(\tilde{t}_r)}{dt}\frac{\partial \mathbf{q}(\mathbf{x}_r,\mathbf{p})}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}(\mathbf{x}(t_r,\mathbf{p}),\mathbf{p})}{\partial \mathbf{p}} + \lambda(\tilde{t}_r)\left(\frac{d\mathbf{s}(\tilde{t}_r,\mathbf{p})}{d\mathbf{p}} - \frac{\partial \mathbf{j}(\mathbf{x}_r,\mathbf{p})}{\partial \mathbf{p}}\right)\right)\right.\\ \left. + \left(\frac{d\lambda(\tilde{t}_{r+1})}{dt}\frac{\partial \mathbf{q}(\mathbf{x}_{r+1},\mathbf{p})}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}(\mathbf{x}(t_{r+1},\mathbf{p}),\mathbf{p})}{\partial \mathbf{p}} + \lambda(\tilde{t}_{r+1})\left(\frac{d\mathbf{s}(\tilde{t}_{r+1},\mathbf{p})}{d\mathbf{p}} - \frac{\partial \mathbf{j}(\mathbf{x}_{r+1},\mathbf{p})}{\partial \mathbf{p}}\right)\right)\right].\end{aligned} \tag{3.26}$$

The error of Trapezoidal rule is proportional to the $\frac{1}{n}$, so if we choose n is large enough, the above approximation is reasonably considered as an solution accurate enough.

To get $\lambda(t)$, we consider Eq. (3.22), and we choose an inverse time-step for BDF,

$$\begin{aligned}&\frac{\partial \mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial \mathbf{x}} = \left(\lambda(t)\mathcal{G} - \frac{d\lambda(t)}{dt}\mathcal{C}\right),\\ \Rightarrow\ &\frac{\partial \mathbf{f}(\mathbf{x}(t_r,\mathbf{p}),\mathbf{p})}{\partial \mathbf{x}} = \left(\lambda(\tilde{t}_r)\mathcal{G}_r - \frac{\lambda(\tilde{t}_r)-\lambda(\tilde{t}_{r+1})}{-\triangle t}\mathcal{C}_r\right),\\ \Rightarrow\ &\lambda(\tilde{t}_r) = \left(\frac{\partial \mathbf{f}(\mathbf{x}(t_r,\mathbf{p}),\mathbf{p})}{\partial \mathbf{x}} + \lambda(\tilde{t}_{r+1})\frac{\mathcal{C}_r}{\triangle t}\right)\left(\frac{\mathcal{C}_r}{\triangle t} + \mathcal{G}_r\right)^{-1}.\end{aligned} \tag{3.27}$$

The start point is set as what we have discussed: $\lambda(T) = 0$. Calculating by inverse time-steps, we will get $\lambda(0)$ at the end.

Looking back on the Eq. (3.25), we still need to get the approximation of the first-differentiation of λ. Based on what we have got the approximation solution of λ for each time-step, we use the below formulas to get the first-differentiation of λ, which we can find that it also effects on error.

$$\begin{cases} \frac{d\lambda(\tilde{t}_0)}{dt} = \frac{\lambda(\tilde{t}_1 - \tilde{t}_0)}{\triangle t} & \text{when } t = 0, \text{ using FDF} \\ \frac{d\lambda(\tilde{t}_r)}{dt} = \frac{\lambda(\tilde{t}_{r+1} - \tilde{t}_{r-1})}{2\triangle t} & \text{when } t \in (0\ T), \text{ using CDF} \\ \frac{d\lambda(\tilde{t}_{n+1})}{dt} = \frac{\lambda(\tilde{t}_{n+1} - \tilde{t}_n)}{\triangle t} & \text{when } t = T, \text{ using BDF} \end{cases} \tag{3.28}$$

Recalling Eq. (3.25), we can find the error of the this equation consists of two parts: one from the numerical solutions of $\frac{d\lambda(t)}{dt}$ and one from the integral error of Eq. (3.26). Here we try to implement a new approach whose goal is to get a more accurate solution. In this problem, we trying to make the term $\frac{d\lambda(t)}{dt}\frac{\partial \mathbf{q}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial \mathbf{p}}$ equals zero, we already know $\frac{d\lambda(t)}{dt}$ doesn't equal zero basing on the above analysis, so we should find a way to make $\frac{\partial \mathbf{q}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial \mathbf{p}}$ equals zero. This implementation is not a sucessful approach, but it gives us some insights to think about how to minimize the error of the integral method and the extension integral method as well.

Recalling the general DAE:

$$\mathbf{j}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) + \frac{d\mathbf{q}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{dt} = \mathbf{s}(t,p).$$

We know that $\mathbf{q}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})$ explicitly depends on $\mathbf{x}$ and $\mathbf{p}$, actually implicitly depends on t and $\mathbf{p}$, so we write its implicit expression as:

$$\mathbf{y}(t,\mathbf{p}) = \mathbf{q}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}).$$

The general DAE can be rewritten as:

$$\left[\begin{array}{c} \mathbf{j}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) \\ \mathbf{y}(t,\mathbf{p}) - \mathbf{q}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) \end{array}\right] + \frac{d}{dt}\left[\begin{array}{c} \mathbf{y}(t,\mathbf{p}) \\ 0 \end{array}\right] = \left[\begin{array}{c} \mathbf{s}(t,\mathbf{p}) \\ 0 \end{array}\right]. \tag{3.29}$$

We redefine the unknowns are:

$$\tilde{\mathbf{x}}(t,\mathbf{p}) = \left(\begin{array}{c} \mathbf{x}(t,\mathbf{p}) \\ \mathbf{y}(t,\mathbf{p}) \end{array}\right),$$

Also the matrices of DAE in Eq. (3.29) are redefined as:

$$\tilde{\mathbf{j}}(\tilde{\mathbf{x}}(t,\mathbf{p}),\mathbf{p}) = \left[\begin{array}{c} \mathbf{j}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) \\ \mathbf{y}(t,\mathbf{p}) - \mathbf{q}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) \end{array}\right],\ \tilde{\mathbf{q}}(\tilde{\mathbf{x}}(t,\mathbf{p})) = \left[\begin{array}{c} \mathbf{y}(t,\mathbf{p}) \\ 0 \end{array}\right],\ \tilde{\mathbf{s}}(t,\mathbf{p}) = \left[\begin{array}{c} \mathbf{s}(t,\mathbf{p}) \\ 0 \end{array}\right]. \tag{3.30}$$

Therefore we get a new DAE for the new unknown:

$$\tilde{\mathbf{j}}(\tilde{\mathbf{x}}(t,\mathbf{p}),\mathbf{p}) + \frac{d\tilde{\mathbf{q}}(\tilde{\mathbf{x}}(t,\mathbf{p}))}{dt} = \tilde{\mathbf{s}}(t,\mathbf{p}).$$

The corresponding sensitivity equation:

$$\frac{d\tilde{\mathbf{j}}(\tilde{\mathbf{x}}(t,\mathbf{p}),\mathbf{p})}{d\mathbf{p}} + \frac{d}{dt}\frac{d\tilde{\mathbf{q}}(\tilde{\mathbf{x}}(t,\mathbf{p}))}{d\mathbf{p}} = \frac{\partial \tilde{\mathbf{s}}(t,\mathbf{p})}{\partial \mathbf{p}}. \tag{3.31}$$

Now we assume the new notation is $\tilde{\lambda}(t)$, we have:

$$
\begin{aligned}
&\int_0^T \tilde{\lambda}(t) \left(\frac{d\tilde{\mathbf{j}}(\tilde{\mathbf{x}}(t,\mathbf{p}),\mathbf{p})}{d\mathbf{p}} + \frac{d}{dt}\frac{d\tilde{\mathbf{q}}(\tilde{\mathbf{x}}(t,\mathbf{p}))}{d\mathbf{p}} - \frac{\partial \tilde{\mathbf{s}}(t,\mathbf{p})}{\partial \mathbf{p}} \right) dt = \mathbf{0}, \\
\Rightarrow \quad & \left[\tilde{\lambda}(t) \frac{d\tilde{\mathbf{q}}}{d\mathbf{p}} \right]\Bigg|_0^T + \int_0^T \left[-\frac{d\tilde{\lambda}(t)}{dt}\frac{d\tilde{\mathbf{q}}}{d\mathbf{p}} + \tilde{\lambda}(t) \left(\frac{d\tilde{\mathbf{j}}}{d\mathbf{p}} - \frac{\partial \tilde{\mathbf{s}}}{\partial \mathbf{p}} \right) \right] dt = \mathbf{0}, \\
\Rightarrow \quad & \left[\tilde{\lambda}(t) \left(\frac{\partial \tilde{\mathbf{q}}}{\partial \tilde{\mathbf{x}}}\hat{\tilde{\mathbf{x}}} + \frac{\partial \tilde{\mathbf{q}}}{\partial \mathbf{p}} \right) \right]\Bigg|_0^T + \int_0^T \left[-\frac{d\tilde{\lambda}(t)}{dt}\frac{\partial \tilde{\mathbf{q}}}{\partial \tilde{\mathbf{x}}}\hat{\tilde{\mathbf{x}}} + \tilde{\lambda}(t) \left(\frac{\partial \tilde{\mathbf{j}}}{\partial \tilde{\mathbf{x}}}\hat{\tilde{\mathbf{x}}} + \frac{\partial \tilde{\mathbf{j}}}{\partial \mathbf{p}} - \frac{\partial \tilde{\mathbf{s}}}{\partial \mathbf{p}} \right) \right] dt = \mathbf{0}, \\
\Rightarrow \quad & \left[\tilde{\lambda}(t) \left(\tilde{\mathcal{C}}\hat{\tilde{\mathbf{x}}} + \frac{\partial \tilde{\mathbf{q}}}{\partial \mathbf{p}} \right) \right]\Bigg|_0^T + \int_0^T \left[\left(-\frac{d\tilde{\lambda}(t)}{dt}\tilde{\mathcal{C}} + \tilde{\lambda}(t)\tilde{\mathcal{G}} \right) \hat{\tilde{\mathbf{x}}} + \tilde{\lambda}(t) \left(\frac{\partial \tilde{\mathbf{j}}}{\partial \mathbf{p}} - \frac{\partial \tilde{\mathbf{s}}}{\partial \mathbf{p}} \right) \right] dt = \mathbf{0}.
\end{aligned}
\tag{3.32}
$$

We use the same trick to choose $\tilde{\lambda}(T) = 0$, so the Eq. (3.32) turns into:

$$
\left(-\tilde{\lambda}(0)\tilde{\mathcal{C}}_0\hat{\tilde{\mathbf{x}}}_0 - \frac{\partial \tilde{\mathbf{q}}_0}{\partial \mathbf{p}} \right) + \int_0^T \left[\left(-\frac{d\tilde{\lambda}(t)}{dt}\tilde{\mathcal{C}} + \tilde{\lambda}(t)\tilde{\mathcal{G}} \right) \hat{\tilde{\mathbf{x}}} + \tilde{\lambda}(t) \left(\frac{\partial \tilde{\mathbf{j}}}{\partial \mathbf{p}} - \frac{\partial \tilde{\mathbf{s}}}{\partial \mathbf{p}} \right) \right] dt = \mathbf{0}. \tag{3.33}
$$

Of course the interested circuit function should be the same, but it is written into another expression:

$$
\tilde{\mathbf{F}}(\tilde{\mathbf{x}}(t,\mathbf{p}),\mathbf{p}) = \mathbf{F}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}).
$$

The sensitivity equation also turns into another expression:

$$
\frac{d\tilde{\mathbf{F}}(\tilde{\mathbf{x}}(t,\mathbf{p}),\mathbf{p})}{d\mathbf{p}} = \int_0^T \left(\frac{\partial \tilde{\mathbf{F}}(\tilde{\mathbf{x}}(t,\mathbf{p}),\mathbf{p})}{\partial \tilde{\mathbf{x}}}\hat{\tilde{\mathbf{x}}} + \frac{\partial \tilde{\mathbf{F}}(\tilde{\mathbf{x}}(t,\mathbf{p}),\mathbf{p})}{\partial \mathbf{p}} \right) dt. \tag{3.34}
$$

If we choose:

$$
\frac{\partial \tilde{\mathbf{F}}(\tilde{\mathbf{x}}(t,\mathbf{p}),\mathbf{p})}{\partial \tilde{\mathbf{x}}} = -\frac{d\tilde{\lambda}(t)}{dt}\tilde{\mathcal{C}} + \tilde{\lambda}(t)\tilde{\mathcal{G}}. \tag{3.35}
$$

The Eq. (3.34) can be written as:

$$
\frac{d\tilde{\mathbf{F}}(\tilde{\mathbf{x}}(t,\mathbf{p}),\mathbf{p})}{d\mathbf{p}} = \left(\tilde{\lambda}(0)\tilde{\mathcal{C}}_0\hat{\tilde{\mathbf{x}}}_0 + \frac{\partial \tilde{\mathbf{q}}_0}{\partial \mathbf{p}} \right) + \int_0^T \left(\frac{\partial \tilde{\mathbf{F}}}{\partial \mathbf{p}} - \tilde{\lambda}(t) \left(\frac{\partial \tilde{\mathbf{j}}}{\partial \mathbf{p}} - \frac{\partial \tilde{\mathbf{s}}}{\partial \mathbf{p}} \right) \right) dt. \tag{3.36}
$$

From this equation, we can find no differential term $\frac{d\tilde{\lambda}(t)}{dt}$ any more. For the Eq. (3.35), the implement is given as:

$$
\frac{\partial \tilde{\mathbf{F}}}{\partial \tilde{\mathbf{x}}} = \begin{pmatrix} \frac{\partial \mathbf{F}}{\partial \mathbf{x}} & \mathbf{0} \end{pmatrix}. \tag{3.37}
$$

From Eq. (3.30), we have:

$$
\tilde{\mathcal{C}}(\tilde{\mathbf{x}}(t,\mathbf{p})) = \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \tilde{\mathcal{G}}(\tilde{\mathbf{x}}(t,\mathbf{p}),\mathbf{p}) = \begin{pmatrix} \mathcal{G}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) & \mathbf{0} \\ -\mathcal{C}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) & \mathbf{I} \end{pmatrix}. \tag{3.38}
$$

Now we investigate $\tilde{\lambda}(t)$ further, if we assume it as:

$$\tilde{\lambda}(t) = \begin{pmatrix} \lambda(t) & \mu(t) \end{pmatrix}. \tag{3.39}$$

Substituting Eq. (3.37), Eq. (3.38) and Eq. (3.39) into Eq. (3.35), the luxury equation can be expressed as:

$$\begin{aligned} \begin{pmatrix} \frac{\partial \mathbf{F}}{\partial \mathbf{x}} & \mathbf{0} \end{pmatrix} &= -\begin{pmatrix} \frac{d\lambda}{dt} & \frac{d\mu}{dt} \end{pmatrix} \begin{pmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \begin{pmatrix} \lambda & \mu \end{pmatrix} \begin{pmatrix} \mathcal{G} & \mathbf{0} \\ -\mathcal{C} & \mathbf{I} \end{pmatrix}, \\ \Rightarrow \frac{\partial \mathbf{F}}{\partial \mathbf{x}} &= \lambda \mathcal{G} - \mu \mathcal{C} \quad and \quad \mu = \frac{d\lambda}{dt}. \end{aligned}$$

It is easy to find out that the above equations are the same as the Eq. (3.22), so we can't avoid the term $\frac{\partial \mathbf{q}}{\partial \mathbf{p}}$ at the end. On the other hand, these two equations can be considered as a DAE as well, but with higher index than the original circuit equation because of $\mu = \frac{d\lambda}{dt}$.

3.5.4 Extension integral method

If the interested circuit function $\mathbf{f}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})$ is not an integral function over time, but just a circuit function at a specific time, we want to reuse the result of the integral method to get the sensitivity of $\mathbf{f}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})$.

The relation of $\mathbf{f}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})$ and $\mathbf{F}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})$ is given:

$$\begin{aligned} \mathbf{f}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) &= \frac{d}{dT} \int_0^T \mathbf{f}(\mathbf{x}(t, \mathbf{p}), \mathbf{p}) dt = \frac{d\mathbf{F}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})}{dT}, \\ \Rightarrow \frac{d\mathbf{f}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})}{d\mathbf{p}} &= \frac{d}{dT} \left(\frac{d\mathbf{F}(\mathbf{x}(t, \mathbf{p}), \mathbf{p})}{d\mathbf{p}} \right). \end{aligned} \tag{3.40}$$

Here we should discuss more details about λ, which is known to be dependent on the choice of its initial point to start, i. e., the choice of T. So it can be said that λ is a function of T as well, which is expressed as:

$$\lambda = \lambda(t, T).$$

We always set $\lambda = 0$ at the start point $t = T$, as shown in section 3.5.3. Hence, for all T

$$\lambda(T, T) \equiv 0.$$

It is obviously found that the function λ is a function of T, specially at the condition of $t \equiv T$. But for every choice of start point T, the function is always set to be equal to zero. Therefore we have:

$$\begin{aligned} \frac{d\lambda(T,T)}{dT} = 0 \quad &\Rightarrow \quad \frac{\partial \lambda(T,T)}{\partial t} \frac{\partial t}{\partial T} + \frac{\partial \lambda(T,T)}{\partial T} = 0, \\ \Rightarrow \frac{\partial \lambda(T,T)}{\partial t} + \frac{\partial \lambda(T,T)}{\partial T} = 0 \quad &\Rightarrow \quad \frac{\partial \lambda(T,T)}{\partial T} = -\frac{\partial \lambda(T,T)}{\partial t}. \end{aligned} \tag{3.41}$$

This is an essential relation for the extension integral method as we will see next. After substituting Eq. (3.25) into Eq. (3.40), the sensitivity equation turns into:

$$\begin{aligned}\frac{d\mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{d\mathbf{p}} &= \frac{d\lambda(0,T)}{dT}\left(\mathcal{C}_0\hat{\mathbf{x}}_0 + \frac{\partial \mathbf{q}(\mathbf{x},0)}{\partial \mathbf{p}}\right) + \frac{d}{dT}\left(\int_0^T \left(\frac{d\lambda(t,T)}{dt}\frac{\partial \mathbf{q}(\mathbf{x},t)}{\partial \mathbf{p}}\right.\right. \\ &\left.\left. + \frac{\partial \mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial \mathbf{p}} + \lambda(t,T)\left(\frac{d\mathbf{s}(\mathbf{x},t)}{d\mathbf{p}} - \frac{\partial \mathbf{j}(\mathbf{x},t)}{\partial \mathbf{p}}\right)\right) dt\right). \qquad (3.42)\end{aligned}$$

Here a theorem is introduced (see [4], pp.108-109) for the integral term of the above equation.

THEOREM : If the functions ϕ and h are defined as below, we have:

$$\begin{aligned}\phi(x) \equiv \int_a^b f(x,u)du \quad &\Rightarrow \quad \phi'(x) = \int_a^b f_x(x,u)du, \\ h(x,y) \equiv \int_x^y f(x,u)du \quad &\Rightarrow \quad h_x(x,y) = \int_a^y f_x(x,u)du, \\ &\Rightarrow \quad h_y(x,y) = f(x,y).\end{aligned}$$

Let $x = x(t),\ y = y(t)$ be a parametrization, then:

$$\begin{aligned}\frac{d}{dt}h(x(t),y(t)) &= h_x(x,y)\frac{dx}{dt} + h_y(x,y)\frac{dy}{dt}, \\ &= \left[\int_a^y f_x(x,u)du\right]\frac{dx}{dt} + f(x,y)\frac{dy}{dt}.\end{aligned}$$

For the special choice $x(t) = t,\ y(t) = F(t) = F(x)$, we derive:

$$\frac{d}{dx}\int_a^{F(x)} f(x,u)du = \int_a^{F(x)} f_x(x,u)du + f(x,F(x))F'(x).$$

In particular, if $F(x) = x$, then:

$$\frac{d}{dx}\int_a^{x} f(x,u)du = \int_a^{x} f_x(x,u)du + f(x,x). \qquad (3.43)$$

¶¶

Following Eq. (3.43) of Theorem 3.5.4, the integral term can be calculated as the below:

$$\begin{aligned}&\frac{d}{dT}\left(\int_0^T \left(\frac{d\lambda(t,T)}{dt}\frac{\partial \mathbf{q}(\mathbf{x},t)}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial \mathbf{p}} + \lambda(t,T)\left(\frac{d\mathbf{s}(\mathbf{x},t)}{d\mathbf{p}} - \frac{\partial \mathbf{j}(\mathbf{x},t)}{\partial \mathbf{p}}\right)\right) dt\right) \\ =& \int_0^T \left(\frac{\partial}{\partial t}\left(\frac{\partial\lambda(t,T)}{\partial T}\right)\frac{\partial \mathbf{q}(\mathbf{x},t)}{\partial \mathbf{p}} + \frac{\partial\lambda(t,T)}{\partial T}\left(\frac{d\mathbf{s}(\mathbf{x},t)}{d\mathbf{p}} - \frac{\partial \mathbf{j}(\mathbf{x},t)}{\partial \mathbf{p}}\right)\right) dt \\ &+ \left(\frac{d\lambda(t,T)}{dt}\frac{\partial \mathbf{q}(\mathbf{x},t)}{\partial \mathbf{p}} + \frac{\partial \mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial \mathbf{p}}\right)\Bigg|_{t=T}\end{aligned}$$

Apart from $\lambda(t,T)$ and $\frac{d\lambda(t,T)}{dt}$, we now also need $\frac{\partial\lambda(t,T)}{\partial T}$ and $\frac{\partial}{\partial t}\left(\frac{\partial\lambda(t,T)}{\partial T}\right)$. So the problem how to determine each $\frac{\partial\lambda(t,T)}{\partial T}$ at each time-step in the domain $[0\ T]$ should be considered here. Recalling the luxury equation for the integral equation Eq. (3.22):

$$\begin{aligned}\frac{\partial\mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial\mathbf{x}} &= \lambda(t,T)\mathcal{G}-\frac{\partial\lambda(t,T)}{\partial t}\mathcal{C},\\ \Rightarrow\frac{\partial}{\partial T}\left(\frac{\partial\mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial\mathbf{x}}\right) &= \frac{\partial}{\partial T}\left(\lambda(t,T)\mathcal{G}\right)-\frac{\partial}{\partial T}\left(\frac{\partial\lambda(t,T)}{\partial t}\mathcal{C}\right),\\ \Rightarrow\mathbf{0} &= \frac{\partial\lambda(t,T)}{\partial T}\mathcal{G}-\frac{\partial^2\lambda(t,T)}{\partial t\partial T}\mathcal{C}.\end{aligned} \tag{3.44}$$

If we define $\mathcal{L}(t)=\frac{\partial\lambda(t,T)}{\partial T}$, the Eq. (3.44) becomes:

$$\mathcal{L}(t)\mathcal{G}-\frac{\partial\mathcal{L}(t)}{\partial t}\mathcal{C}=\mathbf{0}. \tag{3.45}$$

which is the luxury equation for extension integral method, looks very similar to Eq. (3.22).

From the essential relation we got in Eq. (3.41), we have the initial value of the Eq. (3.45):

$$\frac{\partial\lambda(T,T)}{\partial T}=-\frac{\partial\lambda(T,T)}{\partial t}\Rightarrow\mathcal{L}(T)=-\frac{\partial\lambda(T,T)}{\partial t}.$$

which also implies that the start point of function $\mathcal{L}(t)$ is: $t=T$, so we also use BDF for Eq. (3.45) by inverse time-step,

$$\begin{aligned}\mathcal{L}(\tilde{t}_r)\mathcal{G}_r-\frac{\mathcal{L}(\tilde{t}_r)-\mathcal{L}(\tilde{t}_{r+1})}{-\triangle t}\mathcal{C}_r &= \mathbf{0},\\ \Rightarrow\quad \mathcal{L}(\tilde{t}_r)=\mathcal{L}(\tilde{t}_{r+1})\frac{\mathcal{C}_r}{\triangle t}\left(\frac{\mathcal{C}_r}{\triangle t}+\mathcal{G}_r\right)^{-1}&.\end{aligned} \tag{3.46}$$

So the sensitivity equation can be expressed as:

$$\begin{aligned}\frac{d\mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{d\mathbf{p}} &= \mathcal{L}(0)\left(\mathcal{C}_0\hat{\mathbf{x}}_0+\frac{\partial\mathbf{q}(\mathbf{x},0)}{\partial\mathbf{p}}\right)+\left.\left(\frac{d\lambda(t,T)}{dt}\frac{\partial\mathbf{q}(\mathbf{x},t)}{\partial\mathbf{p}}+\frac{\partial\mathbf{f}(\mathbf{x}(t,\mathbf{p}),\mathbf{p})}{\partial\mathbf{p}}\right)\right|_{t=T}\\ &\quad+\int_0^T\left(\frac{d\mathcal{L}(t)}{dt}\frac{\partial\mathbf{q}(\mathbf{x},t)}{\partial\mathbf{p}}+\mathcal{L}(t)\left(\frac{d\mathbf{s}(\mathbf{x},t)}{d\mathbf{p}}-\frac{\partial\mathbf{j}(\mathbf{x},t)}{\partial\mathbf{p}}\right)\right)dt.\end{aligned} \tag{3.47}$$

As before, the final integral term can be approximated using the Trapezoidal rule.

Summarizing:

1. to find $\lambda(t)$ and $\frac{d\lambda}{dt}$, we have to solve Eq. (3.22) with $\lambda(T)=0$.
2. to find $\mathcal{L}(t)$ and $\frac{d\mathcal{L}}{dt}$, we have to solve Eq. (3.45) with $\mathcal{L}(T)=-\frac{\partial\lambda(T,T)}{\partial t}$.

In both problems, we can reuse the LU-decomposition of $\left(\frac{\mathcal{C}}{\triangle t}+\mathcal{G}\right)$ from the original time integration for the circuit solution. The finite differential approximation for $\frac{d\lambda}{dt}$, $\frac{d\mathcal{L}}{dt}$ and the Trapezoidal rule affect the accuracies.

For the four methods of transient analysis, we have a table to illustrate:

Property	DiffM	DM	AdjM	IntM	Extension IntM
Accuracy	+	+	+	+	-
Storage cost	-	-	-	+	+
Programming	+	+	+	-	-
Luxury equation	-	-	-	+	+

Table 3.2: “+" means “good"; “-" means “not so good"

3.5.5 Examples

Linear example

We use the same circuit topology as before, the parameter is set as: $\mathbf{p} = R_1$.

1. Direct method:

$$\begin{pmatrix} -i_{b1} + \frac{v_1 - v_2}{R_1} \\ -\frac{v_1 - v_2}{R_1} + \frac{v_2 - v_3}{R_2} \\ -\frac{v_2 - v_3}{R_2} + i_{b4} \\ v_1 \\ v_3 \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} 0 \\ Cv_2 \\ 0 \\ 0 \\ -Li_{b4} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ s(t) \\ 0 \end{pmatrix},$$

we have the related matrices:

$$\mathcal{G} = \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & -1 & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_2} & -\frac{1}{R_2} & 0 & 0 \\ 0 & -\frac{1}{R_2} & \frac{1}{R_2} & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \quad \mathcal{C} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -L \end{pmatrix}.$$

And

$$\frac{\partial \mathbf{j}}{\partial \mathbf{R_1}} = \begin{pmatrix} -\frac{v_1 - v_2}{R_1^2} \\ \frac{v_1 - v_2}{R_1^2} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \frac{\partial \mathbf{q}}{\partial R_1} = \mathbf{0}, \quad \frac{\partial \mathbf{s}}{\partial R_1} = \mathbf{0}.$$

Following Eq. (3.17), we have:

$$\hat{\mathbf{x}}_{r+1} = \left(\mathcal{G}_{r+1} + \frac{1}{\triangle t} \mathcal{C}_{r+1} \right)^{-1} \left(\frac{1}{\triangle t} \mathcal{C}_r \hat{\mathbf{x}}_r - \frac{\partial \mathbf{j}(\mathbf{x}_{r+1}, \tilde{t}_{r+1})}{\partial \mathbf{p}} \right).$$

2. Adjoint method: the interested functions are defined as before: $f_1 = i_{b3}v_{b3} = \frac{(v_2-v_3)^2}{R_2}$, $f_2 = i_{b1}v_1$. We have:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{pmatrix} 0 & \frac{2(v_2-v_3)}{R_2} & -\frac{2(v_2-v_3)}{R_2} & 0 & 0 \\ i_{b1} & 0 & 0 & v_1 & 0 \end{pmatrix}, \quad \frac{\partial \mathbf{f}}{\partial R_1} = \mathbf{0}.$$

So we have the sensitivity equation for the two functions is:

$$\frac{d\mathbf{f}}{dR_1} = \begin{pmatrix} 0 & \frac{2(v_2-v_3)}{R_2} & -\frac{2(v_2-v_3)}{R_2} & 0 & 0 \\ i_{b1} & 0 & 0 & v_1 & 0 \end{pmatrix} \hat{\mathbf{x}}.$$

3. Integral method: If we assume the integral functions are:

$$\begin{cases} F_1 = \int_0^T i_{b3}v_{b3}dt = \int_0^T \frac{(v_2-v_3)^2}{R_2}dt \\ F_2 = \int_0^T i_{b1}v_1 dt \end{cases} \Rightarrow \begin{cases} f_1 = \frac{(v_2-v_3)^2}{R_2} \\ f_2 = i_{b1}v_1 \end{cases}.$$

At first, we should use luxury equation to get λ at each time-step. Following Eq. (3.27), we have:

$$\lambda(\tilde{t}_r) = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \lambda(\tilde{t}_{r+1})\frac{\mathcal{C}_r}{\triangle t}\right)\left(\frac{\mathcal{C}_r}{\triangle t} + \mathcal{G}_r\right)^{-1},$$

the initial value for this equation is chosen as $\lambda(T) = 0$. Then the Eq. (3.28) is used to get the $\frac{d\lambda(\tilde{t}_r)}{dt}$, at final the Eqs. (3.25) and (3.26) are used to get the sensitivity solution $\frac{d\mathbf{F}}{d\mathbf{p}}$ for the integral functions.

4. Extension integral method:

The interested functions are assumed as:

$$\begin{cases} f_1 = \frac{(v_2-v_3)^2}{R_2} \\ f_2 = i_{b1}v_1 \end{cases} \Rightarrow \begin{cases} f_1 = \frac{d}{dT}\int_0^T \frac{(v_2-v_3)^2}{R_2}dt = \frac{dF_1}{dT} \\ f_2 = \frac{d}{dT}\int_0^T i_{b1}v_1 dt = \frac{dF_2}{dT} \end{cases}.$$

At first, we also need to use the luxury Eq. (3.46) to get $\mathcal{L}(\tilde{t}_r)$:

$$\mathcal{L}(\tilde{t}_r) = \mathcal{L}(\tilde{t}_{r+1})\frac{\mathcal{C}_r}{\triangle t}\left(\frac{\mathcal{C}_r}{\triangle t} + \mathcal{G}_r\right)^{-1},$$

the initial value for this equation is chosen as: $\mathcal{L}(T) = -\frac{\partial \lambda(T,T)}{\partial t}$. Then the similar way as we used to in Eq. (3.28), we get the first-differentialtion of $\frac{d\mathcal{L}}{dt}$. At final, we should use the Eq. (3.47) to get the sensitivity solution $\frac{d\mathbf{f}}{d\mathbf{p}}$.

Nonlinear example

The element R_2 is a nonlinear component with the relation: $i_{b3} = v_{b3}^3 + 2v_{b3}^2 + v_{b3} = (v_2 - v_3)^3 + 2(v_2 - v_3)^2 + (v_2 - v_3)$.

1. Direct method:

$$\begin{pmatrix} -i_{b1} + \frac{v_1 - v_2}{R_1} \\ -\frac{v_1 - v_2}{R_1} + ((v_2 - v_3)^3 + 2(v_2 - v_3)^2 + (v_2 - v_3)) \\ -((v_2 - v_3)^3 + 2(v_2 - v_3)^2 + (v_2 - v_3)) + i_{b4} \\ v_1 \\ v_3 \end{pmatrix} + \frac{d}{dt} \begin{pmatrix} 0 \\ Cv_2 \\ 0 \\ 0 \\ -Li_{b4} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ s(t) \\ 0 \end{pmatrix},$$

we have the related matrices:

$$\mathcal{G} = \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & -1 & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} + (3v_2^2 - 6v_2v_3 + 3v_3^2 + 4v_2 - 4v_3 + 1) & -((3v_2^2 - 6v_2v_3 + 3v_3^2 + 4v_2 - 4v_3 + 1)) & 0 & 0 \\ 0 & -(3v_2^2 - 6v_2v_3 + 3v_3^2 + 4v_2 - 4v_3 + 1) & (3v_2^2 - 6v_2v_3 + 3v_3^2 + 4v_2 - 4v_3 + 1) & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix},$$

$$\mathcal{C} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & C & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -L \end{pmatrix}.$$

And

$$\frac{\partial \mathbf{j}}{\partial \mathbf{R_1}} = \begin{pmatrix} -\frac{v_1 - v_2}{R_1^2} \\ \frac{v_1 - v_2}{R_1^2} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \frac{\partial \mathbf{q}}{\partial R_1} = \mathbf{0}, \quad \frac{\partial \mathbf{s}}{\partial R_1} = \mathbf{0}.$$

Following Eq. (3.17), we have:

$$\hat{\mathbf{x}}_{r+1} = \left(\mathcal{G}_{r+1} + \frac{1}{\triangle t} \mathcal{C}_{r+1} \right)^{-1} \left(\frac{1}{\triangle t} \mathcal{C}_r \hat{\mathbf{x}}_r - \frac{\partial \mathbf{j}(\mathbf{x}_{r+1}, \tilde{t}_{r+1})}{\partial \mathbf{p}} \right).$$

2. Adjoint method: the interested functions are defined as before: $f_1 = i_{b3}v_{b3} = \frac{(v_2 - v_3)^2}{R_2}$, $f_2 = i_{b1}v_1$. We have:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{pmatrix} 0 & \alpha & -\alpha & 0 & 0 \\ i_{b1} & 0 & 0 & v_1 & 0 \end{pmatrix}, \quad \frac{\partial \mathbf{f}}{\partial R_1} = \mathbf{0}.$$

where $\alpha = 4(v_2 - v_3)^3 + 6(v_2 - v_3)^2 + 2(v_2 - v_3)$. So we have the sensitivity equation for the two functions is:

$$\frac{d\mathbf{f}}{dR_1} = \begin{pmatrix} 0 & \frac{2(v_2-v_3)}{R_2} & -\frac{2(v_2-v_3)}{R_2} & 0 & 0 \\ i_{b1} & 0 & 0 & v_1 & 0 \end{pmatrix} \hat{\mathbf{x}}.$$

3. Integral method: If we assume the integral functions are:

$$\begin{cases} F_1 = \int_0^T (i_{b3} v_{b3} dt = \int_0^T ((v_2 - v_3)^4 + 2(v_2 - v_3)^3 + (v_2 - v_3)^2) dt \\ F_2 = \int_0^T i_{b1} v_1 dt \end{cases}$$

$$\Rightarrow \begin{cases} f_1 = (v_2 - v_3)^4 + 2(v_2 - v_3)^3 + (v_2 - v_3)^2 \\ f_2 = i_{b1} v_1 \end{cases} .$$

At first, we should use luxury equation to get λ at each time-step. Following Eq. (3.27), we have:

$$\lambda(\tilde{t}_r) = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \lambda(\tilde{t}_{r+1}) \frac{\mathcal{C}_r}{\triangle t} \right) \left(\frac{\mathcal{C}_r}{\triangle t} + \mathcal{G}_r \right)^{-1},$$

the initial value for this equation is chosen as $\lambda(T) = 0$. Then the Eq. (3.28) is used to get the $\frac{d\lambda(\tilde{t}_r)}{dt}$, at final the Eqs. (3.25) and (3.26) are used to get the sensitivity solution $\frac{d\mathbf{F}}{d\mathbf{p}}$ for the integral functions.

4. Extension integral method:

The interested functions are assumed as:

$$\begin{cases} f_1 = \frac{(v_2-v_3)^2}{R_2} \\ f_2 = i_{b1} v_1 \end{cases} \Rightarrow \begin{cases} f_1 = \frac{d}{dT} \int_0^T \frac{(v_2-v_3)^2}{R_2} dt = \frac{dF_1}{dT} \\ f_2 = \frac{d}{dT} \int_0^T i_{b1} v_1 dt = \frac{dF_2}{dT} \end{cases} .$$

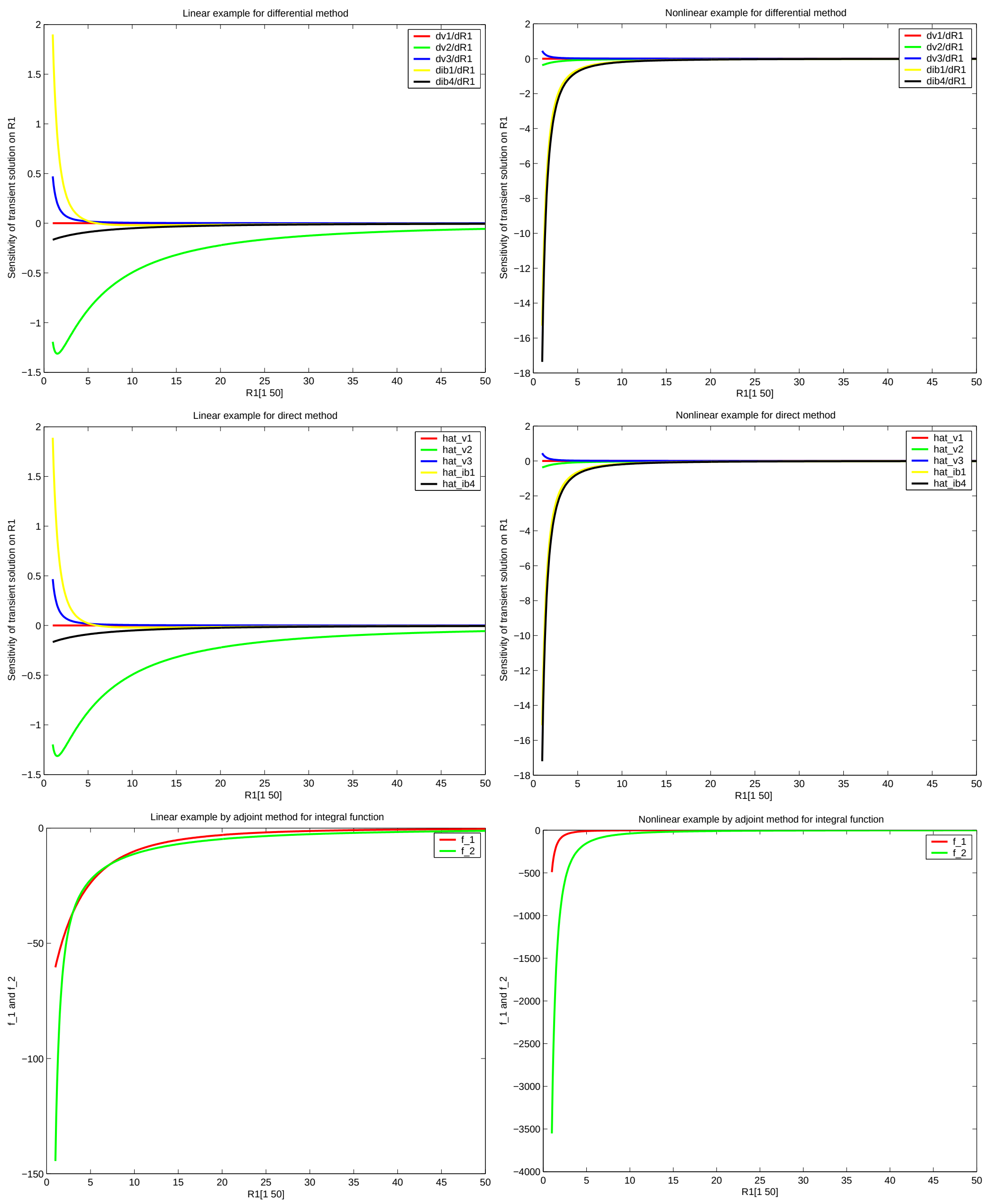
At first, we also need to use the luxury Eq. (3.46) to get $\mathcal{L}(\tilde{t}_r)$:

$$\mathcal{L}(\tilde{t}_r) = \mathcal{L}(\tilde{t}_{r+1}) \frac{\mathcal{C}_r}{\triangle t} \left(\frac{\mathcal{C}_r}{\triangle t} + \mathcal{G}_r \right)^{-1},$$

the initial value for this equation is chosen as: $\mathcal{L}(T) = -\frac{\partial \lambda(T,T)}{\partial t}$. Then the similar way as we used to in Eq. (3.28), we get the first-differentialtion of $\frac{d\mathcal{L}}{dt}$. At final, we should use the Eq. (3.47) to get the sensitivity solution $\frac{d\mathbf{f}}{d\mathbf{p}}$.

Results

The results for linear and nonlinear examples are illustrated below,


Linear example for differential method
Sensitivity of transient solution on R1
R1[1 50]
dv1/dR1
dv2/dR1
dv3/dR1
dib1/dR1
dib4/dR1
Nonlinear example for differential method
Sensitivity of transient solution on R1
R1[1 50]
dv1/dR1
dv2/dR1
dv3/dR1
dib1/dR1
dib4/dR1
Linear example for direct method
Sensitivity of transient solution on R1
R1[1 50]
hat_v1
hat_v2
hat_v3
hat_ib1
hat_ib4
Nonlinear example for direct method
Sensitivity of transient solution on R1
R1[1 50]
hat_v1
hat_v2
hat_v3
hat_ib1
hat_ib4
Linear example by adjoint method for integral function
f_1 and f_2
R1[1 50]
f_1
f_2
Nonlinear example by adjoint method for integral function
f_1 and f_2
R1[1 50]
f_1
f_2

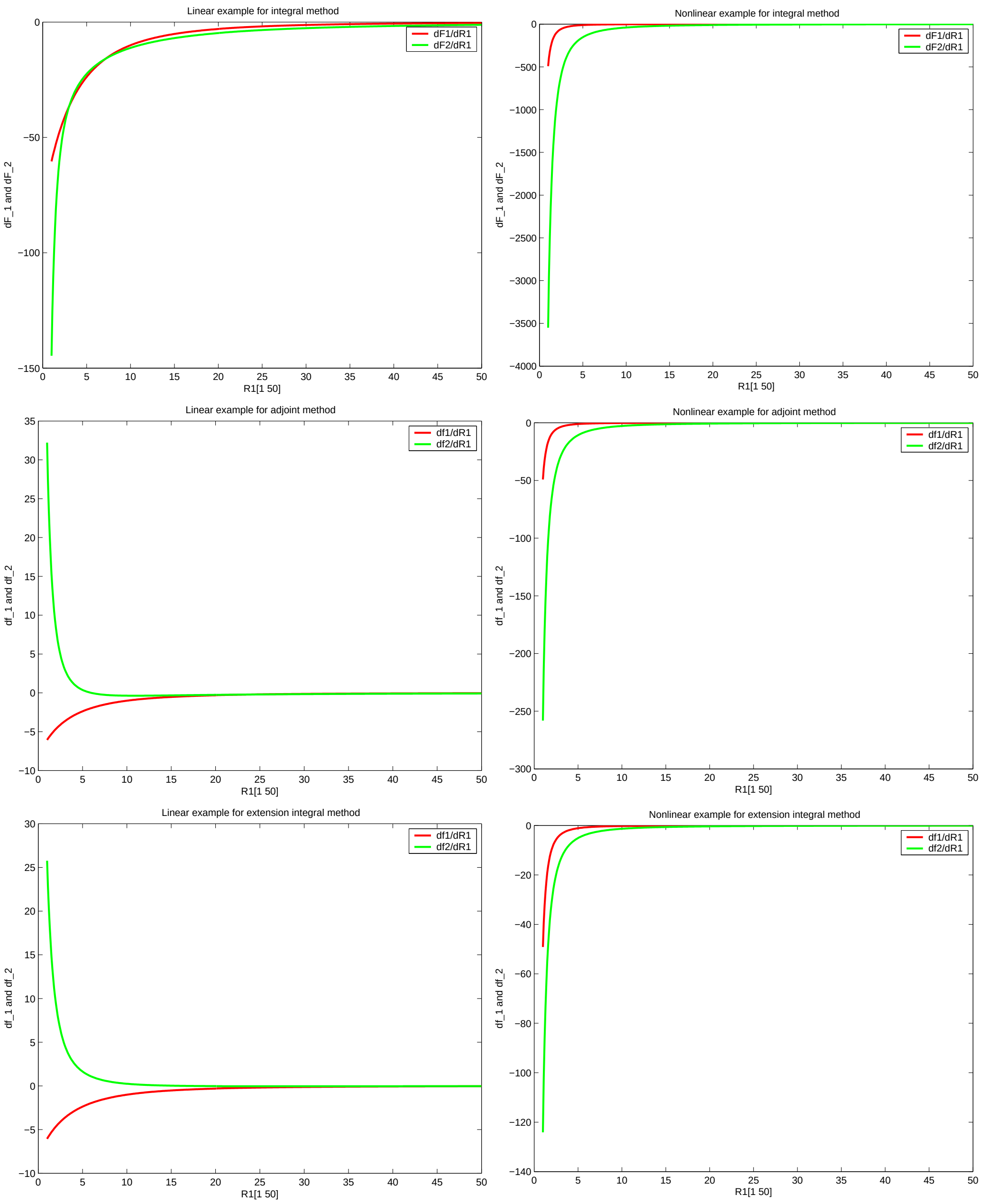


Figure 3.3: Transient sensitivity solution: $\mathbf{s}(t) = 20 + 5\sin 10t$

3.6 Fault analysis

The fault analysis assumes that the circuit topology is fixed. We restrict ourselves to the cases that the effect of all the elements in the circuit are linear elements. If some element are taken away from the circuit, the change for circuit response is significant. How to determine such change is the goal of the fault analysis.

The circuit equation is expressed as:

$$\mathbf{j}(\mathbf{x}, t) + \frac{d\mathbf{q}(\mathbf{x}, t)}{dt} = \mathbf{s}(t).$$

If the elements are all linear elements, the circuit equation also can be expressed as:

$$\begin{aligned} & \mathcal{G}\mathbf{x} + \tfrac{d}{dt}(\mathcal{C}\mathbf{x}) = \mathbf{s}(t), \\ \Rightarrow \quad & \mathcal{G}_{r+1}\mathbf{x}_{r+1} + \tfrac{\mathcal{C}_{r+1}\mathbf{x}_{r+1} - \mathcal{C}_r\mathbf{x}_r}{\triangle t} = \mathbf{s}(\tilde{t}_{r+1}), \\ \Rightarrow \quad & \mathbf{x}_{r+1} = \left(\mathcal{G}_{r+1} + \tfrac{\mathcal{C}_{r+1}}{\triangle t}\right)^{-1} \left(\mathbf{s}(\tilde{t}_{r+1}) + \tfrac{\mathcal{C}_r}{\triangle t}\mathbf{x}_r\right). \end{aligned} \tag{3.48}$$

If we define the term $\left(\mathcal{G} + \frac{\mathcal{C}}{\triangle t}\right)$ as matrix $\mathcal{M}$, the Eq. (3.48) turns into:

$$\mathbf{x}_{r+1} = (\mathcal{M}_{r+1})^{-1} \left(\mathbf{s}(\tilde{t}_{r+1}) + \frac{\mathcal{C}_r}{\triangle t}\mathbf{x}_r\right). \tag{3.49}$$

We know the unknowns of circuit equation consist of node voltages and branch currents. So the contribution of each basic electronic components in the circuit can be expressed as the below table:

Elements	Definition	$\triangle\mathcal{C}$	$\triangle\mathcal{G}$
R	$v = iR$		Ree^T
C	$q = Cv$	Cee^T	
L	$\varphi = Li$	$-L\tilde{e}\tilde{e}^T$	$e\tilde{e}^T + \tilde{e}e^T$
S	$S = E$		$e\tilde{e}^T + \tilde{e}e^T$

Table 3.3: Contributions of linear elements

Hence, R, C, L, S are *Resistor*, *Capacitor*, *Inductor*, *Source* respectively. e and $\tilde{e}$ are the vectors indicating the corresponding relation of node voltages and branch currents respectively. The components of vector e only consist of $1, \ -1$ and 0, respectively indicating the node voltages "entering", "leaving", and not crossing through the element; Meanwhile the components of vector $\tilde{e}$ only consist of 1 and 0, respectively indicating the branch currents across through and not across through element.

Now we illustrate how to determine the change of circuit response if a perturbation happen to the circuit. For simplicity, we assume that the perturbation is from a resistor, only having influence on matrix $\mathcal{M}$. The perturbation is symbolled as $\triangle\mathcal{M} = -Ree^T$ here, which means, the resistor being taken from the circuit.

$Shermann - Morrison - Woodbury(SMW)\ Formula$ should be introduced here:

$\mathbf{THEOREM}:$ Let P and Q be matrices of size $m \times n$ and $n \times m$ respectively; I_m and I_n be identity matrices of rank m and n. We assume that $(I_m - PQ)$ is non-singular. Then $(I_n - QP)$ is non-singular and more precisely:

$$(I_n - QP)^{-1} = I_n + Q(I_m - PQ)^{-1}P. \tag{3.50}$$

Furthermore, let $A,\ U,\ S$ and W be matrices of sizes $n \times n,\ n \times m,\ m \times m$ and $m \times n$ respectively. We assume that A and S are non-singular, and $I_m + SWA^{-1}U$ is non-singular. Then $A + USW$ is non-singular and more precisely:

$$(A + USW)^{-1} = A^{-1} - A^{-1}U(S^{-1} + WA^{-1}U)^{-1}WA^{-1}. \tag{3.51}$$

¶¶

Applying the SMW formula to the change of the matrix $\mathcal{M}$ after a perturbation $\triangle\mathcal{M}$:

$$\begin{aligned}
(\mathcal{M} + \triangle\mathcal{M})^{-1} &= (\mathcal{M} - Ree^T)^{-1} = (\mathcal{M} + e(-R)e^T)^{-1}, \\
&= \mathcal{M}^{-1} - \mathcal{M}^{-1}e\left(-\frac{1}{R} + e^T\mathcal{M}^{-1}e\right)^{-1} e^T\mathcal{M}^{-1}, \\
&= \left(I_{nx} - \mathcal{M}^{-1}e\left(-\frac{1}{R} + e^T\mathcal{M}^{-1}e\right)^{-1} e^T\right)\mathcal{M}^{-1}.
\end{aligned}$$

So the change of solution after perturbation is:

$$\begin{aligned}
\mathbf{x}_{r+1} + \triangle\mathbf{x}_{r+1} &= \left(I_{nx} - (\mathcal{M}_{r+1})^{-1}e\left(-\frac{1}{R} + e^T(\mathcal{M}_{r+1})^{-1}e\right)^{-1} e^T\right)(\mathcal{M}_{r+1})^{-1}\left(\mathbf{s}(t_{r+1}) + \frac{\mathcal{C}_r}{\triangle t}\mathbf{x}_r\right), \\
&= \left(I_{nx} - (\mathcal{M}_{r+1})^{-1}e\left(-\frac{1}{R} + e^T(\mathcal{M}_{r+1})^{-1}e\right)^{-1} e^T\right)\mathbf{x}_{r+1}, \\
\Rightarrow \triangle\mathbf{x}_{r+1} &= -(\mathcal{M}_{r+1})^{-1}e\left(-\frac{1}{R} + e^T(\mathcal{M}_{r+1})^{-1}e\right)^{-1} e^T\mathbf{x}_{r+1}.
\end{aligned} \tag{3.52}$$

Letting $R \to \infty$ means that the resistor becomes an "open". Letting $R \to 0$ means that the resistor becomes a "short".

3.6.1 Examples

We use the linear example as before,

$$
\underbrace{\begin{pmatrix} -i_{b_1} + \frac{v_1 - v_2}{R_1} \\ -\frac{v_1 - v_2}{R_1} + \frac{v_2 - v_3}{R_2} \\ -\frac{v_2 - v_3}{R_2} + i_{b_4} \\ v_1 \\ v_3 \end{pmatrix}}_{\mathbf{j}(\mathbf{x},t)} + \frac{d}{dt} \underbrace{\begin{pmatrix} 0 \\ Cv_2 \\ 0 \\ 0 \\ -Li_{b_4} \end{pmatrix}}_{\mathbf{q}(\mathbf{x},t)} = \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ V(t) \\ 0 \end{pmatrix}}_{s(t)}
$$

where $\mathbf{x} = (v_1 \;\; v_2 \;\; v_3 \;\; i_{b_1} \;\; i_{b_4})^T$.
We can write as

$$
\begin{aligned}
\mathbf{q}(\mathbf{x}, t) &= \mathbf{q}_c(\mathbf{x}, t) + \mathbf{q}_L(\mathbf{x}, t) \\
&= C \begin{pmatrix} 0 & & & & \\ & 1 & & & \\ & & 0 & & \\ & & & 0 & \\ & & & & 0 \end{pmatrix} \mathbf{x} + L \begin{pmatrix} 0 & & & & \\ & 0 & & & \\ & & 0 & & \\ & & & 0 & \\ & & & & -1 \end{pmatrix} \mathbf{x}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{j}(\mathbf{x}, t) &= \mathbf{j}_E(\mathbf{x}, t) + \mathbf{j}_{R_1}(\mathbf{x}, t) + \mathbf{j}_{R_2}(\mathbf{x}, t) + \mathbf{j}_L(\mathbf{x}, t) \\
&= \begin{pmatrix} 0 & & & -1 & \\ & 0 & & & \\ & & 0 & & \\ 1 & & & 0 & \\ & & & & 0 \end{pmatrix} \mathbf{x} + \frac{1}{R_1} \begin{pmatrix} 1 & -1 & & & \\ -1 & 1 & & & \\ & & 0 & & \\ & & & 0 & \\ & & & & 0 \end{pmatrix} \mathbf{x} \\
&\quad + \frac{1}{R_2} \begin{pmatrix} 0 & & & & \\ & 1 & -1 & & \\ & -1 & 1 & & \\ & & & 0 & \\ & & & & 0 \end{pmatrix} \mathbf{x} + \begin{pmatrix} 0 & & & & \\ & 0 & & & \\ & & 0 & & 1 \\ & & & 0 & \\ & & 1 & & 0 \end{pmatrix} \mathbf{x}
\end{aligned}
$$

So we can write the circuit equation as

$$
\mathcal{G}\mathbf{x} + \frac{d}{dt} (\mathcal{C}\mathbf{x}) = s(t)
$$

where

$$
\mathcal{G} = \begin{pmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & -1 & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_2} & -\frac{1}{R_2} & 0 & 0 \\ 0 & -\frac{1}{R_2} & \frac{1}{R_2} & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \quad \mathcal{C} = \begin{pmatrix} 0 & & & & \\ & C & & & \\ & & 0 & & \\ & & & 0 & \\ & & & & -L \end{pmatrix}
$$

If we take resistor R_2 away, the corresponding value will be considered as ZERO, so the perturbation will be

$$\triangle M = \left(-\frac{1}{R_2}\right)\begin{pmatrix} 0 & & & & \\ & 1 & -1 & & \\ & -1 & 1 & & \\ & & & 0 & \\ & & & & 0 \end{pmatrix} = \underbrace{\left(-\frac{1}{R_2}\right)}_{-g}\underbrace{\begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}}_{\mathbf{e}}\underbrace{\begin{pmatrix} 0 & 1 & -1 & 0 & 0 \end{pmatrix}}_{\mathbf{e}^T}$$

Using the Eq.(3.52)

$$\triangle \mathbf{x}_{i+1} = -M^{-1}\mathbf{e}(-\frac{1}{g} + \mathbf{e}^T M^{-1}\mathbf{e})^{-1}\mathbf{e}^T M^{-1}\left(s(t_{i+1}) + \frac{Z_i \mathbf{x}_i}{\triangle t}\right)$$

Chapter 4

Conclusions and future work

This thesis aims to investigate some approaches for the circuit optimization, using sensitivity analysis. The circuit simulation is the basement for such a sensitivity analysis. From what we have discussed in this thesis, we can get some conclusions like below:

1. The circuit equation can be acquired by nodal analysis, which describes the circuit topology efficiently.

2. The DC simulation is a fundament for AC, HB and transient simulation. NR method are used for DC simulation to get the solution.

   For AC simulation, actually we solve some separate linear equations in time domain. For such linear equations, the solutions are easy to be acquired. If the source satisfy the condition $\overline{\tilde{S}_k} = \tilde{S}_{-k}$, we can get the $\tilde{X}_{-k}$ from $\tilde{X}_k$ directly. In practice, this property is very helpful to reduce the computing cost.

   HB simulation is another way for AC condition. It transfers the time domain to the frequency domain by means of Fourier transformation. The general HB equation is a linear equation, but for each frequency, it is a nonlinear problem. For linear circuit, HB equation can be acquired by NR method, but for nonlinear circuit, we need to substituting solution of time domain to $\mathbf{j}$ and $\mathbf{q}$ to get $\mathbf{J}$ and $\mathbf{Q}$ of frequency domain.

   BDF is used for transient analysis. The calculating of the transient simulation starts from DC condition. At each time-step, the equation is solved by NR method. In transient equation, the solution of the last time-step is known and be assumed as the initial guess for the NR method.

3. Using sensitivity analysis, we can know about the change rate of the circuit solutions or interested functions varies with parameters. According to this change rate, we can do optimization on circuit.

4. DC sensitivity can use direct method and adjoint method.

AC sensitivity also use direct method and adjoint method. The direct method is more complicated for its tensor operating. The adjoint method can be acquired using the results of direct method.

HB sensitivity also works on direct method and adjoint method. In this thesis, only theory implementations are given. More thorough analysis should be done in future work.

Compared with other kind sensitivity analyses, transient sensitivity have two more methods: integral method and extension integral method. Though integral method looks complicated than other method, it is a efficient method for integral functions. Integral method avoids to calculate $\hat{x}$ any more, even by such way a new luxury equation is introduced in. Extension integral method used the similar idea of integral method, but it brings into a lager error in results.

5. Fault analysis can help us to predict the circuit response in case the circuit perturbation becomes significant. If the perturbations happens from indictor or capacitor, the fault analysis will be much complicated.

For the future work, we still need some more works on HB analysis, especially on the nonlinear circuit. And some investigations should be done on the precision of extension integral method. Also the situation of perturbation on indictor and capacitor for fault analysis should be worked more. Further, the sensitivity on oscillator should be worked. Such sensitivity is related to frequency, which will be more complicated.

Bibliography

[1] E. J. W. ter Maten, Sensitivity in circuit simulation, Philips ED &T, Unpublished, 2004

[2] Robert Mattheij, Jaap Molenaar, Ordinary differential equations in theory and practice, SIAM, 1996

[3] A.Verhoeven, Transient sensitivity analysis, Philips ED&T, Unpublished, 2004

[4] R. Osserman, Two-dimensional Calculus, Harcourt, Brace & World , Inc., New York, 1968

[5] K.Balla, R.M$\ddot{a}$rz: Linear differential-algebraic equations of index-1 and their adjoint equations, Results in Maths., 2000

[6] M.Caracotsios, W.E.Stewart: Sensitivity analysis of initial value problems with mixed ODEs and algebraic equations, Computers and chemical engineering, 1985

[7] Y.Cao, S.Li, L.Petzold, R.Serban: Adjoint sensitivity for differential-algebraic equations: Part I, the adjoint DAE ayatem, Manuscript, Univ. of California, Santa Barbara, 2000

[8] Y.Cao, S.Li, L.Petzold, R.Serban: Adjoint sensitivity for differential-algebraic equations: the adjoint DAE system and its numerical solution, SIAMJ.Sci. Comput., 2002

[9] A.R.Conn, R.A.Haring, C.Visweswariah, C.W.Wu: Circuit optimization via adjoint lagrangians, Proc. ICCAD, San jose, CD, USA, 1997

[10] A.R.Conn, P.K.Coulman, R.A.Haring, G.L.Morrill, C.Visweswariah, C.W.Wu: Circuit optimization using time-domain sensitivities, IEEE Trans. on CAD of ICs and Syatems, 1998

[11] L.Daldoss, P.Gubian, M.Quarantelli: Tranient sensitivity computation in circuit simulation, Proc. ICCAD-1999, Part VI, 1999

[12] L.Daldoss, P.Gubian, M.Quarantelli: Multiparameter time-domain sensitivity computation, IEEE Trans. on Circuits and Syetems -I: Fund. Theory and Applics, 2001

[13] S.W.Director, R.A.Rohrer: The generalized adjoint network and network sensitivitties, IEEE trans. on circuit theory, 1969

[14] T.Engelen, N.Verdonschot: Design of module Šensitivity; Memo sensitivity_ds■p, Pstar documentation, Philips Research Lab, 1992

[15] W.F.Feehery, J.E.Tolsma, P.I. Barton: Effcient sensitivity analysis of large-scale differential-algebraic system, Appl. Numer. Maths., 1997

[16] M.Gerdts, C.B$\ddot{u}$skens: Consistent initialization of sensitivity matrices for a class of parametric DAE systems, BIT, 2002

[17] D.E.Hocevar, P.yang: Practical issues for implementing transient sensitivity computation, Proc. ICCAD-1984, 1984

[18] A.S.Householder: A survey of some closed methods for inversing matrices, Applied matematics, 1957

[19] S.Li, LPetzold: Software and algorithms for sensitivity analysis of large-scale differential-algebraic systems, J.Comput. and Appl. Maths., 2000

[20] S.Li, L.Petzold, W.Zhu: Sensitivity analysis of differential-algebraic equations: a comparison of methods on a special problem, Appl. Numer. Maths, 2000

[21] T.Maly, L.R.Petzold: Numerical methods and software for sensitivity analysis of differential-algebraic systems, Applied Number, 1996

[22] Y.L.O: Calculation of network sensitivities, Memo sensitivity_tn,mem, Pstar documentation, ED&T AS, Philips research lab, 1985

[23] J.Ogrodzki: Circuit simulation methods and algorithms, CRC Press, Boca Raton, FL, USA, 1994

appendix

.1 Matlab codes for the chapter 3

.1.1 AC sensitivity

```
function Sacanalysis=Sacanalysis(omega,T,style)
% this function is for AC sensitivity analysis;
% omega means the basic frequency of source; T means the test time;
% style==1 means linear condition; ==2 means nonlinear condition;

Capacitor=1;L=10;R2=10;V=20;
%---------- Sweeping of R1
a=1;b=50;delta_R1=0.1;
R1=(a-delta_R1):delta_R1:(b+delta_R1);
for jr1=1:round((b-a+2*delta_R1)/delta_R1+1)
%------------ DC condition
if style==1
    ib1_ini=20/(R1(jr1)+R2);v1_ini=20;v2_ini=(20*R2)/(R1(jr1)+R2);
elseif style==2
    solution=nrdcnl(20,R1(jr1));
    v1_ini=solution(1);v2_ini=solution(2);ib1_ini=solution(3);
end
ib4_ini=ib1_ini;v3_ini=0;
x_ini=[v1_ini v2_ini v3_ini ib1_ini ib4_ini]';
%--------------------------
%==============================
t=0:.01:T;
for j=1:round(T/0.01+1)

C=[0 0 0 0 0;0 Capacitor 0 0 0; 0 0 0 0 0; 0 0 0 0 0; 0 0 0 0 -L];
if style==1
G=[1/R1(jr1) -1/R1(jr1) 0 -1 0; -1/R1(jr1) (R1(jr1)+R2)/(R1(jr1)*R2) -1/R2 0 0;...
   0 -1/R2 1/R2 0 1; 1 0 0 0 0; 0 0 1 0 0];
elseif style==2
    G=[1/R1(jr1) -1/R1(jr1) 0 -1 0 ;...
       -1/R1(jr1) 1/R1(jr1)+3*v2_ini^2-6*v2_ini*v3_ini+3*v3_ini^2+4*v2_ini-4*v3_ini+1 ...
       -3*v2_ini^2+6*v2_ini*v3_ini-3*v3_ini^2-4*v2_ini+4*v3_ini-1 0 0;...
       0 -3*v2_ini^2+6*v2_ini*v3_ini-3*v3_ini^2-4*v2_ini+4*v3_ini-1 ...
       3*v2_ini^2-6*v2_ini*v3_ini+3*v3_ini^2+4*v2_ini-4*v3_ini+1 0 1;...
       1 0 0 0 0 ;0 0 1 0 0];
end
    %%%% hat_DC condition
    if style==1
    y=[1/R1(jr1) -1/R1(jr1) -1; -1/R1(jr1) (R1(jr1)+R2)/(R1(jr1)*R2) 0;1 0 0];
    b_hat=[V/((R1(jr1)+R2)*R1(jr1)) -V/((R1(jr1)+R2)*R1(jr1)) 0]';
    hat_xDC=y\b_hat;
    elseif style==2
    y=[1/R1(jr1) -1/R1(jr1) -1;-1/R1(jr1) 1/R1(jr1)+3*v2_ini^2+4*v2_ini+1 0;1 0 0];
    b_hat=[(v1_ini-v2_ini)/R1(jr1)^2 -(v1_ini-v2_ini)/R1(jr1)^2 0]';
    hat_xDC=y\b_hat;
    end
    hat_xDC_v1=hat_xDC(1);hat_xDC_v2=hat_xDC(2);hat_xDC_ib1=hat_xDC(3);
    hat_xDC_ib4=hat_xDC_ib1;hat_xDC_v3=0;
    hat_xDC=[hat_xDC_v1 hat_xDC_v2 hat_xDC_v3 hat_xDC_ib1 hat_xDC_ib4]';
    %%%%%%


k=1;
iwC=i*omega*k*C;
g=iwC+G;

Eplus=[0 0 0 -0.5*5*i 0]';
```

```
xplus(:,j)=g\Eplus;
xplus_v1=xplus(1,j);xplus_v2=xplus(2,j);xplus_v3=xplus(3,j);xplus_ib1=xplus(4,j);
xplus_ib4=xplus(5,j);

    %%%%%sensitivity for R1
    if style==1
    dg_dp=[-1/R1(jr1)^2 1/R1(jr1)^2 0 0 0;1/R1(jr1)^2 -1/R1(jr1)^2 0 0 0; 0 0 0 0 0;...
           0 0 0 0 0; 0 0 0 0 0];
    hat_xplus(:,j)=-g\(dg_dp*xplus(:,j));
    elseif style==2
    assi=6*xplus_v2-6*xplus_v3+4;
    dg_dx_x2=[0 0 0 0 0;0 assi -assi 0 0;0 -assi assi 0 0; 0 0 0 0 0;0 0 0 0 0]*hat_xDC;
    dg_dx_x3=[0 0 0 0 0;0 -assi assi 0 0;0 assi -assi 0 0; 0 0 0 0 0;0 0 0 0 0]*hat_xDC;
    dg_dx_x=[zeros(5,1) dg_dx_x2 dg_dx_x3 zeros(5,1) zeros(5,1)];
    dg_dp=[-1/R1(jr1)^2 1/R1(jr1)^2 0 0 0;1/R1(jr1)^2 -1/R1(jr1)^2 0 0 0; 0 0 0 0 0;...
           0 0 0 0 0; 0 0 0 0 0];
    hat_xplus(:,j)=-g\((dg_dp+dg_dx_x)*xplus(:,j));
    end
    %%%%%

k=-1;
iwC=i*omega*k*C;
g=iwC+G;

Eminus=[0 0 0 0.5*5*i 0]';
xminus(:,j)=g\Eminus;
xminus_v1=xminus(1,j);xminus_v2=xminus(2,j);xminus_v3=xminus(3,j);xminus_ib1=xminus(4,j);
xminus_ib4=xminus(5,j);
    %%%%%sensitivity for R1
    if style==1
    hat_xminus(:,j)=-g\(dg_dp*xminus(:,j));
    elseif style==2
    assi=6*xminus_v2-6*xminus_v3+4;
    dg_dx_x2=[0 0 0 0 0;0 assi -assi 0 0;0 -assi assi 0 0; 0 0 0 0 0;0 0 0 0 0]*hat_xDC;
    dg_dx_x3=[0 0 0 0 0;0 -assi assi 0 0;0 assi -assi 0 0; 0 0 0 0 0;0 0 0 0 0]*hat_xDC;
    dg_dx_x=[zeros(5,1) dg_dx_x2 dg_dx_x3 zeros(5,1) zeros(5,1)];
    dg_dp=[-1/R1(jr1)^2 1/R1(jr1)^2 0 0 0;1/R1(jr1)^2 -1/R1(jr1)^2 0 0 0; 0 0 0 0 0;...
           0 0 0 0 0; 0 0 0 0 0];
    hat_xminus(:,j)=-g\((dg_dp+dg_dx_x)*xminus(:,j));
    end
    %%%%%

x(:,j)=real(xplus(:,j)*exp(i*omega*t(j))+xminus(:,j)*exp(-i*omega*t(j)));
x(:,j)=x(:,j)+x_ini;
[xrow xcolumn]=size(x);
v1(j)=x(1,j);v2(j)=x(2,j);v3(j)=x(3,j);ib1(j)=x(4,j);ib4(j)=x(5,j);

   %%%%%sensitivity for R1
    hat_x(:,j)=real(hat_xplus(:,j)*exp(i*omega*t(j))+hat_xminus(:,j)*exp(-i*omega*t(j)))+hat_xDC;
    hat_v1(j)=hat_x(1,j);hat_v2(j)=hat_x(2,j);hat_v3(j)=hat_x(3,j);hat_ib1(j)=hat_x(4,j);...
    hat_ib4(j)=hat_x(5,j);
   %%%%%%
end

%================================
xR1(:,jr1)=real(x(:,round(T/0.01+1)));
hat_xR1(:,jr1)=real(hat_x(:,round(T/0.01+1)));
    %%%%%%%%%%%%%%adjoint method
     if style==1
         df_dx=[0 (2*(v2(xcolumn)-v3(xcolumn)))/R2 -(2*(v2(xcolumn)-v3(xcolumn)))/R2 0 0;ib1(xcolumn) 0 0 v1(xcolumn) 0];
         df_dp=[0;0];
         adj_f=df_dx*hat_x(:,xcolumn)+df_dp;
    elseif style==2
         co_dfdx=4*(v2(xcolumn)-v3(xcolumn))^3+6*(v2(xcolumn)-v3(xcolumn))^2+2*(v2(xcolumn)-v3(xcolumn));
         df_dx=[0 co_dfdx -co_dfdx 0 0;ib1(xcolumn) 0 0 v1(xcolumn) 0];
         df_dp=[0;0];
         adj_f=df_dx*hat_x(:,xcolumn)+df_dp;
    end
    %%%%%%
    adj_fR1(:,jr1)=adj_f(:,1);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
[row column]=size(xR1);
v1_R1=xR1(1,:);v2_R1=xR1(2,:);v3_R1=xR1(3,:);ib1_R1=xR1(4,:);ib4_R1=xR1(5,:);
[row column]=size(hat_xR1);
hat_v1_R1=hat_xR1(1,:);hat_v2_R1=hat_xR1(2,:);hat_v3_R1=hat_xR1(3,:);
hat_ib1_R1=hat_xR1(4,:);hat_ib4_R1=hat_xR1(5,:);
%------ end sweeping of R1
%------------ dxR1/dR1
for n=2:(column-1)
    dv1(n-1,:)=(v1_R1(n+1)-v1_R1(n-1))/(2*delta_R1);
    dv2(n-1,:)=(v2_R1(n+1)-v2_R1(n-1))/(2*delta_R1);
    dv3(n-1,:)=(v3_R1(n+1)-v3_R1(n-1))/(2*delta_R1);
```

```
    dib1(n-1,:)=(ib1_R1(n+1)-ib1_R1(n-1))/(2*delta_R1);
    dib4(n-1,:)=(ib4_R1(n+1)-ib4_R1(n-1))/(2*delta_R1);
end
xdR1=[dv1;dv2;dv3;dib1;dib4];
%--------------------------

%----------------------------
%********************plot of AC unknowns on interval of [0 T]

%**** plot of sensitivity on R1
plot(R1(2:(column-1)),dv1,'r-',R1(2:(column-1)),dv2,'g-',R1(2:(column-1)),dv3,'b-',...
     R1(2:(column-1)),dib1,'y-',R1(2:(column-1)),dib4,'c-')
ylabel('Sensitivity of AC solution on R1')
legend('dv1/dR1','dv2/dR1','dv3/dR1','dib1/dR1','dib4/dR1')
xlabel('R1[1 50]')
if style==1
title('Linear example for differential method')
elseif style==2
    title('Nonlinear example for differential method')
end
pause
%**** plot of sensitivity on R1
plot(R1(2:(column-1)),hat_v1_R1(2:(column-1)),'r-',R1(2:(column-1)),hat_v2_R1(2:(column-1)),'g-',...
     R1(2:(column-1)),hat_v3_R1(2:(column-1)),'b-',R1(2:(column-1)),hat_ib1_R1(2:(column-1)),'y-',...
     R1(2:(column-1)),hat_ib4_R1(2:(column-1)),'c-')
ylabel('Sensitivity of AC solution on R1')
legend('hat\_v1','hat\_v2','hat\_v3','hat\_ib1','hat\_ib4')
xlabel('R1[1 50]')
if style==1
title('Linear example for direct method')
elseif style==2
    title('Nonlinear example for direct method')
end
pause

plot(R1(2:(column-1)),adj_fR1(1,(2:(column-1))),'r-',R1(2:(column-1)),adj_fR1(2,(2:(column-1))),'g-')
xlabel('R1')
ylabel('df\_1 and df\_2')
legend('df1/dR1','df2/dR1')
if style==1
title('Linear example for adjoint method')
elseif style==2
    title('Nonlinear example for adjoint method')
end
```

.1.2 Transient sensitivity

```
% this function is for transient analysis;
% for every time step t_i, we will calculate the optimized t_i+1;
function Stanalysis=Stanalysis(omega,T,n,style)
% omega means the fenquency of source; n means the freedom number in limited time;
% T means the expected time; n=0 means DC; n=n+1 means T;
% style==1 means linear condition; style==2 means nonlinear condition;

C=1;L=10;R2=10;
delta_t=T/(n+1);t=delta_t*[0:n+1];

V=20+5*sin(omega*t);
a=1;b=50;
R1=a:.1:b;delta_R1=0.1;
R1=(a-delta_R1):delta_R1:(b+delta_R1);
for jr1=1:round((b-a+2*delta_R1)/delta_R1+1)
%---------DC condition
if style==1
   ib1_DC=20/(R1(jr1)+R2);v1_DC=20;v2_DC=(20*R2)/(R1(jr1)+R2);
   %%%%%%%%%%%%%
   y_hat=[1/R1(jr1) -1/R1(jr1) -1; -1/R1(jr1) (R1(jr1)+R2)/(R1(jr1)*R2) 0;1 0 0];
   b_hat=[20/((R1(jr1)+R2)*R1(jr1)) -20/((R1(jr1)+R2)*R1(jr1)) 0]';
   hat_xDC=y_hat\b_hat;
   %%%%%%%%%%%%%
elseif style==2
     [solution, hat_xDCnl]=nrdcnl(20,R1(jr1));
     v1_DC=solution(1);v2_DC=solution(2);ib1_DC=solution(3);
     %%%%%%%%%%%%%
     y=[1/R1(jr1) -1/R1(jr1) -1;-1/R1(jr1) 1/R1(jr1)+3*v2_DC^2+4*v2_DC+1 0;1 0 0];
     b_hat=[(v1_DC-v2_DC)/R1(jr1)^2 -(v1_DC-v2_DC)/R1(jr1)^2 0]';
     hat_xDC=y\b_hat;
     %%%%%%%%%%%%%
end
```

```
ib4_DC=ib1_DC;v3_DC=0;
x_DC=[v1_DC v2_DC v3_DC ib1_DC ib4_DC]';
v1_ini=v1_DC;v2_ini=v2_DC;v3_ini=v3_DC;ib1_ini=ib1_DC;ib4_ini=ib4_DC;
v1=v1_ini;v2=v2_ini;v3=v3_ini;ib1=ib1_ini;ib4=ib4_ini;

    %%%%%%%%%%%%%%%
    hat_xDC_v1=hat_xDC(1);hat_xDC_v2=hat_xDC(2);hat_xDC_ib1=hat_xDC(3);
    hat_xDC_ib4=hat_xDC_ib1;hat_xDC_v3=0;
    hat_xDC=[hat_xDC_v1 hat_xDC_v2 hat_xDC_v3 hat_xDC_ib1 hat_xDC_ib4]';
    hat_x=hat_xDC;
    %%%%%%%%%%%%%%%
%---------------------
for jt=1:n+1 %start time loop
  for j=1:20 %start NR loop
    x=[v1 v2 v3 ib1 ib4]';

    if style==1
       jnl=[-ib1+(v1-v2)/R1(jr1) -(v1-v2)/R1(jr1)+(v2-v3)/R2...
         -(v2-v3)/R2+ib4 v1-V(jt) v3]';
    elseif style==2
       jnl=[-ib1+(v1-v2)/R1(jr1) -(v1-v2)/R1(jr1)+(v2-v3)^3+2*(v2-v3)^2+v2-v3...
            -(v2-v3)^3-2*(v2-v3)^2-v2+v3+ib4 v1-V(jt) v3]';
    end

    qnl=[0 C*(v2-v2_ini)/delta_t 0 0 -L*(ib4-ib4_ini)/delta_t]';
    f=jnl+qnl;

    if style==1
       y=[1/R1(jr1) -1/R1(jr1) 0 -1 0 ;-1/R1(jr1) (R1(jr1)+R2)/(R1(jr1)*R2)+C/delta_t -1/R2 0 0 ;...
          0 -1/R2 1/R2 0 1 ;1 0 0 0 0 ;0 0 1 0 -L/delta_t ];

    elseif style==2
       y=[1/R1(jr1) -1/R1(jr1) 0 -1 0 ; -1/R1(jr1) ...
          1/R1(jr1)+3*v2^2-6*v2*v3+3*v3^2+4*v2-4*v3+1+C/delta_t ...
          -3*v2^2+6*v2*v3-3*v3^2-4*v2+4*v3-1 0 0;...
          0 -3*v2^2+6*v2*v3-3*v3^2-4*v2+4*v3-1 3*v2^2-6*v2*v3+3*v3^2+4*v2-4*v3+1 0 1;...
          1 0 0 0 0 ;0 0 1 0 -L/delta_t ];
    end

    x=x-inv(y)*f;
    approx(:,j)=x;
    if j>1
    error(:,j-1)=approx(:,j)-approx(:,j-1);
    nerror(j-1)=norm(error(:,j-1));
     if nerror(j-1)<=1e-005
       break
     end
    end
  end %############# end NR loop

    iniapprox=approx(:,j);
    v1=iniapprox(1);v2=iniapprox(2);v3=iniapprox(3);ib1=iniapprox(4);ib4=iniapprox(5);
    %%set as the initial value of the next timestep
    v1_ini=iniapprox(1);v2_ini=iniapprox(2);v3_ini=iniapprox(3);ib1_ini=iniapprox(4);ib4_ini=iniapprox(5);
    %%the values of the t_i timestep
    sol(:,jt)=iniapprox;

    %%%%%%%%%%%
    if style==1
        cal_C=[0 0 0 0 0;0 C 0 0 0; 0 0 0 0 0; 0 0 0 0 0; 0 0 0 0 -L];
        cal_G=[1/R1(jr1) -1/R1(jr1) 0 -1 0; -1/R1(jr1) (R1(jr1)+R2)/(R1(jr1)*R2) -1/R2 0 0;...
               0 -1/R2 1/R2 0 1; 1 0 0 0 0; 0 0 1 0 0];
        dj_dp=[-(v1-v2)/R1(jr1)^2 (v1-v2)/R1(jr1)^2 0 0 0]';
        hat_x=(cal_C/delta_t+cal_G)\((cal_C/delta_t)*hat_x-dj_dp);
        hat_X(:,jt)=hat_x;
    elseif style==2
        cal_C=[0 0 0 0 0;0 C 0 0 0; 0 0 0 0 0; 0 0 0 0 0; 0 0 0 0 -L];
        cal_G=[1/R1(jr1) -1/R1(jr1) 0 -1 0 ;...
           -1/R1(jr1) 1/R1(jr1)+3*v2^2-6*v2*v3+3*v3^2+4*v2-4*v3+1 ...
           -3*v2^2+6*v2*v3-3*v3^2-4*v2+4*v3-1 0 0;...
           0 -3*v2^2+6*v2*v3-3*v3^2-4*v2+4*v3-1 ...
           3*v2^2-6*v2*v3+3*v3^2+4*v2-4*v3+1 0 1;...
           1 0 0 0 0 ;0 0 1 0 0];
        dj_dp=[-(v1-v2)/R1(jr1)^2 (v1-v2)/R1(jr1)^2 0 0 0]';
        hat_x=(cal_C/delta_t+cal_G)\((cal_C/delta_t)*hat_x-dj_dp);
        hat_X(:,jt)=hat_x;
    end
    %%%%%%%%%%

end  %############## end time loop

x=[x_DC sol];
v1=x(1,:);v2=x(2,:);v3=x(3,:);ib1=x(4,:);ib4=x(5,:);
```

```
xR1(:,jr1)=sol(:,n+1);

%%%%%%%%%%
hat_X=[hat_xDC hat_X];
hat_xR1(:,jr1)=hat_X(:,n+2);
%%%%%%%%%%

    %%%%%%%%%%%%%%%adjoint method
    for ja=1:n+2
     if style==1
         df_dx=[0 (2*(v2(ja)-v3(ja)))/R2 -(2*(v2(ja)-v3(ja)))/R2 0 0;ib1(ja) 0 0 v1(ja) 0];
         df_dp=[0;0];
         adj_f(:,ja)=df_dx*hat_X(:,ja)+df_dp;
     elseif style==2
         co_dfdx=4*(v2(ja)-v3(ja))^3+6*(v2(ja)-v3(ja))^2+2*(v2(ja)-v3(ja));
         df_dx=[0 co_dfdx -co_dfdx 0 0;ib1(ja) 0 0 v1(ja) 0];
         df_dp=[0;0];
         adj_f(:,ja)=df_dx*hat_X(:,ja)+df_dp;
     end
   end
   int_adj_f_ini=0;
   for ja=1:n+1
   int_adj_f=delta_t/2*(adj_f(:,ja)+adj_f(:,ja+1));
   int_adj_f=int_adj_f+int_adj_f_ini;
   int_adj_f_ini=int_adj_f;
   end
    %%%%%%
    adj_fR1(:,jr1)=adj_f(:,n+2);
    int_adj_fR1(:,jr1)=int_adj_f;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %%%%%%%%%%% integral method

    lambda_T=zeros(2,5);
    for jn=n+1:-1:1
    if style==1
        df_dx=[0 (2*v2(jn)-2*v3(jn))/R2 -(2*v2(jn)-2*v3(jn))/R2 0 0;ib1(jn) 0 0 v1(jn) 0];
    elseif style==2
        assi=4*(v2(jn)-v3(jn))^3+6*(v2(jn)-v3(jn))^2+2*(v2(jn)-v3(jn));
        df_dx=[0 assi -assi 0 0;ib1(jn) 0 0 v1(jn) 0];
        cal_G=[1/R1(jr1) -1/R1(jr1) 0 -1 0 ;...
           -1/R1(jr1) 1/R1(jr1)+3*v2(jn)^2-6*v2(jn)*v3(jn)+3*v3(jn)^2+4*v2(jn)-4*v3(jn)+1 ...
           -3*v2(jn)^2+6*v2(jn)*v3(jn)-3*v3(jn)^2-4*v2(jn)+4*v3(jn)-1 0 0;...
           0 -3*v2(jn)^2+6*v2(jn)*v3(jn)-3*v3(jn)^2-4*v2(jn)+4*v3(jn)-1 ...
           3*v2(jn)^2-6*v2(jn)*v3(jn)+3*v3(jn)^2+4*v2(jn)-4*v3(jn)+1 0 1;...
           1 0 0 0 0 ;0 0 1 0 0];
    end
        lambda=(df_dx+lambda_T*(cal_C/delta_t))*inv(cal_C/delta_t+cal_G);
        lambda_T=lambda;
        f1_lambda(:,jn)=lambda(1,:)';
        f2_lambda(:,jn)=lambda(2,:)';
    end %end backward for lambda from T to DC;
    f1_lambda=[f1_lambda zeros(5,1)];
    f2_lambda=[f2_lambda zeros(5,1)];
    %-----------the 1st-differertion for lambda
    lambda_0=[f1_lambda(:,1)';f2_lambda(:,1)'];lambda_1=[f1_lambda(:,2)';f2_lambda(:,2)'];
    d_lambda=(lambda_1-lambda_0)/delta_t;
    d_f1_lambda(:,1)=d_lambda(1,:)';d_f2_lambda(:,1)=d_lambda(2,:)';
    lambda_Tminus1=[f1_lambda(:,n+1)';f2_lambda(:,n+1)'];lambda_T=[f1_lambda(:,n+2)';f2_lambda(:,n+2)'];
    d_lambda=(lambda_T-lambda_Tminus1)/delta_t;
    d_f1_lambda(:,n+2)=d_lambda(1,:)';d_f2_lambda(:,n+2)=d_lambda(2,:)';
    for jl=2:n+1
        lambda_jlminus1=[f1_lambda(:,jl-1)';f2_lambda(:,jl-1)'];
        lambda_jlplus1=[f1_lambda(:,jl+1)';f2_lambda(:,jl+1)'];
        d_lambda=(lambda_jlplus1-lambda_jlminus1)/(2*delta_t);
        d_f1_lambda(:,jl)=d_lambda(1,:)';d_f2_lambda(:,jl)=d_lambda(2,:)';
    end
    %--------------

    cal_L_T=-[d_f1_lambda(:,n+2)';d_f2_lambda(:,n+2)'];
    for jn=n+1:-1:1
     if style==2
        cal_G=[1/R1(jr1) -1/R1(jr1) 0 -1 0 ;...
           -1/R1(jr1) 1/R1(jr1)+3*v2(jn)^2-6*v2(jn)*v3(jn)+3*v3(jn)^2+4*v2(jn)-4*v3(jn)+1 ...
           -3*v2(jn)^2+6*v2(jn)*v3(jn)-3*v3(jn)^2-4*v2(jn)+4*v3(jn)-1 0 0;...
           0 -3*v2(jn)^2+6*v2(jn)*v3(jn)-3*v3(jn)^2-4*v2(jn)+4*v3(jn)-1 ...
           3*v2(jn)^2-6*v2(jn)*v3(jn)+3*v3(jn)^2+4*v2(jn)-4*v3(jn)+1 0 1;...
           1 0 0 0 0 ;0 0 1 0 0];
    end
        cal_L=(cal_L_T*(cal_C/delta_t))*inv(cal_C/delta_t+cal_G);
        cal_L_T=cal_L;
        f1_cal_L(:,jn)=cal_L(1,:)';
        f2_cal_L(:,jn)=cal_L(2,:)';
```

```
    end %end backward for cal_L from T to DC;
    f1_cal_L=[f1_cal_L -d_f1_lambda(:,n+2)];
    f2_cal_L=[f2_cal_L -d_f2_lambda(:,n+2)];

    %-----------the 1st-differertion for cal_L on t
    cal_L_0=[f1_cal_L(:,1)';f2_cal_L(:,1)'];cal_L_1=[f1_cal_L(:,2)';f2_cal_L(:,2)'];
    d_cal_L=(cal_L_1-cal_L_0)/delta_t;
    d_f1_cal_L(:,1)=d_cal_L(1,:)';d_f2_cal_L(:,1)=d_cal_L(2,:)';
    cal_L_Tminus1=[f1_cal_L(:,n+1)';f2_cal_L(:,n+1)'];cal_L_T=[f1_cal_L(:,n+2)';f2_cal_L(:,n+2)'];
    d_cal_L=(cal_L_T-cal_L_Tminus1)/delta_t;
    d_f1_cal_L(:,n+2)=d_cal_L(1,:)';d_f2_cal_L(:,n+2)=d_cal_L(2,:)';
    for jl=2:n+1
        cal_L_jlminus1=[f1_cal_L(:,jl-1)';f2_cal_L(:,jl-1)'];
        cal_L_jlplus1=[f1_cal_L(:,jl+1)';f2_cal_L(:,jl+1)'];
        cal_L=(cal_L_jlplus1-cal_L_jlminus1)/(2*delta_t);
        d_f1_cal_L(:,jl)=d_cal_L(1,:)';d_f2_cal_L(:,jl)=d_cal_L(2,:)';
    end
    %-------------

    %-------------
    int_ini=zeros(2,2); int_ini_ext=zeros(2,2);
    for jn=1:n+1
        lambda=[f1_lambda(:,jn)';f2_lambda(:,jn)'];
        lambda_plus=[f1_lambda(:,jn+1)';f2_lambda(:,jn+1)'];
        d_lambda=[d_f1_lambda(:,jn)';d_f2_lambda(:,jn)'];
        d_lambda_plus=[d_f1_lambda(:,jn+1)';d_f2_lambda(:,jn+1)'];
        cal_L=[f1_cal_L(:,jn)';f2_cal_L(:,jn)'];
        cal_L_plus=[f1_cal_L(:,jn+1)';f2_cal_L(:,jn+1)'];
        d_cal_L=[d_f1_cal_L(:,jn)';d_f2_cal_L(:,jn)'];
        d_cal_L_plus=[d_f1_cal_L(:,jn+1)';d_f2_cal_L(:,jn+1)'];
        if style==1
          dj_dp=[-(v1(jn)-v2(jn))/R1(jr1)^2 0;(v1(jn)-v2(jn))/R1(jr1)^2 -(v2(jn)-v3(jn))/R2^2;0 (v2(jn)-v3(jn))/R2^2; 0 0;0 0];
          df_dp=[0 -(v2(jn)-v3(jn))^2/R2^2;0 0];
          dj_dp_plus=[-(v1(jn+1)-v2(jn+1))/R1(jr1)^2 0;(v1(jn+1)-v2(jn+1))/R1(jr1)^2 -(v2(jn+1)-v3(jn+1))/R2^2;...
                      0 (v2(jn+1)-v3(jn+1))/R2^2; 0 0;0 0];
          df_dp_plus=[0 -(v2(jn+1)-v3(jn+1))^2/R2^2;0 0];
          int=delta_t/2*((-lambda*dj_dp+df_dp)+(-lambda_plus*dj_dp_plus+df_dp_plus));

          int_ext=delta_t/2*(-cal_L*dj_dp-cal_L_plus*dj_dp_plus);

        elseif style==2
            dj_dp=[-(v1(jn)-v2(jn))/R1(jr1)^2 0; (v1(jn)-v2(jn))/R1(jr1)^2 0; 0 0;0 0;0 0];
            dq_dp=[0 0; 0 v2(jn);0 0;0 0;0 0];
            df_dp=[0 0;0 0];

            dj_dp_plus=[-(v1(jn+1)-v2(jn+1))/R1(jr1)^2 0; (v1(jn+1)-v2(jn+1))/R1(jr1)^2 0; 0 0;0 0;0 0];
            dq_dp_plus=[0 0; 0 v2(jn+1);0 0;0 0;0 0];
            df_dp_plus=[0 0;0 0];
            int=delta_t/2*((d_lambda*dq_dp-lambda*dj_dp+df_dp)+(d_lambda_plus*dq_dp_plus-lambda_plus*dj_dp_plus+df_dp_plus));

            int_ext=delta_t/2*((d_cal_L*dq_dp-cal_L*dj_dp)+(d_cal_L_plus*dq_dp_plus-cal_L_plus*dj_dp_plus));
        end
       int=int+int_ini;int_ini=int;

       int_ext=int_ext+int_ini_ext;int_ini_ext=int_ext;
    end

    if style==1
        first_term=lambda_0*[cal_C*hat_X(:,1) zeros(5,1)];
        int=int+first_term;

        first_term_ext=cal_L_0*[cal_C*hat_X(:,1) zeros(5,1)];
        right_term_ext=[0 -(v2(n+2)-v3(n+2))^2/(R2^2);0 0];
        int_ext=int_ext+first_term_ext+right_term_ext;
    elseif style==2
        dq_dx_hat_x=[cal_C*hat_X(:,1) zeros(5,1)];
        dq_dp_0=[0 0; 0 v2(1);0 0;0 0;0 0];
        first_term=lambda_0*(dq_dx_hat_x+dq_dp_0);
        int=int+first_term;

        first_term_ext=cal_L_0*(dq_dx_hat_x+dq_dp_0);
        d_lambda_T=[d_f1_lambda(:,n+2)';d_f2_lambda(:,n+2)'];
        dq_dp_T=[0 0; 0 v2(n+2);0 0;0 0;0 0];df_dp_T=[0 0;0 0];
        right_term_ext=d_lambda_T*dq_dp_T+df_dp_T;
        int_ext=int_ext+first_term_ext+right_term_ext;
    end
    int_fR1(:,jr1)=int(:,1);
    int_fR1_ext(:,jr1)=int_ext(:,1);

end %############ end R1 loop

%----------- dsolR1/dR1
[row column]=size(xR1);
```

```
v1_R1=xR1(1,:);v2_R1=xR1(2,:);v3_R1=xR1(3,:);ib1_R1=xR1(4,:);ib4_R1=xR1(5,:);
%------------ dxR1/dR1
for n=2:(column-1)
    dv1(n-1,:)=(v1_R1(n+1)-v1_R1(n-1))/(2*delta_R1);
    dv2(n-1,:)=(v2_R1(n+1)-v2_R1(n-1))/(2*delta_R1);
    dv3(n-1,:)=(v3_R1(n+1)-v3_R1(n-1))/(2*delta_R1);
    dib1(n-1,:)=(ib1_R1(n+1)-ib1_R1(n-1))/(2*delta_R1);
    dib4(n-1,:)=(ib4_R1(n+1)-ib4_R1(n-1))/(2*delta_R1);
end
%-------------------------

%**** plot of transient sensitivity on R1
plot(R1(2:(column-1)),dv1,'r-',R1(2:(column-1)),dv2,'g-',...
     R1(2:(column-1)),dv3,'b-',R1(2:(column-1)),dib1,'y-',R1(2:(column-1)),dib4,'k-','linewidth',2)
xlabel('R1[1 50]')
ylabel('Sensitivity of transient solution on R1')
legend('dv1/dR1','dv2/dR1','dv3/dR1','dib1/dR1','dib4/dR1')
if style==1
title('Linear example for differential method')
elseif style==2
    title('Nonlinear example for differential method')
end
pause

plot(R1(2:(column-1)),hat_xR1(1,2:(column-1)),'r-',R1(2:(column-1)),hat_xR1(2,2:(column-1)),'g',...
     R1(2:(column-1)),hat_xR1(3,2:(column-1)),'b-',R1(2:(column-1)),hat_xR1(4,2:(column-1)),'y-',...
     R1(2:(column-1)),hat_xR1(5,2:(column-1)),'k-','linewidth',2)
xlabel('R1[1 50] ')
ylabel('Sensitivity of transient solution on R1')
legend('hat\_v1','hat\_v2','hat\_v3','hat\_ib1','hat\_ib4')
if style==1
  title('Linear example for direct method')
elseif style==2
    title('Nonlinear example for direct method')
end
pause

plot(R1(2:(column-1)),int_adj_fR1(1,(2:(column-1))),'r-',R1(2:(column-1)),int_adj_fR1(2,(2:(column-1))),'g-','linewidth',2)
xlabel('R1[1 50]')
ylabel('f\_1 and f\_2')
legend('f\_1','f\_2')
if style==1
title('Linear example by adjoint method for integral function')
elseif style==2
title('Nonlinear example by adjoint method for integral function')
end
pause

plot(R1(2:(column-1)),int_fR1(1,(2:(column-1))),'r-',R1(2:(column-1)),int_fR1(2,(2:(column-1))),'g-','linewidth',2)
xlabel('R1[1 50]')
ylabel('dF\_1 and dF\_2')
legend('dF1/dR1','dF2/dR1')
if style==1
title('Linear example for integral method')
elseif style==2
    title('Nonlinear example for integral method')
end
pause

plot(R1(2:(column-1)),adj_fR1(1,(2:(column-1))),'r-',R1(2:(column-1)),adj_fR1(2,(2:(column-1))),'g-','linewidth',2)
xlabel('R1[1 50]')
ylabel('df\_1 and df\_2')
legend('df1/dR1','df2/dR1')
if style==1
    title('Linear example for adjoint method')
elseif style==2
    title('Nonlinear example for adjoint method')
end
pause

plot(R1(2:(column-1)),int_fR1_ext(1,(2:(column-1))),'r-',R1(2:(column-1)),int_fR1_ext(2,(2:(column-1))),'g-','linewidth',2)
xlabel('R1[1 50]')
ylabel('df\_1 and df\_2')
legend('df1/dR1','df2/dR1')
if style==1
    title('Linear example for extension integral method')
elseif style==2
    title('Nonlinear example for extension integral method')
end
```