Exploiting Matrix Sparsity for Symbolic Analysis

ZDENĚK KOLKA, VIERA BIOLKOVÁ*, DALIBOR BIOLEK+
*)Department of Radio Electronics, +)Department of Microelectronics
Brno University of Technology
Purkyňova 118, 612 00 Brno
CZECH REPUBLIC
kolka@feec.vutbr.cz

Abstract: - This paper deals with a method for symbolic approximation that exploits the sparsity of circuit matrix to achieve an acceptable speed for large circuits. The method is based on a simplification of the equations of circuit models of linear or linearized circuits in the frequency domain. The simplified model is then analyzed symbolically. The algorithm proposed has been developed with the aim of obtaining maximum computational efficiency.

Key-Words: - Approximate symbolic analysis, linear circuits, circuit decomposition

1 Introduction

Symbolic analysis represents a supplement to the numerical analysis in the field of linearized electronic circuits. An analytical expression for network function can provide more information than multiple Spice simulations. It is well known the classical (exact) symbolic analysis is applicable only to very small and rather idealized circuits. It serves as a tool for basic functional verification. However, an analysis of practical transistor-level circuits by this method is unfeasible. The size of exact symbolic expressions grows exponentially with the number of nodes [1]. For blocks with ten or more transistors the expression can contain more than 10^{10} terms. Such a huge expression is practically useless. The results are for the designer to be interpreted in order to get insightful information about the circuit. Thus, it is absolutely necessary that the expression should be simple enough to be interpretable.

Since the early 90s, a combination of numerical and symbolic analysis has been extensively studied [1]. This approach consists in generating approximate symbolic expressions whose size is acceptable for human interpretation. The simplification is based on the omission of insignificant terms that practically do not influence the numerical value of the expression. Thus it requires the knowledge of at least approximate values of circuit parameters. Simplicity is achieved at the expense of expression accuracy. The validity of simplified results is usually guaranteed on a given frequency interval and (rarely) for a certain range of parameters of network elements. The Simplification Before Generation (SBG) method, which simplifies the circuit model, is the most powerful one [1], [4]. It somehow mimics what a designer does during pen-and-paper calculations but with an exact control over approximation errors. The method consists in two main operations:

- sequence of simplification steps of circuit matrix or graphs [1], [4],
- generation of symbolic solution.

The expected size of resulting expression is several lines. Thus, there are no special requirements for the symbolic solution. A variety of known methods can be used [1].

2 Basic algorithm of Simplification

Let us suppose, for simplicity, that the circuit consists of generalized admittance-type elements, i.e. each one can be described by

$$i_a = y v_b \ . \tag{1}$$

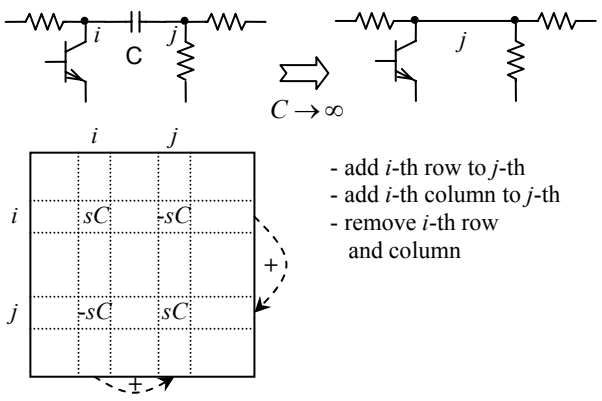

Fig. 1 Simplification step for $C \to \infty$.

Setting $y \to 0$ or $y \to \infty$ eliminates the element from the system of network equations, i.e. simplifies its symbolic solution. For example, a coupling capacitor can be regarded as short circuit at high

frequencies. Setting $C \to \infty$ reduces the size of the circuit matrix by 1, Fig. 1. Practical algorithms use more sophisticated approach that consists in modification of circuit matrix or voltage and current graphs but the basic idea remains the same [1], [4].

Each simplification step introduces a certain error. Simplification operations with the lowest error are carried out first. The process stops when the maximum approximation error is reached. Let the circuit be originally described by n_O linear equations. After r simplification steps the number of equations decreases to n_S, Fig. 2.

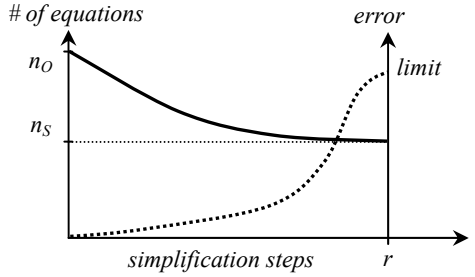

Fig. 2 Gradual simplification of circuit model.

The linear circuit is described by

$$\mathbf{H}\,\mathbf{v} = \mathbf{b} \ , \tag{2}$$

where $\mathbf{H} \in C^{nxn}$, $\mathbf{v}, \mathbf{b} \in C^n$. For simplicity, the nodal formulation will be assumed. Any network function K can be obtained as a quotient of two algebraic cofactors of $\mathbf{H}$

$$K = (-1)^{\alpha} \frac{\det(\mathbf{H}_1)}{\det(\mathbf{H}_2)} \ . \tag{3}$$

Matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ are derived from $\mathbf{H}$ by means of adding and deleting some rows and columns. Details can be found in textbooks of circuit theory. Assume, for example, a conductor g between nodes i and j. Both determinants in (3) can be expanded as

$$\det(\mathbf{H}_1) = \det(\mathbf{H}_1') + g\left(\Delta_{\mathrm{H1}i:i} + \Delta_{\mathrm{H1}j:j} - \Delta_{\mathrm{H1}i:j} - \Delta_{\mathrm{H1}j:i}\right), \tag{4}$$

where $\mathbf{H}_1$' does not contain g and $\Delta_{\mathrm{H1}i:j}$ is a cofactor. Equation (4) enables us to rewrite (3) as

$$K = \frac{ag + b}{cg + d} \ . \tag{5}$$

Having all the elements of (5) we can determine numerically the influence of $g \to 0$ and $g \to \infty$ without recomputing (3). Everything we need is the first-order cofactors of a regular matrix which can be computed numerically by a simple inversion as

$$\mathbf{A}_{\Delta} = \det(\mathbf{A})\left(\mathbf{A}^{-1}\right)^T \ , \tag{6}$$

where each element of $\mathbf{A}_{\Delta}$ is the respective algebraic cofactor of $\mathbf{A}$. The computation of (6) is rather expensive for large circuits. For a matrix of order n it requires

$$2(n-1)^3 \tag{7}$$

"long" operations for both the numerator and the denominator [2]. It is shown in [1] that (6) can be updated "lazily" after performing several simplification steps. It should be done at least during each reduction of the matrix order (as in Fig. 1). Thus the computational cost of (6) is

$$N_{(6)} \geq 2\sum_{i=n_S}^{n_O} (i-1)^3 = \frac{1}{2}(n_O^4 - n_S^4) + O({n_O}^3, {n_S}^3) \ . \tag{8}$$

A more effective algorithm is based on the Sherman-Morrison-Woodbury formula (9) [1]. The matrix change during each simplification step can be interpreted as adding some difference matrix to $\mathbf{H}$ in (2). For example, the deletion of a term in $\mathbf{H}$ is equivalent to adding its opposite value.

Let $\mathbf{A} \in C^{nxn}$ be a regular matrix and $\mathbf{B} = \mathbf{A} + \mathbf{u}\mathbf{v}^T$, where $\mathbf{u}, \mathbf{v} \in C^n$. Then

$$\mathbf{B}^{-1} = \mathbf{A}^{-1} - \frac{1}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1} \ . \tag{9}$$

If $\mathbf{u}$ and $\mathbf{v}$ contain only one nonzero element, then the cost of (9) is

$$N_{(9)} \approx n^2 \tag{10}$$

operations. The update of (6) should be performed after each simplification step, i.e. the total cost including the initial inversion is

$$N_A \leq r\,{n_O}^2 + {n_O}^3 \tag{11}$$

not counting possible order reduction of $\mathbf{H}$. The number of circuit elements for practical circuits ranges from two to four times the number of nodes [2]. If the majority of elements was removed then

$$N_A \leq (3 \div 5)\,{n_O}^3 \ . \tag{12}$$

3 Block Decomposition with Gradual Inverse Matrix Update

The approximate symbolic analysis of large circuits should exploit the sparsity of system matrix by means of partitioning the circuit into smaller blocks. A block is a group of nodes and elements tightly coupled but connected outside with only a rather small number of branches [3]. All practical circuits are inherently composed of simpler blocks.

Let us consider, for simplicity, a large circuit partitioned into M equal blocks with m nodes by means of nodal decomposition [3]. Nodes inside each block are called *internal* and nodes where blocks interact are called *block nodes*. Let the circuit have n_B block nodes, $n_B << n_O$, $m \approx (n_O - n_B)/M$.

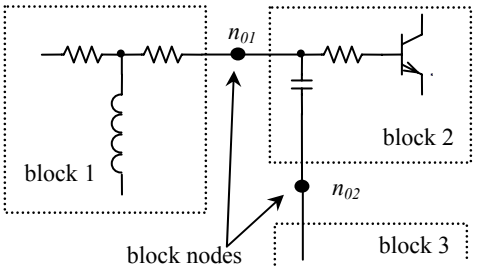

Fig. 3 Nodal decomposition of a circuit.

If we reorder the circuit variables, starting from the voltages $\mathbf{v}_i$ of internal nodes, and terminating with the voltages $\mathbf{v}_0$ of block nodes, then the matrix $\mathbf{H}$ takes Border Block Diagonal form [3]

$$\begin{bmatrix} \mathbf{Y}_1 & & & & \mathbf{P}_1 \\ & \mathbf{Y}_2 & & & \mathbf{P}_2 \\ & & \ddots & & \vdots \\ & & & \mathbf{Y}_M & \mathbf{P}_M \\ \mathbf{Q}_1 & \mathbf{Q}_2 & \cdots & \mathbf{Q}_M & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_M \\ \mathbf{v}_0 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{b}_0 \end{bmatrix}, \quad (13)$$

where only $\mathbf{b}_0$ is nonzero because external ports are considered among block nodes.

The simplification procedure processes the blocks successively. The admissible approximation error for each block is a fraction of the error for the whole circuit. Blocks may be simplified in several rounds until the maximum error is reached. The block under simplification is considered as symbolic while the others are represented numerically. Suppose that block 1 is being simplified. The voltages of other blocks can be expressed numerically as

$$\begin{aligned} \mathbf{v}_2 &= -\mathbf{Y}_2^{-1}\,\mathbf{P}_2\,\mathbf{v}_0 \\ \mathbf{v}_3 &= -\mathbf{Y}_3^{-1}\,\mathbf{P}_3\,\mathbf{v}_0 \quad . \\ &\dots \end{aligned} \quad (14)$$

Then, system (13) can be reduced to

$$\begin{bmatrix} \mathbf{Y}_1 & \mathbf{P}_1 \\ \mathbf{Q}_1 & -\mathbf{Q}_2\mathbf{Y}_2^{-1}\mathbf{P}_2 - \mathbf{Q}_3\mathbf{Y}_3^{-1}\mathbf{P}_3 - \ldots + \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_0 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{b}_0 \end{bmatrix}. \quad (15)$$

The structure of symbolic entries into matrices $\mathbf{Y}_1$, $\mathbf{P}_1$, $\mathbf{Q}_1$, and $\mathbf{R}$ corresponds to the structure of standard Nodal Analysis matrix, see Fig. 3. System (15) represents $m + n_B$ mixed symbolic-numerical equations. The simplification procedure is applied to the symbolic terms of (15). Since $\mathbf{v}_0$ includes the voltages of external ports, we have control over the numerical value, i.e. approximation error, of the network function of the whole circuit. The algorithm is as follows:

1. Initial computation of $\mathbf{Q}_i\mathbf{Y}_i^{-1}\mathbf{P}_i$ for all blocks [$M(m^3 + n_B m(n_B + m))$ operations].
2. Setting up of (15) for the block under simplification, and computation of cofactor matrices (6) [$2(m + n_B)^3$ operations].
3. Simplification steps for the selected block [$2(m + n_B)^2$ operations for each step].
4. Update of $\mathbf{Q}_i\mathbf{Y}_i^{-1}\mathbf{P}_i$ for the simplified block [$m^3 + n_B m(n_B + m)$ operations at most].
5. Repeat 2 – 4 for other blocks until the maximum approximation error is reached.

The total complexity can be estimated as a sum of operations for steps 1 – 5. Taking into account the inequality $n_B << n_O$ the number of operations is

$$N_B \approx \frac{{n_O}^3}{M^2}\left(2 + \frac{c}{M}\right), \quad (16)$$

where c is the number of complete simplification cycles.

4 Conclusions

The method of gradual submatrix inversion update fully exploits the sparsity of matrices for large electronic circuits. It requires fewer operations than standard full-matrix approaches.

5 Acknowledgement

This research has been supported by the Grant Agency of the Academy of Sciences of the Czech Republic under the contract no. B2813301.

References:

[1] HENNIG, E.: *Symbolic Approximation and Modeling Techniques for Analysis and Design of Analog Circuits*, PhD dissertation, Univ. of Kaiserslautern, 2000.

[2] VLACH, J.-SINGHAL, K.: *Computer Methods for Circuit Analysis and Design*, New York: Van Nostrad Reinhold, 1994.

[3] OGRODZKI, J.: *Circuit Simulation Methods and Algorithms*, CRC Press, Boca Raton, 1994.

[4] KOLKA, Z.-VLK, M.: *Combined Approach to Symbolic Approximation*, Proc of 7th WSEAS Intern. Conf. on CIRCUITS, Greece, 2003.