Computer-Aided Analysis of Microwave Circuits

VITO A. MONACO, MEMBER, IEEE, AND PAOLO TIBERIO, MEMBER, IEEE

(Invited Paper)

Abstract—**The most relevant techniques that have either found or should find useful application in analyzing microwave circuit performances in the frequency domain are surveyed. The particular needs of the microwave engineer are briefly discussed. Circuit equation formulations in terms of voltages and currents and wave variables are presented and the solution of the set of circuit equations by sparse-matrix techniques is illustrated. Methods based on multiport connection are also reviewed.**

The techniques for computing the first- and second-order sensitivity are illustrated and a comparison is made between the direct method and the transpose-matrix method, which is in certain cases similar to the method based on the adjoint circuit.

I. Introduction

THE PROGRESS registered in recent years in the field of computer-aided design has been considerable and conceptually important, so much so that the computer is no longer considered an auxiliary aid for checking the validity of a solution obtained in other ways, but rather as an indispensable instrument during all circuit design phases. Present-day computer programs, in fact, permit determination not only of the component parameter nominal values but also their maximum permitted spreads in relation to given tolerances on circuit response functions and to the required production yield when a large number of identical circuits must be realized. This is made possible by the availability of analysis programs that, besides being rapid, also allow precise determination of network functions without limitations on component composition or on circuit topology.

This paper describes the methods and algorithms that are the basis for the most important and known programs for analyzing linear circuits in frequency domain. Though giving most attention to the methods specially conceived for microwave circuits with distributed elements, a description is also given of those that are the basis for lumped-element circuit analysis programs since, with appropriate artifices and modifications, they could be adapted for the analysis of distributed component circuits. A comparison between the various methods is also made with a view, above all, to advising the reader of the different limitations deriving from them regarding circuit topology and component composition.

No indications are given on the structure and use of the programs since, being well aware that the employment of a program is strictly conditioned by its simplicity in use, we do not believe this is essential for this paper. In fact, there are no theoretical difficulties that prevent programs from meeting the user's requirements when these are clearly defined [1]. An analysis program, whether used as a routine of larger programs (optimization and tolerance assignment programs) or as an independent instrument for the designer, must have rapid execution, easy input data preparation, and clear output data presentation. Fast execution above all is required when a number of analyses have to be made of one circuit with different component values, as happens in optimization processes and in component tolerance assignment. The other requirement, that is, the simplicity of man–machine interaction, is desirable both to overcome the designer's natural reluctance to use something new and to reduce the time spent in input–output operations [2]. It must be possible, in particular, to modify component values and circuit topology without changing the complete data file. Besides, the printed output data must contain not only the required network functions but also the input data describing the circuit so that it is possible to recognize even some time afterwards the circuit to which they refer. The possibility of obtaining results graphically by means of the same printer or, even better, by means of a plotter or a cathode ray tube display is, finally, a useful though not indispensable convenience.

To analyze a circuit means computing a certain number of response functions in terms of the component parameters, circuit topology, and independent excitations being given. The response functions may be determined both in terms of voltages and currents in some nodes and branches of the circuit, as is usually done with circuits composed of lumped components, and in terms of incident and reflected waves at some ports, as is often done with microwave circuits. Correspondingly, circuit equations may be formulated either in terms of voltages and currents or of normalized wave variables. In the first case, the components are defined by means of admittance, impedances, or dependent generators; each constitutes a branch that is connected between the nodes. In the second case, the components are multiports connected through pairs of ports. The circuit description is effected by means of topological matrices that indicate the nodes between which the branches are connected or the pairs of adjacent ports.

Circuit analysis, that is, determination of the voltages and currents or of the normalized waves, implies the solution of a system of equations whose number, according

Manuscript received July 20, 1973; revised December 7, 1973. This work was supported by the National Research Council of Italy.

The authors are with the Istituto di Elettronica, Università di Bologna, Centro Interazione Operatore-Calcolatore (CNR), Bologna, Italy.

to the method adopted, depends on the number of nodes and branches of the network or on the number of component ports. This might cause limitations in the maximum number of circuit components in relation to computer memory capacity and to the maximum running time imposed.

The classical solution methods require a number of operations proportional to the cube of the number of equations [3]. Various methods have been proposed to reduce computing time by taking advantage of the sparsity of the coefficient matrices of the equation system. There are two basic approaches. The first consists of progressive elimination of all variables not needed for the requested network functions. This approach has been most widely used in microwave circuit analysis since only the external port variables are generally of interest With the second approach, computing time reduction is obtained by carrying out only the arithmetical operations with nonzero operands. It is based on examination of the sparsity structure of the coefficient matrix and on generation of a code or of a set of pointers and indices that provide the "key" to execution of the arithmetical operations. It was first used for power distribution network analysis and subsequently for lumped and distributed circuit analysis.

A limitation on the use of many existing programs for analyzing microwave circuits derives from the fact that only two-terminal components are permitted. Because of this, all the circuit components must be described by means of lumped-element equivalent circuits. For transistors at high frequency, numerous models have been proposed [4]–[7]. However, it is not always easy in the microwave field to characterize active and passive components by means of lumped-element equivalent circuits, but it is always possible to determine the parameters of any component by direct measurements at its ports. In this regard it is observed that both manual and automatic instruments exist today that permit measurements of scattering parameters on broad frequency bands both precisely and very quickly [8]. In the opinion of the authors, shared also by others [1], [9] the possibility of defining circuit components by means of measured parameters must therefore be considered as a necessary property of programs for microwave circuit analysis.

Alternatively, the port parameters of many passive components (microstrip transmission lines, etc.) can be determined in terms of their geometrical dimensions and the electrical characteristics of the materials forming them. In this case it is enough to insert appropriate routines in the analysis program to calculate the component parameters. In this paper, however, no indication is given regarding the operation of these routines since we consider only the problem of determining circuit performance in terms of the electrical parameters.

In Section II, the formulation of circuit equations is described when branch voltages and currents and node voltages are considered as unknowns, and the tableau method and the nodal admittance matrix are explained along with the modified tableau and the mixed method. In Section III, the circuit equation formulation is given when normalized waves at component ports are considered as unknowns and the connection scattering matrix is described. Subsequently, in Section IV, the methods based on multiport connection and, therefore, on the computation of the port matrix of the complete circuit are given. A comparison is also made between the methods, bringing out the inconveniences of some, particularly in relation to accuracy in calculation and the difficulties that might arise in describing the circuits. In Section V, the sparse-matrix techniques for solving the system of the equations are discussed. The method based on execution code generation is dealt with in detail; this is particularly suitable for microwave circuits since their dimensions are not usually very great. A comparison is also made between the computation times required to analyze a single circuit by means of two programs, one being based on the method of connecting components in pairs, the other on the generation and execution of the code that solves the system of the equations describing the circuit.

Finally, Section VI is devoted to computation of network function sensitivities with respect to component parameters.

The direct and the transpose-matrix methods are described and a discussion is presented on the computing effort required by each one. The equivalence that in certain cases exists between the transpose-matrix method and the adjoint circuit method is also shown.

To conclude, the results of analyses on some circuits are given and the problem of component tolerance assignment in relation to permitted performance tolerances is briefly described. This problem is of particular interest to industry, since circuit cost and mass production yield are often greatly dependent on component value spreads.

II. Circuit Analysis in Terms of Voltages and Currents

The solution of an electrical circuit in terms of voltages and currents may be achieved in several ways depending on the variables assumed as unknowns. In lumped-element circuits the variables to be computed are the branch voltages and currents and the voltages between each node and the reference node.

Allowing only one circuit component per branch and applying Ohm's law for each branch, a set of b equations is obtained between vectors $\boldsymbol{V}_b$ and $\boldsymbol{I}_b$ of branch voltages and currents at the b branches:

$$\begin{array}{cc} & \begin{array}{cc} \leftarrow b \rightarrow & \leftarrow b \rightarrow \end{array} \\ b \updownarrow & \begin{bmatrix} \boldsymbol{Y}_b & -\boldsymbol{Q}_I \\ -\boldsymbol{Q}_V & \boldsymbol{Z}_b \\ \boldsymbol{\mu} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\beta} \end{bmatrix} \end{array} \cdot \begin{bmatrix} \boldsymbol{V}_b \\ \boldsymbol{I}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_0 \\ \boldsymbol{E}_0 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \qquad (1)$$

where $\boldsymbol{E}_0$ and $\boldsymbol{J}_0$ are the vectors of independent voltage and current generators; the $\boldsymbol{Q}_V$ and $\boldsymbol{Q}_I$ matrices are made up of rows whose elements are all 0 except for the 1 in the entry relative to the branch to which the row refers. The four groups of equations derive from branches containing, respectively: y elements, independent current sources, and voltage-controlled current sources; z elements, independent voltage sources, and current-controlled voltage sources; voltage-controlled voltage sources; current-controlled current sources. By way of example, the set of equations describing the circuit in Fig. 1 is

$$\begin{bmatrix} 0 & y_m & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & y_5 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & z_2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & \mu & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ E_{10} \\ 0 \\ 0 \end{bmatrix}.$$

The constraints imposed by the topology supply the other equations necessary to define the circuit completely. Applying the Kirchhoff voltage law to all circuit branches and the Kirchhoff current law to all the circuit nodes, except the reference node, the following two topological relations can be obtained:

$$\boldsymbol{A}\boldsymbol{V}_N - \boldsymbol{V}_b = \boldsymbol{0} \tag{2}$$

$$\boldsymbol{A}^T\boldsymbol{I}_b = \boldsymbol{0} \tag{3}$$

where $\boldsymbol{V}_N$ is the vector of the node voltages, $\boldsymbol{A}$ the branch node incidence matrix [10], and $\boldsymbol{A}^T$ its transpose. System (2) contains b equations, each specifying the nodes between which each branch is connected. Each row of $\boldsymbol{A}$ thus contains 1 and -1 in the entries corresponding to the two nodes, all the other elements being 0. System (3) contains n equations indicating which are the branches connected to each of the n nodes excluding the reference one. Each a_{ij} of $\boldsymbol{A}$ has $\pm$ 1 in the entries corresponding to the branches connected to the node, the sign depending on the branch orientation.

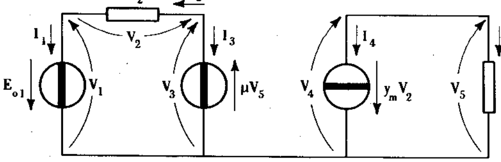


Fig. 1. Circuit chosen as example to show set of equations obtained by applying Ohm's law to each branch.

Collecting together systems (1)–(3) and indicating by $\boldsymbol{1}$ a $(b \times b)$ unit matrix, one obtains the set

$$\boldsymbol{T}\cdot\begin{bmatrix} \boldsymbol{V}_N \\ \boldsymbol{I}_b \\ \boldsymbol{V}_b \end{bmatrix} = \begin{array}{c|ccc} & \overset{n}{\longleftrightarrow} & \overset{b}{\longleftrightarrow} & \overset{b}{\longleftrightarrow} \\ \hline b\updownarrow & \boldsymbol{A} & \boldsymbol{0} & -\boldsymbol{1} \\ n\updownarrow & \boldsymbol{0} & \boldsymbol{A}^T & \boldsymbol{0} \\ & \boldsymbol{0} & -\boldsymbol{Q}_I & \boldsymbol{Y}_b \\ b & \boldsymbol{0} & \boldsymbol{Z}_b & -\boldsymbol{Q}_V \\ & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\mu} \\ & \boldsymbol{0} & \boldsymbol{\beta} & \boldsymbol{0} \end{array}\cdot\begin{bmatrix} \boldsymbol{V}_N \\ \boldsymbol{I}_b \\ \boldsymbol{V}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{J}_0 \\ \boldsymbol{E}_0 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{4}$$

where $\boldsymbol{T}$ is the tableau matrix of the circuit [11].

The application of sparse-matrix techniques to the circuit equations of system (4) is the sparse-tableau method, which has recently met with much approval, particularly for the analysis of nonlinear circuits in the time domain.

A method based on the solution of a system with a smaller number of equations consists in assuming only the node voltage vector $\boldsymbol{V}_N$ as unknown and in expressing system (3) in terms of these and of branch admittances:

$$\boldsymbol{Y}\boldsymbol{V}_N = \boldsymbol{J}_{N0} \tag{5}$$

where $\boldsymbol{J}_{N0}$ is the node impressed current vector and $\boldsymbol{Y}$ the nodal admittance matrix. The kth equation derives from the application of Kirchhoff's current law to node k. Referring to the circuit in Fig. 2(a), one gets, for example, the following equation for node 2:

$$-y_2V_1 + (y_2 + y_3 + y_8)V_2 - y_3V_4 = J_{10}. \tag{6}$$

$\boldsymbol{Y}$ may be obtained by means of the following rules easily deduced from (3).

1) Each diagonal term y_{ii} is the sum of all the admittances connected to node i.

2) Each off-diagonal term y_{ij} is the negative sum of all the transadmittances relative to current generators connected to node i, controlled by node-j voltage, and of all the admittances connected between node i and j.

The nodal admittance matrix does not lend itself to describing circuits including current-dependent current generators and dependent or independent voltage generators except by using special artifices such as the introduction of extra dummy nodes and dummy components [12]–[14]. By way of example, Fig. 3 shows that a current-dependent voltage generator may be represented by introducing two extra nodes and positive and negative impedances [12]. These artifices, however, besides pro-

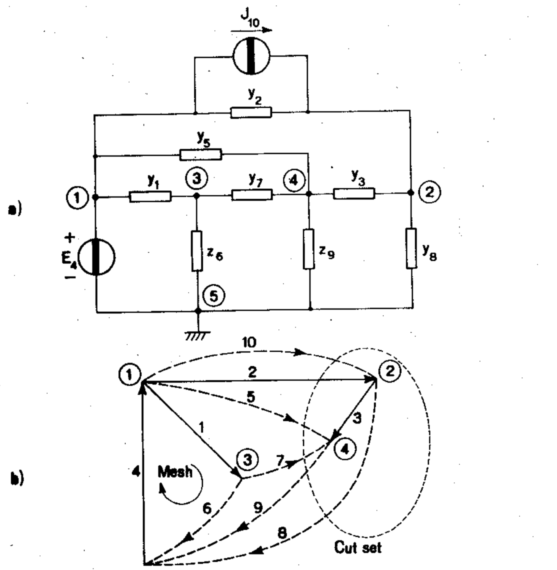
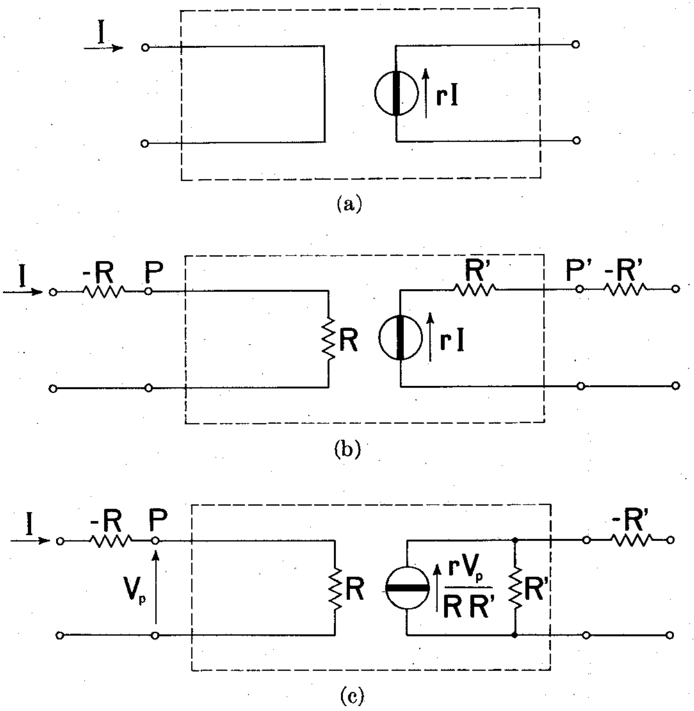


Fig. 2. (a) Lumped-element circuit. (b) Lumped-element circuit graph.



(a)

(b)

(c)

Fig. 3. Current-dependent voltage generator and its modeling with dummy nodes and components.

ducing an increase in the Y-matrix size, cause the presence of coefficients equal to 0 on the main diagonal; this second fact, as we shall see later, must be kept in mind when solving system (5). Moreover, roundoff errors might arise in the solution of (5) when, for example, the voltages of two nodes connected by one or more branches assume values very close to each other or when the Y coefficients assume widely different values.

These inconveniences are not manifested with the previously described tableau method. For the same reasons the modified tableau method [15], [16] and the mixed method [17], which are described hereunder, are, in some cases, preferable. In both these methods, in order to describe the topology, circuit branches are classified as tree branches and links.[1] Each link identifies a mesh composed of the link itself and those specific tree branches required to close the circuit. Each tree branch identifies a cut set that consists of this tree branch itself and the set of links cut by a closed line [see Fig. 2(b)]. Kirchhoff's voltage law applied to the meshes and Kirchhoff's current law applied to the cut sets give the topological relations

$$[C \quad 1_l]\begin{bmatrix} V_t \\ V_l \end{bmatrix} = 0 \qquad (7)$$

$$[1_t \quad -C^T]\begin{bmatrix} I_t \\ I_l \end{bmatrix} = 0 \qquad (8)$$

where V_t, I_t, and V_l, I_l are the voltage and current vectors of the tree branches and links, respectively; 1_l and 1_t are unit matrices, C is the branch-mesh matrix in which $c_{ij} = \pm 1$ if the branch j is in the mesh defined by link i, the sign depending on the branch and mesh orientation; $c_{ij} = 0$ if the tree branch is not in the mesh i; C^T is the transpose of C.

Separating the variables relative to the tree branches and the links, system (1) may be rewritten in the following form:

$$[H_1 \quad H_2]\begin{bmatrix} I_t \\ V_l \\ V_t \\ I_l \end{bmatrix} = [K] \qquad (9)$$

where the meaning of the symbols H_1, H_2, and K can be deduced immediately from (1).

Collecting together the topological systems (7), (8), and the branch constitutive relations expressed by (9), one obtains the following system:

$$\left[\begin{array}{cc:cc} 1_t & 0 & 0 & -C^T \\ 0 & 1_l & C & 0 \\ \hline \multicolumn{2}{c:}{H_1} & \multicolumn{2}{c}{H_2} \end{array}\right]\begin{bmatrix} I_t \\ V_l \\ \hline V_t \\ I_l \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \hline K \end{bmatrix} \qquad (10)$$

whose coefficient matrix is the modified tableau.

The method based on this approach, which has recently been proposed, allows shorter computation time in its authors' opinion [16] compared to that required by the complete tableau, without loss of solution accuracy.

[1] A tree is made up of a set of branches that, in the circuit graph, make it possible to reach all the nodes without forming closed paths.

Another method requiring tree search and the consequent subdivision of the circuit branches into tree branches and links is the well-known mixed method on which ECAP II program is based [17]. The circuit branches are subdivided into two types, y and z: generally, y type with low admittance values and z type with low impedance values. A tree is built up by choosing the circuit branches in the following order: independent voltage generators, dependent voltage generators, y branches, and z branches. Independent voltage generators may not be chosen as links, nor current generators as tree branches, since these choices might imply a violation of the Kirchhoff laws. Thus the branches are classified in four categories: tree admittance branches, tree impedance branches, admittance links, and impedance links. The tree admittance branch voltages and the impedance link currents are assumed to be independent variables, and the circuit equations are written by applying the Kirchhoff current law for the cut sets identified by the tree admittance branches, and the Kirchhoff voltage law for the meshes identified by the impedance links.

By way of example for the cut set identified by branch 2 and for the mesh identified by link 6 in the circuit of Fig. 2(b), the following equations can be written:

$$\begin{gathered} y_2V_2 + y_5(V_2 + V_3) + y_7(V_2 + V_3 - V_1) - I_8 - I_9 = J_{10} \\ V_1 + Z_6I_6 = -E_4. \end{gathered} \tag{11}$$

For the complete circuit the system can be written

$$\begin{bmatrix} \boldsymbol{Y}_t & \boldsymbol{N}_{yz} \\ \boldsymbol{N}_{zy} & \boldsymbol{Z}_l \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{V}_{yt} \\ \boldsymbol{I}_{zl} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{yt} \\ \boldsymbol{E}_{zl} \end{bmatrix} \tag{12}$$

where $\boldsymbol{V}_{yt}$ and $\boldsymbol{I}_{zl}$ are, respectively, the tree admittance branch voltage vector and the impedance link current vector; $\boldsymbol{Y}_t$ and $\boldsymbol{N}_{yz}$ are the coefficient matrices deriving from the cut-set equations, $\boldsymbol{N}_{zy}$ and $\boldsymbol{Z}_l$ the coefficient matrices from the mesh equations; $\boldsymbol{J}_{yt}$ is the vector of the cut-set equivalent independent current generators, and $\boldsymbol{E}_{zl}$ the vector of the mesh equivalent independent voltage generators.

System (12) cannot generally be obtained by simple rules without searching the tree or using the topological matrices. This is not, however, the proper place to go into greater detail and the reader is referred to the works by Branin [10], [17].

III. Circuit Analysis in Terms of Wave Variables

The behavior of a microwave circuit may also be described in terms of the normalized wave variables at the ports of the component multiports.[2] Given a circuit with m component multiports, for each component with scattering matrix $\boldsymbol{S}_i$ the vectors $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ of incident and reflected waves at its n_i ports are related by the following equation:

$$\boldsymbol{b}_i = \boldsymbol{S}_i\boldsymbol{a}_i. \tag{13}$$

An independent generator is described, instead, by the relation

$$b_g = S_g a_g + c_g \tag{14}$$

where c_g is the impressed wave.

Considering all the components, we have a system of linear equations whose matrix form is

$$\boldsymbol{b} = \boldsymbol{S}\boldsymbol{a} + \boldsymbol{c} \tag{15}$$

where

$$\boldsymbol{a} = \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \vdots \\ \boldsymbol{a}_m \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_m \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \vdots \\ \boldsymbol{c}_m \end{bmatrix}$$

and

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{S}_1 & \cdot & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \boldsymbol{S}_i & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & \cdot & \boldsymbol{S}_m \end{bmatrix}. \tag{16}$$

The connections between the m components impose constraints on the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$; in fact, incident and reflected waves at ports j and k connected together must satisfy the following relations (see Fig. 4):

$$a_j = b_k \qquad a_k = b_j$$

if the normalization numbers are the same. The relations for all the circuit component ports may be put in the form

$$\boldsymbol{b} = \boldsymbol{\Gamma}\boldsymbol{a} \tag{17}$$

where $\boldsymbol{\Gamma}$ is the connection matrix whose elements are all null except the 1's in the entries corresponding to pairs of adjacent ports [22]–[25].[3]

Substituting (17) into (15) and solving for $\boldsymbol{a}$ by setting

$$\boldsymbol{W} = \boldsymbol{\Gamma} - \boldsymbol{S} \tag{18}$$

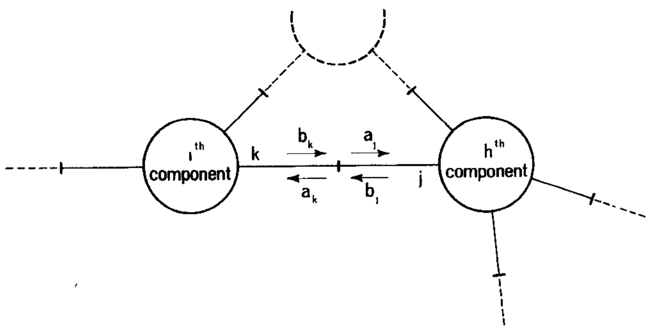


Fig. 4. Constraints imposed by connection between adjacent ports.

[2] It is always possible to transform a lumped-element network into a circuit made up of multiports connected through pairs of ports [18]–[21].

[3] It would be easy to define the connection matrix, also, when different normalization numbers are chosen for adjacent ports [26].

we have

$$\boldsymbol{W a} = \boldsymbol{c} \tag{19}$$

where $\boldsymbol{c}$ is the vector of the impressed waves and $\boldsymbol{W}$ is the connection scattering matrix. Its main diagonal elements are the reflection coefficients at component ports; the other $\boldsymbol{W}$ elements are all null except those relative to ports belonging to the same component (the transmission coefficients) and those relative to ports connected together (the $\boldsymbol{\Gamma}$ elements). Numerical values of nonzero elements change with the frequency except the 1's indicating connections.

IV. Multiport Connection Methods

Considerable reduction of computing time and memory space requirements in analyzing large circuits can be achieved by dividing the circuit into subcircuits and calculating for each of them one of its port matrices. Subsequently, the subcircuits are interconnected and, finally, the matrix relative to the ports of the complete circuit is determined.

A method based on these principles has been proposed by Murray-Lasso for the BELLNAP program [12], [27] employing the indefinite admittance matrix, which is simply the nodal admittance matrix defined in Section II, but referred to a datum node outside the circuit. The nodes of every subcircuit are divided into internal and external. By eliminating the variables relative to the internal nodes, one obtains the indefinite admittance matrix relative to the external nodes.

Every indefinite admittance matrix is subdivided as follows:

$$\begin{bmatrix} \boldsymbol{I}_e \\ \\ \boldsymbol{I}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y}_{ee} & \boldsymbol{Y}_{ei} \\ \\ \boldsymbol{Y}_{ie} & \boldsymbol{Y}_{ii} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_e \\ \\ \boldsymbol{V}_i \end{bmatrix} \tag{20}$$

where $\boldsymbol{I}_e$ and $\boldsymbol{V}_e$ are the vectors of the currents and voltages relative to the external nodes, and $\boldsymbol{I}_i$ and $\boldsymbol{V}_i$ are those relative to the internal nodes. Since no generators are connected to the internal nodes, we can take $\boldsymbol{I}_i = 0$ and, solving with respect to $\boldsymbol{V}_i$ in the second set of equations of (20), one has

$$\boldsymbol{V}_i = -\boldsymbol{Y}_{ii}^{-1}\boldsymbol{Y}_{ie}\boldsymbol{V}_e. \tag{21}$$

Substituting (21) in the first set of (20) one gets

$$\boldsymbol{I}_e = \boldsymbol{Y}_e\boldsymbol{V}_e \tag{22}$$

with

$$\boldsymbol{Y}_e = \boldsymbol{Y}_{ee} - \boldsymbol{Y}_{ei}\boldsymbol{Y}_{ii}^{-1}\boldsymbol{Y}_{ie} \tag{23}$$

which represents the indefinite admittance matrix with respect to the external ports of each subcircuit. When the subcircuits are connected together, the indefinite admittance matrix for the external nodes of the complete circuit is calculated using the same procedure.

Some programs that use the S-matrix formulation [23], [24] are also based on the same principle, that is, on determining the port matrix of the circuit obtained by interconnecting a number of components. These programs are mainly microwave circuit oriented.

Evaluation of the scattering matrix of a circuit composed of multiports connected through pairs of ports in terms of component S parameters is effected by dividing component ports into connected and nonconnected ones. Then by partitioning system (15) and letting $\boldsymbol{c} = 0$, since independent generators will be considered connected to external ports, one obtains

$$\begin{bmatrix} \boldsymbol{b}_p \\ \\ \boldsymbol{b}_c \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}_{pp} & \boldsymbol{S}_{pc} \\ \\ \boldsymbol{S}_{cp} & \boldsymbol{S}_{cc} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_p \\ \\ \boldsymbol{a}_c \end{bmatrix} \tag{24}$$

where $\boldsymbol{a}_p$, $\boldsymbol{b}_p$ and $\boldsymbol{a}_c$, $\boldsymbol{b}_c$ are the normalized waves at the p external ports and at the c internal connected ones. The constraints between connected ports yield

$$\boldsymbol{b}_c = \boldsymbol{\Gamma a}_c \tag{25}$$

where $\boldsymbol{\Gamma}$ is the connection matrix previously defined in Section III. From (24) and (25), by first eliminating $\boldsymbol{b}_c$, one obtains

$$\boldsymbol{a}_c = (\boldsymbol{\Gamma} - \boldsymbol{S}_{cc})^{-1}\boldsymbol{S}_{cp}\boldsymbol{a}_p. \tag{26}$$

Then, by eliminating $\boldsymbol{a}_c$,

$$\boldsymbol{b}_p = \boldsymbol{S}_p\boldsymbol{a}_p \tag{27}$$

where

$$\boldsymbol{S}_p = \boldsymbol{S}_{pp} + \boldsymbol{S}_{pc}(\boldsymbol{\Gamma} - \boldsymbol{S}_{cc})^{-1}\boldsymbol{S}_{cp} \tag{28}$$

is the scattering matrix of the microwave circuit at its p external ports. It can be determined in terms of the S matrices of its component multiports by relation (28). For a circuit comprising many components, the computing time is, however, too long due primarily to the inversion of a matrix with order equal to the number of connected ports.

Great running time reduction is obtained in the program SCAMAT [23] by connecting the m component multiports of the complete circuit two at a time and determining the S matrix of the resulting subcircuit every time. In such a manner, after (m-1) applications of (28), the S matrix of the complete network is computed. The number of arithmetical operations necessary to compute the S matrix for a given circuit topology depends on the order according to which the components are connected to each other.

In order to minimize computing time a connection sequence based on an easily programmable rule has been proposed that implies a number of algebraic operations at all times very near to the minimum, whatever the circuit topology. The rule consists of connecting every time the two components whose resultant multiport has the smallest number of ports.

The application of this method to circuits including lumped elements is done by transforming every component having n terminals into a multiport with n ports. The transformation is carried out by associating a ground terminal with each component terminal, and these terminal pairs are considered as ports [19], [20]. When a component has one of its n terminals connected to ground, it is considered a multiport having (n-1) ports; all the ports in such a case derive by pairing the ground terminal with the other (n-1) terminals. The multiports thus obtained are connected together to form the network. Since the program connects only pair of ports, auxiliary multiports that transform every node with k branches into a multiport with k ports are introduced. These auxiliary multiports are parallel tees (Fig. 5) introduced at the rate of (k-2) tees for every node connecting k branches. The introduction of the auxiliary multiports and the transformation of every n-terminal component into an (n-1) port or n port is made automatically by the program through interpretation of the input data.

In addition to the characterizing parameters, the data for every component indicate to which nodes its terminals are connected. The interconnection modalities between real and auxiliary components are then found and the connection sequence stated. After this preliminary phase, which is performed only once for a given circuit with m components, the execution phase begins and consists of applying (28) (m-1) times for every frequency point.

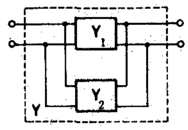
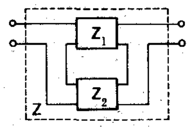
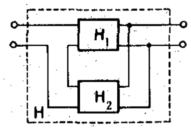
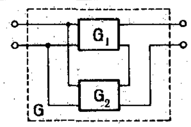
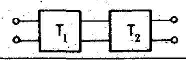
A similar formulation has been adopted in the program General [9]. It requires the user to decompose the circuit into wire-coupling multiports and original components, then a formula similar to (28) is applied repeatedly, starting from the last subnetwork. The circuit decomposition required appears, however, too cumbersome for the user.

Determination of the matrix of the multiport resulting from the connection of two components can be effected with more simple rules than (23) or (28) if matricial representation of the components is suitably chosen. Indeed, rules that determine the matrix of the multiport resulting from the connection of two multiport components are well known [28]. For the various types of connections, they require the components to be defined according to different matricial representations. The simplicity of these connection rules has brought about their widespread use in a number of general and special-purpose programs [9], [29]–[35]. In these programs the matrix relative to the external ports of the complete circuit is determined by connecting the components in twos, each time determining the matrix of the resulting multiports in accordance with rules codified in library routines. Table I shows, by way of example, some of the most widely used rules in the case of connections between 2-port components. It may, however, happen that the matricial representation required by the type of the connection to be made may not exist for some components. Besides, the method may call for numerous matrix transformations, which sometimes cause a loss of accuracy.

The types of connections and the sequence in which they are made are generally decided by the user. This last, apparently difficult, operation is considerably simplified by using high-level languages for circuit description. A particularly interesting one is MARTHA, proposed by Penfield [21], [35]. It consists of defining a certain number of wiring operators by means of which connections between components may be simply and concisely identified and described. The order of writing the operators establishes the order for carrying out the connections by the program. By way of example the simple circuit in Fig. 6 is described using the wiring operators shown in Table I. The number of wiring operators permitted by

TABLE I

VARIOUS TYPES OF CONNECTION BETWEEN TWO PORTS—MATRIX OPERATIONS INVOLVED AND MARTHA WIRING OPERATORS

CONNECTION TYPE		Matrix Operation	MARTHA Wiring Operator
Parallel		$Y = Y_1 + Y_2$	WPP
Series		$Z = Z_1 + Z_2$	WSS
Series-parallel		$H = H_1 + H_2$	WSP
Parallel-series		$G = G_1 + G_2$	WPS
Cascade		$T = T_1 \times T_2$	WC

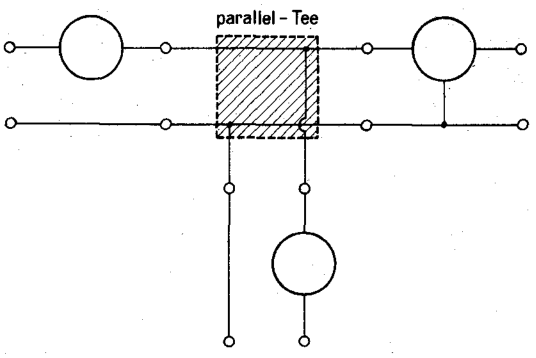


Fig. 5. Parallel tee as auxiliary multiport.

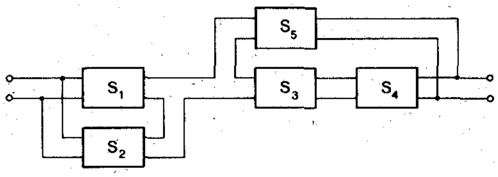


Fig. 6. Example of circuit description by means of MARTHA language. MARTHA description: (s_1 WPS s_2)WC((s_3 WC s_4)WSP s_5).

MARTHA is such as to allow considerable generality in the type of circuit that may be analyzed.

V. Sparse-Matrix Approach

Many methods have been proposed for the solution of linear systems with sparse coefficient matrices [36]-[39]. Evaluation of their efficiency is not, however, one of the aims of this work, so the description, though brief, is limited to the LU-factorization method [3], which is most frequently applied in the field of electric networks. According to this method the solution of the system $\boldsymbol{Mx} = \boldsymbol{c}$ is carried out first by factoring $\boldsymbol{M}$ into the product of two matrices, a lower triangular matrix $\boldsymbol{L}$ and an upper triangular matrix with 1's on the diagonal $\boldsymbol{U}$:

$$\boldsymbol{M} = \boldsymbol{LU} = \begin{bmatrix} l_{11} & \cdot & 0 & \cdot & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ l_{k1} & \cdot & l_{kk} & \cdot & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ l_{i1} & \cdot & l_{ik} & \cdot & l_{ii} & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ l_{n1} & \cdot & l_{nk} & \cdot & l_{ni} & \cdot & l_{nn} \end{bmatrix} \cdot \begin{bmatrix} 1 & \cdot & u_{1k} & \cdot & u_{ij} & \cdot & u_{1n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 1 & \cdot & u_{kj} & \cdot & u_{kn} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & \cdot & 1 & \cdot & u_{jn} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & 0 & \cdot & 0 & \cdot & 1 \end{bmatrix}. \qquad (29)$$

The elements of $\boldsymbol{L}$ and $\boldsymbol{U}$ are determined by recurrence formulas as follows:

$$\begin{aligned} l_{ik} &= m_{ik} - \sum_{\mu=1}^{k-1} l_{i\mu}u_{\mu k}, && i \geq k \\ u_{kk} &= 1 \\ u_{kj} &= (m_{kj} - \sum_{\mu=1}^{k-1} l_{k\mu}u_{\mu j})/l_{kk}, && j > k \end{aligned} \qquad (30)$$

for $k = 1,2,\cdots,n$, where n is the matrix dimension. Then by solving the two triangular systems

$$\boldsymbol{Ly} = \boldsymbol{c} \qquad \boldsymbol{Ux} = \boldsymbol{y} \qquad (31)$$

using "forward elimination" on $\boldsymbol{Ly} = \boldsymbol{c}$ and "back substitution" on $\boldsymbol{Ux} = \boldsymbol{y}$, the unknown vector $\boldsymbol{x}$ is obtained [3]. This is performed by applying the recurrence formulas:

$$\begin{aligned} y_j &= (c_j - \sum_{\mu=1}^{j-1} l_{j\mu}y_\mu)/l_{jj}, && j = 1,2,\cdots,n \\ x_j &= y_j - \sum_{\mu=j+1}^{n} u_{j\mu}x_\mu, && j = n,n-1,\cdots,1. \end{aligned} \qquad (32)$$

All the elements of $\boldsymbol{L}$ and $\boldsymbol{U}$ can be stored in a matrix:

$$\boldsymbol{Q} = \boldsymbol{L} + \boldsymbol{U} - \boldsymbol{1} \qquad (33)$$

$\boldsymbol{1}$ being unit matrix. Any q_{jk} of $\boldsymbol{Q}$ is 0 if both m_{jk} of $\boldsymbol{M}$ and all the products $q_{ji}q_{ik}$, $1 \leq i \leq \min\{j-1,k-1\}$ are 0. Therefore, the number of nonzeros in $\boldsymbol{Q}$ depends on the ordering of rows and columns in $\boldsymbol{M}$, as is discussed later. The nonzeros in $\boldsymbol{Q}$ in the entries corresponding to zeros in $\boldsymbol{M}$ are currently called "fills."

A great reduction in execution time is obtained with the reduced Crout method [40], according to which only the nonzero operands are considered in computing the nonzeros of $\boldsymbol{Q}$. Two different strategies are used to this end. The first consists of creating a sequence of indices and pointers that establish the type of operation to be carried out and the position of the operands in the tables where the operands are stored [41]-[43]. The second method consists of generating a no-loop no-branch code containing the instructions to perform all the necessary operations.

The second method is very fast in execution but requires large memory space due to the need to store the generated code and high central processing unit time to generate the code itself. On the other hand, the first method, though slower in execution, requires less time for preparation of the sequence of indices and pointers and much less memory space. Memory requirements make the code generation method unsuitable for very large networks like power distribution networks [44], while it is widespread for electronic circuit analysis both with lumped [11], [16] and distributed elements [25].

The solution code may be generated either in machine language or in a high-level language such as Fortran. In the first case it requires shorter execution time but has the inconvenience of being machine dependent. Generation time, execution time, and storage requirements depend on the length of the code, which depends in turn on the number of nonzeros in $\boldsymbol{Q}$ and $\boldsymbol{c}$ and, therefore, on the number and position of nonzeros in $\boldsymbol{M}$.

With the aim of minimizing code length, numerous reordering algorithms of the $\boldsymbol{M}$-matrix coefficients have been proposed. In all these algorithms the accuracy problem must be kept in mind; this requires the rows and the columns of the $\boldsymbol{M}$ matrix to be rearranged so that the diagonal elements are nonzero, since they are used as divisors in (30) and (32). To this end, depending on the circuit matrix adopted, some rules are established to avoid pivoting on the critical elements.

For circuits described by the nodal admittance matrix ($\boldsymbol{M} \equiv \boldsymbol{Y}$), the diagonal term m_{ii} represents the sum of admittances connected to the ith node and, therefore, it is different from 0 except in particular cases.[4] For this reason the reordering algorithms for $\boldsymbol{Y}$ matrix keep these terms on the diagonal while ordering the rows and columns to minimize the code length. This is equivalent to circuit node renumbering. This procedure is based on the supposition that nonzero elements on the $\boldsymbol{Y}$ diagonal are a sufficient guarantee that the elements on the $\boldsymbol{Q}$ diagonal

[4] When certain components are modeled by introducing extra dummy nodes or dummy components, the corresponding diagonal positions cannot be pivoted [13].

will be nonzero. The node renumbering algorithms most frequently used for the Y matrix are reported in the following [41], [43], [44].

1) The nodes are ordered so that the number of nonzeros in the corresponding rows are nondecreasing.

2) At every jth step the node is selected whose corresponding row has fewer nonzeros on the right of the $(j - 1)$th entry.

3) Select as jth step the node whose corresponding row and column, when used in factorization, would cause the smallest number of fills.

For the connection scattering matrix $\boldsymbol{W}$, on the other hand, as described in Section III, the diagonal elements represent the reflection coefficients of the circuit component ports and, therefore, being very near to 0 in matched conditions, cannot be chosen as pivots. However, in system (19) every row of $\boldsymbol{W}$ contains the constant 1, deriving from $\boldsymbol{\Gamma}$, which could be an ideal pivot because it allows great precision, independent of frequency, and, at the same time, divisions are avoided. In reality, roughly half the 1's are modified in the course of the factorization process, but rarely do the modified values become 0 and only in anomalous cases. This method has been adopted in a recently realized program for microwave circuit analysis [25] with the following ordering strategy: the pair of rows relative to adjacent ports are considered together and ordered so that each pair has a number of nonzeros not greater than that of the successive one; in every pair the row with fewer nonzeros precedes the other; the columns are then ordered to place all the 1's of $\boldsymbol{\Gamma}$ on diagonal.

The programs adopting the code generation techniques are, therefore, structured in two phases. In the first, after data input and interpretation, coefficient matrix row and column ordering is established according to the algorithm used and the solution code is generated. In the second phase the matrix coefficient values are determined for every frequency point and then the code is executed, giving the unknown variable values in terms of which the requested network functions are determined.

The time required for the first phase is generally very high, but it should be remembered that for a given circuit the code is generated only once, while the code itself is executed many times and may be repeated at any later time if it is stored in a permanent file. Analogous considerations may be done for the structure of the programs creating a sequence of pointers and indices.

However, for very simple circuits not requiring many analysis repetitions, it may be more convenient both for computing time and memory space to use programs of the type described in Section IV, whose preliminary phase is less complex. In order to supply quantitative information the circuit in Fig. 7, representing a thin-film strip-line branching filter [25], has been analyzed with two programs realized by the authors: the BMT program adopting a sparse-matrix technique and the SCAMAT program effecting connection of multiports two at a time. The relative computing time versus the number of executions is shown in Fig. 8.

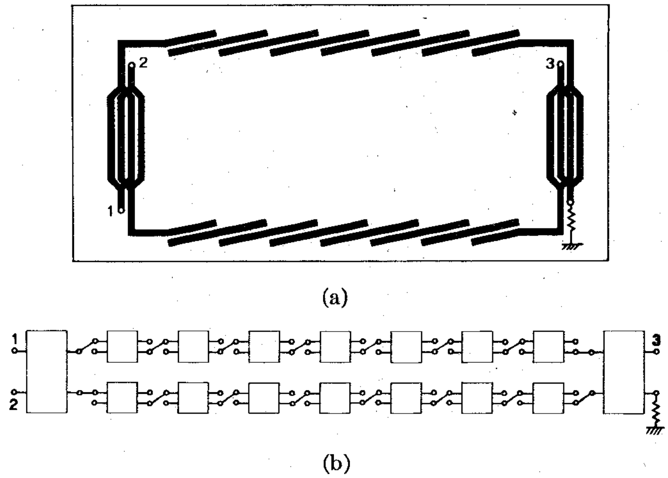


Fig. 7. (a) Thin-film strip-line branching filter. (b) Description for analysis program.

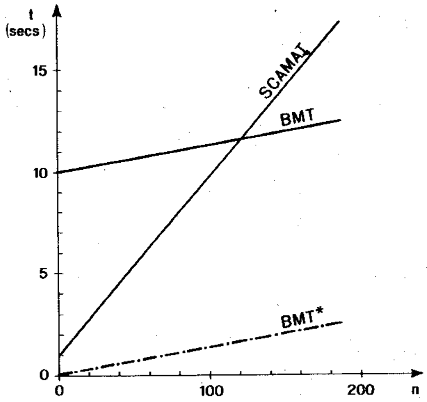


Fig. 8. Computation time required by the two programs SCAMAT and BMT versus number of analyses effected for branching filter in Fig. 6. Dotted line refers to case for which execution code has already been generated and compiled.

This figure shows that for the circuit analyzed, the first phase of the BMT program is ten times slower, while the second phase is seven times faster. Then, when more than 120 executions (different frequency points or parameters values) are required, the code generation method is more convenient. However, when the code is already available, the dotted line must be considered and the convenience of the BMT program is evident.

VI. First- and Second-Order Sensitivity

A performance generally required of circuit analysis programs is the computation of the partial derivatives (sensitivities) of the network functions with respect to the variables on which the component parameters depend. It is useful for the designer to know them both in order to have an indication of circuit criticality relative to the variables themselves, and in order to utilize the sensitivities in calculating particular network functions (group delay, for example) and in optimization processes.

Simple determination of the sensitivities by means of the variational method may cause computation errors that depend on the intrinsic precision of the program and the increment chosen for the independent variables. Computing time may, besides, be too high when derivatives with respect to many parameters are requested. It is for these reasons that other techniques are used in the important programs; among these the adjoint circuit method has met with much favor in recent years [45]–[61]. It must, however, be noted that the most convenient technique depends on the analysis method used by the program. The best known among recently proposed methods are illustrated below, and a comparison is made between them in relation to the adopted method of analysis.

Where circuit analysis consists of the solution of a system of the type

$$\boldsymbol{M}\boldsymbol{x} = \boldsymbol{c} \tag{34}$$

where $\boldsymbol{x}$ is the circuit variable vector (voltages, currents, or normalized waves) and $\boldsymbol{M}$ is a matrix that takes topology and circuit composition into account (nodal admittance, matrix, mixed matrix, connection scattering matrix, etc.), computation of the partial derivatives of the vector $\boldsymbol{x}$ with respect to any single parameter p on which the circuit components depend may be carried out by differentiating (34):

$$\boldsymbol{M}\frac{\partial \boldsymbol{x}}{\partial p} = -\frac{\partial \boldsymbol{M}}{\partial p}\boldsymbol{x} + \frac{\partial \boldsymbol{c}}{\partial p}. \tag{35}$$

This equation may be solved for $\partial \boldsymbol{x}/\partial p$ by evaluating the right-hand side vector:

$$\boldsymbol{c}' = -\frac{\partial \boldsymbol{M}}{\partial p}\boldsymbol{x} + \frac{\partial \boldsymbol{c}}{\partial p} \tag{36}$$

which may be calculated when $\partial \boldsymbol{M}/\partial p$ and $\partial \boldsymbol{c}/\partial p$ are known and $\boldsymbol{x}$, by solving (34), has been determined.

Determination of vector $\partial \boldsymbol{x}/\partial p$ by means of (35) requires only forward and back (FB) substitution since the *LU* factorization already done for the solution of (34) may be utilized. If sparse-matrix techniques are adopted, FB substitution may be performed by executing the same code (or the same set of pointers and indices) as that generated for the analysis, provided that the code has been generated without considering the sparsity of vector $\boldsymbol{c}$, since its sparseness structure is generally different from that of $\boldsymbol{c}'$.

This direct method permits determination of the sensitivity of all the elements of $\boldsymbol{x}$ with respect to a single parameter p, and the computing effort involved is only slightly higher than that of the original analysis. If the sensitivities of all elements of $\boldsymbol{x}$ are requested with respect to several parameters, as many FB substitutions are required as there are parameters.

If the sensitivities of only one element of $\boldsymbol{x}$ are requested with respect to many parameters, it may be convenient to adopt the adjoint network method proposed by Director and Rohrer [45], [46], deriving it from Tellegen's theorem [62], [63]. This method has in recent years been subject of numerous studies [45]–[61] and the reader is referred to them for detailed information. Here, a method is presented that provides (in certain cases) results similar to the adjoint method. The derivation is quite straightforward and involves only matrix operations [25], [65].

Indicating by

$$\boldsymbol{\gamma}_j^T = [0 \quad 0 \cdots 0 \quad 1 \quad 0 \cdots 0] \tag{37}$$

a row vector whose elements are all null except a 1 in position j, we have

$$\frac{\partial x_j}{\partial p} = \boldsymbol{\gamma}_j^T \frac{\partial \boldsymbol{x}}{\partial p}. \tag{38}$$

Keeping (35) in mind this becomes

$$\frac{\partial x_j}{\partial p} = -\boldsymbol{\gamma}_j^T \boldsymbol{M}^{-1}\left(\frac{\partial \boldsymbol{M}}{\partial p}\boldsymbol{x} - \frac{\partial \boldsymbol{c}}{\partial p}\right) = \boldsymbol{\xi}_j^T\left(\frac{\partial \boldsymbol{M}}{\partial p}\boldsymbol{x} + \frac{\partial \boldsymbol{c}}{\partial p}\right) \tag{39}$$

having indicated by

$$\boldsymbol{\xi}_j = (\boldsymbol{\gamma}_j^T \boldsymbol{M}^{-1})^T = (\boldsymbol{M}^T)^{-1}\boldsymbol{\gamma}_j \tag{40}$$

the vector of the unknowns of a system of equations whose coefficient matrix is equal to the transpose of that characterizing the circuit being analyzed and having $\boldsymbol{\gamma}_j$ as its right-hand side vector.

Supposing for simplicity $\partial \boldsymbol{c}/\partial p = 0$, (39) is reduced to

$$\frac{\partial x_j}{\partial p} = -\boldsymbol{\xi}_j^T \frac{\partial \boldsymbol{M}}{\partial p}\boldsymbol{x} \tag{41}$$

which can be used to compute sensitivity of variable x_j with respect to any p parameter when vectors $\boldsymbol{x}$ and $\boldsymbol{\xi}_j$ have been computed by solving systems (34) and (40).

This method, which in the following is called the transpose-matrix method, is in some cases similar to the adjoint method. In fact considering, for instance, the connection scattering matrix (that is, letting $\boldsymbol{M} \equiv \boldsymbol{W}$) we have from (18)

$$\boldsymbol{W}^T = (\boldsymbol{\Gamma} - \boldsymbol{S})^T = \boldsymbol{\Gamma} - \boldsymbol{S}^T$$

being $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}^T$; thus $\boldsymbol{W}^T$ may be interpreted as the connection scattering matrix of a new circuit with the same topology as the one being examined and components whose scattering matrices are the transposes of the corresponding ones in the original circuit. In this case, $\boldsymbol{\xi}_j^T$ represents the vector of the incident waves at the new circuit component ports with excitations stated by $\boldsymbol{\gamma}_j$. The equivalence between this method and the adjoint circuit method is then evident [48].

The same consideration can be made with reference to the nodal admittance matrix. Indeed, $\boldsymbol{\xi}_j$ in this case may be interpreted as the nodal voltage vector of the adjoint circuit, which, as it is known, has a matrix equal to the transpose of the original one and is excited as established by $\boldsymbol{\gamma}_j$ [51].

When the mixed method is adopted for analysis, the transpose of the coefficient matrix in (12) does not coincide with the matrix of the adjoint circuit due to the sign reversals of its off-diagonal submatrices. However, the transpose matrix method can be equally adopted and the results are identical to those derived by means of the adjoint network.[5]

In order to make the reader aware of the computing effort in determining sensitivity with (41), we observe that computation of $\boldsymbol{\xi}_j$ involves only a single FB substitution [14], [64] for the following reason. Since

$$\boldsymbol{M}^T = (\boldsymbol{L}\boldsymbol{U})^T = \boldsymbol{U}^T\boldsymbol{L}^T = \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{U}}$$

it is not necessary to repeat factorization in order to compute $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{U}}$, as these are obtainable through transposition of $\boldsymbol{U}$ and $\boldsymbol{L}$, respectively. When sparse-matrix techniques are used with code generation it should be noted that the FB-substitution code generated for the analysis cannot be utilized; thus a modified code should be generated for the FB substitution applied to $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{U}}$. If, however, $\boldsymbol{M}^T$ has the same sparseness structure as $\boldsymbol{M}$, differing at the most in the values of some nonzero coefficients, then $\boldsymbol{\xi}_j$ can be determined starting from $\boldsymbol{M}^T$ by executing the *LU*-factorization and FB-substitution codes already generated for the analysis. When $\boldsymbol{M}$ represents the connection scattering matrix $\boldsymbol{W}$ and if all the parameters of the components are considered to be different from 0 the analysis code may be used; the same thing takes place with the nodal admittance matrix if we consider as nonzero the coefficients in locations symmetrical to those of the dependent generators. Thus with two complete code executions in addition to the supplementary operations[6] required by (41), the sensitivity of one variable x_j may be computed for many parameters.

A quantitative comparison between this method and the direct one cannot be effected in general. It is, however, noted that the direct method is preferable when the derivatives of many variables with respect to few parameters are required, while the transpose-matrix method is more convenient when the derivatives of few variables with respect to many parameters are to be calculated. However, it is observed that if sparse-matrix techniques are used, the choice between the two methods may be conditioned by other factors such as, for example, general organization of the program.

Calculation of sensitivity by the direct method and the adjoint circuit are not equally convenient when multiport connection methods, described in Section IV, are adopted for the analysis. In this case, in fact, the variables relative to internal nodes or ports of the circuit are not normally calculated. It is, however, always possible to include routines in the program that, on the basis of suitable algorithms [56], [57], [66], make it possible to calculate the voltage and currents or the wave variables relative to the internal ports in terms of component parameters and of the impressed vectors. These routines may, however, bring about a considerable increase in program complexity and thus, considering also that these programs are usually utilized for analyzing circuits that are not too large and that have a particular topology, it may be convenient to do the sensitivity computation by means of variational techniques. These techniques have the advantage of directly supplying the sensitivity of the required network functions, which are often complicated functions of the derivatives $\partial \boldsymbol{x}/\partial p$.

To compute second-order sensitivity, (35) may be differentiated with respect to a new variable q and, for simplicity, supposing $\boldsymbol{c}$ independent of p and q, one obtains the second-order sensitivity expression

$$\frac{\partial^2 \boldsymbol{x}}{\partial p \partial q} = - \boldsymbol{M}^{-1}\left(\frac{\partial^2 \boldsymbol{M}}{\partial p \partial q} x + \frac{\partial \boldsymbol{M}}{\partial p}\frac{\partial \boldsymbol{x}}{\partial q} + \frac{\partial \boldsymbol{M}}{\partial q}\frac{\partial \boldsymbol{x}}{\partial p}\right). \qquad (42)$$

As in the first-order sensitivity case, by introducing vector $\boldsymbol{\gamma}_j$ the following relation is obtained:

$$\frac{\partial^2 x_j}{\partial p \partial q} = - \boldsymbol{\xi}_j{}^T \frac{\partial^2 \boldsymbol{M}}{\partial p \partial q} x - \boldsymbol{\xi}_j{}^T\left(\frac{\partial \boldsymbol{M}}{\partial p}\frac{\partial \boldsymbol{x}}{\partial q} - \frac{\partial \boldsymbol{M}}{\partial q}\frac{\partial \boldsymbol{x}}{\partial p}\right) \qquad (43)$$

which allows computation of the second-order sensitivity of variable x_j with respect to parameters p and q.

The application of (43) in determining the second-order sensitivity is very convenient, especially when the analysis program is based on the generation of the *LU*-factorization and FB-substitution codes. In fact it involves two complete code executions for determining vectors $\boldsymbol{x}$ and $\boldsymbol{\xi}_j$ and two more executions of the FB-substitution code for evaluating $\partial \boldsymbol{x}/\partial p$ and $\partial \boldsymbol{x}/\partial q$. When the second-order sensitivities of the same variable x_j with respect to m different parameters have to be computed, the FB-substitution code must be executed m times. For a more detailed discussion the reader is referred to [78].

VII. Results of Some Analyzed Circuits

The analysis programs based on the methods described in the preceding sections allow determination for every assigned set of component parameter values and for every frequency point the values of the circuit variables considered as unknowns. In terms of these variables, the response functions of the circuit required by the user must be computed. For this reason, programs usually contain library routines for computation of the most common functions such as: voltage and current insertion gain; input and output impedances; loss attenuation; reflection coefficients at circuit ports; etc. It is often interesting also to determine group delay and/or the sensitivities of the above functions with respect to certain parameters; to this end one must also know the partial derivatives of the circuit variables with respect to the parameters themselves. When the functions to be calculated are not contained in the library, it must be possible for the user to insert new specially written routines into the program.

In order to give the reader an idea of the functions that

[5] See Branin [65] for a detailed discussion.

[6] The sparsity of $\partial \mathbf{M}/\partial p$ and $\partial \mathbf{c}/\partial p$ may be taken into account.

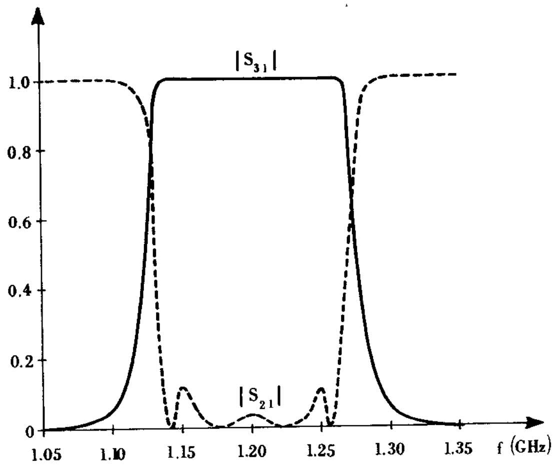


Fig. 9. Computed transmission coefficients S_{31} and S_{21} of branching filter in Fig. 6.

may be requested, the results of the analyses carried out by the authors on some circuits are reported. Numerous other examples of considerable interest are described in the works mentioned in the References.

With reference to the circuit already shown in Fig. 7 representing a branching filter implemented on an alumina ceramic substrate by thin-film technology, the amplitudes of the transmission coefficients S_{31} and S_{21} have been computed and plotted versus frequency in Fig. 9. The analysis was done with the BMT program, which utilizes the connection scattering matrix $\boldsymbol{W}$ to describe the circuit and adopts the sparse-matrix technique with code generation to determine the normalized wave vector. The circuit has been described for the program as shown in Fig. 6(b), connecting port 1 to a matched generator with impressed wave $c_g = 1$, ports 2 and 3 to matched loads, and all the other ports to open-circuit terminations. The transmission coefficients S_{21} and S_{31} coincide, in this case, with the waves b_2 and b_3 reflected by ports 2 and 3:

$$S_{21} = (b_2)_{a_1=1} \qquad S_{31} = (b_3)_{a_1=1}.$$

The condition $a_1 = 1$ is imposed by the generator connected to port 1. The S parameters of the coupled-transmission microstrip have been computed by means of routines [67] associated to the program in terms of the geometric dimensions and electrical characteristics. For the same circuit, using the direct method illustrated in Section VI, computations have been made for group delay:

$$\tau_{31} = \frac{\partial \beta_{31}}{\partial \omega} = \mathrm{Im}\left(\frac{1}{b_3}\frac{\partial b_3}{\partial \omega}\right)$$

where $\beta_{31} = \measuredangle b_3$ and the magnitude sensitivity M_ϵ with respect to the permittivity ϵ of the ceramic substrate is

$$M_\epsilon = \frac{\partial \ln |b_3|}{\partial \ln \epsilon} = \epsilon \,\mathrm{Re}\left[\frac{1}{b_3}\frac{\partial b_3}{\partial \epsilon}\right].$$

The computed results are shown in Fig. 10.

The circuit in Fig. 11 has been analyzed by the SCAMAT program, which is based on the multiport connection method after transformation of the circuit components into multiports by means of the introduction of auxiliary multiports. It represents a broad-band transistor amplifier for a 360-channel frequency division multiplexing (FDM) system. Transistors and transformers have been characterized by measured parameters. The computed voltage insertion gain G and return losses ρ_i and ρ_0 at input and output ports are shown in Fig. 12. In the same figure the values measured at several frequency points are also given; the discrepancy between computed and measured values are due to the inability of the instrument to measure very high values of return loss.

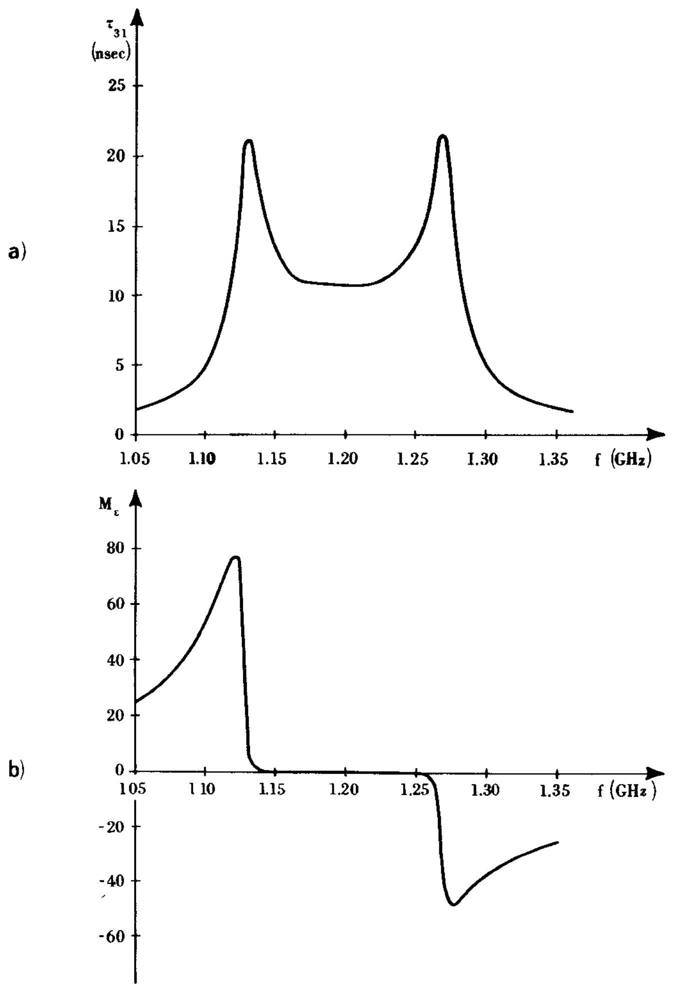


Fig. 10. (a) Group delay τ_{31} and (b) sensitivity M_ϵ, with respect to permittivity ϵ of transmission coefficient S_{31} of filter in Fig. 6 versus frequency.

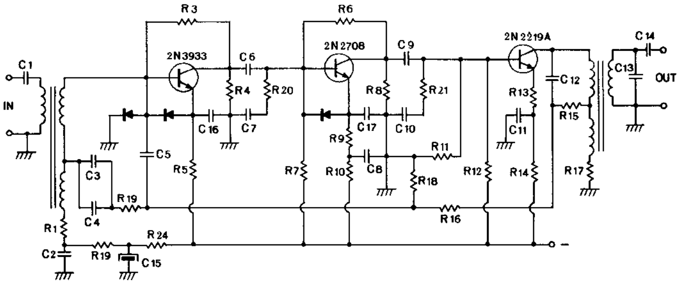


Fig. 11. Broad-band transistor amplifier for 960-channel FDM system (courtesy of Telettra Laboratory).

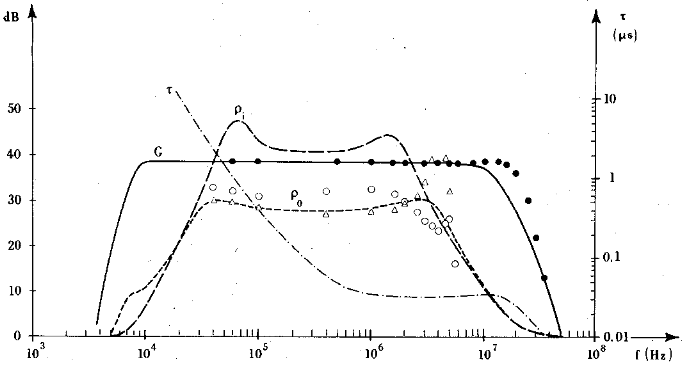


Fig. 12. Computed voltage insertion gain G and reflection coefficients at input and output ports of amplifier in Fig. 10. Experiment points—●: G; ○: ρ_i; △: ρ_o.

An application of analysis programs as routines of larger programs is supplied with reference to the circuit in Fig. 13, which represents a microstrip negative-resistor transistor amplifier tunable in the band 2.05–2.35 GHz, the envelope of its computed tuned voltage insertion gain being shown in Fig. 14. For this amplifier the transistor parameter tolerances and the production yield of a large number of circuits had to be determined given the assigned tolerances of permittivity ϵ and thickness δ of the ceramic substrate and the specified circuit performance [75]. These are expressed by the following relations: a) tuned insertion gain $G = 12$–16 dB in the RF band $B = 2.05$–2.35 GHz; b) 1-dB bandwidth $B_w \geq 50$ MHz at any tuning frequency; c) tunability in the whole RF band by a trimmer capacitor $C_{\text{tun}} = 0.2$–2 pF.

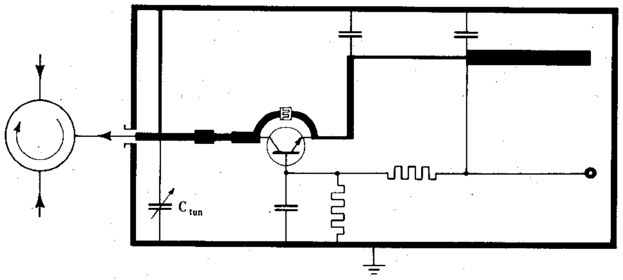


Fig. 13. Thin-film strip-line negative-resistance transistor amplifier (courtesy of Telettra Laboratory).

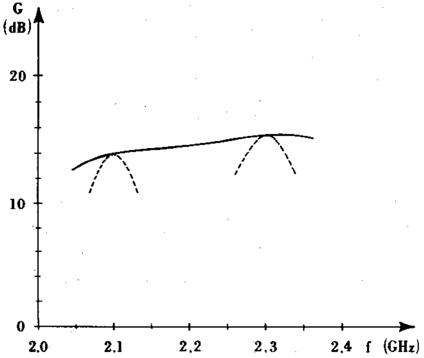


Fig. 14. Envelope of computed tuned voltage insertion gain of amplifier in Fig. 12.

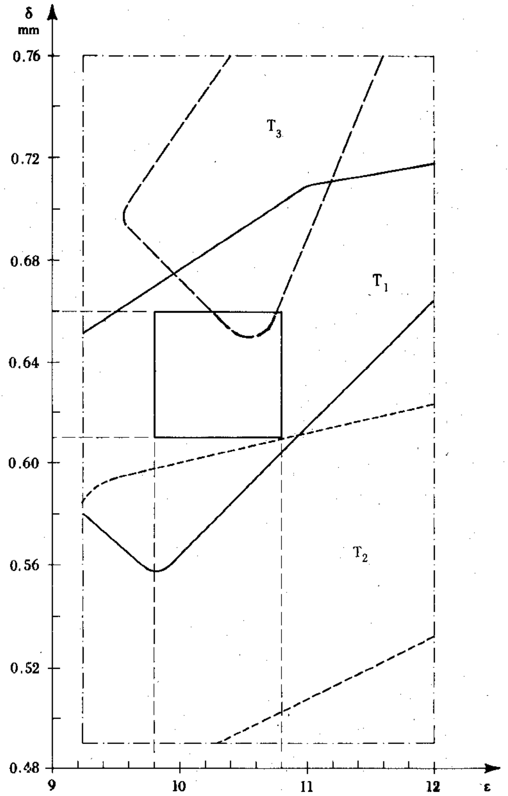


Fig. 15. Contours of acceptable regions in (ϵ,δ) plane for circuit in Fig. 12, with three different transistors.

To this end, the acceptable regions have been determined in the plane (ϵ,δ) and for three transistors T_1, T_2, and T_3 with a program that, by means of repeated analyses (effected with BMT) searches for the contours of these regions. These contours are shown in Fig. 15 where the region of the possible values of ϵ and δ is shaded. It is seen from the same figure that since the whole area lies within the acceptable region relative to T_1, the production yield is 100 percent, while it is almost completely nonexistent for transistors T_2 and T_3. In this way it has been possible to determine the admissible tolerances for the transistor parameters.

The component tolerance assignment in relation to a given production yield is a problem being studied at present, as can be seen from recently published works [68]–[77]. For this, as in the case of optimization problems, it is very important that the programs for network function computation and the respective sensitivity be particularly fast, as is possible by adopting solution methods based on the sparse-matrix techniques described above.

VIII. Conclusions

The methods most frequently adopted by analysis programs of linear circuits in frequency domain have been described, giving greater attention to those best suited for use in microwave circuits. A comparison of the methods has been made emphasizing execution speed as well as limitations imposed on the component nature and the circuit topology.

Section V has been dedicated to the solution methods based on sparse-matrix techniques.

Determination of sensitivity by the direct and the transpose-matrix methods has been dealt with, and the convenience of using one or the other method in relation to the number of parameters and analysis methods has been discussed.

In Section VII some examples of analyzed circuits have been shown, referring, in particular, to the problem of component tolerance assignment.

Acknowledgment

The authors wish to thank the reviewers and Dr. M. Blasgen of the IBM T. J. Watson Center, Visiting Professor of the University of Bologna, for useful suggestions, and Prof. G. Vannucchi, Director of Telettra Laboratory, for giving permission to publish the analyzed circuits.

References

[1] P. E. Green, M. K. McPhun, M. A. Murray-Lasso, and A. E. Smoll, "Automatic general-purpose microwave circuit analysis programs (*Panel Discussion*)," *IEEE Trans. Microwave Theory Tech.* (*Special Issue on Computer-Oriented Microwave Practices*), vol. MTT-17, pp. 527–533, Aug. 1969.

[2] M. L. Dertouzos, G. P. Jessel, and J. R. Stinger, "CIRCAL-2: General-purpose on-line circuit design," *Proc. IEEE* (*Special Issue on Computers in Design*), vol. 60, pp. 39–48, Jan. 1972.

[3] D. A. Calahan, *Computer Aided Circuit Design.* New York: McGraw-Hill, 1972.

[4] W. Baechtold, W. Kotyczka, and M. J. O. Strutt, "Computerized calculation of small signal and noise properties of microwave transistors," *IEEE Trans. Microwave Theory Tech.* (*Special Issue on Computer-Oriented Microwave Practices*), vol. MTT-17, pp. 614–619, Aug. 1969.

[5] G. J. Herskovitz and R. B. Shilling, Ed., *Semiconductor Device Modeling for Computer-Aided Design.* New York: McGraw-Hill, 1972.

[6] R. I. Ollins and S. J. Ratner, "Computer-aided design and optimization of a broad-band high-frequency monolithic amplifier," *IEEE J. Solid-State Circuits* (*Special Issue on Analog Integrated Circuits*), vol. SC-7, pp. 487–492, Dec. 1972.

[7] *Proc. Symp. Essential Characteristics of Semiconductor Devices and Methods of Measurements* (Palermo, Italy, Dec. 1972) (in Italian).

[8] Hewlett-Packard, Application Note 117-1, 1969.

[9] P. Bodharamik, L. Besser, and R. W. Newcomb, "Two scattering matrix programs for active circuit analysis," *IEEE Trans. Circuit Theory* (*Special Issue on Active and Digital Networks*), vol. CT-18, pp. 610–619, Nov. 1971.

[10] F. H. Branin, Jr., "Computer methods of network analysis," *Proc. IEEE* (*Special Issue on Computer-Aided Design*), vol. 55, pp. 1787–1801, Nov. 1967.

[11] G. D. Hatchel, R. K. Brayton, and F. G. Gustavson, "The sparse tableau approach to network analysis and design," *IEEE Trans. Ctrcuit Theory* (*Special Issue on Computer-Aided Circuit Design*), vol. CT-18, pp. 101–113, Jan. 1971.

[12] M. A. Murray-Lasso, "Black-box models for linear integrated circuits," *IEEE Trans. Educ.* (*Special Issue on Educational Aspects of Circuit Design by Computer—I*), vol. E-12, pp. 170–180, Sept. 1969.

[13] D. A. Calahan, "Numerical considerations for implementation of a nonlinear transient circuit analysis program," *IEEE Trans. Circuit Theory* (*Special Issue on Computer-Aided Circuit Design*), vol. CT-18, pp. 66–73, Jan. 1971.

[14] L. Nagel and R. A. Rohrer, "Computer analysis of nonlinear circuits, excluding radiation (CANCER)," *IEEE J. Solid-State Circuits* (*Special Issue on Computer-Aided Circuit Analysis and Device Modeling*), vol. SC-6, pp. 166–182, Aug. 1971.

[15] W. T. Weeks, A. J. Jimenez, G. W. Mahoney, H. Qassemzadah, and T. R. Scott, "Network analysis using a sparse tableau with tree selection to increase sparseness," in *Proc. 1973 IEEE Int. Symp. Circuit Theory* (Toronto, Canada, 1973), pp. 165–168.

[16] W. T. Weeks *et al.*, "Algorithms for ASTAP—A network-analysis program," *IEEE Trans. Circuit Theory* (*Special Issue on Computer-Aided Design*) (Corresp.), vol. CT-20, pp. 628–634, Nov. 1973.

[17] F. H. Branin, Jr., G. R. Hogsett, R. L. Lunde, and L. E. Kugel, "ECAP II—A new electronic circuit analysis program," *IEEE J. Solid-State Circuits* (*Special Issue on Computer-Aided Circuit Analysis and Device Modeling*), vol. SC-4, pp. 146–166, Aug. 1971.

[18] E. DeCastro, *Applied Electronics* (in Italian), preliminary ed. Torino, Italy: UTET, 1974.

[19] V. A. Monaco and P. Tiberio, "A method for automatic scattering matrix computation of a lumped, linear network," *Alta Freq.*, vol. 38, pp. 906–912, Nov. 1969.

[20] ——, "On the transformation of a lumped element linear network into a circuit composed of multiports," *Alta Freq.*, vol. 38, pp. 1013–1014, Nov. 1970.

[21] P. Penfield, Jr., "Description of electrical networks using wiring operators," *Proc. IEEE* (*Special Issue on Computers in Design*), vol. 60, pp. 49–53, Jan. 1972.

[22] G. C. Corazza. "Scattering matrix," (in Italian), *Alta Freq.*, vol. 32, pp. 393–414, June 1963.

[23] V. A. Monaco and P. Tiberio, "Automatic scattering matrix computation of microwave circuits," *Alta Freq.*, vol. 39, pp. 59–64, Feb. 1970.

[24] ——, "A computer program for circuit analysis from dc to microwave using scattering parameters," in *Dig. Tech Papers IEEE Symp. Circuit Theory* (Dec. 1970), pp. 119–120.

[25] F. Bonfatti, V. A. Monaco, and P. Tiberio, "Microwave circuit analysis by sparse matrix techniques," presented at the IEEE G-MTT Int. Microwave Symp., Boulder, Colo., June 1973; also this issue, pp. 264–269.

[26] H. J. Carlin, "The scattering matrix in network theory," *IRE Trans. Circuit Theory* (*Scattering Matrix Issue*), vol. CT-3, pp. 88–97, June 1956.

[27] M. A. Murray-Lasso, "Analysis of linear integrated circuits by digital computer using black-box techniques," in *Computer-Aided Integrated-Circuit Design*, G. J. Herskowitz, Ed. New York: McGraw-Hill, 1968, pp. 113–159.

[28] L. P. Huelsman, *Circuits, Matrices, and Linear Vector Spaces.* New York: McGraw-Hill, 1963.

[29] L. Besser, "Combine *S* parameters with time-sharing," *Electron. Des.*, vol. 16, pp. 62–68, Aug. 1968.

[30] W. N. Parker, "DIPNET, a general distributed parameter network analysis program," *IEEE Trans. Microwave Theory Tech.* (*Special Issue on Computer-Oriented Microwave Practices*), vol. MTT-17, pp. 495–505, Aug. 1969.

[31] P. L. Green, "General purpose programs for the frequency domain analysis of microwave circuits," *IEEE Trans. Microwave Theory Tech.* (*Special Issue on Computer-Oriented Microwave Practices*), vol. MTT-17, pp. 506–514, Aug. 1969.

[32] T. N. Trick and J. Vlach, "Computer-aided design of broadband amplifiers with complex loads," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-18, pp. 541–547, Sept. 1970.

[33] M. E. Mokari-Bolhassan and T. N. Trick, "Computer aided design of distributed-lumped-active networks," *IEEE Trans. Circuit Theory* (*Special Issue on Computer-Aided Circuit Design*) (Corresp.), vol. CT-18, pp. 187–190, Jan. 1971.

[34] V. G. Gelnovatch and I. L. Chase, "DEMON: An optimal seeking computer program for the design of microwave circuits," *IEEE J. Solid-State Circuits* (*Special Issue on Microwave Circuits*), vol. SC-5, pp. 303–309, Dec. 1970.

[35] P. Penfield, Jr., *MARTHA Users' Manual.* Cambridge, Mass.: M.I.T. Press, 1971.

[36] *Sparse Matrix Symp. Proc.* Yorktown Heights, N. Y.: IBM, Rep. RA-1, Mar. 1969.

[37] J. K. Reid, Ed., *Large Sparse Sets of Linear Equations.* New York: Academic, 1970.

[38] D. J. Rose and R. A. Willoughby, Ed., *Sparse Matrices and Their Applications.* New York: Plenum, 1972.

[39] I. DeLotto, G. Pierini, and P. Tiberio, "Sparse matrix techniques in computer-aided design," in *Proc. Int. Computing Symp. European Ass. Comput. Mach.* (Davos, Switzerland, Oct. 1973).

[40] F. G. Gustavson, W. Liniger, and R. A. Willoughby, "Symbolic generation of an optimal Crout algorithm for sparse system of linear equations," *J. Ass. Comput. Mach.*, vol. 17, pp. 87–109, Jan. 1970.

[41] W. F. Tinney and J. W. Walker, "Direct solutions of sparse network equations by optimally ordered triangular factorization," *IEEE Proc.* (*Special Issue on Computer-Aided Design*), vol. 55, pp. 1801–1809, Nov. 1967.

[42] A. Chang, "Application of sparse matrix methods in electric power system analysis," in *Sparse Matrix Symp. Proc.* Yorktown Heights, N. Y.: IBM, Rep. RA-1, Mar. 1969.

[43] R. D. Berry, "An optimal ordering of electronic circuit equa-

tions for a sparse matrix solution," *IEEE Trans. Circuit Theory* (*Special Issue on Computer-Aided Circuit Design*), vol. CT-18, pp. 40–50, Jan. 1971.

[44] A. M. Erisman and G. E. Spies, "Exploiting problem characteristics in the sparse matrix approach to frequency domain analysis," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 260–265, May 1972.

[45] S. W. Director and R. A. Rohrer, "Automated network design—The frequency-doman case," *IEEE Trans. Circuit Theory*, vol. CT-16, pp. 330–337, Aug. 1969.

[46] ——, "The generalized adjoint network and network sensitivities," *IEEE Trans. Circuit Theory*, vol. CT-16, pp. 318–323, Aug. 1969.

[47] G. A. Richards, "Second-derivative sensitivity using the concept of the adjoint network," *Electron. Lett.*, vol. 5, pp. 398–399, Aug. 1969.

[48] V. A. Monaco and P. Tiberio, "On linear network scattering matrix sensitivity," *Alta Freq.*, vol. 39, pp. 193–195, Feb. 1970.

[49] G. C. Temes, "Exact computation of group delay and its sensitivities using adjoint-network concepts," *Electron. Lett.*, vol. 6, pp. 483–485, July 1970.

[50] G. C. Temes and R. N. Gadenz, "Simple technique for the prediction of dissipation-induced loss distortion," *Electron. Lett.*, vol. 6, pp. 836–837, Dec. 1970.

[51] J. W. Bandler and R. E. Seviora, "Current trends in network optimization," *IEEE Trans. Microwave Theory Tech.* (*1970 Symposium Issue*), vol. MTT-18, pp. 1159–1170, Dec. 1970.

[52] G. Iuculano, V. A. Monaco, and P. Tiberio, "Network sensitivities in terms of scattering parameters," *Electron. Lett.*, vol. 7, pp. 53–55, Jan. 1971.

[53] J. W. Bandler and R. E. Seviora, "Computation of sensitivities for noncommensurate networks," *IEEE Trans. Circuit Theory* (*Special Issue on Computer-Aided Circuit Design*) (Corresp.), vol. CT-18, pp. 174–178, Jan. 1971.

[54] R. N. Gadenz and G. C. Temes, "Computation of dissipation induced loss distortion in lumped/distributed networks," *Electron. Lett.*, vol. 7, pp. 258–260, May 1971.

[55] R. A. Rohrer, L. Nagel, R. Meyer, and L. Weber, "Computationally efficient electronic-circuit noise calculations," *IEEE J. Solid-State Circuits* (*Special Issue on Computer-Aided Circuit Analysis and Device Modeling*), vol. SC-6, pp. 204–213, Aug. 1971.

[56] G. Iuculano, V. A. Monaco, and P. Tiberio, "Automatic computation of microwave circuit sensitivity coefficients," in *Proc. 1971 European Microwave Conf.* (Stockholm, Sweden, Aug. 1971).

[57] ——, "A computer program for sensitivity and group-delay evaluation of linear networks," *Alta Freq.*, vol. 40, pp. 873–880, Nov. 1971.

[58] J. W. Bandler and R. E. Seviora, "Wave sensitivities of networks," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-20, pp. 138–147, Feb. 1972.

[59] R. N. Gadenz and G. C. Temes, "Efficient hybrid and state space analysis of the adjoint network," in *Proc. 1972 IEEE Int. Symp. Circuit Theory* (Los Angeles, Calif., Apr. 1972), pp. 184–188.

[60] V. A. Monaco and P. Tiberio, "Two properties for circuit sensitivity in terms of scattering parameters," *Electron. Lett.*, vol. 8, pp. 382–383, Aug. 1972.

[61] G. C. Temes, R. M. Ebers, and R. N. Gadenz, "Some applications of the adjoint network concept in frequency domain analysis and optimization," *Comput. Aided Des.*, pp. 129–134, Apr. 1972.

[62] B. D. H. Tellegen, "A general network theorem, with applications," *Philips Res. Rep.*, vol. 7, pp. 259–269, Aug. 1952.

[63] P. Penfield, Jr., R. Spence, and S. Duinker, "A generalized form of Tellegen's theorem," *IEEE Trans. Circuit Theory*, vol. CT-17, pp. 302–305, Aug. 1970.

[64] S. W. Director, "*LU* factorization in network sensitivity computations," *IEEE Trans. Circuit Theory* (*Special Issue on Computer-Aided Circuit Design*) (Corresp.), vol. CT-18, pp. 184–185, Jan. 1971.

[65] F. H. Branin, Jr., "Network sensitivity and noise analysis simplified," *IEEE Trans. Circuit Theory* (*Special Issue on Large-Scale Networks*), vol. CT-20, pp. 285–288, May 1973.

[66] C. Rauscher, "A fast evaluation of *S*-parameter sensitivities," private communication.

[67] V. Rizzoli, "The calculation of scattering parameters for coupled microstrip arrays of any cross section," *Alta Freq.*, vol. 42, pp. 191–199, Apr. 1973.

[68] E. M. Butler, "Realistic design using large-change sensitivities and performance contours," *IEEE Trans. Circuit Theory* (*Special Issue on Computer-Aided Circuit Design*), vol. CT-18, pp. 58–66, Jan. 1971.

[69] E. DeCastro, G. Iuculano, and V. A. Monaco, "Component value spread and network function tolerances: An optimal design procedure," *Alta Freq.*, vol. 40, pp. 867–872, Nov. 1971.

[70] J. W. Bandler, "Optimization of design tolerances using nonlinear programming," in *Proc. 6th Annu. Princeton Conf. Information Sciences and Systems*, 1972, pp. 655–659; also, to be published in *J. Optimiz. Theory Appl.*

[71] J. F. Pinel and K. A. Roberts, "Tolerance assignment in linear networks using nonlinear programming," in *Proc. 1972 Int. Symp. Circuit Theory* (Los Angeles, Calif., Apr. 1972), pp. 129–183.

[72] B. J. Karafin, "The optimum assignment of component tolerances for electrical networks," *Bell Syst. Tech. J.*, vol. 50, pp. 1225–1242, Apr. 1972.

[73] J. W. Bandler and P. C. Liu, "Automated network design with optimal tolerances," *IEEE Trans. Circuits Syst.*, vol. CAS-21. pp. 219–222, Mar. 1974.

[74] A. R. Thorbjornsen and S. W. Director, "Computer-aided tolerance assignment for linear circuits with correlated element," in *Proc. 1973 Int. Symp. Circuit Theory* (Toronto, Canada, Feb. 1973).

[75] E. Marazzi, V. A. Monaco, A. M. Brini, and V. Solaro, "Computer-aided realistic design of a thin-film microwave reflection-amplifier and production yield optimization," in *Proc. 1973 European Microwave Conf.* (Brussels, Belgium, Sept. 1973).

[76] J. W. Bandler, "The tolerance problem in optimal design," in *Proc. 1973 European Microwave Conf.* (Brussels, Belgium, Sept. 1973).

[77] R. N. Gadenz, G. Rezai-Fakhr, and G. C. Temes, "A method for the computation of large tolerance effects," *IEEE Trans. Circuit Theory* (*Special Issue on Computer-Aided Design*), vol. CT-20, pp. 704–708, Nov. 1973.

[78] F. Bonfatti, V. A. Monaco, and P. Tiberio, "Fast computation of microwave circuit response functions and sensitivities," in *Proc. 5th Colloq. Microwave Communication* (Budapest, Hungary), June 1974.