Microwave Circuit Analysis by Sparse-Matrix Techniques

FLAVIO BONFATTI, STUDENT MEMBER, IEEE, VITO A. MONACO, MEMBER, IEEE, AND PAOLO TIBERIO, MEMBER, IEEE

Abstract—**Linear microwave circuit analysis by sparse-matrix techniques is discussed. Optimum equation ordering and pivoting are proposed to reduce execution time, storage requirement, and to improve accuracy. A direct first-order sensitivity method using sparse-matrix techniques is proposed and compared with the adjoint network method. Details on the implemented program and a numerical example are given.**

Introduction

THE ADVANTAGES which derive from the use of scattering matrices in microwave circuit analysis and design have been clearly recognized for some time [1]–[6]. They arise from the greater convenience of describing circuit performance in terms of normalized waves, and from the possibility offered by scattering parameters of being easily measured on wide frequency bands by automatic measure systems [7], and this allows the characterization of those components which, especially at very high frequencies, cannot be schematized by lumped element circuits. These interesting features gain great weight when the design concerns circuits to be made by modern techniques (such as thin-film, thick-film, or integrated circuits) since experimental iteration techniques cannot be adopted and the circuit must be defined with all details before its realization.

In this paper, a suitable mathematical formulation is proposed, both for normalized waves and for first-order sensitivity. A matrix W is defined which completely describes the circuit by topological information and component S parameters. Pivoting and optimal equation ordering are discussed to ensure good precision and to reduce the computation effort. Finally, a program which determines the requested network functions on the basis of a complete circuit description is presented and a numerical example is given to prove its efficiency.

Problem Formulation

For every microwave circuit component with n_k ports, a system of n_k equations can be written

$$\boldsymbol{b}_k = \boldsymbol{S}_k \boldsymbol{a}_k \tag{1}$$

$\boldsymbol{S}_k$ being its scattering matrix and $\boldsymbol{a}_k$ and $\boldsymbol{b}_k$ the vectors of incident and reflected waves at its n_k ports. A generator, instead, is described by the relation

$$b_g = S_g a_g + c_g \tag{2}$$

to take the impressed wave c_g into account, too.

Collecting together the equations relative to all the m components and generators, a system describing the circuit with all the elements uncoupled is obtained:

$$\boldsymbol{b} = \boldsymbol{S}\boldsymbol{a} + \boldsymbol{c} \tag{3}$$

where

$$\boldsymbol{a} = \begin{vmatrix} a_1 \\ \vdots \\ a_i \\ \vdots \\ a_m \end{vmatrix}, \quad \boldsymbol{b} = \begin{vmatrix} b_1 \\ \vdots \\ b_i \\ \vdots \\ b_m \end{vmatrix}, \quad \boldsymbol{c} = \begin{vmatrix} c_1 \\ \vdots \\ c_i \\ \vdots \\ c_m \end{vmatrix},$$

$$\boldsymbol{S} = \begin{vmatrix} S_1 & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & S_i & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & S_m \end{vmatrix} \tag{4}$$

$\boldsymbol{a}_i$, $\boldsymbol{b}_i$, and $\boldsymbol{c}_i$ being incident, reflected, and impressed wave vectors relative to the ith component and $\boldsymbol{S}_i$ its scattering matrix.

The connections between various components imposed by circuit topology introduce constraints on incident and reflected waves at adjacent ports which may be put in the form

$$\boldsymbol{b} = \boldsymbol{\Gamma}\boldsymbol{a} \tag{5}$$

$\boldsymbol{\Gamma}$ being the connection matrix. Its elements are all zeros except those in the entries corresponding to adjacent ports, which are 1's if normalization numbers are the same [3], [4], [6].

From (3) and (5), by setting

$$\boldsymbol{W} = \boldsymbol{\Gamma} - \boldsymbol{S} \tag{6}$$

one obtains

$$\boldsymbol{a} = \boldsymbol{W}^{-1}\boldsymbol{c} \qquad \boldsymbol{b} = \boldsymbol{\Gamma}\boldsymbol{W}^{-1}\boldsymbol{c} \tag{7}$$

which completely describe the circuit behavior and allow determination of the waves $\boldsymbol{a}$ and $\boldsymbol{b}$ at all the component ports when impressed waves $\boldsymbol{c}$ are given.

$\boldsymbol{W}$ is hereafter referred to as the *connection scattering* matrix; its order

$$n = \sum_{i=1}^{m} n_i$$

equals the total number of component ports.

Manuscript received June 25, 1973; revised September 19, 1973. This work was supported by the Italian Research Council (CNR).

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

Sparse-Matrix Solution and Equation Ordering

A discussion on the properties of matrix $\boldsymbol{W}$ is given in this section in order to find the most convenient technique to solve system (7).

The computational effort can be greatly reduced by taking into account the following.

1) The only nonzero elements of matrix $\boldsymbol{W}$ are: the diagonal ones, which are the reflection coefficients at component ports; those in the entries corresponding to ports belonging to the same component, which are the transmission coefficients between the two ports; and those corresponding to ports connected together, which take the constant value 1.

2) The sparseness structure of $\boldsymbol{W}$ is fixed and does not depend on the frequency.

3) Numerical values of nonzero elements may change with the frequency except the 1's indicating connections.

4) Vector $\boldsymbol{c}$ has nonzero terms only in positions relative to generators.

The fixed sparseness structure of $\boldsymbol{W}$ and $\boldsymbol{c}$ makes the Crout method very convenient for solving system (7), especially when it has to be solved many times with changed coefficient values. The method consists of factoring $\boldsymbol{W}$ into two matrices:

$$\boldsymbol{W} = \boldsymbol{L}\boldsymbol{U} \tag{8}$$

$\boldsymbol{L}$ being lower triangular and $\boldsymbol{U}$ upper triangular with 1's on the diagonal. Then, by the forward and back substitutions,

$$\boldsymbol{L}\boldsymbol{y} = \boldsymbol{c} \qquad \boldsymbol{U}\boldsymbol{a} = \boldsymbol{y} \tag{9}$$

derived from (7), all the wave variables may be obtained.

The elements of matrices $\boldsymbol{L}$ and $\boldsymbol{U}$ and vectors $\boldsymbol{a}$ and $\boldsymbol{y}$ are determined by recurrent formulas [9].

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

$$\boldsymbol{T} = \boldsymbol{L} + \boldsymbol{U} - \boldsymbol{E}$$

$\boldsymbol{E}$ being the unity matrix. It is simple to see that any t_{jk} of $\boldsymbol{T}$ is zero if both w_{jk} of $\boldsymbol{W}$ and all the products $t_{ji}t_{ik}$, with $1 \leq i \leq \min\{j-1, k-1\}$, are zero. The nonzeros generated by LU factorization are hereafter called *fills* in accordance with current usage. The number of fills depends on the $\boldsymbol{W}$ sparseness structure.

A great reduction in execution time can be obtained with the reduced Crout method [10] according to which only the nonzero operands are considered in LU factorization and in forward–back (FB) substitutions. The operations involved depend only on the sparseness structure of $\boldsymbol{W}$ and are therefore always the same for a given circuit topology.

Since a great number of analyses have usually to be performed for the same circuit, it is very convenient to implement the reduced Crout method by generating a code containing only those statements which are strictly necessary to execute the LU factorization and the FB substitutions relative to that circuit.

Code length depends mainly on the structure of $\boldsymbol{W}$ and, to a lesser degree, on $\boldsymbol{c}$ sparsity. It can be strongly reduced by an appropriate ordering of the rows and columns of $\boldsymbol{W}$ before generating the code. Particular attention must, however, be paid because precision depends on the values of the pivots l_{ii} of the reordered matrix.

Predetermined pivoting might cause, for some frequency points, a loss of accuracy due to roundoff errors, since the values of component parameters change with the frequency. However, every row of $\boldsymbol{W}$ contains the constant 1 deriving from $\boldsymbol{\Gamma}$ which could be an ideal pivot because it allows great precision [9] independent of frequency and, at the same time, divisions are avoided. Indeed, about a half of the 1's are modified in the course of the factorization process but rarely may the value of any drop to zero and then only in anomalous cases.

In order to find the best ordering strategy for matrix $\boldsymbol{W}$, a computer program has been implemented which, by simulating the LU factorization, determines the number of fills occurring and the operations involved. With this program many ordering algorithms have been tested on several microwave circuits and the average ratio between the nonzeros in $\boldsymbol{T}$ and in $\boldsymbol{W}$ has been assumed as the index of ordering efficiency. Some of the tested strategies have been expressly set up for matrix $\boldsymbol{W}$ and others derived from those proposed by different authors with reference to the nodal admittance matrix [11], [17]. These have been modified by ordering columns so as to place 1's on the main diagonal, since the pivoting question is not generally taken into account in sparse solution of the nodal admittance matrix.

A comparative examination based on the efficiency index, speed, and simplicity of implementation, indicated that the most convenient algorithm for the microwave circuits described by system (7) is the following [8].

1) The couple of rows relative to adjacent ports are considered together and ordered so that each couple has a number of nonzeros not greater than that of the successive one; in every couple the row with fewer nonzeros precedes the other.

2) The columns are then ordered so as to place all the 1's of $\boldsymbol{\Gamma}$ on the main diagonal.

First-Order Sensitivity

Sensitivities of incident and reflected wave vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ with respect to any variable x on which the parameters of one or more components depend, are generally very useful to the circuit designer, both in evaluating the dependence of circuit functions on some variables, and in circuit optimization.

Inasmuch as circuit analysis is carried out by sparse-matrix solution with code generation, two methods can find useful application. One, the direct method, is convenient when the sensitivities of normalized waves at many ports with respect to one variable are required.

The other, based on adjoint circuit concepts, is preferable when sensitivities of only one normalized wave with respect to many variables have to be computed [6], [12], [13], [18]. In order to justify this declaration, the two methods are now briefly described.

Partial derivation of (7) with respect to a generic variable x gives

$$\frac{\partial \boldsymbol{a}}{\partial x} = -\boldsymbol{W}^{-1}\frac{\partial \boldsymbol{W}}{\partial x}\boldsymbol{W}^{-1}\boldsymbol{c} = \boldsymbol{W}^{-1}\frac{\partial \boldsymbol{S}}{\partial x}\boldsymbol{a} = \boldsymbol{W}^{-1}\boldsymbol{c}_s \qquad (10)$$

the connection matrix $\boldsymbol{\Gamma}$ being independent of x and

$$\boldsymbol{c}_s = \frac{\partial \boldsymbol{S}}{\partial x}\boldsymbol{a} = \left|\begin{array}{c} \dfrac{\partial \boldsymbol{S}_1}{\partial x}\boldsymbol{a}_1 \\ \vdots \\ \dfrac{\partial \boldsymbol{S}_j}{\partial x}\boldsymbol{a}_j \\ \vdots \\ \dfrac{\partial \boldsymbol{S}_m}{\partial x}\boldsymbol{a}_m \end{array}\right|. \qquad (11)$$

Computation of wave sensitivities according to the direct method consists in the application of (10) which differs from (7) due to substitution of the impressed wave vector $\boldsymbol{c}$ by $\boldsymbol{c}_s$. It can be computed in terms of matrices $\partial \boldsymbol{S}_j/\partial x$ of all the components depending on x and in terms of vectors $\boldsymbol{a}_j$ of the incident waves at their ports. These last are obtained from the circuit analysis described in the previous section.

Sparse-matrix solution applied to (10) requires the LU factorization of $\boldsymbol{W}$ and FB substitution according to vector $\boldsymbol{c}_s$. The former having been executed for analysis does not require any computation while the latter consists of a number of arithmetical operations corresponding to the execution of the FB-substitution code already generated for the analysis. Sparsity of $\boldsymbol{c}_s$, however, depends on variable x and is completely different from the sparsity of impressed wave vector $\boldsymbol{c}$. The FB-substitution code generated for the analysis may be utilized for sensitivity only if, taking no account of the $\boldsymbol{c}$ sparseness, a complete code has been generated. Generation of the complete FB-substitution code, however, while it does not produce a substantial code increase, could become useful even when analyses of the same circuit with different excitation conditions[1] have to be effected.

Thus the evaluation of all wave sensitivities with respect to a given variable x implies an extra central processing unit (CPU) time which is very short compared to that required by the analysis alone; for every new variable x, one more FB substitution is necessary.

The adjoint circuit method may be deduced very simply from (10). In fact, if only the sensitivity of wave a_j at port j has to be determined, by letting $\boldsymbol{\gamma}_j^T$ be a row vector with all the elements zero except a 1 in position j

$$\boldsymbol{\gamma}_j^T = |\,0\cdots1\cdots0\,|, \qquad 1 \leq j \leq n \qquad (12)$$

one obtains from (10)

$$\frac{\partial a_j}{\partial x} = \boldsymbol{\gamma}_j^T\frac{\partial \boldsymbol{a}}{\partial x} = \boldsymbol{\gamma}_j^T\boldsymbol{W}^{-1}\frac{\partial \boldsymbol{S}}{\partial x}\boldsymbol{a} = \boldsymbol{\alpha}^T\frac{\partial \boldsymbol{S}}{\partial x}\boldsymbol{a} \qquad (13)$$

where vector

$$\boldsymbol{\alpha} = (\boldsymbol{\gamma}_j^T\boldsymbol{W}^{-1})^T = (\boldsymbol{W}^T)^{-1}\boldsymbol{\gamma}_j = \boldsymbol{\mathcal{W}}^{-1}\boldsymbol{\gamma}_j \qquad (14)$$

can be interpreted as the incident wave vector of a circuit described by a matrix $\boldsymbol{\mathcal{W}} = \boldsymbol{W}^T = (\boldsymbol{\Gamma} - \boldsymbol{S}^T)$ and excited according to $\boldsymbol{\gamma}_j$. Such a circuit has the same connection matrix $\boldsymbol{\Gamma}$ as the original one but its components have scattering matrices $\boldsymbol{S}_1^T,\boldsymbol{S}_2^T\cdots\boldsymbol{S}_n^T$ equal to the transpose of the corresponding ones in the original circuit.

The effort in computing sensitivity by (13) is, at first glance, greater than that due to (10), because determination of $\boldsymbol{\alpha}$ by (14) implies the transposition of component matrices and the execution of the LU-factorization code in addition to the execution of the FB-substitution code.[2]

When vector $\boldsymbol{\alpha}$ of incident waves at the adjoint circuit ports has, however, been obtained, the sensitivity of a_j with respect to any variable can be computed only by applying (13) for different variables $x_1,x_2,\cdots,x_m$.

In conclusion, the adjoint network method might be preferred when sensitivities with respect to many variables have to be computed. However, if sparse-matrix solution with code generation is adopted, the extra time required by the direct method could not be so long as to make the adjoint one preferable to it, even taking into account that the direct method furnishes the sensitivities of all waves $\boldsymbol{a}$ at the same time.

Program Description

A program (BMT) has been implemented which supplies all the requested network functions starting from the complete description of the circuit to be analyzed. It has been divided into three phases which can each be executed independently in order to obtain good flexibility of utilization and, on the other hand, considerable reduction of central memory requirement.

Phase 1—Data Input and Interpretation

The input data are organized in three groups as follows.

1) Command instructions to choose the output options (wave vectors, circuit response functions, S parameters, sensitivities, group delay, etc.).

[1] This is the case when the circuit scattering matrix with reference to all external ports has to be computed.

[2] Execution of the LU-factorization code might be avoided by taking into account that from (8) we have $\boldsymbol{\mathcal{W}} = \boldsymbol{W}^T = \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{U}} = (\boldsymbol{L}\boldsymbol{U})^T = \boldsymbol{U}^T\boldsymbol{L}^T$ and vector $\boldsymbol{\alpha}$ might be obtained by a forward substitution applied to the lower triangular matrix and a backward one applied to the upper triangular matrix $\boldsymbol{\mathcal{U}} = \boldsymbol{L}^T$ [14]. These substitutions, however, could not be done by the FB-substitution code generated for solving (9) because the sparseness structure of $\boldsymbol{L}$ and $\boldsymbol{U}$ is generally different from that of $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{U}}$, respectively, and a new code would have to be generated. Such a new FB-substitution code generation would, however, be too cumbersome compared to the execution of the LU-factorization code applied to $\boldsymbol{\mathcal{W}}$. The same code generated for factorizing $\boldsymbol{W}$ can, in fact, be usefully employed also for $\boldsymbol{\mathcal{W}}$, due to the symmetrical structure of this matrix.

2) Data describing circuit topology (number of components, port numbering for every component, connections between ports).

3) Data relative to the component parameters (directly measured S parameters, other data in terms of which the S parameters can be calculated by library routines or by routines directly provided by the user).

The command instructions are first analyzed and coded into orders for the program, providing diagnostic messages when they are not correct. Subsequently, the topological data are assembled in tables to be used in phase 2 and, on the basis of the command instructions, component S parameters and their derivatives are computed for every frequency point using group 3) data. These parameters, together with those read as measured data, are organized in files corresponding to the different frequency points and stored in an auxiliary memory to be utilized in phase 3.

Phase 2—Ordering and Code Generation

This phase consists in the execution of a two-pass compiler-like program. It translates the circuit topological description into a Fortran code containing all the operations whose execution gives the requested unknown vectors in terms of the $\boldsymbol{W}$ elements and excitation vectors.

On the basis of the circuit topological description, the first pass determines the ordering of equations according to the adopted algorithm and produces an intermediate file containing a description of the nonzero positions in the matrix obtained by reordering $\boldsymbol{W}$. Moreover, it combines every nonzero with the address of the location, in the file arranged during phase 1, which contains the value of the corresponding S parameter. To avoid memory waste, all these data are stored in tables pseudodynamically allocated by a group of routines set up for that purpose [15]. Their use allows the request of tables whose size is determined during program execution, and the release of memory space reserved for tables no longer required.

The second pass generates a Fortran code formed by the factorization and substitution routines and their main program. The generated code is stored in an auxiliary memory to reduce central memory occupation even further. The decision to generate the code in Fortran language derives from its machine independence even if the use of a machine language would have allowed faster execution.

When the generated code is complete, it is brought back into the central memory and compiled. The object code which is obtained is automatically stored in a temporary file by the computer operating system; on user request, it is possible to store it in a private permanent file.

Phase 3—Loading and Execution

The object code is loaded into the central memory together with the precompiled library routines which have been written to manage the procedures chosen by the command instructions. The most general execution procedure is that relative to the computation of sensitivity; in accordance with the direct method adopted it involves, for every frequency point, the following operations.

1) Execution of the LU-factorization code utilizing the S parameters arranged in phase 1.

2) Execution of the FB-substitution code utilizing the vector of impressed waves $\boldsymbol{c}$. So, the normalized waves $\boldsymbol{a}$ and $\boldsymbol{b}$ at every component port are obtained.

3) Computation of any response functions requested.

4) Evaluation of the vector $\boldsymbol{c}_s$, defined in (11), for each variable x.

5) Execution of the FB-substitution code for every $\boldsymbol{c}_s$ vector, each being utilized as an excitation vector. Every execution gives the sensitivities of incident waves at all the component ports relative to a variable x.

If sensitivities have not to be computed, the procedure terminates after the third point.

Phase 3 is not necessarily preceded by both the others. In fact, on user request, it is possible to store the object code in a permanent file and utilize it subsequently to perform new analyses of circuits with the same topology but with different parameter values. Therefore, the extra CPU time necessary to generate and compile the code is spent only once, even if many analyses are executed and also, at different times. The whole procedure is shown in the flow diagram of Fig. 1.

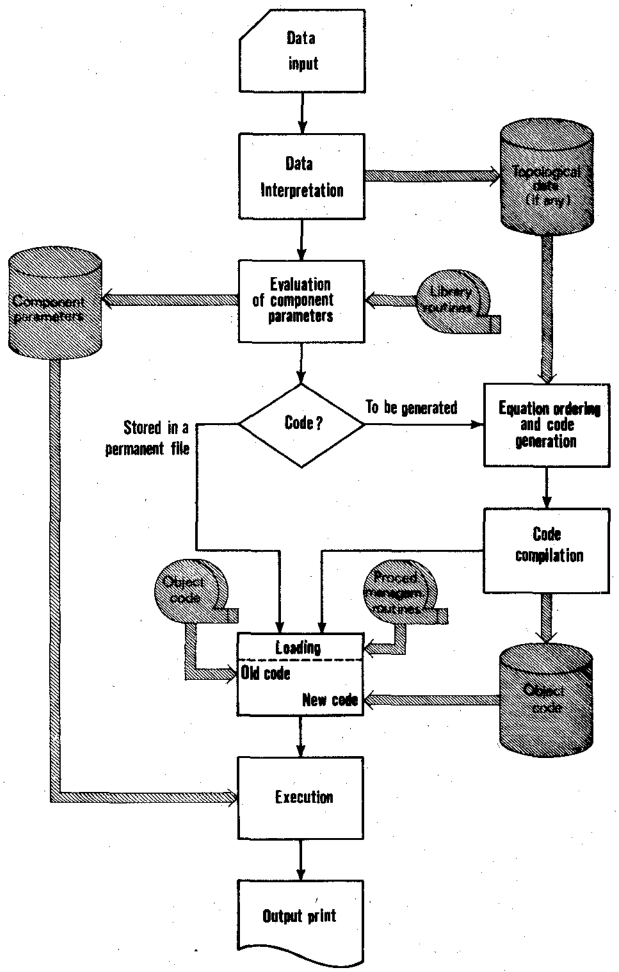

Fig. 1. BMT program organization.

A Numerical Example

In order to give quantitative information on the program, the thin-film strip-line branching filter in Fig. 2(a) has been analyzed. It has been described for the program as shown in Fig. 2(b) with port 1 connected to a generator, ports 2 and 3 to loads, and all the others to open-circuit terminations.

The transmission coefficients

$$S_{21} = |S_{21}| \exp(j\beta_{21})$$

$$S_{31} = |S_{31}| \exp(j\beta_{31})$$

have been computed and their moduli plotted versus frequency in Fig. 3. Moreover, the group delay of S_{31}

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

and the sensitivities of $|S_{31}|$

$$M_{\epsilon} = \frac{\partial \ln |S_{31}|}{\partial \ln \epsilon} = \epsilon \,\mathrm{Re}\left[\frac{1}{S_{31}} \frac{\partial S_{31}}{\partial \epsilon}\right]$$

$$M_{\delta} = \frac{\partial \ln |S_{31}|}{\partial \ln \delta} = \delta \,\mathrm{Re}\left[\frac{1}{S_{31}} \frac{\partial S_{31}}{\partial \delta}\right]$$

relative to permittivity ϵ and thickness δ of the microstrip ceramic substrate, have been determined and plotted versus frequency in Figs. 4, 5 and 6, respectively. The thin-film coupled-line component S parameters have been determined from geometrical description by routines [16] associated to the program.

In the example given, $\boldsymbol{W}$ is a 96 × 96 matrix with 384 nonzeros. The CDC6600 CPU time for ordering $\boldsymbol{W}$ and

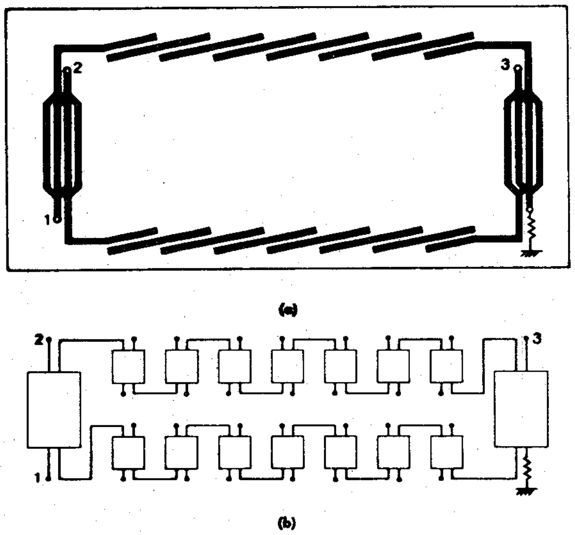

Fig. 2. (a) Thin-film strip-line branching-filter. (b) Its description for program BMT.

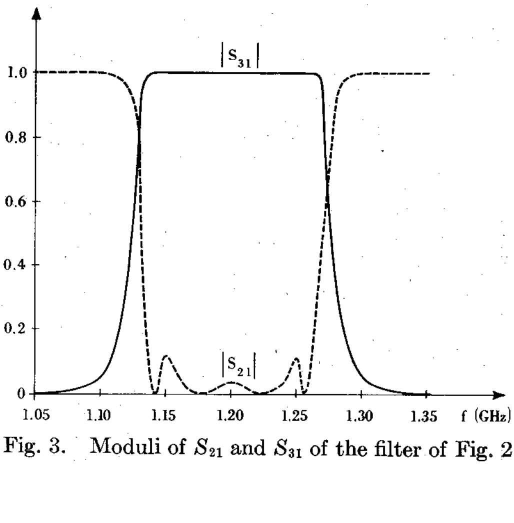

Fig. 3. Moduli of S_{21} and S_{31} of the filter of Fig. 2.

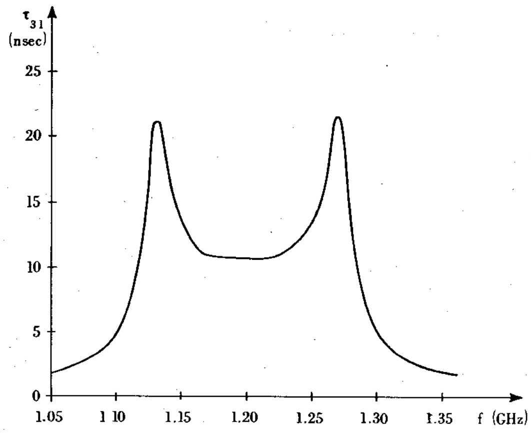

Fig. 4. Group delay of S_{31}.

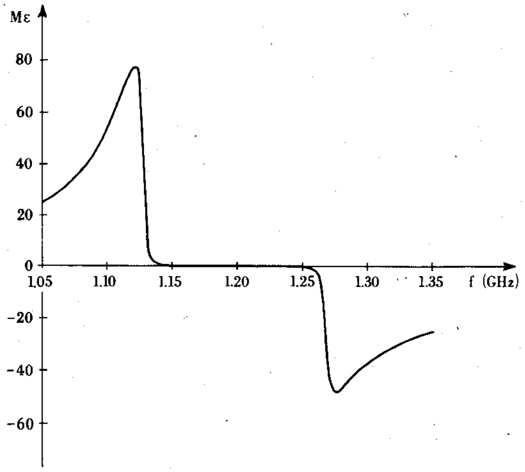

Fig. 5. Sensitivity of $|S_{31}|$ relative to permittivity ϵ of the microstrip ceramic substrate.

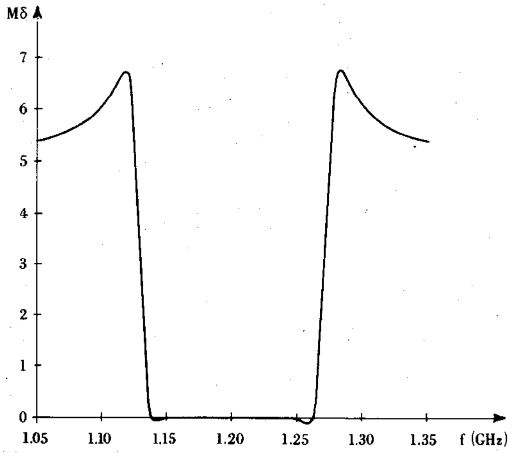

Fig. 6. Sensitivity of $|S_{31}|$ relative to thickness δ of the microstrip ceramic substrate.

for generating and compiling the factorization and substitution code is about 10 s. The generated code is composed of 1410 statements and the fills number 139. The time spent in executing the *LU*-factorization code once is about 8 ms, while that for the FB-substitution code is about 5 ms. Therefore, the time required to evaluate the normalized waves at all the ports is about 13 ms for every frequency point. Maximum memory, about 30k words, was required during generated code compilation, including the 16k words of the compiler.

Conclusions

The connection scattering matrix $\boldsymbol{W}$, which completely describes a microwave circuit by topological information and the component S parameters, has been defined.

A microwave circuit analysis technique based on the reduced Crout method applied to $\boldsymbol{W}$, has been discussed. It consists in generating a Fortran code composed of the statements which are strictly necessary to execute the *LU* factorization and FB-substitution procedures. The advantage in employing the generated code itself for computing group delays and sensitivities has been shown and the relative procedures examined.

The organization of program BMT, which supplies various network functions (wave vectors, circuit response functions, sensitivities, group delay, etc.), starting from the complete description of the circuit to be analyzed, has been presented. The efficiency of the program has been proved with a numerical example.

Acknowledgment

The authors wish to thank Dr. V. Rizzoli for furnishing the application example and for the useful discussions on the computed results.

References

[1] Hewlett-Packard, Application Note 95, Sept. 1968.

[2] 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.

[3] ——, "Automatic scattering matrix computation of microwave circuits," *Alta Freq.*, vol. 39, pp. 59–64, Feb. 1970.

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

[5] 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.

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

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

[8] F. Bonfatti, V. A. Monaco, and P. Tiberio, "Microwave circuit analysis by sparse matrix techniques," in *Proc. 1973 IEEE G-MTT Int. Microwave Symp.* (Boulder, Colo.), June 1973.

[9] B. Noble, *Applied Linear Algebra.* Englewood Cliffs, N. J.: Prentice-Hall, 1969.

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

[11] 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. 260–264, May 1972.

[12] S. W. Director and R. A. Rohrer, "The generalized adjoint network and network sensitivities," *IEEE Trans. Circuit Theory*, vol. CT-16, pp. 318–323, Aug. 1969.

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

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

[15] F. Bonfatti, G. Bucci, and P. Tiberio, "Dynamic allocation of data tables in FORTRAN programming," Univ. Bologna, Bologna, Italy, CNR-CIOC Internal Rep. 16, Apr. 1973.

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

[17] B. Dembart and A. M. Erisman, "Hybrid sparse-matrix methods," *IEEE Trans. Circuit Theory* (*Special Issue on Computer-Aided Design*), vol. CT-20, pp. 641–649, Nov. 1973.

[18] F. H. Branin, Jr., "Network sensitivity and noise analysis simplified," *IEEE Trans. Circuit Theory*, vol. CT-20, pp. 285–288, May 1973.