Efficient and Systematic Solution of Real and Complex Eigenvalue Problems Employing Simplex Chain Vertices Searching Procedure

Jerzy Julian Michalski, *Member, IEEE*, and Piotr Kowalczyk

***Abstract*—This paper presents a novel method that is very efficient in solving multidimensional real and complex eigenvalue problems, commonly employed in electromagnetic analysis, which can be transformed into a nonlinear equation. The concept is realized as root tracing process of a real or complex function of N variables in the constrained space. We assume that the roots of the continuous function of N variables lie on the continuous $(N-1)$-dimensional hyperplane. The method uses regular N and $(N-1)$-Simplexes, at which vertices the considered function changes its sign. Based on $(N-1)$-Simplex, the function is evaluated at two new points that are vertices of new regular N-Simplexes for which $(N-1)$-Simplex is one of its $(N-1)$-faces. The algorithm, with the usage of stack, runs in an iterative mode tracing the roots inside the volume of the considered simplexes. As a result, the algorithm creates a chain of simplexes in the constrained region. The proposed algorithm is optimal in the sense of the number of function evaluations. The numerical results, real and complex dispersion characteristics of chosen microwave guides, have proven the versatility and efficiency of the proposed algorithm.**

***Index Terms*—Algorithms, eigenvalues and eigenfunctions, ferrites, nonlinear equations, optical waveguides.**

I. Introduction

MANY electromagnetic problems are generally described by an intricate eigenvalue problem, and its solution determines a value of an unknown parameter, (e.g., propagation coefficient or resonance frequency). Most frequently the problem can be transformed into a nonlinear equation with respect to this parameter.

All nonlinear equations can be expressed in a form involving a real or complex function of many variables $F(x)=0$, where $x \in R^N$. The problem becomes much more complicated if the number of variables N is increasing. Moreover, in numerous electromagnetic problems, the function is defined as a determinant of a high-order matrix. The evaluation of such functions can be very time consuming. Thus, it is very important to minimize the number of the function evaluations. Moreover, for some structures, such as ferrite lines, periodic structures, nonuniform waveguides, and any open lines, a dispersion equation is defined in complex domain (complex or leaky modes).

Generally, finding the roots of a function is the most common numerical problem. There is a large number of standard numerical procedures that can be successfully applied to such tasks: the Newton iteration method, Muller interpolation method [1], and secant method (Newton's method where derivatives are replaced with finite differences). Despite a large number of known root-finding techniques, new algorithms are still arising. In recent years, a few new methods for finding the roots of real and complex functions were published [2]–[4]. However, most of them are based on iterative methods and concern improving the accuracy of finding a singular root, which are usually inefficient in electromagnetic analysis. For example, in microwave transmission line theory, the dispersion characteristic is defined as a propagation coefficient in a function of frequency $\gamma(f)$. Propagation coefficients with corresponding frequency points are the roots of the dispersion equation $d(f,\gamma)=0$. In order to find the dispersion curve $\gamma(f)$ (in a specified frequency band), the propagation coefficients are usually calculated using the root-finding algorithms for each frequency point. However, this process is very time consuming. Moreover, if two or more dispersion characteristics are lying close to each other, the roots can be improperly classified (to a different characteristic). To avoid these disadvantages, a root-tracing algorithm can be utilized. The procedure starts from the known root and runs iteratively, tracing subspace where a function changes its sign along a certain path, surrounding all roots in a constrained region.

The algorithm of root tracing for a real function of two variables was published years ago [5]. This method, although very practical, is not optimal in the sense of the number of function evaluations. Some of them are "empty shots"—the points calculated in a search space where the considered function does not change the sign comparing to the function sign from last iteration step. Moreover, this method works with a real function of two variables only. In the proposed approach, the algorithm can be applied for both complex and real functions.

The method presented in this paper is free of the above-mentioned disadvantages because it is based on the simplex chain vertices searching (SCVS) procedure. For a function of N variables, each new calculated point in search space generates up to (f_0-1) new regions consisting of roots of the function (the subspaces where considered function changes sign). The value f_0 is defined as the binominal coefficient $C\binom{N+1}{1}$.

Manuscript received December 21, 2010; revised May 16, 2011; accepted May 24, 2011. Date of publication July 25, 2011; date of current version September 14, 2011. This work was supported by the Polish National Ministry for Science and Higher Education (Decision 736/N-COST/2010/0) under the "New Optimization Methods and Their Investigation for the Application to Physical Microwave Devices That Require Tuning" project performed within the COST Action RFCSET IC0803.

J. J. Michalski is with Research and Development, TeleMobile Electronics Ltd., Gdynia 81-451, Poland.

P. Kowalczyk is with Microwave and Antenna Engineering, Technical University, Gdańsk 81-451, Poland.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMTT.2011.2160277

0018-9480/$26.00 © 2011 IEEE

The main goals of this paper are to present a novel root-tracking algorithm and show its efficiency for solving electromagnetic problems. To verify the validity of the proposed method, a few chosen types of microwave waveguide lines were analyzed.

The main concept of the method is illustrated in Section II where the general algorithm for N-variables is presented. More detailed considerations are presented for two and three variables cases. The following section presents the applications of the proposed algorithm—the numerical results of the dispersion equation solution in the real and complex domain, for the case of two and three variables. Comments are presented and some aspects of the algorithm are summarized at the end of this paper.

II. Algorithm

Generally, N-Simplex is an N-dimensional convex hull polytope with $N+1$ vertices [6] considered in $N+1$ dimensional space. For example, 0-Simplex is a point, a 1-Simplex is a line segment, a 2-Simplex is a triangle, a 3-Simplex is a tetrahedron, and a 4-Simplex is a pentachoron. In geometry, simplex is a N-dimensional analog of a triangle.

In this approach, we consider regular $(N-1)$-Simplexes and N-Simplexes in N-dimensional space E. Let us denote N-Simplex by Δ^N and its set of vertices by $V^N = \{(v^1, \cdots, v^{N+1}) \in R^N\}$. Each vertex v^n is defined by N coordinates $v^n(x_1^n, \cdots, x_N^n)$. The upper index n represents the vertex number and the lower one represents the number of the coordinate in the N-dimensional space E. Let us now define (in this space) a continuous function F and the equation $F(x_1, x_2, \ldots, x_N) = 0$. We assume the constrained search space E for each variable $x_{L_i} < x_i < x_{R_i}$, $i = 1, \ldots, N$. We denote, by Δ_0^N, the N-Simplex at whose vertices the considered function obtains all the values of the same sign, and its set of vertices is denoted by V_0^N. The following definition $\Delta_\pm^N$ will be utilized for N-Simplexes at which vertices the function values are of different signs. For this case, the set of vertices will be denoted by $V_\pm^N$.

In the algorithm (see Fig. 1), we start with N initial points $V_\pm^{N-1}$ (which create regular simplex $\Delta_\pm^{N-1}$) for which the considered function obtains the values of different signs. Using the vertices of $\Delta_\pm^{N-1}$, we define two new points $v^{(N+1)(1)}$ and $v^{(N+1)(2)}$, where function should be evaluated. These new points, with the rest of vertices of $\Delta_\pm^{N-1}$, create in space E the two new regular N-Simplexes $\Delta_\pm^N$. This new regular N-Simplex is constructed from a regular $(N-1)$-Simplex by connecting a new vertex to all original vertices by the common edge length. The coordinates $v^{(N+1)(1)}(x_1^{(N+1)(1)}, \cdots, x_N^{(N+1)(1)})$ and $v^{(N+1)(2)}(x_1^{(N+1)(2)}, \cdots, x_N^{(N+1)(2)})$ of new vertices can be found from the following system of N quadratic equations[1]

$$\left\{e^{m(N+1)} = e\right\}, \qquad m = 1, \ldots, N \tag{1}$$

where $e^{m(N+1)} = \left(\sum_{i=1}^N \left(x_i^m - x_i^{N+1}\right)^2\right)^{1/2}$ is an edge length between mth and the searched $(N+1)$th vertex, e is an edge length of the base $(N-1)$-Simplex and consequently of the N-Simplex and x_i^m—i-th coordinate of m-th vertex.

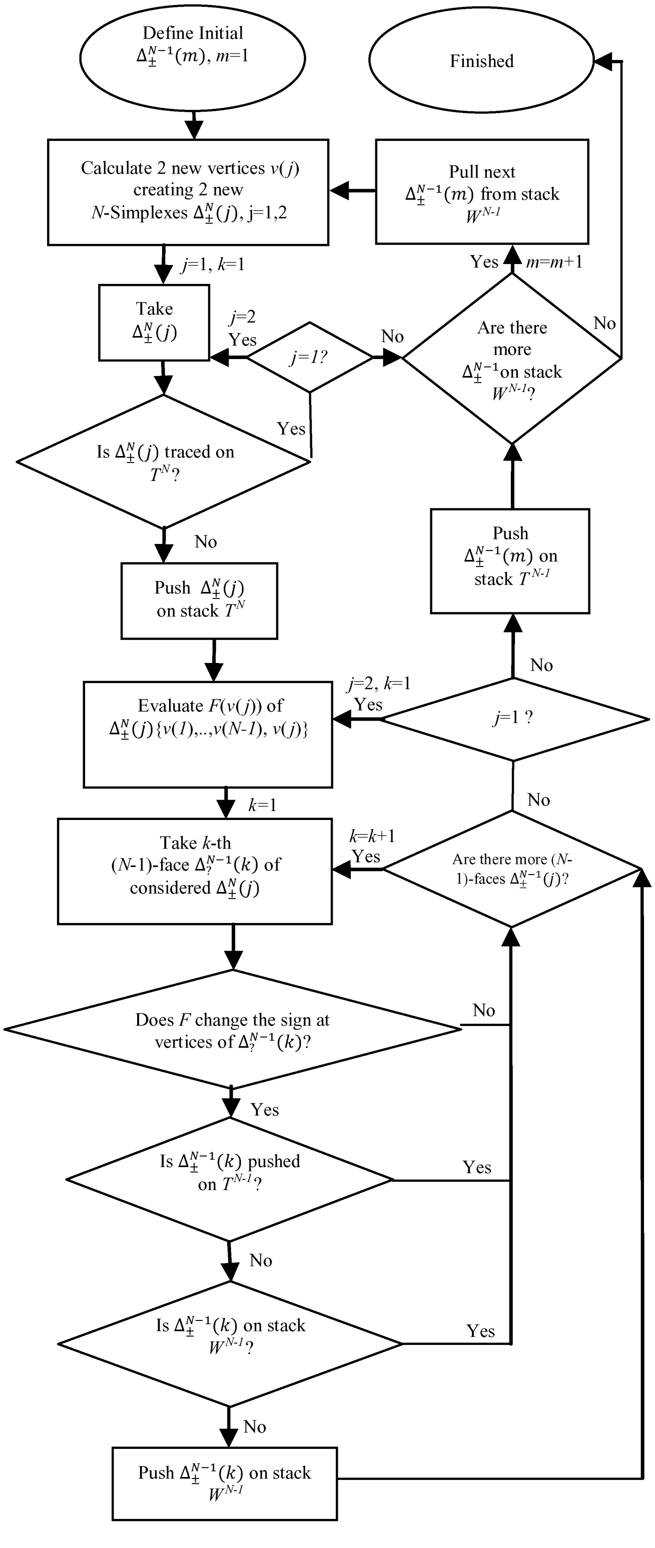
Define Initial $\Delta_\pm^{N-1}(m)$, m=1
Finished
Calculate 2 new vertices $v(j)$ creating 2 new N-Simplexes $\Delta_\pm^N(j)$, j=1,2
Pull next $\Delta_\pm^{N-1}(m)$ from stack W^{N-1}
j=1, k=1
Yes
m=m+1
Take $\Delta_\pm^N(j)$
j=2 Yes
j=1?
No
Are there more $\Delta_\pm^{N-1}$ on stack W^{N-1}?
No
Yes
Is $\Delta_\pm^N(j)$ traced on T^N?
No
Push $\Delta_\pm^N(j)$ on stack T^N
Push $\Delta_\pm^{N-1}(m)$ on stack T^{N-1}
No
j=2, k=1 Yes
Evaluate $F(v(j))$ of $\Delta_\pm^N(j)\{v(1),..,v(N-1), v(j)\}$
j=1 ?
k=1
No
k=k+1 Yes
Take k-th (N-1)-face $\Delta_?^{N-1}(k)$ of considered $\Delta_\pm^N(j)$
Are there more (N-1)-faces $\Delta_\pm^{N-1}(j)$?
Does F change the sign at vertices of $\Delta_?^{N-1}(k)$?
No
Yes
Is $\Delta_\pm^{N-1}(k)$ pushed on T^{N-1}?
Yes
No
Is $\Delta_\pm^{N-1}(k)$ on stack W^{N-1}?
Yes
No
Push $\Delta_\pm^{N-1}(k)$ on stack W^{N-1}

Fig. 1. Block diagram of SCVS algorithm.

Considering all $(N-1)$-faces of new constructed simplexes $\Delta_\pm^N$ that are in fact another $\Delta_\pm^{N-1}$ or Δ_0^{N-1} simplexes, we select such simplexes $\Delta_\pm^{N-1}$ at which vertices the function changes its sign. Number L of the new simplexes Δ^{N-1} depends on the

[1] It can be proven that such system of quadratic equations, if $e^{mn} = a$ for $m, n = 1, 2, \ldots, N+1$; $m \neq n$; $a \in R$, always has two solutions in a real domain.

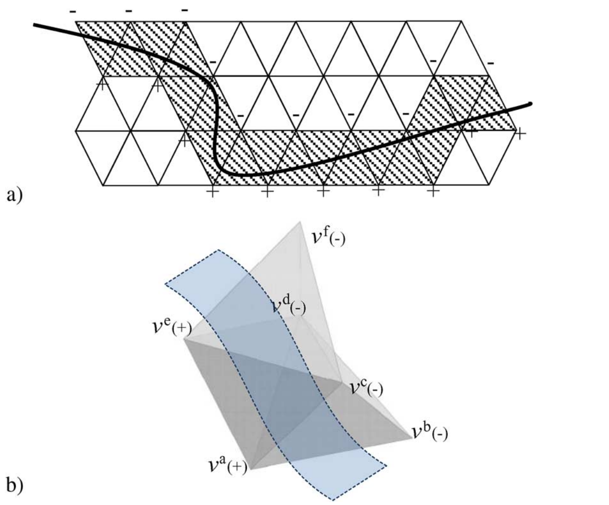

Fig. 2. (a) Run of the algorithm for $N = 2$ variables. All lined triangles define a tracked region (simplex chain) where function changes sign and the continuous line represents exact roots of the function. In this case roots lie in 2-D space on 2-D curve. (b) Run of the algorithm for $N = 3$ variables, where we present the root's plane through three 3-simplexes: $(v^a, v^b, v^c, v^d), (v^a, v^c, v^d, v^e), (v^c, v^d, v^e, v^f)$. Considering the sign of the function at vertices, we see that roots lie in 3-D space on 3-D plane on the following surfaces of triangles (2-simplexes): (v^a, v^b, v^c), (v^a, v^b, v^d), (v^a, v^c, v^d), (v^c, v^d, v^e), (v^d, v^e, v^f), and (v^a, v^c, v^e).

simplexes dimension and can be calculated with the following formula:

$$L = 2(f_o - 1) \tag{2}$$

where f_0 is the number of $(N-1)$-faces of the single N-Simplex. In general, the number of m-faces of the N-Simplex is equal to the binomial coefficient[2] $C\binom{N+1}{m+1}$. The vertices of new created simplexes $\Delta_{\pm}^{N-1}$ are used as the initial points to run the algorithm in an iterative mode. In one iteration step, we consider only one newly created simplex $\Delta_{\pm}^{N-1}$. The rest is push on the stack, waiting for evaluation. This way, we create a simplex chain locking a new sub-region in space E, where considered function changes its sign. Simultaneously, we build a database where we store the information on the already traced simplexes $\Delta_{\pm}^{N-1}$. Before we evaluate the function at a new vertex, we check if the simplex with this vertex was already evaluated.

The exemplary run of the algorithm for $N = 2$ and $N = 3$ is presented in Fig. 2. All lined triangles (2-Simplexes) are the tracked regions at which vertices the function is evaluated. An analogous example [see Fig. 2(a)] is presented in [5]. The main difference between the method presented in [5] and our approach is a definition of grid (in our approach squares are replaced with triangles).

A. Algorithm for Two Variables

We assume that the roots of a real function of two variables lie on the continuous curve C in 2-D real space (Fig. 3). We consider initial 1-Simplex in 2-D space E [line segment between points v^1 and v^2 in Fig. 3(a)]. At vertices $v^1(x_1^1, x_2^1)$ and $v^2(x_1^2, x_2^2)$, the considered function obtains different signs: $\operatorname{sign}(F(v^1)) \neq \operatorname{sign}(F(v^2))$. The dotted curve C represents exact roots of the function. According to the presented above procedure, and based on (1), we get the following system of two quadratic equations:

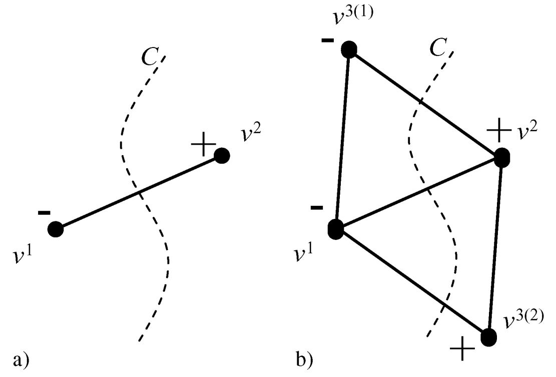

Fig. 3. Procedure for two variables. (a) Initial 1-Simplex $\{v^1, v^2\}$. (b) New evaluation points $v^{3(1)}, v^{3(2)}$ as the vertices of new 2-Simplexes: $\{v^1, v^2, v^{3(1)}\}$ and $\{v^1, v^2, v^{3(2)}\}$.

$$\{e^{m3} = e\}, \qquad m = 1, \ldots, 2 \tag{3}$$

where $e^{m3} = \left((x_1^m - x_1^3)^2 + (x_2^m - x_2^3)^2\right)^{1/2}$. The solution is the set of coordinates of the new vertices $v^{3(1)}(x_1^{3(1)}, x_2^{3(1)})$ and $v^{3(2)}(x_1^{3(2)}, x_2^{3(2)})$

$$\begin{aligned} x_1^3 &= \frac{\pm\sqrt{3}\left(x_2^1 - x_2^2\right)}{2} + \frac{x_1^1 + x_1^2}{2} \\ x_2^3 &= \frac{\pm\sqrt{3}\left(x_1^2 - x_1^1\right)}{2} + \frac{x_2^1 + x_2^2}{2}. \end{aligned} \tag{4}$$

Using these two new points we create, from one 1-Simplex [see Fig. 3(a)] $V^1 = \{v^1, v^2\}$, two new regular 2-Simplexes [see Fig. 3(b)] $V^{2(1)} = \{v^1, v^2, v^{3(1)}\}$ and $V^{2(2)} = \{v^1, v^2, v^{3(2)}\}$. The newly calculated vertices $v^{3(1)}$ and $v^{3(2)}$ with the remaining vertices v^1, v^2 create four additional 1-Simplexes: $\{v^1, v^{3(1)}\}$, $\{v^2, v^{3(1)}\}$, $\{v^1, v^{3(2)}\}$, and $\{v^2, v^{3(2)}\}$. The first two are the 1-faces of the simplex $V^{2(1)}$; the third and fourth are the 1-faces of $V^{2(2)}$.

By evaluating the values of the function at new vertices $v^{3(1)}$ and $v^{3(2)}$, we can select 1-Simplexes in which the function changes sign. In the case presented in Fig. 3(b), there are $\{v^2, v^{3(1)}\}$ and $\{v^1, v^{3(2)}\}$. One is pushed on stack and the second is used in the next algorithm iteration for another new vertices calculation of the next two 2-Simplexes. To avoid a situation where the function is evaluated twice, before we push the simplexes on stack, we check if they have been already traced or put on stack.

B. Algorithm for Three Variables

We assume that the roots of the real function of three variables lie on the continuous surface in 3-D space, which is presented in Fig. 4. In this case, we consider the initial 2-Simplex in 3-D space E [see Fig. 4(a)] $V^2 = \{v^1, v^2, v^3\}$. According to (1), we get the following system of three quadratic equations:

$$\{e^{m4} = e\}, \qquad m = 1, \ldots, 3 \tag{5}$$

[2] The number of m-faces of an N-simplex can be found in column $(m+1)$ of row $(N+1)$ of Pascal's triangle.

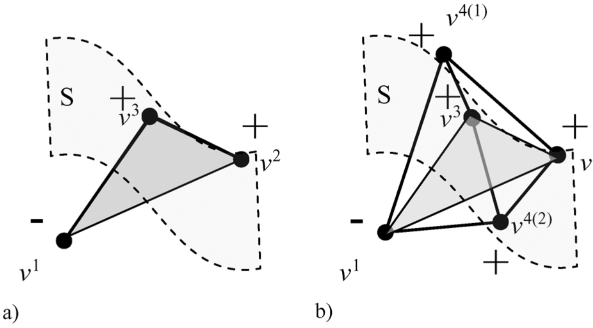

Fig. 4. Case for three variables. (a) Initial 2-Simplex $\{v^1, v^2, v^3\}$. (b) New function evaluation points $v^{4(1)}, v^{4(2)}$ as the vertices of new 3-Simplexes: $\{v^1, v^2, v^3, v^{4(1)}\}$ and $\{v^1, v^2, v^3, v^{4(2)}\}$.

where $e^{m4} = \left((x_1^m - x_1^4)^2 + (x_2^m - x_2^4)^2 + (x_3^m - x_3^4)^2\right)^{1/2}$.

The solutions are the coordinates of new two vertices $v^{4(1)}(x_1^{4(1)}, x_2^{4(1)}, x_3^{4(1)})$ and $v^{4(2)}(x_1^{4(2)}, x_2^{4(2)}, x_3^{4(2)})$ that create two new regular 3-Simplexes [see Fig. 4(b)] $V^{3(1)} = \{v^1, v^2, v^3, v^{4(1)}\}$ and $V^{3(2)} = \{v^1, v^2, v^3, v^{4(2)}\}$. In this case, six new 2-Simplexes $\Delta^2_?$ need to be considered as candidates changing, in its volume, the sign of the function. By analyzing the sign of the function at all vertices of new 2-Simplexes $\Delta^2_?$, we select four of them $\{v^1, v^3, v^{4(1)}\}, \{v^1, v^2, v^{4(1)}\}, \{v^1, v^3, v^{4(2)}\}, \{v^1, v^2, v^{4(2)}\}$. Three of them are pushed on stack and one is used in the next algorithm iteration. Before we push simplexes on stack, we check if they have been already traced or put on stack. The procedure can be easily generalized and applied for functions of more variables.

C. Root Tracing Condition

If a real function at two different points v^i and v^j in a N-dimensional space evaluates values with a different sign

$$\operatorname{sign}\left(F(v^i)\right) \neq \operatorname{sign}\left(F(v^j)\right) \tag{6}$$

then the root of the function lies between points v^i and v^j if

$$\lim(e^{ij}) \to 0 \tag{7}$$

where

$$e^{ij} = \|v^i - v^j\| \tag{8}$$

is an Euclidian distance between two points v^i and v^j defined as

$$\|v^i - v^j\| = \sqrt{\sum_{n=1}^{N} \left(x_n^i - x_n^j\right)^2}. \tag{9}$$

For a complex function, the condition (6) must be extended as

$$\begin{cases} \operatorname{sign}\left(\mathfrak{Re}\left(F(v^i)\right)\right) \neq \operatorname{sign}\left(\mathfrak{Re}\left(F(v^j)\right)\right) \\ \operatorname{sign}\left(\mathfrak{Im}\left(F(v^i)\right)\right) \neq \operatorname{sign}\left(\mathfrak{Im}\left(F(v^j)\right)\right) \end{cases} \tag{10}$$

where $\mathfrak{Re}$ and $\mathfrak{Im}$ indicate the real and imaginary part of the function value, respectively. Considering (7), we have to minimize the initial simplex edge length to ensure proper run of the root tracing. If an initial simplex edge is too long, the algorithm run can be unexpectedly terminated, e.g., due to the multiple roots located between the simplex vertices v.

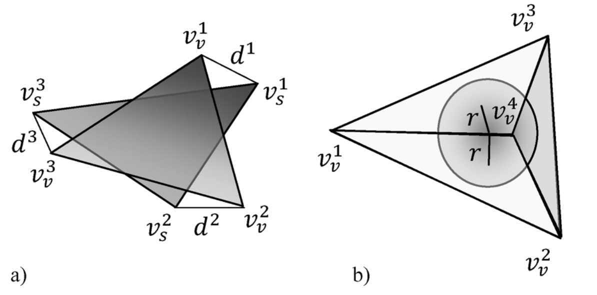

Fig. 5. (a) Summed difference between vertices as a criterion of stack searching for 2-Simplex (11). (b) Inradius of the inscribed sphere inside the 3-Simplex used in a criterion of stack searching (12).

D. Stack Realization

The algorithm uses three stacks (Fig. 1). On the first one, denoted by T^N, the traced N-Simplexes (the coordinates of its geometrical center) consisting in its volume region with the roots are pushed. On the stacks T^{N-1} and W^{N-1}, the coordinates of the vertices of the traced (already evaluated) and waiting for evaluation $(N-1)$-Simplexes are pushed, respectively. In all cases, before simplex is pushed on stack, the stack is searched to check if the simplex is already on it. For the purpose of searching the stack, the simplex similarity criterion needs to be introduced. If $N = 2$, for searching the stacks T^1 and W^1, the following criterion has been used:

$$\sum_{n=1}^{3} \|v_v^n - v_s^n\| = \sum_{n=1}^{3} d_n < e \tag{11}$$

where v_v^n and v_s^n indicate the nth vertex of the verified and searched simplex on stack, respectively. This concept is depicted in Fig. 5(a).

In general, when $N > 2$, for searching T^N, the following criterion is applied:

$$X^N < 2r^N e \tag{12}$$

where X^N is the Euclidian distance between geometrical centers of checked N-Simplex and these searched on stack. The ith coordinate of the geometrical center of the N-Simplex is defined as follows:

$$x_i^c = \frac{\sum_{m=1}^{N+1} x_i^m}{N+1}. \tag{13}$$

The r^N represents the inradius of the inscribed sphere inside the N-Simplex [see Fig. 5(b)] defined as

$$r^N = \frac{h^N}{N+1} \tag{14}$$

where $h^N = e\prod_{n=2}^{N} \sqrt{(n^2-1)/n^2}$.

III. Numerical Result

To verify the presented algorithm, the numerical calculations were carried out. For three chosen microwave transmission

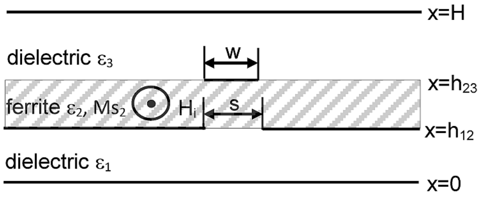

Fig. 6. Cross section of a slab dielectric waveguide of microstrip-sloted FCL. Line dimensions [mm]: $h_{12} = 1, h_{23} = 1.5, H = 3, w = 0.5, s = 1$ [mm]. Material parameters: $\varepsilon_1 = 1, Ms_2 = 340$ kA/m, $H_i = 0, \varepsilon_2 = 13.5, \varepsilon_3 = 1$.

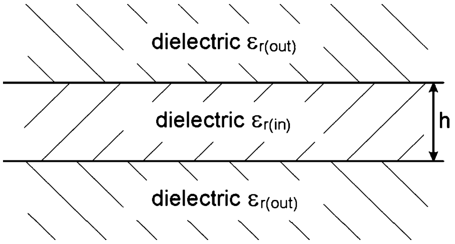

Fig. 7. Cross section of a slab dielectric waveguide. Line dimensions $h = 10$ [μm].

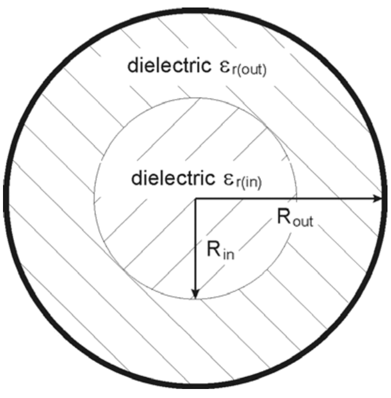

Fig. 8. Cross section of a partially loaded cylindrical waveguide. Line dimensions $R_{\text{in}} = 6.35$, $R_{\text{out}} = 10$ [μm]. Material parameters $\varepsilon_{r(\text{out})} = 1$, $\varepsilon_{r(\text{in})} = 10$.

lines, real and complex dispersion characteristics were found. The first investigated line is the ferrite coupled line (FCL), presented in Fig. 6. The mathematical model of the FCL has been presented, e.g., in [7] and [12]. For this line, the dispersion equation is defined as a determinant of the matrix $detM(f, \gamma) = 0$. The dimension of the matrix M depends on the number of series elements of the tangent currents J, on the strip w, and tangent electrical fields E in slot s taken in computation. Moreover, the elements of the matrix M are defined as integrals. In this case, an evaluation of the function takes a relatively long time.

To show the validity of the presented procedure for problems defined in the complex domain, the next two lines were investigated. The first one is a slab dielectric waveguide (unshielded) presented in Fig. 7. For this structure, guided and leaky modes were considered [8]. The second one is a partially loaded cylindrical waveguide shown in Fig. 8. In this case, guided and complex modes were analyzed [9].

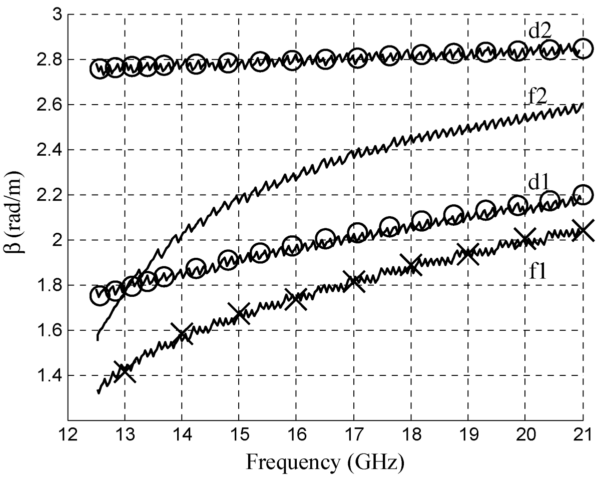

Fig. 9. Dispersion characteristics $F(f, \beta) = 0$ of microstrip-sloted ferrite and dielectric lines (Fig. 6). Simplex edge length 0.0539. X-marks indicate roots calculated with the use of the secant method. The centers of circles indicate the points calculated with the use of CST EM simulator [10].

A. 2D Real-Valued Problem

In the first step, to verify the presented method for the FCL line (Fig. 6), the real roots are calculated. The equation we considered is of the form $F(f, \gamma) = 0$, where $\gamma = j\beta$. We assumed the following limits on variables: frequency $f\epsilon(12.5, 21)$ [GHz] and propagation constant $\beta\epsilon(1, 3)$ [rad/m]. Four dispersion characteristics have been calculated: two of them represent the two first modes for the dielectric line d1, d2 (in this case, magnetization saturation $Ms_2 = 0$ [kA/m]); another two characteristics are the two first modes for the ferrite line ($Ms_2 = 340$ [kA/m]). All computed dispersion characteristics are presented in Fig. 9. To demonstrate the effect of inaccuracies, we have intentionally chosen quite a long edge of the traced simplexes ($e = 0.0539$). As one can see, the curves are not smooth. To obtain the smoother curves, the edge of the initial simplex must be decreased. To validate our method, one characteristic (f1) was calculated additionally using the secant method (points are indicated by X-marks). In calculations, in both methods, we used the same dispersion equation. In order to verify obtained results, the characteristics of dielectric line (d1, d2) were further calculated using the commercial electromagnetic simulator CST MWS [10]; the calculated points are the centers of the circles in Fig. 9.

The next investigated waveguide line is presented in Fig. 7. In the analysis, the following material parameters were assumed $\varepsilon_{r(\text{in})} = 9$ and $\varepsilon_{r(\text{out})} = 1$, and the following limits of the computational domain were imposed: frequency $f\epsilon(0, 2.4)$ [GHz] and propagation constant $\beta\epsilon(1, 3)$ [rad/m].

Due to shorter traced simplexes edge length $e = 0.004$, comparing to FCL line, the presented characteristics are much more smooth (calculated with higher accuracy) (see Fig. 10).

B. 3-D Real-Valued Problem

The next dispersion equation we consider is again the FCL line presented in Fig. 6. Now, we consider the dispersion equation with three variables. Additionally, besides

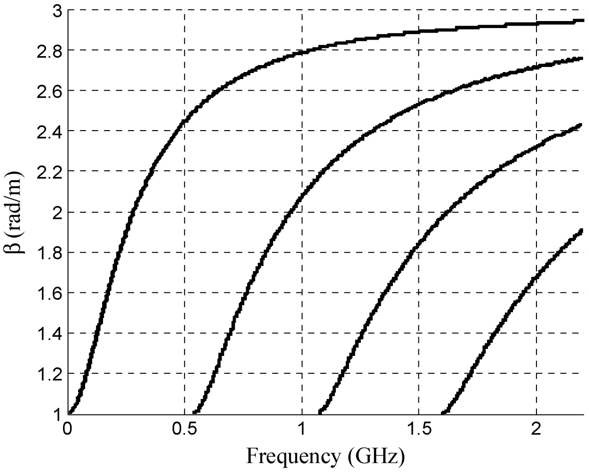

Fig. 10. Dispersion characteristics $F(f,\beta) = 0$ of the slab dielectric waveguide. Line dimension like in Fig. 7. Material parameters $\varepsilon_{r(\mathrm{in})} = 9$ and $\varepsilon_{r(\mathrm{out})} = 1$. Four first pure β modes. Simplex edge length $e = 0.004$.

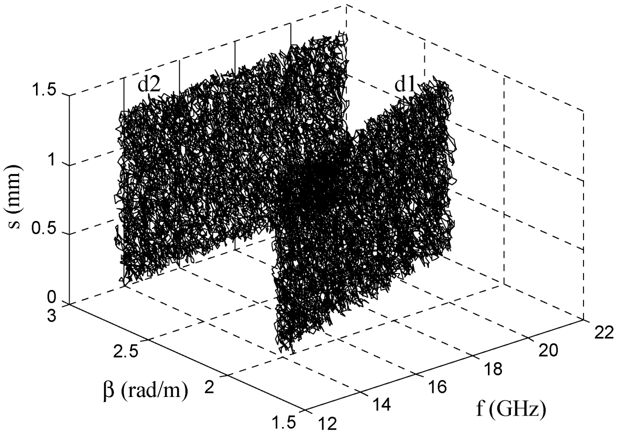

Fig. 11. Dispersion characteristics of microstrip-sloted dielectric line (Fig. 6) in a function of slot width s. Simplex edge length: 0.1118.

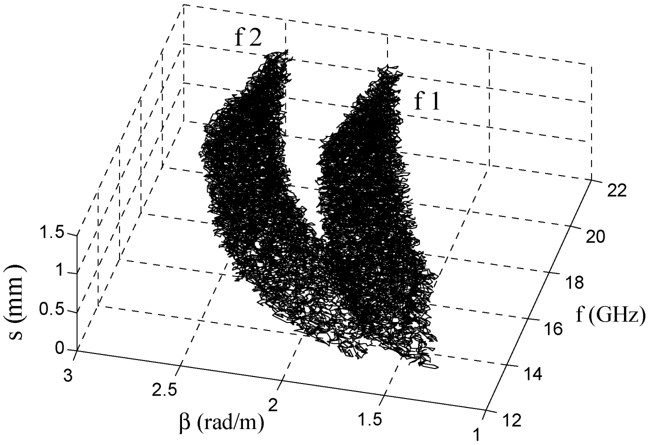

Fig. 12. Dispersion characteristics of microstrip-sloted ferrite line (Fig. 6) in a function of slot width s. Simplex edge length 0.1118.

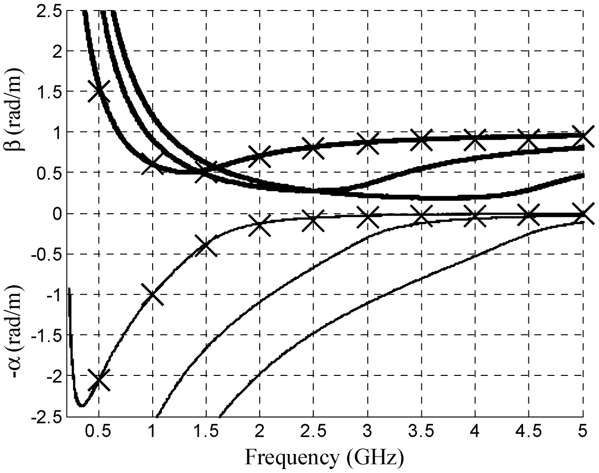

Fig. 13. Dispersion characteristics for the first three leaky modes of slab dielectric waveguide. Material parameters $\varepsilon_{r(\mathrm{in})} = 1$ and $\varepsilon_{r(\mathrm{out})} = 9$. Simplex edge length $e = 0.004$. X-marks indicate roots calculated with the use of Muller's method.

frequency and propagation constant β, we change the width of the slot s, solving the dispersion equation $F(f, j\beta, s) = 0$. We define the following limits for variables: frequency $f\epsilon(12.5, 21)$ [GHz], propagation constant $\beta\epsilon(1, 3)$ [rad/m], slot width $s\epsilon(0.2, 1.5)$ [mm]. In Fig. 11 and 12, dispersion characteristics d1, d2 for dielectric and ferrite f1, f2 lines are presented, respectively. The maximum error of the calculated roots is defined by the edge length of the considered simplexes and for all four presented characteristics is equal to 0.1118.

C. Complex-Valued Problem

As mentioned above, introducing condition (10), the algorithm can be very simply adopted to the analysis in complex domain. For calculating the complex dispersion characteristics in a function of frequency $\gamma(f)$, we work with $N = 3$ variables: f, α, β. Variables α and β represent real and imaginary parts of the complex value γ, and in this method we treat them independently. The complex dispersion characteristics for the lines depicted in Fig. 7 and 8 are calculated. The first line (the slab dielectric waveguide) is calculated for two different material parameters $\varepsilon_{r(\mathrm{in})} = 1, \varepsilon_{r(\mathrm{out})} = 9$ (Fig. 13) and $\varepsilon_{r(\mathrm{in})} = 9, \varepsilon_{r(\mathrm{out})} = 1$ (Fig. 14). The calculations were performed for simplex edge length $e = 0.004$, which reflected in a very good accuracy of the calculated characteristics. In both cases, as can be observed, the leaky waves propagate in the lines in considered frequency band. To validate our method, one characteristic was calculated additionally using Muller's method; the points are indicated by X-marks.

The last analyzed structure was a partially loaded cylindrical waveguide, depicted in Fig. 8. In this line (for a frequency band from about 3.25 up to 5.4 GHz), complex modes are propagated. Above this frequency band, the pure real modes are propagated (see Fig. 15). In order to verify obtained results, these characteristics were further calculated using commercial electromagnetic simulator HFSS [11]; the calculated points are the centers of the circles in Fig. 15.

IV. Comparison With Other Known Root-Finding Methods and Comments on SCVS Accuracy

Root-finding methods that can be found in the literature, like Newton's, secant, bisection method, etc., in their basic form,

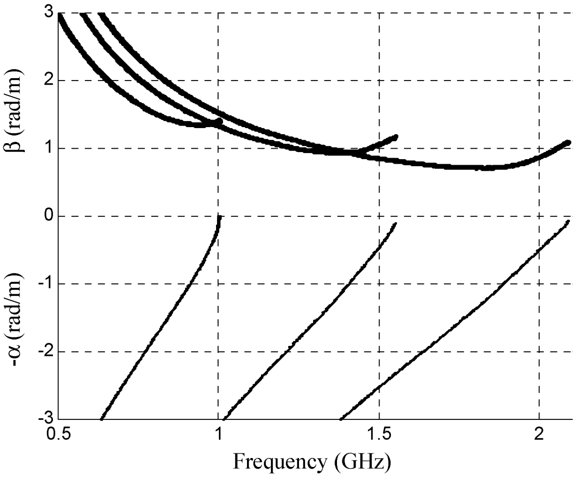

Fig. 14. Dispersion characteristics for the three first leaky modes of the slab dielectric waveguide. Material parameters $\varepsilon_{r(\mathrm{in})} = 9$ and $\varepsilon_{r(\mathrm{out})} = 1$. Simplex edge length 0.004.

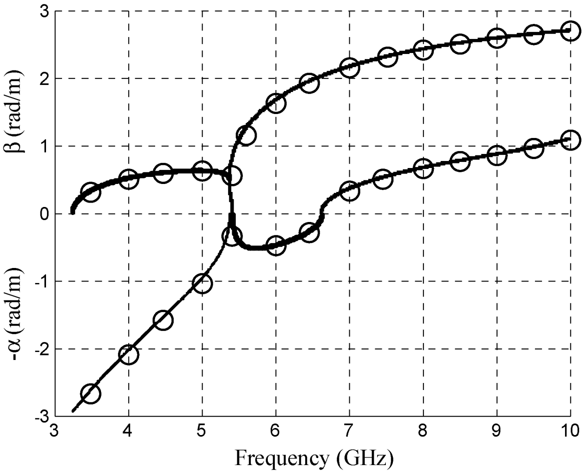

Fig. 15. Dispersion characteristics of two first complex mode for a partially loaded cylindrical waveguide. Simplex edge length $e = 0.004$. The centers of circles indicate the points calculated with the use of the HFSS EM simulator [11].

work with the functions of one variable $f(x)$. All these methods can approach the root with any given precision, repeating the iteration until the root is approximated sufficiently well (with the assumed tolerance ε). Some methods like Newton's method and the secant method can be generalized to multidimensional problems.

The multivariable Newton's and Broyden's (generalized secant method) methods require the calculation of the Jacobian matrix, which increases the numerical cost. In most cases of numerical solutions to electromagnetic problems, we do not have the functions in an explicit analytical form, which results in inefficiency of these methods.

In eigenvalue problems, the root-finding methods for one variable can be successfully applied to root searching in case of the two-variable function $f(x_1, x_2)$. When using these methods we approach the root $f(x_1, x_2) = 0$ searching through one variable, e.g., x_2, assuming a fixed value for the second variable x_1 (Fig. 16).

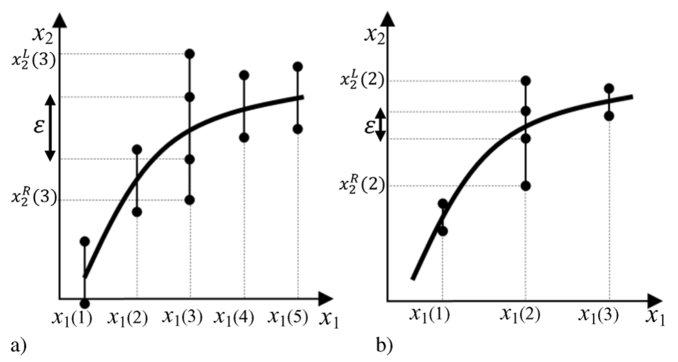

Fig. 16. Standard root-finding method applied for the function of two variables. Exact roots are located on the bold curve. (a) Case of smaller step in variable x_1 and bigger tolerance ε (smaller root accuracy). (b) Case of bigger step in variable x_1 and smaller tolerance ε (better root accuracy).

In many cases, the values of variable x_1 are discretized with a defined step $\Delta x_1 = x_1(i+1) - x_1(i)$, which is selected independently of the assumed searched root accuracy ε for variable x_2. Depending on the method, the root-searching procedure for a certain fixed value of $x_1(i)$ starts with one or two initial function values neighboring the root $f^L = (x_1(i), x_2^L(i))$, $f^R = (x_1(i), x_2^R(i))$.

Our SCVS algorithm is not a typical method for root searching. It starts close to one known root, found with the described tolerance e, and traces the other roots iteratively with the same tolerance e.

Considering the SCVS algorithm in case of the function of two variables $f(x_1, x_2) = 0$, we start with two initial points v_1 and v_2, which define the initial 1-Simplex. The distance between them $e = \|v_1 - v_2\|$, which is the length e of the simplex edge, defines the tolerance of traced roots (Fig. 17). It means that the SCVS algorithm does not converge from the far position(s) of the searched root to the place where the root is approximated with the assumed tolerance. The next root is found with the same tolerance e in one iteration only. Moreover, in this algorithm, we cannot treat the variables x_1 and x_2 independently. The simplex edge e maintains the mutual dependences between values of these variables during the whole run of the algorithm (Fig. 17).

To compare the efficiency (in terms of the number of function evaluations and run time) between our method and the other known methods, we have performed numerical calculations of dispersion characteristics of selected transmission lines.

A. *Efficiency Comparison—2-D Real-Valued Problem*

First, we tested a case for root searching of the real function of two variables. To compare, we used our SCVS algorithm and two other commonly used methods, i.e., the secant method and the method employed in the MATLAB environment ("fzero" command[3]). Table I presents the results of the number of function evaluations and corresponding times of the whole algorithm run during the calculation of the dispersion curve f1 from Fig. 9.

[3] The "fzero" command applied in MATLAB environment uses the combination of bisection, secant, and inverse quadratic interpolation methods.

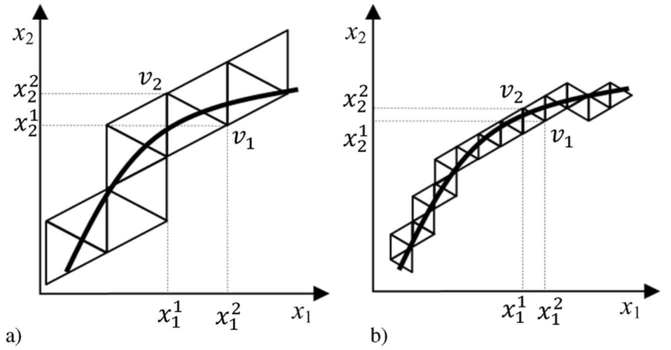

Fig. 17. Run of SCVS algorithm for the function of two variables $f(x_1, x_2)$. Exact roots are located on the bold curve. (a) Case of bigger simplex edge length $e = \|v_1 - v_2\|$ (bigger tolerance, smaller root accuracy). (b) Case of smaller simplex edge length (smaller tolerance, bigger root accuracy).

TABLE I
COMPARISON OF METHODS EFFICIENCY IN TERMS OF ITERATIONS NUMBER/RUN TIME

Method	Tol.=0.053	Tol.=0.01
Secant	1008 / 9 min. 53 sec.	8572 / 1h 18 min
Matlab “fzero”	752 / 7 min. 12 sec.	4218 / 38 min
SCVS	335 / 3 min 19 sec.	1962 / 20 min

We did the tests with two root-searching tolerances, namely, 0.053 and 0.01. First, we ran our SCVS method with the edge length of the traced simplexes equal 0.053. Our algorithm required 335 iterations (function evaluations) to find 335 roots within the band $f\epsilon(12.5, 21)$ (see Fig. 9). Next, we ran the secant and “fzero” algorithms with the frequency step $\Delta f = 0.02537$, which guarantees 335 frequency points within the considered frequency band. We performed the same tests with the tolerance equal 0.01. All the results are collected in Table I. While analyzing the results, we can observe that our algorithm requires the least number of function evaluations, and in consequence, the shortest run time.

B. *Efficiency Comparison—Complex-Valued Problem*

To check how efficient our algorithm is when tracing complex roots, we compared the results with the ones obtained from the Muller’s method [1] (i.e., the secant method generalized to search for complex roots).

The results for a very narrow frequency band within $f \in (4, 4.03)$ GHz for the line from Fig. 8 are presented below in Fig. 18.

First, we ran the SCVS algorithm with the simplex edge length $e = 0.004$. In the frequency band $f \in (4, 4.03)$, our algorithm required only 51 function evaluations. Next, we ran Muller’s algorithm with the frequency step $\Delta f = 0.00588$, which guarantees 51 frequency points within the considered frequency band. Muller’s algorithm required 204 function evaluations (four for every frequency point) (see Table II). The results are depicted in Fig. 18. Having analyzed the results of both algorithms, we concluded that Muller’s algorithm generates roots that create a very smooth curve with regularly discretized frequency, which cannot be observed in our method.

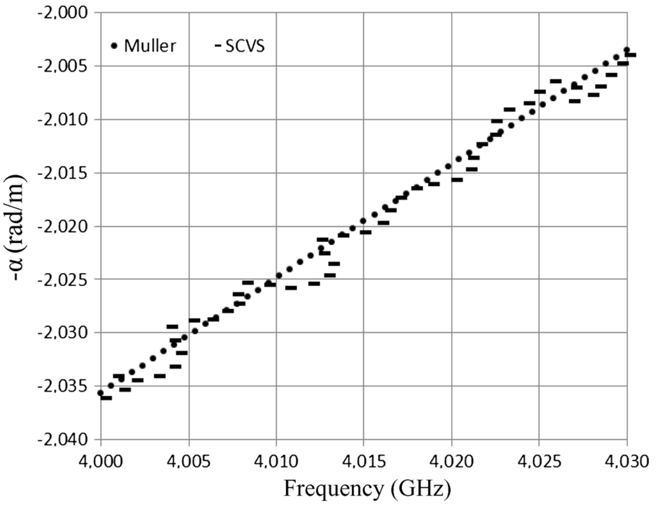

Fig. 18. Dispersion characteristics (α part) of loaded cylindrical waveguide. Simplex edge length $e = 0.004$, Muller algorithm accuracy $a = 0.004$.

TABLE II
COMPARISON OF METHODS EFFICIENCY IN TERMS OF ITERATIONS NUMBER

Method	Tol.=0.004
Muller	204
SCVS	51

Our method requires four times less function evaluations with the presumed accuracy.

V. CONCLUSION

In this paper, we proposed a new approach for efficient and systematic solution of the eigenvalue problems. A novel SCVS algorithm for the roots tracing of real and complex functions of many variables is presented. The algorithm is optimal in the sense of the number of function evaluations. In each iteration, a new generated evaluation point determines a new region where the function has the roots. The accuracy of the procedure depends on the edge length of the initial simplex, and can be defined at the beginning. A few exemplary applications of the presented method for cases of two and three variables for real and complex functions are shown.

ACKNOWLEDGMENT

The authors want to thank Dr. W. Marynowski, Department of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, Gdańsk, Poland, for the provided results, obtained with the use of CST MWS.

REFERENCES

[1] D. E. Muller, “A method for solving algebraic equations using an automatic computer,” *Math. Tables and Other Aids to Computat.*, vol. 10, pp. 208–215, 1956.

[2] F. Liu, X. Chen, and Z. Huang, “Parallel genetic algorithm for finding roots of complex functional equation,” *Pervasive Comput. Appl.*, pp. 542–545, Jul. 26–27, 2007.

[3] X. Zhang, M. Wan, and Z. Yi, “A constrained learning algorithm for finding multiple real roots of polynomial,” in *Comput. Intell. Design*, Oct. 17–18, 2008, vol. 2, pp. 38–41.

[4] I. Alolyan, “An algorithm for finding all zeros of vector functions,” *Bull. Australian Math. Soc.*, vol. 77, pp. 353–363, 2008.

[5] M. Mrozowski, “An efficient algorithm for finding zeros of a real function of two variables,” *IEEE Trans. Microw. Theory Tech.*, vol. 36, no. 3, pp. 601–604, Mar. 1988.

[6] H. S. M. Coxeter, *Regular Polytopes*, 3rd ed. New York: Dover, 1973.

[7] J. Michalski, M. Mazur, and J. Mazur, “Scattering in a section of ferrite-coupled microstrip lines: Theory and application in nonreciprocal devices,” *Proc. Inst. Elect. Eng.—Microw., Antennas, Propag.*, vol. 149, no. 56, pp. 286–290, Oct.–Nov. 2002.

[8] P. Kowalczyk and M. Mrozowski, “A new conformal radiation boundary condition for high accuracy finite difference analysis of open waveguides,” *Opt. Exp.*, vol. 15, no. 20, pp. 12605–12618, 2007.

[9] M. Mrozowski*, Guided Electromagnetic Waves: Properties and Analysis*. Baldock, U.K.: Res. Studies Press Ltd., 1997.

[10] CST Microwave Studio. CST, 2011. [Online]. Available: https://www.cst.com

[11] HFSS 3-D Full-Wave Electromagnetic Field Simulation. ANSYS, 2011. [Online]. Available: http://www.ansoft.com/products/hf/hfss/

[12] W. Marynowski and J. Mazur, “Treatment of the three strip coplanar lines on the ferrite,” in *XVII Int. Microw., Radar, Wireless Commun. Conf. (MIKON 2008)*, Wroclaw, Poland, 2008.

Jerzy Julian Michalski (M’07) was born in Rypin, Poland, in 1974. He received the M.S. degree in electrical engineering from the Gdańsk University of Technology, Gdańsk, Poland, in 1998, and the Ph.D. degree in electrical engineering from the Telecommunications Research Institute, Warsaw, Poland, in 2002.

He is currently with Research and Development, TeleMobile Electronics Ltd., Gdynia, Poland. He has authored numerous software tools for GSM, DCS, and EDGE communication systems for diagnosis of damages in the base-station modules. His research has thus far concerned omega pseudochiral materials and ferrite materials longitudinally magnetized in applications for microwave devices. His main research interest concerns new efficient methods based on artificial intelligence algorithms applied in the automatic tuning of filters and diplexers.

Piotr Kowalczyk was born in Wejherowo, Poland, in 1977. He received the M.S. degree in applied physics and mathematics and Ph.D. degree in electrical engineering from the Gdańsk University of Technology, Gdańsk, Poland, in 2001 and 2008, respectively.

He is currently with Microwave and Antenna Engineering, Technical University, Gdańsk, Poland. He has authored or coauthored numerous publication in some of the highest impact factor journals. His research is focused on scattering and propagation of electromagnetic-wave problems.