Complex Root Finding Algorithm Based on Delaunay Triangulation

PIOTR KOWALCZYK, Gdansk University of Technology

A simple and flexible algorithm for finding zeros of a complex function is presented. An arbitrary-shaped search region can be considered and a very wide class of functions can be analyzed, including those containing singular points or even branch cuts. The proposed technique is based on sampling the function at nodes of a regular or a self-adaptive mesh and on the analysis of the function sign changes. As a result, a set of candidate points is created, where the signs of the real and imaginary parts of the function change simultaneously. To verify and refine the results, an iterative algorithm is applied. The validity of the presented technique is supported by the results obtained in numerical tests involving three different types of functions.

Categories and Subject Descriptors: G.1.5 [**Numerical Analysis**]: Roots of Nonlinear Equations—*Iterative methods*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Complex root finding, global algorithm, Delaunay triangulation

ACM Reference Format:
Piotr Kowalczyk. 2015. Complex root finding algorithm based on Delaunay triangulation. ACM Trans. Math. Softw. 41, 3, Article 19 (May 2015), 13 pages.
DOI: http://dx.doi.org/10.1145/2699457

1. INTRODUCTION

Though there are many publications on complex root-finding techniques, they generally only consider a narrow class of functions or a restricted region of analysis. Standard schemes for local root finding like Newton's [Abramowitz and Stegun 1972], Muller's [Press et al. 1992] or simplex [Dantzig 1963] methods require initial points to start the routine. If the function has local extrema the starting point must be quite accurate to ensure convergence of the process. On the other hand, algorithms tracking the root in a function with an extra parameter are very efficient [Gritton et al. 2001; Michalski and Kowalczyk 2011], but their flexibility is limited and again initial points are required. Further, some problems can appear for multiple-valued functions (e.g., square or cube roots) in the neighborhood of branch cuts. Nevertheless, in general, local roots can be successfully estimated using one of the known algorithms.

Global root finding, on the other hand, is significantly more difficult. For simple polynomial functions, algorithms based on the Sturm sequence method enhanced by the Routh theorem [Pinkert 1976] or on the splitting circle method introduced by Schonhage [1982], can be applied with a very high efficiency. In many technical problems, however, the polynomial approximation is insufficient. A generalization of these procedures is proposed in Long and Jiang [1998] and Wu et al. [2010], but the function still needs to be free of singularities and branch cuts in the analyzed region, and the same limitation applies to mesh methods [Wan 2011]. To the author's knowledge, there

This work was supported under funding for Statutory Activities for the Faculty of Electronics, Telecommunication and Informatics, Gdansk University of Technology.
Author's address: Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Narutowicza 11/12, 80233 Gdansk.

DOI: http://dx.doi.org/10.1145/2699457

is no flexible and effective global root-finding method. Therefore, in particular cases, it becomes necessary to implement difficult and complex routines (e.g., based on the genetic algorithm [Yu-Bo 2009]).

In this article, a simple algorithm is presented that allows zeros of a complex function to be found in an arbitrary-shaped region. A very wide class of functions can be analyzed, including those containing singular points or even branch cuts. The proposed technique is based on sampling of the function using a regular or a self-adaptive mesh and analysis of the sign changes between the nodes. As a result, a set of candidate points is created, where the signs of the real and imaginary parts of the function change simultaneously. Next, the points are verified using Cauchy's Argument Principle. Finally, the accuracy of the estimates of the verified roots is improved. For this, many algorithms are suitable, but their efficiency strongly depends on the type of function. In this article, a simple iterative method based on the rational function is proposed.

The validity of the technique presented is supported by the results obtained in numerical tests involving three different types of functions commonly used in electromagnetic and optical engineering.

2. ALGORITHM

The algorithm can be divided into three separate stages: preliminary estimation, verification and final refinement. The preliminary estimation procedure provides a set of candidate points where the signs of the real and imaginary parts of the function change simultaneously. In the second stage, the validity of the solution is checked using Cauchy's Argument Principle and, finally, in the third stage, the accuracy of the estimates of the roots is improved.

2.1. Simple Preliminary Estimation

Denoting the function considered by $f(z)$ and the search region by $\Omega \subset \mathbb{C}$, the main idea of the method can be simply described in terms of the following steps:

(1) In the first step, the smallest distance Δr between all the roots, singularities and branch cuts is assumed.
(2) Next, region Ω is covered with a triangular mesh and the nodes are collected in a set of points denoted by $V = \{v_1, v_2, \ldots, v_N\}$. A honeycomb arrangement (equilateral triangles) of the points provides the highest efficiency of the algorithm, however any other configuration is possible, until the longest side of all the triangle is smaller than Δr.
(3) The function is evaluated for each point v_n and the values are stored in set $\boldsymbol{F} = \{f_1, f_2, \ldots, f_N\}$.
(4) For each of the triangles, the real and imaginary parts of the function can be approximated separately by a plane determined by three points (vertices of the triangle). The approximation is not holomorphic, but it is continuous in Ω. Given this, two curves (consisting of line segments) representing zeros of the real and imaginary parts of the function can be constructed: $C_R = \{z \in \Omega : Re(f(z)) = 0\}$ and $C_I = \{z \in \Omega : Im(f(z)) = 0\}$. For the example function, these curves are presented in Figure 1.
(5) All the points where C_R and C_I curves cross (or are closer then Δr) are collected in set $S = \{s_1, s_2, \ldots, s_M\}$ and the preliminary estimation process is complete.

At this stage of the algorithm, set S contains roots as well as singularities of the function or artificial roots at branch cuts (the sign changes can be a result of cuts only).

It is clear that the accuracy of the candidate points is determined by the parameter Δr. The results are, however, improved in the last stage of the algorithm. At this stage, the value of Δr only needs to be sufficient to separate different solutions. It should be

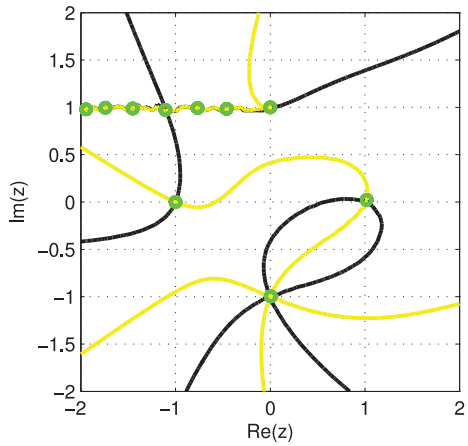

Fig. 1. The curves C_R (black) and C_I (yellow) are obtained in the preliminary estimation process for $f(z) = (z-i)^{1/2}(z+1)(z+i)^2(z-1)^{-1}$. The set S contains 10 candidate points (green circles).

noted that multiroots and single roots which are located in a distance smaller than Δr are indistinguishable for the algorithm.

An important advantage of the method is the simple parallelization of the calculations in Step (3). In this step, the function can be evaluated at quite high number of points. Hence, this part of the algorithm can be very time consuming and the possibility of parallelizing the process may be very useful.

2.2. Preliminary Estimation Based on Delaunay Triangulation

Suppose the evaluation of the function is time consuming, so that some extra time can be spent on finding the best locations of nodes of the mesh. This technique is particularly efficient for more complex forms of function (especially evaluated numerically).

In such a case, the mesh is modified iteratively, however, the main idea of the preliminary estimation is the same. The improved version of the algorithm is similar, but it consists of more steps:

(1) In the first step, region Ω is covered with a dense initial mesh and the nodes are collected in a set of points denoted by $V = \{v_1, v_2, \ldots, v_N\}$. Usually, as it is shown in the numerical examples, the four vertices of a rectangle are sufficient to initiate the procedure.
(2) By applying Delaunay triangulation to set V, its points become the vertices of triangles. These triangles must completely cover the domain Ω. To improve the efficiency of this process a modified version of the triangulation can be applied[1] [Miller et al. 2011].
(3) The function is evaluated for each point v_n and the values are stored in set $F = \{f_1, f_2, \ldots, f_N\}$.
(4) The triangles with edges of a length greater than the assumed accuracy Δr are analyzed. If the signs of the real or imaginary parts of the function are different at the ends of any of these edges, then an extra point is added to V, namely, a midpoint of the corresponding edge.[2]

[1]The algorithm produces a Delaunay mesh with guaranteed optimal mesh size and quality.
[2]Splitting the edge at the midpoint is more effective than at the approximated root, especially in the first few retriangulations. In one triangle (along one edge), the signs of the real and imaginary parts of the functions

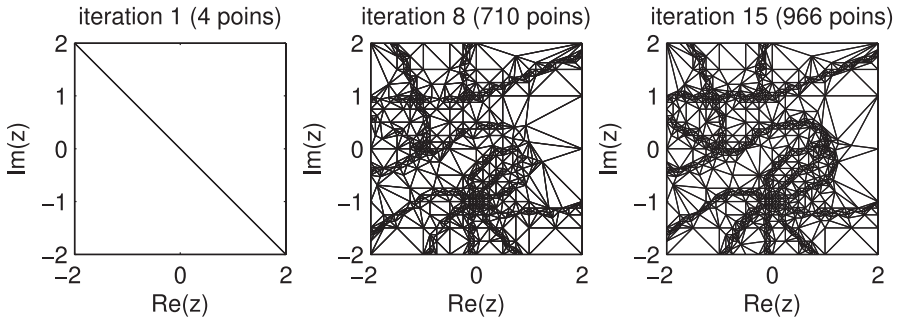

Fig. 2. An example of the preliminary estimation algorithm application for $f(z) = (z-i)^{1/2}(z+1)(z+i)^2(z-1)^{-1}$ and $\Omega = \{z \in \mathbb{C} : -2 < \Re(z) < 2 \wedge -2 < \Im(z) < 2\}$ is presented. For $\Delta r = 0.1$, the procedure stops after 15 iterations and 966 evaluations of the function.

(5) If in the previous step, any new points have been added to V, the procedure is repeated from Step (2). An example iteration process is illustrated[3] in Figure 2.
(6) For each of the triangles, the real and imaginary parts of the function can be approximated separately by a plane determined by three points (vertices of the triangle). The approximation is not holomorphic, but it is continuous in Ω. Given this, two curves (consisting of line segments) representing zeros of the real and imaginary parts of the function can be constructed: $C_R = \{z \in \Omega : Re(f(z)) = 0\}$ and $C_I = \{z \in \Omega : Im(f(z)) = 0\}$ (exactly like in previous section).
(7) All the points where C_R and C_I curves cross (or are closer then Δr) are collected in set $S = \{s_1, s_2, \ldots, s_M\}$ and the preliminary estimation process is complete (exactly like in previous section).

To improve the efficiency of the preliminary process, it is convenient to remove any new points which are very close to the old ones (already present in V), before the next application of Delaunay triangulation (after Step (4)). If branch cuts of the function are present in the considered region there can be quite a large number of points in S, and it is efficient to reduce this number by grouping the points using the technique described in the appendix.

Also in this variant of the preliminary estimation the simple parallelization of the calculations in Step (3) is possible. In this step, the function is only evaluated at new points of V; nevertheless, the number of calculations required may be quite high. Owing to the assumption of complexity of the function, this part of the algorithm is the most time consuming.

2.3. Verification

Once set S is obtained, the points must be classified to avoid false solutions. This is a very important part of the entire process. The iterative method improving the solution accuracy, applied directly at candidate points, can get stuck in a local minimum or not converge at all. Moreover, without the verification stage it is not possible to determine whether all the roots have been found.

---

can change more than once and this intuitive approach turns out to significantly increase a possibility of missing some roots.

[3]The functions used to illustrate the process is $f(z) = (z-i)^{1/2}(z+1)(z+i)^2(z-1)^{-1}$.

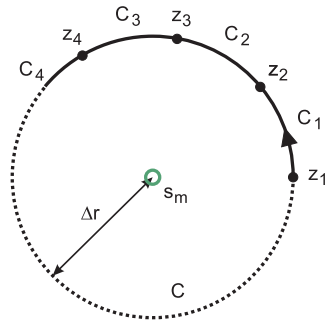

Fig. 3. Contour C is divided into P segments.

In order to verify the validity of the root, Cauchy's Argument Principle is applied for each point s_m. According to this principle, the integral

$$q = \frac{1}{2\pi i} \oint_C \frac{f'(z)}{f(z)} dz, \tag{1}$$

represents a change in the argument of the function $f(z)$ over a closed contour C. In general q is a sum of all zeros counted with their multiplicities, minus the sum of all poles counted with their multiplicities. For a multiple-valued function (e.g., a square or cube root), the integration must involve all Riemann surfaces to close the curve.

Assuming that the contour C is a circle of radius Δr centered at s_m and the candidate point represents only one root or singularity, the value of q can have only fixed values. If the function is single-valued, the parameter q can be:

—a positive integer - root of order q,
—a negative integer - singularity of order $-q$,
—zero - regular point.

If the function is multiple-valued, the parameter q can be fractional, as C can be open at the branch cut: the integration is performed over one Riemann surface only and the discontinuity at the cut is neglected (e.g., for the integral over $s_m = 0$, $q = 1/2$ for the square root function and $q = 1/3$ for the cube root). In this case, the calculations must be performed for each Riemann surface and the values of q (obtained for different surfaces) summed. Only this approach can unambiguously determine the nature of the candidate point.

In numerical calculations, it is convenient to divide the contour C into P segments $C = \bigcup_{p=1}^{P} C_p$ (see Figure 3). Writing the function in the exponential form $f(z) = M(z)\exp(i\Psi(z))$, the parameter q can be expressed as follows:

$$\begin{aligned} q &= \frac{1}{2\pi i} \sum_{p=1}^{P} \int_{C_p} \left[\frac{M'(z)}{M(z)} + i\Psi'(z) \right] dz \\ &= \frac{1}{2\pi i} \left\{ \sum_{p=1}^{P-1} \left[\ln \frac{M(z_{p+1})}{M(z_p)} + i\Psi(z_{p+1}) - i\Psi(z_p) \right] + \ln \frac{M(z_1)}{M(z_P)} + i\Psi(z_1) - i\Psi(z_P) \right\}, \end{aligned} \tag{2}$$

where $z_p \in C$.

Table I. The Verification of the Example set S Presented in Figure 1

m	s_m	q	P	Verification
1	$1.02 + 0.02i$	-1	9	singularity
2	$-1.00 - 0.00i$	1	9	root
3	$0.00 + 1.00i$	0.5	54	root
4	$0.00 - 1.00i$	2	19	root
5	$-1.74 + 1.00i$	$3.53 \cdot 10^{-17}$	102	regular point
6	$-1.95 - 0.98i$	$-1.33 \cdot 10^{-17}$	102	regular point
7	$-1.45 + 0.99i$	$-4.42 \cdot 10^{-18}$	102	regular point
8	$-0.46 + 0.99i$	$-1.33 \cdot 10^{-17}$	102	regular point
9	$-1.11 + 0.97i$	$-5.74 \cdot 10^{-17}$	102	regular point
10	$-0.77 + 0.99i$	$-1.77 \cdot 10^{-17}$	102	regular point

As a result, the integral becomes the algebraic sum of the partial increases or decreases in the argument of the function along contour C

$$q = \frac{1}{2\pi} \sum_{p=1}^{P} \Delta\Psi_p, \tag{3}$$

where $\Delta\Psi_p = \Psi(z_{p+1}) - \Psi(z_p)$ for $p = 1, \ldots, P-1$ and $\Delta\Psi_P = \Psi(z_1) - \Psi(z_P)$.

The procedure is simple if no branch cuts a cross the contour. In such case, the points z_p can be distributed evenly along C; otherwise, the cuts must be detected and eliminated. Let the argument changes at the branch cut be less than $\Delta\theta$ (e.g., if the function contains $\sqrt[R]{(\cdot)}$ then $\Delta\theta = 2\pi/R$). Then, the segments C_p whose $|\Delta\Psi_p|$ is greater than $\Delta\theta$ are divided in half, until the length of C_p is greater than ε. Finally, these very short segments (of length less than ε) are neglected in the sum (3). This operation eliminates the improper $\Delta\Psi_p$ (which, in fact, represents an artificial change in the argument at a cut).

Table I summarizes the results of the verification process, performed for the example set S. Initially, the contour is divided into $P = 3$ segments, and $\varepsilon = 2.220446049250313 \cdot 10^{-16}$ (machine precision). Here, the value of parameter q for the point s_3 is $q = 0.5 + 0.5$ (0.5 for each of two Riemman surfaces $w = \pm\sqrt{z}$).

2.4. Limitations

The regular mesh has a very clear guarantee of correctness - if the step Δr is smaller than the smallest distance between the roots and poles of the function then none of the roots can be missed. If the distance is too long and the function changes sign two (or more, but even) times then the approximation of the function is improper. As a result some of the roots can be omitted. Unfortunately, just like for other established methods (e.g., bisection for real function of one variable), there is no clear recipe for the estimation of such Δr a priori. In practice, Δr must be chosen by a user experimentally (in sequential iterations).

The verification process described in the previous section checks the validity of the solutions, however it does not guarantee that all the roots are found. To reduce the risk of root missing the integration (1) over the whole considered area can be applied. Assuming that C is a boundary of Ω the integral (1) must be equal to the sum of all parameters q, obtained for each s_m. Unfortunately, if the contour integration over the considered region is zero then the region can be free of roots and poles, but there is still a risk that the region contains an equal number of roots and poles. The integration can be extended to higher moments [Lamparillo and Sorrentino 1975], which can further reduce that risk (the second moment eliminates the problem for single pair of root-poles

[Zieniutycz 1983]). However, it must be emphasized that none of those procedures can guarantee that all the roots will be found. For the Delaunay-based technique, the risk of root missing can be higher due to irregular discretization of the domain (a self-adaptive mesh in preliminary estimation).

2.5. Final Refinement

Once the roots have been obtained with reasonably good accuracy, the estimates can be improved by many different techniques; these include Newton's, Muller's, or simplex methods. When the considered region contains singularities, however, a rational function approximation is more suitable. In this article, an iteration technique based on a rational function is proposed. The rate of convergence of this method is about 1.84 (the same as Muller technique [Stewart 1994]). However, the asymptotic error constant depends on the analyzed function and can be smaller (than, e.g., Muller technique) for functions containing singularities.

The simplest rational function can be written in the following form:

$$P(z) = \frac{z-a}{bz-c}, \tag{4}$$

where parameters a, b and c are complex numbers.

The procedure is performed for each root from set S and includes the following steps:

(1) For a fixed s_m the three nearest points are chosen from V and denoted by v_α, v_β, and v_γ. The corresponding values of the function can be obtained from F and denoted by f_α, f_β, and f_γ, respectively.
(2) Substituting the points and their values into the rational function (4), the following matrix equation can be formulated

$$\begin{bmatrix} v_\alpha f_\alpha & -f_\alpha & 1 \\ v_\beta f_\beta & -f_\beta & 1 \\ v_\gamma f_\gamma & -f_\gamma & 1 \end{bmatrix} \begin{bmatrix} b \\ c \\ a \end{bmatrix} = \begin{bmatrix} v_\alpha \\ v_\beta \\ v_\gamma \end{bmatrix}. \tag{5}$$

(3) Since the function in (4) is equal to zero for $z = a$, a new approximation of the root can be obtained from the solution of Eq. (5), as $v_\eta = a$.
(4) The function value for the new point is evaluated as $f_\eta = f(v_\eta)$.
(5) The magnitudes of the function at v_α, v_β, and v_γ are compared and the point with the highest value is replaced by v_η.
(6) If the distances between points v_α, v_β, and v_γ are greater than a given accuracy ε the process is repeated from (2).

As was mentioned at the beginning of this section, the refinement can be performed using different algorithms and its efficiency varies depending on the function type. The simple nature of the proposed algorithm means that the refinement process can be helped by other techniques (in Steps (2) and (3)) without affecting the structure of the whole root-finding process.

3. NUMERICAL EXAMPLES

In order to demonstrate the validity of the method, the algorithm is applied to three different problems from the fields of microwave and optical engineering. The algorithm is implemented in the Matlab Environment (the Delaunay triangulation function is built-in), so the code is very simple. Since none of the known (global) algorithms can be used for a multiple-valued function containing branch points and singularities no comparison of efficiency is provided. Unlike simple functions used to illustrate the algorithm (see Figure 2 and Figure 1), all functions selected are more complex and time consuming to evaluate, so the time required for iterative refinement of the triangular

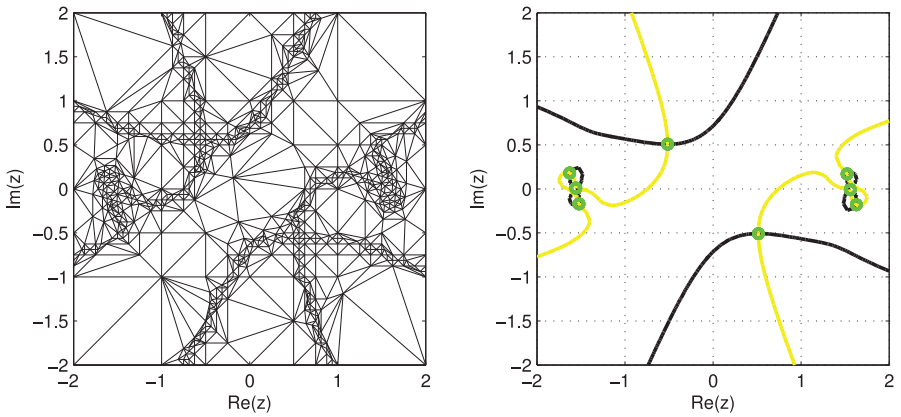

Fig. 4. The final triangulation (left-hand side) and the real-zero and imaginary-zero curves (right-hand side) for the surface wave propagation problem.

Table II. The Results of the Preliminary Estimation and Verification for the Surface Wave Propagation Problem

m	s_m	q	P	verification
1	$-0.52+0.51i$	1	9	root
2	$0.52-0.51i$	1	9	root
3	$-1.52-0.17i$	1	9	root
4	$1.52+0.17i$	1	9	root
5	$-1.56+0.01i$	-2	17	singularity
6	$1.56-0.01i$	-2	17	singularity
7	$-1.63+0.18i$	1	9	root
8	$1.63-0.18i$	1	9	root

mesh is small compared to the total CPU time for the entire process. To assess the algorithm efficiency, the number of function evaluations is given and compared, for each analyzed function type, with the case involving a dense regular initial mesh.

3.1. Surface Waves in a Microstrip Antenna

One of the most frequently cited examples of complex transcendental equations in microwave electronics originates from the analysis of surface waves in microstrip lines or antennas [Long and Jiang 1997; Wu et al. 2010]. The function obtained from the eigenvalue problem describing propagation in such structures can be written in the following form:

$$f(z) = \varepsilon_r^2 z^2 + z^2 \tan^2 z - \varepsilon_r^2 (k_0 h)^2 (\varepsilon_r \mu_r - 1), \tag{6}$$

where $k_0 = 2\pi f/c$ and $c = 3 \cdot 10^8$[m/s]. The typical set of material parameters is $\varepsilon_r = 5 - 2i$ and $\mu_r = 1 - 2i$, while for the numerical analysis, h and f are assigned values of 0.01[m] and 1[GHz], respectively.

The function (6) is analyzed in the region $\Omega = \{z \in \mathbb{C} : -2 < \Re(z) < 2 \wedge -2 < \Im(z) < 2\}$ with $\Delta r = 0.1$. In the preliminary estimation process, the function is evaluated 693 times (14 iterations). The results from this first stage and the verification are presented in Figure 4 and Table II.

The accuracy of the solutions is improved in the final refinement assuming $\varepsilon = 2.220446049250313 \cdot 10^{-16}$ (machine precision) and the final results are listed in Table III (the 2 points representing singularities having been rejected).

Table III. The Final Results for the Surface Wave Propagation Problem

m	$\overline{s_m}$	$\lvert f(\overline{s_m})\rvert$	$\lvert f(\overline{s_m}+\delta_m)\rvert$	K
1	$-0.515113098774213 + 0.507111597183436i$	$5.62 \cdot 10^{-15}$	$3.06 \cdot 10^{-9}$	6
2	$0.515113098774213 - 0.507111597183436i$	$5.62 \cdot 10^{-15}$	$3.06 \cdot 10^{-9}$	6
3	$-1.520192977783856 - 0.173670452372664i$	$2.07 \cdot 10^{-14}$	$1.24 \cdot 10^{-7}$	9
4	$1.520192977783856 + 0.173670452372664i$	$2.07 \cdot 10^{-14}$	$1.24 \cdot 10^{-7}$	9
7	$-1.624715288303687 - 0.182095877325762i$	$8.95 \cdot 10^{-14}$	$1.25 \cdot 10^{-7}$	9
8	$1.624715288303687 + 0.182095877325762i$	$8.95 \cdot 10^{-14}$	$1.25 \cdot 10^{-7}$	9

Note: K is a number of iterations in the final refinement and $\delta_m = \overline{s_m} \cdot 10^{-10}$.

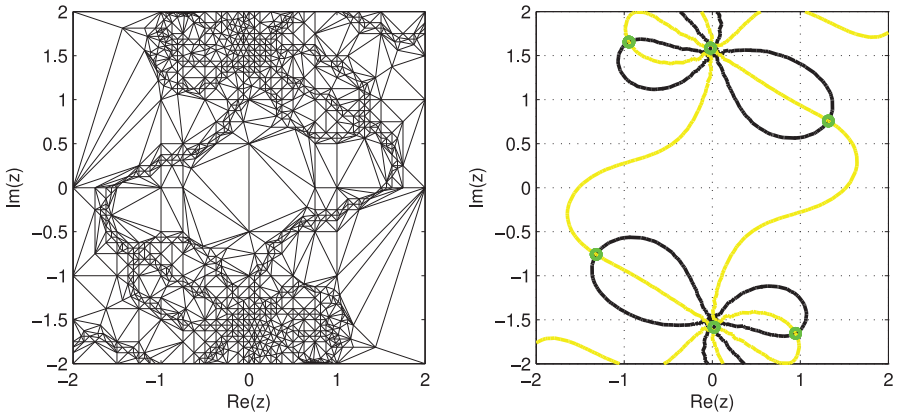

Fig. 5. The final triangulation (left-hand side) and the real-zero and imaginary-zero curves (right-hand side) for the complex waves propagation problem.

In electromagnetic waveguide problems, the sign of the solution represents the direction of propagation, hence both signs are meaningful.

3.2. Leaky Waves in an Optical Fiber

The second example concerns propagation processes (specifically radiation) in a standard optical fiber of radius $R = 0.5[\mu\text{m}]$. Constant values are assigned to the refractive index of the core, $n_2 = 2.9$, and the cladding, $n_1 = 1.55$. In addition, in this example, to ensure continuity of the fields at the boundary the following determinant function must be equal to zero [Ghatak and Thyagarajan 1998]:

$$f(z) = \begin{vmatrix} H_m^{(2)}(\kappa_1 R) & 0 & -J_m(\kappa_2 R) & 0 \\ 0 & n_1 H_m^{(2)}(\kappa_1 R)0 & 0 & -n_2 J_m(\kappa_2 R) \\ \frac{zmH_m^{(2)}(\kappa_1 R)}{R\kappa_1^2} & \frac{k_0 n_1 H_m'^{(2)}(\kappa_1 R)}{\kappa_1} & -\frac{zmJ_m(\kappa_2 R)}{R\kappa_2^2} & -\frac{k_0 n_2 J_m(\kappa_2 R)}{\kappa_2} \\ \frac{k_0 n_1^2 H_m'^{(2)}(\kappa_1 R)}{\kappa_1} & -\frac{zmH_m^{(2)}(\kappa_1 R)}{\kappa_1^2} & -\frac{k_0 n_2^2 J_m'(\kappa_2 R)}{\kappa_2} & \frac{zmJ_m(\kappa_2 R)}{\kappa_2^2} \end{vmatrix}, \tag{7}$$

where z represents a propagation constant, $J_m(\cdot)$ is a Bessel function of the first kind and $H_m^{(2)}(\cdot)$ is a Hankel function of the second kind. The coefficients κ_1 and κ_2 are defined as follows $\kappa_1 = \sqrt{z^2 + k_0^2 n_1^2}$, $\kappa_2 = \sqrt{z^2 + k_0^2 n_2^2}$, where $k_0 = 2\pi f/c$.

The numerical tests are performed for $m = 1$ and $f = 50[\text{THz}]$. The region considered is $\Omega = \{z \in \mathbb{C} : -2 < \Re(z) < 2 \wedge -2 < \Im(z) < 2\}$ and Δr is assigned a value of 0.1.

In this case, the preliminary estimation requires 15 iterations and 1107 evaluations of the function (see Figure 5).

Table IV. The Results of the Preliminary Estimation and Verification for the Leaky Wave Propagation Problem

m	s_m	q	P	verification
1	$1.32+0.76i$	1	19	root
2	$-0.02+1.59i$	-3	58	singularity
3	$-0.95+1.66i$	1	18	root
4	$-1.32-0.76i$	1	19	root
5	$0.02-1.59i$	-3	58	singularity
6	$0.95-1.66i$	1	18	root

Table V. The Final Results for the Leaky Wave Propagation Problem

m	$\overline{s_m}$	$\lvert f(\overline{s_m})\rvert$	$\lvert f(\overline{s_m}+\delta_m)\rvert$	K
1	$1.317970518320856+0.758576155985806i$	$2.72\cdot 10^{-16}$	$1.27\cdot 10^{-10}$	9
3	$-0.946410082621815+1.654062749423636i$	$3.26\cdot 10^{-16}$	$2.28\cdot 10^{-10}$	9
4	$-1.317970518320856-0.758576155985806i$	$2.72\cdot 10^{-16}$	$1.27\cdot 10^{-10}$	9
6	$0.946410082621815-1.654062749423636i$	$3.26\cdot 10^{-16}$	$2.28\cdot 10^{-10}$	9

Note: K is a number of iterations in the final refinement and $\delta_m=\overline{s_m}\cdot 10^{-10}$.

The results obtained for the function (7) are listed in Tables IV and V. The same results can be obtained directly from discreet methods of computational electrodynamics (then roots s_m are represented by eigenvalues of the corresponding matrix operator) [Kowalczyk and Mrozowski 2007].

3.3. Complex Waves in an Inhomogeneous Circular Waveguide

The last example concerns a complex wave propagation problem in a circular waveguide of radius b, coaxially loaded (lossless dielectric with permittivity ε_r) with a cylinder of radius a [Mrozowski and Mazur 1992; Mrozowski 1997]. To ensure continuity of the fields at the boundary, the following determinant function must be equal to zero:

$$f(z)=\begin{vmatrix} J_m(\kappa_1 a) & 0 & -J_m(\kappa_2 a) & -Y_m(\kappa_2 a) & 0 & 0 \\ 0 & \frac{J_m(\kappa_1 a)}{\eta_1} & 0 & 0 & -\frac{J_m(\kappa_2 a)}{\eta_2} & -\frac{Y_m(\kappa_2 a)}{\eta_2} \\ \frac{zmJ_m(\kappa_1 a)}{a\kappa_1^2} & -\frac{\omega\mu_1 J_m'(\kappa_1 a)}{\kappa_1\eta_1} & -\frac{zmJ_m(\kappa_2 a)}{a\kappa_2^2} & -\frac{zmY_m(\kappa_2 a)}{a\kappa_2^2} & \frac{\omega\mu_2 J_m'(\kappa_2 a)}{\kappa_2\eta_2} & \frac{\omega\mu_2 Y_m'(\kappa_2 a)}{\kappa_2\eta_2} \\ -\frac{\omega\varepsilon_1 J_m'(\kappa_1 a)}{\kappa_1} & -\frac{zmJ_m(\kappa_1 a)}{a\kappa_1^2\eta_1} & \frac{\omega\varepsilon_2 J_m'(\kappa_2 a)}{\kappa_2} & \frac{\omega\varepsilon_2 Y_m'(\kappa_2 a)}{\kappa_2} & \frac{zmJ_m(\kappa_2 a)}{a\kappa_2^2\eta_2} & \frac{zmY_m(\kappa_2 a)}{a\kappa_2^2\eta_2} \\ 0 & 0 & J_m(\kappa_2 b) & Y_m(\kappa_2 b) & 0 & 0 \\ 0 & 0 & \frac{zmJ_m(\kappa_2 b)}{b\kappa_b^2} & \frac{zmY_m(\kappa_2 b)}{b\kappa_b^2} & -\frac{\omega\mu_2 J_m'(\kappa_2 b)}{\kappa_2\eta_2} & -\frac{\omega\mu_2 Y_m'(\kappa_2 b)}{\kappa_2\eta_2} \end{vmatrix}, \quad (8)$$

where z represents a propagation constant, and $J_m(\cdot)$ and $Y_m(\cdot)$ are the first and second kind of Bessel functions, respectively. Further, the coefficients κ_1 and κ_2 are defined as follows $\kappa_1=\sqrt{z^2+\omega^2\varepsilon_r/c^2}$, $\kappa_2=\sqrt{z^2+\omega^2/c^2}$, where $\omega=2\pi f$, $\eta_1=120\pi/\sqrt{\varepsilon_r}[\Omega]$ and $\eta_2=120\pi\,[\Omega]$.

The numerical tests are performed for $a=6.35$[mm], $b=10$[mm], $\varepsilon_r=10$, $m=1$ and $f=5$[GHz]. The region considered $\Omega=\{z\in\mathbb{C}:-2<\Re(z)<2\wedge -2<\Im(z)<2\}$ and Δr is assigned a value of 0.1.

Figure 6 shows the final mesh of the preliminary estimation process. It requires 1376 evaluations of the function over 11 iterations. The results of the preliminary estimation and the verification are reported in Table VI and the improved values of the roots in Table VII.

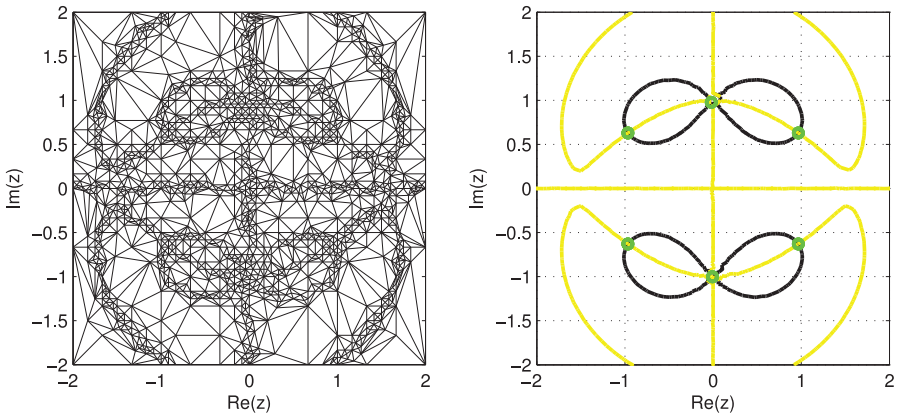

Fig. 6. The final triangulation (left-hand side) and the real-zero and imaginary-zero curves (right-hand side) for the complex wave propagation problem.

Table VI. The Results of the Preliminary Estimation and Verification for the Complex Wave Propagation Problem

m	s_m	q	P	Verification
1	$0.97 + 0.63i$	1	9	root
2	$0.00 + 1.02i$	-2	18	singularity
3	$-0.97 + 0.63i$	1	9	root
4	$-0.97 - 0.63i$	1	9	root
5	$0.00 - 1.00i$	-2	18	singularity
6	$0.97 - 0.63i$	1	9	root

Table VII. The Final Results for the Complex Wave Propagation Problem

m	$\overline{s_m}$	$\lvert f(\overline{s_m})\rvert$	$\lvert f(\overline{s_m} + \delta_m)\rvert$	K
1	$0.966423024599417 + 0.629233974556967i$	$2.35 \cdot 10^{-22}$	$4.16 \cdot 10^{-17}$	9
3	$-0.966423024599417 + 0.629233974556967i$	$2.35 \cdot 10^{-22}$	$4.16 \cdot 10^{-17}$	9
4	$-0.966423024599417 - 0.629233974556967i$	$2.35 \cdot 10^{-22}$	$4.16 \cdot 10^{-17}$	9
6	$0.966423024599417 - 0.629233974556967i$	$2.35 \cdot 10^{-22}$	$4.16 \cdot 10^{-17}$	9

Note: K is a number of iterations in the final refinement and $\delta_m = \overline{s_m} \cdot 10^{-10}$.

3.4. Efficiency of Delaunay Triangulation

Since the effectiveness of the presented algorithm has been shown, a few comments about its efficiency should be added. As previously mentioned, the efficiency depends on the considered function. If the evaluation time of the function at a single point is short, then iterative retriangulation can be inefficient (since the time involved in retriangulation may be longer that the time of evaluation the function at extra points on a dense mesh). However, for more complex functions (especially these obtained from numerical analysis), the retriangulation time can be negligibly short in comparison to the function evaluation. In Table VIII, the CPU time of the whole process and the preliminary estimation for each tests considered here is compared with a case of a hexagonal mesh (a pitch $\Delta r = 0.1$). In the latter case, no retriangulation is necessary. It can be seen that for all presented examples, the number of points at which the function is evaluated is siginficantly smaller. Since the preliminary estimation is a major part of the whole process, it can be expected that when the function evaluation

Table VIII. Computation Time of Total Process and Preliminary Estimation for Two Variants: Based on Delaunay Triangulation and on Regular Dense Mesh (Intel(R) Core(TM) i3 3.07GHz)

	Iterative Delaunay Triangulation		Regular Mesh	
Example	Preliminary Estimation	Total Process	Preliminary Estimation	Total Process
Surface Waves	0.32s (693 points)	0.91s	1.10s (2040 points)	1.69s
Leaky Waves	1.17s (1107 points)	2.01s	2.27s (2040 points)	3.11s
Complex Waves	2.46s (1376 points)	3.48s	3.87s (2040 points)	4.89s

is very time consuming, the total duration of the calculation can be also significantly reduced.

4. CONCLUSIONS

In this article, a simple (global) complex root finding algorithm is presented. The algorithm can be applied to a very wide class of functions and for arbitrary-shaped regions. The results are verified and its accuracy can be controlled. The examples presented show that the algorithm can be successfully applied to functions containing singularities and branch cuts, which is the case in many real-world technical problems. In order to illustrate the efficiency of the scheme presented, a number of function evaluations is given for each example.

The possibility of parallelizing the algorithm enables its efficiency to be significantly improved for difficult and complex functions, especially those estimated numerically.

APPENDIX

According to the assumption that Δr is the minimum distance between the roots, the points in set S cannot be located any closer. However, for higher orders of the root or singularities (or at a branch cut), the preliminary estimation process can generate several candidate points a very short distance apart. This is a result of multiple crossing of the curves C_R and C_I. In such cases, it is convenient to eliminate some redundant points from S. To cope with this problem, the following routine can be applied.

(1) For each point s_m, a weight $w_m = 1$ is assigned.
(2) For each pair of points s_i and s_j, a distance $d_{i,j} = |s_i - s_j|$ is calculated.
(3) If the smallest distance $d_{i,j}$ is greater than Δr the algorithm stops. Otherwise, both points are replaced by $s_k = \frac{s_i w_i + s_j w_j}{w_i + w_j}$ $(w_k = w_i + w_j)$ and the algorithm returns to Step (2).

ACKNOWLEDGMENTS

The author would like to thank Dr. W. Marynowski for numerous numerical tests and Prof. M. Mrozowski for helpful suggestions and anonymous reviewer for valuable comments.

REFERENCES

Milton Abramowitz and Irene A. Stegun. 1972. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover Publications, New York, NY.

George B. Dantzig. 1963. *Linear Programming and Extensions*. Princeton University Press, Princeton, N.J.

Ajoy Ghatak and K. Thyagarajan. 1998. *An Introduction to Fiber Optics*. Cambridge University Press, New York.

Kenneth S. Gritton, J. D. Seader, and Wen-Jing Lin. 2001. Global homotopy continuation procedures for seeking all roots of a nonlinear equation. *Comput. Chem. Eng.* 25, 7–8, 1003–1019. DOI:http://dx.doi.org/10.1016/S0098-1354(01)00675-5

Piotr Kowalczyk and Michal Mrozowski. 2007. A new conformal radiation boundary condition for high accuracy finite difference analysis of open waveguides. *Opt. Express* 15, 20, 12605–12618. DOI:http://dx.doi.org/10.1364/OE.15.012605

P. Lamparillo and R. Sorrentino. 1975. The ZEPLS program for solving characteristic equations of electromagnetic structures (Computer Program Descriptions). *IEEE Trans. Microwave Theory Tech.* 23, 5, 457–458.

Yunliang Long and Hongyan Jiang. 1997. Numerical analysis of surface wave on microstrip antenna with lossy substrate. *Electron. Lett.* 33, 19, 1592–1593. DOI:http://dx.doi.org/10.1049/el:19971103

Yunliang Long and Hongyan Jiang. 1998. Rigorous numerical solution to complex transcendental equations. *Int. J. Infrared Millimeter Waves* 19, 5, 785–790.

Jerzy J. Michalski and Piotr Kowalczyk. 2011. Efficient and systematic solution of real and complex eigenvalue problems employing simplex chain vertices searching procedure. *IEEE Trans. Microwave Theory Tech.* 59, 9, 2197–2205. DOI:http://dx.doi.org/10.1109/TMTT.2011.2160277

Gary L. Miller, Todd Phillips, and Donald R. Sheehy. 2011. Beating the spread: Time-optimal point meshing. In *Proceedings of the 27th Annual Symposium on Computational Geometry (SoCG'11)*. ACM, New York, 321–330. DOI:http://dx.doi.org/10.1145/1998196.1998252

Michal Mrozowski. 1997. *Guided Electromagnetic Waves: Properties and Analysis*. Electronic and Electrical Engineering Research Studies Computer Methods in Electromagnetics, Vol. 3, Research Studies Press, New York, NY.

Michal Mrozowski and Jerzy Mazur. 1992. Matrix theory approach to complex waves [in shielded lossless guides]. *IEEE Trans. Microwave Theory Tech.* 40, 4, 781–785.

James R. Pinkert. 1976. An exact method for finding the roots of a complex polynomial. *ACM Trans. Math. Softw.* 2, 4.

William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. 1992. *Numerical Recipes in Fortran 77: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK.

Arnold Schonhage. 1982. The fundamental theorem of algebra in terms of computational complexity. Tech. Rep. Mathematisches Institut der Universitat Tubingen.

G. W. Stewart. 1994. On the convergence of multipoint iterations. *Numer. Math.* 68, 1, 143–147.

L. Wan. 2011. A new method to find full complex roots of a complex dispersion equation for light propagation. ArXiv e-prints (Aug. 2011).

Changying Wu, Jianzhou Li, Gao Wei, and Jiadong Xu. 2010. A novel method to solve the complex transcendental equation for the permittivity determination in short-circuited line. In *PIERS Proceedings*.

Tian Yu-Bo. 2009. Solving complex transcendental equations based on swarm intelligence. *IEEJ Trans. Elect. Electron. Eng.* 4, 6, 755–762. DOI:http://dx.doi.org/10.1002/tee.20477

W. Zieniutycz. 1983. Comments on: The ZEPLS program for solving characteristic equations of electromagnetic structures. *IEEE Trans. Microwave Theory Tech.* 31, 5, 420.

Received June 2013; revised January 2014 and August 2014; accepted September 2014