Journal of Computational and Applied Mathematics 314 (2017) 1–9

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

On root finding algorithms for complex functions with branch cuts

Piotr Kowalczyk

Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Narutowicza 11/12, 80233 Gdansk, Poland

ARTICLE INFO

Article history:
Received 10 June 2016
Received in revised form 17 October 2016

Keywords:
Complex root finding
Branch cuts
Branch points

ABSTRACT

A simple and versatile method is presented, which enhances the complex root finding process by eliminating branch cuts and branch points in the analyzed domain. For any complex function defined by a finite number of Riemann sheets, a pointwise product of all the surfaces can be obtained. Such single-valued function is free of discontinuity caused by branch cuts and branch points. The roots of the new function are the same as the roots of original multi-valued variety, while the verification of them is much easier. Such approach can significantly improve the efficiency (as well as the effectiveness) of the root finding algorithms. The validity of the presented technique is supported by the results obtained from numerical tests.

1. Introduction

A number of practical issues may be formulated in terms of nonlinear equations defined in a complex domain. Essentially, the problem is reduced to finding roots of a complex function. Numerous functions of this kind can be rather complicated and multi-valued; for instance, in optics, microwave engineering and acoustics (especially the guiding or scattering of waves). Unfortunately, root finding algorithms are very sensitive to branch cuts/points, such that the procedures become inefficient or even ineffective.

In general, the root-finding algorithms can be divided into three main groups. The first one concerns standard local schemes, which improve the accuracy of the root (if its initial value is roughly known), such as Newton's [1], Muller's [2] or the simplex [3] methods. In the neighborhood of a branch cut (due to discontinuity), the iteration process may not converge; worse still, it may converge to a point that in fact is not a root at all. In Table 1, the first five iterations of the Muller method for the function $f(z) = \sin(\sqrt{z^2+1}) - z$ are presented. The process does not converge in terms of the root located at the branch cut in $z_0 = 1.599978334033766i$, but it systematically moves away from the solution.

The second group includes algorithms that track the root as a function with an extra parameter [4,5]. Such an approach can be very efficient. However, it is based on following the sign changes of the function, so it is also limited to regions without branch cuts.

The last group concerns global algorithms, which means that, when using these techniques all roots inside a fixed region should be found. For simple polynomial functions, algorithms, which are based on the Sturm sequence method and enhanced by the Routh theorem [6], or are based on the splitting circle method introduced by Schonhage [7], can be applied with a very high degree of efficiency. The generalization of these procedures is proposed in [8,9], but the function still needs to be free of singularities and branch cuts in the analyzed region. The same limitation applies to the mesh methods [10].

E-mail addresses: pio.kow@gmail.com, piotr.kowalczyk@pg.gda.pl.

http://dx.doi.org/10.1016/j.cam.2016.10.015
0377-0427/

Table 1
First five iterations of the Muller method with the following initial points: $z_{-2} = 0.1 + 1.6i$, $z_{-1} = -0.1 + 1.7i$ and $z_0 = -0.1 + 1.5i$.

Iteration	z_k	$\|f(z_k)\|$
1	$-0.290121154355435 + 1.543243599391047i$	3.095872687013406
2	$0.264934875828936 + 0.411462293942946i$	0.655698045459433
3	$0.509909361321198 + 0.556376715035444i$	0.551564593660939
4	$0.994392302844264 + 0.835059692040596i$	0.677079121380443
5	$0.878943230435087 + 0.047238667494617i$	0.101068126040518

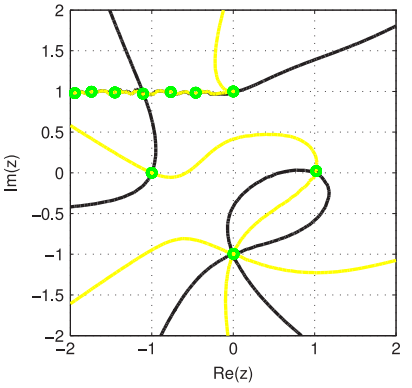

Fig. 1. The curves C_R (black) and C_I (yellow) are obtained in the preliminary estimation process for $f(z) = (z-i)^{1/2}(z+1)(z+i)^2(z-1)^{-1}$. The set S contains 10 candidate points (green circles). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

A new global technique, based on an approximation of the function on a triangular mesh, has recently been proposed in [11]. The technique consists of three main stages: preliminary estimation, verification and final refinement. In the first step, the real and the imaginary parts of the function are linearly approximated on a regular (or self-adoptive) mesh of size Δr. Then two curves representing the zeros of its real and imaginary parts can be obtained (see Fig. 1). In turn, the "candidate points" (denoted by set S) are evaluated as intersections of these two curves.

In the second stage all the candidates must be verified, which means checking if the candidate is a root, a singularity or a regular point. The verification of a complex root candidate can be quite complicated (signs, which change in the real and imaginary parts of the function in a fixed region, do not guarantee a root in this area). In order to verify the validity of the root, Cauchy's argument principle can be applied. 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 an integer number and represents a sum of all zeros counted with their multiplicities, minus the sum of all poles counted with their multiplicities. For a multi-valued function, the integration must involve all Riemann surfaces to close the curve. For example, values of parameter q obtained for function $f(z) = (z-i)^{1/2}(z+1)(z+i)^2(z-1)^{-1}$ are as follows (see Fig. 1): $q = -1$, if C is a circle of radius $\Delta r = 0.1$ and the center at $z_C = 1$; $q = 1$, if C is a circle of radius $\Delta r = 0.1$ and the center at $z_C = -1$; $q = 2$, if C is a circle of radius $\Delta r = 0.1$ and the center at $z_C = -i$; and $q = 0.5$ if C is a circle of radius $\Delta r = 0.1$ and the center at $z_C = i$. To obtain an integer value of q in the latter case, both Riemann sheets must be analyzed and the results of the integrations over the contour C on both sheets must be summed up (to close the contour). For 6 other candidates (located at the cut) parameters q are zero, which means that the function has two roots and one singularity in the considered region (it can be confirmed by integration over the boundary of the whole region).

In the last stage the verified candidates become starting points for local iteration process based on a rational function approximation (in regions $|z - z_C| < \Delta r$). In this stage any local technique can be applied or it can be skipped if the size Δr is sufficiently small.

The described algorithm is versatile and can be very useful in practical applications. However, it is based on sign changes, such that it is still inefficient at the branch cuts. Along the cut, a large number of candidate points is generated. Finally, although most of these candidate points will be rejected, all Riemann sheets of the function must be analyzed (which involves a rather complex implementation and a very time-consuming process) before the decision of discarding the candidate is made. Summarizing, there is no general algorithm which is effective and efficient for complex functions with branch cuts.

Obviously there are some possibilities to avoid the problem with branch points/cuts. A localization of the cut can be changed by redefining the function (e.g., in Matlab environment for the square root function, the cut is located on the negative real axis, but it can be simply modified to any half-line with the initial point at the origin). However, the branch

point cannot be changed, while such an operation is useless for roots located close to this point. Moreover, a reformulation of the problem can be ineffective. For instance, in many issues a square root becomes an argument of a trigonometric or a special function. An inversion of such an expression, if ever it is possible, leads to the worst conditioned problem (inverse trigonometric functions have an infinite number of Riemann sheets).

The above considerations lead to the conclusion that efficient root-finding processes cannot be performed separately on a single Riemann sheet. In this article, a simple method is presented that allows for eliminating branch cuts and branch points in the analyzed domain. For any complex function defined by a finite number of Riemann sheets, a pointwise product of all the surfaces can be obtained. Such single-valued function is free of discontinuity caused by branch cuts and branch points. The roots of the new function are the same as the roots of multi-valued original one. Also, the verification of the roots for such function is much easier: only integer values of parameter q can be obtained from a single integration over the contour C. This also holds for the global verification of the whole considered region (it is necessary to verify the total number of roots and singularities).

The idea presented in this paper can be applied to any standard root-finding algorithm (local, global or tracing). The validity of the presented technique is supported by the results obtained from numerical tests.

2. Pointwise product of Riemann sheets

Before presenting the benefits of the mentioned technique the following theorem should be introduced.

Theorem. *Let us assume that $S_n(z)$ ($n \in \mathcal{N} = \{1, \ldots, N\}$) are all Riemann sheets of the continuous multi-valued function $f(z)$ defined on $\Omega \subset \mathbb{C}$. Let us also assume that L ($L \subset \Omega$) is a set of all points located on the branch cuts of the function. A single-valued function $F(z)$ can be defined in Ω as a pointwise product of all Riemann sheets*

$$F(z) = \prod_{n=1}^{N} S_n(z). \tag{2}$$

Then:

(1) *function $F(z)$ is continuous in Ω (including branch cut L),*
(2) *all roots of each $S_n(z)$ are roots of $F(z)$ and, vice versa, each root of $F(z)$ is a root of one of $S_n(z)$.*

Proof. (1) If $S_n(z)$ are continuous in $\Omega \setminus L$, then their product is also continuous in this region. The proof, therefore, can be focused only on the cuts and stems directly from the mechanism of the formation of Riemann sheets:

$$\forall n \in \mathcal{N}, \exists m \in \mathcal{N}, \forall z_L \in L \lim_{z \to z_L} S_n(z) = S_m(z_L)$$

$$\Rightarrow \forall z_L \in L \lim_{z \to z_L} \prod_{n=1}^{N} S_n(z) = \prod_{n=1}^{N} S_n(z_L). \tag{3}$$

(2) This proof is trivial since $F(z)$ is a pointwise product of all Riemann sheets.

In practice, the above theorem means that $F(z)$ is free of all discontinuities caused by branch points and branch cuts and can be analyzed instead of all Riemann sheets separately. For instance, let us consider the following function

$$f(z) = 1 + z \sin\left(\sqrt{z^2 - 1}\right). \tag{4}$$

Such a function can be represented by two Riemann sheets. For example, in Matlab environment (the square root is defined with the cut at the negative real axis) two Riemann sheets are

$$S_1(z) = 1 + z \sin\left(\sqrt[\star,2]{z^2 - 1}\right) \tag{5}$$

and

$$S_2(z) = 1 + z \sin\left(-\sqrt[\star,2]{z^2 - 1}\right) \tag{6}$$

where $\sqrt[\star,2]{z} = e^{\frac{1}{2}\mathrm{Log}z}$, $\mathrm{Log}z = \log|z| + i\mathrm{Arg}z$ and $(-\pi < \mathrm{Arg}z \leq \pi)$ represents a standard Matlab function "*sqrt(z)*". In Fig. 2, all the cuts are shown (imaginary axis and partially real axis). In this case, the pointwise product function $F(z)$ consists of only two terms: $F(z) = S_1(z)S_2(z)$. The result is presented in Fig. 3, which shows that there are no cuts in the function $F(z)$.

It should be noted that, in some special cases, the assumptions of the theorem can be alleviated. Not all Riemann sheets must be involved in order to achieve a continuous function in a fixed region: the condition in the theorem is sufficient, but not necessary. In general, all the sheets that are continuous in Ω can be neglected in $F(z)$ and analyzed separately. Such a "reduction" can be also very useful for algorithms whose efficiency is sensitive to higher orders of the root (the artificial multiplication in the pointwise product can unnecessary increase the root order if it is located on more than one Riemann sheet). In practice, this reduction can be difficult and will be efficient only in very special cases (e.g., the time-consuming evaluation of the function or a high number of sheets).

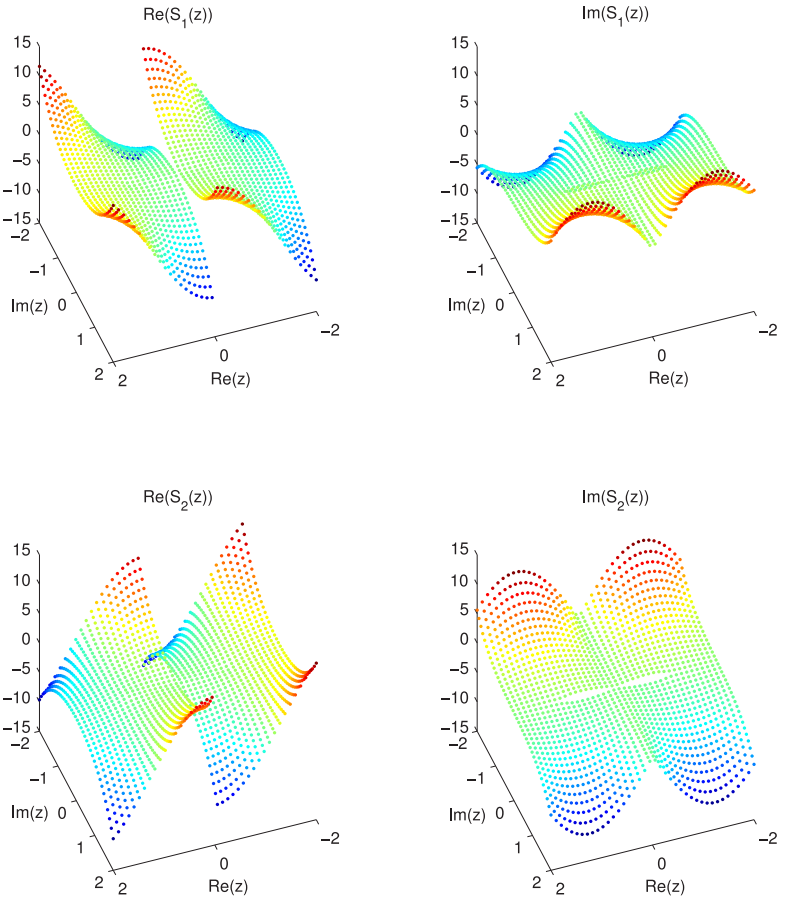

Fig. 2. Two Riemann sheets (5) and (6) of function (4).

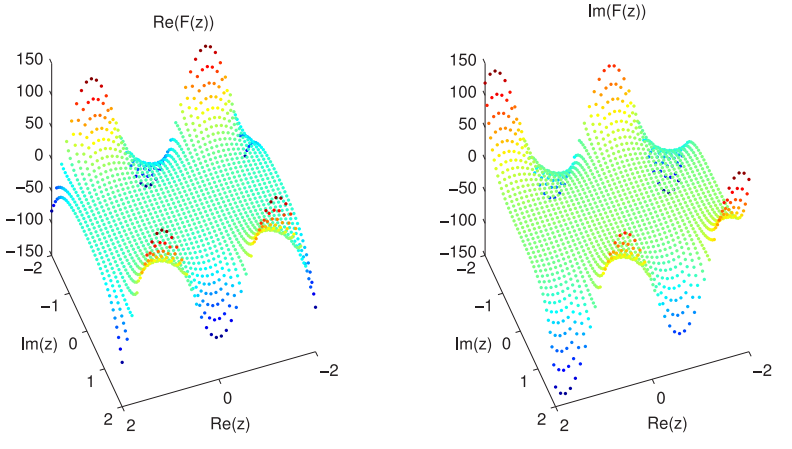

Fig. 3. The pointwise product function $F(z)$ obtained from Riemann sheets (5) and (6).

3. Numerical tests

In order to show the advantages of the presented idea, Muller's method was once again applied to find a complex root of the function $f(z) = \sin(\sqrt{z^2+1}) - z$ with the same initial points. This time, however, the pointwise product function was utilized (both Riemann sheets were simultaneously considered): $F(z) = \left(\sin(\overset{\star,2}{\sqrt{z^2+1}}) - z\right)\left(-\sin(\overset{\star,2}{\sqrt{z^2+1}}) - z\right)$. The results of the first five iterations are displayed in Table 2 (compare with Table 1).

Table 2
First five iterations of the Muller method for pointwise product function.

Iteration	z_k	$\lvert f(z_k)\rvert$
1	$0.002650832933975 + 1.600125211264367i$	0.012041399549327
2	$0.000068049906734 + 1.599989025354078i$	0.000312358654393
3	$-0.000000016348834 + 1.599978305787640i$	0.000000147987263
4	$-0.000000000000002 + 1.599978334033757i$	0.000000000000042
5	$0.000000000000000 + 1.599978334033765i$	0.000000000000002

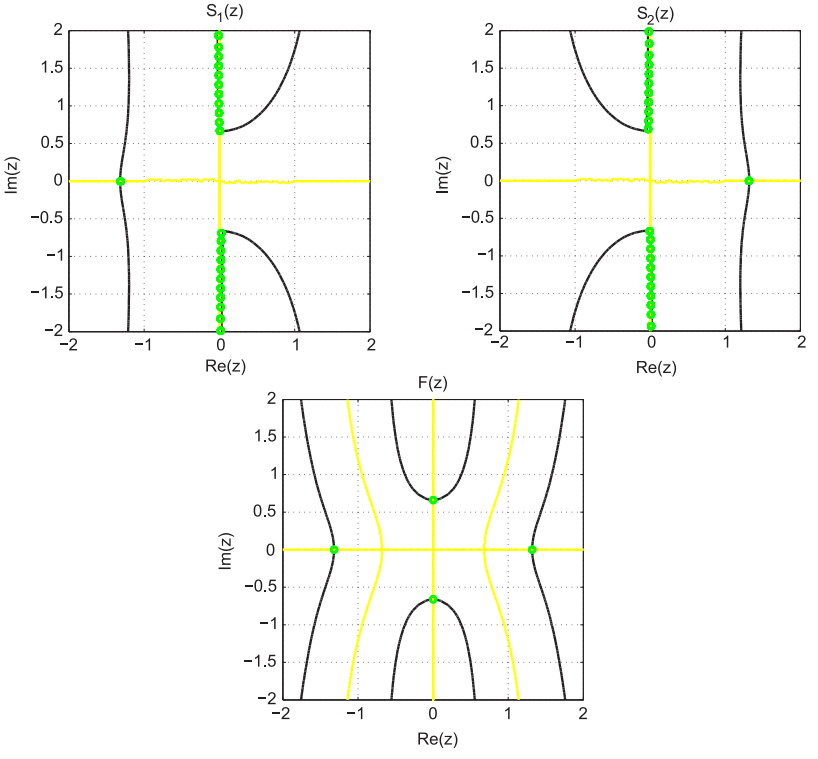

Fig. 4. The results obtained for two Riemann sheets (5) and (6), and the pointwise product function. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To test the proposed technique for more complex problems, the algorithm described in [11] was utilized due to its versatility and flexibility.

Firstly, the example (4), which was introduced in the previous section, was examined. The function was analyzed in the region $\Omega = \{z \in \mathbb{C} : -2 < \Re(z) < 2 \wedge -2 < \Im(z) < 2\}$ with the mesh size $\Delta r = 0.1$. In the first step, two Riemann sheets were analyzed separately (two separate execution of the algorithm [11] for both Riemann sheets), then the idea behind the pointwise product function was utilized. In Fig. 4, two curves represent zeros of real (black) and imaginary (yellow) parts of the function. All candidate points were evaluated based on intersections of these curves (green circles).

For the first Riemann sheet, 23 candidate points were found. All of them were verified with Cauchy's argument principle (1). For point $z_1 = -1.3192$, parameter $q_1 = 1$ (with the machine precision $\varepsilon = 2.220446049250313 \cdot 10^{-16}$) and z_1 is a single root of (4). For 22 candidates, as the parameters were not integer, the second Riemann sheet should be analyzed to specify their types.

For the second Riemann sheet, the results were similar, i.e., 23 candidate points. Only one single root at $z_2 = 1.3192$ could be verified directly from the verification of the second sheet. For the other 22 candidates, the q parameters were not integers. Summing up the values of parameters q obtained for both the sheets (for each candidate) two extra roots were found: i.e., $z_3 = 0.6627i$ and $z_4 = -0.6627i$.

The pointwise product function was than analyzed. In this case, only four candidate points were found and verified as a single roots. The process was more than 25 times faster than the separate analysis of both sheets. It is the result of singularly performing of the routine [11] and with a significantly smaller number of candidates (estimation of candidates and its verification is time-consuming), while there is no need to combine the partial results. The comparison of efficiency of these two approaches is presented in Table 3 (computations performed using Intel(R) Core(TM)i7-2600K CPU @ 3.40 GHz, RAM 16 GB).

Table 3
The comparison of efficiency between the regular algorithm and the pointwise product approach for the first example.

	Single Riemann sheets			Pointwise product
	$S_1(z)$	$S_2(z)$	Total	
Number of function evaluations for preliminary estimation	917	922	1839	1548
Number of function evaluations for verification	2417	2415	4832	80
Number of candidate points	23	23	46	4
Number of verified roots	1	1	4[a]	4
Computational time (s)	7.5	7.6	15.1	0.6

[a] Two of these roots cannot be verified from a single Riemann sheet.

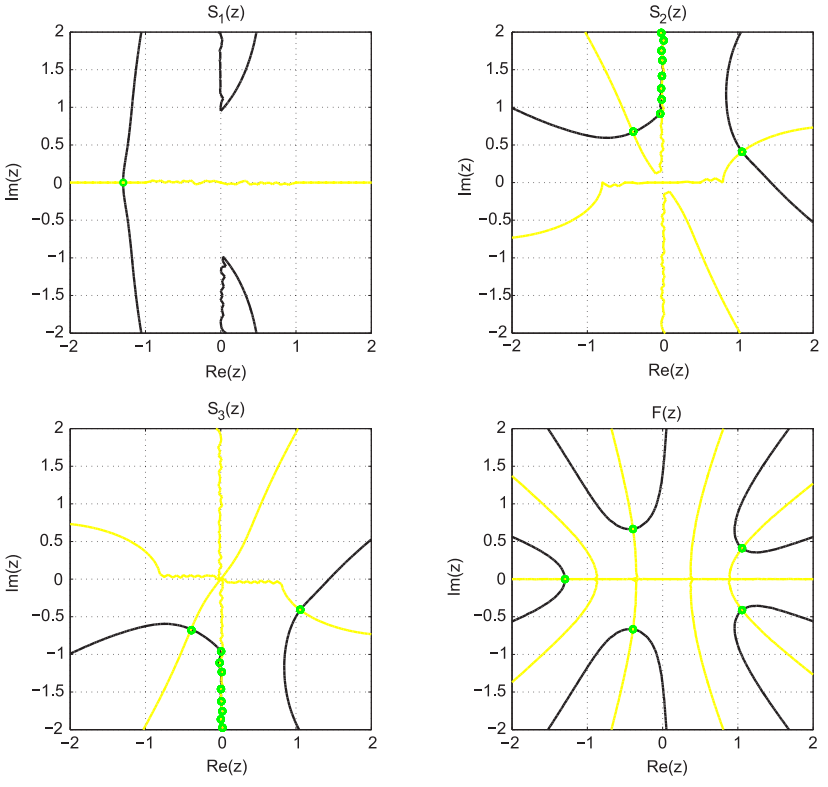

Fig. 5. The results respectively obtained for three Riemann sheets (8)–(10), and the pointwise product function.

In the second example, a cubic root function was involved in order to show the ability of the method for higher order roots

$$f(z) = 1 + z \sin\left(\sqrt[3]{z^2 - 1}\right). \tag{7}$$

In this case, three Riemann sheets must be considered

$$S_1(z) = 1 + z \sin\left(\overset{\star,3}{\sqrt{}}{z^2 - 1}\right), \tag{8}$$

$$S_2(z) = 1 + z \sin\left(\overset{\star,3}{\sqrt{}}{z^2 - 1}\, e^{2\pi i/3}\right) \tag{9}$$

and

$$S_3(z) = 1 + z \sin\left(\overset{\star,3}{\sqrt{}}{z^2 - 1}\, e^{4\pi i/3}\right) \tag{10}$$

where $\overset{\star,3}{\sqrt{}}{z} = e^{\frac{1}{3}\mathrm{Log}z}$ represents a cubic root function in Matlab environment (cut at the negative real axis).

Similar to the previous example, the function was analyzed in the region $\Omega = \{z \in \mathbb{C} : -2 < \Re(z) < 2 \wedge -2 < \Im(z) < 2\}$ with the mesh size $\Delta r = 0.1$. The results are presented in Fig. 5.

For the first Riemann sheet, only one candidate point was found and verified as a single root, namely, $z_1 = -1.2971$. From the second sheet, 10 candidates were assigned and only two of these points were directly confirmed as single roots

Table 4
The comparison of efficiency between the regular algorithm and the pointwise product approach for the second example.

	Single Riemann sheets				Pointwise product
	$S_1(z)$	$S_2(z)$	$S_3(z)$	Total	
Number of function evaluations for preliminary estimation	707	835	880	2422	1534
Number of function evaluations for verification	20	948	948	1916	100
Number of candidate points	1	10	10	21	5
Number of verified roots	1	2	2	5	5
Computational time (s)	2.0	5.3	6.2	13.5	0.6

($z_2 = 1.0565 + 0.4109i$ and $z_3 = -0.3940 + 0.6663i$). For the third sheet, 10 candidate points were also found, while two of them were verified as single roots ($z_4 = 1.0565 + 0.4109i$ and $z_5 = -0.3940 + 0.6663i$). Summing up the partial results of the verification from three Riemann sheets, the remaining candidates were eliminated.

The same solution was obtained from analysis of the pointwise product function, but about 22 times faster: see Table 4. As in the previous example, the algorithm [11] was applied only once, with no partial results and no redundant candidates to verify.

Finally, a real-world technical problem was solved: in other words, guiding of electromagnetic waves in dielectric slabs is one of the fundamental issues of optics and microwave engineering. Let us consider an electromagnetic wave guided in an air gap ($\varepsilon_{r1} = 1$) between two dielectric slabs of relative permittivity ($\varepsilon_{r2} = 9$) [12]. The thickness of the gap was $b = 2$ nm, whereas the frequency corresponded to wavelength $\lambda_0 = 1.5$ nm in a vacuum. The effective refractive index (represented here by z) can be found from the conditions fulfilled by electromagnetic fields at the boundary. For even *TE* modes in this structure the following determinant must be zero (for more layers, the determinant size would be bigger):

$$f(z) = \begin{vmatrix} \sin\left(\frac{b\pi}{\lambda_0}\sqrt{\varepsilon_{r1} - z^2}\right) & 1 \\ \sqrt{\varepsilon_{r1} - z^2}\cos\left(\frac{b\pi}{\lambda_0}\sqrt{\varepsilon_{r1} - z^2}\right) & \sqrt{z^2 - \varepsilon_{r2}} \end{vmatrix}. \tag{11}$$

For such a function, involving square roots in four places many different Riemann sheets should be introduced (exactly $16 = 2^4$). However, due to a special form of the function, most of them can be eliminated (cosine is an even function, while signs in rows or columns can be factored out of the determinant). Eventually, two Riemann sheets may be taken into account (the rest differs from those only involving the global sign)

$$S_1(z) = \begin{vmatrix} \sin\left(\frac{b\pi}{\lambda_0}\ \overset{\star,2}{\sqrt{\varepsilon_{r1} - z^2}}\right) & 1 \\ \overset{\star,2}{\sqrt{\varepsilon_{r1} - z^2}}\cos\left(\frac{b\pi}{\lambda_0}\ \overset{\star,2}{\sqrt{\varepsilon_{r1} - z^2}}\right) & \overset{\star,2}{\sqrt{z^2 - \varepsilon_{r2}}} \end{vmatrix} \tag{12}$$

and

$$S_2(z) = \begin{vmatrix} \sin\left(-\frac{b\pi}{\lambda_0}\ \overset{\star,2}{\sqrt{\varepsilon_{r1} - z^2}}\right) & 1 \\ -\overset{\star,2}{\sqrt{\varepsilon_{r1} - z^2}}\cos\left(-\frac{b\pi}{\lambda_0}\ \overset{\star,2}{\sqrt{\varepsilon_{r1} - z^2}}\right) & -\overset{\star,2}{\sqrt{z^2 - \varepsilon_{r2}}} \end{vmatrix}. \tag{13}$$

As in the two previous examples, the function was analyzed in the region $\Omega = \{z \in \mathbb{C} : -2 < \Re(z) < 2 \wedge -2 < \Im(z) < 2\}$ with the mesh size $\Delta r = 0.1$. The results are shown in Fig. 6.

From the first Riemann sheet, 36 candidate points were found and only four of them could be directly verified as a single root ($z_1 = 0.3194 + 1.0907i$, $z_2 = 0.3194 - 1.0907i$, $z_3 = -0.3194 + 1.0907i$ and $z_4 = -0.3194 - 1.0907i$). For the second sheet, 26 candidates were obtained, of which none could be confirmed. Summing up the partial results of the verification from two Riemann sheets two extra points were verified as single roots ($z_5 = 1$ and $z_6 = -1$).

The same six points were obtained for the pointwise product function, but almost 100 times faster: see Table 5. As in the previous examples, the algorithm [11] was applied only once, such that there were no partial results and no redundant candidates.

4. Conclusions

In this paper, a simple technique of elimination of branch cuts and branch points in a complex function is presented. The method can be applied to any standard technique of root-finding, including local, global and tracing algorithms. Such approach can significantly improve efficiency (or even effectiveness) of root-finding algorithms. The idea presented in this article allows for the algorithm to be performed only once for a multi-valued function (instead of a separate analysis of N

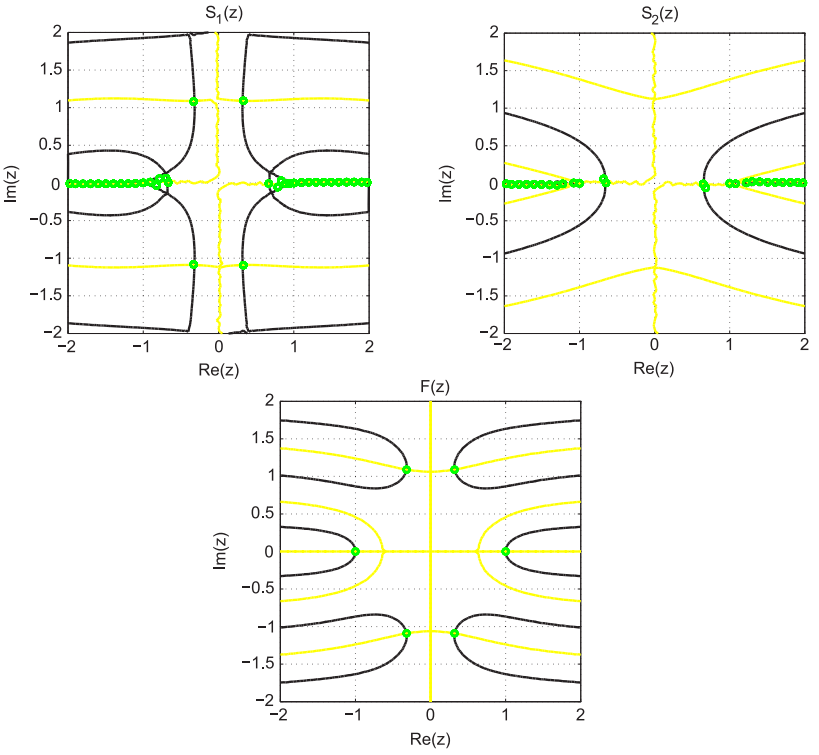

Fig. 6. The results obtained for two Riemann sheets between (12) and (13), and the pointwise product function, respectively.

Table 5
The comparison of efficiency between the regular algorithm and the pointwise product approach for the third example.

	Single Riemann sheets			Pointwise product
	$S_1(z)$	$S_2(z)$	Total	
Number of function evaluations for preliminary estimation	1578	1145	2723	1929
Number of function evaluations for verification	2900	2132	5032	240
Number of candidate points	36	26	62	6
Number of verified roots	4	0	6[a]	6
Computational time (s)	49.3	20.3	69.6	0.7

[a] Two of these roots cannot be verified from a single Riemann sheet.

Riemann sheets). Moreover, the number of candidate points is significantly smaller and the verification is much simpler, which essentially reduces the computational time (up to two orders of magnitude) and the complexity of the algorithm. The examples show that the technique can be successfully applied to functions (containing branch cuts or branch points), which are common in many real-world technical problems.

Acknowledgment

This work was supported under funding for Statutory Activities for the Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology.

References

[1] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, New York, NY, USA, 1972.
[2] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes in Fortran 77:The Art of Scientific Computing, Cambridge University Press, England, 1992.
[3] G.B. Dantzig, Linear Programming and Extensions, Princeton University Press, Princeton, 1963.
[4] K.S. Gritton, J. Seader, W.-J. Lin, Global homotopy continuation procedures for seeking all roots of a nonlinear equation, Comput. Chem. Eng. 25 (7–8) (2001) 1003–1019. http://dx.doi.org/10.1016/S0098-1354(01)00675-5.
[5] J.J. Michalski, P. Kowalczyk, Efficient and systematic solution of real and complex eigenvalue problems employing simplex chain vertices searching procedure, IEEE Trans. Microw. Theory Tech. 59 (9) (2011) 2197–2205. http://dx.doi.org/10.1109/TMTT.2011.2160277.
[6] J.R. Pinkert, An exact method for finding the roots of a complex polynomial, ACM Trans. Math. Softw. 2 (4) (1976) 351–363. http://dx.doi.org/10.1145/355705.355710.

[7] A. Schonhage, The fundamental theorem of algebra in terms of computational complexity, Technical Report, Mathematisches Institut der Universitat, Tubingen, 1982.
[8] Y. Long, H. Jiang, Rigorous numerical solution to complex transcendental equations, Int. J. Infrared Millim. Waves 19 (5) (1998) 785–790.
[9] C. Wu, J. Li, G. Wei, J. Xu, A novel method to solve the complex transcendental equation for the permittivity determination in short-circuited line, in: PIERS Proceedings, Xi'an, China, 2010, pp. 1764–1767.
[10] L. Wan, A new method to find full complex roots of a complex dispersion equation for light propagation, ArXiv e-prints arXiv:1109.0879.
[11] P. Kowalczyk, Complex root finding algorithm based on delaunay triangulation, ACM Trans. Math. Softw. 41 (3) (2015). http://dx.doi.org/10.1145/2699457, Article 19, 13 pages.
[12] P. Kowalczyk, M. Mrozowski, A new conformal radiation boundary condition for high accuracy finitedifference analysis of open waveguides, Opt. Express 15 (20) (2007) 12605–12618. http://dx.doi.org/10.1364/OE.15.012605.