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	$\lvert f(z_k)\rvert$
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

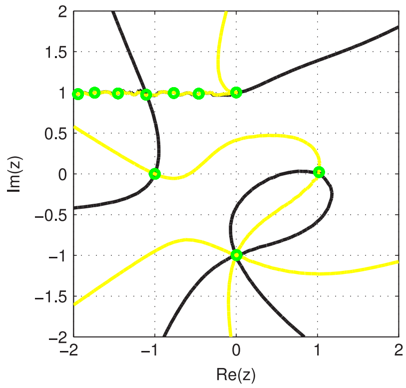

Fig. 1. The curves C_R (black) and C_I (yellow) are obtained in the preliminary estimation process for $f(z) = (z-i)^{1/2}(z+1)(z+i)^2(z-1)^{-1}$. The set S contains 10 candidate points (green circles). (For interpretation of the references to color in this figure legend, the reader is referred to the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