Journal of Computational and Applied Mathematics 228 (2009) 418–423

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

Enclosing all zeros of an analytic function — A rigorous approach

Tomas Johnson, Warwick Tucker [*,1]

Department of Mathematics, Uppsala University, Box 480, Uppsala, Sweden

ARTICLE INFO

Article history:
Received 10 May 2007
Received in revised form 14 April 2008

MSC:
primary 65G20
65E05
secondary 65H05

Keywords:
Rigorous numerics
Argument principle
Root finding
Interval analysis

ABSTRACT

We present a method to find all zeros of an analytic function in a rectangular domain. The approach is based on finding guaranteed enclosures rather than approximations of the zeros. Well-isolated simple zeros are determined fast and with high accuracy. Clusters of zeros can in many cases be distinguished from multiple zeros by applying the argument principle to sufficiently high-order derivatives of the function. We illustrate the proposed method through five examples of varying levels of complexity.

1. Introduction

Finding approximate zeros of analytic functions is an important and well-studied problem. The case of polynomial zeros has been studied thoroughly, see e.g. [9,17,16]. Methods to find the number of zeros of a general analytic function are addressed in e.g. [8,13,19].

The method presented in [8] is based on the argument principle, and uses validated integration of contour integrals. It is rigorous since the error terms from the numerical quadrature are enclosed via interval arithmetic. The method described in [4] uses a bisection scheme to find enclosures of all zeros within a given rectangle, but is not rigorous. Combining the basic ideas of both mentioned papers, and introducing several improvements, we obtain an adaptive, rigorous method for locating enclosures of all zeros of an analytic function within a given rectangle. Several examples of the method's performance are presented below.

2. The general strategy

We will base our method on the argument principle (see e.g. [1]) restricted to rectangular domains R. If f is a meromorphic function in $R \subset \mathbb{C}$ not having any zeros or poles on the simple closed counter-clockwise oriented contour ∂R, we have

$$I(f;R) = \frac{1}{2\pi \mathrm{i}} \int_{\partial R} \frac{f'(z)}{f(z)}\,\mathrm{d}z = N - P. \tag{1}$$

Here, N and P are the number of f's zeros and poles (counting multiplicities), respectively, inside R. Seeing that we will only consider analytic functions, we always have $P = 0$. Thus, in order to determine the number of zeros of f via (1) using a

* Corresponding author. Fax: +46 018 471 32 01.
E-mail addresses: johnson@math.uu.se (T. Johnson), warwick@math.uu.se (W. Tucker).

[1] Research fellow of the Royal Swedish Academy of Sciences.

0377-0427/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2008.10.014

computer, it suffices to produce an enclosure $E(f;R)$ of $I(f;R)$ such that its real part has a diameter of less than one:

$$I(f;R) \in E(f;R) \quad \text{and} \quad \operatorname{diam}(\Re(E(f;R))) < 1.$$

Once we have established the unique integer value k of $I(f;R)$, we distinguish three cases:

a. $(k = 0)$ The rectangle R contains no zeros of f.
b. $(k = 1)$ The rectangle R contains exactly one zero of f.
c. $(k > 1)$ The rectangle R contains at least one, and at most k zeros of f.

In case (a), there is nothing to do: we simply remove R from further study. In case (b), however, we might want to refine the enclosure of the unique zero. This is done by a local search, first heuristically using Newton's method applied to f, and finally rigorously by a verification process described below. In case (c), we first attempt to shrink the domain as in (b), but using a Newton search applied to $f^{(k-1)}$. If this fails we generally bisect the rectangle along its widest side, and re-examine the two subrectangles separately according to (1). Only when a rectangle has reached a minimum size do we attempt to distinguish the case of a multiple zero from a cluster of simple zeros. This procedure is described below.

The outcome from this scheme is a list of rectangles, each having an associated integer: $\{R_i, k_i\}_{i=1}^m$, where $k_i = I(f;R_i)$. Note that $R = \cup_{i=1}^m R_i$, and $\operatorname{int}(R_i) \cap \operatorname{int}(R_j) = \emptyset$ for $i \neq j$. In the case $\max_i k_i \leq 1$, we have managed to isolate all zeros of f, and proved that they are all simple.

3. Computational aspects

When computing $I(f;R)$ via (1), we will decompose the contour of integration into its four line segments $\partial R = \gamma_1 + \cdots + \gamma_4$. Each line segment is parametrized by a linear function: $z_j\colon [0,1] \to \mathbb{C}$ defined by $z_j(t) = \gamma_j^-(1-t) + \gamma_j^+ t$, where γ_j^- and γ_j^+ are endpoints of γ_j chosen so that the positive orientation of ∂R is preserved.

With this parametrization, we arrive at the identity

$$I(f;R) = \frac{1}{2\pi \mathrm{i}} \sum_{j=1}^{4} \int_0^1 \frac{f'(z_j(t))}{f(z_j(t))} z_j'(t)\,\mathrm{d}t. \tag{2}$$

In what follows, we will concentrate on one of the four integrals appearing on the right-hand side of (2). In order to ease the reading, we will suppress the index j, and write $g(t) = f'(z(t))z'(t)/f(z(t))$. The task at hand, then, is to compute an enclosure of the integral

$$\tilde{I}(g;[0,1]) = \int_0^1 g(t)\,\mathrm{d}t. \tag{3}$$

We require that the enclosure of the imaginary part of (3) has a diameter less than $\pi/2$. This is achieved by adaptively inserting nodes t_k within the domain of integration $[0, 1]$.

The numerical quadrature will be based on Simpson's three-point method: given three consecutive nodes $t_{2k}, t_{2k+1}, t_{2k+2}$ satisfying $t_{2k+1} = (t_{2k} + t_{2k+2})/2$, we have

$$\int_{t_{2k}}^{t_{2k+2}} g(t)\,\mathrm{d}t = \underbrace{\frac{t_{2k+2} - t_{2k}}{6}\,(g(t_{2k}) + 4g(t_{2k+1}) + g(t_{2k+2}))}_{\text{approximation}} \underbrace{-\frac{(t_{2k+2} - t_{2k})^5}{2880} g^{(4)}(s_{2k})}_{\text{remainder}}, \tag{4}$$

where s_{2k} is some number between t_{2k} and t_{2k+2}. We will account for the remainder term by enclosing the range of $g^{(4)}$ over the subdomain $[t_{2k}, t_{2k+2}]$:

$$g^{(4)}(s_{2k}) \in \operatorname{range}\left(g^{(4)};[t_{2k}, t_{2k+2}]\right) = \{g^{(4)}(t)\colon t \in [t_{2k}, t_{2k+2}]\} \subseteq G^{(4)}([t_{2k}, t_{2k+2}]). \tag{5}$$

Here, $G^{(4)}$ is a set-valued, interval extension of $g^{(4)}$. This is obtained by a combination of complex interval arithmetic and automatic differentiation (see [2,5,11,12,14,18]).

In order to reach the desired quality of the global enclosure, we demand that the diameter of the imaginary part of each local enclosure of (5) be less than $(t_{2k+2} - t_{2k})\pi/2$. This is not hard to obtain seeing that the width of the enclosure comes from the remainder term, which contains a $(t_{2k+2} - t_{2k})^5$ factor.

One major strength of this approach is that the user need only provide the original function f and the domain R. The program automatically generates the re-parametrization g, as well as its interval extension G, and all necessary set-valued derivatives $G^{(k)}$ with mathematical rigour.

3.1. Improving enclosures of simple zeros

As soon as we encounter a region R_i that contains a unique simple zero, we attempt to improve the bounds via a Newton search. Taking the midpoint of the domain $z_0 = \operatorname{mid}(R_i)$ as starting point, we generate the sequence

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}. \tag{6}$$

This usually converges to some point z^* within a few iterates. Since this is obtained from a (non-validated) floating point iteration, we have to prove a posteriori that z^* indeed is a good approximation to the unique zero of f. This is done in three steps: (i) we verify that z^* belongs to the original enclosing domain R_i; (ii) we shrink R_i to a very small rectangle R_i^* containing z^*; (iii) we show that $I(f; R_i^*) = 1$. If (i)–(iii) hold, then we have proved that there is a zero in R_i^*, and that this is the only zero of f in R_i. If the validation fails, we return to the main program, and continue to bisect R_i.

3.2. Multiple zeros versus clusters

Given a rectangle R_i of minimal size, satisfying $I(f; R_i) = k > 1$, we would like to know whether R_i contains multiple zeros or not. To prove numerically that an analytic function has a multiple zero is, in general, not possible. What is possible, however, is to prove that no multiple zeros reside within R_i. This is achieved by applying the argument principle to the jth derivative $f^{(j)}$ for $j = 1, \ldots, k-1$. If $I(f^{(j)}; R_i) = 0$, the function f does not have any zeros of order $j+1$ within R_i. If successful, this procedure can establish the existence of several simple zeros within R_i. As mentioned earlier, the required derivatives are generated via automatic differentiation.

3.3. Improving enclosures of multiple zeros

As soon as we encounter a region R_i, satisfying $I(f; R_i) = k > 1$, we attempt to shrink R_i via a Newton search on $f^{(k-1)}$. Taking the midpoint of the domain $z_0 = \text{mid}(R_i)$ as starting point, we generate the sequence

$$z_{n+1} = z_n - \frac{f^{(k-1)}(z_n)}{f^{(k)}(z_n)}. \tag{7}$$

If the domain contains a zero of degree k, this sequence usually converges to some point z^* within a few iterates. Again, we have to prove a posteriori that z^* indeed is a good approximation to the simple zero of $f^{(k-1)}$. This is done in three steps: (i) we verify that z^* belongs to the original enclosing domain R_i; (ii) we shrink R_i to a very small rectangle R_i^* containing z^*; (iii) we show that $I(f; R_i^*) = k$. If (i)–(iii) hold, then we have proved that there are k zeros in R_i^*, and that these are the only zeros of f in R_i. If the validation fails, we return to the main program, and continue to bisect R_i.

4. Examples

All computations were performed on a Intel Xeon 2.0 GHz, 64 bit computer with 7970 MB of RAM. The program was compiled with `gcc`, version 3.4.6. The software for complex interval Taylor arithmetic was provided by the `CXS-C` package, version 2.1.1 (see [3,6]).

Example 1. Our first example is purely academic, and illustrates the adaptivity of the proposed method:

$$f_1(z) = z^{11} - a, \quad \left(a = \frac{1}{2} + \mathrm{i}\frac{\sqrt{3}}{2}\right).$$

This polynomial has exactly 11 zeros — all of them simple, and evenly distributed on the unit circle in $\mathbb{C}$. Starting with the domain $R = [-3, 3] + \mathrm{i}[-3, 3]$, 28 subdivisions and 11 Newton searches were needed to determine all zeros with nine decimals. The entire run time was 40 s, see Fig. 1. On average, we needed roughly 4 s per zero.

Example 2. Our second example is taken from [17] (see p. 119), and is a fifth order polynomial with clustered zeros:

$$f_2(z) = 70(z^2 - 2z + 3)^2 \left(z - \left(1 + \mathrm{i}\frac{99}{70}\right)\right).$$

This polynomial has two zeros of multiplicity two at $z = 1 \pm \mathrm{i}\sqrt{2}$, and one simple zero at $z = 1 + \mathrm{i}\frac{99}{70}$. Note that $|\sqrt{2} - \frac{99}{70}| < 7.3 \times 10^{-5}$. This means that three of the five zeros are clustered together. Searching over the domain $R = [-10, 10] + \mathrm{i}[-10, 10]$, 40 subdivisions and three Newton searches were needed to determine all the zeros (multiple and simple) with nine decimals, see Fig. 2. The entire run time was 27 s, which corresponds to roughly 5 s per zero on average.

Example 3. Next, we consider an example from [4],

$$f_3(z) = z^{50} + z^{12} - 5\sin(20z)\cos(12z) - 1,$$

which we prove to have exactly 424 zeros on the domain $R = [-20.3, 20.7] + \mathrm{i}[-5, 5.1]$ — all of them simple, see Fig. 3. A total of 1321 bisections were required in order to find the zeros with 9 decimals. The run time for this program was 63 min, indicating that our method is slow for functions with a large number of clustered zeros. On average, we needed roughly 9 s per zero.

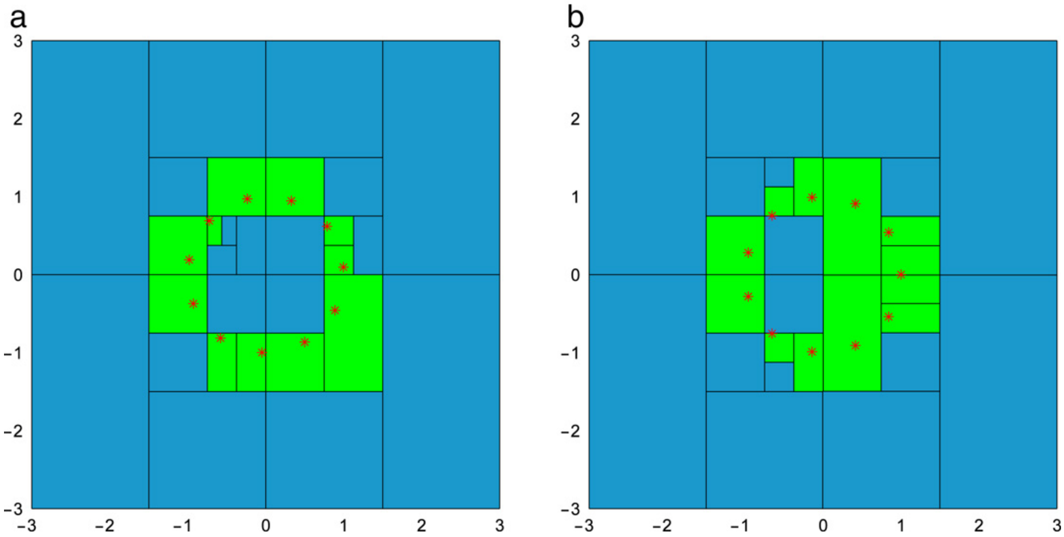

Fig. 1. Throughout the search, a subrectangle can be discarded (blue) or contracted (green). The final enclosures (too small to appear in the illustration) are marked with red stars. (a) When $a = \frac{1}{2} + i\frac{\sqrt{3}}{2}$, the boundaries of the subrectangles do not contain any zeros of f_1. (b) When $a = 1$, however, the zero at $z = 1$ requires that two subrectangles be slightly enlarged during the search. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

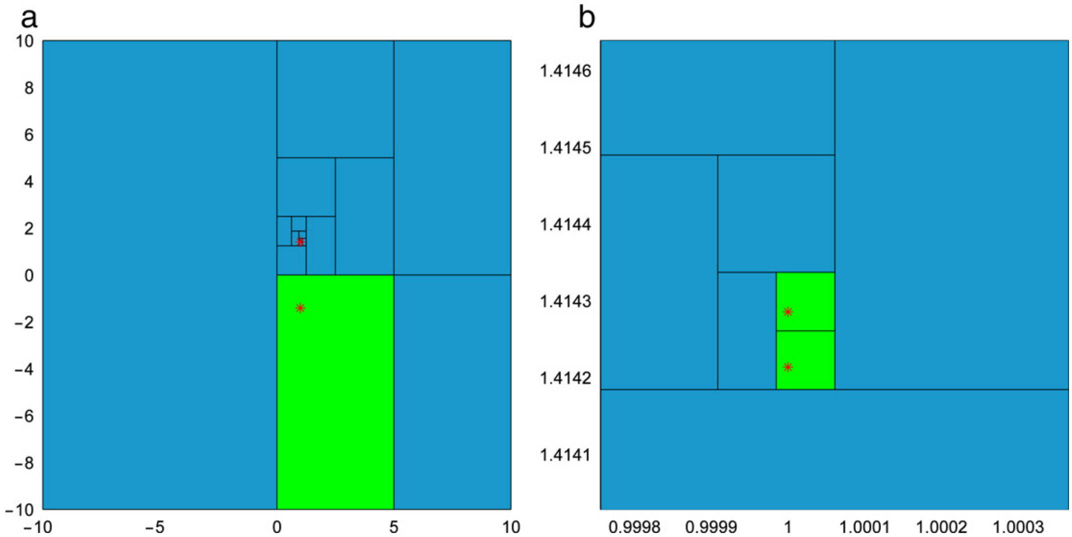

Fig. 2. (a) The two regions of five zeros of f_2. (b) A close-up of the three zeros with positive imaginary parts.

Example 4. Our second last example, also from [4], is based on a model for the stability of a flow inside of an annular combustion chamber:

$$f_4(z) = z^2 + Az + Be^{-Tz} + C.$$

The relevant parameter values are $A = -0.19435$, $B = 1000.41$, $C = 522\,463$, and $T = 0.005$. Using the same domain as in [4], $R = [-15\,000, 5000] + i[-15\,000, 15\,000]$, we prove that f_4 has exactly 24 zeros inside R — all of them simple, see Fig. 4. Since all of these are well separated, only 67 bisections were needed to isolate the zeros before the Newton step, and another 24 during the Newton search procedure. Thus, a total of 48 Newton searches had to be done in order to find the zeros with 9 decimals. The run time was 64 s, which means roughly 2 s per zero.

Comparing these results to the non-validated method in [4], one should note that both methods actually find all roots of f_3 and f_4 in the respective domains. We, however, can also *prove* that this really is the case.

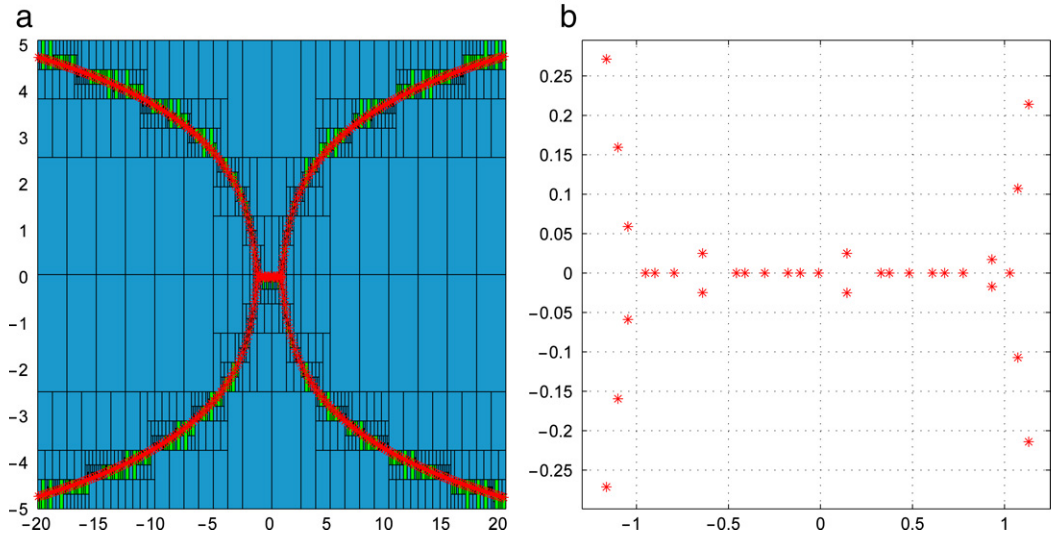

Fig. 3. (a) The 424 zeros of f_3 with the discarded/contracted regions. (b) A close-up on the zeros near the origin.

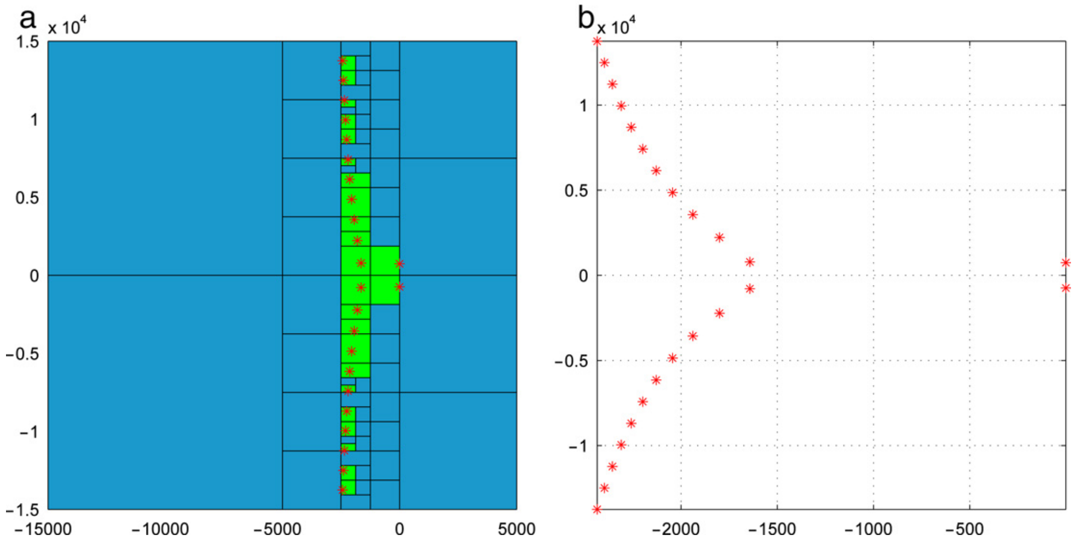

Fig. 4. (a) The 24 zeros of f_4 with the discarded/contracted regions. (b) A close-up on the zeros.

Example 5. As our final example, to illustrate that the presented method also works for complicated functions, we consider the Riemann Zeta function (see e.g. [10])

$$\zeta(s) = \sum_{n=1}^{\infty} n^{-s}.$$

To find an interval extension of $\zeta(s)$ we approximate it using the Euler–Maclaurin summation formula (see e.g. [7]), and enclose the error terms and their derivatives with interval extensions. This gives the following interval extension of $\zeta(s)$:

$$\zeta(s) \in \sum_{n=1}^{N} n^{-s} + \frac{1}{2}(1+N)^{-s} + \frac{(1+N)^{1-s}}{s-1} + \sum_{k=1}^{R} \frac{B_{2k}}{(2k)!} \prod_{l=0}^{2k-2}(s+l)(1+N)^{-s-2k+1} + \prod_{l=0}^{2R}(s+l)G(s,R,N), \tag{8}$$

where B_{2k} denote the even Bernoulli numbers and $G(s, R, N)$ is an interval extension of the remainder. Interval extensions of the derivatives of G are entered by hand, so that automatic differentiation can be used for the entire formula (4).

$$G(s,R,N) = 2/(2\pi)^{2R+1}\zeta(2R)(\sigma+2R)^{-1}(N+1)^{-(\sigma+2R)}I$$

$$G^{(k)}(s,R,N) = (\ln^k(1+N)G(s,R,N) + k(\sigma+2R)^{-1}G^{(k-1)}(s,R,N))I,$$

where $I = [-1, 1] + \mathrm{i}[-1, 1]$.

Table 1
A summary of the performance for the five examples.

Function	Zeros	Bisections	CPU time
f_1	11	28	00:00:40
f_2	5	40	00:00:27
f_3	424	1321	01:03:00
f_4	24	91	00:01:04
f_5	29	95	10:30:00

Examining the domain $[0.49, 0.51] + i[0, 100]$ (which encloses a portion of the critical line $\Re(s) = \frac{1}{2}$), we found 29 zeros — all simple. These were all determined with nine correct decimals. The total run time was 1h 30 min on seven parallel processors, and called for a total of 95 bisections. The complexity of these computations are illustrated by the tremendous increase in run time: on average, it took 22 min per zero. Of course, there exist much more effective methods for locating the zeros of the Riemann Zeta function, see e.g. [15].

5. Conclusions

We have presented a validated method which produces enclosures of all zeros of an analytic function in a bounded rectangular domain. Simple, well-spaced zeros are generally determined reasonably fast, and with high accuracy. Our method is also able to disprove the existence of multiple roots in a cluster of zeros. Its main strength, however, is that *all* zeros are accounted for: it is mathematically impossible for the algorithm to miss a zero.

As illustrated by the performance in the third and fifth examples (see Table 1), the time-consuming part of our algorithm is caused either by the number of bisections, or by a very expensive function evaluation. An obvious, partial, remedy to the latter would be to store all function evaluations. These could then be re-used when inserting new nodes in the adaptive quadrature scheme.

The proposed method is very user-friendly seeing that only the function itself, and no derivatives, are required by the user. Future research will aim at generalizing the method to arbitrary triangulated domains.

References

[1] L. Ahlfors, Complex Analysis, 1st ed., McGraw Hill, 1953.
[2] G. Alefeld, J. Herzberger, Introduction to Interval Computations, Academic Press, New York, 1983.
[3] CXSC — C++ eXtension for Scientific Computation, version 2.0. Available from: http://www.math.uni-wuppertal.de/org/WRST/xsc/cxsc.html.
[4] M. Dellnitz, O. Schütze, Q. Zheng, Locating all the zeros of an analytic function in one complex variable, J. Comput. Appl. Math. 138 (2) (2002) 325–333.
[5] A. Griewank, Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, SIAM Frontiers in Applied Mathematics, Philadelphia, 2000.
[6] R. Hammer, M. Hocks, U. Kulisch, D. Ratz, C++ Toolbox for Verified Computing, Springer-Verlag, New York, 1995.
[7] D. Hejhal, Lecture notes in complex analysis, available at the Ångström library, Uppsala University.
[8] J. Herlocker, J. Ely, An automatic and guaranteed determination of the number of roots of an analytic function interior to a simple closed curve in the complex plane, Reliab. Comput./Nadezhn. Vychisl. 1 (3) (1995) 239–249. English, Russian summary.
[9] J. Hubbard, D. Schleicher, S. Sutherland, How to find all roots of complex polynomials by Newton's method, Invent. Math. 146 (1) (2001) 1–33.
[10] A.E. Ingham, The Distribution of Prime Numbers, Cambridge University Press, 1932.
[11] R.E. Moore, Interval Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1966.
[12] R.E. Moore, Methods and Applications of Interval Analysis, SIAM Studies in Applied Mathematics, Philadelphia, 1979.
[13] A. Neumaier, An existence test for root clusters and multiple roots, Z. Angew. Math. Mech. 68 (6) (1988) 256–257.
[14] A. Neumaier, Interval Methods for Systems of Equations, in: Encyclopedia of Mathematics and its Applications, vol. 37, Cambridge Univ. Press, Cambridge, 1990.
[15] A.M. Odlyzko, A. Schoenhage, Fast algorithms for multiple evaluations of the Riemann zeta function, Trans. Amer. Math. Soc. 309 (1998) 797–809.
[16] V. Pan, Solving a polynomial equation: Some history and recent progress, SIAM Rev. 39 (1997) 187–220.
[17] M.S. Petković, Iterative Methods for Simultaneous Inclusion of Polynomial Zeros, in: Lecture Notes in Mathematics, vol. 1387, Springer-Verlag, 1989.
[18] M.S. Petković, L.D. Petković, Complex Interval Arithmetic and its Applications, in: Mathematical Research, vol. 105, Wiley-VCH, Verlag, Berlin, GmbH, Berlin, 1998.
[19] J. Rokne, Automatic error bounds for simple zeros of analytic functions, Comm. ACM 16 (1973) 101–104.