On a Fast and Accurate Method to Enclose All Zeros of an Analytic Function on a Triangulated Domain

Tomas Johnson[1] and Warwick Tucker[2]

[1] Dept. of Mathematics, Uppsala University
[2] Dept. of Mathematics, University of Bergen
tomas.johnson@math.uu.se
warwick.tucker@math.uib.no

Abstract. We present a fast hybrid method designed to enclose all zeros of an analytic function on a triangulated domain. The method consists of three parts: first the zeros are isolated (up to some resolution) using a combination of winding number computations and bisections; in the second step we approximate the location of each zero using a floating point complex Newton method; in the final step we validate the existence of an exact zero near each approximate zero by recalculating the winding number on a small enclosing triangle. The output of the program is a subset of a refinement of the original triangulation of the domain, together with the number of zeros on each remaining triangle. To demonstrate the method, we apply it on a few non–trivial examples.

Key words: Rigorous numerics, argument principle, root finding, interval analysis, triangulation.

1 Introduction

Finding approximate zeros of univariate analytic functions is an important and well–studied problem. The case of polynomial zeros has been studied thoroughly, see e.g. [8, 14, 15]. Methods to find the number of zeros of a general univariate analytic function are addressed e.g., in [7, 9, 12, 18].

The methods presented in [7, 9] are based on the argument principle; the first uses validated integration of contour integrals, whereas the second computes the winding number by discretising the boundary. Both methods are rigorous since the error terms from the numerical quadrature and summation, respectively, 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 the above–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 triangulation.

The main motivation for our approach is that the natural way to represent a domain is by a triangulation of it. In most cases such a representation is not

adapted to auto–validated algorithms, which are typically based on interval representations. For an introduction to interval computations and analysis, see e.g. [2, 10, 11, 13, 17]. The reason is that normally one needs to bound the ranges of some functions on the entire domain, which is therefore covered by (axis parallel) boxes, suitable for interval computations. When determining the zeros of an analytic function, however, we only need boundary data and point data, therefore it is possible to use the canonical representation of the domain. In addition, there are fast floating point algorithms designed to solve the root finding problem. If possible, one should try to use these to improve the efficiency of the auto–validated algorithm, see [16] for a general discussion of hybrid methods.

2 The General Strategy

We will base our method on the argument principle (see e.g. [1]) on a domain $|K|$, where K is a triangulation,

$$|K| = \bigcup_{T \in K} T, \quad \mathrm{int}(T) \cap \mathrm{int}(\widetilde{T}) = \emptyset, \text{ for } \mathrm{T} \neq \widetilde{\mathrm{T}}.$$

Note, with a triangulation we denote throughout the paper a literal triangulation, i.e. a triangle consists of three points that are connected with straight lines. In principle it would we possible to implement the algorithm allowing for curvilinear triangulations. That would, however, make the adaptive procedure which refines the triangulation much more complicated.

If f is a meromorphic function in $T \in K$, without zeros or poles on ∂T, then the *winding number* of f on T, $I(f;T)$, is well–defined, see Section 3.1. On the simple, closed, counter–clockwise oriented contour ∂T, we have

$$I(f;T) = N - P. \tag{1}$$

Here, N and P are the number of f's zeros and poles (counting multiplicities), respectively, inside T. Seeing that we only consider analytic functions, we always have $P = 0$. A consequence of this is that we can ignore the orientation of the triangles, which simplifies our algorithms. Thus, in order to determine the number of zeros of f via (1), it suffices to produce an enclosure $E(f;T)$ of $I(f;T)$ such that its diameter is less than one:

$$I(f;T) \in E(f;T) \qquad \text{and} \qquad \mathrm{diam}\big(E(f;T)\big) < 1. \tag{2}$$

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

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

In case (a), there is nothing to do: we simply remove T from further study. In case (b), however, we might want to refine the enclosure of the isolated 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 triangle along its widest side, and re–examine one of the two subtriangles. Note that we get the winding number of f on the second subtriangle for free, since their sum is the winding number of the old triangle.

The outcome from this scheme is a list of triangles, a subset of a refinement of the original triangulation, each having an associated integer: $\{(T_i, k_i)\}_{i=1}^{m}$, where $k_i = I(f; T_i)$. In the case $\max_i k_i = 1$, we have managed to isolate all the zeros of f and prove that they are all simple.

3 Computational Aspects

3.1 Computing the Winding Number

To compute the winding number of a function f on a single triangle T, one can either, as is done in [7], compute the contour integral

$$I(f;T) = \frac{1}{2\pi} \int_{\partial T} \frac{f'(z)}{f(z)}\, dz, \tag{3}$$

or, as is done in [9], compute the accumulated change in the argument along ∂T,

$$I(f;T) = \frac{1}{2\pi} \sum_{n=1}^{N} \arg\left(\frac{f(c_{n+1})}{f(c_n)}\right), \tag{4}$$

where $\{c_n\}_{n=1}^{N}$ is a discretisation of the boundary such that

$$|\Delta \arg_{\gamma_n}(f(z))| < \pi, \tag{5}$$

which can be assured by testing

$$0 \notin f(\gamma_n), \tag{6}$$

where γ_n is the line segment between c_n and c_{n+1}. This requirement also ensures that there are no zeros on the boundary. If (6) is not satisfied, this is interpreted as if there was a zero on the boundary, and the zeros inside of the triangle have to be enclosed by the triangle from the previous subdivision level. If (4) is not satisfied by the original triangulation, the algorithm fails to integrate on the corresponding triangle. In such a case, unlikely in practice, the user has to change the representation of the domain and redo the computation.

We use the latter method, (4), initiated with a discretisation point in each corner of the triangle T. The boundary of T is thereafter subdivided until conditions (2) and (6) are satisfied, which validates (4), and gives the unique number

$$k = E(f;T) \cap \mathbb{N}.$$

3.2 Improving the Enclosures of the Zeros

The procedure described here is applied to zeros of any order, but to keep the presentation simple we only describe the case of a simple zero. As soon as we encounter a triangle T that contains a unique simple zero, we attempt to improve the bounds via a Newton search; the necessary derivatives are computed using automatic differentiation [5]. Taking the midpoint of the triangle $z_0 = \mathrm{mid}(T)$ as starting point, we generate the sequence

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

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

For all of the examples we have tried, the addition of this floating point procedure to our algorithm decreases the run–time of the program by a factor between 15 and 330, compared to a bisection only algorithm, see Section 4.

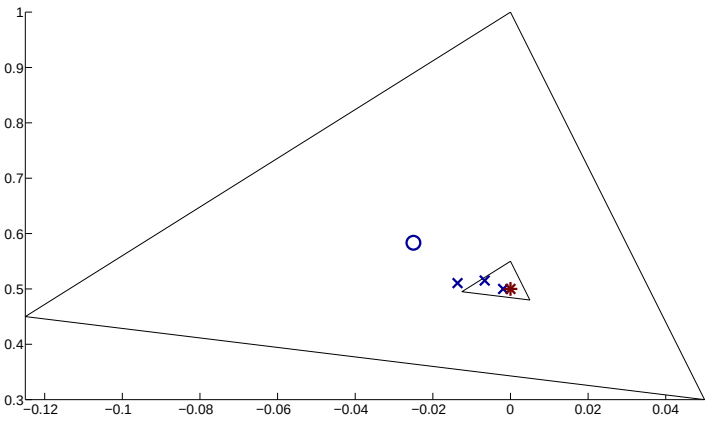

Fig. 1. Improving the enclosure of a zero, the circle is the starting point of the Newton iteration, the approximate zero is marked by a star. Intermediate points are marked by crosses.

3.3 Multiple zeros versus clusters

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

4 Examples

All computations were performed on a Intel Xeon 2.0 Ghz, 64bit computer with 7970Mb 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]).

4.1 Example 1

We begin by providing an example on a box, that is, a domain adapted to interval representations, to show that our approach also works well for such domains. We consider the following function, as in [4],

$$f_1(z) = z^2 + Az + Be^{-Tz} + C, \tag{8}$$

where the relevant parameter values are $A = -0.19435$, $B = 1000.41$, $C = 522463$, and $T = 0.005$. The domain is the rectangle $R = [-5000, 5000] + i[-15000, 15000]$ triangulated with two triangles, separated by the diagonal from the lower left corner to the upper right corner. All 24 zeros are located with 5 decimals in 3 seconds. The final triangulation consists of 91 triangles, see Figure 2. Removing the Newton search from the algorithm increases the run-time to 392 seconds.

4.2 Example 2

Our second example regards a non-entire function on the unit disk. We consider a rational function with isolated zeros on a circle slightly inside of the unit circle, and isolated poles slightly outside of it, see Figure 3.

$$f_2(z) = \frac{(z - i0.0067)^{37} - \frac{1}{\sqrt{2}}}{z^{200} - 1.1}. \tag{9}$$

Note that, $\left(\frac{1}{\sqrt{2}}\right)^{\frac{1}{37}} \approx 0.9907$, and $(1.1)^{\frac{1}{200}} \approx 1.0005$. Using an initial triangulation of the unit disk consisting of 8177 triangles, all 37 zeros are located with 6

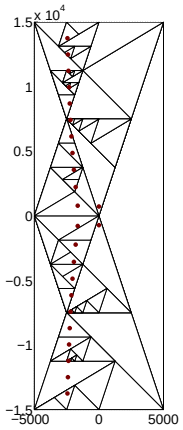

Fig. 2. Illustration of the result from Example 1, the points are the approximations to the zeros.

decimals in 115 seconds. The final triangulation consists of 26669 triangles, see Figure 4. Removing the Newton search from the algorithm increases the run-time to 1720 seconds.

4.3 Example 3

Next, we consider an example from [4], which we multiply by

$$\sin\left(\frac{z^2}{(z^2-(1+i)^2)(z^2-(1-i)^2)}\right)$$

in order to introduce four poles at $\pm 1 \pm i$, and a double zero at the origin. The function is studied on a star-shaped domain, which does not contain the poles, see Figure 5. The initial triangulation consists of 162 triangles.

$$\begin{aligned} f_3(z) = &\left(z^{50}+z^{12}-5\sin(20z)\cos(12z)-1\right) \\ &\times \sin\left(\frac{z^2}{(z^2-(1+i))(z^2-(1-i))}\right). \end{aligned} \tag{10}$$

We locate the 34 zeros inside of the domain, including the double zero at the origin, with 6 decimals in 18 seconds. The final triangulation consists of 232 triangles. Removing the Newton search from the algorithm increases the run-time to 5964 seconds.

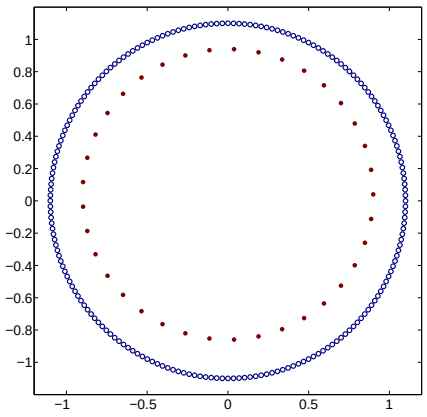

Fig. 3. The zeros and poles, marked by points and circles, respectively, for the function in Example 2. The distance between the zeros and poles is exaggerated in the figure.

5 Conclusions

We have presented a validated method to enclose and approximate all zeros of an analytic function on the natural representation of its domain, that is, on a triangulation of it. The method is fast, especially for well-spaced zeros. In addition, we are able to distinguish between clusters and multiple zeros, by disproving the existence of a multiple zero.

Function	Function Calls with Newton	Function Calls without Newton
f_1	22 333	12 636 415
f_2	430 259	11 016 424
f_3	31 230	11 174 209

Table 1. Summary of the performance

References

1. L. Ahlfors, Complex Analysis. McGraw Hill, 1st ed., 1953
2. G. Alefeld, and 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

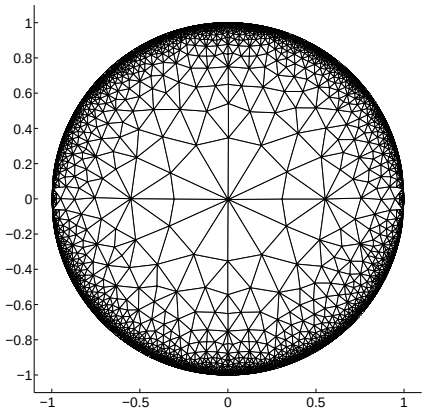

Fig. 4. Illustration of the final triangulation from Example 2.

4. M. Dellnitz, O. Schütze, Q. Zheng, Locating all the zeros of an analytic function in one complex variable. J. Comput. Appl. Math. 138 (2002), no. 2, 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, and D. Ratz, C++ Toolbox for Verified Computing, Springer-Verlag, New York, 1995.
7. 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. (English, Russian summary) Reliab. Comput./Nadezhn. Vychisl. 1 (1995), no. 3, 239–249.
8. J. Hubbard, D. Schleicher, S. Sutherland, How to find all roots of complex polynomials by Newton's method. Invent. Math. 146 (2001), no. 1, 1–33.
9. K.H. Ko, T. Sakkalis and N.M. Patrikalakis A reliable algorithm for computing the topological degree of a mapping in R2 App. Math. Comp., Volume 196, Issue 2, 1 March 2008, Pages 666-678
10. R.E. Moore, Interval Analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1966.
11. R.E. Moore, Methods and Applications of Interval Analysis, SIAM Studies in Applied Mathematics, Philadelphia, 1979.
12. A. Neumaier, An existence test for root clusters and multiple roots. Z. Angew. Math. Mech. 68 (1988), no. 6, 256–257.
13. A. Neumaier, Interval Methods for Systems of Equations. Encyclopedia of Mathematics and its Applications 37, Cambridge Univ. Press, Cambridge, 1990
14. V. Pan, Solving a polynomial equation: some history and recent progress SIAM Review 39, 187–220, 1997.
15. M.S. Petković. Iterative Methods for Simultaneous Inclusion of Polynomial Zeros. Lecture Notes in Mathematics 1387, Springer-Verlag, 1989.
16. M.S. Petković, J. Herzberger, Hybrid inclusion algorithms for polynomial multiple complex zeros in rectangular arithmetic. Appl. Numer. Math. 7 (1991), no. 3, 241–262.

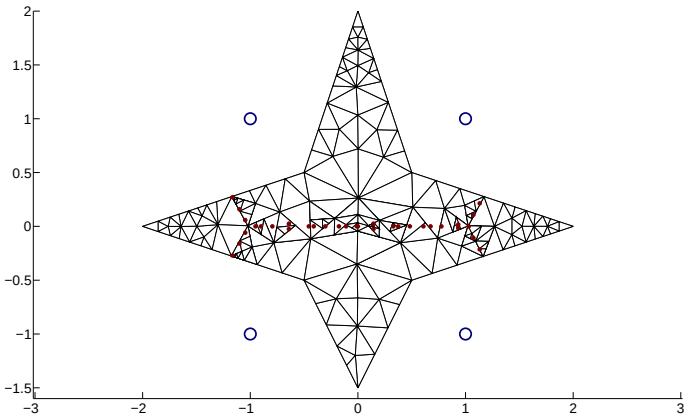

Fig. 5. Illustration of the result from Example 3, the points are the approximations to the zeros, the outer circles are the poles.

17. M.S. Petković, L.D. Petković. Complex interval arithmetic and its applications. Mathematical Research, 105. Wiley-VCH Verlag Berlin GmbH, Berlin, 1998.
18. J. Rokne. Automatic error bounds for simple zeros of analytic functions Comm. ACM 16, 101–104, 1973.