

# Algorithm 987: MANBIS—A C++ Mathematical Software Package for Locating and Computing Efficiently Many Roots of a Function: Theoretical Issues

DIMITRA-NEFELI A. ZOTTOU, DIMITRIS J. KAVVADIAS, FROSSO S. MAKRI,  
and MICHAEL N. VRAHATIS, University of Patras

MANBIS is a C++ mathematical software package for tackling the problem of computing the roots of a function when the number of roots is very large (of the order of hundreds or thousands). This problem has attracted increasing attention in recent years because of the broad variety of applications in various fields of science and technology. MANBIS applies the bisection method to obtain an approximate root according to a pre-determined accuracy. Thus, the only computable information required is the algebraic signs of the considered function, which is the smallest amount of information (one bit of information) necessary for the purpose needed, and not any additional information. MANBIS is able to compute very efficiently a user-given percentage of roots and draws its strength from the fact that the roots are expected to be many. Furthermore, MANBIS is capable of estimating without any additional function computational cost the total number of roots within the user-given interval. Our approach can also be efficiently applied in cases where the distribution of the roots is not known. This article is accompanied by another article where the user manual, some implementation details, and some examples are included.

Categories and Subject Descriptors: G.1.0 [Numerical Analysis]: General—*Numerical algorithms*; G.4 [Mathematical Software]—*Algorithm design and analysis*

General Terms: Algorithms

Additional Key Words and Phrases: Root finding, zero finding, many zeroes, expected behavior, bisection-based methods, imprecise function values, counting and computing roots, very large problems

## ACM Reference format:

Dimitra-Nefeli A. Zottou, Dimitris J. Kavvadias, Frosso S. Makri, and Michael N. Vrahatis. 2018. Algorithm 987: MANBIS—A C++ Mathematical Software Package for Locating and Computing Efficiently Many Roots of a Function: Theoretical Issues. *ACM Trans. Math. Softw.* 44, 3, Article 35 (April 2018), 7 pages.

<https://doi.org/10.1145/3155744>

## 1 INTRODUCTION

In this article we give the theoretical basics of a software package called *MANBIS* written in C++, which implements an algorithm for computing a user-given percentage  $q \in (0, 100)$  of the total number of roots of a continuous function  $f : \mathbb{R} \rightarrow \mathbb{R}$  within a user-defined interval  $[a, b] \subset \mathbb{R}$ .

For the computation of a root with an accuracy  $\varepsilon \in (0, 1)$ , MANBIS applies a bisection method (Vrahatis 1988a, 1988b) that requires only the algebraic signs of the function values to be computed,

Authors' address: D.-N. A. Zottou, D. J. Kavvadias, F. S. Makri, and M. N. Vrahatis, Department of Mathematics, University of Patras, GR-26110 Patras, Greece; emails: {nefeli.zottou, djk, makri, vrahatis}@math.upatras.gr.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2018 ACM 0098-3500/2018/04-ART35 \$15.00

<https://doi.org/10.1145/3155744>

and thus it can be applied to problems with imprecise function values. This is an important issue in cases, among others, where the function value follows as a result of an infinite series (e.g., Bessel, Airy functions), because it can be shown (Vrahatis et al. 1997) that the sign of a function value stabilizes after a relatively small number of terms of the series and the calculations can be sped up considerably.

Apart from the special function problem areas mentioned previously, the problem of the computation of many roots is related to other very important problems including the famous Riemann's hypothesis and Elbert's conjecture, among others (see Kavvadias et al. (2005)). For a brief discussion of the importance of the massive computation of roots or extrema of a function in one variable, we refer the interested reader to Kavvadias et al. (2005).

Of course, there exist various theoretical and practical approaches to the root-finding problem, but to our knowledge none that is focused on functions with many roots and, in addition, with the property of allowing a kind of trade-off between the computational cost and the number of discovered roots.

The main issues of the proposed package are summarized as follows:

- (1) MANBIS is the first package (to the best of our knowledge) for computing very "cheaply" a percentage of the roots within the user-given interval when the roots are expected to be many. The algorithm benefits from the abundance of roots in the given interval and discovers a fraction of them with little computational cost *per root*. This per root computational cost increases as the required fraction increases, and it reaches that of an exhaustive search of the interval when the fraction is 100%. Therefore, its features are best exploited when the user is interested in computing many (but not all) of the roots and possibly searches the intervals between two discovered roots using other methods.
- (2) MANBIS proceeds solely by requiring only the smallest amount of information (one bit of information) necessary for the purpose needed and not any additional information.
- (3) MANBIS is capable of estimating without any additional function computational cost the total number of roots within the user-given interval. This estimation is improved after each iteration by taking into consideration the newly discovered roots.
- (4) MANBIS is capable of estimating in advance the total computational cost required for approximating a root with a given accuracy. Thus, in each iteration, the total computational cost for computing all required roots is known beforehand.
- (5) As MANBIS is a bisection-based algorithm, it behaves very well in the presence of noise in the given function. The bisection method only requires the sign of the function to proceed correctly, and this retains its correct value even in the presence of large amplitude noise. In addition, randomly changing the roots by shifting them left or right by a small margin (not exiting  $\epsilon$ , so two adjacent roots do not overlap) results in exactly the same behavior as before. We have tested this in several instances, including example fun100 (see Zottou et al. (2018)).

In Section 2, our main algorithm, MANBIS, along with its theoretical basis, is presented. In particular, the root-finding methods used are described, along with the mathematical framework for estimating the total number of roots and the termination criteria. In Section 3, a synopsis and some concluding remarks are presented.

## 2 MANBIS ALGORITHM AND ITS THEORETICAL BASIS

In this section, we briefly present the necessary theoretical results, concepts, and methods used in the MANBIS algorithm.

## 2.1 Root-Finding Methods Used

A simple and very useful criterion for the existence of a root of a continuous function  $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$  in some interval  $(a, b)$  is Bolzano's existence criterion:

$$f(a)f(b) < 0 \iff \operatorname{sgn} f(a) \operatorname{sgn} f(b) = -1,$$

where  $\operatorname{sgn}$  denotes the three-valued sign function. This criterion is well known and widely used, and it can be generalized to higher dimensions (see Heindl (2016) and Vrahatis (1989, 2016)). Note that when this criterion is not fulfilled, then either no root exists or there is an even number of simple roots within the considered interval (i.e., whenever  $f(x) = 0$ , we also have  $f'(x) \neq 0$ ).

MANBIS implements a bisection method that is based on the Bolzano existence criterion to compute the roots. Specifically, it uses the following simplified version described in Vrahatis (1988a):

$$x_{i+1} = x_i + c \operatorname{sgn} f(x_i) / 2^{i+1}, \quad i = 0, 1, \dots, \quad (1)$$

where  $x_0 = a$  and  $c = \operatorname{sgn} f(a) (b - a)$ . The sequence (1) converges with certainty to a root  $r \in (a, b)$  if for some  $x_i$ ,  $i = 1, 2, \dots$  holds that

$$\operatorname{sgn} f(x_0) \operatorname{sgn} f(x_i) = -1.$$

Furthermore, the number of iterations  $\nu$  required to obtain an approximate root  $r^*$  such that  $|r - r^*| \leq \varepsilon$  for some  $\varepsilon \in (0, 1)$  is given by

$$\nu = \lceil \log_2(b - a) \varepsilon^{-1} \rceil. \quad (2)$$

Instead of the iterative formula (1), we can also use

$$x_{i+1} = x_i - \hat{c} \operatorname{sgn} f(x_i) / 2^{i+1}, \quad i = 0, 1, \dots, \quad (3)$$

where  $x_0 = b$  and  $\hat{c} = \operatorname{sgn} f(b) (b - a)$ .

The reasons for choosing the bisection method given by the iterative scheme (1) or (3) are as follows:

- (1) It is a globally convergent method in the sense that it converges to a root from remote initial guesses.
- (2) It converges with certainty within the given interval  $(a, b)$ .
- (3) It has a great advantage since it is worst-case optimal. In other words, it possesses asymptotically the best possible rate of convergence in the worst case (Sikorski 2001). This means that it is guaranteed to converge within the predefined number of iterations, and, moreover, no other method has this important property.
- (4) Using relation (2), we may predetermine the number of iterations that are required for the attainment of an approximate root to a given accuracy.
- (5) It requires only the algebraic signs of the function values to be computed, as is evident from (1) or (3); thus, it can be applied to problems with imprecise function values.

## 2.2 Estimating the Total Number of Roots and Termination Criteria

The termination criterion of the MANBIS algorithm requires the knowledge of the total number of roots,  $N$ , in the considered interval. The algorithm estimates the total number of roots without any additional computational cost regarding function evaluations while simultaneously computing the roots themselves. The estimation of the total number of roots  $N$  is revised and improved after each iteration, when new roots have been discovered. Thus, at the first stages of the algorithm, there is only a rough estimation of the number of roots (none at the beginning); however, as the algorithm proceeds, the additional information allows us to be more accurate in our prediction.

The use of the statistical estimation becomes more important and useful when MANBIS is used to solve many root-finding problems of functions drawn from a space of functions with uniformly distributed roots. It is under these circumstances that the statistical estimations become valid in the long run.

The algorithm, which MANBIS implements, sets as a goal the percentage of the roots in the interval that need to be computed before the algorithm halts. The algorithm constantly updates its estimation of the total number of roots as it discovers new ones; if the required percentage is achieved, the algorithm terminates. The theory behind this is described in Kavvadias et al. (2005). In practice, however, some instances have displayed a behavior where, at the early stages, the number of discovered roots is very small to constitute a meaningful sample; as a consequence in such cases, the statistical estimations may fail. To remedy this behavior, in the current implementation, an additional (not present in the previous reference) stopping criterion has been added to the algorithm, which reduces the risk of an early termination. This may occur before achieving the required percentage of the actual number of roots as a result of a false estimation that the roots in the interval are fewer than what they really are. This additional criterion takes into account the difference between two consecutive estimations; if they differ by a small margin, then the additional stopping criterion is considered fulfilled. This criterion is controlled by a parameter the user sets before the execution of the program.

Of course, the statistics are valid when the functions have uniformly distributed roots, but it is important to stress that the software package can be used efficiently for solving a root-finding problem even when the distribution of the roots of the function is unknown. In this case, however, the statistical estimation may fail. In such a case, the user is advised to repeat the procedure more than once with a different required percentage and/or more strict allowed difference of estimations in two consecutive samplings and then observe possible variations in the estimations. In general, larger percentages are to be trusted more, even in the case of an unknown distribution. The estimation with the largest number of roots should be kept.

For our estimation method, we need the following two propositions, proved in Kavvadias et al. (2005), which we restate here for completeness.

**PROPOSITION 2.1.** *Under the assumption that the roots are randomly and uniformly distributed, let the total number of roots in the considered interval be  $N$ . Then the probability  $p_{\text{odd}}$  that a subinterval of length  $\ell$  contains an odd number of roots is given by*

$$p_{\text{odd}} = \frac{1 - (1 - 2\ell)^N}{2}. \quad (4)$$

**PROOF.** See Kavvadias et al. (2005).  $\square$

**PROPOSITION 2.2.** *Assume that in a family of  $m$  nonoverlapping subintervals all of the same length  $\ell$ ,  $k$  subintervals have opposite signs at their endpoints. Then, with probability at least  $(1 - \alpha)$ , for any  $\alpha$  between 0 and 1, the probability  $p_{\text{odd}}$  that any subinterval of length  $\ell$  has an odd number of roots is between*

$$p_{\text{lower}} \leq p_{\text{odd}} \leq p_{\text{upper}}, \quad (5)$$

where

$$p_{\text{lower}} = \frac{k - z_{\alpha/2}}{m} \sqrt{\frac{k(m-k)}{m}}, \quad (6)$$

and

$$p_{\text{upper}} = \frac{k + z_{\alpha/2}}{m} \sqrt{\frac{k(m-k)}{m}}, \quad (7)$$

where  $z_{\alpha/2}$  is a constant which can be determined from

$$\text{Prob}(-z_{\alpha/2} \leq Z \leq z_{\alpha/2}) = 1 - \alpha, \quad (8)$$

where  $Z$  is a random variable having the standard normal distribution.

PROOF. See Kavvadias et al. (2005). □

For example, when  $\alpha = 0.05$ , then  $z_{\alpha/2} = 1.96$ . We use the preceding confidence interval to estimate  $p_{\text{odd}}$  and then use (4) to estimate  $N$ . This is done by computing two values for  $N$ , calling them  $N_{\text{lower}}$  and  $N_{\text{upper}}$ . The first one is computed by solving the equation (with respect to  $N$ )

$$p_{\text{lower}} = \frac{1 - (1 - 2\ell)^N}{2}, \quad (9)$$

and the second is computed by solving the equation

$$p_{\text{upper}} = \frac{1 - (1 - 2\ell)^N}{2}. \quad (10)$$

Notice that in some cases,  $N_{\text{upper}}$  and even  $N_{\text{lower}}$  can be infinite. This will happen when insufficient data are available for determining exact values for the confidence interval. But for our purposes, this is immaterial in the sense that in either of these cases, we continue subdividing the intervals.

### 2.3 MANBIS Algorithm

In this section, we present a detailed step-by-step description of the main algorithm (see Algorithm 1). In Algorithm 1, the following input values are required:

- (1) A continuous function  $f : \mathbb{R} \rightarrow \mathbb{R}$
- (2) An interval  $[a, b] \subset \mathbb{R}$
- (3) An accuracy  $\varepsilon \in (0, 1)$
- (4) The desired percentage  $q \in (0, 100)$  of the number of roots within  $(a, b)$ .

On output, the algorithm gives the computed roots stored in set  $S$  whose cardinality  $d = |S|$ , which of course determines the total number of the computed roots.

Based on the approach of Section 2.2, MANBIS uses the following termination criteria. When, after some iterations, both ends of the confidence interval are finite, we estimate the total number of roots  $N$  within the considered interval using the following relation:

$$N = \frac{1}{2}(N_{\text{lower}} + N_{\text{upper}}). \quad (11)$$

Assume now that the user desires the percentage  $q \in (0, 100)$  of the total number of roots to be computed within the considered interval  $(a, b)$ . Thus, MANBIS terminates when the following criterion is fulfilled:

$$d \geq \frac{q}{100} N, \quad (12)$$

where  $d$  is the number of the computed roots. A more strict termination criterion can be formed by choosing  $N$  to be  $N_{\text{upper}}$ .

*Remark 2.3.* The preceding criteria that are used in MANBIS require that a percentage of the total number of roots must be discovered before terminating the algorithm independently of the accumulated computational cost. An additional criterion could be formed by terminating the algorithm if a predefined “budget” of function evaluations has expired.

---

**ALGORITHM 1: MANBIS:** Computing **MANy** roots using a **BIS**ection method

---

**Input:** A function  $f : \mathbb{R} \rightarrow \mathbb{R}$ ; an interval  $[a, b] \subset \mathbb{R}$ ; an accuracy  $\varepsilon \in (0, 1)$ ; a percentage  $q \in (0, 100)$ .**Output:** The set  $S$  of computed roots; the cardinality  $d = |S|$ .Set  $d = |S| = 0$ ;Set  $i = 0$ ;Set  $tc = \text{.FALSE.}$ ;Divide the interval  $[a, b]$  into two equal subintervals;**repeat**    Store the subintervals with opposite signs at their endpoints into set  $A$ ;    Set  $k = |A|$ ;    Store the subintervals with the same signs at their endpoints into set  $B$ ;    Set  $i = i + 1$ ;    Set  $m = 2^i$ ;    **for each interval in  $A$  do**        Compute one root using bisection (1) or (3) with accuracy  $\varepsilon$  and add in each iteration the obtained subintervals whose endpoints are of the same sign into set  $B$ ;        Store the computed root into set  $S$ ;        Set  $d = d + 1$ ;    **end**    Set  $\ell = 2^{-i}$  and by using the parameters  $m = 2^i$  and  $k$  estimate the total number of roots  $N$  using relations (9), (10), and (11), where  $p_{\text{lower}}$  and  $p_{\text{upper}}$  are given in Proposition 2.2;    **if** ( $N_{\text{lower}}$  is finite) *and* ( $N_{\text{upper}}$  is finite) *and* ( $d \geq qN/100$ );        **then**            Set  $tc = \text{.TRUE.}$ ;        **else**            Replace in set  $B$  each subinterval of largest size by its two halves;        **end****until**  $tc = \text{.TRUE.}$ ;

---

### 3 SYNOPSIS AND CONCLUDING REMARKS

In this article, the theoretical background of the MANBIS software package has been described. This package computes a user-given percentage of the total number of roots according to the user-given accuracy of the user-defined continuous real function of one variable within the user-given interval.

The termination criteria of MANBIS require knowledge of the total number of roots in the given interval. MANBIS, without any additional function computational cost and in parallel with its main task of computing roots, also estimates the total number of roots. This estimation is revised and improved after each iteration, as new additional roots are computed. Thus, in the initial stages, there is only a rough estimation of the number of roots; however, as the algorithm proceeds, a more accurate estimate is generated. Furthermore, in cases where the number of roots is very large (of the order of hundreds or thousands), the estimation becomes better. This holds in cases where the roots are uniformly distributed. Thus, this estimation becomes more important and useful when MANBIS is used to solve many root-finding problems of functions drawn from a space of functions with uniformly distributed roots (see Riemann's  $\zeta$ -function (Kavvadias et al. 2005)).

It is important to point out, however, that MANBIS can also be applied in cases where the roots are not uniformly distributed in the considered interval or when the distribution of the roots is

unknown. In general, this information is not available a priori. However, a natural choice when no additional information about the distribution of the roots of the function is given is to consider the roots uniformly distributed in the defined interval.

The computational results of MANBIS are reported in Zottou et al. (2018). As a general observation, these results show that the behavior of MANBIS in practice is, to a large extent, as anticipated by its theoretical analysis, although some fine-tuning was required to tackle some practical issues.

## ACKNOWLEDGMENTS

We wish to express our appreciation and thanks to Dr. Tim Hopkins and the referees for their constructive and valuable comments.

## REFERENCES

- G. Heindl. 2016. Generalizations of theorems of Rohn and Vrahatis. *Reliable Comput.* 21, 109–116.
- D. J. Kavvadias, F. S. Makri, and M. N. Vrahatis. 2005. Efficiently computing many roots of a function. *SIAM J. Sci. Comput.* 27, 1, 93–107.
- K. Sikorski. 2001. *Optimal Solution of Nonlinear Equations*. Oxford University Press, New York, NY.
- M. N. Vrahatis. 1988a. Solving systems of nonlinear equations using the nonzero value of the topological degree. *ACM Trans. Math. Softw.* 14, 4, 312–329.
- M. N. Vrahatis. 1988b. CHABIS—a mathematical software package for locating and evaluating roots of systems of nonlinear equations. *ACM Trans. Math. Softw.* 14, 4, 330–336.
- M. N. Vrahatis. 1989. A short proof and a generalization of Miranda’s existence theorem. *Proc. Am. Math. Soc.* 107, 701–703.
- M. N. Vrahatis. 2016. Generalization of the Bolzano theorem for simplices. *Topology Appl.* 202, 40–46.
- M. N. Vrahatis, T. N. Grapsa, O. Ragos, and F. A. Zafiroopoulos. 1997. On the localization and computation of zeros of Bessel functions. *Z. Angew. Math. Mech.* 77, 467–475.
- D.-N. A. Zottou, D. J. Kavvadias, F. S. Makri, and M. N. Vrahatis. 2018. Algorithm 987: MANBIS—a C++ mathematical software package for locating and computing efficiently many roots of a function: Usage and practical issues. *Collected Algorithms of the ACM*. <http://calgo.acm.org/987.zip>.

Received March 2016; revised July 2017; accepted October 2017