Effective Subdivision Algorithm for Isolating Zeros of Real Systems of Equations, with Complexity Analysis

Juan Xu*  
LMIB-School of Mathematics and Systems Science,  
Beihang University  
Beijing, China  
xujuan@buaa.edu.cn

Chee Yap†  
Courant Institute, NYU  /New York, NY  
yap@cs.nyu.edu

ABSTRACT
We describe a new algorithm Miranda for isolating the simple zeros of a function $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ within a box $B_0 \subseteq \mathbb{R}^n$. The function $f$ and its partial derivatives must have interval forms, but need not be polynomial. Our subdivision-based algorithm is “effective” in the sense that our algorithmic description also specifies the numerical precision that is sufficient to certify an implementation with any standard BigFloat number type. The main predicate is the Moore-Kioustelidis (MK) test, based on Miranda’s Theorem (1940). Although the MK test is well-known, this paper appears to be the first synthesis of this test into a complete root isolation algorithm.

We provide a complexity analysis of our algorithm based on intrinsic geometric parameters of the system. Our algorithm and complexity analysis are developed using 3 levels of description (Abstract, Interval, Effective). This methodology provides a systematic pathway for achieving effective subdivision algorithms in general.

KEYWORDS
Root Isolation; System of Real Equations; Certified Computation; Subdivision Algorithms; Miranda Theorem; Moore-Kioustelidis Test; Effective Certified Algorithm; Complexity Analysis

ACM Reference Format:
Juan Xu and Chee Yap. 2019. Effective Subdivision Algorithm for Isolating Zeros of Real Systems of Equations, with Complexity Analysis. In 2019 Int’l Symposium on Symbolic & Algebraic Computation (ISSAC ’19), July 15–18, 2019, Beijing, China. ACM.

1 INTRODUCTION
Solving multivariate zero-dimensional systems of equations is a fundamental task with many applications. We focus on the problem of isolating simple real zeros of a real function

$$f = (f_1, \ldots, f_n): \mathbb{R}^n \rightarrow \mathbb{R}^n$$

within a given bounded box $B_0 \subseteq \mathbb{R}^n$. We do not require $f$ to be polynomial, only each $f_i$ and its partial derivatives have interval forms. We require that $f$ has only isolated simple zeros in $2B_0$, which is the box sharing the same center of $B_0$ and of with twice that of $B_0$. We call $B_0$ the region-of-interest (ROI) of the input instance. This formulation of root isolation is called a local problem in [14], in contrast to the global problem of isolating all roots of $f$. The local problem is very important in higher dimensions because the global problem has complexity that is exponential in $n$. In geometric applications we typically can identify ROI’s and can solve the corresponding local problem much faster than the global problem. Moreover, if $f$ is not polynomial, the global problem might not be solvable: E.g., $f = \sin x, n = 1$. But it is solvable as a local problem as in [29].

In their survey of root finding in polynomial systems, Sherbrooke and Patrikalakis [26] noted 3 main approaches: (1) algebraic techniques, (2) homotopy, (3) subdivision. They objected to the first two approaches on “philosophical grounds”, meaning that it is not easy in these methods to restrict its computation to some ROI $B_0$. Of course, one could solve the global problem and discard solutions that do not lie in $B_0$. But its complexity would not be a function of the roots in $2B_0$. Such local complexity behavior are provable in the univariate case (e.g., [4]), and we will also show similar local complexity in the algorithm of this paper.

Focusing on the subdivision approach, we distinguish two types of subdivision: algebraic and analytic. In algebraic subdivision, $f$ is polynomial and one exploits representations of polynomials such as Bernstein form or B-splines [7, 11, 12, 22, 26]. Analytic subdivision [15, 23, 27] supports a broader class of functions; this is formalized in [29] and includes all the functions obtained from composition of standard elementary functions or hypergeometric functions. Many

---

* Partially supported by the National Natural Science Foundation of China (NSFC 11771034).
† Partially supported by NSF Grants Nos. CCF-1564132; Chinese Academy of Science (Beijing) President’s Fellowship Initiative (2018); Beihang International Visiting Professor Program No. Z2018060; and Special Fund for Guangxi Bagui Scholars WBS 2014-01.

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.

ISSAC ’19, July 15–18, 2019, Beijing, China
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6084-5/19/07...
https://doi.org/10.1145/3326229.3326270

Sometimes, an algorithm is called “local” if it works in small enough neighborhoods (like Newton iteration), and “global” if no such restriction is needed. Clearly, this is a different local/global distinction.
algebraic algorithms come with complexity analysis, while the analytic algorithms typically lack such analysis, unless one views convergence analysis as a weak form of complexity analysis. This lack is natural because many analytic algorithms are what theoretical computer science call “heuristics” with no output guarantees. Any guarantees would be highly conditional (cf. [27]). To our knowledge, the existing subdivision algorithms, both the algebraic ones and the analytic ones, suffer from a gap: they require an input $\varepsilon > 0$ to serve as termination criterion [7, 11, 12, 22, 26]. Without this additional $\varepsilon$, the termination of the algorithms becomes unclear.

1.1 Generic Root Isolation Algorithms

It is useful to formulate a “generic algorithm” for local root isolation (cf. [19]). We postulate 5 abstract modules: three box tests (exclusion $C_0$, existence $EC$, Jacobian $JC$) and two box operators (subdivision and contraction). Our tests (or predicates, which we use interchangeably) are best described using a notation: for any set $B \subseteq \mathbb{R}^n$, $ #(B) = \#_f(B)$ denotes the number of roots, counted with multiplicity, of $f$ in $B$. These tests are abstractly defined by these implications:

$$
\begin{align*}
C_0(B) & \iff #(B) = 0, \\
EC(B) & \iff #(B) \geq 1, \\
JC(B) & \iff #(B) \leq 1.
\end{align*}
$$

Unlike exact predicates, these tests are “one-sided” (cf. [29]) since their failure may have no implications for the negation of the predicate. For root isolation, we need both $EC(B)$ and $JC(B)$ to prove uniqueness. These 3 tests can be instantiated in a variety of ways. The exclusion test $C_0(B)$ is instantiated differently depending on the type of subdivision: exploiting the convex hull property of Bernstein coefficients (in algebraic case) or using interval forms of $f$ (in analytic case). For $EC$, we can use various tests coming from degree theory or fixed point theory (e.g., [3]). This paper is focused on a test based on Miranda’s Theorem. The Jacobian test $JC$ is related to the determinant of the Jacobian matrix but more geometric operators (see [15, 23]). For instance, $[23, \text{Chapter 5}]$ gives detailed sufficient condition for the strong convergence of the operators. But it is still unproven that when a box is sufficiently small, the operators will give a definite result either to exclude the box or to confirm the uniqueness of a root in it. Therefore, an extra $\varepsilon$ is necessary to ensure the termination of the algorithms. But the dependence on $\varepsilon$ naturally results in two issues: the output boxes may not be isolating, i.e., they may contain no root, or more than one roots. In this paper, we present an algorithm that makes up this gap.

1.2 How to derive effective algorithms

In this paper, we describe Miranda, a subdivision algorithm for root isolation, roughly along the above outline. We forgo the use of the contraction operator as it will not Figure in our analysis. For simplicity, assume that all our boxes are hypercubes (equi-dimensional boxes): this means our subdivision splits each box into $2^n$ children. With a little more effort, our analysis can handle boxes with bounded aspect ratios and thus support the bisection-based algorithms. As noted, termination depends on instantiations of our 3 tests: our exclusion and Jacobian tests are standard in the interval

```
\begin{algorithm}
\textbf{Simple Isolate($f, B_0$)}
\begin{algorithmic}
\State \textbf{Output}: sequence of isolating boxes for roots in $B_0$
\State $Q \leftarrow \{B_0\}$
\While {$Q \neq \emptyset$}
\State $B \leftarrow Q$.pop()
\If {$C_0(B)$ continue;}
\State $\triangleleft$ discard $B$ and repeat loop
\EndIf
\If {$EC(B) \land JC(B)$}
\State $\triangleleft$ $B$ has a unique root
\State output $B$ and continue;
\EndIf
\If {$w(N(B) \land B) < w(B)$}
\State $\triangleleft$ if contraction succeeds
\State $Q$.push($B$)
\Else
\State $Q$.push(subdivide($B$))
\EndIf
\EndWhile
\end{algorithmic}
\end{algorithm}
```

Maintains a queue $Q$ of candidate boxes. Initially, $Q$ contains only the ROI $B_0$, the algorithm loops until $Q$ is empty.
literature. Our existence test, called MK test, is from Moore-Kioustadellidis (MK) [20]. Our algorithm is similar\(^3\) to the one in [18, Appendix]. In the normal manner of theoretical algorithms, one would proceed to “prove that \textit{Miranda} is correct and analyze its complexity”. This will be done, but the way we proceed is aimed at some broader issues discussed next.

\textbf{Effectivity:} how could we convert a mathematically precise algorithm (like \textit{Miranda}) into an “effective algorithm”, i.e., certified and implementable. One might be surprised that there is an issue. The non-triviality of this question can be illustrated from the history of isolating univariate roots: for about 30 years, it is known that the “benchmark problem” of isolating all the roots of an integer polynomial with \(L\)-bit coefficients and degree \(n\) has bit-complexity \(O(n^2L)\), a bound informally described as “near-optimal”. This is achieved by the algorithm of Schönhage and Pan (1981-1992). But this algorithm has never been implemented. What has been implemented? Basically, it is the formidable problem of mapping algorithms in the Real RAM model [2] or BSS model [6] into a bit-based Turing-computable model – see [31].

In contrast, recent progress in subdivision algorithms for univariate roots finally succeeded in achieving comparable complexity bounds of \(O(n^3(L+\log n))\), and such algorithms were implemented shortly after! Thus, these subdivision algorithms were “effective”. For two parallel developments of this, see [17, 25] for the case of real roots, and to [4, 5, 14] for complex roots. What is the power conferred by subdivision? We suggest this: \textit{the subdivision framework provides a natural way to control the numerical precision necessary to ensure correct operations of the algorithm}. Moreover, the typical one-sided tests of subdivision avoid the “Zero Problem” and can be effectively implemented using approximations with suitable rounding modes.

In this paper, we capture this pathway to effectivity by introducing 3 Levels of (algorithmic) Abstractions: (A) \textbf{Abstract Level}, (I) \textbf{Interval Level}, and (E) \textbf{Effective Level}. We normally identify Level (A) with the mathematical description of an algorithm or Real RAM algorithms. At level (I), the set extensions of the functions are replaced by the interval forms (see Section 2 for definitions). At the Effective Level, the algorithm approximates real numbers by BigFloat or dyadic numbers, i.e., \(\mathbb{Z}[\frac{1}{2}]\). As illustration, consider the exclusion test \(C_0(B)\) (viewed as abstract) has correspondences in the next three levels:

\begin{align*}
\text{(A): } & C_0(B) \equiv 0 \notin f(B) \\
\text{(I): } & \square C_0(B) \equiv 0 \notin \square f(B) \\
\text{(E): } & C_0(B) \equiv 0 \notin \square f(B)
\end{align*}

where \(f(B)\) is the exact range of \(f\) on \(B\), \(\square f(B)\) is the interval form of \(f\), and \(\square f(B)\) is the effective form where the endpoints are dyadic numbers. The 3 range functions here are related as \(f(B) \subseteq \square f(B) \subseteq \square f(B)\).

An abstract algorithm \(A\) is first mapped into an interval algorithm \(\square A\). But the algorithm still involves real numbers. So we must map \(\square A\) to an effective algorithm \(\square A\). Correctness must ultimately be shown at the Effective Level; the standard missing link in numerical (even “certified”) algorithms is that one often stops at Abstract or Interval Levels.

We need to mention that the effectivity of an algorithm has no implications for the efficiency of the algorithm.

\textbf{Complexity:} The complexity of analytic algorithms is often restricted to convergence analysis. But in this paper, we will provide bounds on complexity as a function of the geometry of the roots in \(2B_0\). This complexity can be captured at each of our 3 levels, but we always begin by proving our theorems at the Abstract Level, subsequently transferred to the other levels. Although it is the Effective Level that really matters, it would be a mistake to directly attempt such an analysis at the Effective level: that would obscure the underlying mathematical ideas, incomprehensible and error prone. The 3-level description enforces an orderly introduction of new concerns appropriate to each level. Like structured programming, the design of effective algorithms needs some structure. Currently, outside of the subdivision framework, it is hard to see a similar path way to effectivity.

\subsection{1.3 Literature Survey}

There is considerable literature associated with each of our three tests: the exclusion test comes down to bounding range of functions, a central topic in Interval Analysis [24]. The Jacobian test is connected to the question of local injectivity of functions, the Bieberbach conjecture (or de Branges Theorem), Jacobian Conjecture, and theory of univalent functions. In our limited space, we focus on the “star” of our 3 tests, i.e., the existence test. It is the most sophisticated of the 3 tests in the sense that some nontrivial global/topological principle is always involved in existence proofs. In our case, the underlying principle is the fixed point theorem of Brouwer, in the form of Miranda’s Theorem (1940), and intimately related to degree theory.

We compare two box tests \(C\) and \(C'\) in terms of their relative \textbf{efficacy}: say \(C\) is \underline{as efficacious as} \(C'\), written \(C \succeq C'\), if for all \(B\), \(C'(B)\) succeeds implies that \(C(B)\) succeeds. The relative efficacy of several existence tests have been studied [3, 9, 10, 13]. Goldststein considers four common existence tests, and argues that “in practice” there is an efficacy hierarchy

\begin{equation}
(IN) \succeq (HS) \succeq (FLS) \succeq (K)
\end{equation}

where (K) refers to Krawczyk, (HS) to Hansen-Sengupta, (FLS) to Frommer-Lang-Schnurr, and (IN) to Interval-Newton. Note that (K), (HS) and (IN) are all based on Newton-type operators (see (2)). Our Moore-Kioustadellidis (MK) test is essentially (FLS). We say “essentially” because the details of defining the tests may vary to render the comparisons invalid. In our MK tests, we evaluate \(f\) on each box face using the Mean Value Form expansion at the center of the face. But the above analysis assumes that the expansion is at the center of the box, which is less accurate. But we may also compare these tests in terms of their complexity (measured by the worst case number of arithmetic operations, or number of function evaluations); a complexity-efficacy tradeoff may be
expected. Finally, evaluating these tests in isolation does not tell us how they might perform in the context of an algorithm. It is therefore premature to decide on the best existence test.

1.4 Overview
In section 2, we introduce some basic concepts of interval arithmetic and establish notations. Section 3 introduces the key existence test based on Miranda’s theorem. Section 4 proves conditions that ensure the success of these existence test. Section 5 introduces two Jacobian tests. Section 6 describes our main algorithm. Section 7 is the complexity analysis of our algorithm. Section 8 discusses about the effective version of the algorithm. We conclude in Section 9. All proofs can be found in [28].

2 INTERVAL FORMS
We first establish notations for standard concepts of interval arithmetic. Bold fonts indicate vector variables: e.g., \( \mathbf{f} = (f_1, \ldots, f_n) \) or \( \mathbf{x} = (x_1, \ldots, x_n) \).

Let \( \mathbb{R} \) denote the set of compact intervals in \( \mathbb{R} \). Extend this to \( \mathbb{R}^n \) for the set of compact \( n \)-boxes. In the remaining paper, we assume that all \( n \)-boxes are hypercubes (i.e., the width in each dimension is the same). For any box \( B \in \mathbb{R}^n \), let \( m_B = m(B) \) denote its center and \( w_B = w(B) \) be the width of any dimension. Besides boxes, we will also use ball geometry: let \( \Delta = \Delta(a, r) \subseteq \mathbb{R}^n \) denote the closed ball centered at \( a \in \mathbb{R}^n \) of radius \( r > 0 \). If \( r \leq 0 \), \( \Delta(a, r) \) is just the point \( a \). For any \( k > 0 \), let \( kB \) denote the box centered at \( m(B) \) of width \( k \cdot w(B) \), called the \( k \)-dilation of \( B \). The \( k \)-dilation \( k\Delta \) of \( \Delta \) is defined likewise.

Let \( A, B \subseteq \mathbb{R}^n \) be two sets. We will quantify their “distance apart” in two ways: their usual Hausdorff distance is denoted \( d(A, B) \) and their separation, \( \inf \{ ||a - b|| : a \in A, b \in B \} \) is denoted as \( dsep(A, B) \). Note that \( d \) is a metric on closed subsets of \( \mathbb{R}^n \) but \( dsep(A, B) \) is not a metric.

Consider two kinds of extensions of a function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \). First, the set extension of \( f \) refers to the function (still denoted by \( f \)) that maps \( S \subseteq \mathbb{R}^n \) to \( f(S) := \{ f(x) : x \in S \} \). The second kind of extension is not unique: an interval form of \( f \) is any function \( \square f : \mathbb{R}^n \rightarrow \mathbb{R} \). Note that \( \square f \) is a metric on closed subsets of \( \mathbb{R}^n \) but \( dsep(A, B) \) is not a metric.

The notation \( \square f \) is generic; we use subscripts to indicate specific interval forms. Thus, the mean value form of \( f \) is

\[
\square_{mf}(f)(B) = f(m(B)) + \nabla f(B) \cdot (B - m(B))
\]

where \( \nabla f \) is the gradient of \( f \). The box \( B - m(B) \) is now centered at the origin, i.e., \( m(B - m(B)) = 0 \). The appearance of the generic \( \nabla f(B) \) in the definition of \( \square_{mf} \) means that \( \square f \) is still not fully specified. In our complexity analysis, we assume that for any interval form, if not fully specified, will have at least linear convergence. Next, we intend to convert the interval form \( \square_m \) to some effective version \( \square_M \).

3 MIRANDA AND MK TESTS
In the rest of this paper, we fix

\[
f := (f_1, \ldots, f_n) : \mathbb{R}^n \rightarrow \mathbb{R}^n
\]

to be a \( C^2 \)-function (twice continuously differentiable), and \( f \) and its partial derivatives have interval forms. We further postulate that \( f \) has only finitely many simple zeros in \( 2B_0 \) where \( B_0 \) is the bounded region of interest. A zero \( \alpha \) of \( f \) is simple if the Jacobian matrix \( J_f(\alpha) \) is non-singular. For any set \( S \subseteq \mathbb{R}^n \), let \( \text{Zero}(S) \) denote the multiset of zeros of \( f \) in \( S \). We assume that \( f \) is analytic and its zeros are counted with the proper multiplicity. Then \#(S) is the size of the multiset \( \text{Zero}(S) \). We may write \( \text{Zero}(S) \) and \#(S) when \( f \) is understood.

We consider a classical test from Miranda (1940) to confirm that a box \( B \in \mathbb{R}^n \) contains a zero of \( f \). If the box \( B \) is written as \( B = \prod_{i=1}^n I_i \) with \( I_i = [a_i, a_i^+] \), then it has two \( i \)-th faces, namely

\[
B_i^- := I_1 \times \cdots \times I_{i-1} \times [a_i^-, a_i^+] \times I_{i+1} \times \cdots \times I_n
\]

and \( B_i^+ \), defined similarly. Write \( B_i^{\pm} \) to mean either \( B_i^- \) or \( B_i^+ \). Consider the following box predicate called the simple Miranda Test:

\[
\text{MT}_f(B) \equiv \bigcap_{i=1}^n (f_i(B_i^+) > 0) \land (f_i(B_i^-) < 0)
\]

where \( f \) is given in (4). The following result is classic:

**Proposition 1.** [Miranda (1940)]

If \( \text{MT}_f(B) \) holds then \#(B) \geq 1.

Next, we introduce the MK Test \( \text{MK}(B) = \text{MT}_f(B) \) that amounts an application of the simple Miranda test to the box \( 2B \), using a preconditioned form of \( f \):

---

**Abstract MK Test**

**Input:** \( f \) and box \( B \)

**Output:** \text{true} iff \( \text{MK}(B) \) succeeds

1. Take a point \( m \in B \) with \( J_f^{-1}(m) \) well-defined.
2. Construct a “preconditioned version” \( g \):

\[
g 
\leftarrow J_f^{-1}(m)f = (g_1(x), \ldots, g_n(x))
\]

3. Apply the Simple Miranda Test to \( g \) over \( 2B \):

For \( i = 1, \ldots, n \):

\[
\text{If } g_i(2B_i^+) \leq 0 \text{ or } g_i(2B_i^-) \geq 0, \text{ (*) return false}
\]

4. Return true.

---

The notation “\( 2B_i^{\pm} \)” in (*) refers to faces of the box \( 2B \), not the 2-dilation of the faces of \( B \). Here “MK” refers to Moore and Kious (2010); the preconditioning idea first appearing in [16]. The MK Test was first introduced in [18]. Notice that in \( \text{MK}(B) \), the Miranda test is performed on \( 2B \) instead of \( B \). It is intended to address the difficult case where the root is close to the boundary of a box.

\[\text{\footnote{We call it “simple” as we ignore some common generalizations that allow an interchange of ”< 0” with ”> 0”, or replace } f \text{ by } \sigma(f) = (f_{\sigma(1)}, \ldots, f_{\sigma(n)}) \text{ for any arbitrary permutation } \sigma \text{ of the indices.}}\]
Note that \( \mathbb{M}(B) \) is mathematically exact and generally not implementable (even if it were possible, we may still prefer approximations). We first define its interval form, denoted \( \mathbb{M}(B) \); simply by replacing \( g_k(2B^*_i) \) in line (*) by interval forms \( \mathbb{D}_g(2B^*_i) \). Finally, we must define the effective form \( \mathbb{M}(B) \) (in Section 8).

4 ON SURE SUCCESS OF MK TEST

The success of the MK test implies the existence of roots. In this section, we show that the MK test succeeds on any box containing a root and of size smaller than a specific bound.

We need preliminary facts about mean value forms. Given \( x, y \in \mathbb{R}, \) the notation \( x \pm y \) denotes a number of the form \( x + \theta y \), where \( 0 \leq |\theta| \leq 1 \); thus \( "\pm" \) hides the implicit \( \theta \) in the definition. This notation is not symmetric: \( x \pm y \) and \( y \pm x \) are generally different. This notation extends to matrices: let \( A = (a_{ij})_{i,j=1}^n \) and \( B = (b_{ij})_{i,j=1}^n \) be two matrices, then \( A \pm B := (a_{ij} \pm b_{ij})_{i,j=1}^n \). Also, let \( |x| \) denote the vector \((|x|_1, \ldots, |x|_n)\) where \( x = (x_1, \ldots, x_n) \). For \( x, y \in \mathbb{R}^n \), write \( [x, y] \) to denote the line segment connecting \( x \) and \( y \). We write \( \|x\| \) and \( \|A\| \) for the infinity norms of vector \( x \) and matrix \( A \). The magnitude of any bounded set \( S \subseteq \mathbb{R} \) is defined as \( |S| := \sup \{|x| : x \in S\} \). For a bounded convex set \( C \subseteq \mathbb{R}^n \), define the matrix \( K(C) \) with entries \((K(C))_{ij}\) where

\[
K(C)_{ij} := \sum_{k=1}^n \left| \frac{\partial^2 f}{\partial x_i \partial x_k} (C) \right|, (6)
\]

Below, \( C \) may be a disc \( \Delta \) or a line \([x, y]\). Denote by \( J_f(x) \) the Jacobian matrix of \( f \) at \( x \). We write \( J_f(x) \) as \( J(x) \) when \( f \) is understood. The following is a simple application of the Mean Value Theorem (MVT):

**Lemma 2 (MVT).** Given two points \( x, y \in \mathbb{R}^n \), we have:

\[
\begin{align*}
& (a) \quad J(x) = J(y) \pm K([x, y])\|x - y\|, \\
& (b) \quad f(x) - f(y) = J(y) \pm K([x, y])\|x - y\| \cdot (x - y).
\end{align*}
\]

4.1 Sure Success of abstract MK Test

In this and the next subsection, we consider boxes that contain a root \( \alpha \) of \( f \). We prove conditions that ensures the success of the MK Test. We first prove this for the abstract test \( \mathbb{M}(B) \). The next section extends this result to the interval test \( \mathbb{M}(B) \).

The key definition here is a bound \( \lambda_1(\alpha) \) which depends on \( \alpha \) and \( f \). We prove that if \( w(B) \leq \lambda_1(\alpha) \), then the abstract MK test will succeed on \( B \). By a critical point we mean \( \alpha \in \mathbb{R}^n \) where the determinant of \( J(\alpha) \) is zero. By definition, a root \( \alpha \) of \( f \) is simple if \( \alpha \) is not a critical point.

Suppose \( S_1 \) and \( S_2 \) are two bounded sets in \( \mathbb{R}^n \). Define

\[
\|J^{-1}(S_1)\| := \sup_{x \in S_1} \|J^{-1}(x)\| \quad \text{and} \quad \|J^{-1}(S_1) \cdot K(S_2)\| := \sup_{x \in S_1, y \in S_2} \|J^{-1}(x) \cdot K(y)\|.
\]

We see that both \( \|J^{-1}(S_1)\| \) and \( \|J^{-1}(S_1) \cdot K(S_2)\| \) are finite if \( S_1 \) does not contain a critical point of \( f \). Consider the following function

\[
s(r) := r - \frac{1}{27n\|J^{-1}(\Delta(\alpha, 2\sqrt{n}r)) \cdot K(\Delta(\alpha, 2\sqrt{n}r))\|}. (7)
\]

We then define \( \lambda_1(\alpha) \) to be the smallest \( r \) such that \( s(r) = 0 \), i.e., \( \lambda_1(\alpha) := \arg \min \{s(r) : s(r) = 0\} \).

**Lemma 3.** For any simple root \( \alpha \) of \( f \), \( \lambda_1(\alpha) \) is well-defined.

From now on, let \( \Delta_\alpha \) denote the disc

\[
\Delta_\alpha := \Delta(\alpha, 2\sqrt{n}\lambda_1(\alpha)). \tag{8}
\]

The following lemma corrects a gap in the appendix of [18].

**Lemma 4.** Let \( B \) be a box containing a simple root \( \alpha \) of \( f \) and \( m \in B \) with \( J^{-1}(m) \) well-defined. If \( w_B \leq \lambda_1(\alpha) \), the preconditioned system \( g_B := J^{-1}(m)f = (g_1, \ldots, g_n) \) satisfies that for all \( i = 1, \ldots, n \),

\[
g_i(2B^*_i) \geq \frac{w_B}{4}, \quad g_i(2B^*_i) \leq -\frac{w_B}{4}.
\]

4.2 Sure Success of Interval MK Test

We now extend the previous subsection on the abstract MK Test \( \mathbb{M}(B) \) to the interval version \( \mathbb{M}(B) \) where the set extensions in the sign test (*) are replaced by the interval forms. Again, assume \( B \) is a box containing exactly one root \( \alpha \) of \( f \). We will give \( \lambda_2(\alpha) \) which is analogous to \( \lambda_1(\alpha) \) and prove that if \( w_B \leq \lambda_2(\alpha) \), then \( \mathbb{M}(B) \) will succeed.

To prove the existence of such a \( \lambda_2(\alpha) \) as mentioned above, we need to make some assumptions on the properties of the box functions. As in [21], a box function \( \mathbb{D}f \) is called Lipschitz in a region \( S \subseteq \mathbb{R}^n \) if there exists a constant \( L \) such that

\[
w(\mathbb{D}f(B)) \leq L \cdot w(B), \quad \forall B \subseteq S. \tag{9}
\]

We call any such \( L \) a Lipschitz constant of \( \mathbb{D}f \) on \( S \). For our theorem, we need to know the specific box function in order to derive a Lipschitz constant. Consider the mean value form \( \mathbb{D}f \) on a region \( S \subseteq \mathbb{R}^n \).

**Lemma 5.** Let \( f \) be a continuously differentiable function defined on a convex region \( S \subseteq \mathbb{R}^n \). Then \( \sum_{k=1}^n \left| \mathbb{D} \frac{\partial f}{\partial x_k}(S) \right| \) is a Lipschitz constant for \( \mathbb{D}f \) on \( S \).

For the sign tests of \( \mathbb{M}(B) \):

\[
\mathbb{D}g_i(2B^*_i) > 0 \quad \text{and} \quad \mathbb{D}g_i(2B^*_i) < 0
\]

where \( g_i \) is the \( i \)-th component of the system \( J(m)^{-1}f \). We use the mean value form \( \mathbb{D}g_i(2B^*_i) = g_i(m(2B^*_i)) + \mathbb{D}g_i(2B^*_i) \cdot (m(2B^*_i) - 2B^*_i) \) and assume that the components of \( \mathbb{D}g_i(2B^*_i) \) are evaluated via the linear combination of \( \mathbb{D}g_j(2B^*_i) \) for \( j, k = 1, \ldots, n \).

We now prove that if \( B \) is small enough, \( \mathbb{M}(B) \) will succeed. Recalling the Hausdorff distance \( q(I, J) \) on intervals, we have this bound from [23].

**Proposition 6.** Let \( f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R} \) be a continuously differentiable function. Then

\[
q(\mathbb{D}f, f(B)) \leq 2w_B \sum_{i=1}^n w(\mathbb{D} \frac{\partial f}{\partial x_i}). \tag{10}
\]

For the next theorem, define

\[
\hat{\lambda}_1(\alpha) := \frac{1}{64n^2L_\alpha \cdot \|J^{-1}(\Delta_\alpha)\|}. \tag{11}
\]
where \( L_\alpha \) is a Lipschitz constant for \( \frac{\partial f_k}{\partial x_\alpha} \) on \( \Delta_\alpha \) (for all \( j,k = 1, \ldots, n \)).

**Theorem 7.** Let \( B \) be a box containing a simple root \( \alpha \) of width \( w_B \leq \lambda_1(\alpha) \) and \( \mathbf{m} \in B \) with \( J^{-1}(\mathbf{m}) \) well-defined.

(a) If \( w(\frac{\partial g_j}{\partial x_\alpha})^{(2B)} \leq \frac{1}{2w_B} \) for each \( j = 1, \ldots, n \), then \( \Box\tilde{MK}(B) \) succeeds with \( g_B := J^{-1}(\mathbf{m})f \).

(b) If \( w_B \leq \lambda_2(\alpha) \) with \( \lambda_2(\alpha) := \min \{ \lambda_1(\alpha), \tilde{\lambda}_1(\alpha) \} \), then \( \Box\tilde{MK}(B) \) succeeds.

5 TWO JACOBIAN CONDITIONS

We define the **Jacobi test** as follows:

\[
\BoxJC(B) \equiv 0 \notin \det(J_f(3B)).
\]  

The order of operations in \( \det(J_f(3B)) \) should be clearly understood: first we compute the interval Jacobian matrix \( J_f(3B) \), i.e., entries in this matrix are the intervals \( \frac{\partial f_k}{\partial x_\alpha} \) in \( 3B \). Then we compute the determinant of the interval matrix. Also note that we use \( 3B \) instead of \( B \). The following is well-known in interval computation (see [1, Corollary to Theorem 12.1]):

**Proposition 8.** * Jacobi test

If \( \BoxJC(B) \) holds then \( \#_f(3B) \leq 1 \).

We next introduce the following **strict Jacobi test**: \[
\BoxJC_s(B) \equiv 0 \notin \boxed{(\det J_f)(3B)} \tag{12}
\]

where \( (\det J_f)(x) \) denotes the expression obtained by computing the determinant of the Jacobian matrix \( J_f(x) \) with functional entries \( \frac{\partial f_k}{\partial x_\alpha}(x) \). Finally, we evaluate \( (\det J_f)(x) \) on \( 3B \). Note that \( \BoxJC_s(B) \Rightarrow \BoxJC(B) \) and so the strict test is more efficacious. Unfortunately, it is known that \( \BoxJC_s(B) \) does not imply \( \#_f(3B) \leq 1 \). Nevertheless, we now show that it can serve as a uniqueness test in conjunction with the MK test:

**Theorem 9.**

If both \( \BoxJC_s(B) \) and \( \BoxMK(\frac{3}{2}B) \) succeed then \( \#_f(3B) = 1 \).

6 THE MIRANDA ALGORITHM

Our main algorithm for root isolation is given in Figure 1. We use \( \BoxMK(B) \) and \( \BoxJC(B) \) (respectively) for its existence and Jacobian tests. It remains to specify the exclusion test \( \BoxCo(B) \):

\[
\BoxCo(B) \equiv (\exists i = 1, \ldots, n)[0 \notin f_i(B)] \tag{14}
\]

The algorithm in Figure 1 is abstract. To introduce the interval version \( \Box\tilde{Miranda} \), just replace the abstract tests by their interval analogues: \( \Box\tilde{MK}(B), \Box\tilde{Co}(B) \) and \( \BoxJC(B) \). It amounts to replacing the set theoretic function in the abstract definition by their interval analogues:

- \( \Box\tilde{Co}(B) \): \( \exists i = 1, \ldots, n \) such that \( 0 \notin f_i(B) \);
- \( \Box\tilde{JC}(B) \): \( 0 \notin \boxed{\det(J_f(3B))} \);
- In the definition of \( \Box\tilde{MK}(B) \) (Section 3), replace each \( g_i(2B)^\pm \) by \( g_i(2B)^\pm \).

For the effective version, we use the tests \( \Box\tilde{MK}(B), \Box\tilde{Co}(B) \) and \( \Box\tilde{JC}(B) \), which will be discussed in Section 8.

**Abstract Miranda**

**Output:** Queue \( P \) of non-overlapping isolating boxes of \( f \) s.t.

\[
\BoxZero_f(B_0) \subseteq \bigcup_{B \in P} \BoxZero_f(B) \subseteq \BoxZero_f(2B_0)
\]

1. Initialize output queue \( P \leftarrow \emptyset \) and priority queue \( Q \leftarrow \{B_0\} \).
2. While \( Q \neq \emptyset \) do:
   3. Remove a biggest box \( B \) from \( Q \).
   4. If \( \BoxCo(B) \) succeeds, continue;
   5. If \( \BoxJC(B) \) succeeds then
      6. Initialize new queue \( Q' \leftarrow \{B\} \).
      7. While \( Q' \neq \emptyset \) do:
         8. \( B' \leftarrow Q'.pop() \).
         9. If \( \BoxCo(B') \) fails then
            10. If \( \BoxMK(B') \) succeeds then
               11. \( P.add(2B') \).
               12. Discard from \( Q \) the boxes contained in \( 3B \).
               13. Break
               14. \( Q.push(subdivide(B)) \).
            Else
               15. \( Q.push(subdivide(B)) \).

**Figure 1:** Root Isolation Algorithm

Termination of each version of Miranda follows from the complexity analysis below. We first show the partial correctness:

**Theorem 10 (Partial Correctness).** If \( \tilde{Miranda} \) halts, the output queue \( P \) is correct.

7 COMPLEXITY UPPER BOUNDS

In this section, we derive a lower bound \( \lambda > 0 \) on the size of boxes produced by Miranda. That is, any box \( B \) with width \( w(B) \leq \lambda \) will either be output or rejected. This implies that the subdivision tree is no deeper than \( \log_2(w(B)/\lambda) \), yielding an upper bound on computational complexity. This bound \( \lambda \) will be expressed in terms of quantities determined by the zeros in \( 2B_0 \). We first prove this for the abstract Miranda, then extend the results to \( \tilde{Miranda} \) and \( \tilde{Miranda} \). From the algorithm, we see that a box \( B \) is output if \( \sim \BoxCo(B) \land \BoxJC(B) \land \BoxMK(B) \) holds in line 10; it is rejected if one of the 2 following cases is true: (1) \( \BoxCo(B) \) holds or (2) \( B \) is contained in \( 3B' \) where \( B' \) is an output box, as indicated in line 12. The boxes that contain a root of \( f \) will finally be verified by the former predicate and the boxes that contain no root of \( f \) will eventually be rejected in one of the 2 cases.

To prove the existence of such a \( \lambda \), we need to look into the tests \( \BoxCo(B), \BoxJC(B) \) and \( \BoxMK(B) \). We will give bounds \( \lambda_{\tilde{MK}}, \lambda_{\tilde{JC}} \) and \( \lambda_{\tilde{Co}} \) for the 3 tests respectively and show that for any box \( B \) produced in the algorithm

1. If \( \#(B) > 0 \) it will pass \( \BoxMK(B) \) when \( w_B \leq \lambda_{\tilde{MK}} \);
2. If \( \#(B) \leq 1 \), it will pass \( \BoxJC(B) \) when \( w_B \leq \lambda_{\tilde{JC}} \);
3. If \( \#(B) = 0 \), there are 2 cases: (a) if \( B \) keeps a certain distance from the roots, it passes \( \BoxCo(B) \) when \( w_B \leq \lambda_{\tilde{Co}} \); (b) if \( B \) is close enough to the roots, it will be rejected by line 12 of the algorithm when \( w_B \leq \frac{1}{4} \min \{ \lambda_{\tilde{MK}}, \lambda_{\tilde{JC}}, \lambda_{\tilde{Co}} \} \).
We have essentially proved item (1) in the Section 4. More precisely, for each root \( \alpha \), we had defined a constant \( \lambda_2(\alpha) \) in Theorem 7. We now set

\[
\lambda_\infty := \min_{\alpha \in \text{Zero}(2B_0)} \lambda_2(\alpha). \tag{15}
\]

### 7.1 Sure Success for \( C_0(B) \) and \( JC(B) \)

We study conditions to ensure the success of the tests \( JC \) and \( C_0 \). We will introduce constants \( \lambda_\infty, \lambda_0 \) in analogy to (15).

First consider \( JC(B) \). Let box \( B \) contain a simple root \( \alpha \). By Mean Value Theorem, \( w(\frac{\partial f}{\partial x_j}(3B)) \leq 3w_B \cdot K(3B)_{i,j} \) (see (6) for definition of \( K(3B)_{i,j} \)). Since \( \alpha \in 3B \), it holds \( \frac{\partial f}{\partial x_j}(3B) \leq \frac{\partial f}{\partial x_j}(\alpha) + 3w_B \cdot K(3B)_{i,j} \) for all \( j = 1, \ldots, n \). Denoting \( U := \max_{1 \leq i, j \leq n} \left| \frac{\partial f}{\partial x_j}(\alpha) \right| \) and \( V := \max_{1 \leq i, j \leq n} K(3B)_{i,j} \), we get \( \frac{\partial f}{\partial x_j}(3B)) \leq U + 3v_B \cdot w_B \) and \( w(\frac{\partial f}{\partial x_j}(3B)) \leq 3v_Bw_B \). By applying the rules \( w(I_1 + I_2) = w(I_1) + w(I_2) \) and \( w(I_1 \cdot I_2) \leq w(I_1) \cdot |I_2| + w(I_2) \cdot |I_1| \) where \( I_1, I_2 \) are intervals, we may verify by induction that \( w(\prod_{i=1}^n \left( \frac{\partial f}{\partial x_j}(3B) \right)) \leq 3^n V(1 + 3v_BV)^{n-1} w_B \) for any permutation \( \sigma \). Hence, it follows \( w(\det(J_f(3B))) \leq 3^n \cdot n! \cdot V(1 + 3v_BV)^{n-1} w_B \).

Set \( \lambda_3(\alpha) \) to be the smallest positive root of the equation

\[
|\det(J_f(\alpha))| - 3n \cdot n! \cdot V(1 + 3v_BV)^{n-1} \cdot x = 0. \tag{16}
\]

The following lemma implies the existence of \( \lambda_3(\alpha) \):

**Lemma 11.** If box \( B \) contains a simple root \( \alpha \) and \( w_B < \lambda_3(\alpha) \) then \( JC(B) \) succeeds.

Thus we may choose \( \lambda_\infty := \min_{\alpha \in \text{Zero}(2B_0)} \lambda_3(\alpha) \) and set \( \ell_1 := \min \{ \lambda_\infty, \lambda_0 \} \).

**Lemma 12 (Lemma A).** If \( \#(B) > 0 \) and \( w_B \leq \ell_1 \) then \( MK(B) \) and \( JC(B) \) holds.

**Corollary 13.** Each root in \( B_0 \) will be output in a box of width \( > 3\ell_1/2 \).

Let \( R_0 := 2B_0 \) be a region that excludes discs around roots:

\[ R_0 := 2B_0 \setminus \bigcup_{\alpha \in \text{Zero}(2B_0)} \Delta(\alpha, \ell_1) \]

where \( \Delta \) is the interior of \( \Delta \). Define the zero set of \( f_i \) as \( S_i \) for \( i = 1, \ldots, n \) and define \( d_0 := \inf_{p \in R_0} \max_{i=1}^n \text{sep}(p, S_i) \). Since all the roots in \( 2B_0 \) are removed from the set \( R_0 \), we can verify that \( \max_{i=1}^n \text{sep}(p, S_i) > 0 \) for all \( p \in R_0 \). Combining with the compactness of \( R_0 \), we obtain \( d_0 > 0 \). Finally we set

\[
\lambda_0 := \frac{d_0}{2\sqrt{m}}. \tag{17}
\]

**Lemma 14 (Lemma B).** Suppose \( \#(B) = 0 \). If \( \text{sep}(m_B, \text{Zero}(2B_0)) \geq \ell_1 \), \( C_0(B) \) succeeds when \( w_B \leq \lambda_0 \). If \( \text{sep}(m_B, \text{Zero}(2B_0)) < \ell_1 \), \( C_0(B) \) succeeds when \( w_B \leq \frac{1}{2} \min \{ \lambda_0, \ell_1 \} \).

The Lemma follows naturally from Lemma A and B:

**Lemma 15 (Lemma C).** Every box produced by the \textit{Miranda} has width \( \geq \frac{1}{2} \min \{ \lambda_0, \lambda_\infty, \lambda_0 \} \).

### 7.2 Sure Success for \( \square C_0(B) \) and \( J^C(B) \)

We now consider the interval tests \( J^C \) and \( C_0 \) under the assumption that the underlying interval forms involved are Lipschitz. Let \( \Lambda \) be a global Lipschitz constant for \( \square f_i \) and \( \square \frac{\partial f_i}{\partial x_j} \) for all \( i, j = 1, \ldots, n \) in \( 3B_0 \). We will develop corresponding bounds \( \lambda_{\square f_i}, \lambda_{\square \frac{\partial f_i}{\partial x_j}} \). Observe that if we replace the bounds \( \lambda_{\square f_i}, \lambda_{\square \frac{\partial f_i}{\partial x_j}} \) in the abstract version by the bounds \( \lambda_{\square f_i}, \lambda_{\square \frac{\partial f_i}{\partial x_j}} \), all the statements and proofs in the previous section remain valid. So in this section, we do not repeat the statements, but only give the bounds \( \lambda_{\square f_i} \) and \( \lambda_{\square \frac{\partial f_i}{\partial x_j}} \).

First look at the test \( \squareJC(B) \). With the same arguments as in abstract level, we obtain

\[
\lambda_{\squareJC} := \min_{\alpha \in \text{Zero}(2B_0)} \lambda_3(\alpha)
\]

where \( \lambda_3(\alpha) \) is the smallest positive root of the equation

\[
|\det(J_f(\alpha))| - 3n \cdot n! \cdot |J(3B)|^{n-1} \cdot x = 0. \tag{18}
\]

With \( \lambda_{\square f_i} \) and \( \lambda_{\square \frac{\partial f_i}{\partial x_j}} \) we get an interval analogue of Lemma A:

**Lemma 16 (Lemma A).** If \( \#(B) > 0 \) and \( w_B \leq \ell_1 \) with \( \ell_1 := \min \{ \lambda_{\square f_i}, \lambda_{\square \frac{\partial f_i}{\partial x_j}} \} \), then \( \squareJC(B) \) succeeds. Next look at the test \( \square C_0(B) \). Arguing as in the abstract level, we only consider the boxes in the region \( R_0 := 2B_0 \setminus \bigcup_{\alpha \in \text{Zero}(2B_0)} \Delta(\alpha, \ell_1) \).

**Lemma 17 (Lemma B).** Suppose \( \#(B) = 0 \). Let \( \ell_1 := \min \{ \lambda_{\square f_i}, \lambda_{\square \frac{\partial f_i}{\partial x_j}} \} \). If \( \text{sep}(m_B, \text{Zero}(2B_0)) \geq \ell_1 \), then \( \square C_0(B) \) succeeds when \( w_B \leq \lambda_{\square f_i} \). If \( \text{sep}(m_B, \text{Zero}(2B_0)) < \ell_1 \), then \( \square C_0(B) \) succeeds when \( w_B \leq \frac{1}{2} \min \{ \lambda_{\square f_i}, \ell_1 \} \).

Combining Lemma A and Lemma B, we obtain:

**Lemma 18 (Lemma C).** Every box produced by the \textit{Miranda} has width \( \geq \frac{1}{2} \min \{ \lambda_{\square f_i}, \lambda_{\squareJC} \} \).

### 8 EFFECTIVE MIRANDA

We now extend our results from \( \squareMK(B) \) and \( \squareJC(B) \) to \( \squareMiranda(B) \) by introducing the effective tests \( \squareMK(B) \) and \( \squareJC(B) \) and \( \squareMiranda(B) \).

Recall that the difference between the effective algorithms and the interval ones is that the former uses dyadic numbers instead of real numbers. In \( \squareMiranda \), this difference is reflected in 2 places: (1) The preconditioning matrix \( J^{-1}(m) \) in \( \squareMK(B) \) is approximated by \( \tilde{J}^{-1}(m) \) with dyadic entries (2) each interval form \( h(B) \) (including \( \delta_i h(B) \)) is outwardly rounded to the effective form \( \delta_i h(B) \) with dyadic endpoints.

The main issue in \( \squareMiranda \) is the accuracy of \( J^{-1}(m) \) and \( h(B) \). Based on the following requirements, we claim that similar lower bounds as in Section 7 for \( \squareMiranda \) is still achievable. We require

\[
\| \tilde{J}^{-1}(m) - J^{-1}(m) \| \leq \frac{1}{12n^2 \cdot \|J(m)\|}. \tag{R1}
\]
\[ q(B(B)) \leq \frac{1}{32} w_B. \]  

More precisely, for (R2) we require that in \( \Delta \), \( q(f_i(B)) \leq \frac{1}{32} w_B \) for \( f_i \in [1, n] \); in \( \Delta \), \( q(g_i(B)) \leq \frac{1}{32} w_B \) for \( g_i \in [1, n] \); in \( \Delta \), \( q(J_i(B)) \leq \frac{1}{32} w_B \) for each entry \( J_i(B) \) of \( J(B) \).

It is easy to verify that, for \( \Delta \), \( \Delta \), we can get similar bound as \( \Delta \) and \( \Delta \) by simply replacing \( L \) with \( L + \frac{1}{4} \). For \( \Delta \), similar bound as \( \Delta \) can also be obtained by modified the definition of \( \lambda_1(\alpha) \).

9 CONCLUSION

We have provided the first effective subdivision algorithm \textbf{Miranda} for isolating simple real roots of a system of equations \( f = 0 \), provided \( f \) and its derivatives have interval forms. Our result are novel for its completeness (previous algorithms need \( \varepsilon \)-termination and has no isolation guarantees), its generality (going beyond the polynomial case), and its complexity analysis (going beyond termination proofs). We also contributed to the theory of subdivision algorithms by formalizing a 3-level description to provide a pathway from abstract algorithms to effective ones. Given that many existing numerical algorithms still lack effective versions, this is a promising line of work. In the future, we plan to implement and develop our algorithm into a practical tool.

ACKNOWLEDGMENTS

The first author wishes to thank her supervisor Professor Dongming Wang for his support and encouragement. The second author is thankful for hospitality and support during a sabbatical leave: Professor Jing Yang (Guangxi University of Nationalities, Nanning); Drs. Xiaoshan Gao and Jiansen Cheng (Chinese Academy of Sciences, Beijing); Professors Dongming Wang and Chenqi Mou (Beihang University).

REFERENCES

[1] Olivier Aberth. \textit{Introduction to Precise Numerical Methods}. Elsevier Inc, Oxford, UK, second edition, 2007.
[2] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. \textit{The Design and Analysis of Computer Algorithms}. Addison-Wesley, Reading, Massachusetts, 1974.
[3] Götz Alefeld, Andreas Frommer, Gerhard Heindl, and Jan Mayer. On the existence theorems of Kantorovich, Miranda and Borsuk. \textit{Electronic Transactions on Numerical Analysis}, 17:102–111, 2004.
[4] Ruben Becker, Michael Sagraloff, Vikram Sharma, Juan Xu, and Chee Yap. Complexity analysis of root finding for a complex polynomial. In 41st \textit{Int'1 Symp. Symbolic and Alge. Comp.}, pages 71–78, 2016. \textit{ISSAC 2016}. July 20–22, Wilfrid Laurier University, Waterloo, Canada.
[5] Ruben Becker, Michael Sagraloff, Vikram Sharma, and Chee Yap. A near-optimal subdivision algorithm for complex root isolation based on Pellet test and Newton iteration. \textit{J. Symbolic Computation}, 86:51–96, May–June 2018.
[6] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. \textit{Complexity and Real Computation}. Springer-Verlag, New York, 1998.
[7] Gershon Elber and Myung-Soo Kim. Geometric constraint solver using multivariate rational spline functions. In \textit{Proc. 6th ACM Symp. on Solid Modeling and Applications}, pages 1–10. ACM Press, 2001.
[8] Peter Franek and Stefan Ratschan. Effective topological degree computation based on interval arithmetic. \textit{CoRR}, abs/1207.6331, 2012.
[9] Andreas Frommer and Bruno Lang. Existence Tests for Solutions of Nonlinear Equations Using Borsuk’s Theorem. \textit{SIAM J. Numer. Anal.}, 43(3):1348–1361, 2005.
[10] Andreas Frommer, Bruno Lang, and Marco Schnurr. A comparison of the Moore and Miranda existence tests. \textit{Computing}, 72(3-4):349–354, 2004.
[11] Jürgen Garloff and Andrew P. Smith. Investigation of a subdivision based algorithm for solving systems of polynomial equations. \textit{J. Nonlinear Analysis: Series A Theory and Methods}, 47(1):167–178, 2001.
[12] Jürgen Garloff and Andrew P. Smith. Solution of systems of polynomial equations by using Bernstein expansion. In G. Alefeld, S. Rump, J. Rohn, and T. Yamamoto, editors, \textit{Symbolic Algebraic Methods and Verification Methods ( Dagstuhl 1999)}, pages 87–97. Springer, Vienna, 2001.
[13] Alexandre Goldshtein. Comparison of the Hansen-Sengupta and the Frommer-Lang-Schnurr existence tests. \textit{Computing}, 79(1):53–69, 2007.
[14] Rémi Imbach, Victor Pan, and Chee Yap. Implementation of a near-optimal complex root clustering algorithm. In \textit{Proc. Int’1 Congress on Mathematical Software, 2018.}. 6th ICMS, Notre Dame University, July 24-27, 2018.
[15] R Baker Kearfott. \textit{Rigorous global search: continuous problems}, volume 13. Springer Science & Business Media, 2013.
[16] John B Kioustelidis. Algorithmic error estimation for approximate solutions of nonlinear systems of equations. \textit{Computing}, 19(4):313–320, 1978.
[17] Alexander Kobel, Fabrice Rouillier, and Michael Sagraloff. Computing real roots of real polynomials ... and now for real! In \textit{41st Int’l Symp. Symbolic and Alge. Comp.}, pages 303–310, 2016. July 19-22, Waterloo, Canada.
[18] Jyh-Ming Lien, Vikram Sharma, Gert Vegter, and Chee Yap. Isotopic arrangement of simple curves: An exact numerical approach based on subdivision. In \textit{ICMS 2014}, pages 277–282. Springer, 2014. LNCS No. 8592. Download from http://cs.nyu.edu/exact/papers/ for version with Appendices and details on MK Test.
[19] Long Lin and Chee Yap. Adaptive isotopic approximation of nonsingular curves: the parameterizability and nonlocal isotopy approach. \textit{Discrete and Comp. Geom.}, 45(4):769–795, 2011.
[20] R. E. Moore and J. B. Kioustelidis. A simple test for accuracy of approximate solutions to nonlinear (or linear) systems. \textit{SIAM J. Numer. Anal.}, 17(4):521–529, 1980.
[21] Ramon E. Moore. \textit{Methods and Applications of Interval Analysis}. SIAM, Philadelphia, PA, 1995. Second Reprint.
[22] B. Mourrain and J.-P. Pavone. Subdivision methods for solving polynomial equations. \textit{J. Symbolic Computation}, 44(3):292–306, 2009.
[23] Arnold Neumaier. \textit{Interval Methods for Systems of Equations}. Cambridge University Press, Cambridge, 1990.
[24] Helmut Ratschek and Jon Rokne. \textit{Computer Methods for the Range of Functions}. Horwood Publishing Limited, Chichester, West Sussex, UK, 1984.
[25] Michael Sagraloff and Kurt Mehlhorn. Computing real roots of real polynomials. \textit{J. Symbolic Computation}, 73:46–86, 2016.
[26] Evan C. Sherbrooke and Nicholas M. Patilakalis. Computation of the solutions of nonlinear polynomial systems. \textit{Computer Aided Geometric Design}, 10:379–405, 1993.
[27] Pascal van Hentenryck, David McAllester, and Deepak Kapur. Solving polynomial systems using a branch and prune approach. \textit{SIAM J. Numer. Analysis}, 32(4):797–827, 1995.
[28] Juan Xu and Chee Yap. Effective Subdivision Algorithm for Isolating Zeros of Real Systems of Equations, with Complexity Analysis. \textit{arXiv e-prints}, page arXiv:1905.03505, May 2019.
[29] C. Yap, M. Sagraloff, and V. Sharma. Analytic root clustering: A complete algorithm using soft zero tests. In \textit{The Nature of Computation, Logic, Algorithms, Applications}, volume 7921 of \textit{LNCS}, pages 434–444. Springer, 2013.
[30] Chee K. Yap. Symbolic treatment of geometric degeneracies. \textit{J. Symbolic Computation}, 10:349–370, 1990.
[31] Chee K. Yap. In praise of numerical computation. In S. Albers, H. Alt, and S. Näher, editors, \textit{Efficient Algorithms}, volume 5760 of \textit{LNCS}, pages 308–407. Springer-Verlag, 2009.