Journal of Computational and Applied Mathematics 280 (2015) 377–395

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

Root refinement for real polynomials using quadratic interval refinement

Michael Kerber *, Michael Sagraloff

MPI for Informatics, Campus E 1.4, 66123 Saarbrücken, Germany

ARTICLE INFO

Article history:
Received 23 January 2014
Received in revised form 21 November 2014

Keywords:
Root isolation
Root approximation
Quadratic interval Refinement
Complexity analysis

ABSTRACT

We consider the problem of approximating all real roots of a square-free polynomial f with real coefficients. Given isolating intervals for the real roots and an arbitrary positive integer L, the task is to approximate each root to L bits after the binary point. Abbott has proposed the quadratic interval refinement method (QIR for short), which is a variant of Regula Falsi combining the secant method and bisection. We formulate a variant of QIR, denoted AQIR ("Approximate QIR"), that considers only approximations of the polynomial coefficients and chooses a suitable working precision adaptively. It returns a certified result for any given real polynomial, whose roots are all simple. In addition, we propose several techniques to improve the asymptotic complexity of QIR: We prove a bound on the bit complexity of our algorithm in terms of the degree of the polynomial, the size and the separation of the roots, that is, parameters exclusively related to the geometric location of the roots. For integer coefficients, our variant improves, in theory and practice, the variant with exact integer arithmetic. Combining our approach with multipoint evaluation, we obtain near-optimal bounds that essentially match the best known theoretical bounds on root approximation as obtained by very sophisticated algorithms.

1. Introduction

The problem of computing the real roots of a polynomial in one variable is one of the best studied problems in mathematics. If one asks for a *certified* method that finds all roots, it is common to write the solutions as a set of disjoint *isolating* intervals, each containing exactly one root; for that reason, the term *real root isolation* is common in the literature. For integer polynomials, simple, though efficient, methods for this problem have been presented, for instance, based on Descartes' Rule of Signs [1], or on Sturm's theorem [2,3]. The majority of these methods exclusively performs exact arithmetic over rational numbers, and thus returns a certified result. Recently, the focus of research shifted to polynomials with real coefficients, which are approximated during the algorithm. It is worth remarking that this approach does not just generalize the integer case but also leads to practical [4,5] and theoretical [6] improvements of it.

We consider the related *real root refinement problem*: assuming that isolating intervals of a polynomial are known, *refine* them to a width of 2^{-L} or less, where $L \in \mathbb{N}$ is an additional input parameter. The combination of root isolation and root refinement, which we call *strong root isolation*, yields a certified approximation of all roots of the polynomial to an absolute precision of 2^{-L} or, in other words, to L bits after the binary point in binary representation. Abbott's *quadratic interval refinement* method [7] (QIR for short) is a hybrid of the bisection and the secant method, which eventually converges quadratically (see Section 3). A straight-forward approach for refinement is to apply QIR to each isolating interval until the

* Corresponding author. Tel.: +49 683193251005.
E-mail addresses: mkerber@mpi-inf.mpg.de (M. Kerber), msagralo@mpi-inf.mpg.de (M. Sagraloff).

http://dx.doi.org/10.1016/j.cam.2014.11.031
0377-0427/

width becomes small enough. This approach combines the advantages of being certified (the refined interval is guaranteed to contain a root), simple (in terms of implementation) and adaptive (the refinement process might switch to quadratic convergence much earlier than predicted by theory) and has therefore been used in several implementations where accurate and certified approximations of real roots are required. Another advantage is that the polynomial is only queried through polynomial evaluations; for instance, derivatives as in Newton iteration are not needed. The QIR approach has been analyzed in [8] for integer polynomials: for a polynomial of degree d and τ-bit coefficients and m real roots, it requires $\tilde{O}(d^4\tau^2 + md^2L)$ bit operations to refine all roots to L bits.

1.1. Our contributions

Our work generalizes [8] from integer to arbitrary real coefficients, without assuming that exact operations on real numbers are available at unit cost. Instead, our approach works only with approximations of the input and exclusively performs approximate but certified arithmetic, that is, in each step of our algorithm, errors which result from the use of approximate arithmetic are bounded and taken into account using interval arithmetic. We assume the existence of an oracle which, for an arbitrary positive integer ρ, provides approximations of the coefficients of the input polynomial to an error of less than $2^{-\rho}$. We also quantify the size of ρ in the worst case. We obtain an algorithm, called AQIR ("Approximate QIR") that shares the traits of being certified, simple, and adaptive with its exact counterpart, given that arbitrary approximations of the coefficients are accessible. We analyze the bit complexity of AQIR and give a bound that depends on the degree of the polynomial, the size of its largest root and the separation of its roots; see Theorem 22 and Section 2 for the precise statement. We are not aware of a similar in-depth analysis of any related approach for real coefficients. Our analysis proceeds in a similar way as [8], splitting the sequence of QIR steps in the refinement process into a *linear sequence*, where the method behaves like bisection in the worst case, and into a *quadratic sequence*, where the interval is converging quadratically towards the root; for technical reasons, we introduce an initial *normalization phase* that modifies the intervals to guarantee the efficiency of our refinement strategy.

Remarkably, AQIR does not only generalize, but also improves the integer case: We obtain a bit complexity of $\tilde{O}(d^3\tau^2 + mdL)$ if all coefficients are τ-bit integers. Compared to [8], we get rid of one factor of d by a different approach for evaluating the sign of f at rational points, which is the main operation in the refinement procedure. For an interval of size $2^{-\ell}$, the evaluation of f at the endpoints of the interval has a complexity of $\tilde{O}(d^2(\tau + \ell))$ with exact rational arithmetic because the function values can consist of up to $d(\tau + \ell)$ bits. However, we show that we can still compute the sign of the function value with certified numerical methods using the substantially smaller working precision of $O(d\tau + \ell)$ if the distance to the closest root is not much smaller than $2^{-\ell}$. Thus, the crucial modification of AQIR is to ensure that the boundaries of an isolating interval are sufficiently far away from the root that it contains; the details are described in Sections 4–6. Moreover, we show by experimental comparisons (Section 7) that our asymptotic improvement is also reflected in practice: the ratio of running times of exact and approximate QIR is proportional to the degree. Although AQIR is tailored to an accessible complexity analysis and does not yield an efficient practical implementation without further modifications, our experiments show that using approximate arithmetic is crucial for any efficient implementation of QIR.

We can further reduce the complexity of our method using the technique of *fast approximate multipoint evaluation* [9–11]. With that approach, performing one refinement step on all isolating intervals simultaneously has the same cost (up to logarithmic factors) as a single refinement step with classical evaluation. This yields a bit complexity of $\tilde{O}(d^3\tau^2 + dL)$ for integer polynomials; the bound for polynomials with arbitrary real coefficients also scales like $\tilde{O}(dL)$ for large L. Notice that this bound is optimal up to logarithmic factors if L is the dominating factor and $m = \Theta(d)$ because the output complexity is $\Theta(mL)$. We consider fast approximate multi-point evaluation as a purely theoretical tool to improve the bit (and also the arithmetic) complexity. Although not hard to implement, an *efficient* realization is not easy to achieve and its advantages are unlikely to show up for instances that are currently feasible. For this reason, we refrained from implementing the corresponding variant of AQIR.

1.2. Related work

The problem of accurate root approximation is omnipresent in mathematical applications; certified methods are of particular importance in the context of computations with algebraic objects, for instance, when computing the topology of algebraic curves [12,13] or when solving systems of multivariate equations [14,15].

The theoretical complexity of approximating all complex roots has been investigated by Pan [16–18]. His approach combines the splitting circle method [19] with techniques from numerical analysis (i.e. Newton iteration, Graeffes method, discrete Fourier transforms) and fast algorithms for polynomial and integer multiplication. For polynomials with integer coefficients, this yields an algorithm with a bit complexity of $\tilde{O}(d^2\tau + dL)$ for approximating all roots to an accuracy of 2^{-L}, and, for square-free polynomials with arbitrary real coefficients, it still scales like $\tilde{O}(dL)$ for large L; see [20,21] for details. Our bound for AQIR matches this complexity if L is the dominant factor, but it is still inferior by a factor of $d\tau$ in the first term. Still, we think that this does not turn our analysis obsolete because we show that almost optimal asymptotic bounds can also be achieved with an easily implementable and practically efficient method. In contrast, as Pan admits in [22], the splitting-circle method is difficult to implement and so is the complexity analysis when taking rounding errors in inter-