Journal of Symbolic Computation 65 (2014) 79–110

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

On the complexity of the Descartes method when using approximate arithmetic

Michael Sagraloff

MPI for Informatics, Saarbrücken, Germany

ARTICLE INFO

Article history:
Received 12 July 2013
Accepted 4 December 2013
Available online 30 January 2014

Keywords:
Root isolation
Descartes method
Subdivision methods
Numerical computation
Complexity bounds
Approximate coefficients

ABSTRACT

In this paper, we introduce a variant of the Descartes method to isolate the real roots of a square-free polynomial $F(x) = \sum_{i=0}^{n} A_i x^i$ with arbitrary real coefficients. It is assumed that each coefficient of F can be approximated to any specified error bound. Our algorithm uses approximate arithmetic only, nevertheless, it is certified, complete and deterministic. We further provide a bound on the complexity of our method which exclusively depends on the geometry of the roots and not on the complexity of the coefficients of F. For the special case, where F is a polynomial of degree n with integer coefficients of maximal bitsize τ, our bound on the bit complexity writes as $\tilde{O}(n^3\tau^2)$. Compared to the complexity of the classical Descartes method from Collins and Akritas (based on ideas dating back to Vincent), which uses exact rational arithmetic, this constitutes an improvement by a factor of n. The improvement mainly stems from the fact that the maximal precision that is needed for isolating the roots of F is by a factor n lower than the precision needed when using exact arithmetic.

1. Introduction

Computing the roots of a univariate polynomial can be considered as one of the fundamental problems in computational algebra, and numerous approaches have been proposed in the last decades to solve this problem. In this paper, we focus on the problem of *isolating the real roots* of a square-free polynomial $F \in \mathbb{R}[x]$ with arbitrary real coefficients. More precisely, given approximations of the coefficients of F to an arbitrary precision, we aim to compute disjoint intervals $J_1, \ldots, J_m$ such that each J_i contains exactly one root of F and such that their union contains all real roots of F. For polynomials with integer coefficients, the so-called Descartes method (or "Vincent–Collins–Akritas"

E-mail address: msagralo@mpi-inf.mpg.de.

0747-7171/$ – see front matter
http://dx.doi.org/10.1016/j.jsc.2014.01.005

method),[1] first introduced by Collins and Akritas (1976), constitutes one of the simplest and most efficient algorithms. In order to better understand the contribution of this paper, we briefly review the algorithm: It starts with an interval $\mathcal{I}$ containing all real roots of F and recursively proceeds as follows: For an interval $I=(a,b)\subset\mathcal{I}$, Descartes' Rule of Sign is used to test I for roots of F. If it yields that the number m of roots contained in I equals zero, I is discarded. If it yields that $m=1$, then I is stored as an isolating interval. In all other cases, I is subdivided into two equally sized subintervals $I_\ell:=(a,m(I))$ and $I_r:=(m(I),b)$, where $m(I)$ denotes the midpoint of I. For a polynomial F of degree n with integer coefficients of bit-size τ, the Descartes method induces a recursion tree of size $O(n(\tau+\log n))$, where the latter bound has shown to be optimal (Eigenwillig et al., 2006). For Descartes' Rule of Signs, we need to compute the polynomial[2]

$$F_{I,\mathrm{rev}}(x):=(x+1)^n\cdot F\left(\frac{ax+b}{x+1}\right). \tag{1.1}$$

Using asymptotically fast Taylor shifts (Gerhard, 2004; von zur Gathen and Gerhard, 1997; Schönhage, 1982), the cost for this computation is bounded by

$$\tilde{O}\left(n^2\left(\log^+\max\left(|a|,|b|\right)+\log^+|b-a|^{-1}\right)\right)=\tilde{O}\left(n^3\tau\right) \tag{1.2}$$

bit operations,[3] where we define $\log^+(x):=\log\max(2,|x|)\geqslant 1$ for all $x\in\mathbb{C}$ and $\log:=\log_2$. The bound in (1.2) follows from the fact that we have to perform $\tilde{O}(n)$ arithmetic operations and that $F_{I,\mathrm{rev}}$ has rational coefficients of bit-size $O(n(\log^+\max(|a|,|b|)+\log^+|b-a|^{-1}))=\tilde{O}(n^2\tau)$. Multiplication of the bound on the recursion tree and the bound (1.2) on the bit complexity for the computations at each node yields the bound $\tilde{O}(n^4\tau^2)$ on the overall bit complexity of the Descartes method.

The advantages of the Descartes method are its simplicity and that the size of the recursion tree adapts well to the geometric locations of the roots, that is, the recursion tree becomes large if and only if some of the roots are clustered. A disadvantage of the Descartes method is that the exact computation of the polynomials $F_{I,\mathrm{rev}}$ needs a precision of $\tilde{\Theta}(n^2\tau)$ in the worst case, whereas separating the roots from each other needs only $\tilde{O}(n\tau)$ bits. In fact, the binary representation of the endpoints of all isolating intervals returned by the algorithm needs no more than $\tilde{O}(n\tau)$ bits. This brings up the question whether approximate computation of the polynomials $F_{I,\mathrm{rev}}$ yields any improvement with respect to the precision demand during the computation and, thus, also with respect to the bit complexity of the Descartes method. This question has been addressed in a series of previous papers: Johnson and Krandick (1997) introduced a hybrid method that uses interval arithmetic based on floating point computation (up to a certain fixed precision) to compute the polynomials $F_{I,\mathrm{rev}}$. This allows to determine the signs of the coefficients of $F_{I,\mathrm{rev}}$ (and, thus, to use Descartes' Rule of Signs) for most of the considered intervals within the subdivision process by using approximate arithmetic, whereas, for the remaining intervals, the method falls back to exact computation. Hence, floating point arithmetic is used as a filter which allows to decrease the precision demand for most intervals, however, no improvement is achieved with respect to worst case bit complexity. Rouillier and Zimmermann (2004) modified the latter approach by arbitrarily increasing the working precision at each stage of the algorithm. It is currently one of the fastest algorithms in practice (e.g. the univariate solver in Maple is based upon this method), however, no result on the needed precision demand and its computational complexity is known, and we expect that, without further modifications,

[1] There exist numerous discussions (e.g. Akritas, 2008) about whether "Descartes method" is the correct term since Descartes did not introduce any algorithm to isolate the roots but (only) a method to estimate the number of positive roots of a univariate polynomial (i.e. Descartes' Rule of Signs). However, because of the fact that the algorithm from Collins and Akritas (based on ideas dating back to Vincent) exclusively uses this rule as inclusion and exclusion predicate, it is reasonable to name the algorithm after Descartes without using the possessive "s" following his name.

[2] Descartes' Rule of Sign states that the number m of roots contained in I is upper bounded by the number v of sign changes in the coefficient sequence of $F_{I,\mathrm{rev}}$ and that $v\equiv m \bmod 2$. For more details, we refer to Section 2.6.

[3] According to Cauchy's Root Bound (see e.g. Yap, 2000), we can assume that $\mathcal{I}\subset(-1-2^\tau,1+2^\tau)$, and thus $\max(1,|a|,|b|)\leqslant 1+2^\tau$. In addition, Descartes method does not subdivide intervals of size less than half of the minimal distance between two distinct roots of F (i.e. the separation σ_F of F), and $\log\max(1,\sigma_F)=O(n(\tau+\log n))$; see Section 2.6 for details.