Theoretical Computer Science 681 (2017) 138–145

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Accelerated approximation of the complex roots and factors of a univariate polynomial

Victor Y. Pan [a,b,*], Elias Tsigaridas [c]

[a] Department of Mathematics and Computer Science, Lehman College of the City University of New York, Bronx, NY 10468, USA
[b] Ph.D. Programs in Mathematics and Computer Science, The Graduate Center of the City University of New York, New York, NY 10036, USA
[c] INRIA, Paris-Rocquencourt Center, PolSys, Sorbonne Universités, UPMC Univ. Paris 06, PolSys, UMR 7606, LIP6, F-75005, Paris, France

ARTICLE INFO

Article history:
Available online 31 March 2017

Keywords:
Polynomial equation
Roots
Root-finding
Root-refinement
Power sums
Complexity

ABSTRACT

The algorithms of Pan (1995) [20] and Pan (2002) [22] approximate the roots of a complex univariate polynomial in nearly optimal arithmetic and Boolean time but require a precision of computing that exceeds the degree of the polynomial. This causes numerical stability problems when the degree is large. We observe, however, that such a difficulty disappears at the initial stage of the algorithms, and in our present paper we extend this stage to root-finding within a nearly optimal arithmetic and Boolean complexity bounds provided that some mild initial isolation of the roots of the input polynomial has been ensured. Furthermore our algorithm is nearly optimal for the approximation of the roots isolated in a fixed disc, square or another region on the complex plane rather than all complex roots of a polynomial. Moreover the algorithm can be applied to a polynomial given by a black box for its evaluation (even if its coefficients are not known); it promises to be of practical value for polynomial root-finding and factorization, the latter task being of interest on its own right. We conclude with summarizing our algorithms and their extension to the approximation of isolated multiple roots and root clusters.

1. Introduction

The classical problem of univariate polynomial root-finding has been central in Mathematics and Computational Mathematics for about four millennia since the Sumerian times, and is still important for Signal and Image Processing, Control, Geometric Modeling, Computer Algebra, and Financial Mathematics. It is closely linked to the approximation of linear and nonlinear factors of a polynomial, which is also important on its own right because of the applications to the time series analysis, Weiner filtering, noise variance estimation, covariance matrix computation, and the study of multi-channel systems (see Wilson (1969) [34], Box and Jenkins (1976) [6], Barnett (1983) [1], Demeure and Mullis (1989 and 1990) [8,9], Van Dooren (1994)) [32].

Solution of both problems within nearly optimal arithmetic and Boolean complexity bounds (up to polylogarithmic factors) have been obtained in Pan (1995) [20] and Pan (2002) [22], but the supporting algorithms require a precision of computing that exceeds the degree of the input polynomial, and this causes numerical stability problems when the degree is large.

* Corresponding author.
E-mail addresses: victor.pan@lehman.cuny.edu (V.Y. Pan), elias.tsigaridas@inria.fr (E. Tsigaridas).

http://dx.doi.org/10.1016/j.tcs.2017.03.030
0304-3975/

The most popular packages of numerical subroutines for complex polynomial root-finding, such as MPSolve 2000 (see Bini and Fiorentino (2000) [3]), EigenSolve 2001 (see Fortune (2002) [10]), and MPSolve 2012 (see Bini and Robol (2014) [5]) employ alternative root-finders based on functional iterations (namely, Ehrlich–Aberth's and WDK, that is, Weierstrass', also known as Durand–Kerner's) and the QR algorithm applied to eigen-solving for the companion matrix of the input polynomial. The user considers these root-finders practically superior by relying on the empirical data about their excellent convergence, even though these data have no formal support. To their disadvantage, these algorithms compute the roots of a polynomial in an isolated region of the complex plane not much faster than all its roots.

We re-examine the subject, still assuming input polynomials with complex coefficients, and show that the cited deficiency of the algorithms of [20] and [22] disappears if we modify the initial stage of these algorithms and apply them under some mild assumptions about the initial isolation of the root sets of the input polynomial. Moreover, like the algorithms of [20] and [22] and unlike WDK and Ehrlich–Aberth's algorithms, the resulting algorithms are nearly optimal for the approximation of the roots in an isolated region of the complex plane.

Next we briefly comment on our results. In the next sections we elaborate upon them, deduce the computational cost estimates, and outline some natural extensions.

Recall that polynomial root-finding iterations can be partitioned into two stages. At first a crude (although reasonably good) initial approximations to all roots or to a set of roots are relatively slowly computed. Then these approximations are refined faster by means of the same or distinct iterations.

Our algorithm applies at the second stage and is nearly optimal, under both arithmetic and Boolean complexity models and under mild initial isolation of every root, some roots or some root sets. Such an isolation can be observed at some stages of root approximation by Ehrlich–Aberth's and WDK algorithms, but with no estimates for the computational cost of reaching isolation. Towards the solution with controlled computational cost, one can apply advanced variants of Weyl's Quad-tree construction of Weyl 1924 [33], successively refined in Henrici 1974 [12], Renegar 1987 [28], and Pan 2000 [21].

The algorithm of the latter paper computes all roots of a polynomial or its roots in a fixed isolated region at a nearly optimal arithmetic cost, and it is nearly optimal for computing the initial isolation of the roots as well; the paper [21] does not estimate the Boolean cost of its algorithm, but most of its steps allow rather straightforward control of the precision of computing.[1]

Having reached an initial isolation, we can apply our present algorithm which supports nearly optimal complexity estimates and if properly implemented has very good chances to become the user's choice.

Besides the root-finding applications, our algorithm can be a valuable ingredient of the polynomial factorization algorithms. Recall that one can extend factorization to root-finding (see Schönhage (1982) [29,20,22], and the present paper), but also root-finding to factorization (see Pan (2012) [23]). Our algorithm can be technically linked to those of [29,20,22]; our results (Theorem 8 and 10) could be also viewed as an extension of the recent record and nearly optimal bounds for the approximation of the real roots [26,27], see also [17].

An interesting challenge is the design of a polynomial factorization algorithms that both are simple enough for practical implementation and support factorization at a nearly optimal computational complexity. An efficient solution outlined in our last section combines our present algorithms with the one of Pan 2012 [23] and McNamee and Pan 2013 [18, Section 15.23], which is a simplified version of the efficient but very much involved algorithm of Kirrinnis (1998) [13].

Like our present algorithm, these solution algorithms are nearly optimal and remain nearly optimal when they are applied to a polynomial given by a black box for its evaluation, even when its coefficients are not known.

Organization of the paper We recall the relevant definitions and some basic results in the remainder of this section and in the next section. In Section 3 we present our main algorithm, prove its correctness, and estimate its arithmetic cost when it is applied to the approximation of a single root and d simple isolated roots of a dth degree polynomial. In Section 4 we extend our analysis to estimate the Boolean cost of these computations. In our concluding Section 5 we summarize our results and outline their extension to factorization of a polynomial and to root-finding in the cases of isolated multiple roots and root clusters.

Some definitions

- For a polynomial $u = u(x) = \sum_{i=0}^{d} u_i x^i$, the norms $||u||_\gamma$ denote the norms $||\mathbf{u}||_\gamma$ of its coefficient vector $\mathbf{u} = (u_i)_{i=0}^{d}$, for $\gamma = 1, 2, \infty$.
- $D(X, r)$ denotes the complex disc $\{x : |x - X| \le r\}$.
- "ops" stands for "arithmetic operations".
- DFT(q) denotes the discrete Fourier transform at q points. It can be performed by using $O(q\log(q))$ ops.

[1] More recent variations of the Quad-tree algorithm have been studied by various authors under the name of subdivision algorithms for polynomial root-finding.