Journal of Symbolic Computation 47 (2012) 153–166

Contents lists available at SciVerse ScienceDirect

Journal of Symbolic Computation

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

SqFreeEVAL: An (almost) optimal real-root isolation algorithm

Michael A. Burr [a,c,1], Felix Krahmer [b,c]

[a] Fordham University, 441 East Fordham Road, Bronx, NY 10458, USA

[b] Hausdorff Center for Mathematics, Universität Bonn, Endenicher Allee 60, 53115 Bonn, Germany

[c] Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, USA

ARTICLE INFO

Article history:
Received 4 April 2011
Accepted 17 August 2011
Available online 14 September 2011

Keywords:
Continuous amortization
Adaptive analysis
Subdivision algorithm
Integral analysis
Amortization
Root isolation

ABSTRACT

Let f be a univariate polynomial with real coefficients, $f \in \mathbb{R}[X]$. Subdivision algorithms based on algebraic techniques (e.g., Sturm or Descartes methods) are widely used for isolating the real roots of f in a given interval. In this paper, we consider a simple subdivision algorithm whose primitives are purely numerical (e.g., function evaluation). The complexity of this algorithm is adaptive because the algorithm makes decisions based on local data. The complexity analysis of adaptive algorithms (and this algorithm in particular) is a new challenge for computer science. In this paper, we compute the size of the subdivision tree for the SqFreeEVAL algorithm.

The SqFreeEVAL algorithm is an evaluation-based numerical algorithm which is well-known in several communities. The algorithm itself is simple, but prior attempts to compute its complexity have proven to be quite technical and have yielded sub-optimal results. Our main result is a simple $O(d(L+\ln d))$ bound on the size of the subdivision tree for the SqFreeEVAL algorithm on the benchmark problem of isolating all real roots of an integer polynomial f of degree d and whose coefficients can be written with at most L bits.

Our proof uses two amortization-based techniques: first, we use the algebraic amortization technique of the standard Mahler–Davenport root bounds to interpret the integral in terms of d and L. Second, we use a continuous amortization technique based on an integral to bound the size of the subdivision tree. This paper is the first to use the novel analysis technique of continuous amortization to derive state of the art complexity bounds.

E-mail addresses: mburr1@fordham.edu (M.A. Burr), felix.krahmer@hcm.uni-bonn.de (F. Krahmer).

[1] Tel.: +1 718 817 3298; fax: +1 718 817 3236.

0747-7171/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jsc.2011.08.022

1. Introduction

In this paper, we show that the size of the subdivision tree for the simple, evaluation-based, numerical algorithm `SqFreeEVAL` has size $O(d(L + \ln d))$ for the benchmark problem of isolating all of the real roots of an integer polynomial of degree d whose coefficients can be represented by at most L bits. Under the mild assumption that $L \geq \ln d$, this complexity simplifies to the optimal size of $O(dL)$, see Eigenwillig et al. (2006, Section 3.3) for a proof of optimality. The optimality and simplicity of the `SqFreeEVAL` algorithm imply that it may be a useful algorithm in practical settings. The bound on the size of the subdivision tree is achieved via a straight-forward and elementary argument. The two main techniques which are used in the computation are algebraic amortization, in the form of Mahler–Davenport bounds, and continuous amortization, in the form of an integral technique as presented in Burr et al. (2009).

1.1. `EVAL`-type algorithms

The `SqFreeEVAL` algorithm which we study in this paper is a specific example of what we call an `EVAL`-type algorithm. These algorithms are so named because they are based on function evaluation: `EVAL`-type algorithms take, as input, functions which allow some subset of the following two predicates: first, these functions and their derivatives can be evaluated at a countable dense subset of their domain. In this paper, the domain will be the real numbers and the countable dense subset will be the dyadic integers. Second, these functions and their derivatives can be approximated on intervals in such a way that the approximation converges as the input intervals converge to a point. In this paper, the approximation is derived from interval arithmetic on a Taylor sequence. The simplest and most well-known example of an `EVAL`-type algorithm is Lorensen and Cline's marching cube algorithm (Lorensen and Cline, 1987).

`EVAL`-type algorithms are typically studied because of their simplicity and generality. These algorithms are fairly general because their inputs can be extended to more general analytic functions. In particular, many analytic functions have interval arithmetic available to them, and, therefore, it is possible to approximate these functions on intervals. In addition, with the limited predicates available to `EVAL`-type algorithms, most of the techniques which are used in these algorithms are analytically based (as opposed to algebraically based). These algorithms are simple because, in many cases, `EVAL`-type algorithms are based on simple recursive bisection algorithms. Such algorithms iteratively subdivide an initial domain until each set in the resulting partition of the initial domain satisfies a (usually simple) terminal condition. Bisection algorithms are common in computer graphics (Boier-Martin et al., 2005) as well as in computational science and engineering applications (Domain Decomposition Methods, 2011). Bisection algorithms are of particular interest because they are adaptive; they perform more bisections near difficult features and fewer bisections elsewhere. However, this adaptivity makes the complexity analysis of such algorithms more difficult because the subdivision tree may have a few deep paths while the remainder of the tree remains modest in size.

`EVAL`-type algorithms have been studied in the univariate case in Henrici (1970), Yakoubsohn (2005), Yap and Sagraloff (2011), Burr et al. (in preparation) and Burr et al. (2009), in the bivariate and trivariate cases in Lorensen and Cline (1987), Snyder (1992), Plantinga and Vegter (2004), Plantinga (2006), Lin and Yap (2009) and Burr et al. (2012), and in the multivariate case in Galehouse (2009) and Dedieu and Yakoubsohn (1992). All of these algorithms are devoted to approximating algebraic (and in some cases analytic varieties) in the real or complex settings. The algorithms in Burr et al. (in preparation) and Burr et al. (2009) are designed to find all real roots of a polynomial or analytic function while the algorithms in Henrici (1970), Yakoubsohn (2005) and Yap and Sagraloff (2011) are designed to find the complex roots of a polynomial or analytic function (note that Henrici (1970) is only designed to find a single root of a polynomial). Each of these algorithms is very closely related to the `SqFreeEVAL` algorithm considered in this paper; the main differences are in the setting, in the type of subdivisions performed, and in various preprocessing steps. We give a more detailed account of these algorithms in the next section. The two-dimensional `EVAL`-type algorithm (Plantinga and Vegter, 2004; Plantinga, 2006) was presented for approximating smooth and bounded varieties. It was