J. Symbolic Computation (2002) **00**, 1–33
doi:10.1006/jsco.2002.0531
Available online at http://www.idealibrary.com on IDEAL®

Univariate Polynomials: Nearly Optimal Algorithms for Numerical Factorization and Root-finding*

VICTOR Y. PAN†

Mathematics and Computer Science Department, Lehman College, CUNY, Bronx, NY 10468, U.S.A.

To approximate all roots (zeros) of a univariate polynomial, we develop two effective algorithms and combine them in a single recursive process. One algorithm computes a basic well isolated zero-free annulus on the complex plane, whereas another algorithm numerically splits the input polynomial of the nth degree into two factors balanced in the degrees and with the zero sets separated by the basic annulus. Recursive combination of the two algorithms leads to computation of the complete numerical factorization of a polynomial into the product of linear factors and further to the approximation of the roots. The new root-finder incorporates the earlier techniques of Schönhage, Neff/Reif, and Kirrinnis and our old and new techniques and yields nearly optimal (up to polylogarithmic factors) arithmetic and Boolean cost estimates for the computational complexity of both complete factorization and root-finding. The improvement over our previous record Boolean complexity estimates is by roughly the factor of n for complete factorization and also for the approximation of well-conditioned (well isolated) roots, whereas the same algorithm is also optimal (under both arithmetic and Boolean models of computing) for the worst case input polynomial, whose roots can be ill-conditioned, forming clusters. (The worst case complexity bounds for root-finding are supported by our previous algorithms as well.) All algorithms allow processor efficient acceleration to achieve solution in polylogarithmic parallel time.

0. Introduction

0.1. some history of polynomial root-finding

Numerical root-finding for a univariate polynomial is a classical problem which had remained the central and most influential for the development of mathematics since the Sumerian times in the third B.C. and well into the 19th century A.D. The very ideas of abstract thinking and using mathematical notation are largely due to the study of this problem. Furthermore, this study has motivated the introduction of some fundamental concepts of mathematics (such as irrational, negative and complex numbers, algebraic groups, fields, and ideals) and has substantially influenced the earlier development of numerical computing. For instance, the regula falsi method appeared in the Ahmes papirus about 1500 B.C. as a method for solving quadratic equations. We refer the reader to Bell (1940), Boyer (1968) and Pan (1997) on this fascinating development. The areas

*The results of this paper were presented at the ACM International Symp. on Symbolic and Algebraic Computation (ISSAC'01).

†E-mail: `vpan@lehman.cuny.edu`

0747–7171/02/000001 + 33 $35.00/0

influenced by the polynomial root-finding problem in the 19th and 20th centuries included meromorphic functions, algebraic curves, and structured matrices (Householder, 1970; Gauss, 1973; Henrici, 1974); furthermore, Weyl's quadtree root-finder alone (Weyl, 1924) has made an impact on computational geometry, image processing, template matching, and the n-body particle simulation (Samet, 1984; Greengard, 1988; Senoussi, 1994). Presently, polynomial root-finding is still a major research topic with highly important applications to computer algebra, in particular to the solution of polynomial systems of equations (Kapur and Lakshman, 1992; Blum *et al.*, 1997; Pan, 1997; Mourrain and Pan, 1998, 2000) (cf. also Pan, 1998, 2001c; Pan and Chen, 1999 on the applications to the computation of approximate polynomial gcds and the algebraic eigenproblem).

It was already understood by the Greeks at about 500 B.C. that even the solution of the equation $x^2 - 2 = 0$ must involve non-arithmetic operations. Ruffini and Abel proved that the inclusion of radicals is generally insufficient for root-finding for polynomials of degree $n \geq 5$; Galois proved this even for specific simple polynomials such as $x^5 - 4x - 2$. Therefore, we arrive at the numerical problem of approximating the roots. Smale (1981) and then Schönhage (1982b) raised the question of devising optimal numerical polynomial root-finders, which would use smaller computational time, and proposed some effective algorithms. In the vast literature on root-finding (McNamee, 1993, 1997), including thousands of publications, not many items are devoted to this important problem, however see Bini and Pan (to appear), Kim and Sutherland (1994), Kirrinnis (1998), Neff and Reif (1994, 1996), Pan (1987, 1995a,b, 1996, 2000), Renegar (1987), Schönhage (1982b) and Smale (1981, 1985). The decisive steps to yield optimality were made quite recently.

0.2. Recent progress, our results and techniques, and some further improvements

The root-finder in Pan (1995a, 1996) relies on recursive numerical splitting of the input polynomial $p(x)$ into the product of pairs of balanced degree (and ultimately linear) factors (the ratios of the degrees lie between 12 and 1/12, say) and reaches optimal (up to polylogarithmic factors) bounds on the asymptotic arithmetic and Boolean time of root-finding for the worst case input polynomial. This case covers polynomials with ill-conditioned (clustered) roots, typically appearing in the result of the numerical truncation of the coefficients of polynomials with multiple roots. The bounds on the computational precision and the Boolean (bit-operation) cost of performing the algorithm, however, are too high. They are off by the factor of n, the degree of $p(x)$, from the information lower bounds at the auxiliary stage of polynomial factorization and for the practically important case of root-finding for the input polynomials with well-conditioned (isolated) roots. Factorization is computed by means of splitting $p(x)$ into the product of non-constant factors and then splitting recursively the nonlinear factors. The splitting and factorization problems are of independent interest due to their applications to time series analysis, Weiner filtering, noise variance estimation, covariance matrix computation, and the study of multi-channel systems (Wilson, 1969; Box and Jenkins, 1976; Barnett, 1983; Demeure and Mullis, 1989, 1990; Van Dooren, 1994).

In the present work, we simplify the construction in Pan (1995a, 1996) where a higher precision of computing of the order of at least bn bits is maintained throughout. This is indeed required for approximating the ill-conditioned zeros of $p(x)$ (see Fact 1.1 in Pan, 1996) and its higher order derivatives $p^{(l)}(x)$ unless $n - l = o(n)$. Now we observe that the factorization of $p(x)$ and $p^{(l)}(x)$ does not require such a high computational precision (the order of b bits is sufficient) except for the stage of approximating a single well isolated zero or cluster of zeros of $p^{(l)}(x)$. Here we need a higher precision to define the basic splitting annuli (or discs) but the Boolean cost at this stage is dominated by the overall Boolean cost of our factorization algorithm, which is by the factor of n less than the root-finding cost. Thus we rearrange the computations in Pan (1995a, 1996) respectively, to decrease the precision of computing throughout (except for the cited stage).

With our improved algorithm, we keep optimality (up to polylog factors) of the root-finding for the worst case input polynomial but simultaneously reach it in the case of well-conditioned roots (zeros) as well as for the complete factorization of a polynomial. In both cases, the improvement of our bit-operation (Boolean) cost bounds vs. Pan (1995a, 1996) is by the factor of n, due to the decrease of the computational precision (the arithmetic cost bounds are nearly linear already in Pan (1995a, 1996)). Technical statement of these results is in Section 2.1 (in Part II).

All algorithms of this paper allow work (processor) efficient parallel acceleration. This yields polylogarithmic parallel time bounds preserving work (processor) optimality up to polylog factors.

Apparently, our asymptotic bit-operation cost bounds can be improved by roughly the logarithmic factor if one applies fast integer arithmetic based on the binary segmentation techniques (cf. Schönhage, 1982a,b; Bini and Pan 1994, Section 3.3; Kirrinnis, 1998). Asymptotically, these techniques are slightly superior to the FFT-based arithmetic, on which we rely in our estimates. At the stage of splitting into two factors, an acceleration by the logarithmic factor was also claimed in Bini *et al.* (to appear); if confirmed, this improvement can apparently be combined with the one obtained via binary segmentation and then our construction would immediately accommodate both improvements for the factorization and root-finding. Further comparison with some related works is given in Section 0.3.

Our root-finder remains nearly optimal (up to polylogarithmic factors) even for the more limited tasks of approximating a single root or a few roots of a polynomial, but in these cases the computational cost is slightly lower and the implementation is simpler in our distinct approaches, which use no splitting (Pan, 1987, 2000).

Our algorithms are quite involved, and their implementation would require a non-trivial work, incorporating numerous known implementation techniques and tricks (Bini and Fiorentino, 2000; Fortune, 2001; Bini and Pan, to appear). We do not touch this vast domain here and just briefly comment on the precision of computing.

Our algorithms involve the shifts of the variable (or equivalently of the origin), its scaling, and approximation of the root radii, that is, the distances of the unknown roots from a selected complex point. These techniques have low arithmetic and Boolean cost and are customary for reducing the study to the canonical cases, say where all roots lie in the unit disc $\{x : |x| \leq 1\}$ (see Renegar, 1987; Pan, 1996; Kirrinnis, 1998). On the other hand, using these techniques requires precision of computation of the order of n or $n \log n$ bits, which creates an implementation problem for larger n. Although the Chinese remainder algorithm and Schönhage–Strassen's algorithm (Schönhage and Strassen, 1971) overcome this problem in principle (at least at the asymptotic complexity level), the problem is still substantial for numerical implementation, which is most efficient using the single or double IEEE precision. Thus the current champions in practical numerical root-finding rely on Jenkins–Traub's, modified Laguerre's and modified Newton's algorithms, variations of the Weierstrass method, and the QR algorithm for the companion matrix (McNamee, 1993, 1997; Fortune, 2001; Pan, 2002; Bini and Pan, to appear). We note, however, that our splitting approach has all the potential to be effective for lower precision computation of polynomial factorization (see Malajovich and Zubelli, 1997, Bini *et al.*, to appear) and that the shift-free variations of our algorithms (say in the beginning of Section 2.3) have the same promise (in both cases substantial implementation work would be required).

Our algorithms involve several advanced techniques, some of independent interest such as the reversion of Graeffe's lifting with incorporating the Padé approximation. The power of Graeffe's iteration and the Padé approximation as numerical computation tools was already recognized and exploited in Pan (1996) and then in Malajovich and Zubelli (1997), Pan (1998, 2001c), Schönhage (2000) and Gemignani (2001); we further refine these techniques and suggest exploiting them more widely in algebraic/numerical computations. Our study here, and in Pan (1996), relies on direct ties of the Padé approximation with Toeplitz/Hankel matrix computations and its better numerical stability (at the level of $O(n)$-bit or $O(n \log n)$-bit precision) vs. the Euclidean algorithm. This suggests undertaking systematic revision of the known applications of the Euclidean algorithm in computer algebra (such as the computation of the maximal square-free factor of a univariate polynomial) with its possible replacement by computing Padé approximations. Similar potentials could be investigated for Graeffe's process and its reverse descending process.

0.3. SOME RELATED WORKS

The study of polynomial root-finding is related to various areas of pure and applied mathematics as well as the theory and practice of computing and has huge bibliography (McNamee, 1993, 1997; Pan, 1997; Bini and Pan, to appear). We focus on an important aspect of this study, that is, the computational complexity of the solution under the arithmetic and Boolean (bit-operation) models. As we mentioned, the modern interest in this aspect of the study is due to Schönhage (1982b) and Smale (1981). McNamee (1993, 1997) is a good source for the bibliography; the unpublished manuscript (Schönhage, 1982b) is an important landmark work but is sparse in citations of the preceding works.

Let us extend our comparison with the related works begun in Section 0.2. In Kirrinnis (1998) and Schönhage (1982b) algorithms for splitting a polynomial into the product of two factors were studied extensively, assuming a sufficiently high relative width of the basic root-free annulus, that is, assuming higher isolation of the two respective sets of the roots of the two factors from each other. No balancing of the degrees of the factors was achieved, which implied the extra factor of $n/\log n$ in the Boolean cost estimates. Further improvements by the order of magnitude were due to relaxing the assumption of the isolation (by reversing Graeffe's lifting process with using the Padé approximation) (Pan, 1995a, 1996) and to achieving balanced splitting (Gel'fond, 1958; Coppersmith and Neff, 1994; Neff and Reif, 1994; Pan, 1995a, 1996). The techniques in the papers Kirrinnis (1998), Neff and Reif (1994) and Schönhage (1982b) are more important and should last longer than the computational complexity estimates. Kirrinnis (1998) reached the same bound on the Boolean (bit-operation) cost as in Pan (1995a, 1996) but only under the very strong requirement of blowing up the precision of computing to the order of $(1+\frac{1}{r \log n})n^2$ bits, r being the minimum distance between the distinct roots. That is, this assumption requires computations with a very high precision of the order of n^2 bits even where all roots are well isolated from each other. Moreover under this assumption, the precision must dramatically increase further to handle any cluster of the roots. Otherwise the algorithms in Kirrinnis (1998) and Schönhage (1982b) support the arithmetic and Boolean cost bounds that exceed the bounds in Pan (1995a, 1996) by roughly the factors of n^2 and n, respectively.

Coppersmith and Neff (1994), Gel'fond (1958) and Neff and Reif (1994) contributed advanced techniques for balancing the degrees of the factors produced by splittings, which were an important part of our algorithms in Part II but defined no construction that would have supported nearly optimal complexity estimates. Theorems in Coppersmith and Neff (1994) and Gel'fond (1958) are on the complexification of Rolle's theorem, not on root-finding, whereas the construction in Neff and Reif (1994) still requires further extensive and technically non-trivial work to yield the complexity bounds of Pan (1995a, 1996). One problem is the reliance of Neff and Reif (1994) on the straightforward complete splitting of higher order derivatives of $p(x)$ into the product of all their linear factors, avoided in Pan (1995a, 1996), and this already implies the extra factor of n^δ for a positive δ in the parallel and sequential time bounds. Furthermore, the construction in Neff and Reif (1994) does not include the recursive process of Algorithm 1.2.1 and the recursive root radii process (RRRP) algorithm in Section 2.3 and relies on the approximation of the root radii with a higher precision. In our Remarks 1.3.6 and 2.6.2 we note a dramatic increase of the precision and the Boolean computational cost resulting from this apparently natural but actually too crude treatment of two delicate problems. As a relatively minor deficiency, the problem of massive clusters of roots was not addressed in Neff and Reif (1994), in particular neither (a, B, ψ)-splitting discs, as in Pan (1995a, 1996), nor any alternative for them were introduced.

The paper, Neff and Reif (1996) very closely follows the earlier work of Pan (1995a) but complements it with the Boolean complexity analysis of the descending process. We refer the reader to Bini and Pan (to appear) for our critical comments to this analysis.

Some interesting recent extensions of the result in Pan (1996) were cited in Section 0.2. In addition, Schönhage (2000) and Gemignani (2001) have found further applications of the Padé approximation as a reliable numerical tool for computing polynomial reciprocals and multipoint evaluation, respectively. Malajovich and Zubelli (1997, 2001) further explored a similar property of Graeffe's iteration for splitting a polynomial over a root-free annulus on the complex plane. If the relative width of the annulus is high enough, then their algorithm splits a polynomial by using $O(nb^2 \log b)$ bit-operations, which is favorable where b is relatively small, say $\log b = O(\log \log n)$.

0.4. ORGANIZATION OF THE PAPER

Our root-finder is built on the top of several highly developed constructions and incorporates and improves their advanced techniques. Since Schönhage (1982b) already has 72 pages and Kirrinnis (1998) has 67 pages, this ruled out a self-contained presentation of our root-finding algorithm not mentioning any complete survey of the root-finding field, which includes thousands of publications (McNamee, 1993, 1997). (See a more complete exposition in Bini and Pan (to appear).) Furthermore, the statement of the problem and the final complexity results are relatively compact but the techniques supporting our root-finder cannot be unified easily. At least two groups of very different techniques are involved. We partition our paper respectively into two parts, each with separate enumeration of its sections, equations, theorems, etc. In Part I we describe splitting algorithms, which have importance of their own. In Part II we combine the splitting results of Part I with the search for the basic annuli in a recursive process of nearly optimal factorization and root-finding and state the resulting complexity estimates.

In Section 1.1 we state some definitions, the known theorems, and our own results on splitting a polynomial into two factors over a fixed root-free annulus. In Section 1.2 we define our lifting/descending process for this splitting. We also estimate the arithmetic cost of performing this process. The correctness of the algorithm under the stated computational precision bound is shown in Section 1.3, followed by estimating the bit-operation complexity of splitting. The analysis includes the error estimates for the perturbation of the Padé approximation involved; we deduce them in Sections 1.4 and 1.5. We cover the extensions of our splitting over the unit circle to any basic circle in Section 1.6 and to the complete numerical factorization of a polynomial in Section 1.7. In Section 2.1 (in Theorem 2.1.1 and Corollary 2.1.2) we state our main results on polynomial factorization and root-finding. In Section 2.2, we define the basic concept of the balanced splitting annuli and discs. In Section 2.3, we compute basic splitting annuli for a large class of input polynomials; for the remaining polynomials our algorithms confine most of their roots to a small disc. In Sections 2.4 and 2.5, we recall the results in Neff and Reif (1994) on the computation of the roots of higher order derivatives $p^{(l)}(x)$ of an input polynomial $p(x)$ as a means of balancing the degrees in splitting. In Section 2.6, we yield the same goal without computing the roots of $p^{(l)}(x)$, which enables us to decrease the computational cost and to arrive at our main results stated in Theorem 2.1.1 and Corollary 2.1.2.

Part I: Preconditioned Splitting into Factors

1.1. SPLITTING THEOREMS

We begin with some definitions.

$$p = p(x) = \sum_{i=0}^{n} p_i x^i = p_n \prod_{j=1}^{n} (x - z_j), \qquad p_n \neq 0, \tag{1.1.1}$$

$$A = A(X, r_-, r^+) = \{x : r_- \leq |x - X| \leq r^+\}, \tag{1.1.2}$$

$$|u| = \|u(x)\| = \|u(x)\|_1 = \sum_i |u_i| \text{ for } u = u(x) = \sum_i u_i x^i. \tag{1.1.3}$$

((1.1.3) defines the 1-norm of a polynomial.) $\mu(b)$denotes the number of bit-operations sufficient to multiply two integers modulo 2^b+1. The algorithm in Schönhage and Strassen (1971) supports the asymptotically best bound

$$\mu(b) = O((b \log b) \log \log b) \qquad \text{as } b \to \infty, \tag{1.1.4}$$

but for smaller b it is superseded by the algorithm of Karatsuba and Ofman (1963) and the classical algorithm, which support the bounds $\mu(b) = O(b^{\log_2 3})$, $\log_2 3 = 1.58\ldots$, and $\mu(b) = O(b^2)$, respectively.

A polynomial p is given with its complex coefficients, but they turn into real ones if we shift from $p(x)$ to $p(x)\bar{p}(x)$, where $\bar{p}(x) = \sum_i \bar{p}_i x^i$ and $\bar{a}$ denotes the complex conjugate of a. In fact we only use the coefficients of $p(x)$ represented with some bounded number of bits, that is, within a certain fixed precision: b successive bits of a coefficient are produced by using $O(b)$ bit operaions. W.l.o.g. Kirrinnis (1998), Pan (1996) and Renegar (1987) let all roots (zeros) z_j of p satisfy the bounds

$$|z_j| \leq 1, \qquad j = 1, \ldots, n. \tag{1.1.5}$$

"op" is an arithmetic operation, a comparison of two real numbers, or the computation of the values $|z|$ and $|z|^{1/k}$ for a complex z and a positive integer k. $\psi = r^+/r_-$ is the *relative width* of the annulus A in (1.1.2) and the *isolation ratio* of its internal disc

$$D = D(X, r_-) = \{x : |x| \leq r_-\}; \tag{1.1.6}$$

this disc is called *ψ-isolated* (Pan, 1987, 1997). Hereafter, log stands for $\log_2$.

The root-finders in Pan (1995a, 1996) recursively combine *preprocessing* and *splitting*. Preprocessing algorithms compute the basic annulus A for balanced splitting with relative width

$$\psi = \frac{r^+}{r_-} \geq 1 + \frac{c}{n^d} \tag{1.1.7}$$

for two real constants $c > 0$ and d. Splitting algorithms first compute a *crude initial splitting* and then *refine* it by means of Newton's iterative process. They have been completely developed in several papers (see Delves and Lyness, 1967; Grau, 1971; Schönhage, 1982b; McNamee, 1993, 1997; Kirrinnis, 1998).

Let us first state the basic splitting results of Schönhage and Kirrinnis and then our main splitting theorem.

Theorem 1.1.1. (Kirrinnis, 1998 and Schönhage, 1982b) *Suppose we are given a polynomial p of* (1.1.1), (1.1.5)*, a positive integer $k, k < n$, real $c > 0, d$,*

$$N = N(n, d) = \begin{cases} n & \textit{for } d \leq 0, \\ n \log n & \textit{for } d > 0, \end{cases} \tag{1.1.8}$$

$\theta > 1$, and $b \geq N$, an annulus $A = A(X, r_-, r^+)$ of (1.1.2), (1.1.7) *such that*

$$|z_j| \leq r_- \quad \textit{for } j \leq k, \qquad |z_j| \geq r^+ \quad \textit{for } j > k, \tag{1.1.9}$$

and two polynomials $\widetilde{F}$ (monic, of degree k, with all its roots lying in the disc $D = D(X, r_-)$) and $\widetilde{G}$ (of degree $n - k$, with all its roots lying outside the disc $D(X, r^+)$) satisfying

$$|p - \widetilde{F}\widetilde{G}| \leq 2^{-\tilde{c}N}|p| \tag{1.1.10}$$

for a fixed and sufficiently large constant $\tilde{c}$. Then it is sufficient to perform $O((n \log n)$ $\log b)$ ops with $O(b)$-bit precision or $O(\mu(bn))$ bit-operations for $\mu(b)$ of (1.1.4)*, to compute the coefficients of two polynomials $F^* = F^*(x)$ (monic, of degree k, and having all its roots lying in the disc $D = D(X, \theta r_-)$) and $G^* = G^*(x)$ (of degree $n - k$ and having all its roots lying outside the disc $D(X, r_+/\theta)$) such that*

$$|p - F^*G^*| \leq 2^{-b}|p|. \tag{1.1.11}$$

We use an equivalent version of the theorem where we relax assumption (1.1.5) and linearly transform the variable x (we shift X into the origin) to ensure that

$$X = 0, \qquad qr_- = 1, \qquad q = r^+, \qquad \psi = q^2 \tag{1.1.12}$$

for some $q > 1$. Then each of the concentric annuli $A(0, r_1, r_2)$ for $r_- \leq r_1 \leq r_2 \leq r^+$, including the unit circle $A(0, 1, 1) = C(0, 1) = \{x, |x| = 1\}$, splits the polynomial p and separates the factors F^* and G^* from one another. The computation of the factors F^* and G^* in Theorem 1.1.1 satisfying (1.1.11) is called *splitting the polynomial p over the unit circle* as well as over any root-free annulus containing this circle. Similarly we define splitting over any other circle or root-free annulus on the complex plane.

THEOREM 1.1.2. (SCHÖNHAGE, 1982b) (CF. DELVES AND LYNESS, 1967; MCNAMEE, 1993, 1997, AND APPENDIX A OF PAN, 1995b) *Given a polynomial* p *of* (1.1.1)*, a real* N*, and an annulus* $A = A(0, r_-, r^+)$ *such that* (1.1.2), (1.1.7)–(1.1.9), (1.1.12) *hold, it is sufficient to perform* $O(M \log M)$ *ops with* $O(N)$*-bit precision or* $O((M \log M)\mu(N))$ *bit-operations to compute a pair of initial splitting polynomials* $\widetilde{F}$ *(monic, of degree* k*, with all roots lying in the disc* $D = D(0, 1/\psi)$*) and* $\widetilde{G}$ *(of degree* $n-k$*, and with all roots lying outside* $D(0, \psi)$*) satisfying* (1.1.10)*. Here, we have (cf.* (1.1.7)*)*

$$M = n + N/(\psi - 1) = \begin{cases} O(n) & \text{for } d \le 0, \\ O(n^{1+d} \log n) & \text{for } d > 0. \end{cases}$$

Thus the initial splitting can be computed in nearly optimal time (up to a polylog factor) if $d \le 0$ but not so if $d > 0$. Theorems 1.1.1 and 1.1.2 are implicit in Kirrinnis (1998) and Schönhage (1982b), although the stated assumptions are slightly different and the op/precision count is not included. As in Pan (1995a, 1996), we rely on a lifting/descending process to reduce the case of a positive d to the case of $d = 0$ but yield a substantially stronger result, that is, by the factor of n we improve the bounds on the computational precision and the Boolean cost vs. Pan (1995a, 1996).

THEOREM 1.1.3. *Under the assumptions of Theorem* 1.1.2*, it is sufficient to perform* $O((n \log n)\ (\log^2 n + \log b))$ *ops with* $O(b)$*-bit precision to compute the coefficients of the two polynomials* F^* *and* G^* *of Theorem* 1.1.1 *satisfying* (1.1.11)*. The computation involves* $O((\mu(b) n \log n)\ (\log^2 n + \log b))$ *bit-operations.*

Technically, in this part we refine the analysis of the lifting/descending process, which, in spite of its crucial role in the design of nearly optimal polynomial root-finders, remains essentially unknown to the computer algebra community.

Our analysis (cf. also Bini and Pan, to appear) is technically involved but finally reveals surprising numerical stability (in terms of the asymptotic relative errors of the order of 2^{-cn}) of the Padé approximation (provided that the zeros of the input analytic function are isolated from its poles) and of Graeffe's lifting process, and this observation is a springboard for our current progress in polynomial factorization and root-finding.

1.2. A LIFTING/DESCENDING PROCESS

Theorem 1.1.2 enables splitting a polynomial p at a low cost over a circle well isolated from the zeros of p. Theorem 1.1.3 partly relaxes the isolation requirement. It is sufficient if the circle lies in even a narrow root-free annulus. Then we recursively apply the so-called Graeffe's root-squaring process, whereby each step squares the isolation ratio of the annulus. This process was discovered by Dandelin, rediscovered by Lobachevsky shortly afterwards, and later on was popularized by Graeffe (Householder, 1970). In $O(\log n)$ steps, the relative width of the root-free annulus grows from $1 + c/n^d$ to 4 and above, and we may apply Theorem 1.1.2 to split the lifted polynomial into two factors. Then we recursively descend down to the original polynomial by reversing Graeffe's lifting process. We observe that the input of every Graeffe's step is defined by the $(n-k, k)$ entry of the Padé approximation table for a rational function formed by the output of this step. This immediately reduces every descending step to the computation of a Padé approximation. It is known that this computation has small arithmetic cost, but it is quite surprising that its asymptotic Boolean (bit-operation) cost turns out to be low as well. A technical description of the algorithm supporting Theorem 1.1.3 follows next, and then we estimate its arithmetic cost (in this section) and Boolean cost (in Section 1.3).

Algorithm 1.2.1. *Recursive lifting, splitting, and descending.*

Input: *Positive* c, r_-, r^+, *real* $\tilde{c}$ *and* d, *and the coefficients of a polynomial* p *satisfying* (1.1.1), (1.1.7), (1.1.9), and (1.1.12).

Output: *Polynomials* F^* (*monic and of degree* k) *and* G^* (*of degree* $n-k$), *split by the unit circle* $C(0,1)$ *and satisfying bound* (1.1.10) *for* $\epsilon = 2^{-\tilde{c}N}$ *and* N *of* (1.1.8).

Computation: *Stage 1* (*recursive lifting*). *Write* $q_0 = p/p_n$, *compute the integer*

$$u = \lceil d \log n + \log(2/c) \rceil, \tag{1.2.1}$$

and u *times recursively apply Graeffe's root-squaring step*

$$q_{l+1}(x) = (-1)^n q_l(-\sqrt{x}) q_l(\sqrt{x}), \qquad l = 0, 1, \ldots, u-1. \tag{1.2.2}$$

(*Note that* $q_l = \prod_{i=1}^{n}(x - z_i^{2^l})$, $l = 0, 1, \ldots, u$, *so* $D(0,1)$ *is a* ψ^{2^l}*-isolated disc for* q_l, *for all* l.)

Stage 2 (*splitting* q_u). *Deduce from* (1.2.1) *that* $\psi^{2^u} > 4$ *and apply Theorem* 1.1.2 *to split the polynomial* $p_u = q_u/|q_u|$ *into two factors* F_u^* *and* $\widetilde{G}_u$ *over the unit circle such that*

$$|q_u - F_u^* G_u^*| = \epsilon_u |q_u|, \qquad \epsilon_u \le 2^{-CN} \tag{1.2.3}$$

for $G_u^* = |q_u| \widetilde{G}_u$ *and a sufficiently large constant* $C = C(c, d)$.

Stage 3 (*recursive descending*). *Start with the latter splitting of* q_u *and recursively split the polynomials* q_{u-j} *of* (1.2.2) *over the unit circle, for* $j = 1, \ldots, u$. *Output the computed approximations* $F^* = F_0^*$ *and* $G^* = p_n G_0^*$ *to the two factors* F *and* G *of the polynomial* $p = p_n q_0 = FG$. (*The approximation error bounds are specified later on.*)

Remark 1.2.2. The algorithm applies Theorem 1.1.2 only at Stage 2, where the computations are not costly because the zeros of the polynomial p_u are isolated from the unit circle, due to (1.1.7), (1.1.9), and (1.1.12) for $1/(\psi - 1) = O(1)$ and p replaced by p_u.

Let us specify Stage 3.

Stage 3 (recursive descending). Step j, $j = 1, 2, \ldots, u$.

Input: The polynomial q_{u-j} (computed at Stage 1) and the computed approximations F_{u-j+1}^* and G_{u-j+1}^* to the factors F_{u-j+1} and G_{u-j+1} of the polynomial q_{u-j+1}, which is split over the unit circle. (The approximations are computed at Stage 2 for $j = 1$ and at the preceding, $(j-1)$st, descending step of Stage 3 for $j > 1$.)

Computation: Approximate the pair of polynomials $F_{u-j}(x)$ and $G_{u-j}(-x)$ as the pair filling the $(k, n-k)$th entry of the Padé approximation table for a fixed meromorphic function. That is, given polynomials q_{u-j} and G_{u-j+1}^* (approximating the factor G_{u-j+1} of q_{u-j+1}), first approximate the polynomial $M_{u-j}(x) \bmod x^{n+1}$,

$$\begin{aligned} M_{u-j}(x) &= q_{u-j}(x)/G_{u-j+1}(x^2) \\ &= (-1)^{n-k} F_{u-j}(x)/G_{u-j}(-x). \end{aligned} \tag{1.2.4}$$

Then solve the Padé approximation problem (cf. Problem 5.2b (PADÉ) in Bini and Pan (1994, Chapter 1) or Problem 2.9.2 in Pan (2001a)) for the input polynomial

$M_{u-j}(x) \bmod x^{n+1}$ to obtain the polynomials $F^*_{u-j} = F^*_{u-j}(x)$ (approximating F_{u-j}), $G^*_{u-j}(-x)$, and thus $G^*_{u-j} = G^*_{u-j}(x)$ (approximating G_{u-j}) such that

$$|F^*_{u-j}G^*_{u-j} - q_{u-j}| = \epsilon_{u-j}|q_{u-j}|, \qquad \epsilon_{u-j} \leq 2^{-\tilde{c}N}, \tag{1.2.5}$$

for $\tilde{c}$ of (1.1.10), where $q_{u-j} = F_{u-j}G_{u-j}$, $\deg F^*_{u-j} = k$, the polynomial F^*_{u-j} is monic, and $\deg G^*_{u-j} \leq n-k$. Then improve the computed approximations of F_{u-j} by F^*_{u-j} and of G_{u-j} by G^*_{u-j}, by applying Theorem 1.1.1 with p replaced by q_{u-j}, F^* by F^*_{u-j}, and G^* by G^*_{u-j}. In the refinement, ϵ_{u-j} remains the value of the order of $1/2^{O(n \log n)}$ for $j < u$, whereas the bound $\epsilon_0 < 2^{-b}$ is ensured at the last (uth) step.

Output of step j: Of the two computed factors, F^*_{u-j} and G^*_{u-j}, only the latter one is used at the subsequent descending step, although at the last step, both F^* and G^* are output. Having completed step $j = u$, stop; for $j < u$, go to the $(j+1)$st step.

The equations

$$G_{u-j+1}(x^2) = (-1)^{n-k}G_{u-j}(x)G_{u-j}(-x)$$

and

$$\gcd(F_{u-j}(x), G_{u-j}(-x)) = 1$$

together with (1.2.4) immediately imply the correctness of Algorithm 1.2.1 performed with infinite precision and no rounding errors provided bound (1.2.5) holds true for $\epsilon_{u-j} = 0$ for all j (that is, $F^*_{u-j} = F_{u-j}$, $G^*_{u-j} = G_{u-j}$ for all j).

We next estimate the arithmetic complexity of Algorithm 1.2.1.

Stage 1: $O(un \log n) = O(n \log^2 n)$ ops at the $u = O(\log n)$ lifting steps, each is a polynomial multiplication (we use the FFT based algorithms).

Stage 2 (for $\epsilon_u = 1/2^{O(n \log n)}$): a total of $O(n \log^2 n)$ ops, by Theorems 1.1.1 and 1.1.2.

Stage 3: $O(n \log^2 n)$ ops for the computation of the polynomials $M_{u-j+1}(x) \bmod x^{n+1}$ for all j, $j = 1, \ldots, u$ (this is polynomial division modulo x^{n+1} for each j) and $O(n \log^3 n)$ ops for the computation of the $(k, n-k)$th entries of the Padé approximation tables for the polynomials $M_{u-j+1}(x) \bmod x^{n+1}$ for $j = 1, \ldots, u$.

For every j, the latter computation is reduced to solving a non-singular Toeplitz linear system of $n-k$ equations (see, e.g., Bini and Pan (1994, Chapter 2, equation (5.6)), for $z_0 = 1$ or Proposition 9.4 where $s(x) = 1$ or Pan (2001a, Algorithm 2.11.1)); the Padé output entry is filled with a non-degenerating pair of polynomials $(F_{u-j}(x), G_{u-j}(-x))$. Non-singularity and non-degeneration follow because the polynomials $F_{u-j}(x)$ and $G_{u-j}(-x)$ have no common roots (zeros) and, therefore, have only constant common divisors; we extend this property to their approximations in the next section. The input coefficients of the auxiliary non-singular Toeplitz linear systems (each of $n-k$ equations) are exactly the coefficients of the input polynomial $M_{u-j}(x) \bmod x^{n+1}$ of the Padé approximation problem.

To solve the u Toeplitz linear systems (where $u = O(\log n)$), we first symmetrize them and then apply the MBA algorithm of Morf and Bitmead/Anderson (cf. Bini and Pan, 1994, Chapter 2, Theorem 13.1; Pan, 2001a, Chapter 5 or Pan and Wang, 2002). The symmetrization ensures positive definiteness and therefore weak numerical stability (Bunch, 1985). $O(n \log^3 n)$ ops are sufficient in the u steps of Stage 3. Summarizing, we arrive at the *arithmetic cost estimates* of Theorem 1.1.3.

We perform all computations by Algorithm 1.2.1 with the precision of $O(n \log n)$ bits except for the refinement of the approximate initial splitting of the polynomial $q_0(x)$. There, we require (1.1.11) for a fixed $\epsilon = 2^{-b}$, $b \geq N$, and use computations with the

b-bit precision. To prove Theorem 1.1.3, it remains to show that under the cited precision bounds, Algorithm 1.2.1 remains correct, that is, bound (1.2.5) holds for a fixed and sufficiently large $\tilde{c}$. We show this in the next three sections.

1.3. PRECISION AND COMPLEXITY ESTIMATES

Our goal is to prove that the computational precision of $O(N)$ bits and the bounds of the order of 2^{-cN} on the values ϵ_{u-j} of (1.2.5) for $j = 0, 1, \ldots, u$ are sufficient to support Algorithm 1.2.1. We first recall the following theorem.

THEOREM 1.3.1. (SCHÖNHAGE, 1985, THEOREM 2.7) *Let*

$$p = p_n \prod_{j=1}^{n}(x - z_j), \qquad p^* = p_n^* \prod_{j=1}^{n}(x - z_j^*),$$

$$|p^* - p| \le \nu|p|, \qquad \nu < 2^{-7n},$$

$$|z_j| \le 1,\ j = 1, \ldots, k; \qquad |z_j| \ge 1,\ j = k+1, \ldots, n.$$

Then, up to reordering z_j^*, we have

$$|z_j^* - z_j| < 9\sqrt[n]{\nu},\ j = 1, \ldots, k;$$

$$|1/z_j^* - 1/z_j| < 9\sqrt[n]{\nu},\ j = k+1, \ldots, n.$$

By applying the theorem for $p = q_{u-j} = F_{u-j}G_{u-j}$, $p^* = F_{u-j}^*G_{u-j}^*$, we obtain the following result.

COROLLARY 1.3.2. *Let* (1.1.1), (1.1.9), (1.1.12), (1.2.1), *and* (1.2.5) *hold and let* $\epsilon_{u-j} < \min\{2^{-7n}, ((\psi-1)\theta/9)^n\}$ *for all* j *and a fixed* θ, $0 \le \theta < 1$. *Then for all* j, $j = 0, 1, \ldots, u$, *all roots of the polynomials* $F_{u-j}^*(x)$ *and the reciprocals of all roots of the polynomials* $G_{u-j}^*(x)$ *lie inside the disc* $D(0, \theta + (1-\theta)/\psi)$. *For* $\psi - 1 \ge c/n^d$, $c > 0$, *the latter properties of the roots are ensured already where* $\epsilon_{u-j} \le 1/n^{O(N)}$ *for all* j.

Let us estimate the error of splitting $q_{u-j}(x)$ in terms of the approximation error for splitting $q_{u-j+1}(x)$.

PROPOSITION 1.3.3. *Suppose that* $|F_{u-j+1}^*G_{u-j+1}^* - q_{u-j+1}| \le \epsilon_{u-j+1}|q_{j-j+1}|$ *for some real* ϵ_{u-j+1} *and a monic polynomial* F_{u-j+1}^* *of degree* k. *Let the Padé approximation problem be solved exactly* (*with infinite precision and no rounding errors*) *for the input polynomial* $M_{u-j}^*(x) = (q_{u-j}(x)/G_{u-j+1}^*(x^2)) \bmod x^{n+1}$. *Let* F_{u-j}^*, G_{u-j}^* *denote the solution polynomials and let* ϵ_{u-j} *be defined by* (1.2.5). *Then we have* $|F_{u-j}^* - F_{u-j}| + |G_{u-j}^* - G_{u-j}| \le \epsilon_{u-j+1}2^{O(n\log n)}$, $\epsilon_{u-j} = \epsilon_{u-j+1}2^{O(n\log n)}$.

Due to the latter proposition, it is sufficient to choose the value ϵ_{u-j} of (1.2.5) of the order of $\epsilon_{u-j+1}2^{-\tilde{c}N}$ for a large positive $\tilde{c}$ to ensure splitting q_{u-j} within an error bound (1.1.10), that is, small enough to allow the subsequent refinement based on Theorem 1.1.1.

The next theorem of independent interest is used in the proof of Proposition 1.3.3. It estimates the perturbation error of the Padé approximation problem. Generally, the input perturbation causes unbounded output errors but in our special case the roots of the output pair of polynomials are isolated from the unit circle.

THEOREM 1.3.4. *Let us be given two integers, k and n, $n > k > 0$, three positive constants C_0, γ, and ψ,*

$$\psi > 1, \tag{1.3.1}$$

and six polynomials F, f, G, g, M and m. Let the following relations hold:

$$F = \prod_{i=1}^{k}(x - \hat{z}_i), \qquad |\hat{z}_i| \le 1/\psi, \quad i = 1, \dots, k, \tag{1.3.2}$$

$$G = \prod_{i=k+1}^{n}(1 - x/\hat{z}_i), \qquad |\hat{z}_i| \ge \psi, \quad i = k+1, \dots, n \tag{1.3.3}$$

(*cf.* (1.1.9), (1.1.12)),

$$F = MG \bmod x^{n+1}, \tag{1.3.4}$$

$$F + f = (M + m)(G + g) \bmod x^{n+1}, \tag{1.3.5}$$

$$\deg f \le k, \tag{1.3.6}$$

$$\deg g \le n - k, \tag{1.3.7}$$

$$|m| \le \gamma^n (2 + 1/(\psi - 1))^{-C_0 n}, \qquad \gamma < \min\{1/128, (1 - 1/\psi)/9\}. \tag{1.3.8}$$

Then there exist two positive constants C and C^* independent of n and such that if $|m| \le (2 + 1/(\psi - 1))^{-Cn}$, then

$$|f| + |g| \le |m|(2 + 1/(\psi - 1))^{C^* n}. \tag{1.3.9}$$

The proof of Theorem 1.3.4 is elementary but quite long. It is given in the next two sections, where we also specify the constant C_0.

PROOF OF PROPOSITION 1.3.3. The relative error norms ϵ_{u-j} and ϵ_{u-j+1} are invariant in scaling the polynomials. For convenience, we drop all the subscripts of F, F^*, G, q and q^* and use scaling that makes the polynomials F, F^*, $G_{\text{rev}} = x^{n-k}G(1/x)$, and $G^*_{\text{rev}} = x^{n-k}G^*(1/x)$ monic, that is, $F = \prod_{j=1}^{k}(x - z_j)$, $F^* = \prod_{j=1}^{k}(x - z_j^*)$, $G = \prod_{j=k+1}^{n}(1 - x/z_j^*)$, $G^* = \prod_{j=k+1}^{n}(1 - x/z_j^*)$, $q = FG$, $q^* = F^*G^*$. The polynomials q and q^* are no longer assumed to be monic (see Remark 1.3.5.). Furthermore, by (1.3.1)–(1.3.3) and Corollary 1.3.2, we may assume that $|z_j| < 1$, $|z_j^*| < 1$, for $j \le k$, whereas $|z_j^*| > 1$, $|z_j| > 1$, for $j > k$. Therefore, $1 \le |F| < 2^k$, $1 \le |F^*| < 2^k$, $1 \le |G| < 2^{n-k}$, $1 \le |G^*| < 2^{n-k}$, $1 < |q| < 2^n$, $1 < |q^*| < 2^n$.

For any positive r, let us deduce that

$$\begin{aligned} &\left\|\frac{1}{G_{u-j+1}(x)} \bmod x^{r+1}\right\| \le \|(1 - x)^{k-n} \bmod x^{r+1}\| \\ &= \sum_{i=0}^{r} \binom{n - k + i - 1}{n - k - 1} = \binom{n - k + r}{r} < 2^{n-k+r}. \end{aligned} \tag{1.3.10}$$

Indeed, write $(-x)^{n-k}/G_{n-k}(x) = \sum_{i=0}^{\infty} g_i/x^i$. Now, (1.3.10) follows when we observe for each i that $|g_i|$ reaches its maximum where $z_i = 1$, that is, where $(-x)^{n-k}/G_{n-k}(x) = x^{n-k}/(1 - x)^{n-k}$.

Likewise, we have

$$\|(1/G^*_{u-j+1}(x)) \bmod x^r\| < 2^{n-k+r}.$$

We apply a bound of Section 10 of Schönhage (1982b) to obtain that

$$|G^*_{u-j+1} - G_{u-j+1}| \le \epsilon_{u-j+1} 2^{O(N)}.$$

Now write

$$\Delta_{u-j+1} = \left(\frac{1}{G^*_{u-j+1}} - \frac{1}{G_{u-j+1}}\right) = \frac{G_{u-j+1} - G^*_{u-j+1}}{G_{u-j+1}G^*_{u-j+1}},$$

summarize the above estimates, and obtain that

$$\|\Delta_{u-j+1}(x) \bmod x^r\| \le \epsilon_{u-j+1} 2^{O(n \log n)}$$

for $r = O(n)$.

Next write $m_{u-j} = m_{u-j}(x) = (M^*_{u-j}(x) - M_{u-j}(x)) \bmod x^{n+1}$ and combine our latter bound with (1.2.4) and with the bound $|q_{u-j}| \le 2^n$ to obtain that $|m_{u-j}| \le \epsilon_{u-j+1} 2^{O(N)}$. By combining this estimate with the ones of Theorem 1.3.4, we obtain the first bound of Proposition 1.3.3,

$$\Delta_{F,G} = |F^*_{u-j} - F_{u-j}| + |G^*_{u-j} - G_{u-j}| \le \epsilon_{u-j+1} 2^{O(N)}.$$

Now we easily deduce the second bound,

$$\begin{aligned}
\epsilon_{u-j} &= |F^*_{u-j} G^*_{u_j} - F_{u-j} G_{u-j}| \\
&\le |F^*_{u-j}(G^*_{u-j} - G_{u-j}) + (F^*_{u-j} - F_{u-j}) G_{u-j}| \\
&\le |F^*_{u-j}| \cdot |G^*_{u-j} - G_{u-j}| + |F^*_{u-j} - F_{u-j}| \cdot |G_{u-j}| \\
&\le \max\left\{|F^*_{u-j}|, |G_{u-j}|\right\} \Delta_{F,G} \\
&\le \epsilon_{u-j+1} 2^{O(N)}. \ \square
\end{aligned}$$

Similarly to Proposition 1.3.3, we may prove that any perturbation of the coefficients of the polynomial q_{u-j} within the relative norm bound of the order of $1/2^{O(N)}$ causes a perturbation of the factors of q_{u-j+1} within the relative error norm of the same order.

Proposition 1.3.3 and Theorem 1.3.4 together show that the relative errors of the order of $O(N)$ bits do not propagate in the descending process of Stage 3 of Algorithm 1.2.1.

To complete the proof of Theorem 1.1.3, it remains to show that the relative precision of $O(N)$ bits for the output of the descending process of Algorithm 1.2.1 can be supported by the computations with rounding to the precision of $O(N)$ bits. To yield this goal, one may apply the tedious techniques in Schönhage (1982a) (cf. also Schönhage, 1982b; Kirrinnis, 1998). Alternatively we apply the backward error analysis to all the polynomial multiplications and divisions involved, to simulate the effect of rounding errors of these operations by the input perturbation errors. This leads us to the desired estimates simply via the invocation of Theorem 1.3.4 and Proposition 1.3.3, except that we need some distinct techniques at the stages of the solution of Toeplitz linear systems of equations associated with the Padé problem.

To extend our analysis to these linear systems, we recall that they are non-singular because the Padé problem does not degenerate in our case. Moreover, Theorem 1.3.4 bounds the condition number of the problem. Furthermore, we solve the Padé problem by applying the cited MBA algorithm to the symmetrized linear systems. (The symmetrization squares the condition number, which requires doubling the precision of the computation, but this is not substantial for proving our estimate of $O(N)$ bits.) We then recall that the algorithm only operates with some displacement generators of the matrices appearing in a recursive block triangular factorization of the matrices defined by the entries of the Padé input, $M^*_{u-j}(x) \bmod x^{n+1}$, and that this algorithm is proved to be sufficiently stable numerically (Bunch, 1985). It follows that $O(N)$-bit precision of the computation is sufficient at the stages of solving Padé problems as well, and we arrive at Theorem 1.1.3. □

Remark 1.3.5. One could have expected an even greater increase of the precision required at the lifting steps of (1.2.2). Indeed, these steps generally cause rapid growth of the ratio of the absolutely largest and the absolutely smallest coefficients of the input polynomial. The growth, however, does not affect the precision of computing because all our error norm bounds are relative to the norms of the polynomials. Technically, to control the output errors, we apply scaling, to make the polynomials F, F^*, $G_{\rm rev}$ and $G^*_{\rm rev}$ monic, and then continue as in the proof of Proposition 1.3.3, where the properties (1.1.9) of the roots of the input polynomials are extended to the approximations to the roots, due to Corollary 1.3.2.

Remark 1.3.6. Consider modifications of the descending stage of Algorithm 1.2.1 based on either or both of the two following equations applied for all j:

$$\begin{aligned} F_{u-j}(x) &= \gcd(q_{u-j}(x), F_{u-j+1}(x^2)), \\ G_{u-j}(x) &= \gcd(q_{u-j}(x), G_{u-j+1}(x^2)), \qquad j = 1, \ldots, u. \end{aligned}$$

Here and hereafter, $\gcd(u(x), v(x))$ denotes the monic greatest common divisor (gcd) of the two polynomials $u(x)$ and $v(x)$. In this modification of Algorithm 1.2.1, the Padé computation is replaced by the polynomial gcd computation. This produces the same output as in Algorithm 1.2.1 if we assume infinite precision of computing. The approach was originally introduced in the proceedings paper (Pan, 1995a) but in its journal version (Pan, 1996) was replaced by the one based on the Padé computation. The replacement enabled more direct control over the propagation of the perturbation errors (cf. Theorem 1.3.4). We refer the reader to Brent *et al.* (1980), Bini and Pan (1994) and Pan (2001a) on the correlation among both approaches and the solution of the associated Toeplitz linear system of equations. To yield a fast solution based on the gcd approach and without reduction to a Toeplitz system, one should apply the fast Euclidean algorithm, as proposed in Pan (1995a) and then in Neff and Reif (1996). In this case, however, each descending step (1.2.4) is replaced by a recursive Euclidean process, prone to severe problems of numerical stability (cf. Schönhage, 1985; Emiris *et al.*, 1996, 1997) and to blowing up the precision and the Boolean cost of the computations. In Bini and Pan (to appear) we give some critical comments on an attempt to avoid this problem made in Neff and Reif (1996).

1.4. Perturbation error bounds for Padé approximation

Corollary 1.4.7, to be proved in this section, implies Theorem 1.3.4 in the case where assumption (1.3.6) is replaced by the inequality

$$\deg f < k. \tag{1.4.1}$$

We need some auxiliary estimates.

Proposition 1.4.1. (Mignotte, 1974) *If* $p = p(x) = \prod_{i=1}^{l} f_i, \deg p \leq n$, *and all* f_i *are polynomials, then* $\prod_{i=1}^{l} |f_i| \leq 2^n \max_{|x=1|} |p(x)| \leq 2^n |p|_2$.

The next two results extend the ones of Schönhage (1982b).

Proposition 1.4.2. *For a fixed pair of scalars,* $\psi \geq 1$ *and* β, *let*

$$p = \beta \prod_{i=1}^{k}(x - z_i) \prod_{i=k+1}^{n} (1 - x/z_i),$$

where $|z_i| \leq 1/\psi$ *for* $i \leq k$; $|z_i| \geq \psi$ *for* $i > k$ (*cf.* (1.1.9), (1.1.12)). *Then*

$$|\beta| \geq |p|/(1 + 1/\psi)^n.$$

Proof. The assumed factorization of the polynomial p yields the inequality

$$|p|/|\beta| \leq \left(\prod_{i=1}^{k} |x - z_i|\right) \prod_{i=k+1}^{n} \|1 - x/z_i\|,$$

where neither of the n factors on the right-hand side exceeds $1 + 1/\psi$. □

Proposition 1.4.3. *Let* (1.1.9) *hold for some* $r^+ = \psi \geq 1$, $r_- = 1/\psi$. *Then*

$$|p|\left(\frac{\psi - 1}{\psi + 1}\right)^n \leq \min_{|x|=1} |p(x)| \leq |p|.$$

Proof. The upper bound on $\min_{|x|=1} |p(x)|$ is obvious. To prove the lower bound, recall the equation of Proposition 1.4.2 and deduce that

$$|p(x)| \geq |\beta| \prod_{i=1}^{k} |x - z_i| \prod_{i=k+1}^{n} |1 - x/z_i| \text{ for all } x.$$

Substitute the bounds $|x| = 1$, (1.1.9) and (1.1.12) and obtain that

$$|p(x)| \geq (1 - 1/\psi)^n |\beta|.$$

Now substitute the bound on $|\beta|$ of Proposition 1.4.2 and arrive at Proposition 1.4.3. □

Proposition 1.4.4. *Let* $f(x)$ *and* $F(x)$ *be two polynomials having degrees at most* $k - 1$ *and* k, *respectively. Let* $R(x)$ *be a rational function having no poles in the disc* $D(0, 1) = \{x, |x| \leq 1\}$. *Then, for any complex* x, *we have*

$$\int_{|x|=1} R(t)\frac{F(t) - F(x)}{t - x} dt = 0,$$

and if $F(x) \neq 0$ *for* $|x| = 1$, *then*

$$f(x) = \frac{1}{2\pi\sqrt{-1}} \int_{|x|=1} \frac{f(t)}{F(t)} \cdot \frac{F(x) - F(t)}{x - t} dt.$$

Proof. (Cf. Pólya and Szegö, 1925, III, Chapter 4, No.163; Kirrinnis, 1992, Proof of Lemma 4.6) The first equation of Proposition 1.4.4 immediately follows from the Cauchy theorem on complex contour integrals of analytic functions (Ahlfors, 1979). Cauchy's integral formula (Ahlfors, 1979) implies the second equation of Proposition 1.4.4 for every x equal to a zero of $F(x)$. If $F(x)$ has k distinct zeros, then the second equation is extended identically in x, since $f(x)$ has a degree less than k. The confluence argument enables us to extend the result to the case of a polynomial $F(x)$ having multiple zeros. □

Now we are prepared to estimate the norms $|f|$ and $|g|$ from above.

Proposition 1.4.5. *Let a constant ψ and six polynomials F, f, G, g, M and m satisfy relations* (1.3.1)–(1.3.7), (1.4.1). *Let*

$$v(x) = (G(x) + g(x))G(x)m(x) \bmod x^{n+1}, \qquad \deg v \le n. \tag{1.4.2}$$

Then we have

$$|f| \le k\tau|F|, \qquad \tau = \max_{|x|=1} \left| \frac{v(x)}{F(x)G(x)} \right|.$$

Proof. Subtract (1.3.4) from (1.3.5) and obtain that

$$f(x) = (M(x) + m(x))g(x) + m(x)G(x) \bmod x^{n+1}.$$

Multiply this equation by the polynomial G and substitute

$$F(x) = G(x)M(x) \bmod x^{n+1}$$

into the resulting equation, to arrive at the equation

$$G(x)f(x) = F(x)g(x) + (G(x) + g(x))G(x)m(x) \bmod x^{n+1}.$$

Observe that $\deg(Gf - Fg) \le n$, due to (1.3.2), (1.3.3), (1.3.6) and (1.3.7), and deduce that

$$Gf = Fg + v, \tag{1.4.3}$$

for the polynomial v of (1.4.2). It follows that

$$f = \frac{gF}{G} + \frac{v}{G}.$$

Combine the latter equation with Proposition 1.4.4 for $R(t) = g(t)F(t)/G(t)$ and deduce that

$$f = \frac{1}{2\pi\sqrt{-1}} \int_\Gamma \frac{v(t)}{F(t)G(t)} \cdot \frac{F(x) - F(t)}{x - t} dt.$$

Proposition 1.4.5 follows from this equation applied to the polynomial f coefficient-wise. □

Let us further refine our bound on $|f|$. Combine (1.3.2) and (1.3.3) with Proposition 1.4.3 and obtain that $\min_{|x|=1}|F(x)G(x)| \ge \frac{(\psi-1)^n}{(\psi+1)^n}|p|$. Now, because $\max_{|x|=1}|v(x)| \le |v|$, obtain from Proposition 1.4.5 that

$$|f| \le k|F| \cdot |v|/\phi_-^n|p|, \quad \phi_- = (\psi - 1)/(\psi + 1) = 1 - 2/(\psi + 1). \tag{1.4.4}$$

Next, let us bound the norm $|g|$ from above.

Proposition 1.4.6. *Assume relations* (1.3.1)–(1.3.7) *and* (1.4.1)–(1.4.3). *Then*

$$|g| \le 2^n \phi_+^{n-k}(|f| + \phi_+^{n-k}|m|)/(1 - 2^n \phi_+^{n-k}|m|),$$

where $\phi_+ = 1 + 1/\psi < 2$.

Proof. Combine the relations $\deg g \le n-k$ and $\deg F = k$ (cf. (1.3.2) and (1.3.7)) with Proposition 1.4.1 for $l = 2$ and obtain the bound $|F|\cdot|g| \le 2^n|Fg|$. Therefore, $|g| \le 2^n|Fg|$ because $|F| \ge 1$ (see (1.3.2)). On the other hand, (1.4.3) implies that $|Fg| \le |G|\cdot|f| + |v|$. Combine the two latter bounds to obtain that $|g| \le 2^n(|G|\cdot|f|+|v|)$. Deduce from (1.4.2) that $|v| \le |G+g|\cdot|G|\cdot|m|$. Substitute the bound $|G| \le \phi_+^{n-k}, \phi_+ = 1 + 1/\psi$, implied by (1.3.3), and deduce that

$$|v| \le (\phi_+^{n-k} + |g|)\phi_+^{n-k}|m|, \qquad |g| \le 2^n \phi_+^{n-k}(|f| + (\phi_+^{n-k} + |g|)|m|). \tag{1.4.5}$$

Therefore, we have

$$(1 - 2^n \phi_+^{n-k}|m|)|g| \le 2^n \phi_+^{n-k}(|f| + \phi_+^{n-k}|m|),$$

and Proposition 1.4.6 follows. □

Corollary 1.4.7. *Assume relations* (1.3.1)–(1.3.7), (1.4.1) *and let*

$$2^n \phi_+^{n-k}|m| \le 1/2, \tag{1.4.6}$$

$$k2^n(\phi_+/\phi_-)^n \phi_+^{n-k}|m| \le |p|/4. \tag{1.4.7}$$

Then we have

$$|f| \le 4k(\phi_+/\phi_-)^n \phi_+^{n-k}|m|/|p|, \tag{1.4.8}$$

$$|g| \le 2^{n+1}(1 + 4k(\phi_+/\phi_-)^n/|p|)\phi_+^{2n-2k}|m|, \tag{1.4.9}$$

$$1 \le |p| \le \phi_+^n \tag{1.4.10}$$

for $\phi_- = 1 - 2/(\psi+1)$ *of* (1.4.4) *and for*

$$\phi_+ = 1 + 1/\psi < 2, \qquad \phi_+/\phi_- = (\psi+1)^2/((\psi-1)\psi). \tag{1.4.11}$$

Proof. Combine Proposition 1.4.6 with inequality (1.4.6) and obtain that

$$|g| \le 2^{n+1}(|f| + \phi_+^{n-k}|m|)\phi_+^{n-k}. \tag{1.4.12}$$

Combine (1.4.4), (1.4.5), and the bound $|F| \le \phi_+^k$, implied by (1.3.2), and obtain that

$$|f| \le k(\phi_+/\phi_-)^n(\phi_+^{n-k} + |g|)|m|/|p|.$$

Combining the latter inequality with (1.4.12) implies that

$$|p|\cdot|f| \le k(\phi_+/\phi_-)^n(1 + 2^{n+1}(|f| + \phi_+^{n-k}|m|))\phi_+^{n-k}|m|.$$

Therefore,

$$|f|\cdot(|p| - k2^{n+1}(\phi_+/\phi_-)^n|m|\phi_+^{n-k}) \le \ k(\phi_+/\phi_-)^n(1 + 2^{n+1}\phi_+^{n-k}|m|)\phi_+^{n-k}|m|.$$

Substitute (1.4.6) on the right-hand side and (1.4.7) on the left-hand side and obtain (1.4.8). Combine (1.4.8) and (1.4.12) and obtain (1.4.10). Combine (1.3.2) and (1.3.3) and obtain (1.4.9). □

1.5. Local non-singularity of Padé approximations

In this section, we prove Theorem 1.3.4 by using the following immediate consequence of Corollary 1.4.7.

Corollary 1.5.1. *Let all the assumptions of Theorem* 1.3.4 *hold, except possibly for* (1.3.6)*, and let relations* (1.4.1), (1.4.6) and (1.4.7) *hold. Then bound* (1.3.9) *holds for a sufficiently large constant* C^*.

Due to Corollary 1.5.1, it remains to prove (1.4.1) under (1.3.8) in order to complete the proof of Theorem 1.3.4.

By the Frobenius theorem (Gragg, 1972, Theorem 3.1), there exists a unique rational function F/G satisfying (1.3.4) for any given polynomial M and any pair of integers k and n such that $0 \le k \le n, \deg F \le k, \deg G \le n-k$. Assuming further that the polynomials F and G have no common non-constant factors and that the polynomial F is monic, we uniquely define the pair of the polynomials F and G (unless M is identically 0), which we call *the normalized pair filling the* $(k, n-k)$*th entry* of the Padé table for a polynomial M.

Now, suppose that equations (1.3.1)–(1.3.7) hold and let (F, G) and $(F+f, G+g)$ be two normalized pairs filling the $(k, n-k)$th entry of the Padé table for the meromorphic functions M and $M+m$, respectively, where deg $F = k$. Then, clearly, we have (1.4.1) if and only if

$$\deg(F+f) = k. \tag{1.5.1}$$

Let (F_δ, G_δ) denote the normalized pair filling the $(k, n-k)$th entry of the Padé table for $M+m+\delta$, where δ is a perturbation polynomial. Even if (1.5.1) does not hold, there always exists a sequence of polynomials $\{\delta_h\}, h = 1, 2, \ldots$, such that $|\delta_h| \to 0$ as $h \to \infty$ and

$$\deg F_{\delta_h} = k \text{ for } h = 1, 2, \ldots. \tag{1.5.2}$$

(Indeed, the coefficient vectors of polynomials δ for which $\deg F_\delta < k$ form an algebraic variety of dimension k in the space of the $(k+1)$st dimensional coefficient vectors of all polynomials of degree at most k.)

Due to (1.5.2), we have $\deg f_{\delta_h} < k$. Therefore, we may apply Corollary 1.5.1 to the polynomials $M + m + \delta_h$ and obtain that the coefficient vectors of all polynomials F_{δ_n} and G_{δ_h} are uniformly bounded as follows:

$$|F_{\delta_h} - F| + |G_{\delta_h} - G| \le \left(2 + \frac{1}{\psi - 1}\right)^{C_1 n} |m + \delta_h| \tag{1.5.3}$$

provided that $|m + \delta_h| \le (2 + 1/(\psi-1))^{-C_0 n}$. Because of this bound, there exists a subsequence $\{h(i), i = 1, 2, \ldots\}$ of the sequence $h = 1, 2, \ldots$, for which the coefficient vectors $(\mathbf{F}^T_{\delta_{h(i)}}, \mathbf{G}^T_{\delta_{h(i)}})^T$ of the polynomials $F_{\delta_{h(i)}}, G_{\delta_{h(i)}}$ converge to some $(n+2)$nd dimensional vector $(\mathbf{F}^{*T}, \mathbf{G}^{*T})^T$. Let F^*, G^* denote the associated polynomials and let us write

$$F + f = F^*, \qquad G + g = G^*. \tag{1.5.4}$$

Because $\delta_{h(i)} \to 0$, we immediately extend (1.5.3) and obtain that

$$F^*(x) = (M(x) + m(x))G^*(x) \bmod x^{n+1}$$

and

$$|f| + |g| = |F^* - F| + |G^* - G| \le \left(2 + \frac{1}{\psi - 1}\right)^{C_1 n} |m| \tag{1.5.5}$$

provided that

$$|m| \le \left(2 + \frac{1}{\psi - 1}\right)^{-C_0 n}.$$

To complete the proofs of Theorems 1.3.4 and 1.1.3, it remains to show that deg $f < k$, that is, that deg $F^* = k$ and that the polynomials F^* and G^* of (1.5.4) have only constant common factors. Let us do this by applying Theorem 1.3.1. First combine the bounds (1.4.8) and (1.3.8) (where C_0 satisfies the bound

$$\left(2 + \frac{1}{\psi - 1}\right)^{C_0 n} \ge 4k(\phi_+)^{n-k} \left(\frac{\phi_+}{\phi_-}\right)^n |p|/|F| \tag{1.5.6}$$

for ϕ_- and ϕ_+ in (1.4.4) and (1.4.11)) with Theorem 1.3.1, for p and p^* replaced by F and F^*, respectively, and deduce that the roots of the polynomial $F + f$ deviate from the respective roots of the polynomial F by less than $1 - 1/\psi$, so the polynomial $F + f$ has exactly k roots all lying strictly inside the unit disc $D(0, 1)$. Similarly, obtain that $\deg(G+g) = n-k$ and all the roots of the polynomial $G+g$ lie outside this disc (provided that the constant C_0 in (1.3.8) satisfies the inequality

$$\left(2 + \frac{1}{\psi - 1}\right)^{C_0 n} \ge 2^{n+1} \psi_+^{2n-2k} \left(1 + 4k\left(\frac{\phi_+}{\phi_-}\right)^n \Big/ |p|\right) |m|/|G| \tag{1.5.7}$$

see (1.4.9)), so the polynomial $G+g$ has only constant common factors with $F+f$. This completes the proof of (1.4.1) and, therefore, also the proofs of Theorems 1.3.4 and 1.1.3.

1.6. EXTENSION TO SPLITTING OVER ANY CIRCLE

By the initial scaling of the variable, we may move the roots of a given polynomial into the unit disc $D(0, 1)$. Therefore, it is sufficient to consider splitting a polynomial p of (1.1.1) (within a fixed error tolerance ϵ) over any disc $D(X, r)$, with X and r satisfying the bounds $r > 0$ and

$$r + |X| \le 1. \tag{1.6.1}$$

To extend the splitting respectively, we shift and scale the variable x and estimate the new relative error norm bound $\tilde{\epsilon}$ as a function in ϵ, X and r. The following result relates ϵ and $\tilde{\epsilon}$.

In this section, we write $\|u(x)\| = \|u(x)\|_1$ and $\|v(y)\| = \|v(y)\|_1$ (rather than $|u|$ and $|v|$) to show the norms of various polynomials $u(x)$ in x and $v(y)$ in y.

PROPOSITION 1.6.1. *Let relations* (1.1.11) and (1.6.1) *hold. Write*

$$y = rx + X, \tag{1.6.2}$$

$$\tilde{p}(y) = \sum_{i=0}^{n} \tilde{p}_i y^i = \tilde{p}(rx + X) = q(x),$$

$$p(x) = q(x)/\|q(x)\|, \tag{1.6.3}$$

$$\widetilde{F}^*(y) = \widetilde{F}^*(rx + X) = F^*(x) r^k,$$

$$\widetilde{G}^*(y) = \widetilde{G}^*(rx + X) = G^*(x)/(\|q(x)\|r^k),$$
$$\Delta(x) = p(x) - F^*(x)G^*(x),$$
$$\widetilde{\Delta}(y) = \tilde{p}(y) - \widetilde{F}^*(y)\widetilde{G}^*(y).$$

Then (1.6.2) *maps the disc* $D(0,1) = \{x : |x| \leq 1\}$ *onto the disc* $D(X,r) = \{y : |y - X| \leq r\}$*; moreover,*

$$\|\widetilde{\Delta}(y)\| \leq \|\Delta(x)\| \cdot ((1 + |X|)/r)^n \cdot \|\tilde{p}(y)\| \leq \|\Delta(x)\| \cdot ((2 - r)/r)^n \cdot \|\tilde{p}(y)\|. \tag{1.6.4}$$

PROOF. See Pan (2001b). □

1.7. ERROR ESTIMATES FOR RECURSIVE SPLITTING

In this section we recall some results from Schönhage (1982b) for the sake of completeness. Suppose that we recursively split each approximate factor of p over the boundary circle of some well isolated disc we arrive at the factors of the form $(ux + v)^d$. This gives us an approximate factorization

$$p^* = p^*(x) = \prod_{j=1}^{n}(u_j x + v_j). \tag{1.7.1}$$

Let us estimate the norm $|\Delta^*|$ of the residual polynomial $\Delta^* = p^* - p$. (An upper bound δ on this norm implies the upper bound $\kappa_i\delta$ on the error of approximation of root z_i of $p(x)$ where κ_i is the condition number of z_i under the same norm $\|\cdot\|_1$.) We begin with an auxiliary result.

PROPOSITION 1.7.1. (SCHÖNHAGE, 1982b, Section 5) *Let*

$$\Delta_k = |p - f_1, \ldots, f_k| \leq k\epsilon|p|/n, \tag{1.7.2}$$
$$\Delta = |f_1 - fg| \leq \epsilon_k|f_1|, \tag{1.7.3}$$

for some non-constant polynomials $f_1, \ldots, f_k, f$ *and* g *and for*

$$\epsilon_k \leq \epsilon|p|/\left(n\prod_{i=1}^{k}|f_i|\right). \tag{1.7.4}$$

Then

$$|\Delta_{k+1}| = |p - fgf_2 \ldots f_k| \leq (k+1)\epsilon|p|/n. \tag{1.7.5}$$

PROOF. See Pan (2001b). □

Write $f_1 = f$, $f_{k+1} = g$. Then (1.7.5) turns into (1.7.2) for k replaced by $k+1$. Now split one of the factors f_i as in (1.7.3), apply Proposition 1.7.1, and recursively split p into factors of smaller degrees until we arrive at (1.7.1), where

$$|\Delta^*| = |p^* - p| \leq \epsilon|p|. \tag{1.7.6}$$

Let us call this computation the *Recursive Splitting Process* provided that it starts with $k = 1$ and $f_1 = p$ and ends with $k = n$.

PROPOSITION 1.7.2. (SCHÖNHAGE, 1982b) *Performing the Recursive Splitting Process for a positive $\epsilon \le 1$, it is sufficient to choose ϵ_k in* (1.7.3) *satisfying*

$$\epsilon_k \le \epsilon/(n2^{n+1}) \tag{1.7.7}$$

for all k to support (1.7.2) *for all $k = 1, 2, \ldots, n$.*

PROOF. See Pan (2001b). □

Part II: Computing Basic Annuli for Splitting

2.1. THE MAIN RESULTS

The algorithms of Part I enable us to reduce the approximation of the roots of a polynomial

$$p(x) = \sum_{i=0}^{n} p_i x^i = p_n \prod_{j=1}^{n} (x - z_j), \qquad p_n \ne 0,$$

in (1.1.1) to the computation of a basic annulus over which the polynomial $p(x)$ can be split effectively into the product of two nonlinear factors, $F(x)$ and $G(x)$. In this part of the paper, we improve the algorithm of Pan (1995a, 1996), which computes a basic annulus and, moreover, ensures that the resulting splitting is *a-balanced*, that is,

$$(1-a)n/2 \le \deg F(x) \le (1+a)n/2, \tag{2.1.1}$$

where a is any fixed constant from the interval

$$5/6 \le a < 1. \tag{2.1.2}$$

We still use the definitions of Part I, including the concepts of *ops* (that is, arithmetic operations + comparisons + the computation of $|z|$ or $|z|^{1/k}$ for complex numbers z and integers $k > 1$), *the splitting of polynomials over (zero-free) annuli or circles, the relative width* $\rho(A) \ge 1$ of an annulus A (the ratio of the radii of its two boundary circles), *ψ-isolated discs* (the internal discs of zero-free annuli having a relative width ψ), and the polynomial *norm* $|u| = \|u(x)\| = \|u(x)\|_1 = \sum_i |u_i|$ for $u(x) = u = \sum_i u_i x^i$ (see (1.1.3)); log still means $\log_2$.

Under the above assumptions, in each step of recursive splitting, we compute two factors of the input polynomial of degree d whose degrees are at most $(1+a)d/2$ (for instance, $11d/12$ for $a = 5/6$). Then we apply the estimates from Part I for the computational complexity of splitting a polynomial over a fixed circle. This gives us upper bounds on the overall arithmetic and Boolean computational cost of the complete factorization of the polynomial $p(x)$ into the product of linear factors and of the approximation of well- and ill-conditioned polynomial zeros. All the bounds are optimal up to polylogarithmic factors.

THEOREM 2.1.1. *Let $p(x) = \sum_{i=0}^{n} p_i x^i = p_n \prod_{j=1}^{n} (x - z_j)$, $p_n \ne 0$, be a polynomial of* (1.1.1) *of degree n given with its coefficients. Let* (1.1.5) *hold, that is,*

$$|z_j| \le 1 \qquad \textit{for all } j.$$

Let b be a fixed real number, $b \ge n \log n$. Then complex numbers z_j^*, $j = 1, \ldots, n$, satisfying

$$\|p(x) - p_n \prod_{j=1}^{n}(x - z_j^*)\| \leq 2^{-b}\|p(x)\| \tag{2.1.3}$$

can be computed by using $O((n\log^2 n)(\log^2 n + \log b))$ *ops performed with the precision of* $O(b)$ *bits or by using* $O((n\log^2 n)(\log^2 n + \log b)\mu(b))$ *bit-operations for* $\mu(b)$ *denoting the bit-operation cost of performing a single op with the b-bit precision; by* (1.1.4) *we have* $\mu(b) = O((b\log b)\log\log b)$.

By a theorem from Schönhage (1982b, Section 19), the approximate factorization (2.1.3) defines approximations z_j^* to the roots z_j of $p(x)$ satisfying

$$|z_j^* - z_j| < 2^{2-b/n}, \qquad j = 1, \dots, n. \tag{2.1.4}$$

Corollary 2.1.2. *Under the assumptions of Theorem* 2.1.1*, its cost bounds apply to the task of computing approximations* z_j^* *to all the roots* z_j *of a polynomial* $p(x)$*, where the approximation errors are bounded according to* (2.1.4).

Bound (2.1.4) covers the worst case polynomials $p(x)$ whose roots may be ill-conditioned, that is, form clusters. The recovery of well-conditioned (isolated) roots of $p(x)$ from factorization (2.1.3) has an approximation error of the order of 2^{-b}.

With no preliminary knowledge about how well (or poorly) the roots of a given polynomial $p(x)$ are isolated from each other, one may apply the algorithm supporting Theorem 2.1.1 and Corollary 2.1.2 and obtain the isolation information by examining the discs $D(z_j^*, 2^{2-b/n})$. To refine the bounds of (2.1.4), one may apply, for instance, the root radii algorithms in Schönhage (1982b) and/or (modifications of) the Weierstrass method (cf. Bini and Pan, to appear or Pan, 2002).

The estimates of Theorem 2.1.1 and Corollary 2.1.2 are nearly optimal. Indeed, even the approximation of a single root of a polynomial $p(x)$ requires at least $(n+1)/2$ arithmetic operations. This follows because the approximation involves $n+1$ coefficients of $p(x)$, whereas each arithmetic operation has two operands and, therefore, may involve at most two parameters. To approximate the n roots, we need at least n arithmetic operations because the algorithm must output n values that are generally distinct. Therefore, the arithmetic cost bound of Theorem 2.1.1 is optimal up to polylogarithmic factors in n. So, also, is the Boolean cost bound, due to Fact 1.1 in Pan (1996). We also recall the lower bound $\Omega(\log b)$ in Renegar (1987), on the number of arithmetic operations required for the approximation of even a single root of $p(x)$ within 2^{-b} under the normalization assumption that $|z_i| \leq 1$ for all i.

Remark 2.1.3. We deduce our bit-operation (Boolean) cost bounds simply by combining the ops and precision bounds and the estimate (1.1.4) on the bit-operation (Boolean) cost of performing an op with the b-bit precision. (To apply the latter estimate, known for the bit-operation cost of an arithmetic operation with integers performed modulo 2^b+1, truncate real and complex operands to b bits and then scale them.) This approach can be immediately extended to yield bit-operation (Boolean) cost estimates based on the other known upper bounds on $\mu(b)$ (cf. Bini and Pan, to appear). It seems that a small further decrease (by the factor of $O(\log n)$) of our bit-operation cost estimates is possible if one applies the refined integer arithmetic based on the binary segmentation techniques (cf. Schönhage, 1982a,b; Bini and Pan, 1994, Section 3.3; Kirrinnis, 1998).

2.2. SOME BASIC DEFINITIONS AND RESULTS

To reach the (nearly optimal) estimates of Theorem 2.1.1, one must balance the degrees of the two output factors in each step of the recursive splitting. If, on the contrary, each splitting produces a linear factor, then $n-1$ splittings and at least the order of n^2 arithmetic operations are necessary.

Generally, it can be very hard to ensure balanced splitting, however. For example, for a polynomial $p(x)=\prod_{i=1}^{k}(x-2^{-i^3}-5/7)G(x)$, where $G(x)$ is a polynomial of degree $n-k=n^{1/3}$, one must separate from each other some roots of $p(x)$ lying in the same disc of radius $1/2^{cn^3}$, for a fixed positive c. (By following Pan (1996), we say that $p(x)$ has a *massive cluster* of roots in such cases.) Then, to yield the balanced splitting, one must perform computations with a precision of the order of n^4 bits, even if we are only required to approximate the roots of $p(x)$ within the error tolerance 2^{-10n}. Such a high precision of computing would not allow us to reach the Boolean complexity bounds of Theorem 2.1.1.

We salvage the optimality (up to polylog factors) only because we do not compute balanced splitting in this case. Indeed, the same point $z=5/7$ approximates (within 2^{-10n}) all but $n-l=O(n^{1/3})$ roots of $p(x)$, and it remains to approximate the remaining $n-l=O(n^{1/3})$ roots of $p(x)$ by working with a polynomial of a degree $O(n^{1/3})$, obtained as the quotient of numerical division of $p(x)$ by $(x-5/7)^l$.

Generalizing the latter recipe, we detect massive clusters and approximate their roots without computing balanced splitting of a given polynomial. Formally, we introduce the concepts of (a,ψ)-*splitting annuli* (basic for balanced splittings) and (a,B,ψ)-*splitting discs* (each covering a massive cluster of the roots to be approximated by a single point, without computing a balanced splitting).

DEFINITION 2.2.1. A disc $D(X,\rho)=\{,|-|\leq\rho\}$ is called an (a,B,ψ)-*splitting disc* for a polynomial $p(x)$ if it is both ψ-isolated and contains more than $(3a-2)n$ roots of $p(x)$ and if ρ satisfies the relations

$$\rho\leq 2^{-B}. \tag{2.2.1}$$

An annulus $A(X,\rho_-,\rho_+)=\{x,\ \rho_-\leq|x-X|\leq\rho_+\}$ is called an (a,ψ)-*splitting annulus* for $p(x)$ if it is free of the roots of $p(x)$ and if its internal disc $D(X,\rho)$ contains exactly k roots of $p(x)$ (counted with their multiplicities) where $\rho_+\geq\psi\rho_-$ and

$$(1-a)n/2\leq k\leq(1+a)n/2 \tag{2.2.2}$$

(cf. (2.1.1)). In the latter case we also call the disc $D(X,\rho_-)$ an (a,ψ)-*splitting disc* for the polynomial $p(x)$. A disc containing exactly k roots of $p(x)$ for k satisfying bounds (2.2.2) is called a-*balanced*.

DEFINITION 2.2.2. The jth root radius for $p(x)$ is the distance r_{n+1-j} from the origin to the jth closest root of $p(x)$. (We have $r_{n+1-j}=|z_j|$, $j=1,\ldots,n$, if the roots z_j of $p(x)$ are enumerated so that $|z_1|\geq|z_2|\geq\cdots\geq|z_n|$.) We write $r_0=\infty$, $r_{n+1}=0$, and $r_j(X)$ for the jth root radius of the polynomial $q(x)=p(x+X)$, obtained from $p(x)$ when the origin is shifted into a complex point X.

We use the following auxiliary result.

Proposition 2.2.3. (Schönhage, 1982b) (Cf. also Pan, 2000) $O(n \log^2 n)$ *ops performed with $O(n)$-bit precision are sufficient to approximate within the relative error bound c/n^d (for any fixed pair of $c > 0$ and $d \geq 0$) all root radii r_j of a polynomial $p(x)$, $j = 1, \dots, n$, as well as all root radii $r_j(X)$ of $q(x) = p(x + X)$ for $j = 1, \dots, n$ and any fixed complex X.*

Remark 2.2.4. Proposition 2.2.3 enables us to extend Theorems 1.1.2 and 1.1.3 to the case where the annulus $A(0, \psi, 1/\psi)$ is replaced by any other splitting annulus with a relative width ψ satisfying (1.1.7). The extension is obtained via the respective linear transformations of the variable x (cf. Section 1.6).

2.3. basic splitting annuli or large root clusters

Let us fix a positive a in (2.1.2) and write

$$\psi = 1 + c/n, \tag{2.3.1}$$

for a fixed positive constant c (to be estimated later on). For a large class of polynomials $p(x)$, their (a, ψ)-splitting discs (and consequently, their balanced splitting) can be computed immediately by means of the approximation of root radii. Indeed, apply Proposition 2.2.3 to compute an (a, ψ)-splitting disc for $p(x)$. Write

$$g(a) = \lfloor (1 - a)n/2 \rfloor, \qquad h(a) = g(a) + \lfloor an \rfloor, \tag{2.3.2}$$

so $0 \leq (1 + a)n/2 - h(a) < 2$,

$$g(a) \geq \lfloor n/12 \rfloor, \qquad h(a) \geq \lfloor n/12 \rfloor + \lfloor 5n/6 \rfloor \qquad \text{for } a \geq 5/6,$$

and let r_i^+ and r_i^- denote the computed upper and lower estimates for $r_i = |z_i|$, $i = 1, \dots, n$. We require that

$$r_i^+/r_i^- \leq \psi^* = 1 + (c/n)^2, \qquad i = 1, \dots, n, \tag{2.3.3}$$

for the same fixed positive c and observe that the discs $D(0, r_i^+)$ are (r_{i-1}^-/r_i^+)-isolated for all i. If

$$r_{i-1}^-/r_i^+ \geq \psi, \tag{2.3.4}$$

for at least one choice of an integer i satisfying

$$g(a) \leq n + 1 - i < h(a), \tag{2.3.5}$$

then the disc $D(0, r_i^+)$ is both a-balanced, due to (2.3.2), and ψ-isolated, due to (2.3.4).

This approach yields the desired (a, ψ)-splitting discs for a very large class of the input polynomials $p(x)$, that is, for those for which bound (2.3.4) holds for some integer i satisfying (2.3.5). To yield *Universal Root-finders* for all input polynomials $p(x)$, it remains to treat the opposite case where bound (2.3.4) holds for no i of (2.3.5). In this case, we still make some progress based on Proposition 2.2.3. Indeed, observe that at least $h(a) - g(a) + 1 = \lfloor an \rfloor + 1$ roots of $p(x)$ lie in the closed annulus

$$A = \{x : r_{n+1-g(a)}^- \leq |x| \leq r_{n+1-h(a)}^+\} \tag{2.3.6}$$

and recall that we have the bounds $r_{i-1}^- < \psi r_i^+$ for all i of (2.3.5). Together with (2.3.3), these bounds imply that the relative width of the annulus A satisfies the bound

$$r_{n+1-h(a)}^+/r_{n+1-g(a)}^- \leq (\psi\psi^*)^{h(a)-g(a)+1}. \tag{2.3.7}$$

Now we apply Proposition 2.2.3 twice, for the origin shifted into the points $2r^-_{n+1-h(a)}$ and $2r^-_{n+1-h(a)}\sqrt{-1}$. Then we either compute a desired (a,ψ)-splitting disc for $p(x)$ or arrive at two additional narrow annuli of radii at most $3r^-_{n+1-h(a)}$, each having a relative width of at most $(\psi\psi^*)^{h(a)-g(a)+1}$ and each containing at least na roots of $p(x)$. Our current goal is the determination of an (a,ψ)-splitting disc for $p(x)$, so it is sufficient to examine the latter case, where each of the three narrow annuli contains more than na roots of $p(x)$.

We have the following simple result, which we only need for $h = 3$.

Proposition 2.3.1. (Neff and Reif, 1994) *Let $S_1, S_2, \ldots, S_h$ denote h finite sets. Let U denote their union and I their intersection. Then*

$$|I| \geq \sum_{i=1}^{h} |S_i| - (h-1)|U|,$$

where $|S|$ denotes the cardinality of a set S.

Due to Proposition 2.3.1, the intersection of the three narrow annuli contains more than $(3a-2)n \geq n/2$ roots of $p(x)$. Since the annuli are narrow, we cover their intersection by a sufficiently small *covering disc*, $D = D(Y, r)$. We ensure that $r < 0.1r^-_{n+1-h(a)}$ (say), by choosing the constant c in (2.3.1) and (2.3.3) sufficiently small.

We could have decreased ψ to $1 + c/n^d$ for a small positive c and $d > 1$ and consequently decreased the radius of the disc to $O(r^-_{n+1-h(a)}/n^{d-1})$, but then an extension of Theorem 2.1.1 would be required to avoid a dramatic growth of the computational cost estimates in the subsequent construction (see Remark 2.6.2). Thus we stay with ψ of (2.3.1) but shift the origin into the center Y of the disc D and apply the same construction again. Furthermore, we repeat this process recursively until we obtain either a desired (a,ψ)-splitting disc for $p(x)$ or a (small) covering disc that contains more than $(3a-2)n$ roots of $p(x)$ and has a radius r bounded from above by $r^-_{n+1-h(a)}(0)/n^d$ for a fixed positive d. This radius may be small enough to enable the computation of an (a, B, ψ)-splitting disc. We always check if this is the case for each computed radius r (see Remark 2.3.5), but generally we cannot count on such a rapid success. Hereafter, we refer to this recursive computation as a RRRP.

Proposition 2.3.2. *Write $X_0 = 0, r_0 = r^+_{n+1-h(a)}$ and let $D(X_i, r_i)$ denote the output covering disc of the ith recursive step of the RRRP. Then we have $5r_i \leq r_{i-1}$ and $|X_i| \geq r_0/2$ for $i = 1, 2, \ldots$ provided that the constant c in* (2.3.1) *and* (2.3.3) *has been chosen small enough.*

Proof. Let w_{i-1} denote the width of the narrow annulus A_i centered in X_i and computed at the $(i-1)$st recursive step. By the construction of this annulus, we have

$$w_{i-1} \leq ((\psi\psi^*)^{h(a)-g(a)+1} - 1)r_{i-1}$$

where $h(a) - g(a) = \lfloor an \rfloor$ (cf. (2.3.2)). Clearly, $(\psi\psi^*)^{an} \to 1$ as $c \to 0$ for c in (2.3.1) and (2.3.3). Therefore, $w_{i-1}/r_{i-1} \to 0$ as $c \to 0$, that is, we may assume that $w_{i-1} \leq \nu^2 r_{i-1}$ for any fixed positive ν. It is easy to verify that the intersection of the annulus A_i with the two other annuli computed at the same recursive step of the RRRP must have diameter at most $\mu\sqrt{w_{i-1}r_{i-1}}$ for some fixed constant μ. Therefore, the radius r_i of the output disc (covering this intersection) is less than $\mu\nu r_{i-1}$. It remains to choose $\nu < 0.2/\mu$ to obtain that $5r_i \le r_{i-1}$. On the other hand, we have $|X_i - X_{i-1}| \to r_{i-1}$ as $c \to 0$, for $i = 1, 2, \ldots$. Together with the bound $5r_i \le r_{i-1}$ and equation $X_0 = 0$, this implies that $|X_i| \ge r_0/2$ for $i = 1, 2, \ldots$. □

COROLLARY 2.3.3. *Under the assumptions of Proposition* 2.3.2, *we have* $2|X_i| \ge 5^i r_i$, *for* $i = 1, 2, \ldots$.

For a large class of input polynomials $p(x)$, the RRRP outputs (a, ψ)-splitting discs, thus completing our task. In the remaining case, a covering disc D of a smaller size is output. We may use the center of the disc D as a generally crude approximation to more than $n/2$ roots of $p(x)$. The same algorithm can be extended to improve the latter approximations, decreasing the approximation error with the linear rate. This is too slow for us, however. We follow a distinct strategy: we specify and satisfy a condition under which the RRRP never outputs a disc that covers the intersection of the three narrow annuli, so a desired (a, ψ)-splitting disc must be output.

REMARK 2.3.4. Application of the root radii algorithm enables us to compute a desired (a, B, ψ)-splitting disc for $p(x)$ (see Definition 2.2.1) as soon as we detect that the value $B_k = 2^{-B}/(\psi\psi^*)^{n-k+1}$ exceeds the radius r of some computed disc $D(X, r)$ containing k roots of $p(x)$ where $k > (3a-2)n$. Indeed, in this case, we shift the origin into the point X, compute the values r_i^- and r_i^+ for $i = 1, 2, \ldots, n-k+1$ and write $r_0^- = \infty$. Then we choose the maximal i such that $i \le n-k+1$ and $r_{i-1}^-/r_i^+ \ge \psi$ and observe that $r_i^+ \le 2^{-B}$ and that the disc $D(X, r_i^+)$ for such i is ψ-isolated and, therefore, is a desired (a, B, ψ)-splitting disc for $p(x)$. The comparison of the above values B_k with the radii of all computed discs containing more than $k \ge (3a-2)n$ roots of $p(x)$ is assumed by default to be a part of all our algorithms (to simplify their description, we do not cite this comparison explicitly). Without making these comparisons, we would have lost our control over the precision and the Boolean cost of computing and would have allowed them to blow-up.

2.4. A (t, s)-CENTER OF A POLYNOMIAL AS A ROOT OF ITS HIGHER ORDER DERIVATIVE

We recall the following result from Coppersmith and Neff (1994).

THEOREM 2.4.1. *Let an integer* l *satisfy* $0 < l < n-1$, *let a disc* $D(X, r)$ *contain at least* $l+1$ *roots of a polynomial* $p(x)$ *of degree* n, *and let* $s \ge 2 + 1/\sin(\pi/(n-l))$. *Then the disc* $D(X, (s-2)r)$ *contains a root of* $p^{(l)}(x)$, *the* l*th order derivative of* $p(x)$. *Furthermore, the same property holds for* $l = n-1$ *and any* $s \ge 3$.

REMARK 2.4.2. Theorem 2.4.1 extends to the complex polynomial Rolle's classical theorem about a root (zero) of the derivative of a real function. A distinct and much earlier extension of this theorem to the complex case, due to Gel'fond (1958), supports our nearly optimal asymptotic complexity estimates of Theorem 2.1.1 as well, although with slightly larger overhead constants hidden in the O notation of these estimates. On the other hand, application of more advanced techniques in Coppersmith and Neff (1994) enables a further decrease of the parameter s in Theorem 2.4.1 and, consequently, a further decrease of the latter constants. Namely, by using non-trival techniques based on properties of symmetric polynomials, the result of Theorem 2.4.1 was extended in Coppersmith and Neff (1994) to any s exceeding $2 + c\max\{(n-l)^{1/2}(l+1)^{-1/4}, (n-l)(l+1)^{-2/3}\}$, for $l = 2, 3, \ldots, n-1$ and for some constant c. This extension allows one to replace an s of the order of n in Theorem 2.4.1, by an s of the order of $n^{1/3}$.

Hereafter, we assume that

$$l = \lfloor (3a-2)n \rfloor, \qquad n - l = \lceil (3-3a)n \rceil, \tag{2.4.1}$$

and s satisfies the assumption of Theorem 2.4.1. By combining (2.1.2) and (2.4.1), we obtain that $l \geq \lfloor n/2 \rfloor$, $l + 1 > n/2$. In particular, one may choose

$$a = 5/6, \qquad l = \lfloor n/2 \rfloor, \qquad n - l = \lceil n/2 \rceil. \tag{2.4.2}$$

Definition 2.4.3. (Neff and Reif, 1994) A disc $D(X, r)$ is called *t-full* if it contains more than t roots of $p(x)$. A point Z is called a *(t, s)-center for $p(x)$* if it lies in the dilation $D(X, sr)$ of any t-full disc $D(X, r)$.

Proposition 2.4.4. (Neff and Reif, 1994) *Let $t \geq n/2$ and let $s > 2$. If a complex set S has a non-empty intersection with the dilation $D(X, (s-2)r)$ of any t-full disc $D(X, r)$, then this set S contains a (t, s)-center for $p(x)$.*

Proof. Let $D(X, r)$ be a t-full disc for $p(x)$ of the minimum radius and let Z be a point of the set S lying in the disc $D = D(X, (s-2)r)$. Let $D(Y, R)$ be another t-full disc for $p(x)$. Then $R \geq r$, and since $t \geq n/2$, this disc intersects $D(X, r)$. Therefore, the disc $D(Y, sR)$ covers the disc D and, consequently, the point Z, which is, therefore, a (t, s)-center for $p(x)$. □

Proposition 2.4.4 and Theorem 2.4.1 together imply the next result.

Corollary 2.4.5. *If s satisfies the assumptions of Theorem* 2.4.1 *for $n+1 > l+1 > n/2$, then at least one of the $n - l$ roots of the lth order derivative of $p(x)$ is an (l, s)-center for $p(x)$.*

2.5. (t, s)-centers and splitting a polynomial

Now suppose that we apply the RRRP from Section 2.3 in the case where the origin is initially shifted into a (t, s)-center Z for $p(x)$ and where $t/n = 3a - 2 \geq 1/2$. Then after sufficiently many recursive steps, an (a, ψ)-splitting disc must be output. Indeed, otherwise, according to Corollary 2.3.3, for every positive i the ith recursive step must output a covering disc $D(X_i, r_i)$ containing more than $(3a-2)n$ roots of $p(x)$ where $5^i r_i \leq 2|X_i|$. Then it follows that

$$sr_i < |X_i|, \tag{2.5.1}$$

already for some $i = O(\log s)$. The latter inequality implies that the origin cannot lie in the disc $D(X_i, sr_i)$, in contradiction to our assumption that the origin is (or has been shifted into) a (t, s)-center for $p(x)$.

This gives us an algorithm (hereafter referred to as THE DISC/CENTER ALGORITHM) that computes an (a, ψ)- or an (a, B, ψ)-splitting disc for $p(x)$ as soon as we have a (t, s)-center for $p(x)$ where $t \geq n/2$.

It is easy to extend this algorithm to the case where an approximation to a (t, s)-center for $p(x)$ is available within a small absolute error, say, being less than

$$\rho^* = 2^{-2B}/s. \tag{2.5.2}$$

The extension relies on the following result.

PROPOSITION 2.5.1. *Suppose that an unknown $((3a-2)n, s)$-center for $p(x)$ lies in a disc $D(0, \rho^*)$. Suppose that the Disc/Center Algorithm applied at the origin (rather than at such a center) does not output an (a, ψ)-splitting disc for $p(x)$ but yields a covering disc $D = D(X, r)$, which is $((3a-2)n)$-full for $p(x)$. Then*

$$|X| \leq sr + \rho^*. \tag{2.5.3}$$

PROOF. A $((3a-2)n, s)$-center for $p(x)$ lies in both discs $D(X, sr)$ and $D(0, \rho^*)$. These two discs have a non-empty intersection because $3a-2 \geq 1/2$, and hence $|X| \leq sr + \rho^*$. □

By Proposition 2.5.1, application of the Disc/Center Algorithm should output a desired (a, ψ)- or (a, B, ψ)-splitting disc for $p(x)$ as soon as we have $sr_i < |X_i| - \rho^*$, which for a small ρ^* in (2.5.2) is almost as mild a bound as (2.5.1).

Due to Corollary 2.4.5, an (l, s)-center for $p(x)$ can be found among the $n-l$ roots of the lth order derivative $p^{(l)}(x)$ for l of (2.4.1). Suppose that the set, Z_l^*, of sufficiently close approximations to these roots within ρ^* in (2.5.2) is available, but we do not know which of them is a (t, s)-center for $p(x)$, for $t \geq n/2$. Then, clearly, we still may compute a desired splitting disc by applying the Disc/Center Algorithm with the origin shifted into each of the $n-l$ approximations to the $n-l$ roots of $p^{(l)}(x)$. Alternatively, we may apply an implicit binary search (Neff and Reif, 1994), which enables us to shift the origin into at most $\lceil \log(n-l) \rceil$ candidate approximation points Y_i. We call the latter algorithm (using implicit binary search) the IBS ALGORITHM.

In spite of the acceleration by roughly the factor of $(n-l)/\log(n-l)$ vs. application of the Disc/Center Algorithm at every point of S_0, the latter algorithm still reduces the approximation of the roots of $p(x)$ to the approximation of the roots of a higher order derivative $p^{(l)}(x)$ (at first) and then of two factors of $p(x)$, denoted $F(x)$ and $G(x)$. Due to the extra stage of the approximation of the roots of $p^{(l)}(x)$, which precedes the computation of a splitting disc for $p(x)$, the overall upper bounds on both sequential and parallel time of polynomial root-finding increase by the factor of n^δ for some positive δ. In the next section, we show how to avoid this costly stage.

2.6. THE ROOTS OF HIGHER ORDER DERIVATIVES ARE NOT REQUIRED

Suppose that we have an (a, ψ)-splitting disc, $D(Y, R)$, for the lth order derivative $p^{(l)}(x)$ for l of (2.4.1). Then we may shift the origin into Y and apply the RRRP Algorithm in Section 2.3, repeating the recursive process until either a desired (a, ψ)-splitting disc for $p(x)$ is computed or the dilation $D(X_i, sr_i)$ of a covering disc $D(X_i, r_i)$ lies either entirely in the disc $D(Y, \psi R)$ or entirely in the exterior of the disc $D(Y, R)$. The latter property of the dilation of the disc $D(X_i, r_i)$ is clearly ensured if the width $(\psi - 1)R$ of the computed annulus $\{x : R \le |x| \le \psi R\}$ (which is free of the roots of $p^{(l)}(x)$) exceeds the diameter $2sr_i$ of the disc $D(X_i, sr_i)$.

Let z denote a (t,s)-center for $p(x)$ such that $p^{(l)}(z) = 0$, $t \ge n/2$. Let $p^{(l)}(x) = f_l(x)g_l(x)$, where $f_l(x)$ and $g_l(x)$ are two polynomials, $f_l(x)$ has all its roots in the disc $D(Y,R)$, and $g_l(x)$ has no roots in the disc $D(Y,\psi R)$. Then we have $f_l(z) \ne 0 = g_l(z)$ if the dilation $D(X_i, sr_i)$ of the covering disc $D(X_i, r_i)$ has an empty intersection with the disc $D(Y,R)$, and we have $f_l(z) = 0 \ne g_l(z)$ if $D(X_i, sr_i) \subseteq D(Y,\psi R)$. Therefore, the considered application of the RRRP Algorithm enables us to discard one of the two factors, $f_l(x)$ or $g_l(x)$, and to narrow the search of a (t,s)-center z for $p(x)$ to the set of the roots of the remaining factor of $p^{(l)}(x)$. By continuing recursively, we compute either an (a,ψ)- or an (a,B,ψ)-splitting disc for $p(x)$. Indeed our search for a (t,s)-center for $p(x)$ where $t \ge n/2$ ends with outputting an (a,ψ)-splitting disc for $p(x)$ in $O(\log(n-l))$ recursive steps. To the advantage of this approach, instead of all roots of $p^{(l)}(x)$ it requires approximation of only one factor of $p^{(l)}(x)$ and the root radii of $p^{(l)}(x+X)$ for some shift value X. Arriving at a (t,s)-center enables a low cost reduction of the original problem of the complete factorization of a polynomial $p(x)$ to two similar problems for its two factors, $F(x)$ and $G(x)$ satisfying the balancing assumption (2.1.1). By combining this approach with the recursive splitting algorithms of Part I, we arrive at Theorem 2.1.1.

We next specify the resulting factorization algorithm first describing two subroutines for splitting based on Theorems 1.1.1 and 1.1.2.

Subroutine Split$(v(x), B_v, A)$.

Input: A polynomial $v(x)$ of degree n_v, real B_v, and an annulus $A = A(X, r_-, r_+) = \{x : r_- \le \|x - X\| \le r_+\}$ on the complex plane, for positive r_- and r_+ and a complex X.

Output: Two polynomials, $f^*(x)$ (monic, with all its roots lying in the disc $D(X, r_-^*)$) and $g^*(x)$ (with all its roots lying outside the disc $D(X, r_+^*)$), for $r_-^* = qr_-$, $r_+^* = r_+/q$, $r_+/r_- = q^4$, satisfying

$$\|f^*(x)g^*(x) - v(x)\| \le 2^{-B_v}\|v(x)\|. \tag{2.6.1}$$

Subroutine Refine $(v(x), B_v, A, \epsilon)$.

Input: A positive ϵ and the output of Subroutine Split $(v(x)), B_v, A)$ where at least one of the factors f^* or g^* is linear.

Output: An approximation within the error bound ϵ to the single root of $v(x)$ corresponding to linear factor f^* or g^*.

Algorithm 2.6.1. *DISC*$(p(x), a, B, 2^N)$.

Input: *Polynomial* $p(x) = \sum_{i=0}^{n} p_i x^i$ *of* (1.1.1), $p_n \ne 0$, *real* a, B, c, ψ, ψ^*, *and* s (*provided that* a *satisfies* (2.1.2), c, ψ, *and* ψ^* *satisfy* (2.3.1) *and* (2.3.3),

$$B > Cn \log n \tag{2.6.2}$$

for a sufficiently large constant C, *and* s *satisfies the assumption of Theorem* 2.4.1), *and subroutines Split*$(v(x), B_v, A)$ *and Refine*$(v(x), B_v, A, \epsilon)$ *specified above.*

Output:

(*a*) *Either an* (a,ψ)*-splitting disc for* $p(x)$ *or*
(*b*) *an* (a, B, ψ)*-splitting disc for* $p(x)$.

INITIALIZATION. *Write* $v(x) = p^{(l-1)}(x)$ *for* $l = \lfloor (3a-2)n \rfloor$ *of* (2.4.1).

COMPUTATIONS:

Stage 1. *Substitute* $n_v = \deg v(x)$ *for* n, ψ_v *for* ψ, *and* ψ_v^* *for* ψ^* *in* (2.3.1) *and* (2.3.3), *to define* ψ_v *and* ψ_v^*. *Then apply Algorithm* 2.6.1 *for* $p(x) = v(x)$ *and* B *replaced by* $2B \log s$, *that is, apply the algorithm* $DISC(v(x), a, 2B \log s, c)$ *for* c *in* (2.3.1) *and* (2.3.3), *which outputs an* (a, ψ_v)*- or an* $(a, 2B \log s, \psi_v)$*-splitting disc for* $v(x)$*; denote this disc* $D = D(C_v, R_v)$. *Shift the origin into its center* C_v *and go to Stage* 2.

Stage 2. *Apply the RRRP Algorithm in Section* 2.3 *to the input polynomial* $p(x)$. *Stop if an* (a, B, ψ)*- or an* (a, ψ)*-splitting disc for* $p(x)$ *is output. Otherwise stop in* i *recursive steps for the minimal* i *such that the RRRP produces a covering disc* $D(X_i, r_i)$ *with radius* r_i *less than* $(\psi_v - 1)R_v/s$, *where* $(\psi_v - 1)R_v$ *is the width of the annulus produced at Stage* 1*; in this case invoke the Subroutine Split*$(v(x), B_v, A(C_v, R_v, \psi R_v))$ *with*

$$B_v = C^* B \log s, \tag{2.6.3}$$

for a sufficiently large constant C^*, *and go to Stage* 3.

Stage 3. *Suppose that at Stage* 2 *the RRRP outputs a covering disc* $D(X_i, r_i)$. *Write either* $v(x) = f^*(x)$, *if the dilation* $D(X_i, sr_i)$ *intersects the disc* D, *or* $v(x) = g^*(x)$, *otherwise. If* $\deg f^* = 1$ *or* $\deg g^* = 1$, *invoke the Subroutine Refine*$(v(x), B_v, A, sr_i/2)$. *Then go to Stage* 1.

To see the *correctness of Algorithm* 2.6.1, observe that according to our policy, at Stage 3, we discard the "wrong" factor of $v(x)$ and stay with the "right" one keeping a (t, s)-center for $p(x)$ among its roots. (If we compute an $(a, 2B \log s, \psi_\nu)$-splitting disc for $v(x)$, then a (t, s)-center must lie in this disc, and we compute the desired splitting disc for $p(x)$ based on Proposition 2.5.1.) By Theorems 1.1.1 and 1.1.2, which support the Subroutine Split$(v(x), B_v, A)$, a relatively wide subannulus of the basic annulus isolates from each other the two sets of the roots of the two approximate factors computed in every splitting of $v(x)$ (or $p(x)$). The degree of each computed factor of $v(x)$ is bounded from above by a fixed fraction of $\deg v(x)$. Therefore, Algorithm 2.6.1 must terminate in $O(\log(n - l))$ passes through Stage 3. At termination, it must output an (a, B, ψ)- or an (a, ψ)-splitting disc for $p(x)$. By estimate (19.3) in Schönhage (1982b), our bounds (2.6.1)–(2.6.3) ensure the relative bounds of the order of $B \log s$ on the error norms of the computed approximations $f^*(x)$ and $g^*(x)$ to the factors of $v(x)$. Together with the known perturbation Theorem 1.3.1, this implies that a (t, s)-center for $p(x)$ is closely approximated by a root of the selected factor. Furthermore, by Corollary 2.4.5 and Proposition 2.5.1, the center C of an $(a, 2B \log s, \psi)$-splitting disc for $v(x)$ computed by Algorithm 2.6.1 closely approximates a (t, s)-center for $p(x)$, so C itself is a (t, s^*)-center for $p(x)$ where $s^* = s + 1$, say.

To estimate the *cost of the computation* by the algorithm, observe that the entire computation is reduced to the application of the Subroutine Split to the auxiliary polynomials $v(x)$ of rapidly decreasing degree, followed by the single invocation of the Subroutine Refine, the shifts of the variable x, and the approximation within relative error bound $O(1/n)$ of all root radii of the polynomials $p(x)$ and $v(x)$. The shifts and the root radii approximation require only $O(n \log^2 n)$ ops performed with $O(n \log n)$-bit precision per recursive step, that is, $O(n \log^3 n)$ ops with $O(n \log n)$-bit precision at all $O(\log n_v)$ recursive steps. This is dominated by the computational cost estimate for all applications of the Subroutine Split; the latter estimate also dominates the computational cost of a single application of the Subroutine Refine (due to Theorems 1.1.2 and 1.1.3). By Theorem 1.1.3 and Remark 2.2.4, each application involves $O((n_\nu \log n_\nu)(\log^2 n_\nu + \log B_\nu))$ ops performed with $O(B_\nu)$-bit precision.

Now we are ready to prove Theorem 2.1.1.

Proof of Theorem 2.1.1. By the above argument, the cost of the computation of an (a,ψ)- or (a,B,ψ)-splitting disc for the polynomial $p(x)$ is dominated by the cost of the subsequent balanced splitting of this polynomial, that is, $O((n \log n)(\log^2 n + \log b))$ ops performed with $O(b)$-bit precision where we choose $b = B \geq n \log n$. Recursive extension of the balanced splitting has depth $O(\log n)$ due to its balancing. The above cost bound applies at every level of the recursive process, and we arrive at Theorem 2.1.1. □

Remark 2.6.2. As we mentioned earlier, application of the RRRP as a block of Algorithm 2.6.1 can be replaced by a single step of the root radii approximation algorithm but with $\psi = 1 + c/n^d$ and $\psi^* = 1 + c/n^{d+1}$ for $p(x)$ and with $\psi_v = 1 + c/n_v^{d_v}$ and $\psi_v^* = 1 + c/n_v^{d_v+1}$ for $v(x)$, for a positive c and larger $d > 1$ and $d_v > 1$. In this case, however, we must have $n^d \geq s n_v^{d_v}$ to ensure the bounds of (2.5.1)–(2.5.3). For $n_v = l$ of (2.4.1), this means $n^d \geq \lceil (3a-2)n \rceil^{d_v} s > \Theta n^{d_v + 1/3}$ for a fixed positive Θ and for $s = O(n^{1/3})$ (cf. Remark 2.4.2). To compute a splitting disc for the polynomial $p^{(l)}(x)$, we have to apply Algorithm 2.6.1 to the polynomials $p^{(l_i)}(x)$ where $l_0 = l$, $l_{i+1} = \lceil (3a-2)l_i \rceil$, $i = 0, 1, \ldots$ (cf. (2.4.1)). This would involve ψ_i-isolated discs for $p^{(l_i)}(x)$ with $\psi_i - 1$ of the order of $1/n^{d-i/3}$. We must generally deal with the number of recursive steps i of the order of $\log n$, which means that the exponent d must also be of this order. That is, the considered modification of the RRRP and Algorithm 2.6.1 would require splitting $p(x)$ over an annulus with a relative width of the order of $1/n^{\log n}$. To yield this splitting, we must extend Proposition 2.2.3 and the lifting/descending construction. Then the resulting estimated arithmetic time-cost of factorization and root-finding would have increased a little (at least by the factor of $\log n$), but what is much worse, the computation of the required splitting would have involved unreasonably high bit-precision, of the order of $n^{\log n}$, and, therefore, would have dramatically blown-up the Boolean (bit-operation) cost.

Acknowledgements

I thank the volume editor for inviting my submission and the referee for helpful comments. This paper is supported by NSF grant CCR 9732206 and PSC-CUNY awards 669363, 61393-0030, and 62435-0031. I thank the ACM for permission to publish the paper in a journal.

References

Ahlfors, L. (1979). *Complex Analysis.* New York, McGraw-Hill.

Barnett, S. (1983). *Polynomial and Linear Control Systems.* New York, Marcel Dekker.

Bell, E. T. (1940). *The Development of Mathematics.* New York, McGraw-Hill.

Bini, D. A., Fiorentino, G. (2000). Design, analysis, and implementation of a multiprecision polynomial rootfinder. *Numer. Algorithms*, **23**, 127–173.

Bini, D., Gemignany, L., Meini, B. Computations with infinite Toeplitz matrices and polynomials, to appear.

Bini, D., Pan, V. Y. (1994). *Polynomial and Matrix Computations, v. 1: Fundamental Algorithms.* Boston, Birkhäuser.

Bini, D., Pan, V. Y. *Polynomial and Matrix Computations, v. 2: Fundamental and Practical Algorithms.* Boston, Birkhäuser, to appear.

Blum, L., Cucker, F., Shub, M., Smale, S. (1997). *Complexity and Real Computations*. New York, Springer.
Box, G. E. P., Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control*. San Francisco, CA, Holden-Day.
Boyer, C. A. (1968). *A History of Mathematics*. New York, Wiley.
Brent, R. P., Gustavson, F. G., Yun, D. Y. Y. (1980). Fast solution of Toeplitz systems of equations and computation of Padé approximations. *J. Algorithms*, **1**, 259–295.
Bunch, J. R. (1985). Stability of methods for solving Toeplitz systems of equations. *SIAM J. Sci. Stat. Comput.*, **6**, 349–364.
Coppersmith, D., Neff, C. A. (1994). Roots of a polynomial and its derivatives. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 271–279. New York, ACM Press and Philadelphia, SIAM Publications.
Delves, L. M., Lyness, J. N. (1967). A numerical method for locating zeros of an analytic function. *Math. Comput.*, **21**, 543–560.
Demeure, C. J., Mullis, C. T. (1989). The Euclid algorithm and the fast computation of cross-covariance and autocovariance sequences. *IEEE Trans. Acoust. Speech Signal Process.*, **37**, 545–552.
Demeure, C. J., Mullis, C. T. (1990). A Newton–Raphson method for moving-average spectral factorization using the Euclid algorithm. *IEEE Trans. Acoust. Speech Signal Process.*, **38**, 1697–1709.
Emiris, I. Z., Galligo, A., Lombardi, H. (1996). Numerical univariate polynomial GCD. In *Proceedings of AMS-SIAM Summer Seminar: Mathematics of Numerical Analysis: Real Number Algorithms (Park City, Utah, 1995), Lectures in Applied Mathematics*, **32**, 323–343. Providence, RI, American Mathematical Soceity.
Emiris, I. Z., Galligo, A., Lombardi, H. (1997). Certified approximate polynomial gcds. *J. Pure Appl. Algebr.*, **117/118**, 229–251.
Fortune, S. (2001). Polynomial root finding using iterated eigenvalue computation. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC '01)*, pp. 121–128. New York, ACM Press.
Gauss, C. F. (1973). *Werke,* Band X. New York, Georg Olms Verlag.
Gel'fond, A. (1958). *Differenzenrechnung, Deutsher Verlag Der Wissenschaften*, Berlin (Russian edition: Moscow, 1952).
Gemignani, L. (2001). A generalized Graeffe's iteration for evaluating polynomials and rational functions. In *Proceedings of International Symposium on Symbolic and Algebraic Computation (ISAAC '01)*, pp. 143–149. New York, ACM Press.
Gragg, W. B. (1972). The Padé table and its relation to certain algorithms of numerical analysis. *SIAM Rev.*, **14**, 1–62.
Grau, A. A. (1971). The simultaneous improvement of a complete set of approximate factors of a polynomial. *SIAM J. Numer. Anal.*, **8**, 425–438,.
Greengard, L. (1988). *Rapid Evaluation of Potential Fields in Particle Systems*. Cambridge, MA, MIT Press.
Henrici, P. (1974). *Applied and Computational Complex Analysis 1*. New York, Wiley.
Householder, A. S. (1970). *The Numerical Treatment of a Single Nonlinear Equation*. New York, McGraw-Hill.
Kapur, D., Lakshman, Y. N. (1992). In Donald, B., Kapur, D., Mundy, J. eds, *Elimination Methods: An Introduction, Symbolic and Numerical Computation for Artificial Intelligence*, pp. 45–89. New York, Academic Press.
Karatsuba, A., Ofman, Yu. (1963). Multiplication of multidigit numbers on automata. *Sov. Phys.–Dokl.*, **7**, 595–596.
Kim, M.-H., Sutherland, S. (1994). Polynomial root-finding algorithms and branched covers. *SIAM J. Comput.*, **23**, 415–436.
Kirrinnis, P. (1998). Polynomial factorization and partial fraction decomposition by simultaneous Newton's iteration. *J. Complexity*, 378–444.
Kirrinnis, P. (1992). Fast computation of numerical partial fraction decompositions and contour integrals of rational functions. In Wang, P. S. ed., *Proceedings of International Symposium on Symbolic and Algebraic Computation (ISSAC 92)*, pp. 16–26. New York, ACM Press.
Malajovich, G., Zubelli, J. P. (1997). A fast and stable algorithm for splitting polynomials. *Comput. Math Appl.*, **33**, 1–23.
Malajovich, G., Zubelli, J. P. (2001). Tangent Groeffe iteration. *Numer. Math.*, **89**, 749–782.
McNamee, J. M. (1993). Bibliography on roots of polynomials. *J. Comput. Appl. Math.*, **47**, 391–394.
McNamee, J. M. (1997). A supplementary bibliography on roots of polynomials. *J. Comput. Appl. Math.*, **78**; also `http://www.elsevier.nl/homepage/sac/cam/mcnamee/index.html`.
Mignotte, M. (1974). An inequality about factors of polynomials. *Math. Comput.*, **28**, 1153–1157.
Mourrain, B., Pan, V. Y. (May 1998). Asymptotic acceleration of solving polynomial systems. In *Proceedings of 27th Annual ACM Symposium on Theory of Computing*, pp. 488–496. New York, ACM Press.

Mourrain, B., Pan, V. Y. (2000). Multivariate polynomials, duality and structured matrices. *J. Complexity*, **16**, 110–180.
Neff, C. A. (1994). Specified precision polynomial root isolation is NC. *J. Comput. Syst. Sci.*, **48**, 429–463.
Neff, C. A., Reif, J. H. (1994). An $O(n^{l+\epsilon})$ algorithm for the complex root problem. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pp. 540–547. Los Alamitos, CA, IEEE Computer Society Press.
Neff, C. A., Reif, J. H. (1996). An efficient algorithm for the complex roots problem. *J. Complexity*, **12**, 81–115.
Pan, V. Y. (1987). Sequential and parallel complexity of approximate evaluation of polynomial zeros. *Comput. Math. Appl.*, **14**, 591–622.
Pan, V. Y. (May 1995a). Optimal (up to polylog factors) sequential and parallel algorithms for approximating complex polynomial zeros. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, pp. 741–750. New York, ACM Press.
Pan, V. Y. (1995b). Deterministic improvement of complex polynomial factorization based on the properties of the associated resultant. *Comput. Math. Appl.*, **30**, 71–94.
Pan, V. Y. (1996). Optimal and nearly optimal algorithms for approximating polynomial zeros. *Comput. Math. Appl.*, **31**, 97–138.
Pan, V. Y. (1997). Solving a polynomial equation: some history and recent progress. *SIAM Rev.*, **39**, 187–220.
Pan, V. Y. (1998). Approximate polynomial gcds, Padé approximation, polynomial zeros, and bipartite graphs. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 68–77. New York, ACM Press and Philadelphia, SIAM Publications.
Pan, V. Y. (2000). Approximating complex polynomial zeros: modified quadtree (Weyl's) construction and improved Newton's iteration. *J. Complexity*, **16**, 213–264.
Pan, V. Y. (2001a). *Structured Matrices and Polynomials: Unified Superfast Algorithms.* Boston, Birkhäuser and New York, Springer.
Pan, V. Y. (2001b). Univariate polynomials: nearly optimal algorithms for factorization and rootfinding. In *Proceedings of the International Symposium on Symbolic and Algorithmic Computation*, pp. 253–267. New York, ACM Press.
Pan, V. Y. (2001c). Computation of approximate polynomial GCD's and an extension. *Inf. Comput.*, **167**, 71–85.
Pan, V. Y. (2002). Exploiting matrix structure and eigendecomposition for polynomial rootfinding, preprint.
Pan, V. Y., Chen, Z. Q. (1999). The complexity of the matrix eigenproblem. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pp. 507–516. New York, ACM Press.
Pan, V. Y., Wang, X. (2002). Improved computations with structured matrices, preprint.
Pólya, G., Szegö, G. (1925). *Aufgaben und Lehrsätze aus der Analysis.* Berlin, Springer.
Renegar, J. (1987). On the worst-case arithmetic complexity of approximating zeros of polynomials. *J. Complexity*, **3**, 90–113.
Samet, H. (1984). The quadtree and related hierarchical data structures. *Comput. Surv.*, **16**, 187–260.
Schönhage, A. (1982a). Asymptotically fast algorithms for the numerical multiplication and division of polynomials with complex coefficients. In *Proceedings of EUROCAM*, Marseille, LNCS, pp. 3–15. Berlin, Springer.
Schönhage, A. (1982b). *The Fundamental Theorem of Algebra in Terms of Computational Complexity*, Mathematics Department, University of Tübingen, Tübingen, Germany.
Schönhage, A. (1985). Quasi-gcd computations. *J. Complexity*, **1**, 118–137.
Schönhage, A. (2000). Variations on computing reciprocals of power series. *Inf. Process. Lett.*, **74**, 41–46.
Schönhage, A., Strassen, V. (1971). Schnelle multiplikation grosse Zahlen. *Computing*, **7**, 281–292.
Senoussi, H. (1994). A quadtree algorithm for template matching on pyramid computer. *Theor. Comput. Sci.*, **136**, 387–417.
Smale, S. (1981). The fundamental theorem of algebra and complexity theory. *Bull. Am. Math. Soc.*, **4**, 1–36.
Smale, S. (1985). On the efficiency of algorithms of analysis. *Bull. Am. Math. Soc.*, **13**, 87–121.
Van Dooren, P. M. (1994). Some numerical challenges in control theory. In *Linear Algebra for Control Theory,* Volume 62 of *IMA Vol. Math. Appl*, Springer.
Weyl, H. (1924). Randbemerkungen zu Hauptproblemen der Mathematik, II, Fundamentalsatz der Algebra and Grundlagen der Mathematik. *Math. Z.*, **20**, 131–151.
Wilson, G. T. (1969). Factorization of the covariance generating function of a pure moving-average process. *SIAM J. Numer. Anal.*, **6**, 1–7.

Received 13 November 2001
Accepted 30 March 2002