JOURNAL OF COMPLEXITY **14**, 378–444 (1998)
ARTICLE NO. CM980481

# Partial Fraction Decomposition in $\mathbb{C}(z)$ and Simultaneous Newton Iteration for Factorization in $\mathbb{C}[z]$

Peter Kirrinnis

*Universität Bonn, Institut für Informatik II, Römerstr. 164, D-53117 Bonn, Germany*
E-mail: kirr@cs.uni-bonn.de

Received July 1996

The subject of this paper is fast numerical algorithms for factoring univariate polynomials with complex coefficients and for computing partial fraction decompositions (PFDs) of rational functions in $\mathbb{C}(z)$. Numerically stable and computationally feasible versions of PFD are specified first for the special case of rational functions with all singularities in the unit disk (the "bounded case") and then for rational functions with arbitrarily distributed singularities. Two major algorithms

toring polynomials in $\mathbb{C}[z]$ to an algorithm for PFD. The second algorithm is a Newton iteration for simultaneously improving the accuracy of all factors in an approximate factorization of a polynomial resp. all partial fractions of an approximate PFD of a rational function. Algorithmically useful starting value conditions for the Newton algorithm are provided. Three subalgorithms are of independent interest. They compute the product of a sequence of polynomials, the sum of a sequence of rational functions, and the modular representation of a polynomial. All algorithms are described in great detail, and numerous technical auxiliaries are provided which are also useful for the design and analysis of other algorithms in computational complex analysis. Combining the splitting circle method with simultaneous Newton iteration yields favourable time bounds (measured in bit operations) for PFD, polynomial factoring, and root calculation. In particular, the time bounds for computing high accuracy PFDs, high accuracy factorizations, and high accuracy approximations for zeros of squarefree polynomials are linear in the output size (and hence optimal) up to logarithmic factors.
© 1998 Academic Press, Inc.

## 1. INTRODUCTION

### 1.1. *The Problem and Some History*

Computing factors or roots of polynomials with complex coefficients and partial fraction decompositions (PFDs) of rational functions are fundamental issues of computational complex analysis. Both problems are closely related: The factorization problem is a subproblem of the PFD problem. Conversely, some factorization methods use subroutines which are in fact PFD algorithms.

Algorithms for the root finding problem are legion. The most important locally convergent algorithm is Newton's method. Smale (1981, 1986), Shub and Smale (1985, 1986), Renegar (1985), and Friedman (1989) have investigated its efficiency and probability of success with random starting points.

Prominent classical examples for globally convergent root finding algorithms are given by Weyl (1924) and Lehmer (1961). Henrici (1974, Chap. 6) describes a variety of classical techniques, including methods for finding regions containing zeros.

In the last years, there has been great progress in the development of asymptotically fast (in the sense of worst case sequential or parallel arithmetic or bit complexity) deterministic, globally convergent root finding algorithms.

These algorithms can be divided into two classes: Algorithms like those given by Renegar (1987), Pan (1994), and Schönhage (1996) compute a moderate precision approximation for one root with a variant of some classical root finding algorithm and use Newton iteration for high precision root approximations. Further roots are then computed after deflation.

The other approach is to split a polynomial $p$ into two factors $f$ and $g$ corresponding to disjoint sets of zeros of $p$ and then proceed recursively. Algorithms of this type are Schönhage's "splitting circle method" (1982b; 1986, Sect. 3), Neff's parallel algorithm (1994) showing for the first time that the root finding problem is in NC, and the recent algorithms by Neff and Reif (1996) and Pan (1996).

In practical implementations, the ubiquitous Newton's method and the Jenkins–Traub algorithm (1970) are widely used. Variants of Schönhage's "splitting circle method" have been implemented by Gourdon (1993, 1996) and by Schönhage (1997).

Compared with the vast literature on polynomial root finding and its complexity, little can be found about numerical PFD. Most PFD algorithms described in the literature assume that the roots of the denominator are known. Until now, the numerical PFD problem has not been analyzed in terms of bit complexity, and the relation of numerical stability and complexity has not been investigated.

This paper presents fast numerical algorithms for computing factors or roots of complex polynomials and PFDs of rational functions in one complex variable. Rigorous proofs for the correctness of the algorithms and for favourable asymptotic time bounds with respect to bit complexity (see Subsection 1.2) are given. The numerical methods and auxiliaries (e.g., error estimates) presented in this paper are independent of the computational model and hence useful for various approaches to computational complex analysis. Although the focus is on bit complexity, the algorithms will also result in good time bounds when arithmetic operations are counted, using, e.g., the model proposed by Blum, Shub, and Smale (1989).

An appropriate specification for approximate PFD (APFD) is obtained by extending the concept of approximate factorization of polynomials as specified, e.g., in Schönhage (1982b, 1986). Additional conditions are introduced to specify a concept which is numerically stable.

The first PFD algorithm is an extension of Schönhage's "splitting circle method." The factorization algorithms by Neff and Reif (1996) and Pan (1996) are based on similar techniques. It is plausible that these can be extended for PFD alike, but this needs further investigation.

The second algorithm is a multidimensional Newton method which improves the accuracy of all factors in an approximate factorization of a polynomial resp. all partial fractions of an APFD simultaneously. This algorithm uses ideas of Grau's (1971) and Schönhage's (1982b). Starting values for this Newton iteration can be computed by the extended splitting circle method. Two explicit and algorithmically testable sufficient conditions are provided for an approximate factorization resp. PFD to be a suitable starting value for this Newton algorithm.

For computing roots with high precision, i.e., up to an error which is significantly smaller than their distance, the combination of both algorithms yields a time bound for factorization which is linear in the output size up to logarithmic factors and hence almost optimal.

Some auxiliary algorithms are of independent interest. They compute the product of a sequence of polynomials, the sum of a sequence of rational functions, and the residues of a polynomial modulo a given set of moduli. These algorithms extend the well-known divide and conquer methods for computing symmetric functions and for the multiple evaluation of polynomials described, e.g., in Borodin and Munro (1975, Sect. 4.5).

The Introduction is structured as follows: Subsection 1.2 is an introduction to bit complexity. Subsections 1.3–1.6 summarize problem specifications and complexity results. Subsection 1.7 deals with history and related research. Subsection 1.8 discusses implementations and related software. Subsection 1.9 describes the organization of the paper.

## 1.2. *Bit Complexity and Data Representation*

The concept of bit complexity is based on the concept of recursive functions over the natural numbers. It is specified with respect to some class of machines over a finite alphabet, e.g., multitape Turing machines over $\{0, 1\}$, random access machines with various instruction sets, or pointer machines (see, e.g., Schönhage (1980) for specifications). The running time of algorithms is measured with respect to an appropriate cost function for the underlying machine model. "Fast" algorithms are meant to be asymptotically fast with respect to such a model.

Data are encoded as follows: Integers are represented in some standard binary form. The set of "machine numbers" for the representation of complex numbers is the ring of binary rationals (dyadic numbers), $\{(a+ib)\cdot 2^{-s}: a, b, s \in \mathbb{Z}\} \subset \mathbb{C}$. Note that there is no machine number closest to a given complex number. As input numbers, complex numbers are given by *oracles* with arbitrary, but finite precision: An oracle for $\alpha \in \mathbb{C}$ is queried with a parameter $s \in \mathbb{N}$ and returns a binary rational $\tilde{\alpha}$ with $|\alpha - \tilde{\alpha}| < 2^{-s}$ without extra cost.

A polynomial is represented by its formal degree $n$ (all coefficients are allowed to be zero) and a dyadic approximation for its coefficient vector. The same scaling factor is used for all coefficients (block scaling): A polynomial $p \in \mathbb{C}[z]$ is approximated by a polynomial $2^{-s} \cdot P$, where $P \in \mathbb{Z}[i][z]$ and $s \in \mathbb{Z}$. The data stored are $n$, the coefficients of $P$, and the scaling parameter $s$. As input, polynomials are given by oracles. This model covers several other ways of specifying polynomials (e.g., by integer or rational coefficients) as special cases in a straightforward manner.

Equality of two complex numbers is undecidable, because an oracle for the complex number 0 may always return $2^{-(s+1)}$ when queried with error parameter $s$. An appropriate substitute for testing $a = b$ is testing "$\varepsilon$-equality": Given $\varepsilon > 0$, assert (at least) one of $|b-a| < \varepsilon$ or $|b-a| > \varepsilon/2$.

It is undecidable, too, whether two polynomials are relatively prime. Therefore, the concept of the "reduced form" of a rational function is not appropriate here. A rational function $f \in \mathbb{C}(z)$ is hence assumed to be given by *some* pair $(q, p)$ of polynomials such that $f = q/p$.

For a systematic treatise of further aspects of bit complexity in analysis see Ko (1991).

## 1.3. *Partial Fractions: Specifications and a Time Bound*

This section specifies the problems of approximate factorization and APFD and states a time bound for APFD in a standardized special case. For the sake of clarity, the general case is postponed to Subsection 1.4.

The formulation of specifications and complexity results requires some definitions: $\Pi$ denotes the algebra of univariate complex polynomials (polynomial functions), equipped with the $l_1$-norm $|\cdot|$ defined by $|p| := |a_0| + |a_1| + \cdots + |a_n|$ for $p(z) = a_0 z^n + a_1 z^{n-1} + \cdots + a_n$. For $n \in \mathbb{N}$, $\Pi_n$ is the subspace of $\Pi$ of polynomials of degree $\leqslant n$. $\Pi_n^1$ is the subset of $\Pi_n$ of monic polynomials with all roots inside the unit disc.

The *root finding* problem is usually specified as follows: "Given the coefficients of $p$, compute the roots of $p$ up to an error of $\varepsilon$." The concept of *approximate factorization* is more appropriate for many applications. The quality of the factors is measured in terms of the coefficients in the sense of backward analysis:

1.1. Definition. An *approximate factorization* of a polynomial $p \in \Pi_n$ within error $\varepsilon > 0$ is a sequence $(p_1, ..., p_l)$ of approximate factors $p_j \in \Pi_{n_j}$ such that $n_1 + \cdots + n_l = n$ and $|p - p_1 \cdots p_l| < \varepsilon$. This is denoted by "$p \approx p_1 \cdots p_l \operatorname{err} \varepsilon$".

The input polynomial $p$ should be normalized, e.g., $|p| = 1$ or $p \in \Pi_n^1$, because an absolute error bound $\varepsilon$ is prescribed. The relative error bound $\varepsilon \cdot |p|$ could be used instead. As the concept shall be most flexible, the definition specifies only how to measure the error, but it does not require further conditions for $p_j$. Even $l = 1$ is allowed, meaning that $p_1$ is an approximation for $p$. The aim is to compute a *complete* approximate factorization of $p$, i.e., all $p_j$ are linear factors.

A complete factorization $p \approx p_1 \cdots p_n \operatorname{err} \varepsilon$ with $\varepsilon = 2^{-s}$ can be computed in time $O(n^3 + n^2 \cdot s)$ (up to logarithmic factors), see Schönhage (1982b, Theorem 2.1; 1986, Sect. 3.2) and Pan (1996, Sect. 1.8).

If $p = p_1 \cdots p_l$ with pairwise prime factors $p_j$ and $q \in \mathbb{C}[z]$ with $\deg q < \deg p$, then there is a unique PFD $q/p = q_1/p_1 + \cdots + q_l/p_l$ with polynomials $q_1, ..., q_l$ such that $\deg q_j < \deg p_j$ for $j = 1, ..., l$.

The concept of *approximate* PFD is based on the following idea: Let $q_1/p_1 + \cdots + q_l/p_l = \tilde{q}/\tilde{p}$ with the common denominator $\tilde{p} = p_1 \cdots p_l$. The difference between $q/p$ and $\tilde{q}/\tilde{p}$ is measured by both $|q - \tilde{q}|$ and $|p - \tilde{p}|$:

1.2. Definition. Let $p \in \Pi_n$, $q \in \Pi_{n-1}$ and $\varepsilon, \eta > 0$. An *approximate PFD* of the rational function defined by $q/p$ within error $(\varepsilon, \eta)$ is a sequence $(p_1, q_1, ..., p_l, q_l)$ of polynomials such that $p \approx p_1 \cdots p_l \operatorname{err} \varepsilon$, $q_j \in \Pi_{n_j - 1}$, and $|q - q_1 r_1 - \cdots - q_l r_l| < \eta$, where $r_j = (p_1 \cdots p_l)/p_j$ for $j = 1, ..., l$. This is denoted by "$q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, \eta)$."

The goal is to compute an APFD with $p_j(z) = (z - v_j)^{n_j}$, where $v_1, ..., v_l \in \mathbb{C}$ are the different zeros of $p$ and $n_1, ..., n_l$ are their multiplicities. But in such a PFD, the coefficients of the numerators $q_j$ may be large if the roots of different factors $p_j$ of $p$ are close to each other. An example is

$$\frac{1}{z^n \cdot (z - \varepsilon) \cdot (z - 1)} = \frac{a(z)}{z^n} + \frac{b}{z - \varepsilon} + \frac{c}{z - 1}, \tag{1.1}$$

where $b = \varepsilon^{-n}/(\varepsilon - 1)$. This effect must be controlled to avoid numerical instabilities and to obtain PFDs which are useful for applications. The algorithms must be designed such that the zeros of different $p_j$ are sufficiently far away from each other. Therefore, clustered or multiple zeros of the denominator cannot be located with arbitrary precision. The condition that each $p_j$ is a power of a linear factor must be weakened: It is only required that all zeros of a factor $p_j$ are located in a disk of prescribed radius $\vartheta$:

1.3. Definition. For $p \in \Pi_n$, $q \in \Pi_{n-1}$, and $\varepsilon, \eta, \vartheta > 0$, a *radius $\vartheta$ decomposition* of $q/p$ within error $(\varepsilon, \eta)$ is an APFD $q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, \eta)$ such that for $j = 1, ..., l$, any two zeros $v, v'$ of $p_j$ fulfil $|v - v'| < 2 \cdot \vartheta$.

All algorithms described in this paper are based on fast integer multiplication. Let $\psi\colon \mathbb{N} \to \mathbb{R}$ be such that $N$ bit integers (in binary representation) can be multiplied in time $O(\psi(N))$. The best known time bounds are $\psi(N) = N \log(N+1) \log\log(N+2)$ for multitape Turing machines (Schönhage and Strassen, 1971), $\psi(N) = N \log(N+1)$ for successor RAMs under logarithmic cost, and $\psi(N) = N$ for pointer machines (Schönhage, 1980). For a vector $\mathbf{n} = (n_1, ..., n_l)$ of positive integers, $H(\mathbf{n})$ denotes the entropy of the probability vector $(n_1/n, ..., n_l/n)$, where $n = n_1 + \cdots + n_l$:

$$H(\mathbf{n}) := H(n_1, ..., n_l) := -\sum_{j=1}^{l} \frac{n_j}{n} \cdot \log \frac{n_j}{n} \qquad (\leqslant \log n). \tag{1.2}$$

The following theorem states a time bound for the computation of radius $\vartheta$ decompositions in the "bounded case" where all zeros of the denominator $p$ are inside the unit disc.

1.4. Theorem (Radius $\vartheta$ Decomposition, Bounded Case). *Let $p \in \Pi_n^1$ and $q \in \Pi_{n-1}$ with $|q| = 1$. Let $\varepsilon = 2^{-s_0}$, $\eta = 2^{-s_1}$, and $\vartheta = 2^{-\gamma}$ with $s_0$, $s_1$, $\gamma \in \mathbb{N}$. Then a radius $\vartheta$ decomposition $q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, \eta)$ can be computed from the coefficients of $p$ and $q$ in time $O(\psi(n^3 \cdot \log n + n^3 \cdot \gamma + n \cdot H(\mathbf{n}) \cdot (s_0 + s_1)))$. The zeros of different approximate factors $p_j$ have distance at least $\vartheta/(15n)$, and the $|q_j|$ fulfil the estimate $|q_j| \leqslant 2^{\gamma n + n \log n + O(n)}$.*

There are polynomials $p$ for which *any* reasonably precise radius $\vartheta$ decomposition of $1/p$ has approximate factors $p_i$, $p_j$ with $i \neq j$ which have zeros $v$ and $v'$, respectively, such that $|v - v'| \leqslant \vartheta/(2 \cdot n)$. An example is $p(z) = \prod_{j=1}^{n} (z - 2 \cdot \vartheta \cdot j/n)$. Therefore, the lower bound $\vartheta/(15n)$ for the root distance can only be improved by a constant factor. Likewise, one cannot expect a better bound than $|q_j| \leqslant 2^{\gamma n + n \log n + O(n)}$.

## 1.4. *Partial Fractions: The General Case*

In the general case, there are no restrictions but trivial standardizations like $|p| = |q| = 1$. The denominator $p$ may have zeros of arbitrary size. It is even allowed that the coefficient of $z^n$ vanishes, i.e., infinity may be a zero of $p$.

The clue for an appropriate specification of radius $\vartheta$ decomposition in the general case is a suitable distance measure for (large) complex numbers. A computationally feasible ad hoc solution is to represent numbers outside the unit disc by their reciprocals (Schönhage, 1982b, Sect. 19; 1986, Sect. 3.4). Viewing $\mathbb{C}(z)$ as the field of meromorphic functions on the compactification $\hat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ of the complex plane (with the standard topology) suggests another approach: Use one of the canonical representations of $\hat{\mathbb{C}}$, namely Riemann's Sphere $S^2$ or the one dimensional complex projective space $\mathbb{P}^1$, and a canonical metric defined on it.

The classical approach for the one dimensional situation is to use the topological isomorphism between $\hat{\mathbb{C}}$ and Riemann's Sphere $S^2$ via stereographic projection (see, e.g., Henrici, 1974, Sect. 5.3). This model of $\hat{\mathbb{C}}$ is preferred here, because it gives a good impression of the geometry near infinity. The standard metric on $\hat{\mathbb{C}}$ corresponding to this representation is the *chordal* (*stereographic*) *distance* $d_S$ which, in terms of the Euclidean distance, is

$$d_S(z, w) = \frac{2 \cdot |z - w|}{\sqrt{1 + |z|^2} \cdot \sqrt{1 + |w|^2}} \qquad \text{for} \quad z, w \in \mathbb{C}$$

and (1.3)

$$d_S(z, \infty) = \frac{2}{\sqrt{1 + |z|^2}} \qquad \text{for} \quad z \in \mathbb{C}.$$

1.5. Definition. For $p \in \Pi_n$, $q \in \Pi_{n-1}$, and $\varepsilon, \eta, \vartheta > 0$, a *generalized radius $\vartheta$ decomposition* (short: a *$\vartheta$-S-decomposition*) of $q/p$ within error $(\varepsilon, \eta)$ is an APFD $q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, \eta)$ where $d_S(v, v') < 2 \cdot \vartheta$ for any two zeros $v, v'$ of $p_j$ and all $j = 1, ..., l$.

This is a generalization of Definition 1.1, because $d_S$ is equivalent to the Euclidean distance when restricted to a bounded subset of $\mathbb{C}$, see Lemma 2.1. The problem of computing such decompositions can be reduced to the bounded case. The resulting time bound is the same:

1.6. Theorem (Radius $\vartheta$ Decomposition, General Case). *Let $p \in \Pi_n$ and $q \in \Pi_{n-1}$ with $|p| = |q| = 1$. Furthermore let $\varepsilon = 2^{-s_0}$, $\eta = 2^{-s_1}$, and $\vartheta = 2^{-\gamma}$. Then a $\vartheta$-S-decomposition $q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, \eta)$ can be computed in time $O(\psi(n^3 \cdot \log n + n^3 \cdot \gamma + n \cdot H(\mathbf{n}) \cdot (s_0 + s_1)))$. The zeros of different approximate factors $p_j$ have chordal distance at least $\vartheta/(120n)$, and the $|q_j|$ fulfil the estimate $|q_j| \leqslant 2^{\gamma n + n \log n + O(n)}$.*

## 1.5. *Newton Iteration for Factors of Polynomials*

The time bound for PFD stated in Theorem 1.4 is achieved with a combination of two algorithms. The first step is to compute a moderate precision radius $\vartheta$ decomposition by the extended splitting circle method. This algorithm can be used to compute a PFD to full accuracy, but this results in a time bound in which the term $n \cdot H(\mathbf{n}) \cdot (s_0 + s_1)$ is replaced with $n^2 \cdot (s_0 + s_1)$. The better time bound is achieved by increasing the accuracy by multidimensional Newton iteration, applied to the equation $p - p_1 \cdots p_l = 0$. This method is of independent interest.

The algorithm is locally convergent. For a complexity analysis and even more for efficient implementations, quantitative criteria for starting values are needed. It is useful to have criteria which can be verified without much additional work from data which are already available or must be computed anyhow. For $p \in \Pi_n^1$, the algorithm starts with an *initial decomposition* $1/p \approx h_1/p_1 + \cdots + h_l/p_l \operatorname{err}(\varepsilon_0, \eta_0)$. This is an initial decomposition if $\varepsilon_0, \eta_0$ are sufficiently small compared with $1/M$, where $M := \max_j |h_j|$. A simplified starting value condition which guarantees quadratic convergence of the Newton iteration is given by the inequalities $\varepsilon_0 \leqslant 2^{-11n}/M^4$ and $\eta_0 \leqslant 2^{-5n}/M$. For applications, it is useful to study the relation between $M$ and the distribution of the roots of the approximate factors $p_j$. It follows that if the zeros of different approximate factors $p_j$ of $p$ have distance at least $\delta$, then the Newton algorithm converges quadratically if $\varepsilon_0 \leqslant 2^{-22n} \cdot \delta^{4n}$ and $\eta_0 \leqslant 2^{-6n} \cdot \delta^n$. For a precise definition of initial decomposition and further discussion see Subsection 3.5.

The analysis of the Newton algorithm proves the following time bound:

1.7. Theorem (Improvement of Partial Fractions). *Given an initial decomposition* $1/p \approx h_1/p_1 + \cdots + h_l/p_l \operatorname{err}(\varepsilon_0, \eta_0)$ *for* $p \in \Pi_n^1$ *with* $p_j$ *monic and* $s_0, s_1 \in \mathbb{N}$, *an APFD* $1/p \approx \tilde{h}_1/\tilde{p}_1 + \cdots + \tilde{h}_l/\tilde{p}_l \operatorname{err}(2^{-s_0}, 2^{-s_1})$ *can be computed in time* $O(\psi(n \cdot H(\mathbf{n}) \cdot (s_0 + s_1)))$, *where* $\mathbf{n} = (n_1, ..., n_l)$. *The* $\tilde{p}_j$ *are monic, and the estimates* $|p_j - \tilde{p}_j| < M \cdot 2^{3n} \cdot \varepsilon_0$ *and* $|\tilde{h}_j| \leqslant 2M$ *hold.*

If $1/p = h_1/p_1 + \cdots + h_l/p_l$ and $q/p = q_1/p_1 + \cdots + q_l/p_l$, then $q_j = (q \cdot h_j) \bmod p_j = ((q \bmod p_j) \cdot h_j) \bmod p_j$. Using this reduction, once a PFD of $1/p$ is computed, a PFD of $q/p$ for arbitrary $q$ can be computed within the same asymptotic time bound:

1.8. Corollary (Improvement, Arbitrary Numerators). *Given an initial decomposition of* $1/p$ *as above,* $q \in \Pi_{n-1}$ *with* $|q| = 1$, *and* $s_0, s_1 \in \mathbb{N}$, *an APFD* $q/p \approx \tilde{q}_1/\tilde{p}_1 + \cdots + \tilde{q}_l/\tilde{p}_l \operatorname{err}(2^{-s_0}, 2^{-s_1})$ *can be computed in time* $O(\psi(n \cdot H(\mathbf{n}) \cdot (s_0 + s_1)))$. *The* $\tilde{p}_j$ *are monic, and* $|p_j - \tilde{p}_j| < M \cdot 2^{3n} \cdot \varepsilon_0$.

Simply "forgetting" the numerators $\tilde{h}_j$ in Theorem 1.7 results in a time bound for factorization which deserves an extra formulation:

1.9. Corollary. *Given an initial decomposition of* $1/p$ *as above and* $s \in \mathbb{N}$, *an approximate factorization* $p \approx \tilde{p}_1 \cdots \tilde{p}_l \operatorname{err} 2^{-s}$ *with* $\tilde{p}_j$ *monic can be computed in time* $O(\psi(n \cdot H(\mathbf{n}) \cdot s))$.

## 1.6. *Root Finding*

The idea of computing all roots of a complex polynomial simultaneously by multidimensional Newton iteration originates from the end of the 19th century, see Subsection 1.7 for references. The crucial problem is to find starting values (i.e., an initial decomposition) for this iteration (Kaltofen, 1992, Problem 4). One way is using the extended splitting circle method to compute moderate precision approximations for the linear factors of $p$. The precision is chosen such that the roots are isolated, but not significantly higher. Therefore, a better time bound (compared with that of Schönhage (1982b)) is achieved for computing the roots of a polynomial $p$ up to an error which is significantly smaller than the separation $\operatorname{sep}(p)$ of $p$, i.e., the minimum distance of different roots. As existence of multiple roots is undecidable, it is assumed that an a priori lower bound for $\operatorname{sep}(p)$ is known. Combination of Theorem 1.4 with a perturbation theorem for polynomial zeros implies the following result which, for the sake of simplicity, is only stated for polynomials with bounded root radius:

1.10 Theorem (Root Computation). *Let* $p \in \Pi_n^1$ *and* $\gamma \in \mathbb{N}$ *be given such that* $\operatorname{sep}(p) \geqslant 2^{-\gamma}$. *Let* $u_1, ..., u_l$ *denote the different zeros of* $p$, *and let* $s \in \mathbb{N}$. *Then approximations* $v_j$ *for* $u_j$ *with* $|v_j - u_j| < 2^{-s}$ (*and the multiplicities of the* $u_j$) *can be computed simultaneously for all* $j = 1, ..., l$ *in time* $O(\psi(n^3 \cdot \log n + n^3 \cdot \gamma + n^2 \cdot \log n \cdot s))$.

With respect to the accuracy parameter $s$, this time bound is optimal up to logarithmic factors (Pan, 1996, Fact 1.1). As $s \geqslant n$ is a realistic assumption anyhow (error amplification factors of the form $2^{cn}$ are inevitable in the computations), the time bound is optimal, if $n \cdot \gamma / \log n = O(s)$.

For squarefree polynomials, the roots are much less sensitive to errors of the coefficients. This saves a factor $n$ in the "high accuracy term" (proportional to $s$) of the time bound:

1.11. Theorem (Root Computation, Squarefree Polynomials). *In Theorem* 1.10, *let in addition* $p$ *be squarefree. Then the time bound is* $O(\psi(n^3 \cdot \log n + n^3 \cdot \gamma + n \cdot \log n \cdot s))$.

If $s$ is large compared with $\gamma$ and $n$ ($n^2 = O(s)$ and $\gamma \cdot n^2/\log n = O(s)$), then this time bound is linear in the output size (and hence optimal) up to logarithmic factors. A similar result holds for polynomials with roots of multiplicities bounded by a fixed constant.

The time bounds from Theorems 1.10 and 1.11 also hold for computing roots of polynomials with arbitrary root radius, where large roots are represented by their reciprocals or the error is measured by $d_S$. This follows by reduction to the bounded case.

An application of Theorem 1.11 is the computation of high accuracy approximations of the numerical values of algebraic numbers:

1.12 Theorem (Approximation of Algebraic Numbers). *Let $p \in \mathbb{Z}[x]$, $p(x) = a_0 x^n + a_1 x^{n-1} + \cdots + a_n$, $a_0 \neq 0$, and $|a_\nu| \leqslant 2^L$ for $\nu = 0, ..., n$. Then the roots of $p$ can be computed up to an absolute error of $2^{-s}$ in time $O(\psi(n^3 \cdot \log n + n^4 \cdot L + n \cdot \log n \cdot s))$.*

## 1.7. *Related Research and Open Problems*

The bit complexity of the numerical PFD problem has not been investigated before. The author is not even aware of an appropriate problem specification. Therefore, the new concepts of APFD, radius $\vartheta$ decomposition, and $\vartheta$-$S$-decomposition are introduced. There are no complexity results similar to Theorems 1.4 and 1.6.

The work on algorithms for computing all roots of a complex polynomial simultaneously by multidimensional Newton iteration starts with Weierstrass' proof of the fundamental theorem of algebra (1891). Variants and generalizations of this method have been discussed, e.g., by Samelson (1958), Kerner (1966), Schröder (1969), Grau (1971), Aberth (1973), Green *et al.* (1976), Pasquini and Trigiante (1985), Frommer (1988), and Carstensen (1992). These publications analyze the Newton algorithm from the usual point of view of numerical analysis, where floating point operations are counted. The algorithm has neither been described for arbitrary precision nor analyzed in terms of bit complexity. Moreover, the present analysis is the first which provides explicit starting value conditions.

From the plethora of complexity results for root computation, three results should be compared with Theorems 1.10, 1.11, and 1.12. These are Schönhage (1982b), Neff and Reif (1996), and Pan (1996). The other results take different points of view (arithmetic instead of bit complexity, probabilistic algorithms, parallel algorithms) or have time bounds worse than those from the publications mentioned above.

Schönhage's time bound for root computation is $O(\psi(n^3 \cdot \log n + s \cdot n^3))$ (1982b, Theorem 19.2). The major improvement over his method is the application of Newton iteration to all roots *simultaneously*. This saves a factor $n/\log n$ for large $s$. If $p$ admits "enough balanced splittings" (i.e., the factors $f$ and $g$ in the recursive splitting $p = f \cdot g$ both have roughly the same degree), then Schönhage's algorithm runs within the time bound stated in Theorem 1.11 as well, see (Schönhage, 1982b, Sect. 18). For merely *separating* the roots ($s = O(\gamma)$), Schönhage's method is used directly (even with the overhead for computing a PFD). Therefore, the time bound is not improved in the latter case.

The algorithms by Neff and Reif (1996) and Pan (1996) attack the root finding problem from a different direction. They use enhanced techniques for choosing splitting circles and enforce “balanced splittings” in this way. They both achieve time bounds $O(n^3 + n^2 \cdot s)$ up to logarithmic factors. For moderate precision, the Neff/Reif and Pan algorithms are superior to the one proposed here, because the extra effort to compute a PFD only pays for large $s$. For high precision, their time bounds are the same as in Theorem 1.11 (again up to logarithmic factors).

The numerators of the PFD are computed by a subroutine of Schönhage’s splitting circle method which is also used in the Neff/Reif and Pan algorithms. It is plausible that these algorithms can be extended to compute PFDs alike. An answer to whether this would result in a better time bound requires a careful analysis of quantitative details. This is subject to further research.

For integer polynomials, the time bound for merely *separating* the roots exceeds Schönhage’s time bound (1982b, Sect. 20) by a factor of $n$ (the $n^4 \cdot L$ term) due to the overhead for computing a PFD. For high precision (large $s$), the time bound from Theorem 1.12 is superior by a factor $n/\log n$ and optimal up to logarithmic factors.

The explicit or implicit restriction to polynomials with bounded roots is common: In Pan (1996) and Renegar (1987), such bounds are prescribed directly. Other authors prefer $p$ to be monic and the other coefficients of $p$ to be bounded (e.g., Smale, 1981; Renegar, 1985; Neff and Reif, 1996), or the coefficients of $p$ are assumed to be $L$ bit integers for some given $L$ (e.g., Pan, 1987, Sect. 1.2). All these restrictions imply an upper bound for the roots of $p$ (see Subsection 2.2). The general case (roots of arbitrary size) can be treated using a Moebius transform, hence bounding the roots is only a mild restriction.

It is more restrictive to assume that $p$ has bounded integer coefficients, because this implies a lower bound for $\operatorname{sep}(p)$, see Mignotte (1992, Sect. 4.6). Finally, it is common to assume integer polynomials to be squarefree, which can be achieved by replacing $p$ with $p_0 = p/\gcd(p, p')$. For the model used here, this would be a severe restriction because computing polynomial GCDs is a nontrivial task, see Schönhage (1985).

## 1.8. *Complexity Bounds and Implementations*

The aim of this paper is to prove asymptotic results in the sense of a worst case analysis. Therefore, few efforts have been spent on optimizing constant factors. Unless stated otherwise, numerical constants are stated explicitly for the sake of clarity only and may be improved by simple variants of the algorithm. Worst case bounds for rounding errors have been used throughout, though in implementations the accuracy should of course be controlled dynamically. Implementing the algorithms requires a more thorough discussion of details.

There are first steps towards implementations of fast high precision numerical algorithms of this type. An example is Gourdon's (1993) implementation of the splitting circle method in Maple and in the number theory package PARI which is integrated into the Magma system, see Gourdon (1996, III.7) for experimental results.

Schönhage has developed a software system for manipulating multiprecision data which is intended to serve as a basis for an efficient implementation of fast numerical algorithms like the splitting circle method. This system and the algorithms for integer and complex number arithmetic and for basic operations with polynomials implemented for it are described in Schönhage, Grotefeld, and Vetter (1994). An implementation of the splitting circle method on this system is currently being tested.

### 1.9. *Organization of the Paper*

Section 2 starts with a collection of definitions and then explains and justifies the restrictions made for the problem specifications. Section 3 gives high level descriptions of the algorithms for factorization, PFD, and root computation and states asymptotic time bounds for these algorithms. Section 4 is a collection of auxiliary estimates. The remainder of the paper (Sections 5–9) describes and analyzes the algorithms in detail.

Several proofs use similar ideas and are highly technical, but can be spelled out in a straightforward manner once the idea is clear. As too many technical details would obscure the ideas and not provide any new insight, these proofs have been put into an appendix.

## 2. SPECIFICATIONS

### 2.1. *Definitions*

$\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{C}$ denote the sets of natural numbers, integers, and complex numbers, respectively. Let $\mathbb{N}_+ := \mathbb{N}\setminus\{0\}$ and $[l]=\{1, ..., l\}$ for $l\in\mathbb{N}_+$. A *partition* $\mathbf{n}$ of $n\in\mathbb{N}$ into $l$ parts (briefly, *l-partition*; notation, $\mathbf{n}\vdash_l n$) is an integer vector $\mathbf{n}=(n_1, ..., n_l)\in\mathbb{N}^l$ which satisfies $n_1+\cdots+n_l=n$. If $1\leqslant n_1\leqslant\cdots\leqslant n_l$, then $\mathbf{n}$ is called an ordered partition and denoted as $\mathbf{n}\models_l n$. For $\mathbf{n}\vdash_l n$, $H(\mathbf{n})$ denotes the entropy of the probability vector $(n_1/n, ..., n_l/n)$, see (1.2). The cardinality of a finite set $M$ is denoted by $\# M$. The symbols log and ln denote the base-2 logarithm and the natural logarithm, respectively.

The open disk with radius $r>0$ and center $z_0\in\mathbb{C}$ is denoted by $D_r(z_0)$. In particular, $D := D_1(0)$. The closure, boundary, complement, and diameter of a set $G \subset \mathbb{C}$ are denoted by $\bar{G}$, $\partial G$, $\mathscr{C}G$, and $\operatorname{diam}(G) = \sup\{|z-z'|: z, z' \in G\}$, respectively. The distance of two sets $G, G' \subset \mathbb{C}$ is denoted by $\operatorname{dist}(G, G') = \inf\{|z-z'|: z \in G, z' \in G'\}$.

The leading coefficient and the exact degree of a polynomial $p \in \Pi$ are denoted by $\operatorname{lc}(p)$ and $\deg p$, respectively. Besides congruences, the symbol "mod" also denotes the corresponding operator: For $f, p \in \Pi$, $r = f \bmod p$ is the polynomial determined uniquely by $r \equiv f \bmod p$ and $\deg r < \deg p$.

For a linear operator $T: \Pi \to \Pi$, $\beta_n(T)$ denotes the norm of the restriction of $T$ on $\Pi_n$ with respect to $|\cdot|$. For $v \in \mathbb{C}$ resp. $r > 0$, $T_v$ resp. $S_r$ denote the operators of *translation* (*Taylor shift*) by $v$ resp. *scaling* by $r$, $(T_v p)(z) = p(z+v)$ and $(S_r p)(z) := p(rz)$ for $p \in \Pi$. The norms of these operators are $\beta_n(T_v) = (1+|v|)^n$ and $\beta_n(S_r) = \max\{1, r^n\}$. For $p(z) = a_0 z^n + a_1 z^{n-1} + \cdots + a_n$, $R_n p$ denotes the *reverse* polynomial, $(R_n p)(z) := z^n \cdot p(1/z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_0$. When $n$ is understood, the notation $p^* := R_n p$ is used.

The following concepts are useful for locating the roots of a polynomial $p$: $V(p) := \{v \in \mathbb{C} : p(v) = 0\}$ denotes the set of roots of $p$. The *root radius* of $p$ is $\varrho(p) := \max\{|v|: v \in V(p)\}$. The *inner root radius* of $p$ is $\varrho^*(p) := \min\{|v|: v \in V(p)\}$. If $p(z) = a_0 z^n + a_1 z^{n-1} + \cdots$ with $a_0 \neq 0$, then the *center of gravity* of the zeros of $p$ is $\bar{z}(p) := a_1/(n a_0)$. The diameter of $V(p)$ is approximated by the *centered root radius* of $p$ defined by $\bar{\varrho}(p) := \varrho(T_{\bar{z}(p)} p) = \max\{|\bar{z}(p) - v|: v \in V(p)\}$. It is easy to show that $\bar{\varrho}(p) \leqslant \operatorname{diam}(V(p)) \leqslant 2 \cdot \bar{\varrho}(p)$. A "small" disk containing all roots of $p$ is $DV(p) := D_{\bar{\varrho}(p)}(\bar{z}(p))$.

$\Pi_n^0$ denotes the set of monic polynomials of degree $n$. Then $\Pi_n^1 := \{p \in \Pi_n^0 : \varrho(p) \leqslant 1\}$. The perturbation of polynomial zeros is measured by $\Delta(p, \tilde{p}) := \min_{\pi \in S_n} \max_{j=1}^n |u_j - \tilde{u}_{\pi(j)}|$, where $p \in \Pi_n^0$ has zeros $u_1, ..., u_n$, $\tilde{p} \in \Pi_n^0$ has zeros $\tilde{u}_1, ..., \tilde{u}_n$, and $S_n$ denotes the symmetric group of $[n]$.

## 2.2. *On the Specification of the "Bounded Case"*

The algorithms for factorization and PFD and most of the auxiliary algorithms are designed for the standard special case (the "bounded case") $p \in \Pi_n^1$ and (for PFD) $q \in \Pi_{n-1}$ with $|q| = 1$, see Subsection 1.3. The general case (specified in Subsection 1.4 and discussed further in Subsection 2.3) is solved by reduction to the bounded case, see Subsection 3.7 for details. There are several reasons for starting with the bounded case.

Bounds for the size of the numbers involved in the computations are needed to control error propagation and amplification. For this purpose, a restriction like $|p| = |q| = 1$ would be sufficient.

Some subalgorithms (e.g., polynomial division) require that the denominator $p$ of a rational function $f = q/p$ given by polynomials $q \in \Pi$ and $p \in \Pi_n$ actually has the specified degree $n$. In particular, this is crucial for modular representations (Theorem 3.9) and for multidimensional Newton iteration (Theorem 1.7). For this purpose, $p$ is usually required to be monic, but this is not sufficient if the other coefficients of $p$ are much larger than 1. An upper bound for the other coefficients would be an appropriate supplement.

The standardization $\mathrm{lc}(p) = 1$ and $\varrho(p) \leqslant 1$ chosen here is equivalent in the following sense: Vieta's theorem implies that if $p(z) = z^n + a_1 z^{n-1} + \cdots + a_n$, then $|a_m| \leqslant \binom{n}{m} \cdot \varrho(p)^m$ for $1 \leqslant m \leqslant n$, hence $|p| \leqslant (1 + \varrho(p))^n$. Hence bounding the root radius of monic polynomials of degree $n$ by a constant implies a $2^{O(n)}$ upper bound for the coefficients, in particular $|p| \leqslant 2^n$ for $p \in \Pi_n^1$. Such a bound is acceptable for all algorithms discussed here. Conversely, bounds for the "lower" coefficients of $p$ imply bounds for the roots of $p$. A collection of such estimates can be found in Henrici (1974, Sect. 6.4). A typical one is the following: For $p$ as above, let $\sigma = \max_{1 \leqslant m \leqslant n} \sqrt[m]{|a_m|}$. Then $\sigma/n \leqslant \varrho(p) < 2\sigma$. The lower bound follows from the above estimate for $|a_m|$, and the upper bound is Henrici (1974, Corollary 6.4.k).

In the bounded case, the PFD problem can be restricted to the case $\deg q < \deg p$ w.l.o.g.: Otherwise compute $q_0$ and $r$ with $q = q_0 \cdot p + r$ and $\deg r < \deg p$ by polynomial division and compute a PFD of $r/p$. For details, see Subsection 3.6.

## 2.3. *On the Specification of the General Case*

A first step towards a general concept of radius $\vartheta$ decomposition is the discussion of appropriate representations for large roots of polynomials. For polynomials with bounded roots, there is no problem in specifying linear factors in the standard monic form $L_j = z - v_j$, i.e., with the corresponding root given explicitly. But if the roots of $p$ are allowed to be large, then this representation is no longer useful. Besides the fact that large numbers may cause intolerable error amplification, large roots are more sensitive to perturbations of the coefficient vector than small ones.

As an example consider $p(z) = (z-9)^3 = z^3 - 27z^2 + 243z - 729$ and $p_\varepsilon(z) = p(z) - (1-\varepsilon) \cdot z^3 = \varepsilon z^3 - 27z^2 + 243z - 729$ for $0 < \varepsilon < 1$. Then $p_\varepsilon$ approximates $p$ up to a relative error of less than 0.1 percent, i.e., $|p - p_\varepsilon|/|p| < 0.001$, but the roots of $p_\varepsilon$ converge to $\frac{1}{2}(9 \pm i \cdot 3\sqrt{3})$ and infinity for $\varepsilon \to 0$ and seem not to have much in common with the original ones. This shows that in general it does not make sense to use the Euclidean distance as a measure for the error of large roots.

A computationally feasible ad hoc solution for this problem has been proposed by Schönhage (1982b, Sect. 19; 1986, Sect. 3.4): Large numbers in the output are avoided by allowing the linear factors to be of the form $L_j = \alpha_j z + \beta_j$ with $|L_j| = 1$ or, more practical, $1 \leqslant |L_j| \leqslant 2$. Being slightly more restrictive and requiring in addition "$\alpha_j = 1$ or $\beta_j = 1$" suggests representing large roots by their reciprocals and measuring the quality of root approximations by the Euclidean distance of their reciprocals.

Here, the specification for the general case is derived from the special case by replacing the Euclidean distance with the chordal distance $d_S$ defined by (1.3). Both distance measures are equivalent on bounded subsets of $\mathbb{C}$:

2.1. LEMMA. $2/(1+A^2)\cdot|z-w| \leqslant d_S(z,w) \leqslant 2\cdot|z-w|$ *for* $|z|, |w| \leqslant A$.

*Proof.* The estimate follows from the definition of $d_S$ and $2/(1+A^2) \leqslant 2/\sqrt{(1+|z|^2)(1+|w|^2)} \leqslant 2$ for $|z|, |w| \leqslant A$. ∎

A second natural way of introducing a metric on $\hat{\mathbb{C}}$ is to embed $\hat{\mathbb{C}}$ into the one-dimensional projective space $\mathbb{P}^1$. This approach is better suited for multidimensional generalizations. It is based on the following idea.

The homogeneous polynomial corresponding to $p(z)=a_0z^n+a_1z^{n-1}+\cdots+a_n$ is $P(z_0,z_1)=a_0z_1^n+a_1z_1^{n-1}z_0+\cdots+a_nz_0^n\in\mathbb{C}[z_0,z_1]$. A vector $(v_0,v_1)\in\mathbb{C}^2$ is a root of $P$ iff $\lambda\cdot(v_0,v_1)$ is a root of $P$ for all $\lambda\in\mathbb{C}$. Hence, the roots of $P$ are viewed as elements $[v_0:v_1]:=\mathbb{C}\cdot(v_0,v_1)\in\mathbb{P}^1$.

The roots $v\in\hat{\mathbb{C}}$ of $p(z)=P(1,z)$ ($\infty$ is a root of $p$ iff $a_0=0$) correspond to the roots of $P$ via the canonical embedding $\mathbb{C}\ni v\mapsto[1:v]\in\mathbb{P}^1$ and $\infty\mapsto[0:1]$. Then representing a large root $v$ with its reciprocal means choosing the representative $(1/v,1)\in\mathbb{C}^2$ for the corresponding root $[1:v]=[1/v:1]\in\mathbb{P}^1$ of $P$.

The standard metric on the projective $n$-space $\mathbb{P}^n$ is the (up to scaling) unique unitarily invariant Hermitian metric $d_R$, which geometrically is the generalized angle between complex lines $\mathbb{C}\cdot u$ and $\mathbb{C}\cdot v$ given by

$$d_R(\mathbb{C}\cdot u,\mathbb{C}\cdot v)=\arccos\left(\frac{|\langle u,v\rangle|}{\|u\|\cdot\|v\|}\right)\qquad\text{for}\quad u,v\in\mathbb{C}^{n+1}\backslash\{0\}.$$

For reference see Shub and Smale (1993, Sect. I-3). Shub and Smale (1993) have defined another metric, namely the “projective distance”

$$d_P(\mathbb{C}\cdot u,\mathbb{C}\cdot v)=\min_{\lambda\in\mathbb{C}}\frac{\|u-\lambda v\|}{\|u\|}\qquad\text{for}\quad u,v\in\mathbb{C}^{n+1}\backslash\{0\}.$$

It has been shown there (Shub and Smale, 1993, Proposition 4) that $d_P(\mathbb{C}\cdot u,\mathbb{C}\cdot v)=\sin d_R(\mathbb{C}\cdot u,\mathbb{C}\cdot v)$. Composition of $d_P$ with the canonical embedding of $\mathbb{C}$ into $\mathbb{P}^1$ yields—after a short calculation—$d_P([z:1],[w:1])=\frac{1}{2}\cdot d_S(z,w)$. Hence, all three methods of defining a metric on $\hat{\mathbb{C}}$ yield equivalent distance measures:

$$\frac{1}{2}\cdot d_S=d_P\geqslant d_R\geqslant\frac{2}{\pi}\cdot d_P.$$

Thus, there is no essential difference in choosing any of these metrics as a distance measure for large roots of polynomials.

The degree condition $q \in \Pi_{n-1}$ in Theorem 1.6 is sufficiently general, because $p \in \Pi_n$ may have degree less than $n$. The condition $q \in \Pi_{n-1}$ has been chosen for technical convenience. Assuming $q \in \Pi_n$ instead would be more consistent with the concept of the degree of a rational function, $\deg(q/p) = \max\{\deg q, \deg p\}$ for relatively prime $q$ and $p$ (see van der Waerden, 1971, Sect. 73), which is invariant with respect to Moebius transforms.

If $\deg q = m$ is large compared with $n$, it is nevertheless preferable to replace $q$ with a numerator of smaller degree instead of calling the PFD algorithm with an artificially increased degree parameter $m \gg n$. This reduction can be done with polynomial division, but now polynomial division cannot be applied directly, because $p$ may have roots near infinity. One can proceed as follows: Choose $w \in \mathbb{C}$ with $|w| = 1$ such that $w$ is sufficiently far away from the roots of $p$ (cf. Schönhage, 1985, Lemma 2.5). Let $P = (T_w p)^*$ and $Q = (T_w q)^*$ and let $Q_0 \in \Pi_{m-n}$ and $R \in \Pi_{n-1}$ with $Q = Q_0 \cdot P + R$. Transforming back produces $q_0$ and $r$ with $q = q_0 \cdot p + r$. Then compute a PFD of $r/p$.

## 3. TIME BOUNDS AND OUTLINE OF ALGORITHMS

This section briefly describes algorithms and states time bounds for the computational problems stated in the introduction and for several subproblems which are of independent interest. The results are presented in the order in which the algorithms depend on each other.

### 3.1. *Basic Operations with Integers, Complex Numbers, and Polynomials*

All algorithms presented here are based on integer multiplication. Thus the time bounds are expressed in terms of a time bound $\psi$ for this task. It is assumed that $\psi$ fulfils the regularity condition that $\psi(N)/N$ is monotonic. This technical condition is explained in Subsection 4.1. For $x \in (0, \infty)$, $\psi(x)$ means $\psi(\lceil x \rceil)$.

3.1. Lemma (Complex Number Arithmetic). *The multiplication and division of complex numbers a and b up to a relative error of* $2^{-s}$, *i.e., computing some* $c \in \mathbb{C}$ *such that* $|c - ab| < 2^{-s} \cdot |ab|$ *resp.* $|c - a/b| < 2^{-s} \cdot |a/b|$, *is possible in time* $O(\psi(s))$.

*Proof.* This is obvious for multiplication, because complex numbers are represented by binary integers. Quotients of complex numbers are computed by Newton iteration, see Schönhage (1986, (2.2)) or, for details, Knuth (1981, Sect. 4.3.3). ∎

3.2. Lemma (Polynomial Multiplication). *Given $F, G \in \Pi_n$ and $s \in \mathbb{N}$ with $s \geqslant \log(n+1)$, a polynomial $P \in \Pi_{2n}$ with $|P - FG| < 2^{-s} \cdot |F| \cdot |G|$ can be computed in time $O(\psi(n \cdot s))$.*

3.3. Lemma (Discrete Fourier Transform). *Given $F \in \Pi_n$ with $|F| \leqslant 1$, $N \geqslant n+1$, and $s \in \mathbb{N}$, the simultaneous evaluation of $F$ in all $N$th roots of unity up to an error of $2^{-s}$, i.e., computing $(\alpha_0, ..., \alpha_{N-1}) \in \mathbb{C}^N$ such that $\sum_{j=0}^{N-1} |\alpha_j - F(\omega^j)| < 2^{-s}$, where $\omega := \exp(2\pi i/N)$, is possible in time $O(\psi(N \cdot (s + \log N)))$.*

3.4. Lemma (Polynomial Division). *Given $F \in \Pi_m$, $P \in \Pi_n$, with $|F| \leqslant 1 \leqslant |P| \leqslant 2$, $s \in \mathbb{N}$ with $s \geqslant m \geqslant n \geqslant 1$, and $r > 0$ with $\varrho(P) \leqslant r$, an approximate quotient $Q \in \Pi_{m-n}$ and an approximate remainder $R \in \Pi_{n-1}$ with $|F - (Q \cdot P + R)| < 2^{-s}$ can be computed in time $O(\psi(m^2 \cdot \log(1+r) + m \cdot s))$. If $r = O(1)$, then the time bound is $O(\psi(m \cdot s))$.*

3.5. Lemma (Remainder Computation). *In Lemma 3.4, the approximate remainder $R$ is computed such that in addition $|F \bmod P - R| < 2^{-s}$.*

3.6. Lemma (Taylor Shift). *Given $F \in \Pi_n$ with $|F| \leqslant 1$ and $v \in \mathbb{C}$ with $|v| \leqslant 2$, a polynomial $F_1 \in \Pi_n$ with $|F_1 - T_v F| < 2^{-s}$ can be computed in time $O(\psi(n \cdot (s+n)))$.*

For details and explanations see Schönhage (1982a; 1986, Sect. 4.3; 1990, Sect. 3). Lemmas 3.3 and 3.4 are proved in Schönhage (1982a, Sect. 3 resp. 4). Lemma 3.5 follows from the analysis of the division algorithm in Schönhage (1982a), see Appendix A.1 for details. Lemma 3.6 is Lemma 2.3 from Schönhage (1985). An implementation of algorithms matching these time bounds is described in Chapters 6, 8, and 9 of Schönhage, Grotefeld, and Vetter (1994).

## 3.2. *Symmetric Functions, Addition in $\mathbb{C}(z)$, and Modular Representations*

The algorithms by Moenck, Borodin, and Fiduccia (see, e.g., Borodin and Munro (1975, Sect. 4.5) for the computation of symmetric functions and the multiple evaluation of polynomials can be generalized to compute the product of several polynomials $p_1, ..., p_l$ resp. the residues of a polynomial modulo $p_1, ..., p_l$. The same idea can be applied for the addition of rational functions. A special refinement is obtained for factors of different degrees by using Huffman trees, following an idea of Strassen's (1983). The algorithms, which are described and analyzed in Section 5, yield the following time bounds, where $n \in \mathbb{N}$ and $\mathbf{n} \vdash_l n$ (remember $H(\mathbf{n}) \leqslant \log n$):

3.7. Theorem (Multiplication of Polynomials). *Let $p_j \in \Pi_{n_j}$ with $|p_j| \leqslant 2^{n_j}$ and $s \in \mathbb{N}$. Then $\tilde{p} \in \Pi_n$ with $|p_1 \cdots p_l - \tilde{p}| < 2^{-s}$ can be computed in time $O(\psi(n \cdot H(\mathbf{n}) \cdot (s+n)))$.*

3.8. THEOREM (Addition of Rational Functions). *Let $p_j \in \Pi_{n_j}$ with $|p_j| \leqslant 2^{n_j}$, $s \in \mathbb{N}$, $M \geqslant 1$, and $q_j \in \Pi_{n_j - 1}$ with $|q_j| \leqslant M$ for $j \in [l]$. Furthermore, let $r_j := p_1 \cdots p_{j-1} \cdot p_{j+1} \cdots p_l$ for $j \in [l]$, and let $q := q_1 r_1 + \cdots + q_l r_l$. Then an approximation $\tilde{q} \in \Pi_{n-1}$ for $q$ with $|q - \tilde{q}| < 2^{-s}$ can be computed in time $O(\psi(n \cdot H(\mathbf{n}) \cdot (s + n + \log M)))$.*

For a reasonable concept of modular representation, it is necessary that the moduli $p_j$ actually have the specified degree. Here it is required that they are monic and do not have large zeros. Then $p_j \in \Pi^1_{n_j}$ can be assumed w.l.o.g.

3.9. THEOREM (Modular Representation in $\mathbb{C}[z]$). *Let $p_j \in \Pi^1_{n_j}$ for $j \in [l]$, $f \in \Pi_m$ with $m \geqslant n$ and $|f| \leqslant 1$, and $s \in \mathbb{N}$. Then $\tilde{f}_j \in \Pi_{n_j - 1}$ with $|f \bmod p_j - \tilde{f}_j| < 2^{-s}$ can be computed simultaneously for all $j \in [l]$ in time $O(\psi((n \cdot H(\mathbf{n}) + m) \cdot (s + m)))$.*

## 3.3. *Computation of Root Moduli*

Algorithms for root radius computation and related problems are important tools for locating polynomial zeros. Schönhage (1982b, Sects. 14, 15) has proposed fast algorithms for these problems. They use Graeffe's root squaring method.

3.10. LEMMA. *Let $C > 0$. Given $p \in \Pi_n$ with $2^{-Cn} \leqslant \varrho(p) \leqslant 2^{Cn}$ and $\sigma > 0$, a radius $R > 0$ with $Re^{-\sigma} \leqslant \varrho(p) \leqslant Re^{\sigma}$ can be computed in time $O(\psi(n^2 \cdot (\log\log n + \log(1/\sigma)) \cdot \log(1/\sigma)))$. If $2^{-Cn} \leqslant \varrho^*(p) \leqslant 2^{Cn}$, then the same time bound applies for computing the inner root radius, $\varrho^*(p) = 1/\varrho(p^*)$.*

This has been shown by Schönhage (1982b, Theorem 15.1). The additional restriction $2^{-Cn} \leqslant \varrho(p) \leqslant 2^{Cn}$ is not mentioned there. However, such a restriction is necessary and is used implicitly in the proof. See Appendix A.2 for further discussion.

Approximations for the centered root radius $\bar{\varrho}(p) = \varrho(T_{\bar{z}(p)} p)$ can be computed by combining the root radius algorithm with a Taylor shift. The root perturbation caused by this Taylor shift should be small compared with $\bar{\varrho}(p)$. For the applications in this paper, it is sufficient to assert whether $\bar{\varrho}(p)$ exceeds a given bound $\vartheta$ and to compute a fixed precision approximation for $\bar{\varrho}(p)$ only in this case. See Appendix A.2 for further explanations and a proof of the complexity result.

3.11. DEFINITION. Let $p \in \Pi_n$ and $\vartheta > 0$. A *standard $\vartheta$-approximation* for $\bar{\varrho}(p)$ is a real number $\varrho \geqslant \vartheta$ such that at least one of $(\varrho = \vartheta \wedge \bar{\varrho}(p) < \vartheta)$ or $(\varrho > \vartheta \wedge 0.98\varrho < \bar{\varrho}(p) < \varrho)$ holds.

3.12. LEMMA. *Let $p \in \Pi^1_n$ and $\vartheta > 0$. Then a standard $\vartheta$-approximation for $\bar{\varrho}(p)$ can be computed in time $O(\psi(n^2 \cdot (\log\log n + \log(1/\vartheta))))$.*

## 3.4. *A Splitting Circle Method for PFD*

From now on, a modified definition of radius $\vartheta$ decomposition is used for technical reasons: The radius of $V(p_j)$ is replaced by $\bar{\varrho}(p_j)$, because approximations for $\bar{\varrho}(p_j)$ can be computed without knowing the roots of $p_j$.

3.13. Definition. For $p \in \Pi_n$, $q \in \Pi_{n-1}$, and $\varepsilon, \eta, \vartheta > 0$, a *radius $\vartheta$ decomposition* of $q/p$ within error $(\varepsilon, \eta)$ is an approximate partial fraction decomposition $q/p \approx q_1/p_1 \cdots q_l/p_l \operatorname{err}(\varepsilon, \eta)$ which fulfils $\bar{\varrho}(p_j) < \vartheta$ for all $j \in [l]$.

This definition differs only slightly from Definition 1.3, because $\bar{\varrho}(p) \leqslant \operatorname{diam}(V(p)) \leqslant 2 \cdot \bar{\varrho}(p)$. The distance of the roots of different factors is measured as follows:

3.14. Definition. Let $\delta > 0$. A sequence $(p_1, ..., p_l)$ of polynomials is called *$\delta$-separated*, if $\operatorname{dist}(V(p_i), V(p_j)) \geqslant \delta$ for all $i, j \in [l]$ with $i \neq j$, and if $\varrho(p_j) \leqslant 1 - \delta/2$ for all $j \in [l]$. An APFD $q/p \approx q_1/p_1 \cdots q_l/p_l \operatorname{err}(\varepsilon, \eta)$ is called $\delta$-separated if $(p_1, ..., p_l)$ is $\delta$-separated.

For technical reasons, the condition $\varrho(p_j) \leqslant 1$ is strengthened slightly. It is crucial for the algorithms that the numerators of $\delta$-separated PFDs are reasonably bounded, namely by $|q_j| \leqslant 2^{3n} \cdot \delta^{-n}$ (see Lemma 4.12).

The main idea of the splitting circle method is to process the following steps recursively: Given a polynomial $p$, variants of Graeffe's root squaring method are used to compute a "splitting circle" for $p$, i.e., a circle which encloses some, but not all zeros of $p$, and which is sufficiently far away from all zeros of $p$. With a Taylor shift and a scaling operation, the splitting circle is transformed into the unit circle. Now let $p = f \cdot g$, where the zeros of $f$ are inside the unit circle, while those of $g$ are outside. An initial approximation $f_0$ for $f$ is computed by the power sum method of Delves and Lyness (1967). Division of $p$ by $f_0$ yields an initial approximation $g_0$ for $g$. The error in this initial factorization is then decreased by multidimensional Newton iteration using a special version of the general method described in Subsection 3.5. For this Newton iteration an auxiliary polynomial $h$ is used which fulfils the congruence $h \cdot g \equiv 1 \bmod f$ approximately, hence $1/p = h/f + k/g$ (again approximately) for some polynomial $k$. An initial approximation for $h$ is also computed by power sums, and the accuracy of $h$ is increased by a special version of the quadratic iteration described in Subsection 3.5.

The computation of PFDs is hence implicit in Schönhage's algorithm. However, the algorithmic handling of the numerators and the problems which arise when zeros of different factors are close to one another require additional reasoning. Section 6 provides a detailed description of the method, comprising a summary of relevant technical results from Schönhage (1982b), and a careful analysis of quantitative aspects. The analysis results in a preliminary version of Theorem 1.4:

3.15. Theorem (Radius $\vartheta$ Decomposition, Preliminary). *Given $p \in \Pi_n^1$, $q \in \Pi_{n-1}$ with $|q| = 1$, $\varepsilon = 2^{-s_0}$, $\eta = 2^{-s_1}$, and $\vartheta = 2^{-\gamma}$ such that $\varrho(p) \leqslant 1 - \vartheta/(14n)$, a radius $\vartheta$ decomposition $q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, \eta)$ which is $\vartheta/(14n)$-separated can be computed in time $O(\psi(n^3 \cdot \log n + n^3 \cdot \gamma + n^2 \cdot (s_0 + s_1)))$. The $|q_j|$ fulfil the estimate $|q_j| \leqslant 2^{\gamma n + n \log n + O(n)}$.*

Subsection 3.6 explains how to achieve the better time bound stated in Theorem 1.4 and how to deal with numerators of degree $\geqslant n$. Subsection 3.7 treats the general case (arbitrary root radius).

## 3.5. *Newton Iteration*

Due to the special structure of the problem, it is convenient to describe the Newton algorithm in terms of polynomial arithmetic rather than with general matrix calculus. Let $\mathbf{n} \vdash_l n$. For $m \in \mathbb{N}$, consider $\Pi_m^0 = z^m + \Pi_{m-1}$ as an $m$-dimensional affine subspace of $\Pi$. The polynomials $p \in \Pi_n^0$ and $\hat{p}_j \in \Pi_{n_j}^0$ $(j \in [l])$ fulfil the equation $p = \hat{p}_1 \cdots \hat{p}_l$ iff $(\hat{p}_1, ..., \hat{p}_l)$ is a zero of the mapping

$$\alpha\colon \Pi_{n_1}^0 \times \cdots \times \Pi_{n_l}^0 \ni (p_1, ..., p_l) \mapsto p - p_1 \cdots p_l \in \Pi_{n-1}, \tag{3.1}$$

which can also be regarded as a mapping $\mathbb{C}^{n_1} \oplus \cdots \oplus \mathbb{C}^{n_l} \to \mathbb{C}^n$ of the vectors of the coefficients which are not prescribed to be 1. It is necessary for computing a zero $(\hat{p}_1, ..., \hat{p}_l)$ of $\alpha$ by Newton iteration that the Jacobian $J_\alpha$ of $\alpha$ is nonsingular at $(\hat{p}_1, ..., \hat{p}_l)$. This is the case iff the factors $\hat{p}_j$ are pairwise prime. (Recall that for $l = n$, $\det J_\alpha$ is the discriminant of $p$, which is nonzero iff $p$ has no multiple roots.) Then the approximate factors $p_j$ are pairwise prime if they are sufficiently close to $\hat{p}_j$. Newton's method yields the iteration rule $p_j \leftarrow \tilde{p}_j = p_j + \varphi_j$, where the Newton corrections $\varphi_j \in \Pi_{n_j - 1}$ are defined by the equation

$$\frac{p - p_1 \cdots p_l}{p_1 \cdots p_l} = \frac{\varphi_1}{p_1} + \cdots + \frac{\varphi_l}{p_l}.$$

If $|p - p_1 \cdots p_l| < \varepsilon$ with sufficiently small $\varepsilon$, then $|p - \tilde{p}_1 \cdots \tilde{p}_l| < C_1(n) \cdot \varepsilon^2$ with a moderate factor $C_1(n) > 0$. For $l = n$, i.e., if $\hat{p}_j$ are linear factors of $p$, this iteration is the "W-method" described in Section 1 of Pasquini and Trigiante (1985).

The Newton corrections $\varphi_j$ are computed by first computing a PFD

$$\frac{1}{p_1 \cdots p_l} = \frac{h_1}{p_1} + \cdots + \frac{h_l}{p_l} \tag{3.2}$$

with $h_j \in \Pi_{n_j - 1}$ up to a sufficiently small error, and then computing $\varphi_j$ from the congruences $\varphi_j \equiv (h_j \cdot p) \bmod p_j$. In this sense, the polynomials $h_j$ encode the inverse $(J_\alpha(p_1, ..., p_l))^{-1}$ of the Jacobian of $\alpha$. Instead of computing new $h_j$ for each Newton step, they are computed once at the beginning of the iteration and then adapted for new approximate factors of $p$ with the following iteration: The defect of an approximate solution of (3.2) is $d := 1 - h_1 r_1 - \cdots - h_l r_l$, where $r_j := p_1 \cdots p_{j-1} \cdot p_{j+1} \cdots p_l$. Replace $h_j$ with $\tilde{h}_j := (h_j \cdot (1 + d)) \bmod p_j$. Then the new defect $\tilde{d} := 1 - \tilde{h}_1 r_1 - \cdots - \tilde{h}_l r_l$ fulfils the congruence $\tilde{d} \equiv d^2 \bmod p_1 \cdots p_l$ which implies the estimate $|\tilde{d}| \leqslant C_2(n) \cdot |d|^2$ with a moderate factor $C_2(n) > 0$.

Sections 7 and 8 describe the algorithms for refining factors and partial fractions in detail. In particular, Section 8 provides a proof for Theorem 1.7. Corollary 1.8 is proved in Subsection 9.1.

Example (1.1) illustrates what happens when the Jacobian $J_\alpha(p_1, ..., p_l)$ is "nearly singular," i.e., approximate factors $p_i, p_j$ have roots close to each other. The concept of initial decomposition specifies whether an APFD of $1/p$ is sufficiently close to a solution such that quadratic convergence is guaranteed:

3.16. Definition. An *initial decomposition* for $p \in \Pi_n^1$ is an APFD $1/p \approx h_1/p_1 + \cdots + h_l/p_l \operatorname{err}(\varepsilon_0, \eta_0)$ which fulfils the following estimates with $M := \max_j |h_j|$:

$$\varepsilon_0 \leqslant \min\{2^{-9n}/(l^2 M^2), 2^{-4n}/(4l^2 M^4)\}$$

and

$$\eta_0 \leqslant \min\{2^{-4.5n}, 2^{-2n}/M\}.$$

A condition in terms of the roots of the $p_j$ is the following, see Subsection 4.4 for a proof.

3.17. Corollary. *Let $p \in \Pi_n^1$ and $\delta > 0$. Let $\mathbf{n} \vdash_l n$ and $p_j \in \Pi_{n_j}^0$ for $j \in [l]$ such that $(p_1, ..., p_l)$ is $\delta$-separated. Then an APFD $1/p \approx h_1/p_1 + \cdots + h_l/p_l \operatorname{err}(\varepsilon_0, \eta_0)$ is an initial decomposition if*

$$\varepsilon_0 \leqslant \min\{2^{-13n} \cdot \delta^{2n}/(l^2 n^4), 2^{-12n} \cdot \delta^{4n}/(l^2 n^8)\}$$

*and*

$$\eta_0 \leqslant \min\{2^{-4.5n}, 2^{-4n} \cdot \delta^n/n^2\}.$$

## 3.6. *High Accuracy Radius $\vartheta$ Decompositions in the Bounded Case*

The methods of Subsections 3.4 and 3.5 can be combined to provide an even faster algorithm for computing radius $\vartheta$ decompositions: Given a rational function $q/p$, one first computes a moderate precision radius $\vartheta$ decomposition $1/p \approx h_1/p_1 + \cdots + h_l/p_l \operatorname{err}(\varepsilon_0, \eta_0)$ which is $\vartheta/(14n)$-separated (Theorem 3.15). As the APFD is $\vartheta/(14n)$-separated, the $h_j$ are bounded by $\log |h_j| = O(n \log(1/\vartheta))$ (Lemma 4.12). Hence choosing the initial error bounds $\varepsilon_0$ and $\eta_0$ such that $\log(1/\varepsilon_0), \log(1/\eta_0) = O(n \log(n/\vartheta))$ guarantees that this APFD is an initial decomposition. The desired APFD of $q/p$ is now computed by Newton iteration according to Corollary 1.8. The algorithm is described and analyzed in Subsection 9.2. The analysis yields Theorem 1.4.

If $\deg q = m > n$, then one first computes a quotient $q_0 \in \Pi_{m-n}$ and a remainder $r \in \Pi_{n-1}$ with $|q - q_0 \cdot p - r| < \eta/2$ and then an APFD $r/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, \eta/2)$ as above. Together with the details spelled out in Subsection 9.2, this shows

3.18 Corollary (High Degree Numerators). *If in the above situation $q \in \Pi_m$ for some $m \geqslant n$, then a radius $\vartheta$ decomposition $q/p \approx q_0 + q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, \eta)$ which is $\vartheta/(15n)$-separated can be computed in time $O(\psi(n^3 \cdot \log n + n^3 \cdot \gamma + n \cdot H(\mathbf{n}) \cdot (s_0 + s_1) + m \cdot s_1))$. Again $|q_j| \leqslant 2^{\gamma n + n \log n + O(n)}$.*

## 3.7. *Computation of Radius $\vartheta$ Decompositions in the General Case*

The general case is reduced to the bounded case as in Schönhage (1982b, Sect. 7): If $\varrho(p) \leqslant 2$, then the bounded case is applied directly after scaling. If $\varrho^*(p) \geqslant 1/2$, then replace $z$ with $1/z$. This reduces the problem to computing an APFD of $q^*/p^*$, where $q^* = z^{n-1} \cdot q(1/z)$ and $p^* = z^n \cdot p(1/z)$ with $\varrho(p^*) \leqslant 2$. (Note that $f(1/z) = z \cdot q^*(z)/p^*(z)$. The "overhead" factor $z$ disappears with the backward transform.) If neither condition holds, then compute a radius $r \in (1/2, 2)$ which is sufficiently far away from all the roots of $p$ and an APFD $q/p \approx u/f + v/g$ with $\varrho(f) < r$ and $\varrho^*(g) > r$. Then compute APFDs of $u/f$ and $v/g$ according to the first and second case, respectively. Quantitative details of this reduction are discussed in Subsection 9.3. The analysis yields Theorem 1.6.

## 4. BOUNDS, ESTIMATES, AND OTHER AUXILIARIES

## 4.1. *Regularity Conditions for Time Bounds*

The analysis of algorithms involves sums of time bounds for subalgorithms like $\psi(m) + \psi(n)$ or $\psi(n) + \psi(n/2) + \psi(n/4) + \psi(n/8) + \cdots$. The latter is a typical time bound for the repeated application of a subroutine with increasing accuracy. Newton iteration is a prominent example. A regularity condition like the monotonicity of $\psi(n)/n$ implies convenient estimates for such quantities. The following lemma is used tacitly throughout this paper:

4.1. LEMMA. *Let $m, n \in \mathbb{N}$, $C, \gamma > 1$ and $t = \lceil \log_\gamma n \rceil$. Then $\psi(m) + \psi(n) \leqslant \psi(m+n)$, $\psi(C \cdot n) = O(\psi(n))$, and $\sum_{j=0}^{t} \psi(n/\gamma^j) = O(\psi(n))$.*

*Proof.* Let $\psi(n) = n \cdot \chi(n)$. Then $\psi(m) + \psi(n) = m \cdot \chi(m) + n \cdot \chi(n) \leqslant (m+n) \cdot \chi(m+n) = \psi(m+n)$. The second estimate is Brent (1976, Lemma 1.4). The third estimate follows with $\psi(n/2) \leqslant \psi(n)/2$ from condition (1.1) and Lemma 2.1 of Brent (1976). For more information about regularity conditions, see also Brent and Kung (1978, Sect. 1).

## 4.2. *Factors, Quotients, and Remainders of Polynomials*

4.2. LEMMA. *Let $p \in \Pi_n$ with $p = p_1 \cdots p_l$. Then $|p_1| \cdots |p_l| \leqslant 2^{n-1} \cdot |p|$.*

*Proof.* This bound is stated in Schönhage (1986, (3.3)). Generalizations and similar results can be found in Mignotte (1974), in Section 4.4 of Mignotte (1992), and in Schönhage (1982b, Sect. 4). ▮

The error analysis of factorization and PFD algorithms requires bounds for the size of quotients and remainders and error propagation estimates for polynomial division. For the proof of such estimates, the following lemma is used:

4.3. LEMMA. *Let $r > 0$ and $u_1, ..., u_n \in \mathbb{C}$ with $|u_\nu| \leqslant r$ for $\nu \in [n]$. Then the coefficients $c_k$ of the Laurent series*

$$\frac{1}{(1 - u_1/z) \cdots (1 - u_n/z)} = \sum_{k=0}^{\infty} \frac{c_k}{z^k} \qquad (|z| > r)$$

*fulfil the estimate $|c_k| \leqslant \binom{n+k-1}{k} \cdot r^k$.*

*Proof.* Expanding $1/(1 - u_\nu/z)$ into a geometric series yields

$$\sum_{k=0}^{\infty} \frac{c_k}{z^k} = \prod_{\nu=1}^{n} \left(1 + \frac{u_\nu}{z} + \frac{u_\nu^2}{z^2} + \cdots\right).$$

As $|u_\nu| \leqslant r$ for all $\nu$, $c_k$ is bounded by the coefficient of $t^k$ in the series

$$(1 + rt + (rt)^2 + \cdots)^n = \frac{1}{(1 - rt)^n} = \sum_{k=0}^{\infty} \binom{n+k-1}{k} \cdot r^k \cdot t^k. \quad ▮$$

4.4. Lemma. *Let $m \geqslant n \geqslant 1$, $f \in \Pi_m$, $p \in \Pi_n^1$, and $f = q \cdot p + r$ with $q \in \Pi_{m-n}$ and $r \in \Pi_{n-1}$ (which are uniquely determined). Then*

$$|q| \leqslant \binom{m}{n} \cdot |f| \leqslant 2^{m-1} \cdot |f|,$$

$$|r| \leqslant \left(1 + \binom{m}{n} \cdot |p|\right) \cdot |f| \leqslant (1 + 2^{m+n-1}) \cdot |f| \leqslant \frac{3}{4} \cdot 2^{m+n} \cdot |f|.$$

*Proof.* Let $f(z) = \sum_{j=0}^{m} \varphi_j z^j$ and $p(z) = \prod_{\nu=1}^{n} (z - u_\nu)$, where $|u_\nu| \leqslant 1$. Due to Lemma 4.3, the coefficients $c_k$ in the Laurent series of $1/p$ for $|z| > 1$,

$$\frac{1}{(1 - u_1/z) \cdots (1 - u_n/z)} = \sum_{k=0}^{\infty} \frac{c_k}{z^k},$$

are bounded by $|c_k| \leqslant \binom{n+k-1}{k}$. Now in the Laurent series of $f/p$ for $|z| > 1$,

$$\begin{aligned}\frac{f(z)}{p(z)} &= \frac{\varphi_m z^m + \cdots + \varphi_0}{z^n \cdot \prod_{\nu=1}^{n} (1 - u_\nu/z)} \\ &= (\varphi_m z^{m-n} + \cdots + \varphi_n + O(1/z)) \cdot \sum_{k=0}^{\infty} \frac{c_k}{z^k} \\ &= q_{m-n} z^{m-n} + \cdots + q_0 + O(1/z),\end{aligned}$$

obviously $q(z) = q_{m-n} z^{m-n} + \cdots + q_0$ holds. Hence

$$|q| = \sum_{j=0}^{m-n} |q_j| \leqslant |f| \cdot \sum_{k=0}^{m-n} |c_k| \leqslant |f| \cdot \sum_{k=0}^{m-n} \binom{n+k-1}{k} = |f| \cdot \binom{m}{n}.$$

Finally, $\binom{m}{n} \leqslant 2^{m-1}$, $|r| \leqslant |f| + |q| \cdot |p|$, and $p \leqslant 2^n$. ∎

The case $m = 2n - 2$ is typical for many applications. In this case, the bound for $|r|$ simplifies as follows:

4.5. Corollary. *In Lemma* 4.4, *let* $m = 2n - 2$. *Then* $|r| < \frac{1}{8} \cdot 2^{3n} \cdot |f|$.

*Proof.* Lemma 4.4 shows that $|r| \leqslant (1 + \binom{2n-2}{n} \cdot 2^n)$. The sequence $a_n := (1 + \binom{2n-2}{n} \cdot 2^n)/2^{3n}$ is monotonically decreasing, because $\binom{2n}{n+1}/2^{2n+2} < \binom{2n-2}{n}/2^{2n}$. The assertion follows with $a_1 = \frac{1}{8}$. Approximating the binomial coefficient with Stirling's formula yields a slightly better bound, namely $|r| < 0.142 \cdot 2^{3n} \cdot |f|/\sqrt{n}$. ∎

Lemma 4.4 and Corollary 4.5 can be used to estimate how a perturbation of the dividend in a polynomial division propagates into the quotient and remainder. The following lemma studies the effect of errors in the *divisor*:

4.6. Lemma. *Let* $m \geqslant n \geqslant 1$, $p, \tilde{p} \in \Pi_n^1$, *and* $f \in \Pi_m$. *Let* $f = qp + r = \tilde{q}\tilde{p} + \tilde{r}$ *with* $q, \tilde{q} \in \Pi_{m-n}$ *and* $r, \tilde{r} \in \Pi_{n-1}$ (*which are uniquely determined*). *Then*

$$|q - \tilde{q}| \leqslant \binom{m+n-1}{2n} \cdot |f| \cdot |p - \tilde{p}|,$$

$$|r - \tilde{r}| \leqslant \left(\binom{m}{n} + 2^n \cdot \binom{m+n-1}{2n}\right) \cdot |f| \cdot |p - \tilde{p}|.$$

*Proof.* The proof is similar to that of Lemma 4.4: Let $p(z) = \prod_{\nu=1}^n (z - u_\nu)$ and $\tilde{p}(z) = \prod_{\nu=1}^n (z - v_\nu)$ with $|u_\nu|, |v_\nu| \leqslant 1$, $\tilde{p}(z) - p(z) = \sum_{j=0}^{n-1} \varepsilon_j z^j$ and $f(z) = \sum_{j=0}^m \varphi_j z^j$. For $z \in \mathbb{C}$ with $|z| > 1$ let

$$\frac{1}{\prod_{\nu=1}^n ((1 - u_\nu/z)(1 - v_\nu/z))} = \sum_{k=0}^\infty \frac{c_k}{z^k}.$$

Then $|c_k| \leqslant \binom{2n+k-1}{k}$ due to Lemma 4.3. The Laurent series of $f/p - \tilde{f}/\tilde{p}$ for $|z| > 1$ is

$$\begin{aligned}
\frac{f(z)}{p(z)} - \frac{f(z)}{\tilde{p}(z)} &= \frac{f(z)}{p(z) \cdot \tilde{p}(z)} \cdot (\tilde{p}(z) - p(z)) \\
&= \frac{(\varphi_m z^m + \cdots + \varphi_0) \cdot (\varepsilon_{n-1} z^{n-1} + \cdots + \varepsilon_0)}{z^{2n} \cdot \prod_{\nu=1}^n ((1 - u_\nu/z)(1 - v_\nu/z))} \\
&= (\varphi_m z^{m-n-1} + \cdots + \varphi_{n+1} + O(1/z)) \\
&\quad \cdot \left(\varepsilon_{n-1} + \frac{\varepsilon_{n-2}}{z} + \cdots + \frac{\varepsilon_0}{z^{n-1}}\right) \cdot \sum_{k=0}^\infty \frac{c_k}{z^k} \\
&= \eta_{m-n-1} z^{m-n-1} + \cdots + \eta_0 + O(1/z),
\end{aligned}$$

where $q(z) - \tilde{q}(z) = \eta_{m-n-1} z^{m-n-1} + \cdots + \eta_0$. Hence

$$\begin{aligned}
|q - \tilde{q}| = \sum_{j=0}^{m-n-1} |\eta_j| &\leqslant |f| \cdot |p - \tilde{p}| \cdot \sum_{k=0}^{m-n-1} |c_k| \\
&\leqslant |f| \cdot |p - \tilde{p}| \cdot \sum_{k=0}^{m-n-1} \binom{2n+k-1}{k} \\
&= |f| \cdot |p - \tilde{p}| \cdot \binom{m+n-1}{2n}.
\end{aligned}$$

The error $|r-\tilde{r}|$ is now estimated using Lemma 4.4:

$$\begin{aligned}|r-\tilde{r}| &= |qp-\tilde{q}\tilde{p}| \leqslant |q|\cdot|p-\tilde{p}|+|\tilde{p}|\cdot|q-\tilde{q}| \\ &\leqslant \binom{m}{n}\cdot|f|\cdot|p-\tilde{p}|+2^n\cdot\binom{m+n-1}{2n}\cdot|f|\cdot|p-\tilde{p}|. \quad \blacksquare\end{aligned}$$

Combining Lemmas 4.4 and 4.6 yields:

4.7. LEMMA. *Let $m, n, p, \tilde{p}$ and $f$ as in Lemma 4.6. In addition, let $\tilde{f}\in\Pi_m$. Let $f=qp+r$ and $\tilde{f}=\tilde{q}\tilde{p}+\tilde{r}$ with $q, \tilde{q}\in\Pi_{m-n}$ and $r, \tilde{r}\in\Pi_{n-1}$ (which are uniquely determined). Then*

$$|r-\tilde{r}| \leqslant \tfrac{3}{4}\cdot 2^{m+n}\cdot|f-\tilde{f}|+\tfrac{3}{8}\cdot 2^{m+2n}\cdot|f|\cdot|p-\tilde{p}|.$$

*Proof.* Let $f=q\cdot\tilde{p}+\hat{r}$ with $\hat{r}\in\Pi_{n-1}$. Then $|r-\tilde{r}|\leqslant|r-\hat{r}|+|\hat{r}-\tilde{r}|$. Lemma 4.4 implies $|\hat{r}-\tilde{r}|\leqslant\frac{3}{4}\cdot 2^{m+n}\cdot|f-\tilde{f}|$, and Lemma 4.6 yields

$$\begin{aligned}|r-\hat{r}| &\leqslant \left(\binom{m}{n}+2^n\cdot\binom{m+n-1}{2n}\right)\cdot|f|\cdot|p-\tilde{p}| \\ &\leqslant \tfrac{3}{8}\cdot 2^{m+2n}\cdot|f|\cdot|p-\tilde{p}|. \quad \blacksquare\end{aligned}$$

4.8. LEMMA. *Let $p_j\in\Pi^1_{n_j}$, $h_j\in\Pi_{n_j-1}$, and $r_j := p_1\cdots p_{j-1}\cdot p_{j+1}\cdots p_l$ for $j\in[l]$. Then $|1-\sum_{j=1}^l h_j r_j|<1$ implies that the $p_j$ are pairwise prime.*

*Proof.* Otherwise it may be assumed w.l.o.g. that $p_1$ and $p_2$ have a common zero $z$ with $|z|\leqslant 1$. Then $\sum_{j=1}^l h_j(z)\, r_j(z)=0$, but $|1-\sum_{j=1}^l h_j(z)\, r_j(z)|\leqslant |1-\sum_{j=1}^l h_j r_j|<1$, a contradiction.

## 4.3. *Perturbation of Zeros*

The zeros of a polynomial $p$ can be computed from an approximate factorization $p\approx\tilde{p}$ err $\varepsilon$, where $\tilde{p}=p_1\cdots p_n$ with linear factors $p_j$. The crucial question is which error bound $\varepsilon$ for the factorization is sufficient to guarantee a prescribed accuracy for the zeros. The simple example $p(z)=z^n-\varepsilon$, $p_j(z)=z$ (i.e., $\tilde{p}(z)=z^n$) shows that an error $\varepsilon$ with respect to the $l_1$-norm may perturb the zeros by $\sqrt[n]{\varepsilon}$. A well known perturbation bound due to Ostrowski (1966, Appendix A) is $\Delta(p,\tilde{p})\leqslant 2n\cdot\sqrt[n]{\varepsilon}$ for $p, \tilde{p}\in\Pi^1_n$. This estimate has been improved by Schönhage to $\Delta(p,\tilde{p})<4\cdot\sqrt[n]{\varepsilon}$ (Schönhage, 1982b, Theorem 19.1). For a similar result with a different standardization see Schönhage (1985, Theorem 2.7). Schätzle (1990, Korollar 3.3) has proved the "true" factor in this perturbation estimate to be $2+o(1)$:

4.9. LEMMA. *Let $n \in \mathbb{N}_+$ and $0 < \sigma < \frac{1}{2}$. Then for all $p, \tilde{p} \in \Pi_n^1$ with $\varepsilon = |p - \tilde{p}| \leqslant \sigma$, the estimate $\Delta(p, \tilde{p}) < a_{n,\sigma} \cdot \sqrt[n]{|\varepsilon|}$ holds, where*

$$\alpha_n := \sqrt[n]{\binom{n-1}{\lfloor (n-1)/2 \rfloor}} \qquad \text{and} \qquad a_{n,\sigma} := \frac{\alpha_n}{\sqrt[n]{1-\sigma} - \sqrt[n]{\sigma}}.$$

*Moreover, always $\alpha_n \leqslant 2$, and, e.g., $a_{n,\sigma} \leqslant 2.134$ for $\sigma \leqslant 2^{-4n}$, and $a_{n,\sigma} \leqslant 2.008$ for $\sigma \leqslant 2^{-8n}$.*

4.10. LEMMA. *Let $p_j, \tilde{p}_j \in \Pi_{n_j}^1$ for $j \in [l]$. Let $(p_1, ..., p_l)$ be $\delta$-separated and let $0 < \delta_0 \leqslant \delta_1 < \delta - \delta_0$ be such that $\Delta(p_1 \cdots p_l, \tilde{p}_1 \cdots \tilde{p}_l) \leqslant \delta_0$ and $\Delta(p_j, \tilde{p}_j) \leqslant \delta_1$ for all $j \in [l]$. Then $\Delta(p_j, \tilde{p}_j) \leqslant \delta_0$ for all $j \in [l]$, and $(\tilde{p}_1, ..., \tilde{p}_l)$ is $(\delta - 2\delta_0)$-separated.*

*Proof.* Let the roots $u_1, ..., u_n$ of $p_1 \cdots p_l$ and $v_1, ..., v_n$ of $\tilde{p}_1 \cdots \tilde{p}_l$ be numbered such that $|u_\kappa - v_\kappa| \leqslant \delta_0$ for $\kappa \in [n]$. Now let $u_\kappa$ be a root of $p_j$ and $v_\kappa$ be a root of $\tilde{p}_k$. As $\Delta(p_k, \tilde{p}_k) \leqslant \delta_1$, there is a root $u$ of $p_k$ with $|u - v_\kappa| \leqslant \delta_1$. Hence $|u - u_\kappa| \leqslant \delta_0 + \delta_1 < \delta$ and thus $j = k$, because $(p_1, ..., p_l)$ is $\delta$-separated. As $p_j$ and $\tilde{p}_j$ have the same number of zeros, it follows that $M_j := \{\kappa \in [n] : p_j(u_\kappa) = 0\} = \{\kappa \in [n] : \tilde{p}_j(v_\kappa) = 0\}$ for $j \in [n]$, hence $\Delta(p_j, \tilde{p}_j) \leqslant \max_{\kappa \in M_j} |u_\kappa - v_\kappa| \leqslant \delta_0$. ∎

## 4.4. *Initial Decompositions and Root Clusters*

Upper bounds for the numerators $h_j$ in a PFD $1/p = h_1/p_1 + \cdots + h_l/p_l$ can be derived from lower bounds for the distance of the roots of different $p_j$. The following lemma states such an estimate for $l = 2$:

4.11. LEMMA. *Let $G \subset \mathbb{C}$ be open and bounded by finitely many (positively oriented) piecewise regular Jordan curves. Let $k, n \in \mathbb{N}$, $n > k$, $f \in \Pi_k$ with* deg *$f = k$ and $V(f) \subset G$, $g \in \Pi_{n-k}$ with $V(g) \subset \mathscr{C}\bar{G}$, $u \in \Pi_{k-1}$, $v \in \Pi$, $p := fg$, and $q := vf + ug$. Then*

$$u(z) = \frac{1}{2\pi i} \cdot \int_{\partial G} \frac{q(t)}{p(t)} \cdot \frac{f(t) - f(z)}{t - z}\, dt. \tag{4.1}$$

*If $|t| \leqslant R$ for $t \in \partial G$ and $C(R) :=$ max$\{1, R^{k-1}\}$, then*

$$|u| \leqslant \frac{1}{2\pi} \cdot C(R) \cdot |f'| \cdot \int_{\partial G} \left| \frac{q(t)}{p(t)} \right| |dt|. \tag{4.2}$$

*Finally, let* $|q(t)/p(t)| \leqslant A$ *for* $t \in \partial G$. *Then*

$$|u| \leqslant \frac{\operatorname{len}(\partial G)}{2\pi} \cdot A \cdot C(R) \cdot |f'|. \tag{4.3}$$

*Proof.* First let

$$u_1(z) := \frac{1}{2\pi i} \cdot \int_{\partial G} \frac{u(t)}{f(t)} \cdot \frac{f(t)-f(z)}{t-z}\, dt;$$

then $u_1 \in \Pi_{k-1}$, and $u(z) = u_1(z)$ holds for $z \in V(f)$ because of Cauchy's integral formula. Thus $u = u_1$, if $f$ has $k$ distinct zeros. Otherwise $u = u_1$ can be shown using a confluence argument (i.e., by approximating $f$ by polynomials $\tilde{f} \in \Pi_k$ which are squarefree). $V(g) \subset \mathscr{C}\bar{G}$ and Cauchy's theorem imply that

$$\int_{\partial G} \frac{v(t)}{g(t)} \cdot \frac{f(t)-f(z)}{t-z}\, dt = 0,$$

and (4.1) follows with $u = u_1$:

$$\begin{aligned} u(z) &= \frac{1}{2\pi i} \cdot \int_{\partial G} \left( \frac{u(t)}{f(t)} + \frac{v(t)}{g(t)} \right) \cdot \frac{f(t)-f(z)}{t-z}\, dt \\ &= \frac{1}{2\pi i} \cdot \int_{\partial G} \frac{q(t)}{p(t)} \cdot \frac{f(t)-f(z)}{t-z}\, dt. \end{aligned}$$

Now let $f(z) = \varphi_0 z^k + \varphi_1 z^{k-1} + \cdots + \varphi_k$. Then $f(z) - f(t) = \sum_{j=0}^{k} \varphi_{k-j} \cdot (z^j - t^j)$ and

$$\begin{aligned} \frac{f(z)-f(t)}{z-t} &= \sum_{j=0}^{k} \varphi_{k-j} \cdot (z^{j-1} + t z^{j-2} + \cdots + t^{j-1}) \\ &= \sum_{m=0}^{k-1} z^m \sum_{j=m+1}^{k} t^{j-m-1} \varphi_{k-j}. \end{aligned}$$

With (4.1), this yields a representation for the coefficients of $u(z) = \sum_{m=0}^{k-1} c_m z^m$:

$$c_m = \frac{1}{2\pi i} \int_{\partial G} \frac{q(t)}{p(t)} \cdot \sum_{j=m+1}^{k-1} t^{j-m} \varphi_{k-j}\, dt.$$

Now let $|t| \leqslant R$ for $t \in \partial G$. Then

$$\begin{aligned}|c_m| &\leqslant \frac{1}{2\pi} \cdot \int_{\partial G} \left( \left| \frac{q(t)}{p(t)} \right| \cdot \sum_{j=m+1}^{k} |t^{j-m-1}| \cdot |\varphi_{k-j}| \right) |dt| \\ &\leqslant \frac{1}{2\pi} \cdot \sum_{j=m+1}^{k} |R^{j-m-1}| \cdot |\varphi_{k-j}| \cdot \int_{\partial G} \left| \frac{q(t)}{p(t)} \right| |dt|\end{aligned}$$

and hence

$$\begin{aligned}|u| &\leqslant \frac{1}{2\pi} \cdot \sum_{m=0}^{k-1} \sum_{j=m+1}^{k} |\varphi_{k-j}| \cdot R^{j-m-1} \cdot \int_{\partial G} \left| \frac{q(t)}{p(t)} \right| |dt| \\ &\leqslant \frac{1}{2\pi} \cdot \max\{1, R^{k-1}\} \cdot \underbrace{\sum_{m=0}^{k-1} \sum_{j=m+1}^{k} |\varphi_{k-j}|} \cdot \int_{\Gamma} \left| \frac{q(t)}{p(t)} \right| |dt|,\end{aligned}$$

and the sum $\underbrace{\cdots}$ is

$$\sum_{m=0}^{k-1} \sum_{j=m+1}^{k} |\varphi_{k-j}| = k \cdot |\varphi_0| + (k-1) \cdot |\varphi_1| + \cdots + |\varphi_{k-1}| = |f'| \leqslant k \cdot |f|.$$

Now (4.2) follows immediately. Inequality (4.3) follows with $|q(t)/p(t)| \leqslant A$ for $t \in \Gamma$. ∎

Numerators of $\delta$-separated PFDs can now be estimated as follows:

4.12. LEMMA. *Let $p \in \Pi_n^1$ and $q \in \Pi_{n-1}$ with $|q| = 1$, $\varepsilon$, $\eta$, $\delta > 0$, and $l \in [n]$. Furthermore, let $q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, \eta)$, and let $(p_1, ..., p_l)$ be $\delta$-separated. Then*

$$|q_j| \leqslant n_j^2 \cdot (1+\eta) \cdot 2^n \cdot \delta^{-n} \cdot |p_j|. \tag{4.4}$$

*Proof.* Let $\tilde{p} := p_1 \cdots p_l$ and $\tilde{q} = (q_1/p_1 + \cdots + q_l/p_l) \cdot \tilde{p}$. For $j \in [l]$ let

$$G_j := \{ z \in D : \operatorname{dist}(z, V(p_i)) > \operatorname{dist}(z, V(p_j)) \text{ for all } i \neq j \}.$$

Obviously the $G_j$ are open and pairwise disjoint, and $\bigcup_{j=1}^{l} \overline{G_j} = \bar{D}$. Moreover, for $j \in [l]$ always $V(p_j) \subset G_j$, and for every $z \in \partial G_j$ and every zero $u$ of $\tilde{p}$ the estimate $|z - u| \geqslant \delta/2$ holds. Hence $|\tilde{p}(z)| \geqslant (\delta/2)^n$ and $|\tilde{q}(z)/\tilde{p}(z)| \leqslant 2^n \, |q|/\delta^n$ for these $z$.

The length of $\partial G_j$ is estimated as follows: For $y \in V(\tilde{p})$ let

$$V_y := \{ z \in \mathbb{C} : |z - v| > |z - y| \text{ for all } v \in V(\tilde{p}) \backslash \{y\} \},$$

the (open) *Voronoi polygon* of $y$ (cf. Preparata and Shamos, 1985, 5.5). It is easy to see that $\overline{G_j} = \bigcup_{y \in V(p_j)} \overline{V_y \cap D}$. Hence the length of the boundary of $G_j$ is at most $\operatorname{len}(\partial G_j) \leqslant \sum_{y \in V(p_j)} \operatorname{len}(\partial(V_y \cap D))$. As $V_y$ is a convex polygon, $V_y \cap D$ is a convex subset of $D$. Hence $\operatorname{len}(\partial(V_y \cap D)) \leqslant 2\pi$ and thus $\operatorname{len}(\partial G_j) \leqslant 2\pi \cdot \# V(p_j) \leqslant 2\pi n_j$. Finally, $|p_j'| \leqslant n_j \cdot |p_j| < n_j \cdot 2^{n_j}$. These estimates and Lemma 4.11 establish the assertion. ▌

*Proof of Corollary* 3.17. Insert the estimate (4.4) from Lemma 4.12 into Definition 3.16. ▌

## 4.5. *Huffman's Algorithm*

It is convenient for the specification and analysis of the algorithms for modular arithmetic to use the standardized form of Huffman's algorithm introduced in Knuth (1973, p. 404).

Let $n, l \geqslant 2$ and $\mathbf{n} \models_l n$. The *Huffman index* of $\mathbf{n}$ is

$$j_H(\mathbf{n}) := \max\{ j \in [l-1] : n_{j+1} \leqslant n_1 + n_2 \}.$$

If $k = j_H(\mathbf{n})$, then the partition

$$s_H(\mathbf{n}) := (n_3, ..., n_{k+1}, n_1 + n_2, n_{k+2}, ..., n_l) \models_{l-1} n$$

is called the *Huffman successor* of $\mathbf{n}$. Here, $n_1 + n_2$ is placed in the $k$th place in $s_H(\mathbf{n})$.

The following measures for partitions are used: The *depth* $d_j(\mathbf{n})$ and the *path weight* $h_j(\mathbf{n})$ of the $j$th leaf in the Huffman tree are the length of resp. the sum of the weights along the path associated with $n_j$. The leaf itself is not counted. Technically, these numbers are defined recursively: For $l = 1$ let $d_j(\mathbf{n}) = h_j(\mathbf{n}) = 0$. For $l \geqslant 2$ let $k = j_H(\mathbf{n})$ and $\mathbf{n}' = s_H(\mathbf{n})$. Then $d_j(\mathbf{n}) = 1 + d_k(\mathbf{n}')$ and $h_j(\mathbf{n}) = n_1 + n_2 + h_k(\mathbf{n}')$ for $j = 1, 2$, $d_j(\mathbf{n}) = d_{j-2}(\mathbf{n}')$ and $h_j(\mathbf{n}) = h_{j-2}(\mathbf{n}')$ for $3 \leqslant j \leqslant k+1$, and $d_j(\mathbf{n}) = d_{j-1}(\mathbf{n}')$ and $h_j(\mathbf{n}) = h_{j-1}(\mathbf{n}')$ for $k+2 \leqslant j \leqslant l$. The *average weighted path length* is $H_1(\mathbf{n}) = \sum_{j=1}^{l} n_j d_j$. It can be defined recursively by $H_1(\mathbf{n}) = 0$ for $l = 1$ and $H_1(\mathbf{n}) = n_1 + n_2 + H_1(\mathbf{n}')$. Finally, the *maximal path weight* is $H_\infty(\mathbf{n}) = \max_{j=1}^{l} h_j(\mathbf{n})$. These measures satisfy the relations

$$d(\mathbf{n}) := d_1(\mathbf{n}) = d_2(\mathbf{n}) = 1 + d_k(\mathbf{n}') \leqslant l - 1. \tag{4.5}$$

$$H_1(\mathbf{n}) < n \cdot (H(\mathbf{n}) + 1). \tag{4.6}$$

$$H_\infty(\mathbf{n}) < 2.6181 \cdot n. \tag{4.7}$$

Relation (4.5) is an immediate consequence of the construction. The estimate (4.6) of $H_1(\mathbf{n})$ is well known (see, e.g., Aigner, 1988, Theorem 1.6). It is sharp in the sense that $H_1(\mathbf{n}) \geqslant n \cdot H(\mathbf{n})$. Estimate (4.7) is Lemma 2 from Kirrinnis (1994).

## 5. MODULAR POLYNOMIAL ARITHMETIC

This section describes the algorithms for computing the product of a sequence of polynomials, the sum of a sequence of rational functions, and the residues of a polynomial with respect to several moduli. The analysis of the algorithms yields proofs for Theorems 3.7, 3.8, and 3.9. For the first algorithm, the correctness proof and time analysis are explained in detail. Similar arguments can be spelled out for the other algorithms in a straightforward manner. This is done in Appendix A.3.

For convenience, it is assumed w.l.o.g. that polynomials $p_j$ are given such that $\deg p_1 \leqslant \deg p_2 \leqslant \cdots \leqslant \deg p_n$. For the technical description of the algorithms the following notation is used: Let $\mathbf{n} \models_l n$. If $l \geqslant 2$, let $k = j_H(\mathbf{n})$, $\mathbf{n}' = s_H(\mathbf{n})$, and $n_{1,2} := n_1 + n_2$. With the notation introduced in Subsection 4.5, let $d_{1,2}(\mathbf{n}) := d_1(\mathbf{n}) - 1 = d_2(\mathbf{n}) - 1$. Then $d_{1,2}(\mathbf{n}) = d_k(\mathbf{n}')$ because of (4.5). As the algorithms work recursively, *approximations* $\tilde{p}_j$, $\tilde{q}_j$, and $\tilde{f}$ for the "real" data are assumed to be given as inputs.

The product of $l$ polynomials $p_j$ $(j \in [l])$ is computed by first multiplying two polynomials of least degrees, i.e., $p_1$ and $p_2$. The product $p_1 p_2$ is merged into the sequence $p_3, ..., p_l$ after all polynomials with the same degree, and then the product of these $l-1$ polynomials is computed recursively. This corresponds to the standard Huffman tree described in Subsection 4.5.

5.1. Algorithm (Multiplication of Many Polynomials).

**Input:** An ordered partition $\mathbf{n} \models_l n$; an accuracy parameter $s \in \mathbb{N}$; polynomials $\tilde{p}_j \in \Pi_{n_j}$ with $|p_j - \tilde{p}_j| < 2^{-(s + (n - n_j) + 2d_j(\mathbf{n}))}$ $(j \in [l])$.

**Returns:** A polynomial $\tilde{p} \in \Pi_n$ with $|p_1 \cdots p_l - \tilde{p}| < 2^{-s}$.

**Time bound:** $O(\psi(H_1(\mathbf{n}) \cdot (s+n)))$.

```
1. If $l = 1$: {Let $\tilde{p} := p_1$; return.}
2. Compute $\tilde{p}_{1,2} \in \Pi_{n_{1,2}}$ with $|\tilde{p}_1\tilde{p}_2 - \tilde{p}_{1,2}| < 2^{-(s + (n - n_{1,2}) + 2d_1(\mathbf{n}))}$.
3. Compute $\tilde{p}$ by calling Algorithm 5.1 recursively
   with input $(\mathbf{n}'; s; \tilde{p}_3, ..., \tilde{p}_{k+1}, \tilde{p}_{1,2}, \tilde{p}_{k+2}, ..., \tilde{p}_l)$.
```

*Correctness Proof and Time Analysis.* Calling the algorithm recursively in Step 3 is legal, because

$$\begin{aligned}
&|p_1 p_2 - \tilde{p}_{1,2}| \\
&\quad < 2^{-(s+(n-n_{1,2})+2d_1(\mathbf{n}))} + |p_1| \cdot |p_2 - \tilde{p}_2| + |\tilde{p}_2| \cdot |p_1 - \tilde{p}_2| \\
&\quad < 2^{-(s+(n-n_{1,2})+2d_1(\mathbf{n}))} + 2^{n_1} \cdot 2^{-(s+(n-n_2)+2d_2(\mathbf{n}))} \\
&\qquad + (2^{n_2} + 2^{-s}) \cdot 2^{-(s+(n-n_1)+2d_1(\mathbf{n}))} \\
&\quad < 2^{-(s+(n-n_{1,2})+2d_{1,2}(\mathbf{n}))}.
\end{aligned}$$

According to Theorem 3.2, there is a constant $c > 0$ such that for any two polynomials $F, G \in \Pi_\nu$ with $|F|, |G| < 2^{\nu+1}$, a polynomial $P \in \Pi_{2\nu}$ with $|P - FG| < 2^{-N}$ can be computed in time less than $c \cdot \psi(\nu \cdot (N + \nu))$. Now it is proved by induction that for $l \geqslant 2$ the running time of Algorithm 5.1 is bounded by $c \cdot \psi(H_1(\mathbf{n}) \cdot (s + n + 2d(\mathbf{n})))$: This is obvious for $l = 2$. Because of $|\tilde{p}_j| < |p_j| + 2^{-s} < 2^{n_j+1}$ and the definition of $c$, the computing time for Step 2 is bounded by $c \cdot \psi(n_{1,2} \cdot (s + (n - n_{1,2}) + 2d_1(\mathbf{n})) + n_{1,2}) = c \cdot \psi(n_{1,2} \cdot (s + n + 2d(\mathbf{n})))$. By induction, the computing time for Step 3 is bounded by $c \cdot \psi(H_1(\mathbf{n}') \cdot (s + n + 2d(\mathbf{n}')))$. Hence the time bound for the algorithm follows from $H_1(\mathbf{n}) = n_{1,2} + H_1(\mathbf{n}')$ and $d(\mathbf{n}') \leqslant d(\mathbf{n})$. Now the time bound stated in Theorem 3.7 follows from (4.5) and (4.6). ▮

The sum $q/p = \sum_{j=1}^{l} q_j/p_j$ with $\deg q_j < \deg p_j$ is computed by first computing $q_{1,2}/p_{1,2} := q_1/p_1 + q_2/p_2$ and then adding $l-1$ rational functions recursively. For convenience, $|q_j| \leqslant n_j \cdot 2^{n_j} \cdot M$ is assumed instead of $|q_j| \leqslant M$.

5.2. Algorithm (Addition of Rational Functions).

**Input:** An ordered partition $\mathbf{n} \models_l n$; an accuracy parameter $s \in \mathbb{N}$; a real bound $M \geqslant 1$; polynomials $\tilde{p}_j \in \Pi_{n_j}$ and $\tilde{q}_j \in \Pi_{n_j - 1}$ $(j \in [l])$ with

$$\begin{aligned}
|p_j - \tilde{p}_j| &< 2^{-(s+(n-n_j)+3d_j(\mathbf{n}))}/(lM), \\
|q_j - \tilde{q}_j| &< 2^{-(s+(n-n_j)+3d_j(\mathbf{n}))}.
\end{aligned}$$

**Returns:** Polynomials $\tilde{p} \in \Pi_n$ with $|p_1 \cdots p_l - \tilde{p}| < 2^{-s}$ and $\tilde{q} \in \Pi_{n-1}$ with $|q - \tilde{q}| < 2^{-s}$.

**Time bound:** $O(\psi(H_1(\mathbf{n}) \cdot (s + n + \log M)))$.

1. If $l = 1$: $\{\tilde{p} := p_1;\ \tilde{q} := q_1;$ return.$\}$.
2. Compute $\tilde{p}_{1,2} \in \Pi_{n_{1,2}}$ and $\tilde{q}_{1,2} \in \Pi_{n_{1,2}-1}$ with

$$\begin{aligned}
|\tilde{p}_1 \tilde{p}_2 - \tilde{p}_{1,2}| &< 2^{-(s+(n-n_{1,2})+3d_1(\mathbf{n}))}/(nM), \\
|\tilde{q}_1 \tilde{p}_2 + \tilde{q}_2 \tilde{p}_1 - \tilde{q}_{1,2}| &< \eta := 2^{-(s+(n-n_{1,2})+3d_1(\mathbf{n}))}.
\end{aligned}$$

```
3. Compute a polynomial $\tilde q$ with $|q-\tilde q| < 2^{-s}$
   by calling Algorithm 5.2 recursively with input
   $(\mathbf{n}'; s; M; \tilde p_3, ..., \tilde p_k, \tilde p_{1,2}, \tilde p_{k+1}, ..., \tilde p_l; \tilde q_3, ..., \tilde q_k, \tilde q_{1,2}, \tilde q_{k+1}, ..., \tilde q_l)$.
```

The residues $f_j := f \bmod p_j$ for $j \in [l]$ are computed by first computing the residues $f_{1,2} := f \bmod (p_1 p_2), f_3, ..., f_l$ recursively and then computing $f_1$ and $f_2$ from $f_{1,2}$ by polynomial division. Let $h_{1,2}(\mathbf{n}) := h_1(\mathbf{n}) - (n_1 + n_2)$. It is assumed for simplicity that $\varrho(\tilde p_j) \leqslant 1$ for all approximations $\tilde p_j \in \Pi_{n_j}$ for $p_j$. This can be assumed w.l.o.g., because scaling with a factor of 2 amplifies the error by at most $2^n$. The specification of the following algorithm implies the time bound asserted in Theorem 3.9, if $\sigma \geqslant s + n + 2H_\infty(\mathbf{n})$ is chosen:

5.3. Algorithm (Modular Representation).

**Input:** An ordered partition $\mathbf{n} \models_l n$; an accuracy parameter $\sigma \geqslant m$; polynomials $f \in \Pi_m$ with $|f| \leqslant 1$ and $\tilde p_j \in \Pi^1_{n_j}$ with $|p_j - \tilde p_j| < \frac{4}{9} \cdot 2^{n_j - 2n - 2d_j(\mathbf{n})} \cdot 2^{-(\sigma+m)}$.

**Returns:** Polynomials $\tilde f_j \in \Pi_{n_j - 1}$ with $|f_j - \tilde f_j| < 2^{2h_j(\mathbf{n}) + n_j} \cdot 2^{-\sigma}$.

**Time bound:** $O(\psi((m + H_1(\mathbf{n})) \cdot \sigma))$

```
1. If $l = 1$:
   {Compute $\tilde f_1 \in \Pi_{n-1}$ with $|f \bmod \tilde p_1 - \tilde f_1| < 2^{-\sigma+n}$; return.}
2. Compute $\tilde p_{1,2}$ such that
   $|\tilde p_1 \tilde p_2 - \tilde p_{1,2}| < \frac{2}{9} \cdot 2^{n_{1,2} - 2n - d_{1,2}(\mathbf{n})} \cdot 2^{-(\sigma+m)}$.
3. Call Algorithm 5.3 recursively to compute polynomials
   $\tilde f_{1,2}, \tilde f_3, ..., \tilde f_l$ with $|f_j - \tilde f_j| < 2^{2h_j(\mathbf{n}) + n_j} \cdot 2^{-\sigma}$ for $j = 3, ..., l$
   and $|f_{1,2} - \tilde f_{1,2}| < 2^{2h_{1,2}(\mathbf{n}) + n_{1,2}} \cdot 2^{-\sigma}$.
4. For $j = 1, 2$ use polynomial division to compute residues
   $\tilde f_j \in \Pi_{n_j - 1}$ with $|\tilde f_{1,2} \bmod \tilde p_j - \tilde f_j| < \frac{1}{8} \cdot 2^{-(\sigma+m)} \cdot |\tilde f_{1,2}|$.
```

## 6. THE EXTENDED SPLITTING CIRCLE METHOD

### 6.1. *Unit Circle Splitting*

This section summarizes some elements of Schönhage's splitting circle method (1982b, for brevity referred to as [S] in the following) and provides extensions for PFD. The first result deals with the computation of incomplete PFDs in the special case that the roots of the denominator are sufficiently far away from the unit circle.

6.1. Definition. A polynomial $P \in \Pi_n$ is called $(k, \delta, \mu)$-*E-factorable* ("factorable with respect to the unit circle $E = \partial D$") if $|P| = 1$, $1 \leqslant k \leqslant n/2$, and $0 < \delta < \frac{1}{2} \ln 3$ are such that $P$ has $k$ zeros of modulus $\leqslant e^{-\delta}$ and $n-k$ zeros of modulus $\geqslant e^{\delta}$, and if finally $\mu = \min\{|P(t)| : |t| = 1\}$.

For the rest of Subsection 6.1, it is assumed that $P$ is $(k, \delta, \mu)$-$E$-factorable and that $Q \in \Pi_{n-1}$ with $|Q| = 1$. Note that $|P(t)| \leqslant |P|$ for $|t| = 1$ and hence $\mu \leqslant 1$.

6.2. Definition. Let $\varepsilon > 0$. An $\varepsilon$-*E-splitting* of $P$ is given by polynomials $F \in \Pi_k^1$ with $\varrho(F) < 1$, $G \in \Pi_{n-k}$ with $\varrho^*(G) > 1$, and $\operatorname{lc}(G) = \operatorname{lc}(P)$ such that $|P - FG| < \varepsilon$.

Let in addition $\eta > 0$. An $(\varepsilon, \eta)$-$E$-splitting of $Q/P$ is given by an $\varepsilon$-$E$-splitting $(F, G)$ of $P$ and by polynomials $U \in \Pi_{k-1}$ and $V \in \Pi_{n-k-1}$ with $|Q - VF - UG| < \eta$.

6.3. Lemma. *Let $(F, G)$ be an $\varepsilon$-splitting of $P$ with $\varepsilon \leqslant \mu/8$, and let $K, p, q \in \Pi$ with $K = pG + qF$. Then*

$$|FG| < 1 + \varepsilon, \qquad |F| < 2^k, \qquad \textit{and} \qquad |G| < \tfrac{9}{8} \cdot 2^{n-k}. \tag{6.1}$$

$$\textit{If } \deg p < \deg F, \qquad \textit{then} \quad |p| \leqslant \tfrac{8}{7} \cdot (k/\mu) \cdot |F| \cdot |K|. \tag{6.2}$$

$$\textit{If } \deg q^* < \deg G^*, \qquad \textit{then} \quad |q| \leqslant \tfrac{8}{7} \cdot ((n-k)/\mu) \cdot |G| \cdot |K|. \tag{6.3}$$

*Proof.* Condition (6.1) summarizes (10.1), (10.2), and (10.3) from [S]. Condition (6.2) follows from Lemma 4.11, applied to $(f, g, u, v) := (F, G, p, q)$ and the unit disk, when $|F(t)\, G(t)| \geqslant \mu - \varepsilon \geqslant \frac{7}{8} \cdot \mu$ and $|K(t)| \leqslant |K|$ for $|t| = 1$ are taken into account. For the proof of (6.3), use the equation $K^* = p^*G^* + q^*F^*$ and apply Lemma 4.11 analogously to $(f, g, u, v) := (G^*, F^*, q^*, p^*)$. The estimate follows with $|K^*(t)| = |K(1/t)| \leqslant |K|$ and $|F^*(t) G^*(t)| = |F(1/t)\, G(1/t)| \geqslant \mu - \varepsilon$ for $|t| = 1$. ∎

6.4. Theorem (Unit Circle Splitting). *An $(\varepsilon, \eta)$-$E$-splitting of $Q/P$ can be computed in time*

$$O\left(\psi\left(\left(k + \frac{1}{\delta}\right) \cdot \left(n + \log\left(\frac{1}{\mu}\right)\right)^2 + n \cdot \left(\log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{1}{\eta}\right)\right)\right)\right).$$

*Moreover, $|U| \leqslant \frac{9}{7} \cdot k \cdot 2^k/\mu$ and $|V| \leqslant \frac{81}{56} \cdot (n-k) \cdot 2^{n-k}/\mu$.*

*Proof.* Theorem 12.1 of [S] asserts that an $\varepsilon$-$E$-splitting $(F, G)$ of $P$ can be computed within the asserted time bound. The corresponding algorithm computes an auxiliary polynomial $H$ such that $HG \equiv 1 \bmod F$ up to a small error. $H$ can be used to compute $U$ and $V$, using the relations $QH \equiv VHF + UHG \equiv U \bmod F$ and $V = (Q - UG)/F$, which are fulfilled up to small errors. See Appendix A.4 for details.

## 6.2. *How to Find Splitting Circles*

Splitting circles are computed in two steps. First, the center $w$ is chosen such that the distances of the roots of $p$ from this point are widespread, i.e., $\varrho(T_w p)/\varrho^*(T_w p)$ is sufficiently large. The following lemma summarizes the results of [**S**, Sect. 17]. The time bound follows from that for root radius calculation in Lemma 3.10.

6.5. LEMMA. *Let $p \in \Pi_n$ with $\bar{z}(p) = 0$ and $0.98 < \varrho(p) < 1$. Let $w \in \{2, 2i, -2, -2i\}$ be chosen such that $\Delta := \ln(\varrho(T_w p)/\varrho^*(T_w p))$ is maximal. Then $\Delta \geqslant 0.3$. It can be determined which point $w$ to take in time $O(\psi(n^2 \cdot \log\log n))$.*

If a sufficiently large spread of the root moduli of $p$ is guaranteed, a suitable radius $r$ for the splitting circle, which is centered at the origin, can be computed by variants of Graeffe's method. In the following lemma, the results of [**S**, Sect. 16] are summarized. Some details which cannot be seen immediately from [**S**] are explained in Appendix A.5.

6.6. LEMMA. *Let $p \in \Pi_n$ and $r', r'' > 0$ be given such that $\varrho^*(p) \leqslant r' < r'' \leqslant \varrho(p)$, and let $\Delta := \ln(r''/r')$. Then for $\beta < \Delta/4$, an index $k \leqslant n/2$ and a radius $r$ with $r' < r < r''$ can be computed such that at least one of the polynomials $P := S_r p/|S_r p|$ or $P^*$ is $(k, \delta, \mu)$-E-factorable with parameters $\delta$ and $\mu$ which fulfil*

$$\mu > 2^{-Cn}, \qquad \frac{1}{\delta} \leqslant \frac{2k \cdot \lfloor \log n \rfloor}{0.98 \cdot \beta}, \qquad \textit{and} \qquad \frac{1}{\delta} \leqslant \frac{n}{0.98 \cdot \beta}$$

*with a constant factor $C = C(\beta)$ independent of $n$ and $p$. The computation of $r$ and $k$ is possible in time $O(\psi(kn^2 \log n))$.*

## 6.3. *Single Splittings*

For the rest of Section 6, let $p \in \Pi_n^1$ and $q \in \Pi_{n-1}$ with $|q| = 1$. Furthermore, let $\varepsilon = 2^{-s_0}$, $\eta = 2^{-s_1} \leqslant 2^{-2n}$, and $\vartheta = 2^{-\gamma} \leqslant 1$. Assume that $\varrho(p) \leqslant 1 - \vartheta/(14n)$ holds. Finally, let

$$\vartheta_{n,l} := \frac{\vartheta}{7n} \cdot \left(1 - \frac{l}{2n}\right), \qquad \varepsilon_{n,l} := \varepsilon \cdot \frac{l}{n}, \qquad \text{and} \qquad \eta_{n,l} := \eta \cdot \frac{l}{n}.$$

The splitting circle method computes a radius $\vartheta$ decomposition $q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, \eta)$ by repeated computation of "single splittings" which are specified as follows:

6.7. Definition. Assume that a $\vartheta_{n,l}$-separated PFD

$$q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon_{n,l}, \eta_{n,l}) \tag{6.4}$$

with $v := n_1 = \deg p_1 \geqslant 2$ and a standard $\vartheta$-approximation $\varrho > \vartheta$ for $\bar{\varrho}(p_1)$ are given. A *single splitting for* $q_1/p_1$ is given by an index $k \leqslant v/2$, polynomials $f \in \Pi_k^1$, $g \in \Pi_{v-k}^1$, $u \in \Pi_{k-1}$, and $v \in \Pi_{v-k-1}$ such that

$$q/p \approx u/f + v/g + q_2/p_2 + \cdots + q_l/p_l \operatorname{err}(\varepsilon_{n,l+1}, \eta_{n,l+1}) \tag{6.5}$$

is a $\vartheta_{n,l+1}$-separated PFD, and standard $\vartheta$-approximations $\varrho'$ and $\varrho''$ for $\bar{\varrho}(f)$ and $\bar{\varrho}(g)$.

Subsections 6.4–6.7 describe an algorithm for computing single splittings. Its analysis results in the following time bound:

6.8. Lemma. *There are constants* $c, c_1, c_2 > 0$ *such that whenever* $l$, $q_j$, $p_j$, $\varrho$, *and* $v$ *are given as in Definition* 6.7, *a single splitting* $(k, f, g, u, v, \varrho', \varrho'')$ *of* $q_1/p_1$ *can be computed in time* $c_1 \cdot \psi(k \cdot v^2 \cdot \log v) + c_2 \cdot \psi(v \cdot s)$, *where* $s = \max\{s_0, s_1\} + n\gamma + \lceil 2n \log n \rceil + cn$.

## 6.4. *Computation of a Splitting Circle*

The splitting strategy and transforms from [**S**, Sect. 7] are used with adaptations for transforming the numerators. For simplicity, rounding errors in the transforms are neglected. A straightforward, but tedious analysis shows that it is sufficient to perform the following scalings and Taylor shifts within error bounds which are of the same order of magnitude as those for the splitting itself. Hence these transforms can be performed within the same time bounds.

Let $y := \bar{z}(p_1)$ and

$$P_1 := T_y p_1, \qquad Q_1 := T_y q_1, \qquad P_2 := \varrho^{-v} S_\varrho P_1, \qquad Q_2 := S_\varrho Q_1.$$

Then $\bar{z}(P_2) = 0$ and $0.98 < \varrho(P_2) < 1$. The additional factor $\varrho^{-v}$ in the definition of $P_2$ compensates the norm of the scaling operator $S_{1/\varrho}$ in the backward transform. The coefficients of $P_2$ are not too large, because the root radius of $P_1$ is bounded by $\varrho$ (see [**S**, Sect. 7] for details).

The next step is to compute the center of the splitting circle, i.e., a point $w \in \{2, 2i, -2, -2i\}$ with $\Delta := \ln(\varrho(T_w P_2)/\varrho^*(T_w P_2)) \geqslant 0.3$. According to Lemma 6.5, $w$ can be computed in time $O(\psi(v^2 \log\log v))$. Now let

$$P_3 := T_w P_2, \qquad Q_3 := T_w Q_2.$$

Now a radius $r \in (1, 3)$ for the splitting circle and a corresponding index $k \leqslant v/2$ are computed according to Lemma 6.6, where $\beta = \frac{3}{41}$ is chosen (hence in particular $\beta < \Delta/4$). The radius $r$ and the index $k$ can be computed in time $O(\psi(kv^2 \log v))$. Now let

$$P_4 := S_r P_3, \qquad Q_4 := S_r Q_3, \qquad P := P_4/|P_4|, \qquad \text{and} \qquad Q := Q_4/|Q_4|$$

and replace $(P, Q)$ by $(P^*, Q^*)$, if $k > v/2$. Then Lemma 6.6 implies that $P$ is $(k, \delta, \mu)$-$E$-factorable with $\log(1/\mu) = O(v)$ and $1/\delta \leqslant \min\{14v, 7k \lfloor \log v \rfloor\}$.

## 6.5. *Splitting and Backward Transform*

Let $K$ denote the splitting circle in the initial coordinates, $K = D_{r \cdot \varrho}(y + \varrho w)$. Then there is a unique *exact* PFD $q_1/p_1 = u_0/f_0 + v_0/g_0$, where $f_0 \in \Pi^1_k$, $g_0 \in \Pi^1_{v-k}$, $u_0 \in \Pi_{k-1}$, $v_0 \in \Pi_{v-k-1}$, $V(f_0) \subset K$, and $V(g_0) \subset \mathbb{C} \setminus \bar{K}$. According to Lemma 6.6, the construction of $K$ guarantees

$$\operatorname{dist}(V(f_0), \partial K) \geqslant r \cdot \varrho \cdot (1 - e^{-\delta})$$

and

$$\operatorname{dist}(V(g_0), \partial K) \geqslant r \cdot \varrho \cdot (e^{\delta} - 1).$$

With $r > 1$, $\varrho \geqslant \vartheta$, and $\delta \geqslant 1/(14v)$, this implies

$$\operatorname{dist}(V(f_0), V(g_0)) \geqslant r \cdot \varrho \cdot (e^{\delta} + e^{-\delta}) > \vartheta \cdot 2\delta \geqslant \vartheta/(7v).$$

Hence the APFD

$$q/p \approx u_0/f_0 + v_0/g_0 + q_2/p_2 + \cdots + q_l/p_l \operatorname{err}(\varepsilon_{n,l}, \eta_{n,l}) \tag{6.6}$$

is $\vartheta_{n,l}$-separated.

Now an APFD corresponding to the exact PFD $q_1/p_1 = u_0/f_0 + v_0/g_0$ is *computed*: For suitable $\varepsilon', \eta' > 0$, the choice of which is discussed below, an $(\varepsilon', \eta')$-$E$-splitting $(F, G, U, V)$ of $Q/P$ is computed in time

$$O(\psi(k \cdot v^2 \cdot \log v + v \cdot (\log(1/\varepsilon') + \log(1/\eta')))).$$

Then the backward transform is computed. The corresponding operator is $\Omega := T_{-y}S_{1/\varrho}T_{-w}S_{1/r}$. The factors are chosen such that $\mathrm{lc}(f) = \mathrm{lc}(g) = 1$:

$$
\begin{aligned}
F_4 &:= r^k \cdot F, & G_4 &:= r^{-k} \cdot |P_4| \cdot G, \\
U_4 &:= \frac{|Q_4| \cdot r^k}{|P_4|} \cdot U, & V_4 &:= |Q_4| \cdot r^{-k} \cdot V, \\
f &:= \varrho^k \cdot \Omega F_4, & g &:= \varrho^{\nu-k} \cdot \Omega G_4, \\
u &:= \varrho^{k-\nu} \cdot \Omega U_4, & v &:= \varrho^{-k} \cdot \Omega V_4.
\end{aligned}
$$

## 6.6. *Error Propagation and Accuracy Requirements*

The translations and the scalings with the moderate size factors $r$ resp. $1/r$ cause only harmless error amplification factors $2^{O(\nu)}$. For the scaling operator $S_{1/\varrho}$, an additional factor $\beta_{\nu-1}(S_{1/\varrho}) = \varrho^{1-\nu}$ must be accounted in the estimate of $|q_1 - vf - ug|$. As has been illustrated with example (1.1), the norms of $u$ and $v$ can be thus large. So this "blow up effect" must be accepted. It is controlled by bounding $\varrho$ from below. In the estimate for $|p_1 - fg|$, the norm of $S_{1/\varrho}$ is compensated by the additional factor $\varrho^{-\nu}$.

The estimates $|P - FG| < \varepsilon'$ and $|Q - VF - UG| < \eta'$ imply

$$|p_1 - fg| \leqslant \varrho^\nu \cdot \beta_\nu(\Omega) \cdot |P_4| \cdot \varepsilon'$$

and

$$|q_1 - vf - ug| \leqslant \beta_{\nu-1}(\Omega) \cdot |Q_4| \cdot \eta'.$$

Because of $|y| < 1$, $|w| = 2$, and $1 < r < 3$, the estimate

$$\beta_m(\Omega) \leqslant (6/\varrho)^m$$

holds for all $m \in \mathbb{N}$. Moreover,

$$|P_4| \leqslant (|w| + r + 1)^\nu \cdot |P_1| \leqslant 6^\nu \cdot |P_1| \leqslant 12^\nu \cdot |p_1|$$

holds because of $P_4(z) = \varrho^{-\nu} P_1(\varrho(rz + w))$, compare [**S**, Sect. 7]. Instead of the naive estimate $|Q_4| \leqslant |S_r| \cdot |T_w| \cdot |S_\varrho| \cdot |T_y| \cdot |q_1| \leqslant 18^{\nu-1} \cdot |q_1|$, the better bound

$$|Q_4| < (|w| + r)^{\nu-1} \cdot |Q_2| < 5^{\nu-1} \cdot (1 + |y|)^{\nu-1} \cdot |q_1| < 10^{\nu-1} \cdot |q_1|$$

is used, which can be derived like the estimate of $|P_4|$. Altogether:

$$|p_1 - fg| \leqslant 72^{\nu} \cdot \varepsilon' \cdot |p_1| =: \varepsilon'' \cdot |p_1|, \tag{6.7}$$

$$|q_1 - vf - ug| \leqslant (60/\varrho)^{\nu-1} \cdot \eta' \cdot |q_1| =: \eta'' \cdot |q_1|. \tag{6.8}$$

Now it is investigated how to choose $\varepsilon'$ and $\eta'$ to guarantee (6.5) and to ensure that this APFD is $\vartheta_{n,\,l+1}$-separated. The error of the denominator of the new APFD is estimated as

$$|p - fgp_2 \cdots p_l| \leqslant \varepsilon_{n,\,l} + \varepsilon'' \cdot |p_1| \cdot |p_2 \cdots p_l| \leqslant \varepsilon_{n,\,l} + 2^n \cdot \varepsilon''. \tag{6.9}$$

For the estimate of the numerator let

$$\varphi := q_2 p_3 \cdots p_l + \cdots + q_l p_2 \cdots p_{l-1}.$$

The given APFD (6.4) is $\vartheta_{n,\,l}$-separated. Hence

$$\begin{aligned} |q_j| &\leqslant n_j^2 \cdot (1 + \eta_{n,\,l}) \cdot 2^n \cdot \vartheta_{n,\,l}^{-n} \cdot |p_j| \\ &\leqslant n_j^2 \cdot (1 + \eta) \cdot (28n)^n \cdot \vartheta^{-n} \cdot |p_j| \end{aligned}$$

for $j \in [l]$ because of Lemma 4.12. Therefore

$$|\varphi| \leqslant (n_2^2 + \cdots + n_l^2) \cdot (1 + \eta) \cdot (28n)^n \cdot \vartheta^{-n} \cdot |p_2| \cdots |p_l|.$$

This yields

$$\begin{aligned} &|q - (vf + ug) \cdot p_2 \cdots p_l - fg \cdot \varphi| \\ &\quad \leqslant |q_1 - vf - ug| \cdot |p_2 \cdots p_l| + |p_1 - fg| \cdot |\varphi| + \eta_{n,\,l} \\ &\quad \leqslant \eta'' \cdot |q_1| \cdot |p_2 \cdots p_l| + \varepsilon'' \cdot |p_1| \cdot |\varphi| \;\; + \eta_{n,\,l} \\ &\quad \leqslant (1 + \eta) \cdot (56n)^n \cdot \vartheta^{-n} \cdot (\eta'' n_1^2 + \varepsilon'' \cdot (n_2^2 + \cdots + n_l^2)) + \eta_{n,\,l}. \end{aligned} \tag{6.10}$$

Because of (6.9) and (6.10), the conditions

$$\varepsilon'' \leqslant \varepsilon \cdot 2^{-n}/n \tag{6.11}$$

and

$$\varepsilon'', \eta'' \leqslant \left(\frac{\vartheta}{56n}\right)^n \cdot \frac{\eta}{1 + \eta} \cdot \frac{1}{n^3} \tag{6.12}$$

are sufficient for (6.5). Now the root perturbation is investigated: With $\eta \leqslant 2^{-2n}$ and $n \geqslant 2$, (6.12) implies $\varepsilon'' \cdot |p_1| \leqslant (224)^{-n} =: \sigma$. Therefore (with (6.7)), $|p_1 - fg| \leqslant \sigma$, and thus $\Delta(fg, p_1) < 2.01 \cdot \sqrt[\nu]{\varepsilon''}$ because of Lemma 4.9. Hence, if in addition

$$\varepsilon'' \leqslant \left(\frac{\vartheta}{56.3 n^2}\right)^{\nu}, \tag{6.13}$$

then $\Delta(fg, p) < \vartheta/(28n^2)$. The computed APFD (6.5) is $\vartheta_{n,l+1}$-separated, because the corresponding exact PFD (6.6) is $\vartheta_{n,l}$-separated.

## 6.7. *Time Bound for a Single Splitting*

If (6.7) and (6.8) are simplified by inserting $\nu < n$ and $\varrho > \vartheta$, then it follows from (6.11), (6.12), and (6.13) that it is sufficient to choose

$$\varepsilon' \leqslant \min\left\{\left(\frac{\vartheta}{4032 \cdot n}\right)^{n} \cdot \frac{\eta}{2n^3}, \left(\frac{\vartheta}{4054 \cdot n^2}\right)^{\nu}, (144)^{-n} \cdot \frac{\varepsilon}{n}\right\},$$

$$\eta' \leqslant \left(\frac{\vartheta}{3360 \cdot n^2}\right)^{n} \cdot \frac{\eta}{2n^3}.$$

Let $s = \max\{s_0, s_1\} + n\gamma + \lceil 2n \log n \rceil + cn$, where $c$ is chosen such that $\varepsilon' := 2^{-s}$ and $\eta' := 2^{-s}$ fulfil the above estimates. The time for computing $F$, $G$, $U$, and $V$ is at most $O(\psi(kn^2 \log n + ns))$. This follows from Theorem 6.4 and the estimates $\log(1/\mu) = O(n)$ and $1/\delta = O(k \log n)$. It is sufficient to compute the scalings and translations with accuracy parameters of the same order of magnitude as for the splitting. Therefore, the same time bound holds for the transforms. It has been shown above that computing the splitting circle is possible within the same time bound. Finally, it follows from Lemma 3.12 that standard $\vartheta$-approximations $\varrho'$ for $\bar{\varrho}(f)$ and $\varrho''$ for $\bar{\varrho}(g)$ can be computed in time $O(\psi(\nu^2 \log\log \nu + \nu^2\gamma)) = O(\psi(n^2 \log n + ns))$. This completes the proof of Lemma 6.8.

## 6.8. *Radius $\vartheta$ Decomposition by Repeated Splitting*

The time estimate follows the lines of [**S**, Sect. 5]. The case $\nu = n$ of the following lemma yields the assertion of Theorem 3.15.

6.9. Lemma. *Let $p$, $q$, $\varepsilon$, $\eta$, $\vartheta$, $\varepsilon_{n,l}$, $\eta_{n,l}$, $\vartheta_{n,l}$, $l$, $p_j$, $q_j$, and $\nu$ be given as in Definition 6.7, and let a standard $\vartheta$-approximation $\varrho$ for $\bar{\varrho}(p_1)$ be given. From these data, a $\vartheta_{n,l+m-1}$-separated APFD*

$$q/p \approx u_1/f_1 + \cdots + u_m/f_m + q_2/p_2 + \cdots + q_l/p_l \operatorname{err}(\varepsilon_{n,l+m-1}, \eta_{n,l+m-1})$$

*with $\bar{\varrho}(f_j) < \vartheta$ for all $j \in [m]$ can be computed in time*

$$T_1(v) := \tfrac{2}{3}c_1 \cdot \psi(v^3 \log v) + c_2 \cdot \psi(v^2 \cdot s),$$

*where $s$, $c_1$, and $c_2$ are as in Lemma* 6.8.

*Proof.* The lemma is proved by induction in $v$. Let $\mathbf{Z}(v)$ denote the assertion of the lemma. The case $v=1$ is void. If $v \geqslant 2$ and $\varrho > \vartheta$, then computing a single splitting produces an index $k$ with $1 \leqslant k \leqslant v/2$, polynomials $f \in \Pi_k^1$, $g \in \Pi_{v-k}^1$, $u \in \Pi_{k-1}$, and $v \in \Pi_{v-k-1}$ such that

$$q/p \approx u/f + v/g + q_2/p_2 + \cdots + q_l/p_l \operatorname{err}(\varepsilon_{n,\,l+1}, \eta_{n,\,l+1})$$

is a $\vartheta_{n,\,l+1}$-separated APFD, and standard $\vartheta$-approximations $\varrho'$ and $\varrho''$ for $\bar{\varrho}(f)$ and $\bar{\varrho}(g)$. According to Lemma 6.8, this splitting can be computed in time $T_0(v,k) := c_1 \cdot \psi(k \cdot v^2 \cdot \log v) + c_2 \cdot \psi(v \cdot s)$. Now a $\vartheta_{n,\,l+m'}$-separated APFD

$$q/p \approx u_1/f_1 + \cdots + u_{m'}/f_{m'} + v/g + q_2/p_2 + \cdots + q_l/p_l \operatorname{err}(\varepsilon_{n,\,l+m'}, \eta_{n,\,l+m'})$$

with $\bar{\varrho}(f_j) < \vartheta$ for $j \in [m']$ can be computed in time at most $T_1(k)$. This follows from $\mathbf{Z}(k)$. From this APFD, the desired one can be computed in time bounded by $T_1(v-k)$ because of $\mathbf{Z}(v-k)$.

Hence the overall effort is bounded by $T_0(v,k) + T_1(k) + T_1(v-k)$. Let $k_* := n-k \geqslant k$. Then $kn + k^2 + k_*^2 \leqslant n^2$ and $\frac{3}{2}kn^2 + k^3 + k_*^3 \leqslant n^3$. All together,

$$\begin{aligned} &T_0(v,k) + T_1(k) + T_1(v-k) \\ &\quad\leqslant c_1\psi(kv^2 \log v) + c_2\psi(vs) + \tfrac{2}{3}c_1\psi(k^3 \log k) \\ &\qquad + c_2\psi(k^2 s) + \tfrac{2}{3}c_1\psi(k_*^3 \log k_*) + c_2\psi(k_*^2 s) \\ &\quad\leqslant \tfrac{2}{3}c_1\psi(v^3 \log v) + c_2\psi(v^2 s) = T_1(v). \quad \blacksquare \end{aligned}$$

## 7. IMPROVEMENT OF PARTIAL FRACTIONS

This section describes the quadratic iteration for simultaneously improving all numerators in a PFD with respect to a fixed factorization of the denominator into pairwise prime factors. Let $n, l \geqslant 2$, $\mathbf{n} = (n_1, ..., n_l) \vdash_l n$, and $H = H(\mathbf{n})$. The algorithm is based on the following error estimate:

7.1. LEMMA. *Let* $p_j \in \Pi^1_{n_j}$, $h_j \in \Pi_{n_j-1}$, *and* $r_j := p_1 \cdots p_{j-1} \cdot p_{j+1} \cdots p_l$ *for* $j \in [l]$. *Furthermore, let* $d := 1 - \sum_{j=1}^{l} h_j r_j$, *and let* $|d| < 1$. *Finally, let* $\hat{h}_j := (h_j(1+d)) \bmod p_j$. *Then*

$$|\hat{h}_j| \leqslant (1 + \tfrac{3}{16} \cdot 2^{n+2n_j} \cdot |d|) \cdot |h_j|,$$

$$\left|1 - \sum_{j=1}^{l} \hat{h}_j r_j\right| < 2^{3n-3} \cdot |d|^2.$$

*Proof.* The estimate of $|\hat{h}_j|$ follows from $h_j - \hat{h}_j \equiv h_j \cdot d \bmod p_j$ and Lemma 4.4. Let $\hat{d} := 1 - \sum_{j=1}^{l} \hat{h}_j r_j$. Then $\hat{d} \equiv 1 - \hat{h}_j r_j \equiv 1 - (1+d)\, h_j r_j \equiv 1 - (1+d)(1-d) \equiv d^2 \bmod p_j$. Because of Lemma 4.8, the $p_j$ are pairwise prime. The Chinese remainder theorem implies $\hat{d} \equiv d^2 \bmod (p_1 \cdots p_l)$. Now the estimate of the new defect $\hat{d}$ follows from Corollary 4.5. ∎

The $\hat{h}_j$ are computed as follows: Compute $d$ with Algorithm 5.2. Then compute $d_j = d \bmod p_j$ with Algorithm 5.3. Finally, compute $\hat{h}_j$ from $\hat{h}_j = (h_j \cdot (1 + d_j)) \bmod p_j$, using Algorithm 5.3 again. These ideas are the basis of the following algorithm for a single iteration step:

7.2. ALGORITHM (Partial Fraction Decomposition, Single Step).

**Input:** Polynomials $p_j \in \Pi^1_{n_j}$ and $h_j \in \Pi_{n_j-1}$ and real bounds $M \geqslant 1$ and $\eta \in (0, \frac{1}{2})$ such that, with $r_j$ and $d$ as in Lemma 7.1, the estimates $|h_j| \leqslant M$ and $|d| < \eta$ hold.

**Returns:** Polynomials $\tilde{h}_j \in \Pi_{n_j-1}$ with

$$|\tilde{h}j| \leqslant M \cdot (1 + 2^{3n-4} \cdot \eta), \tag{7.1}$$

$$\left|1 - \sum_{j=1}^{l} \tilde{h}_j r_j\right| < 2^{3n-2} \cdot \eta^2. \tag{7.2}$$

**Time bound:** $O(\psi(n \cdot H \cdot (\log(1/\eta) + \log M + n)))$.

```
1. Call Algorithm 5.2 to compute a polynomial $\tilde{d} \in \Pi_{n-1}$ with
   $|d - \tilde{d}| < 2^{2-2n} \cdot \eta^2/(lM)$.
2. Use Algorithm 5.3 to compute polynomials $\tilde{d}_j \in \Pi_{n_j-1}$ for $j \in [l]$
   with $|\tilde{d} \bmod p_j - \tilde{d}_j| < (2^{n+n_j-2} - 1) \cdot 2^{2-2n} \cdot \eta^2/(lM)$.
   /* Then $|d_j - \tilde{d}_j| < 2^{1+n_j-n} \cdot \eta^2/(lM)$. */
3. Compute polynomials $g_j \in \Pi_{2n_j-2}$ for $j \in [l]$
   with $|h_j \tilde{d}_j - g_j| < 2^{1+n_j-n} \cdot \eta^2/l$.
   /* Then $|h_j d_j - g_j| < 2^{2+n_j-n} \cdot \eta^2/l$. */
```

4. For $j$ with $n_j \geqslant 2$, use polynomial division to compute polynomials $f_j \in \Pi_{n_j-1}$ for $j \in [l]$ with $|g_j \bmod p_j - f_j| < 2^{4n_j-n-1} \cdot \eta^2/l$.
   For $j$ with $n_j = 1$, let $f_j = g_j$.
   /* Then $|(h_j d_j) \bmod p_j - f_j| < 2^{4n_j-n} \cdot \eta^2/l$. */
5. Return $\tilde{h}_j := h_j + f_j$.

*Correctness Proof and Time Analysis.* The estimates /* between the comment signs */ follow from Lemma 4.4 (Step 2), $|h_j| \leqslant M$ (Step 3), and Corollary 4.5 (Step 4). With $n_j \leqslant n-1$, the estimate

$$\begin{aligned}|r_j| \cdot |\hat{h}_j - \tilde{h}_j| &= |r_j| \cdot |(h_j d_j) \bmod p_j - f_j| \\ &< 2^{n-n_j} \cdot 2^{4n_j-n} \cdot \eta^2/l \\ &\leqslant 2^{3n-3} \cdot \eta^2/l\end{aligned}$$

follows, and this estimate and Lemma 7.1 imply

$$\left|1 - \sum_{j=1}^{l} \tilde{h}_j r_j\right| \leqslant \left|1 - \sum_{j=1}^{l} \hat{h}_j r_j\right| + \sum_{j=1}^{l} |r_j| \cdot |\hat{h}_j - \tilde{h}_j| \leqslant 2^{3n-2} \cdot \eta^2.$$

Moreover, the inequalities $l \geqslant 2$, $n_j \leqslant n-1$, $M \geqslant 1$, and $\eta \leqslant \frac{1}{2}$ imply

$$\begin{aligned}|\tilde{h}_j| &\leqslant |\hat{h}_j| + |\hat{h}_j - \tilde{h}_j| \\ &\leqslant (1 + \tfrac{3}{16} \cdot 2^{n+2n_j} \cdot \eta) \cdot |h_j| + 2^{4n_j-n} \cdot \eta^2/l \\ &\leqslant M \cdot (1 + 2^{3n-4} \cdot \eta).\end{aligned}$$

The time bound follows from Lemmas 3.2 and 3.4 and from Theorems 3.8 and 3.9. ∎

In the following description and analysis of the iteration, let $C = 2^{3n-2}$. For technical convenience, $\tilde{h}_j \leqslant M \cdot (1 + C \cdot \eta)$ is used instead of (7.1).

7.3. ALGORITHM (Partial Fraction Decomposition, Iteration).

**Input:** Polynomials $p_j \in \Pi^1_{n_j}$ and $h_j \in \Pi_{n_j-1}$; real bounds $M \geqslant 1$ and $\eta_0$ with $0 < \eta_0 \leqslant \min\{2^{-3.5n}, 1/M\}$ such that, with $r_j$ and $d$ as in Lemma 7.1, the estimates $|h_j| \leqslant M$ and $|d| < \eta_0$ hold; an error bound $\eta \in (0, \eta_0]$.

**Returns:** Polynomials $\tilde{h}_j \in \Pi_{n_j}$ with

$$|\tilde{h}_j| \leqslant M \cdot \frac{1 + C \cdot \eta_0}{1 - C \cdot \eta_0} \cdot (1 - C \cdot \eta) < \left(1 + \frac{4}{7} \cdot 2^{3n} \cdot \eta_0\right) \cdot M, \tag{7.3}$$

$$\left|1 - \sum_{j=1}^{l} \tilde{h}_j r_j\right| < \eta. \tag{7.4}$$

**Time bound:** $O(\psi(n \cdot H \cdot \log(1/\eta)))$.

```
1. If $\eta \geqslant C\eta_0^2$: {Let $\eta := \eta_0$, $\hat{M} := M$, and $\hat{h}_j := h_j$; goto 4.}
2. Let $\hat{\eta} := \sqrt{\eta/C}$ and $\hat{M} := M \cdot ((1 + C \cdot \eta_0)/(1 - C \cdot \eta_0)) \cdot (1 - C \cdot \eta)$.
3. Call Algorithm 7.3 recursively
   with input $(p_1, ..., p_l; h_1, ..., h_l; M, \eta_0, \hat{\eta})$
   to compute polynomials $\hat{h}_j \in \Pi_{n_j - 1}$ for all $j \in [l]$
   with $|\hat{h}_j| < \hat{M}$ and $|1 - \sum_{j=1}^{l} \hat{h}_j r_j| < \hat{\eta}$.
4. Call Algorithm 7.2 with input $(p_1, ..., p_l; \hat{h}_1, ..., \hat{h}_l; \hat{M}, \hat{\eta})$
   to compute polynomials $\tilde{h}_j \in \Pi_{n_j - 1}(j \in [l])$
   with $|\tilde{h}_j| \leqslant \hat{M} \cdot (1 + C \cdot \hat{\eta})$ and $|1 - \sum_{j=1}^{l} \tilde{h}_j r_j| \leqslant C \cdot \hat{\eta}^2$.
```

*Correctness Proof and Time Analysis.* The algorithm stops, because in Step 2 always $\hat{\eta} \geqslant 2\eta$. Let $c > 0$ be such that the time for a single step as specified in Algorithm 7.2 is at most $c \cdot \psi(n \cdot H \cdot \log(1/\eta))$, when the algorithm is called with $\eta \leqslant \min\{2^{-3.5n}, 2/M\}$. The following induction shows that the time for Algorithm 7.3 is bounded by $7c \cdot \psi(n \cdot H \cdot \log(1/\eta))$.

If $C\eta_0^2 \leqslant \eta$ ($\leqslant \eta_0$), then Step 4 returns polynomials $\tilde{h}_j$ ($j \in [l]$) with $|\tilde{h}_j| \leqslant M \cdot (1 + C\eta_0) \leqslant M \cdot ((1 + C\eta_0)/(1 - C\eta_0)) \cdot (1 - C\eta)$ and $|1 - \sum_{j=1}^{l} \tilde{h}_j r_j| \leqslant C\eta_0^2 \leqslant \eta$ in time at most $c \cdot \psi(n \cdot H \cdot \log(1/\eta_0)) < 7c \cdot \psi(n \cdot H \cdot \log(1/\eta))$.

If $C\eta_0^2 > \eta$, then by induction the time for Step 3 is bounded by $7c \cdot \psi(n \cdot H \cdot \log(1/\hat{\eta}))$. The error bounds in Step 4 and the equations $C\hat{\eta}^2 = \eta$ and $\hat{M} \cdot (1 + C \cdot \hat{\eta}) = M \cdot ((1 + C\eta_0)/(1 - C\eta_0)) \cdot (1 - C\eta)$ show that the polynomials $\tilde{h}_j$ satisfy the required error estimates. Now with $C\eta_0 \leqslant \frac{1}{8}$,

$$\hat{M} < M \cdot \frac{1 + C\eta_0}{1 - C\eta_0} \leqslant M \cdot \left(1 + \frac{16}{7} \cdot C\eta_0\right) \leqslant \frac{9}{7} \cdot M. \tag{7.5}$$

This implies $\hat{\eta} \leqslant \eta_0 \leqslant \min\{2^{-3.5n}, 1/M\} \leqslant \min\{2^{-3.5n}, 2/\hat{M}\}$.

The time for Step 4 is at most $c \cdot \psi(n \cdot H \cdot \log(1/\hat{\eta}))$. Thus the overall cost is bounded by $8c \cdot \psi(n \cdot H \cdot \log(1/\hat{\eta}))$. The relations $\eta < C\eta_0^2 < 2^{-4n}$ and $C = 2^{3n-2}$ imply that this is at most $7c \cdot \psi(n \cdot H \cdot \log(1/\eta))$. ▮

The specification of Algorithm 7.3 and (7.5) yield:

7.4. Theorem. *Given polynomials $p_j \in \Pi^1_{n_j}$ and $h_j \in \Pi_{n_j-1}$, bounds $M \geqslant 1$ and $0 < \eta \leqslant \min\{2^{-3.5n}, 1/M\}$ such that $|h_j| \leqslant M$ and $|1 - \sum_{j=1}^{l} h_j r_j| \leqslant \eta$, and an accuracy parameter $S \geqslant \log(1/\eta)$, polynomials $\tilde{h}_j \in \Pi_{n_j-1}$ with $|\tilde{h}_j| \leqslant M \cdot (1 + \frac{4}{7} \cdot 2^{3n} \cdot \eta)$ and $|1 - \sum_{j=1}^{l} \tilde{h}_j r_j| \leqslant 2^{-S}$ can be computed in time $O(\psi(n \cdot H \cdot S))$.*

*Generalizations.* The time bound is preserved up to constant factors, if the weaker estimate $\eta_0 \leqslant \min\{2^{-(3+\alpha)n}, \beta/M\}$ with arbitrary constants $\alpha, \beta > 0$ is assumed. It is possible to do without the assumption $\eta_0 \leqslant$ const.$/M$. Then the time bound for the first steps is $O(\psi(n \cdot H \cdot \log M))$. If only $\eta_0 \leqslant 2^{-(3n-1)}$ is required instead of $\eta_0 \leqslant 2^{-(3+\alpha)n}$, one has to take into account $O(\log n)$ additional iteration steps with linear convergence. The time for these steps is at most $O(\psi(n^2 H))$. In any case, the estimate $|\tilde{h}_j| \leqslant M \cdot (1 + 4C\eta_0) \leqslant 3M$ holds. ▮

## 8. IMPROVEMENT OF FACTORIZATIONS

The Newton algorithm is based on the following error estimate:

8.1. Lemma. *Let $p \in \Pi^1_n$, $p_j \in \Pi^1_{n_j}$, and $h_j \in \Pi_{n_j-1}$ for $j \in [l]$. Furthermore, let $\tilde{p} := p_1 \cdots p_l$, $r_j := \tilde{p}/p_j$, and $d := 1 - \sum_{j=1}^{l} h_j r_j$. Let $M \geqslant 1$, $0 < \varepsilon \leqslant 2^{-2n+3}/(l^2 M)$, and $0 < \eta < 1$ such that $|h_j| \leqslant M$, $|p - \tilde{p}| < \varepsilon$, and $|d| < \eta$. Finally, let $\varphi_j := (h_j \cdot p) \bmod p_j$, $\hat{p}_j := p_j + \varphi_j$, $\hat{p} := \hat{p}_1 \cdots \hat{p}_l$, and $\hat{r}_j := \hat{p}/\hat{p}_j$ Then*

$$|p - \hat{p}| \leqslant 2^{3n-3} \cdot \varepsilon \cdot \eta + l^2 \cdot M^2 \cdot 2^{5n-7} \cdot \varepsilon^2, \tag{8.1}$$

$$\left| 1 - \sum_{j=1}^{l} h_j \hat{r}_j \right| \leqslant \eta + l^2 \cdot M^2 \cdot 2^{3n-4} \cdot \varepsilon. \tag{8.2}$$

*Proof.* Lemma 4.8 and $\eta < 1$ imply that the $p_j$ are pairwise prime. For $j \in [l]$, the congruence $\varphi_j \equiv h_j p \equiv h_j \cdot (p - \tilde{p}) \bmod p_j$, $|p_j| \geqslant 1$, and Lemma 4.4 yield the estimate

$$\begin{aligned}|\varphi_j| &\leqslant \left(1+|p_j|\cdot\binom{n+n_j-2}{n_j}\right)\cdot|h_j|\cdot\varepsilon\\ &\leqslant 2^{n+n_j-2}\cdot M\cdot|p_j|\cdot\varepsilon\\ &\leqslant \alpha\cdot|p_j|\cdot\varepsilon,\end{aligned}$$

where $\alpha := M\cdot 2^{2n-3}$. Hence $|\hat{p}_j| \leqslant (1+\alpha\varepsilon)\cdot|p_j|$. Note that $\alpha\varepsilon \leqslant 1/l^2$. Next it is shown that for every set $I\subset[l]$ with $\#I=k$, the estimate

$$\left|\prod_{j\in I}\hat{p}_j - \prod_{j\in I} p_j\right| \leqslant k\cdot(1+\alpha\varepsilon)^{k-1}\cdot\alpha\varepsilon\cdot\prod_{j\in I}|p_j| \tag{8.3}$$

holds. It may be assumed w.l.o.g. that $I=[k]$. Indeed,

$$\begin{aligned}|\hat{p}_1\cdots\hat{p}_k - p_1\cdots p_k| &\leqslant \sum_{j=1}^{k}|\hat{p}_1\cdots\hat{p}_{j-1}\varphi_j p_{j+1}\cdots p_k|\\ &\leqslant k\cdot(1+\alpha\varepsilon)^{k-1}\cdot\alpha\varepsilon\cdot|p_1|\cdots|p_k|.\end{aligned}$$

Now let $\hat{e}_1$ and $\hat{e}_2$ denote the errors of at most first and at least second order in $\varphi_j$:

$$p-\hat{p} = \underbrace{\left(p-\tilde{p}-\sum_{j=1}^{l}\varphi_j r_j\right)}_{=:\,\hat{e}_1} - \underbrace{\left(\hat{p}-\tilde{p}-\sum_{j=1}^{l}\varphi_j r_j\right)}_{=:\,\hat{e}_2}.$$

Then $\hat{e}_1 \equiv p-\tilde{p}-\sum_{k=1}^{l}\varphi_k r_k \equiv p-\tilde{p}-\varphi_j r_j \equiv (p-\tilde{p})\cdot(1-h_j r_j) \equiv (p-\tilde{p})\cdot d \bmod p_j$ for $j\in[l]$, thus from the Chinese remainder theorem, $\hat{e}_1 \equiv (p-\tilde{p})\cdot d \bmod (p_1\cdots p_l)$. With Corollary 4.5 and $(p-\tilde{p})\cdot d\in\Pi_{2n-2}$, this implies

$$|\hat{e}_1| \leqslant 2^{3n-3}\cdot\varepsilon\cdot\eta. \tag{8.4}$$

Because of the equation

$$\begin{aligned}\hat{p}-\tilde{p} &= \sum_{j=1}^{l}\hat{p}_1\cdots\hat{p}_{j-1}\varphi_j p_{j+1}\cdots p_l\\ &= \sum_{j=1}^{l}\varphi_j r_j + \sum_{j=2}^{l}(\hat{p}_1\cdots\hat{p}_{j-1} - p_1\cdots p_{j-1})\,\varphi_j p_{j+1}\cdots p_l,\end{aligned}$$

(8.3), $\alpha\varepsilon \leqslant 1/l^2$, and $(l-1)\cdot(1+1/l^2)^l \leqslant l$ (Appendix A.6, Lemma A.1), the second order error is bounded as

$$\begin{aligned}
|\hat{e}_2| &\leqslant \sum_{j=2}^{l} |\hat{p}_1 \cdots \hat{p}_{j-1} - p_1 \cdots p_{j-1}| \cdot |\varphi_j| \cdot |p_{j+1} \cdots p_l| \\
&\leqslant \binom{l}{2} \cdot (1+1/l^2)^{l-2} \cdot 2^n \cdot (\alpha\varepsilon)^2 \\
&\leqslant 2^{n-1} \cdot l^2 \cdot (\alpha\varepsilon)^2 \\
&\leqslant l^2 \cdot M^2 \cdot 2^{5n-7} \cdot \varepsilon^2. \qquad (8.5)
\end{aligned}$$

For $k = l-1$, (8.3) implies $|r_j - \hat{r}_j| \leqslant 2^{n-1} \cdot l \cdot \alpha \cdot \varepsilon = 2^{3n-4} \cdot l \cdot M \cdot \varepsilon$, hence

$$\left|1 - \sum_{j=1}^{l} h_j \hat{r}_j\right| \leqslant \eta + \sum_{j=1}^{l} |h_j| \cdot |r_j - \hat{r}_j| \leqslant \eta + l^2 \cdot M^2 \cdot 2^{3n-4} \cdot \varepsilon. \quad \blacksquare$$

8.2. Algorithm (Factorization, Newton Step).

**Input:** Polynomials $p \in \Pi_n^1$, $p_j \in \Pi_{n_j}^1$, and $h_j \in \Pi_{n_j-1}$; real bounds $M \geqslant 1$ and $\varepsilon, \eta > 0$ with $\varepsilon \leqslant 2^{1-2n}/(l^2 M)$ and $\eta \leqslant \min\{2^{-3.5n}, 1/M\}$ such that with $\tilde{p}$, $r_j$ and $d$ as in Lemma 8.1, the estimates $|h_j| \leqslant M$, $|p - \tilde{p}| < \varepsilon$, and $|d| < \eta$ hold.

**Returns:** Polynomials $\tilde{p}_j \in \Pi_{n_j}^0$ and $\tilde{h}_j \in \Pi_{n_j-1}$ such that with $\tilde{r}_j := \tilde{p}_1 \cdots \tilde{p}_{j-1} \cdot \tilde{p}_{j+1} \cdots \tilde{p}_l$ the following estimates hold:
$|p_j - \tilde{p}_j| < M \cdot 2^{3n-2} \cdot \varepsilon$, $|\tilde{h}_j| \leqslant M \cdot (1 + 2^{3n} \cdot \eta)$,

$$|p - \tilde{p}_1 \cdots \tilde{p}_l| < l^2 \cdot M^2 \cdot 2^{5n-4} \cdot \varepsilon^2, \qquad (8.6)$$

$$\left|1 - \sum_{j=1}^{l} \tilde{h}_j \tilde{r}_j\right| < l^2 \cdot M^2 \cdot 2^{3n-1} \cdot \varepsilon. \qquad (8.7)$$

**Time bound:** $O(\psi(n \cdot H \cdot (\log(1/\varepsilon))))$.

1. Use Algorithm 7.3 to compute polynomials $\tilde{h}_j \in \Pi_{n_j-1}$ $(j \in [l])$ with $|\tilde{h}_j| \leqslant M \cdot (1 + \frac{4}{7} \cdot 2^{3n} \cdot \eta)$ and $|1 - \sum_{j=1}^{l} \tilde{h}_j r_j| < \eta' := l^2 \cdot M^2 \cdot 2^{2n-5} \cdot \varepsilon$.
2. Use Algorithm 5.3 to compute polynomials $u_j \in \Pi_{n_j-1}$ $(j \in [l])$ with $|u_j - p \bmod p_j| \leqslant \varepsilon_1 := l \cdot M \cdot 2^{2n-5} \cdot \varepsilon^2$.
3. Compute polynomials $v_j \in \Pi_{2n_j-2}$ $(j \in [l])$ with $|v_j - \tilde{h}_j \cdot u_j| \leqslant \varepsilon_2 := l \cdot M^2 \cdot 2^{2n-4} \cdot \varepsilon^2$.

4. For $j \in [l]$ with $n_j \geqslant 2$, compute polynomials $\tilde{\varphi}_j \in \Pi_{n_j-1}$ $(j \in [l])$ with $|\tilde{\varphi}_j - v_j \bmod p_j| \leqslant \varepsilon_{3,j} := l \cdot M^2 \cdot 2^{4n-7} \cdot \varepsilon^2 \cdot |p_j|$ by polynomial division.
   For $j \in [l]$ with $n_j = 1$, let $\tilde{\varphi}_j := v_j$.
5. Return $\tilde{p}_j := p_j + \tilde{\varphi}_j$ $(j \in [l])$.

The correctness of Algorithm 8.2 is proved by combining the theoretical error estimates from Lemma 8.1 with the rounding error bounds from the algorithm. Details can be spelled out as in the correctness proof for Algorithm 7.2. This is technically complicated without using new ideas. Therefore, the proof is postponed to Appendix A.6.

In the following Newton algorithm, a slightly stronger condition than $\varrho(p) \leqslant 1$ is used for the sake of technical accuracy. The correctness proof and time analysis are similar to that of Algorithm 7.3. Details and further comments are given in Appendix A.6.

8.3. ALGORITHM (Factorization, Newton Iteration).

**Input:** A polynomial $p \in \Pi_n^1$ such that $\varrho(\tilde{p}) \leqslant 1$ holds for all $\tilde{p} \in \Pi_n^0$ with $|p - \tilde{p}| \leqslant 2^{-7n}$; polynomials $p_j \in \Pi_{n_j}^1$ and $h_j \in \Pi_{n_j-1}$; real bounds $M \geqslant 1$ and $\varepsilon_0, \eta_0 > 0$ with $\varepsilon_0 \leqslant \min\{2^{-7n}/(l^2 M^2),\ 1/(4l^2M^4)\}$ and $\eta_0 \leqslant \min\{2^{-3.5n},\ 1/M\}$ such that, with $\tilde{p}$, $r_j$, and $d$ as in Lemma 8.1, the estimates $|h_j| \leqslant M$, $|p - \tilde{p}| < \varepsilon_0$, and $|d| < \eta_0$ hold; an error bound $\varepsilon \in (0, \varepsilon_0]$.

**Notation:** Let $C := l^2 \cdot M^2 \cdot 2^{6n}$, $B := l^2 \cdot M^2 \cdot 2^{3n}$, and $M' := M \cdot ((1 + 2^{3n}\eta_0)/(1 - \sqrt{C\varepsilon_0}))$.

**Returns:** Polynomials $\tilde{p}_j \in \Pi_{n_j}^1$ and $\tilde{h}_j \in \Pi_{n_j-1}$ such that the estimates $|p_j - \tilde{p}_j| \leqslant M \cdot 2^{3n} \cdot \varepsilon_0$, $|\tilde{h}_j| \leqslant M' \cdot (1 - \sqrt{C\varepsilon})$, $|p - \tilde{p}_1 \cdots \tilde{p}_l| < \varepsilon$, and $|1 - \sum_{j=1}^{l} \tilde{h}_j \cdot \tilde{r}_j| \leqslant (B/\sqrt{C}) \cdot \sqrt{\varepsilon}$ hold, where $\tilde{r}_j := \tilde{p}_1 \cdots \tilde{p}_{j-1} \cdot \tilde{p}_{j+1} \cdots \tilde{p}_l$.

**Time bound:** $O(\psi(n \cdot H \cdot \log(1/\varepsilon)))$.

1. If $\varepsilon \geqslant C\varepsilon_0^2$:
   {Let $\hat{\varepsilon} := \varepsilon_0$, $\hat{\eta} := \eta_0$, $\hat{M} := M$, $\hat{p}_j := p_j$, and $\hat{h}_j := h_j$ for $j \in [l]$; goto 4.}
2. Let $\hat{\varepsilon} := \sqrt{\varepsilon/C}$, $\hat{\eta} := (B/\sqrt{C}) \cdot \sqrt{\hat{\varepsilon}}$, and $\hat{M} := M' \cdot (1 - \sqrt{C\hat{\varepsilon}})$.
3. Call Algorithm 8.3 recursively
   with input $(p; p_1, ..., p_l; h_1, ..., h_l; M, \varepsilon_0, \eta_0; \hat{\varepsilon})$
   to compute polynomials $\hat{p}_j \in \Pi_{n_j}^1$ and $\hat{h}_j \in \Pi_{n_j-1}$
   such that the estimates $|\hat{h}_j| \leqslant \hat{M}$, $|p - \hat{p}_1 \cdots \hat{p}_l| < \hat{\varepsilon}$,
   and $|1 - \sum_{j=1}^{l} \hat{h}_j \hat{r}_j| < \hat{\eta}$ hold, where $\hat{r}_j := \hat{p}_1 \cdots \hat{p}_{j-1} \cdot \hat{p}_{j+1} \cdots \hat{p}_l$.

4. Call Algorithm 8.2 with input $(p; \hat{p}_j; \hat{h}_j; \hat{M}, \hat{\varepsilon}, \hat{\eta})$ to compute polynomials $\tilde{p}_j \in \Pi^0_{n_j-1}$ and $\tilde{h}_j \in \Pi_{n_j-1}$ for $j \in [l]$ with $|\tilde{h}_j| \leqslant \hat{M} \cdot (1 + 2^{3n} \cdot \hat{\eta})$, $|p - \tilde{p}_1 \cdots \tilde{p}_l| < C\hat{\varepsilon}^2$, and $|1 - \sum_{j=1}^{l} \tilde{h}_j \tilde{r}_j| < B\hat{\varepsilon}$.

*Proof of Theorem* 1.7. A simple way to achieve the additional restriction for $p$ in Algorithm 8.3 is to rescale the problem with a factor of 2: Let $p$, $p_j$, $h_j$, $M$, $\varepsilon_0$, and $\eta_0$ be such that $|h_j| \leqslant M$ and that $1/p \approx h_1/p_1 + \cdots + h_l/p_l \operatorname{err}(\varepsilon_0, \eta_0)$ is an initial decomposition. Let $P(z) := 2^{-n} \cdot p(2z)$, $P_j(z) := 2^{-n_j} \cdot p_j(2z)$, and $H_j(z) := 2^{n-n_j} \cdot h_j(2z)$. Then $P \in \Pi^1_n$, $\varrho(P) \leqslant 1/2$, and hence $\varrho(\tilde{P}) \leqslant 1$ for all $\tilde{P} \in \Pi^0_n$ with $|P - \tilde{P}| \leqslant 2^{-n}$ (see the proof of Lemma 3.12 in Appendix A.2). Moreover, $H_j \in \Pi_{n_j-1}$ with $|H_j| \leqslant 2^{n-1} \cdot |h_j| < 2^n \cdot M$, $|P - P_1 \cdots P_l| < \varepsilon_0$, and $|1 - \sum_{j=1}^{l} H_j R_j| \leqslant 2^{n-1} \cdot \eta_0$, where $R_j = P_1 \cdots P_{j-1} \cdot P_{j+1} \cdots P_l$. Therefore, the relations between $\varepsilon_0$, $\eta_0$, and $M$ imply that $P$, $P_j$, and $H_j$ can be used as starting values in Algorithm 8.3. Hence an APFD $1/P \approx \tilde{H}_1/\tilde{P}_1 + \cdots + \tilde{H}_l/\tilde{P}_l \operatorname{err}(2^{-(s_0+n)}, 2^{-s_1})$ can be computed in time $O(\psi(n \cdot H(\mathbf{n}) \cdot (s_0 + s_1)))$ (note that $s_0, s_1 \geqslant n$). Transforming back with $\tilde{p}_j(z) := 2^{n_j} \cdot \tilde{P}_j(z/2)$ and $\tilde{h}_j(z) := 2^{n_j - n} \cdot \tilde{H}_j(z/2)$ yields the desired APFD $1/p \approx \tilde{h}_1/\tilde{p}_1 + \cdots + \tilde{h}_l/\tilde{p}_l \operatorname{err}(2^{-s_0}, 2^{-s_1})$.

This transformation has been chosen because of its simplicity. Proceeding this way has the disadvantage that the bounds $M$ and $\eta_0$ are increased by a factor of $2^{n-1}$ with the forward transform. Therefore, the bounds for $\varepsilon_0$ and $\eta_0$ required in Definition 3.16 are smaller than in the specification of Algorithm 8.3. This means that using this reduction requires more accurate starting values. Two ways to reduce this overhead are to be more careful with the root estimates (due to the accuracy requirements in Algorithm 8.2, the roots of $\tilde{p}_j$ will not be much larger than 1) or to use variants of Lemmas 4.4 and 4.7 for polynomials $p$ with $\varrho(p) > 1$. Such improvements will not improve the asymptotic time bound by more than a constant factor. ▮

## 9. IMPROVEMENTS, GENERALIZATIONS, AND APPLICATIONS

### 9.1. *Arbitrary Numerators*

This section studies how to compute an APFD of $q/p$ for $p \in \Pi^1_n$ and arbitrary $q \in \Pi_{n-1}$ from an APFD of $1/p$. Throughout Subsection 9.1, let $n, l \in \mathbb{N}_+$ and $\mathbf{n} \vdash_l n$. The idea has been sketched in Subsection 3.5.

9.1. Lemma. *Let $p_j \in \Pi^1_{n_j}$, $h_j \in \Pi_{n_j-1}$, and $r_j := p_1 \cdots p_{j-1} \cdot p_{j+1} \cdots p_l$. Let $M \geqslant 1$ and $0 < \eta < \min\{2^{-n}, 1/M\}$ be such that $|h_j| < M$ and $|1 - \sum_{j=1}^{l} h_j r_j| < 2^{-3n} \cdot \eta$. Finally, let $q \in \Pi_{n-1}$ with $|q| = 1$. Then polynomials $q_j \in \Pi_{n_j-1}$ with $|q - \sum_{j=1}^{l} q_j r_j| < \eta$ can be computed from $h_j$ and $q$ in time $O(\psi(n \cdot H(\mathbf{n}) \cdot \log(1/\eta)))$.*

*Proof.* First, it is estimated how the error in the PFD of $1/p$ propagates into the PFD of $q/p$: Let $p = p_1 \cdots p_l \in \Pi_n^1$, $\hat{q}_j = (q \cdot h_j) \bmod p_j \in \Pi_{n_j - 1}$, and $\hat{q} := \sum_{j=1}^{l} \hat{q}_j r_j$. Then $\sum_{i=1}^{l} \hat{q}_i r_i \equiv \hat{q}_j \cdot r_j \equiv q \cdot h_j \cdot r_j \equiv q \cdot \sum_{i=1}^{l} h_i r_i \bmod p_j$ for $j \in [l]$. Lemma 4.8 shows that the $p_j$ are pairwise prime. With the Chinese remainder theorem, it follows that $q - \hat{q} \equiv q \cdot (1 - \sum_{j=1}^{l} h_j r_j) \bmod p$. Now Corollary 4.5 implies

$$|q - \hat{q}| \leqslant 2^{3n-3} \cdot |q| \cdot \left| 1 - \sum_{i=1}^{l} h_i r_i \right| < \eta/8. \tag{9.1}$$

The second step is to compute approximations $q_j$ for $\hat{q}_j$. Let $\eta_1, \eta_2, \eta_3 > 0$.

(1) Compute $\gamma_j \in \Pi_{n_j - 1}$ with $|\gamma_j - q \bmod p_j| < \eta_1$ for $j \in [l]$.

(2) Compute $g_j \in \Pi_{2n_j - 2}$ with $|g_j - \gamma_j h_j| < \eta_2$ for $j \in [l]$.

(3) Compute $q_j \in \Pi_{n_j - 1}$ with $|q_j - g_j \bmod p_j| < \eta_3$ for $j \in [l]$.

For the error estimate, the exact intermediate results are denoted by $\hat{\gamma}_j := q \bmod p_j$ and $\hat{g}_j := \gamma_j h_j$. Then

$$|g_j - \hat{g}_j| \leqslant \eta_2 + |\hat{\gamma}_j h_j - \gamma_j h_j| < \eta_2 + M \cdot |\hat{\gamma}_j - \gamma_j| < \eta_2 + M \cdot \eta_1$$

and, due to Corollary 4.5,

$$\begin{aligned} |q_j - \hat{q}_j| &< \eta_3 + |(g_j - \hat{g}_j) \bmod p_j| < \eta_3 + 2^{3n_j - 3} \cdot |g_j - \hat{g}_j| \\ &< \eta_3 + 2^{3n_j - 3} \cdot (\eta_2 + M\eta_1). \end{aligned}$$

Finally,

$$\begin{aligned} \left| \hat{q} - \sum_{j=1}^{l} q_j r_j \right| &\leqslant \sum_{j=1}^{l} |\hat{q}_j - q_j| \cdot |r_j| \\ &< \sum_{j=1}^{l} (\eta_3 + \cdot 2^{3n_j - 3} \cdot (\eta_2 + M\eta_1)) \cdot 2^{n - n_j} \qquad (9.2) \\ &\leqslant l \cdot 2^{n-1} \cdot \eta_3 + l \cdot 2^{3n-5} \cdot (\eta_2 + M\eta_1). \qquad (9.3) \end{aligned}$$

Estimates (9.1) and (9.3) show that choosing $\eta_1 < 2^{-3n} \cdot \eta/(lM)$, $\eta_2 < 2^{-3n} \cdot \eta/l$, and $\eta_3 < 2^{-n} \cdot \eta/l$ is sufficient for $|q - \sum_{j=1}^{l} q_j r_j| < \eta$.

For the time estimate, bounds for the size of the polynomials involved are needed. Due to Lemma 4.4, $|\hat{\gamma}_j| \leqslant \frac{3}{8} \cdot 2^{n+n_j} \cdot |q| < 2^{2n}$. (For technical convenience, slightly cruder, but simpler estimates are sufficient and preferred.) Moreover, $|\hat{g}_j| \leqslant |h_j| \cdot |\hat{\gamma}_j| < M \cdot 2^{2n}$ and, due to Corollary 4.5, $|\hat{q}_j| \leqslant 2^{3n_j - 3} \cdot |\hat{\gamma}_j| < M \cdot 2^{5n}$. Now let $s = \lceil \log(1/\eta) \rceil$ $(> \max\{n, \log(1/M)\})$.

The above estimates show that $O(s)$ bit accuracy is sufficient for all computation steps. Thus Step (1) can be performed in time $O(\psi(n \cdot H(\mathbf{n}) \cdot s)$ due to Theorem 3.9, and Steps (2) and (3) can be performed in time $\sum_{j=1}^{l} O(\psi(n_j \cdot s)) = O(\psi(n \cdot s))$ due to Lemmas 3.2 and 3.4. ▌

*Proof of Corollary* 1.8. This proof suffers from the same technical complication as the proof of Theorem 1.7 in Section 8.

For the sake of simplicity, it is required that $\varrho(p) \leqslant 1/2$ holds in addition to the assumptions of Corollary 1.8. The case of general $p \in \Pi_n^1$ is reduced to this case by scaling with a factor of 2. Details of this reduction can be spelled out straightforwardly as in the proof of Theorem 1.7.

Now assume that $\varrho(p) \leqslant 1/2$. Compute an APFD $1/p \approx \tilde{h}_1/\tilde{p}_1 + \cdots + \tilde{h}_l/\tilde{p}_l \operatorname{err}(2^{-s_0}, 2^{-s_1-3n})$ within the asserted time bound according to Theorem 1.7. Then $\varrho(\tilde{p}_j) \leqslant 1$ and $|\tilde{h}_j| < 2M$; w.l.o.g. $2^{-s_1} < 1/(2M)$, and suitable $\tilde{q}_j$ can be computed from $q$ and $\tilde{h}_j$ within the asserted time bound, as has been shown in Lemma 9.1. ▌

## 9.2. *Faster Computation of Radius $\vartheta$ Decompositions*

Here it is shown how to use the splitting circle method to compute an initial decomposition:

*Proof of Theorem* 1.4. Let $p \in \Pi_n^1$, $\varepsilon = 2^{-s_0}$, $\eta = 2^{-s_1}$, and $\vartheta = 2^{-\gamma}$ be such that $\varrho(p) \leqslant 1 - \vartheta/(14n)$. Now let $\vartheta' := \vartheta - 2^{-20}\vartheta^4/n^4$,

$$\varepsilon_0 := \frac{1}{n^{10}} \cdot \left(\frac{\vartheta'}{56n}\right)^{4n}, \qquad \text{and} \qquad \eta_0 := \frac{1}{n^2} \cdot \left(\frac{\vartheta'}{56n}\right)^{n}.$$

Compute a $\vartheta'/(14n)$-separated radius $\vartheta'$ decomposition

$$1/p \approx \tilde{h}_1/\tilde{p}_1 + \cdots + \tilde{h}_l/\tilde{p}_l \operatorname{err}(\varepsilon_0, \eta_0). \tag{9.4}$$

According to Theorem 3.15, this can be done in time $O(\psi(n^3 \log n + n^3\gamma))$. Lemma 4.12 yields the estimate

$$|\tilde{h}_j| \leqslant n_j^2 \cdot (1+\eta_0) \cdot \left(\frac{28n}{\vartheta'}\right)^n \cdot p_j \leqslant n^2 \cdot \left(\frac{56n}{\vartheta'}\right)^n =: M,$$

where obviously $M^2 \geqslant 2^{7n}$. Hence (9.4) is an initial decomposition, from which an improved APFD

$$q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, 2^{-3n}\eta) \tag{9.5}$$

can be computed in time $O(\psi(n \cdot H \cdot (s_0 + s_1 + \log M)))$ because of Corollary 1.8. With $\log M = O(n \log n + n\gamma)$, this is the time bound asserted in Theorem 1.4.

It remains to show that (9.5) is a radius $\vartheta$ decomposition. The zeros of the $p_j$ can deviate a little from those of the $\tilde{p}_j$. Therefore, $\vartheta$ has been replaced by the smaller value $\vartheta'$. This effect is extraordinarily small because of the choice of $\varepsilon_0$: Lemma 4.9 and the estimate $|p_1 \cdots p_l - \tilde{p}_1 \cdots \tilde{p}_l| \leqslant 2\varepsilon_0$ imply $\Delta(p_1 \cdots p_l, \tilde{p}_1 \cdots \tilde{p}_l) < 3 \sqrt[n]{2\varepsilon_0} < 2^{-21}(\vartheta')^4/n^4 =: \delta_0$. This estimate, $|\tilde{p}_j - p_j| \leqslant M \cdot 2^{3n} \cdot \varepsilon_0$, Lemmas 4.9 and 4.10 imply $\Delta(p_j, \tilde{p}_j) < \delta_0$ for all $j \in [l]$. This implies that the PFD (9.5) is a radius $\vartheta' + \delta_0$ decomposition, and is $(\vartheta'/(14n) - 2\delta_0)$-separated. Finally, (9.5) is a $\vartheta/(15n)$-separated radius $\vartheta$ decomposition, because $\vartheta' + \delta_0 \leqslant \vartheta$ and $\vartheta'/(14n) - 2\delta_0 > \vartheta/(15n)$. ▌

## 9.3. *Radius $\vartheta$ Decompositions: The General Case*

The specification of the general case is derived from the bounded case by replacing the centered root radius $\bar{\varrho}(p)$ with the radius of the set of roots of $p$ with respect to the chordal distance $d_S$ defined by (1.3),

$$\bar{\varrho}_S(p) := \tfrac{1}{2} \cdot \max\{d_S(u, v) : p(u) = p(v) = 0\}. \tag{9.6}$$

An APFD $q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, \eta)$ is a generalized radius $\vartheta$ decomposition if $\bar{\varrho}_S(p_j) < \vartheta$ for all $j \in [l]$, see Definition 1.5.

9.2. Definition. Let $\delta > 0$. An APFD $q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, \eta)$ is called *stereographically $\delta$-separated* or *$\delta$-$S$-separated* ($S$ for "stereographical" or "sphere") if $d_S(u, v) \geqslant \delta$ for all roots $u$ of $p_i$ and all roots $v$ of $p_j$, where $i, j \in [l]$ with $i \neq j$.

9.3. Definition. A $(\vartheta, \delta)$-$E$-decomposition ($E$ for "Euclidean") is a radius $\vartheta$ decomposition which is $\delta$-separated. A $(\vartheta, \delta)$-$S$-decomposition is a generalized radius $\vartheta$ decomposition which is $\delta$-$S$-separated.

The lemma below follows directly from the definition of $d_S$:

9.4. Lemma. *For all $r > 0$ and all $z, w \in \hat{\mathbb{C}}$,*

$$\min\{r, 1/r\} \cdot d_S(z, w) \leqslant d_S(rz, rw) \leqslant \max\{r, 1/r\} \cdot d_S(z, w).$$

*Proof of Theorem* 1.6. Let $p, q$ as above, and let $\varepsilon = 2^{-s_0}$, $\eta = 2^{-s_1}$, and $\vartheta = 2^{-\gamma}$. Additional bounds $\varepsilon' = 2^{-s_0'}$, $\varepsilon'' = 2^{-s_0''}$, $\eta' = 2^{-s_1'}$, $\eta'' = 2^{-s_1''}$, $\vartheta' = 2^{-\gamma'}$, ... are chosen below as needed. Fixed precision approximations for $\varrho(p)$ and $\varrho^*(p)$ are computed to distinguish three cases: "Small roots only" ($\varrho(p) \leqslant 2$), "large roots only" ($\varrho^*(p) \geqslant 1/2$), and "small and large roots" ($\varrho(p) \geqslant 1.999$ and $\varrho^*(p) \leqslant 0.501$).

*First Case* (*Small Roots Only*). Assume that $\varrho(p) \leqslant 2$. Let $P_0(z) := p(2z)$, $Q_0(z) := q(2z)$, $\alpha := \mathrm{lc}(P_0)$, and $A := |Q_0|$. Then $P := P_0/\alpha \in \Pi_n^1$ and $Q := Q_0/A \in \Pi_{n-1}$ with $|Q| = 1$. The factors are bounded as follows: $|\alpha| \leqslant |P_0| \leqslant 2^n \cdot |p| = 2^n$ and $A = |Q_0| \leqslant 2^{n-1} \cdot |q| = 2^{n-1}$.

Now let $\vartheta' = \vartheta/4$, $\varepsilon' \leqslant \varepsilon/|\alpha|$, and $\eta' \leqslant \eta/A$. It is sufficient to choose $\varepsilon' \leqslant 2^{-n} \cdot \varepsilon$ and $\eta' \leqslant 2^{1-n} \cdot \eta$. Compute a $(\vartheta', \vartheta'/(15n))$-*E*-decomposition $Q/P \approx Q_1/P_1 + \cdots + Q_l/P_l \operatorname{err}(\varepsilon', \eta')$ with $|Q_j| \leqslant 2^{n\gamma + n \log n + O(n)}$. This can be done in time $O(\psi(n^3 \cdot \log n + n^3 \cdot \gamma + n \cdot H(\mathbf{n}) \cdot (s_0 + s_1)))$ due to Theorem 1.4.

For the backward transform, let $\beta := \alpha^{1/n}$, $p_j(z) := \beta^{n_j} \cdot P_j(z/2)$, and $q_j(z) := A \cdot \beta^{n_j - n} \cdot Q_j(z/2)$ for $j \in [l]$. Then $p(z) - p_1(z) \cdots p_l(z) = \alpha \cdot (P(z/2) - P_1(z/2) \cdots P_l(z/2))$ and $q(z) - q_1(z)\, r_1(z) - \cdots - q_l(z)\, r_l(z) = A \cdot (Q(z/2) - Q_1(z/2)\, R_1(z/2) - \cdots - Q_l(z/2)\, R_l(z/2)$, where, as usual, $r_j := p_1 \cdots p_{j-1} \cdot p_{j+1} \cdots p_l$ and $R_j := P_1 \cdots P_{j-1} \cdot P_{j+1} \cdots P_l$ for $j \in [l]$. Hence $q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(|\alpha| \cdot \varepsilon', A \cdot \eta')$, and the choice of $\varepsilon'$ and $\eta'$ guarantees the required error bounds $(\varepsilon, \eta)$.

Due to the scaling, this APFD of $q/p$ is a $(2\vartheta', 2\vartheta'/(15n))$-*E*-decomposition. Lemma 2.1 implies the estimates $\bar{\varrho}_S(p_j) \leqslant \operatorname{diam}(V(p_j)) \leqslant 2\bar{\varrho}(p_j) \leqslant 4\vartheta'$ and $d_S(z, w) \geqslant \frac{2}{5}\, |z - w| \geqslant 4\vartheta'/(75n)$ for all roots $z$ of $P_i$ and $w$ of $P_j$ and all $i, j \in [l]$ with $i \neq j$. Hence the APFD is a $(4\vartheta', 4\vartheta'/(75n))$-*S*-decomposition, i.e., a $(\vartheta, \vartheta/(75n))$-*S*-decomposition. Finally, the estimates for $|Q_j|$ and $A \leqslant 2^{O(n)}$ yield $|q_j| \leqslant 2^{n\gamma + n \log n + O(n)}$.

*Second case* (*Large Roots only*). Assume that $\varrho^*(p) \geqslant 1/2$. Let $P := p^*$ and $Q := q^*$. Then $|P| = |Q| = 1$ and $\varrho(P) \leqslant 2$. Compute a $(\vartheta, \vartheta/(75n))$-*S*-decomposition $Q/P \approx Q_1/P_1 + \cdots + Q_l/P_l \operatorname{err}(\varepsilon, \eta)$ where the numerators are bounded by $|Q_j| \leqslant 2^{n\gamma + n \log n + O(n)}$. The first case shows that this can be done in time $O(\psi(n^3 \cdot \log n + n^3 \cdot \gamma + n \cdot H(\mathbf{n}) \cdot (s_0 + s_1)))$. Let $p_j := P_j^*$ and $q_j(z) := Q_j^*$. Then $q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\varepsilon, \eta)$ and $|q_j| = |Q_j|$. This APFD of $q/p$ is still a $(\vartheta, \vartheta/(75n))$-*S*-decomposition, because $d_S(z, w) = d_S(1/z, 1/w)$ for $z, w \in \hat{\mathbb{C}}$.

*Third Case* (*Small and Large Roots*). Now $\varrho^*(p) \leqslant 0.501$ and $\varrho(p) \geqslant 1.999$. Then there are $k \leqslant n/2$ and a radius $r \in (0.501, 1.999)$ such that (w.l.o.g.) $P := S_r p/|S_r p|$ is $(k, \delta, \mu)$-*E*-factorable with $\log(1/\mu) = O(n)$, $1/\delta \leqslant 3n$, and $1/\delta = O(k \cdot \log n)$. This follows from Lemma 6.6, applied with $r' = 0.501$, $r'' = 1.999$, $\Delta = \ln(\frac{1999}{501})$, and $\beta := 0.345 < \Delta/4$. The lemma asserts that $k$ and $r$ can be computed in time $O(\psi(kn^2 \log n))$.

Let $z$ and $w$ be roots of $P$ with $|z| < 1$ and $|w| > 1$. Then $|z| \leqslant e^{-\delta}$, $|w| \geqslant e^{\delta}$, and hence $|z - w| \geqslant 2\delta$. This implies $d_S(z, w) > 1/(2n)$ as follows: If $|w| \leqslant 3/2$, then $d_S(z, w) = 2|z - w|/(\sqrt{1 + |z|^2} \cdot \sqrt{1 + |w|^2}) \geqslant 4\delta/(\sqrt{2} \cdot \sqrt{13/4}) > 1/(2n)$. If $|w| > 3/2$, then $d_S(z, w) > d_S(1, 3/2) = 2/\sqrt{26} > 1/3 > 1/(2n)$.

Let $Q = S_r q/|S_r q|$ and $r_* := \max\{r, 1/r\}$. Choose $\varepsilon' = 2^{-n} \cdot \varepsilon$ and $\eta' = 2^{1-n} \cdot \eta$. Then $\varepsilon' < \varepsilon/r_*^n$ and $\eta' < \eta/r_*^{n-1}$, because $r_* < 2$. For notational convenience, let $s := \max\{s_0, s_1\}$.

Compute an $(\varepsilon'/3, \eta'/5)$-$E$-splitting $Q/P \approx U/F + V/G$ according to Theorem 6.4. This is done in time $O(\psi((kn^2 \log n) + n \cdot s))$. Now $\varrho(F) < 1$ and $\varrho^*(G) > 1$, i.e., $U/F$ and $V^*/G^*$ fulfil the assumptions of the bounded case (Theorem 1.4) up to trivial standardization. Before the computation of APFDs of $U/F$ and $V^*/G^*$ is discussed, some bounds for the quantities involved in the computations are derived.

Let $k_* := n - k$. Then the estimates $|F| < 2^k$, $|G| \leqslant \frac{9}{8} \cdot 2^{k_*}$, $|U| \leqslant \frac{9}{7} \cdot k \cdot 2^k/\mu$, and $|V| \leqslant \frac{81}{56} \cdot k_* \cdot 2^{k_*}/\mu$ hold due to Lemma 6.3 and Theorem 6.4. In particular, $\log(|K|) = O(n)$ for $K \in \{F, G, U, V\}$. Let $\alpha := G(0)$. Then $0 < |\alpha| \leqslant |G|$ and $\log(|\alpha|) = O(n)$.

The two subproblems are now transformed to the standard form required in Theorem 1.4: Define $\hat{F} := F$, $\hat{U} := U/|U|$, $\hat{G}(z) := G^*/\alpha$, and $\hat{V}(z) := V^*/|V|$. Then $\hat{F} \in \Pi_k^1$, $\hat{U} \in \Pi_{k-1}$ with $|U| = 1$, $\hat{G} \in \Pi_{k_*}^1$, and $\hat{V} \in \Pi_{k_*-1}$ with $|V| = 1$.

For computing decompositions of $\hat{U}/\hat{F}$ and $\hat{V}/\hat{G}$, let $\vartheta' := \vartheta/(2r_*)$ and choose error bounds $\varepsilon'' \leqslant \min\{\varepsilon'/(3 \cdot (|G| + 1)), \eta'/(5\,|V|)\}$, $\varepsilon''' \leqslant \min\{\varepsilon'/(3\,|\alpha|\,|F|), \eta'/(5\,|U|)\}$, $\eta'' \leqslant \eta'/(5 \cdot |U| \cdot (|G| + 1))$, and $\eta''' \leqslant \eta'/(5 \cdot |V| \cdot (|F| + 1))$. The above estimates show that $\log(1/\chi) \leqslant s + O(n)$ for $\chi \in \{\varepsilon'', \varepsilon''', \eta'', \eta'''\}$ is sufficient.

Hence $(\vartheta', \vartheta'/(15n))$-$E$-decompositions

$$\hat{U}/\hat{F} \approx \hat{Q}_1/\hat{P}_1 + \cdots + \hat{Q}_m/\hat{P}_m \,\mathrm{err}(\varepsilon'', \eta'')$$

and

$$\hat{V}/\hat{G} \approx \hat{Q}_{m+1}/\hat{P}_{m+1} + \cdots + \hat{Q}_l/\hat{P}_l \,\mathrm{err}(\varepsilon''', \eta''')$$

with $|Q_j| \leqslant 2^{n\gamma + n \log n + O(n)}$ for $j \in [l]$ can be computed in time

$$\begin{aligned} &O(\psi(k^3 \log k + k^3\gamma + kH(\mathbf{n}_1)(s+n))) \\ &\qquad + O(\psi(k_*^3 \log k_* + k_*^3\gamma + k_* H(\mathbf{n}_2)(s+n))) \\ &= O(\psi(n^3 \log n + n^3\gamma + nH(\mathbf{n})s)), \end{aligned}$$

where $\mathbf{n}_1 = (n_1, ..., n_m)$, $\mathbf{n}_2 = (n_{m+1}, ..., n_l)$, and $\mathbf{n} = (n_1, ..., n_l)$ denote the vectors of the degrees of the $P_j$.

Now let $P_j := \hat{P}_j$ and $Q_j := |U| \cdot Q_j$ for $1 \leqslant j \leqslant m$. For the other factors $(m+1 \leqslant j \leqslant l)$, let $\beta := \alpha^{1/k_*}$, $P_j := \beta^{n_j} \cdot \hat{P}_j^*$ and $Q_j := |V| \cdot \beta^{n_j - k_*} \cdot \hat{Q}_j^*$.

Finally, let $\tilde{F} := P_1 \cdots P_m$, $\tilde{G} := P_{m+1} \cdots P_l$, $R_j := \tilde{F}/P_j$ for $1 \leqslant j \leqslant m$, and $R_j := \tilde{G}/P_j$ for $m+1 \leqslant j \leqslant l$. Then

$$\begin{aligned}|P - P_1 \cdots P_l| &\leqslant |P - FG| + |F| \cdot |G - \tilde{G}| + |\tilde{G}| \cdot |F - \tilde{F}| \\ &\leqslant \varepsilon'/3 + |F| \cdot |\alpha| \cdot \varepsilon''' + (|G| + |\alpha| \cdot \varepsilon''') \cdot \varepsilon'' \leqslant \varepsilon'\end{aligned}$$

and

$$\begin{aligned}&\left| Q - \sum_{j=1}^{m} Q_j R_j \tilde{G} - \sum_{j=m+1}^{l} Q_j R_j \tilde{F} \right| \\ &\quad \leqslant |Q - UG - VF| + |U| \cdot |G - \tilde{G}| + |V| \cdot |F - \tilde{F}| \\ &\quad\quad + |\tilde{G}| \cdot \left| U - \sum_{j=1}^{m} Q_j R_j \right| + |\tilde{F}| \cdot \left| V - \sum_{j=m+1}^{l} Q_j R_j \right| \\ &\quad \leqslant \eta'/5 + |U| \cdot \varepsilon''' + |V| \cdot \varepsilon'' + |\tilde{G}| \cdot |U| \cdot \eta'' + |\tilde{F}| \cdot |V| \cdot \eta''' \leqslant \eta',\end{aligned}$$

hence $Q/P \approx Q_1/P_1 + \cdots + Q_l/P_l \operatorname{err}(\varepsilon', \eta')$. Now let $i, j \in [l]$ with $i \neq j$; let $w_0, w'_0 \in V(P_i)$ and $w_1 \in V(P_j)$. Then $d_S(w_0, w'_0) \leqslant 2 \cdot |w_0 - w'_0| \leqslant 4 \cdot \vartheta'$ and $d_S(w_0, w_1) \geqslant \min\{|w_0 - w_1|, |1/w_0 - 1/w_1|, 1/(2n)\} \geqslant \vartheta'/(15n)$ due to Lemma 2.1. Hence the APFD of $Q/P$ is a $(2\vartheta', \vartheta'/(15n))$-$S$-decomposition.

To transform back to the original problem, let $A = |S_r p|^{1/n}$ and $B = |S_r q|$, $p_j(z) := A^{n_j} \cdot P_j(z/r)$, and $q_j(z) := B \cdot A^{n_j - n} \cdot Q_j(z/r)$. Then

$$q/p \approx q_1/p_1 + \cdots + q_l/p_l \operatorname{err}(\beta_n(S_{1/r}) \cdot |S_r p| \cdot \varepsilon', \beta_{n\text{-}1}(S_{1/r}) \cdot |S_r p| \cdot \eta').$$

The error bounds are at most

$$\beta_n(S_{1/r}) \cdot |S_r p| \cdot \varepsilon' \leqslant \beta_n(S_{1/r}) \cdot \beta_n(S_r) \cdot |p| \cdot \varepsilon' \leqslant r_*^n \cdot \varepsilon' \leqslant \varepsilon$$

and

$$\beta_{n-1}(S_{1/r}) \cdot |S_r q| \cdot \eta' \leqslant \beta_{n-1}(S_{1/r}) \cdot \beta_{n-1}(S_r) \cdot |q| \cdot \eta' \leqslant r_*^n \cdot \eta' \leqslant \eta.$$

Lemma 9.4 implies that the above APFD of $q/p$ is a $(2r_* \vartheta', \vartheta'/(15 r_* n))$-$S$-decomposition, which is a $(\vartheta, \vartheta/(30 r_*^2 n))$-$S$-decomposition because of $\vartheta' = \vartheta/(4r_*)$ and a $(\vartheta, \vartheta/(120n))$-$S$-decomposition because of $1 \leqslant r_* \leqslant 2$. This completes the proof of Theorem 1.6. ▮

*Comments.* If $\varrho(p) \leqslant 1$ (first case), then it is not necessary to scale the argument. $P_0 := p$ and $Q_0 := q$ are used instead. In practice, scaling with an approximation for $\varrho(p)$ may be better than scaling with a factor of 2. One may prefer other constant factors in the definition of $p_j$, e.g., $p_1(z) := \alpha \cdot P_1(z/2)$ and $p_j(z) := P_j(z/2)$ for $j > 1$, and corresponding $q_j$.

For the sake of clarity, two technical details have been omitted. First, multiplication with factors like $A$, $\alpha$, $\beta^{n_j}$, etc. cannot be performed exactly. It is sufficient to compute the multiplications within error bounds $2^{-c \cdot n} \cdot \varepsilon$ resp. $2^{-c \cdot n} \cdot \eta$ and to replace $\varepsilon', \eta'$ with $\varepsilon'/2$, $\eta'/2$.

The second omission is that root perturbation has not been taken into account in the estimates for $\bar{\varrho}(p_j)$ and for the separation of the roots. This problem can be analyzed like in the proof of Theorem 1.4 in Subsection 9.2. For example, it turns out that in the first case it is sufficient to choose $\vartheta' = \vartheta/4 - \delta$ with a small $\delta$ and $\varepsilon' \leqslant (\vartheta/(\text{const.} \cdot n))$. This does not affect the asymptotic time bound. The lower bound for the distance of the roots of different factors of $p$ can still be guaranteed, because the above lower bound $\vartheta/(15n)$ is a simplification which may as well be replaced by $\vartheta/(14.01n)$. This can also be seen from the proof of Theorem 1.4.

Another reduction from the unbounded to the bounded case is described in Appendix A.7. This reduction is more elegant than the one described here, but it yields substantially worse separation bounds.

## 9.4. *High Precision Root Calculation*

*Proof of Theorem* 1.10. Let $\varepsilon \leqslant \min\{2^{-n(s+3)}, 2^{-n(\gamma+5)}\}$ and $\vartheta = 2^{-(\gamma+3)}$. Compute polynomials $p_j \in \Pi^0_{n_j}$ ($j \in [l]$) with $|p - p_1 \cdots p_l| < \varepsilon$ and $\bar{\varrho}(p_j) < \vartheta$. Due to Theorem 1.4, this can be done within the asserted time bound. It is assumed w.l.o.g. that $\varrho(p_1 \cdots p_l) < 1$. Lemma 4.9 implies $\Delta(p, p_1 \cdots p_l) < \min\{2^{-(s+1)}, \vartheta\}$. Let $\bar{v}_j = \bar{z}(p_j)$ for $j \in [l]$. The estimate $\Delta(p, p_1 \cdots p_l) < \vartheta$ implies that each disk $D_j = D_{2\vartheta}(\bar{v}_j)$ contains at least one zero of $p$. With $\text{sep}(p) > 8\vartheta$ it follows that each $D_j$ contains exactly one zero $u_j$. This zero has multiplicity $n_j = \deg(p_j)$ and fulfils $|u_j - \bar{v}_j| < 2^{-(s+1)}$. Let $p_j = z^{n_j} + a_{j,1} z^{n_j - 1} + \cdots$. Then $\bar{v}_j = a_{j,1}/n_j$, and the required root approximation $v_j$ is obtained by computing this value up to $2^{-(s+1)}$. ▌

*Proof of Theorem* 1.11. Let $\varepsilon \leqslant \min\{2^{-s} \cdot (\frac{7}{8} \cdot \vartheta)^{n-1}, 2^{-n(\gamma+5)}\}$ and $\vartheta := 2^{-(\gamma+3)}$. Compute $p_j \in \Pi^0_{n_j}$ ($j \in [l]$) with $|p - p_1 \cdots p_l| < \varepsilon$, $\bar{\varrho}(p_j) < \vartheta$, and (w.l.o.g.) $\varrho(p_1 \cdots p_l) < 1$. Let the zeros $u_1, ..., u_n$ of $p$ and $v_1, ..., v_n$ of $p_1 \cdots p_l$ be numbered such that $\max_{j \in [n]} |u_j - v_j| =: \delta$ is minimal and $|u_1 - v_1| = \delta$. Lemma 4.9 implies $\delta < \vartheta$. Like in the proof of Theorem 1.10, it follows that each disk $D_j$ contains a root of $p$. As $p$ is squarefree and $\text{sep}(p) \geqslant 8\vartheta$, all $p_j$ are linear factors and $l = n$. Because of $|v_1| < 1$, the estimate $\varepsilon > |p(v_1)| \geqslant \delta \cdot (8\vartheta - \delta)^{n-1} \geqslant \delta \cdot (\frac{7}{8} \cdot \vartheta)^{n-1}$ holds, hence $\delta < 2^{-s}$. The time bound follows from Theorem 1.4 and the definition of $\varepsilon$. ▌

*Proof of Theorem* 1.12. The first step is "to make $p$ squarefree": Compute $f = p/\text{gcd}(p, p')$, This can be done in time $O(n \cdot (\log n)^2 \cdot \log\log n \cdot \psi(l))$. Let $p(z) = a_0 (z - v_1)^{n_1} \cdots (z - v_l)^{n_l}$ with $v_i \neq v_j$ for $i \neq j$.

Then $f(z) = b_0 \cdot (z - v_1) \cdots (z - v_l)$ with some $b_0 \in \mathbb{Z}$. Let $|p|_2$ denote the $l_2$-norm of $p$. Then the Landau–Mignotte-bound (Mignotte, 1992, Theorem 4.4) asserts $|f| \leqslant 2^l \cdot |p|_2 \leqslant 2^l \cdot \sqrt{n} \cdot |p| < 2^l \cdot (n+1)^{3/2} \cdot 2^L < 2^{L+3n}$.

The root separation of $p$ and $f$ is bounded by $\operatorname{sep}(p) = \operatorname{sep}(f) > n^{-(n+2)/2} \cdot |p|_2^{1-n} > n^{-(n+2)/2} \cdot |p|^{1-n} > n^{-(n+2)/2} \cdot 2^{-(L+3n)\cdot(n-1)} > 2^{-nL-4n^2}$ (Mignotte, 1992, Theorem 4.6). The root radius of $p$ and $f$ is at most $r := \varrho(p) = \varrho(f) \leqslant |p| \leqslant n \cdot 2^L$ (Mignotte, 1992, Theorem 4.2(i)).

Now let $F(z) := f(rz)/(b_0 r^n)$. Then $F \in \Pi_n^1$ and $\operatorname{sep}(F) = \operatorname{sep}(f)/r \geqslant 2^{-(n+1)\cdot L - 4n^2 - 3n}$. Theorem 1.11 implies that the roots $v_j/r$ of $F$ can be computed up to $2^{-(s+L+3n)}$ in time $O(\psi(n^3 \cdot \log n + n^4 \cdot L + n \cdot \log n \cdot s))$. The roots $v_j$ of $p$ are obtained from $v_j/r$ by rescaling. ∎

## APPENDIX A: PROOF OF DETAILS

### A.1. *Remainder Computation*

With the notation from Lemma 3.4, Schönhage's algorithm for polynomial division (1982a, Sect. 4) works as follows: The Laurent coefficients of $F/P$ at zero are used to compute an approximation $Q$ for the quotient $\hat{Q}$ with $|Q - \hat{Q}| \leqslant 2^{-s-3}$. An approximate remainder $R$ is then computed from the relation $F - QP = H \cdot z^n + R$. Let $F = \hat{Q}P + \hat{R}$. Then $(\hat{Q} - Q) \cdot P = H \cdot z^n + (R - \hat{R})$. Hence $|\hat{R} - R| \leqslant |H| + |\hat{R} - R| = |H \cdot z^n + (\hat{R} - R)| = |(\hat{Q} - Q) \cdot P| \leqslant |(\hat{Q} - Q)| \cdot |P| \leqslant 2^{-s-2}$, because $|P| \leqslant 2$. The error bound leaves a margin for rounding errors in the computation of $R$.

### A.2. *Root Radius Computation*

*Computation of Extremely Small or Large Root Radii* (*Lemma* 3.10). The root radius of a polynomial $p$ is computed by an algorithm described in Sect. 15 of [S]. The analysis results in the time bound stated in Lemma 3.10. This lemma requires the additional restriction $2^{-Cn} \leqslant \varrho(p) \leqslant 2^{Cn}$ which is not mentioned explicitly in [S]. However, such a condition is needed: Initially, the polynomial $p$ is transformed such that $1 \leqslant \varrho(p) \leqslant O(n)$. This is achieved by replacing $p_m(z)$ with $2^{-k} \cdot p(2^k \cdot z)$ with suitable $k \in \mathbb{Z}$, i.e., $k \geqslant |\log(\varrho(p))| - O(\log n)$. This transform can be computed in time $O(n \cdot |k|)$. This is within the time bound of Lemma 3.10, if $|k| = O(n \cdot (\log\log n + \log(1/\sigma)) \cdot \log(1/\sigma))$. In particular, the above restriction is sufficient.

In the technical description of the root radius algorithm in [S], the scaling is performed after the first root squaring step. The problem remains the same. ∎

*Computing the Centered Root Radius.* The equation $\bar{\varrho}(p) = \varrho(T_{\bar{z}(p)}(p))$ suggests to compute $\bar{\varrho}(p)$ with a Taylor shift and subsequent application of

Lemma 3.10. This involves the following problem: The Taylor shift $p \mapsto p_1 \approx T_{\bar{z}(p)}(p)$ must be computed with a precision which guarantees that $\varrho(p_1)$ equals $\bar{\varrho}(p)$ up to a small relative error. As $\bar{\varrho}(p)$ is unknown, one might have to compute the transform repeatedly with increasing accuracy. For the applications in this paper, it is sufficient to compute an approximation for $\bar{\varrho}(p)$ if this value exceeds a given bound $\vartheta$. This leads to the concept of "standard $\vartheta$ approximation" introduced in Definition 3.11. The accuracy for the translation can be chosen according to $\vartheta$. Lemma 4.9 shows that $O(n \log(1/\vartheta))$ bits are sufficient for the Taylor shift.

*Proof of Lemma* 3.12. The example $p_n(z) := (z-1)^n \cdot (z+1)$ with $\bar{\varrho}(p_n) = 1 + (n-1)/(n+1)$ shows that $\bar{\varrho}(p)$ can be almost twice as large as $\varrho(p)$. Therefore, the polynomial $p$ is rescaled such that the root perturbation Lemma 4.9 can be applied directly: Let $p_1(z) := 4^{-n} \cdot p(4z)$. Then $p_1 \in \Pi_n^1$, $\varrho(p_1) \leqslant 1/4$ and hence $\bar{\varrho}(p_1) \leqslant 2 \cdot \varrho(p) \leqslant 1/2$.

Let $v := \bar{z}(p_1)$ and $\bar{p} := T_v p_1$. Then $\varrho(P) < 1$ for all $P \in \Pi_n$ with $|P - \bar{p}| < 2^{-n}$. This is seen as follows: Assume that $P$ has a zero of modulus at least 1. The roots of polynomials depend continuously on the coefficients. Hence there is $\delta \in (0, 1]$ such that $P_\delta := \bar{p} + \delta \cdot (P - \bar{p})$ has a root $w$ with $|w| = 1$. The error bound implies $|\bar{p}(w)| = |P_\delta(w) - \bar{p}(w)| < 2^{-n}$. On the other hand, all roots $u_v$ of $\bar{p}$ have modulus $\leqslant 1/2$, whence $\bar{p}(w) \geqslant \prod_{v=1}^n |w - u_v| \geqslant 2^{-n}$, a contradiction.

Now let $\vartheta = 2^{-\gamma}$. A standard $\vartheta$ approximation for $\bar{\varrho}(p)$ is computed as follows: First, the center of gravity of the roots of $p$ is shifted into the origin: Compute an approximation $P \in \Pi_n^0$ for $\bar{p}$ such that $|P - \bar{p}| < 2^{-n}$ (then $\varrho(P) < 1$) and $\Delta(P, \bar{p}) \leqslant \vartheta/800$. Due to Lemma 4.9, $|P - \bar{p}| < (\vartheta/C)^n$ with some $C > 0$ is sufficient. Hence the Taylor shift can be computed in time $O(\psi(n^2 \cdot \gamma))$.

The second step is to estimate the order of magnitude of $\varrho(P)$ from the size of the coefficients of $P$. This is done by counting the number of zero bits after the binary point of the coefficients. Let $P(z) = z^m + a_1 z^{n-1} + \cdots + a_n$ and $\sigma := \max_{1 \leqslant m \leqslant n} \sqrt[m]{|a_m|}$ Then $\sigma/n \leqslant \varrho(P) < 2\sigma$ (see Subsection 2.2). A crude approximation for $\log(1/\sigma)$ is obtained as follows: For $m \geqslant 1$, let $c_m := \max\{|\mathrm{Re}(a_m)|, |\mathrm{Im}(a_m)|\}$ and $\alpha_m = \lceil -\log c_m \rceil$, the position of the first nonzero bit after the binary point ($\alpha_m = \infty$, if $c_m = 0$). Then $c_m \leqslant |a_m| \leqslant \sqrt{2} \cdot c_m$ and $2^{-\alpha_m} \leqslant c_m < 2 \cdot 2^{-\alpha_m}$, hence $2^{-\alpha_m} \leqslant |a_m| < 2\sqrt{2} \cdot 2^{-\alpha_m}$. Moreover, $\varrho(P) < 1$ yields $|a_m| < \binom{n}{m} \leqslant 2^{n-1}$ and hence $\alpha_m \geqslant 1 - n$.

Let $\beta \in \mathbb{Z}$ be an approximation for $\min_{1 \leqslant m \leqslant n} \alpha_m/m$ in the sense that $\alpha_m/m \geqslant \beta$ for all $m$ and $\alpha_m/m \leqslant \beta + 1$ for at least one $m$. With the above estimates, it is straightforward to show that $\frac{1}{2} \cdot 2^{-\beta} \leqslant \sigma < 2^{-\beta}$ and hence $2^{-\beta}/(2n) \leqslant \varrho(P) < 2 \cdot 2^{-\beta}$.

Computationally, proceed as follows: Determine whether $\beta \geqslant \gamma + 4$ and if not, compute $\beta$. This simply means to find the position of the first 1 bit in the binary expansion of $\mathrm{Re}(a_m)$ and $\mathrm{Im}(a_m)$ and to divide this number by $m \leqslant n$, This can be done in time $O(\psi(n^2 \cdot \gamma))$.

If $\beta \geqslant \gamma + 4$, then $\varrho(P) \leqslant 2 \cdot 2^{-\beta} \leqslant \vartheta/8$, hence $\bar{\varrho}(p_1) = \varrho(\bar{p}) \leqslant \vartheta/8 + \vartheta/800 < \vartheta/4$ and $\varrho(p) < \vartheta$. In this case, $\vartheta$ is returned as a standard $\vartheta$ approximation for $\varrho(p)$.

If $\beta \leqslant \gamma + 3$, then $2^{-\beta}/(2n) \leqslant \varrho(P) \leqslant 2 \cdot 2^{-\beta}$. To obtain more precise information about $\varrho(P)$, rescale again: $P_1(z) := 2^{\beta n} \cdot P(2^{-\beta} \cdot z)$. Then $P \in \Pi_n^0$ and $1/(2n) \leqslant \varrho(P_1) \leqslant 2$. This scaling is done by shifting the coefficients in time $O(n^2 \cdot \beta) = O(n^2 \cdot \gamma)$.

Now choose $\tau > 1$ with $\tau^2 \leqslant 199/197$ and compute an approximation $R_1$ for $\varrho(P_1)$ with $R_1/\tau \leqslant \varrho(P_1) \leqslant \tau \cdot R_1$. Lemma 3.10 shows that $R_1$ can be computed in time $O(\psi(n^2 \log \log n))$. Let $R := 2^{-\beta} \cdot R_1$, then $R/\tau \leqslant \varrho(P) \leqslant \tau \cdot R$ and $R/\tau - \vartheta/800 \leqslant \varrho(\bar{p}) = \bar{\varrho}(p_1) \leqslant \tau \cdot R + \vartheta/800$ because of $\Delta(P, \bar{p}) \leqslant \vartheta/800$. With $\bar{\varrho}(p) = 4\bar{\varrho}(p_1)$, this means $4R/\tau - \vartheta/200 \leqslant \bar{\varrho}(p) \leqslant 4\tau \cdot R + \vartheta/200$.

If $4\tau \cdot R \leqslant \frac{199}{200} \cdot \vartheta$, then $\bar{\varrho}(p) \leqslant \vartheta$, and $\vartheta$ is returned. Otherwise, $\vartheta/200 < 4\tau R/199$ and hence $\bar{\varrho}(p) < \frac{200}{199} \cdot 4\tau R = =: \varrho$ and $\bar{\varrho}(p) \geqslant 4R/\tau - 4\tau R/199 = (\frac{199}{200} \cdot \tau^{-2} - \frac{1}{200}) \cdot \varrho \geqslant 0.98 \cdot \varrho$ because of the choice of $\tau$, i.e., $\varrho$ is a standard $\vartheta$ approximation for $\bar{\varrho}(p)$. ∎

## A.3. *Modular Arithmetic*

*Correctness Proof and Time Analysis for Algorithm* 5.2. It is obvious that $|q_1 p_2 + q_2 p_1| \leqslant n_{1,2} \cdot 2^{n_{1,2}} \cdot M$. Using the rough estimate $|\tilde{p}_j| < 2 \cdot |p_j|$, it can be shown like in the correctness proof for Algorithm 5.1 that $|p_1 p_2 - \tilde{p}_{1,2}| < 2^{-(s + (n - n_{1,2}) + 3d_{1,2}(\mathbf{n}))}$. With $|\tilde{q}_j| < 2 \cdot |q_j|$, one can estimate

$$\begin{aligned} &|q_1 p_2 + q_2 p_1 - \tilde{q}_{1,2}| \\ &\qquad < \eta + |p_1| \cdot |q_2 - \tilde{q}_2| + |p_2| \cdot |q_1 - \tilde{q}_1| + |\tilde{q}_1| \cdot |p_2 - \tilde{p}_2| + |\tilde{q}_2| \cdot |p_1 - \tilde{p}_1| \\ &\qquad < 7 \cdot 2^{-(s + (n - n_{1,2}) + 3d_1(\mathbf{n}))} \\ &\qquad < 2^{-(s + (n - n_{1,2}) + 3d_{1,2}(\mathbf{n}))}. \end{aligned}$$

Let $c > 0$ be such that two polynomials of degree $\leqslant v$ with norm $\leqslant 1$ can be added or multiplied up to an error of $2^{-N}$ in time $\leqslant c \cdot \psi(vN)$. (Clearly addition is much cheaper than multiplication. The same time bound is used to simplify the analysis. This increases the time bound only by a constant factor.) Then the time for Algorithm 5.2 is bounded by $4c \cdot \psi(H_1(\mathbf{n}) \cdot (s + n + \log(nM) + 3d_1(\mathbf{n}) + 1))$. The time bound follows by induction similar to the proof of the time bound for Algorithm 5.1. This proves Theorem 3.8.

*Correctness Proof and Time Analysis for Algorithm* 5.3. The recursive call of the algorithm in Step 3 is justified by $|p_1 p_2 - \tilde{p}_{1,2}| < \frac{4}{9} \cdot 2^{n_{1,2} - 2n - 2d_{1,2}(\mathbf{n})} \cdot 2^{-(\sigma + m)}$. This estimate is proved like in the correctness proof for Algorithm 5.1.

Now $|f_j - \tilde{f}_j|$ is estimated for $j = 1, 2$. Lemma 4.4 implies that $|f_{1,2}| \leqslant \frac{3}{4} \cdot 2^{m+n_{1,2}}$ and thus (estimated roughly again) $|\tilde{f}_{1,2}| < 2^{m+n_{1,2}}$. Hence because of Lemma 4.7,

$$\begin{aligned}
|f_j - \tilde{f}_j| &\leqslant \tfrac{3}{4} \cdot 2^{n_{1,2}+n_j} \cdot |f_{1,2} - \tilde{f}_{1,2}| \\
&\quad + \tfrac{3}{8} \cdot 2^{n_{1,2}+2n_j} \cdot |f_{1,2}| \cdot |p_j - \tilde{p}_j| + \tfrac{1}{8} \cdot 2^{-(\sigma+m)} \cdot |\tilde{f}_{1,2}| \\
&< \tfrac{3}{4} \cdot 2^{n_{1,2}+n_j} \cdot 2^{2h_{1,2}(\mathbf{n})+n_{1,2}} \cdot 2^{-\sigma} \\
&\quad + \tfrac{3}{8} \cdot 2^{n_{1,2}+2n_j} \cdot \tfrac{3}{4} \cdot 2^{m+n_{1,2}} \cdot \tfrac{4}{9} \cdot 2^{n_j - 2n - 2d_j(\mathbf{n})} \cdot 2^{-(\sigma+m)} \\
&\quad + \tfrac{1}{8} \cdot 2^{m+n_{1,2}} \cdot 2^{-(\sigma+m)} \\
&< \tfrac{3}{4} \cdot 2^{2h_j(\mathbf{n})+n_j} \cdot 2^{-\sigma} + \tfrac{1}{8} \cdot 2^{2n_{1,2}+3n_j-2n-2d_j(\mathbf{n})} \cdot 2^{-\sigma} + \tfrac{1}{8} \cdot 2^{n_{1,2}} \cdot 2^{-\sigma} \\
&\leqslant 2^{2h_j(\mathbf{n})+n_j} \cdot 2^{-\sigma}.
\end{aligned}$$

Because of Theorems 3.2 and 3.4, the time for Steps 2 and 4 is bounded by $c \cdot \psi(n_{1,2} \cdot \sigma)$, if $\sigma \geqslant m \geqslant n$, and the time for Step 1 is bounded by $c \cdot \psi(m \cdot \sigma)$, where $c$ is a suitable constant. Now it is easy to prove by induction that the overall running time of the algorithm is bounded by $c \cdot (\psi(m \cdot \sigma) + \psi(H_1(\mathbf{n}) \cdot \sigma))$.

Theorem 3.9 follows with $\sigma := \max\{s + n + 2H_\infty(\mathbf{n}), m\}$ and $H_\infty(\mathbf{n}) = O(n)$ (estimate (4.7)). ▮

## A.4. *Unit Circle Splitting*

This section provides details about unit circle splitting and completes the proof of Theorem 6.4.

In [**S**, Sect. 12], it is shown that an $\varepsilon_0$-$E$-splitting $(F_0, G_0)$ of $P$ with $\varepsilon_0 := \mu^4/(k^4 \cdot 2^{3n+k+1})$ (see [**S**, (11.4)]) and an auxiliary polynomial $H_0 \in \Pi_{k-1}$ such that $H_0 G_0 \equiv D_0 \bmod F_0$ with $|D_0| \leqslant \eta_0 := \mu^2/(k^2 \cdot 2^{2n})$ can be computed in time $O(\psi((k + (1/\delta)) \cdot (n + \log(1/\mu))^2))$.

Starting with $F_0$, $G_0$, and $H_0$, multidimensional Newton iteration is used to compute sequences $(F_t)$, $(G_t)$, and $(H_t)$ such that $(F_t, G_t)$ is an $\varepsilon_t$-$E$-splitting of $P$ with some $\varepsilon_t \leqslant \varepsilon_0^{1.5^t}$ and $H_t G_t \equiv 1 - D_t \bmod F_t$ with $|D_t| \leqslant 2^{n+k+1} \cdot (k^2/\mu^2) \cdot \varepsilon_{t-1} (\leqslant \eta_0)$. The polynomials $F_t$, $G_t$, and $H_t$ are computed from $F_{t-1}$, $G_{t-1}$, and $H_{t-1}$ in time $O(\psi(n \cdot \log(1/\varepsilon_t)))$. Now let $T$ be the smallest integer such that $\varepsilon_T \leqslant \varepsilon$. Then the time for computing $F := F_T$, $G := G_T$, and $H_T$ from the initial values $F_0$, $G_0$, and $H_0$ is bounded by $\sum_{t=1}^{T} O(\psi(n \cdot \log(1/\varepsilon_t))) = O(\psi(n \cdot \log(1/\varepsilon)))$. Details of this algorithm and its analysis are given in [**S**, Sect. 11].

The numerators $U$ and $V$ are computed from $F$, $G$, and $H_T$ as follows: Let $\eta_1, \eta_2, \eta_3 > 0$. (The choice of these error bounds is discussed below.)

(1) Starting with $H_T$, compute $H \in \Pi_{k-1}$ such that $HG \equiv 1 - D \bmod F$ with $|D| < \eta_1$.

(2) Use polynomial multiplication and division to compute $A \in \Pi_{n-2}$ and $U \in \Pi_{k-1}$ such that the error $E_2 := QH - AF - U$ has norm $|E_2| < \eta_2$.

(3) Use polynomial multiplication and division to compute $V \in \Pi_{n-k-1}$ and $B \in \Pi_{k-1}$ such that the error $E_3 := (Q - UG) - VF - B$ has norm $|E_3| < \eta_3$.

$A$ and $B$ need not be computed, but are only used to estimate the error. Hence, "reduced" division algorithms may be used, which compute only the remainder or quotient, respectively.

The error $|Q - UF - VG| = |B + E_3| \leqslant |B| + \eta_3$ is estimated as follows: The congruence $B \equiv Q - UG - VF - E_3 \equiv Q - QHG + E_2 G - E_3 \equiv QD + E_2 G - E_3 \bmod F$ implies $BG \equiv QDG + E_2 G^2 - E_3 G \bmod F$, i.e., there is $q \in \Pi$ with $BG + qF = QDG + E_2 G^2 - E_3 G =: K$.

Estimate (6.2) from Lemma 6.3 implies $|B| \leqslant \frac{8}{7} \cdot (k/\mu) \cdot |F| \cdot |K|$. Now $|Q| = 1$, the definition of $\eta_1$, $\eta_2$, and $\eta_3$, and the estimates $|F| \cdot |G| \leqslant 2^{n-1} \cdot |F \cdot G| \leqslant \frac{9}{8} \cdot 2^{n-1}$ and $|G| \leqslant \frac{9}{8} \cdot 2^{n-k}$ (see Lemmas 4.2 and 6.3) yield

$$\begin{aligned}|F| \cdot |K| &\leqslant |Q| \cdot |D| \cdot |F| \cdot |G| + |E_2| \cdot |G| \cdot |F| \cdot |G| - |E_3| \cdot |F| \cdot |G| \\ &< \tfrac{9}{8} \cdot 2^{n-1} \cdot (\eta_1 + \eta_3) + \tfrac{81}{64} \cdot 2^{2n-k-1} \cdot \eta_2.\end{aligned}$$

Altogether this yields

$$|Q - UG - VF| < \frac{9}{7} \cdot \frac{k}{\mu} \cdot 2^{n-1} \cdot \left( \eta_1 + \frac{9}{8} \cdot 2^{n-k} \cdot \eta_2 + \frac{3}{2} \cdot \eta_3 \right).$$

The cruder, but simpler estimate $|Q - UG - VF| < 2^{2n} \cdot k \cdot (\eta_1 + \eta_2 + \eta_3)/\mu$ shows that choosing $\eta_1 = \eta_2 = \eta_3 \leqslant \mu \cdot \eta/(3k \cdot 2^{2n})$ yields the desired estimate $|Q - VF - UG| < \eta$.

The estimation of the time needed for the divisions requires bounds for the size of the polynomials. $HG \equiv 1 - D \bmod F$ with $|D| < \eta_1$ and (6.2) yield $|H| \leqslant \frac{8}{7} \cdot (k/\mu) \cdot |F| \cdot (1 + \eta_1)$. $|U|$ and $|V|$ are estimated according to Lemma 6.3, too, using $|UG + VF| < |Q| + \eta = 1 + \eta$ and (w.l.o.g.) $\eta \leqslant \frac{1}{8}$:

$$|U| < \frac{8}{7} \cdot \frac{k}{\mu} \cdot |F| \cdot (1 + \eta) \leqslant \frac{9}{7} \cdot \frac{k}{\mu} \cdot 2^k,$$

$$|V| < \frac{8}{7} \cdot \frac{n-k}{\mu} \cdot |G| \cdot (1 + \eta) \leqslant \frac{9}{7} \cdot \frac{9}{8} \cdot \frac{n-k}{\mu} \cdot 2^{n-k}.$$

According to Lemmas 3.2 and 3.4 and the above norm bounds, the multiplications and divisions to compute $QH$, $U$, $Q-UG$, and $V$ within the prescribed error bounds can be performed in time $O(\psi(n \cdot \log(2^n/(\mu \cdot \eta))))$, which is bounded by the time bound asserted in Theorem 6.4.

## A.5. *Computation of Splitting Circles*

Lemma 6.6 is proved in Sect. 16 of [**S**]. It is beyond the scope of this paper to reproduce this proof completely. The algorithm is described briefly. Details are provided when they cannot be seen immediately from [**S**].

Variants of Graeffe's method are used to compute an approximation for $\varrho_k(p)$ for a given index $k$, or, to compute an index $k$ such that $\varrho_k(p)$ is close to $R$ for a given radius $R$ (see [**S**, Theorems 14.1 and 14.2]). A sufficiently large gap between the roots of $p$ is computed with these algorithms, using binary search. Technically, one proceeds as follows: Let $K := \lfloor n/(2\lfloor \log n \rfloor) \rfloor$ and $\alpha := \Delta/(8\lfloor \log n \rfloor)$. With these parameters, a *splitting scale* $(\delta_1, ..., \delta_{n-1})$ is defined by $\delta_j := \delta_{n-j} := \alpha/j$ for $1 \leqslant j \leqslant K$ and $\delta_j := \beta/(n-2K-1)$ for $K < j < n-K$. Now $k$ and $r$ are computed such that, with $q := 0.98$, the estimates $\ln(r/\varrho_k(p)) \geqslant q\delta_k$ and $\ln(\varrho_{k+1}(p)/r) \geqslant q\delta_k$ hold. Details on how to compute $k$ and $r$ and how to prove $\mu > 2^{-Cn}$ are given in [**S**, Sect. 16]. It may be assumed w.l.o.g. that $k \leqslant n/2$ (otherwise replace $P$ with $P^*$). If $k \leqslant K$, then $1/\delta \leqslant k/(q\alpha) = 8 \cdot k \cdot \lfloor \log n \rfloor/(q\Delta) \leqslant 4 \cdot n/(q\Delta)$. Otherwise $1/\delta \leqslant (n-2K-1)/(q\beta) \leqslant n/(q\beta) < 2 \cdot (K+1) \cdot \lfloor \log n \rfloor/(q\beta) \leqslant 2 \cdot k \cdot \lfloor \log n \rfloor/(q\beta)$.

## A.6. *Newton Iteration for Factorization*

The analysis of the Newton algorithm for factorization uses the following technical fact:

A.1. LEMMA. *$(l-1) \cdot (1+1/l^2)^l \leqslant l$ for all $l \in \mathbb{N}_+$.*

*Proof.* First observe that

$$(l-1) \cdot (1+1/l^2)^l = (l-1) \cdot \sum_{j=0}^{l} \binom{l}{j} \cdot l^{-2j} = l + D,$$

where $D = \sum_{j=1}^{l} l^{1-2j} \cdot \binom{l}{j} - \sum_{j=0}^{l} l^{-2j} \cdot \binom{l}{j}$. The proof is completed by the estimate

$$D < \sum_{j=1}^{l} l^{1-2j} \cdot \binom{l}{j} - \sum_{j=0}^{l-1} l^{-2j} \cdot \binom{l}{j} = \sum_{j=0}^{l-1} l^{-2j} \cdot \left(\binom{l}{j+1}\Big/ l - \binom{l}{j}\right)$$
$$= \sum_{j=0}^{l-1} l^{-2j} \cdot \binom{l}{j} \cdot \left(\frac{l-j}{j+1} \cdot \frac{1}{l} - 1\right) < 0. \quad \blacksquare$$

*Correctness Proof and Time Analysis for Algorithm* 8.2. For the sake of clarity, the specification of the algorithm states a larger bound for $|\tilde{h}_j|$ than is achieved in Step 1. This proof uses the even cruder estimate $|\tilde{h}_j| \leqslant 2M$. Let $E := p - p_1 \cdots p_l$ and $E_j := E \bmod p_j \in \Pi_{n_j-1}$. Then $E_j = p \bmod p_j$, i.e., the estimate in Step 2 reads $|u_j - E_j| \leqslant \varepsilon_1$. Now

$$\begin{aligned} |v_j - \tilde{h}_j E_j| &\leqslant |v_j - \tilde{h}_j u_j| + |\tilde{h}_j| \cdot |u_j - E_j| \\ &\leqslant \varepsilon_2 + 2M\varepsilon_1 = 2\varepsilon_2. \end{aligned} \tag{A.1}$$

Let $\hat{\varphi}_j := (\tilde{h}_j \cdot p) \bmod p_j \in \Pi_{n_j-1}$ and $\gamma := l \cdot M^2 \cdot 2^{4n-6} \cdot \varepsilon$. Then

$$|\tilde{\varphi}_j - \hat{\varphi}_j| \ \leqslant \ \gamma \cdot \varepsilon \cdot |p_j|. \tag{A.2}$$

This is proved as follows: If $n_j \geqslant 2$, then

$$\begin{aligned} |\tilde{\varphi}_j - \hat{\varphi}_j| &\overset{(a)}{\leqslant} \varepsilon_{3,j} + |(v_j - \tilde{h}_j E_j) \bmod p_j| \\ &\overset{(b)}{\leqslant} \varepsilon_{3,j} + \left(1 + \binom{2n_j - 2}{n_j} \cdot |p_j|\right) \cdot 2 \cdot \varepsilon_2. \end{aligned}$$

This follows from (a) the specification of Step 4 and the congruence $\hat{\varphi}_j \equiv \tilde{h}_j p \equiv \tilde{h}_j E_j \bmod p_j$ and (b) Lemma 4.4 and (A.1). Now $|p_j| \geqslant 1$, which implies $1 + \binom{2n_j-2}{n_j} \cdot |p_j| \leqslant 2^{2n_j-2} \cdot |p_j| \leqslant 2^{2n-4} \cdot |p_j|$, and the definitions of $\varepsilon_2$ and $\varepsilon_{3,j}$ yield (A.2). If $n_j = 1$, then $|\tilde{\varphi}_j - \hat{\varphi}_j| = |v_j - \tilde{h}_j E_j| \leqslant 2\varepsilon_2 \leqslant \gamma \cdot \varepsilon \cdot |p_j|$.

The congruence $\hat{\varphi} \equiv \tilde{h}_j E_j \equiv \tilde{h}_j E \bmod p_j$ and Lemma 4.4 imply $|\hat{\varphi}| \leqslant M \cdot 2^{2n-2} \cdot \varepsilon \cdot |p_j|$, hence $|\tilde{\varphi}_j| \leqslant \tilde{\alpha} \cdot |p_j| \cdot \varepsilon$, where $\tilde{\alpha} = M \cdot 2^{2n-2} + \gamma < M \cdot 2^{2n-1}$. This yields the estimate of $|p_j - \tilde{p}_j|$. Moreover, $\tilde{\alpha} \cdot \varepsilon < 1/l^2$. Now let

$$\begin{aligned} p - \tilde{p}_1 \cdots \tilde{p}_l = &\underbrace{\left(p - p_1 \cdots p_l - \sum_{j=1}^{l} \hat{\varphi}_j r_j\right)}_{=:\, \hat{e}_1} \\ &- \underbrace{\left(\sum_{j=1}^{l} (\tilde{\varphi}_j - \hat{\varphi}_j) \cdot r_j\right)}_{=:\, \tilde{e}_1} - \underbrace{\left(\tilde{p}_1 \cdots \tilde{p}_l - p_1 \cdots p_l - \sum_{j=1}^{l} \tilde{\varphi}_j r_j\right)}_{=:\, \tilde{e}_2}. \end{aligned}$$

Then $|\tilde{e}_1| \leqslant \sum_{j=1}^{l} |\tilde{\varphi}_j - \hat{\varphi}_j| \cdot |r_j| \leqslant l \cdot 2^n \cdot \gamma \cdot \varepsilon$. Estimates for $\hat{e}_1$ and $\tilde{e}_2$ are derived exactly like those for $\hat{e}_1$ and $\hat{e}_2$ in the proof of Lemma 8.1, simply replacing $\hat{p}_j$, $\varphi_j$, $h_j$, $\eta$, and $\alpha$ with $\tilde{p}_j$, $\tilde{\varphi}_j$, $\tilde{h}_j$, $\eta'$, and $\tilde{\alpha}$. This yields $|\hat{e}_1| \leqslant 2^{3n-3} \cdot \varepsilon \cdot \eta'$ and $|\tilde{e}_2| \leqslant 2^{n-1} \cdot l^2 \cdot (\tilde{\alpha}\varepsilon)^2$ (compare with (8.4) and (8.5)). These estimates and the definitions of $\tilde{\alpha}$, $\gamma$, and $\eta'$ yield (8.6).

Inequality (8.7) is proved exactly like (8.2): The analog of (8.3) implies $|r_j - \tilde{r}_j| \leqslant l \cdot 2^{n-1} \cdot \tilde{\alpha} \cdot \varepsilon$, hence $|1 - \sum_{j=1}^{l} \tilde{h}_j \tilde{r}_j| \leqslant \eta' + l^2 \cdot M \cdot 2^n \cdot \tilde{\alpha} \cdot \varepsilon$, which yields (8.7) by inserting $\eta'$ and $\tilde{\alpha}$.

The time estimate follows from Lemmas 3.2 and 3.4 and Theorems 3.9 and 7.4. ▌

*Correctness Proof and Time Analysis for Algorithm* 8.3. A first complication is that all approximate factors $p_j$, $\tilde{p}_j$ must have root radius less than one. For arbitrary $p \in \Pi_n^1$, this cannot be expected even for small $\varepsilon$. This is only a trivial technical restriction, but for a rigorous proof of Theorem 1.7, and even more for a correct implementation, technical accuracy requires a slightly sharper restriction for $p$ than just $\varrho(p) \leqslant 1$.

The description of the algorithm is simplified by replacing (8.6) with the more inaccurate estimate $|p - \tilde{p}_1 \cdots \tilde{p}_l| < l^2 \cdot M^2 \cdot 2^{6n} \cdot \varepsilon^2$.

The algorithm stops, because $\hat{\varepsilon} \geqslant 2\varepsilon$. Steps 1–3 need no further explanation. To prove the legality of the call of Algorithm 8.2 in Step 4 one first estimates $\hat{M}$: Because of $\varepsilon_0 \leqslant 2^{-7n}/(l^2 M^2)$, $\eta_0 \leqslant 2^{-3.5n}$, and $n \geqslant 2$, the estimate

$$\hat{M} < M' \leqslant M \cdot \frac{1 + 2^{-n/2}}{1 - 2^{-n/2}/l} \leqslant 2M$$

holds. Hence $\hat{\varepsilon} \leqslant \varepsilon_0 \leqslant 2^{-(2n-1)}/(l^2 \hat{M})$. If $\varepsilon < C\varepsilon_0^2$, it is to show that $\hat{\eta}$ is sufficiently small: Indeed, $\hat{\eta} = lM \sqrt[4]{\varepsilon/C} \leqslant lM \sqrt{\varepsilon_0}$, hence because of the assumptions on $\varepsilon_0$: $\hat{\eta} \leqslant \min\{2^{-3.5n}, 1/\hat{M}\}$.

If $\varepsilon \geqslant C\varepsilon_0^2$, then the estimates

$$|\tilde{h}_j| \leqslant M \cdot (1 + 2^{3n}\eta_0) \leqslant M' \cdot (1 - \sqrt{C\varepsilon_0}) < M' \cdot (1 - \sqrt{C\varepsilon}),$$

$C\varepsilon_0^2 \leqslant \varepsilon$, and $B\varepsilon_0 \leqslant (B/\sqrt{C}) \cdot \sqrt{\varepsilon}$ show that the returned data satisfy the specification.

If $\varepsilon < C\varepsilon_0^2$, then this follows from the definitions of $\hat{\varepsilon}$, $\hat{\eta}$, and $\hat{M}$. The product $\tilde{p}_1 \cdots \tilde{p}_l$ is in $\Pi_n^1$ because of $|p - \tilde{p}_1 \cdots \tilde{p}_l| \leqslant \varepsilon_0$, hence $\tilde{p}_j \in \Pi_{n_j}^1$ for $j \in [l]$. For convenience, $|p_j - \tilde{p}_j|$ is estimated "outside the recursion": $|p_j - \tilde{p}_j| < 2M \cdot 2^{3n-2} \cdot (\varepsilon_0 + C\varepsilon_0^2 + C^3\varepsilon_0^4 + C^7\varepsilon_0^8 + \cdots) < 2M \cdot 2^{3n-2} \cdot \varepsilon_0$. Altogether, the correctness of Algorithm 8.3 is established. The time bound is derived like that for Algorithm 7.3. ▌

## A.7. *Another Reduction to the Bounded Case*

The following reduction of the general to the bounded case is more elegant than the one described in Subsection 9.3, but the separation bound for the roots is substantially smaller: Compute a point $w \in \mathbb{C}$ which is sufficiently far away from all roots of $p$ and let $\bar{p}(z) = p(z+w)$, $\bar{q}(z) = q(z+w)$ and $\bar{f} = \bar{p}/\bar{q}$. Due to Lemma 2.5 of Schönhage (1985), one can choose $w$ with $|w| = 1$ such that the inner root radius $\varrho^*(\bar{p})$ is at least $1/(4n^{3/2})$.

Now let $\hat{f}(z) = \bar{f}(1/z)$, $\hat{p}(z) = \bar{p}^*(z) = (R_n p)(z) = z^n p(1/z)$, and $\hat{q}(z) = (R_{n-1} q)(z) = z^{n-1} q(1/z)$. Then $\hat{p}(z) \in \Pi_n$, $\hat{q}(z) \in \Pi_{n-1}$, $\hat{f}(z) = z \cdot (\hat{q}(z)/\hat{p}(z))$, and $R := \varrho(\hat{p}) \leqslant 4n^{3/2}$. Finally put $P(z) = \hat{p}(R \cdot z)$ and $Q(z) = \hat{q}(R \cdot z)$. Then $\varrho(P) \leqslant 1$, i.e., $P$ and $Q$ fulfil the prerequisites of the bounded case (up to trivial standardization).

Computing a radius $\vartheta'$ decomposition $Q/P \approx Q_1/P_1 + \cdots Q_l/P_l$ for suitably chosen $\vartheta'$ within appropriate error bounds and transforming back yields a generalized radius $\vartheta$ decomposition of $q/p$ which can only be guaranteed to be $\vartheta/O(n^3)$-$S$-separated because of the extra scaling with $R$.

## ACKNOWLEDGMENTS

I am indebted to A. Schönhage for proposing the subject of this paper, pointing out some errors in preliminary drafts, and suggesting several improvements. Helpful comments came from T. Ahrendt, B. Bärk, A. Bellido, O. Geupel, A. Grotefeld, M. Heckler, T. Lickteig, D. Reischert, and E. Vetter.

## REFERENCES

Aberth, O. (1973), Iteration methods for finding all zeros of a polynomial simultaneously, *Math. Comp.* **27**, 339–344.

Aigner, M. (1988), "Combinatorial Search," Wiley-Teubner, Chichester, Stuttgart.

Blum, L., Shub, M., and Smale, S. (1989), On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines, *Bull. Amer. Math. Soc. (N.S.)* **21**, 1–46.

Borodin, A., and Munro, I. (1975), "The Computational Complexity of Algebraic and Numeric Problems," Elsevier, New York.

Brent, R. P. (1976), Fast multiple-precision evaluation of elementary functions, *J. Assoc. Comput. Mach.* **23**, 242–251.

Brent, R. P., and Kung, H. T. (1978), Fast algorithms for manipulating formal power series, *J. Assoc. Comput. Mach.* **25**, 581–595.

Carstensen, C. (1992), On Grau's method for simultaneous factorization of polynomials, *SIAM J. Numer. Anal.* **29**, 601–613.

Delves, L. M., and Lyness, J. N. (1967), A numerical method for locating the zeros of an analytic function, *Math. Comp.* **21**, 543–560.

Friedman, J. (1989), On the convergence of Newton's method, *J. Complexity* **5**, 12–33.

Frommer, A. (1988), A unified approach to methods for the simultaneous computation of all zeros of generalized polynomials, *Numer. Math.* **54**, 105–116.

Gourdon, X. (1993), “Algorithmique du théorème fondamental de l’algèbre,” Rapport de Recherche No. 1852, Institut National de Recherche en Informatique et en Automatique.

Gourdon, X. (1996), “Combinatoire, algorithmique et géométrie des polynômes,” Thèse, École Polytechnique.

Grau, A. A. (1971), The simultaneous improvement of a complete set of approximate factors of a polynomial, *SIAM J. Numer. Anal.* **8**, 425–438.

Green, M., Korsak, A., and Pease, M. (1976), Simultaneous iteration towards all roots of a complex polynomial, *SIAM Rev.* **18**, 501–502.

Henrici, P. (1974), “Applied and Computational Complex Analysis,” Vol. I, Wiley, New York.

Jenkins, M. A., and Traub, J. F. (1970), A three-stage variable-shift iteration for polynomial zeros and its relation to generalized Rayleigh iteration, *Numer. Math.* **14**, 252–263.

Kaltofen, E. (1992), “Polynomial Factorization 1987–1991,” Technical Report 92-1, Rensselaer Polytechnical Institute, Troy, NY.

Kerner, I. O. (1966), Ein Gesamtschrittverfahren zur Berechnung von Nullstellen von Polynomen, *Numer. Math.* **8**, 290–294.

Kirrinnis, P. (1994), An optimal bound for path weights in Huffman trees, *Inform. Proc. Lett.* **51**, 107–110.

Knuth, D. E. (1973), “The Art of Computer Programming. Vol. 1. Fundamental Algorithms,” 2nd ed., Addison–Wesley, Reading, MA.

Knuth, D. E. (1981), “The Art of Computer Programming. Vol. 2. Seminumerical Algorithms,” 2nd ed., Addison–Wesley, Reading, MA.

Ko, K. (1991), “Complexity Theory of Real Functions,” Birkhäuser, Boston, MA.

Lehmer, D. H. (1961), A machine method for solving polynomial equations, *J. Assoc. Comput. Mach.* **8**, 151–162.

Mignotte, M. (1974), An inequality about factors of polynomials, *Math. Comp.* **28**, 1153–1157.

Mignotte, M. (1992), “Mathematics for Computer Algebra,” Springer-Verlag, New York.

Neff, C. A. (1994), Specified precision polynomial root isolation is in NC, *J. Comput. Syst. Sci.* **48**, 429–463.

Neff, C. A., and Reif, J. H. (1996), An efficient algorithm for the complex roots problem, *J. Complexity* **12**, 81–115.

Pan, V. Ya. (1987), Sequential and parallel complexity of approximate evaluation of polynomial zeros, *Comput. Math. Appl.* **14**, 591–622.

Pan, V. Ya. (1994), New techniques for approximating complex polynomial zeros, *in* “Proc. 5th Ann. ACM-SIAM Symp. on Discrete Algorithms,” pp. 260–270.

Pan, V. Ya. (1996), Optimal and nearly optimal algorithms for approximating complex polynomial zeros, *Comput. Math. Appl.* **31**, 97–138.

Pasquini, L., and Trigiante, D. (1985), A globally convergent method for simultaneously finding polynomial roots, *Math. Comp.* **44**, 135–149.

Preparata, F. P., and Shamos, M. I. (1985), “Computational Geometry,” Springer-Verlag, New York.

Ostrowski, A. (1966), “Solution of Equations and Systems of Equations,” Academic Press, New York.

Renegar, J. (1985), On the cost of approximating all roots of a complex polynomial, *Math. Programming* **32**, 319–336.

Renegar, J. (1987), On the worst-case complexity of approximating zeros of polynomials, *J. Complexity* **3**, 90–113.

Samelson, K. (1958), Faktorisierung von Polynomen durch funktionale Iteration, *Bayer. Akad. Wiss. Math.-Natur. Kl. Abh.* **95**.

Schätzle, R. (1990), "Zur Störung der Nullstellen von komplexen Polynomen," Diploma thesis, Univ. Bonn, Math. Inst.

Schönhage, A. (1980), Storage modification machines, *SIAM J. Comput.* **9**, 490–508.

Schönhage, A. (1982a), Asymptotically fast algorithms for the numerical multiplication and division of polynomials with complex coefficients, *in* "Computer Algebra EUROCAM '82 (Marseille (1982))," Lecture Notes in Computer Science, Vol. 144, pp. 3–15, Springer-Verlag, New York.

Schönhage, A. (1982b), "The Fundamental Theorem of Algebra in Terms of Computational Complexity," Technical Report, Univ. Tübingen.

Schönhage, A. (1985), Quasi-GCD computations, *J. Complexity* **1**, 118–137.

Schönhage, A. (1986), Equation solving in terms of computational complexity, *in* "Proc. Internat. Congress of Mathematicians, Berkeley, CA," Vol. 1, pp. 131–153.

Schönhage, A. (1990), Numerik analytischer Funktionen und Komplexität, *in* "Jber. d. Dt. Math.-Verein.," Vol. 92, pp. 1–20.

Schönhage, A. (1996), Ordinary root finding, preprint, Univ. Bonn.

Schönhage, A. (1997), personal communication.

Schönhage, A., and Strassen, V. (1971), Schnelle Multiplikation großer Zahlen, *Computing* **7**, 281–292.

Schönhage, A., Grotefeld, A. F. W., and Vetter, E. (1994), "Fast Algorithms: A Multitape Turing Machine Implementation," B.I. Wissenschaftsverlag, Mannheim.

Schröder, J. (1969), Factorization of polynomials by generalized Newton procedures, *in* "Constructive Aspects of the Fundamental Theorem of Algebra" (B. Dejon and P. Henrici, Eds.), Wiley, London.

Shub, M., and Smale, S. (1985), Computational complexity: On the geometry of polynomials and a theory of cost, Part I, *Ann. Sci. École Norm. Sup. (4)* **18**, 107–142.

Shub, M., and Smale, S. (1986), Computational complexity: On the geometry of polynomials and a theory of cost, II, *SIAM J. Comput.* **15**, 145–161.

Shub, M., and Smale, S. (1993), Complexity of Bezout's theorem. I. Geometric aspects, *J. Amer. Math. Soc.* **6**, 459–501.

Smale, S. (1981), The fundamental theorem of algebra and complexity theory, *Bull. Amer. Math. Soc. (N.S.)* **4**, 1–36.

Smale, S. (1986), Algorithms for solving equations, *in* "Proc. Internat. Congress of Mathematicians, Berkeley, CA," Vol. 1, pp. 172–195.

Strassen, V. (1983), The computational complexity of continued fractions, *SIAM J. Comput.* **12**, 1–27.

van der Waerden, B. L. (1971), "Algebra I," 8. Auflage, Springer-Verlag, Berlin.

Weierstrass, K. (1891), Neuer Beweis des Satzes, dass jede ganze rationale Function einer Veränderlichen dargestellt werden kann als ein Product aus linearen Functionen derselben Veränderlichen, *in* "Gesammelte Mathematische Werke," Vol. 3, Mayer & Müller, Berlin.

Weyl, H. (1924), Randbemerkungen zu Hauptproblemen der Mathematik. II. Fundamentalsatz der Algebra und Grundlagen der Mathematik, *Math. Z.* **20**, 142–150.

*Printed in Belgium*