Accelerated Approximation of the Complex Roots of a Univariate Polynomial (Extended Abstract)
Victor Y. Pan, Elias Tsigaridas

► To cite this version:

HAL Id: hal-00980584
https://inria.hal.science/hal-00980584
Submitted on 18 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Accelerated Approximation of the Complex Roots of a Univariate Polynomial

Extended Abstract

Victor Y. Pan
Depts. of Mathematics and Computer Science
Lehman College and Graduate Center
of the City University of New York
Bronx, NY 10468 USA
victor.pan@lehman.cuny.edu
http://comet.lehman.cuny.edu/vpan/

Elias P. Tsigaridas
INRIA, Paris-Rocquencourt Center,
PoSiS Project
UPMC, Univ Paris 06, LIP6
CNRS, UMR 7606, LIP6
Paris, France
elias.tsigaridas@inria.fr

Highly efficient and even nearly optimal algorithms have been developed for the classical problem of univariate polynomial root-finding (see, e.g., [6, 7, 4], and the bibliography therein), but this is still an area of active research. By combining some powerful techniques developed in this area we devise new nearly optimal algorithms, whose substantial merit is their simplicity, important for the implementation.

We first recall the basic concept of the isolation ratio, central also for [6, 7]. Assume a real or complex polynomial
\[ p(x) = \sum_{i=0}^{d} p_i x^i = p_d \prod_{j=1}^{q} (x - z_j), \quad p_d \neq 0, \]
degree \( d \), an annulus \( A(X, R, r) = \{ x : r \leq |x - X| \leq R \} \) on the complex plane with a center \( X \) and the radii \( r \) and \( R \) of the boundary circles. Then the internal disc \( D(X, r) = \{ x : |x - X| \leq r \} \) is \( R/r \)-isolated and \( R/r \) is its isolation ratio if the polynomial \( p \) has no roots in the annulus. Next we reproduce [10 Corollary 4.5]. It shows that Newton’s iteration converges quadratically to a single simple root of \( p \) if it is initiated at the center of a \( 5d^2 \)-isolated disc that contains just this root.

**Theorem 1.** Suppose both discs \( D(c, r) \) and \( D(c, r/s) \) for \( s \geq 5d^2 \) contain a single simple root \( \alpha \) of a polynomial \( p = p(x) \) of degree \( d \). Then Newton’s iteration
\[ x_{k+1} = x_k - p(x_k)/p'(x_k), \quad k = 0, 1, \ldots \]
converges quadratically to the root \( \alpha \) right from the start provided \( x_0 = c \).

Now suppose that we are given a disc with a single simple zero of \( p \) having an isolation ratio \( 1 + \eta \) for a fixed constant \( \eta > 0 \). Can we increase the ratio to \( 5 \)?

\[ \Delta \text{ such that } r_\Delta / \Delta \geq 5d^2. \]

Indeed in this case \( \Delta \leq 0.2r_\eta / d^2 \), and so the disc \( D(c, \Delta) \) is \( 5d^2 \)-isolated.

We can shift and scale the variable \( x \), and so wlog we assume dealing with a \((1 + t)^2\)-isolated disc \( D(0, r) \) for \( r = 1/(1 + t) \) for a fixed \( t > 0 \), and with polynomial \( p \) having a single simple root \( z_1 \) in this disc. Recall the Laurent expansion,
\[ \frac{p'(x)}{p(x)} = \sum_{j=1}^{d} \frac{1}{x - z_j} = - \sum_{k=1}^{\infty} S_k x^{k-1} + \sum_{k=0}^{\infty} s_k x^{-k-1} \]
\[ = \sum_{h=-\infty}^{\infty} c_h x^h. \]

Here \(|x| = 1, s_0 = 1, s_k = z_1^k, S_k = \sum_{i=0}^{d} z_i^{-k}, k = 1, 2, \ldots \) Consequently \( s_k = z_1^k \), whereas \( S_k \) is the kth power sum of the roots lying inside such a disc. In our case this is a single root, the power sum is the root itself, and we just need its approximation \( c \) within an error at most

\[ \Delta \text{ such that } r_\Delta / \Delta \geq 5d^2. \]

Copyright 20XX ACM X-XXXXX-XX/XX/XX ...$15.00.
Let us estimate the approximation errors. Equations (2) and (3) imply that
\[ s_k^h = \sum_{l=1}^{+\infty} c_{-k-1+l} l^q. \]
Moreover, (2) for \( h = -k - 1 \), \( k \geq 1 \) implies that \( s_k = c_{-k-1} \), whereas (2) for \( h = k - 1, k \geq 1 \) implies that \( S_k = -c_{k-1} \). Consequently
\[ s_k^h - s_k = \sum_{l=1}^{+\infty} (c_{lq-k-1} + c_{-lq-k-1}). \]
We assumed in (3) that \( 0 < k < q - 1 \). It follows that \( c_{-lq-k-1} = s_{lq+k} \) and \( c_{lq-k-1} = -S_{lq-k} \) for \( l = 1, 2, \ldots \). Then
\[ s_k^h - s_k = \sum_{l=1}^{+\infty} (s_{lq+k} - S_{lq-k}). \]
On the other hand \( |s_k| \leq z^h \), \( |S_k| \leq (d-1)z^h \), \( h = 1, 2, \ldots \) where \( z = \max_{1 \leq j \leq d} \min(|z_j|, 1/|z_j|) \), and so \( z \leq \frac{1}{1+d} \) in our case. Substitute these bounds into (3) and obtain \( |s_k^h - s_k| \leq (z^h + (d-1)z^h) \). Therefore it is sufficient to choose \( q \) of order \( \log(d) \) to decrease the error of the approximation to the root \( z \) by a factor of \( gd^k \) for any pair of constants \( g \) and \( h \), and so we can ensure the desired error bound \( \Delta \). To support this computation we only need less than \( d \) additions, followed by \( O(\log(d)) \) evaluations of the polynomial \( p_{\eta}(x) \) of degree \( q - 1 \) at the \( l \)th roots of unity for \( l = O(\log(d)) \). This involves \( O(\log(d)\log(\log(d))) \) ops overall. (Here and hereafter “ops” stand for “arithmetic operations”.) Summarizing we obtain the following estimates.

**Theorem 2.** Suppose the unit disc \( D(0, r) = \{ x : |x| \leq 1 \} \) is \((1+\eta)^2\)-isolated for a fixed \( \eta > 0 \) and contains a single simple root \( z \) of a polynomial \( p = p(x) \) of a degree \( d \). Then it is sufficient to apply less than \( d \) additions and \( O(\log(d)\log(\log(d))) \) other ops to compute a \( 5d^2 \)-isolated subdisc of \( D(0, r) \) containing this root.

Combine Thm. 1 and 2 and obtain the following result.

**Corollary 3.** Under the assumptions of Theorem 2 we can approximate the root \( z \) of the polynomial \( p(x) \) within a fixed positive error bound \( \epsilon < 1 \) by using \( O(\log(d)\log(\log(d)) + d\log(\log(1/\epsilon))) \) ops.

**Corollary 4.** Suppose that we are given \( d \) discs, each containing a single simple root of a polynomial \( p = p(x) \) of degree \( d \) and each being \((1+\eta)^2\)-isolated for a fixed \( \eta > 0 \). Then we can approximate all \( d \) roots of this polynomial within a fixed positive error bound \( \epsilon < 1 \) by using \( O(\log^2(d)\log(\log(1/\epsilon))) \) ops.

**Proof:** Apply the same algorithm that supports Corollary 4 concurrently in all \( d \) given discs, but instead of the \( q \)th roots of unity use \( q \) equally spaced points at the boundary circle of each input disc (that is \( dq = O(d\log(d)) \) points overall) and instead of FFT apply the Moenck–Borodin algorithm for multipoiont polynomial evaluation. Also use it at the stage of performing concurrent Newton’s iteration initialized at the centers of the \( 5d^2 \)-isolated subdiscs of the \( d \) input discs, each subdisc computed by the algorithm that supports Theorem 2. Here we work with the \( d \)th degree polynomial \( p \) rather than with the \( q \)th degree polynomials \( p_q \) because to support transition to polynomials \( p_q \) of the degree \( q \) for \( d \) discs we would need to perform \( d \) shifts and scalings of the variable \( x \). Instead we employ the Moenck–Borodin algorithm, which still enables us to obtain a nearly optimal root-refiner. Technically, in a relatively minor change of our algorithm, we replace the matrix \( \Omega = [\omega^{j(k+1)}] \) in (3) by the matrix \( [c + \omega^{j(k+1)}] \), \( k \) and \( \omega \) where \( c \) is invariant in \( j \) and \( k \). The multiplication of the new matrix by a vector \( \mathbf{v} \) is still reduced to multiplication of the matrix \( \Omega \) by a vector \( \mathbf{v} \) with the additional \( 3d \) ops for computing the vector \( c[1]_j k \mathbf{v} \) and adding it to the vector \( \mathbf{v} \).

The Moenck–Borodin algorithm uses nearly linear arithmetic time, and [2] proved that this algorithm supports multipoint polynomial evaluation at a low Boolean cost as well (see also [13–15]). Consequently our algorithm supporting Corollary 4 can be extended to support a nearly optimal Boolean cost bound for refining all simple isolated roots of a polynomial.

We can immediately relax the assumption that the roots are simple because our proof of Theorem 2 applies to a multiple root as well. Furthermore deduce from the Lucas theorem that the isolation ratio of the basic discs in our algorithms does not decrease when we shift from a polynomial to its derivative and higher order derivatives. Therefore we can just apply Newton’s iteration to the derivative or to a higher order derivative to approximate a double or multiple root, respectively.

**Boolean cost bounds**

Hereafter \( \tilde{O} \) denotes the bit or Boolean complexity ignoring logarithmic factors. To estimate it we apply some results from [10–12], which hold in the general case where the coefficients of the polynomials are known up to an arbitrary precision. In our case the input polynomial is known exactly: the parameter \( \lambda \) to be specified in the sequel could be considered as the working precision.

Let \( p \) be given as a \( \lambda \)-approximation, ie \( \| p - \tilde{p} \|_\infty \leq \lambda \). We compute \( p_\eta \) by using \( d \) additions. This produces a polynomial such that \( \| p_q - p_l \|_\infty \leq \lambda + \tau \| lg d \| + 3/2 \| lg^2 d \| + 5/2 \| lg d + 1 \| + 5 \) for all \( i \) (Lemma 16), and similar bounds hold for \( p_\omega \).

The divisions \( p_\eta(\omega) / p_\omega(\omega) \) output complex numbers such that \( \| p_\eta(\omega) / p_\omega(\omega) \| \leq \tau + 2 \| lg d + 1 \| \) with the logarithm of the error \( \leq \lambda + \tau \| lg d + 3/2 \|^2 \| lg d + 2 \| + 2 \| lg d + 1 \| + 11 \). The final DFT produces numbers such that the logarithms of their magnitudes are not greater than \( \tau + 2 \| lg + 2 \| + 4 \) and the logarithms of their approximation errors are at most \( -\lambda + \tau \| lg(8d) + 3/2 \|^2 \| lg d + 4 \| + 18 \) (Lemma 16).

To achieve an error within \( 2^{-\epsilon} \) in the final result, we perform all the computations with accuracy \( \lambda = \epsilon + \tau \| lg(8d) + 3/2 \|^2 \| lg d + 4 \| + 18 \), that is \( \epsilon = \tilde{O}(\log \| lg d \| + \log^2 d) = \tilde{O}(\epsilon + \tau) \). We perform \( d \) additions at the cost \( \tilde{O}_B(d) \) and perform the rest of computations, that is the 3 DFTs, at the cost \( \tilde{O}_B(d \| lg d \| + \mu(\lambda)) \) or \( \tilde{O}_B(d \| lg d \| + \tau) \) (Lemma 16).
If the root that we want to refine is not in the unit disc, then we replace \( \tau \) in our bounds with \( \delta r \).

We apply a similar analysis from [10] Section 2.3 to the Newton iteration (see also [11] Section 2.3) and arrive at the same asymptotic bounds on the Boolean complexity. Only the overhead constants change because now we perform computations with complex numbers.

The overall complexity is \( \mathcal{O}(d^2 \tau + d\ell) \) and the working precision is \( \mathcal{O}(d\tau + \ell) \).

Here we assume the exact input, that is assume the coefficients of the input polynomials known up to arbitrary precision. For the refinement of the root up to precision of \( L \) bits, we arrive at an algorithm with the complexity in \( \mathcal{O}((d^2\tau + dL)) \).

If we are interested in refining all complex roots, we cannot work anymore with the polynomial \( p_0 \) of degree \( q = \mathcal{O}(\log d) \) unless we add the cost of \( d \) shifts of the initial approximations to the origin. Instead we rely on fast algorithms for multipoint evaluation. Initially we evaluate the polynomial \( p_0 \) of degree \( d \) at \( O(d\log d) \) points, and we assume that \( \log |p|\infty \leq \tau \). These \( d \) points approximate the roots of \( p \), and so their magnitude is at most \( \leq 2^\tau \).

We use the following result of [12, Lemma 21]. Similar bounds appear in [2, 3, 14].

**Lemma 5 (Modular representation).** Assume \( m+1 \) polynomials, \( F \in \mathbb{C}[x] \) of degree \( 2mn \) and \( P_j \in \mathbb{C}[x] \) of degree \( n \), for \( j = 1, \ldots, m \) such that \( \|F\|\infty \leq 2^{\tau_1} \) and all roots of the polynomials \( P_j \) for all \( j \) have magnitude of at most \( 2^\ell \). Furthermore assume \( \lambda \)-approximations of \( F \) by \( \bar{F} \) and of \( P_j \) by \( \bar{P}_j \) such that \( \|F - \bar{F}\|\infty \leq 2^{-\lambda} \) and \( \|P_j - \bar{P}_j\|\infty \leq 2^{-\lambda} \).

Let \( \ell = \lambda - \mathcal{O}(\tau_1\log m + m\log n) \). Then we can compute an \( \ell \)-approximations \( \bar{F}_j \) of \( F_j = F \mod P_j \) for \( j = 1, \ldots, m \) such that \( \|F_j - \bar{F}_j\|\infty \leq 2^{-\ell} \) in \( \mathcal{O}(m(n(\ell + \tau_1 + m\log n))) \).

Using this lemma we bound the overall complexity of multipoint evaluation by \( \mathcal{O}(d(L+d\tau)) \). The same bounds holds at the stage where we perform Newton’s iteration. We need to apply Newton’s operator \( \mathcal{O}(1) \) for each root. Each application of the operators consists of two polynomial evaluations. We perform the evaluations simultaneously and apply Lemma 5 to bound the complexity. On similar estimates for the refinement of the real roots see [11].

**Extensions**

The algorithm of [5] computes at nearly optimal cost \( 6d \) isolated initial discs for all \( d \) roots of a polynomial \( p(x) \). By combining this algorithm with ours we obtain a distinct alternative algorithm, which like the one of [6, 7, 8] supports the record nearly optimal bounds on the Boolean complexity of the approximation of all complex polynomial roots, but has the advantage of allowing substantially simpler implementation.

Finally the same algorithm of [11] approximate the power sums of any number \( m \) of roots (forming, e.g., a single cluster or a number of clusters) in an isolated disc. The algorithm runs at the same cost, already stated and depending just on the isolation ratio. Having the power sums available we can readily compute the coefficients of the factor \( f \) of \( p \) of degree \( d \), whose roots are exactly the roots of \( p \) in this disc: this is a numerically stable algorithm using \( \mathcal{O}(m\log(m)) \) ops (cf. [1] pages 34–35).

**Acknowledgments.** VP is supported by NSF Grant CCF-1116736. ET is partially supported by GeoLMI (ANR 2011 BS03 011 06), HPAC (ANR ANR-11-BS02-013) and an FP7 Marie Curie Career Integration Grant.

1. REFERENCES


