Algorithm for Approximating Complex Polynomial Zeros

Victor Pan
Lehman College, CUNY

June 8, 1998

[summary by Bruno Salvy]

Abstract

An algorithm for approximating complex polynomial zeros is presented. Its complexity is optimal up to polylogarithmic factors and holds the current record.

Finding roots of a complex polynomial numerically in a guaranteed way with a fixed prescribed accuracy is difficult when no approximation is known in advance. This task cannot be performed in a fixed precision environment and implementations in computer algebra systems (where arbitrary precision is available) are seldom able to treat polynomials of degree a few hundreds. However, polynomials of very high degree arise frequently when solving a polynomial system by elimination. The work summarized here provides an algorithm supporting the following theorem.

Theorem 1. *Let $p(x)$ be a monic polynomial of degree n and $z_1, \ldots, z_n$ its zeros, with $|z_i| \leq 1$, $i = 1, \ldots, n$. For a fixed positive b, approximations $z_i^\star$ satisfying*

$$|z_i - z_i^\star| < 2^{-b}, \qquad i = 1, \ldots, n \tag{1}$$

can be computed at a cost bounded by $\tilde{O}(n)$ arithmetic operations and $\tilde{O}(n^2(b+n))$ boolean operations. The notation $\tilde{O}$ means that factors $\log n$, $\log b$ or smaller are neglected.

Much more precise statements, proofs and parallel complexity estimates can be found in [5] and a pedagogical introduction to this area is [6].

The statement of the theorem can be modified to accommodate polynomials which are not monic (by first scaling the coefficients) or with roots of modulus larger than 1 by computing a bound on the moduli (see below) and then scaling the polynomial.

1. Lower Bounds

It is clear that the arithmetical complexity $\tilde{O}(n)$ is optimal, since n coefficients of the input polynomial have to be treated. The boolean complexity $\tilde{O}(n^2(b+n))$ is optimal in the very frequent case $n = O(b)$.

Actually, $O(n^2 b)$ is even a lower bound for the computation of *one* root of polynomials of degree n. This bound follows from the high susceptibility of the roots of a polynomial with respect to the coefficients. For instance, the polynomial $x^n - a$ with a small $a > 0$ has for root $a^{1/n}$. If this root is of order 2^{-b}, changing a to 0 is a change of the nb-th bit of a coefficient that changes the b-th bit of the root. This reasoning extends to other coefficients: let $p = O(n)$ and consider $x^n - ax^p$. Then again a change of a bit at position $O(nb)$ modifies the b-th bit of the solution. Thus b bits of the solution depend on $O(nb)$ bits of each of $O(n)$ coefficients, whence the $O(n^2 b)$ lower bound. This example also illustrates why clusters of zeros defeat many numerical algorithms.

2. Outline of the Algorithm

The algorithm is based on a splitting technique where the polynomial p is split into factors of degree k and $n-k$ with $k=\alpha n$, for some $\alpha \in (1/2,\rho)$, ρ being fixed. Applying this process recursively, any polynomial can be completely factored in $O(\log n)$ steps.

The splitting itself is computed in 3 steps:

1. Find a "splitting" circle not "too close" to roots of p and containing αn of them;
2. Compute the polynomial vanishing at these αn roots;
3. Divide p by this polynomial to obtain the other factor.

Each of these steps has to be performed in $O(n^2 b + n^3)$ boolean operations to yield the theorem.

The factors p_k and p_{n-k} of p are computed numerically. The following two lemmas show how the precision with which they are required can be bounded by ensuring that $\epsilon^\star$ is sufficiently small in the following inequality:

$$\|p(x) - p_k(x)p_{n-k}(x)\| \le \epsilon^\star \|p(x)\|, \tag{2}$$

where $\|q(x)\|$ denotes the sum of the moduli of the coefficients of a polynomial q.

Lemma 1. [8] *If*

$$\left\| p(x) - \prod_{i=1}^{n} (x - z_i^\star) \right\| < \epsilon \|p(x)\|,$$

with $-\log_2 \epsilon \ge bn + n + 2$, *the inequalities* (1) *are satisfied.*

Lemma 2. [8] *Let* $p(x)$, $f_1(x),\ldots,f_k(x)$ *and* $f(x), g(x)$ *be polynomials such that*

$$\|p(x) - f_1(x)\cdots f_k(x)\| \le \epsilon \frac{k}{n} \|p(x)\| \tag{3}$$

$$\|f_1(x) - f(x)g(x)\| \le \epsilon_k \|f_1(x)\|, \tag{4}$$

then

$$\|p(x) - f(x)g(x)f_2(x)\cdots f_k(x)\| \le \epsilon \frac{k+1}{n} \|p(x)\|$$

holds, provided

$$\epsilon_k \le \epsilon \frac{\|p(x)\|}{n \prod_{i=1}^{k} \|f_i(x)\|}.$$

From these lemmas follows that it is sufficient to compute the splitting with $\epsilon^\star \le \epsilon/(n2^n)$ in (2), where ϵ comes from Lemma 1.

The splitting circle method was introduced by Schönhage [8, 9]. We now review the algorithms used in steps 1 and 2, together with the recent progress due to Victor Pan.

3. Numerical Factorization

To simplify the notation, assume the unit circle is a splitting circle for the polynomial $p(x)$. Let $p_k(x)$ be the monic polynomial whose k roots are those roots of p lying inside the circle. The computation of $p_k(x)$ relies on the following integral representation of the power sums s_j of its zeros:

$$s_j = \frac{1}{2i\pi} \int_{|z|=1} \frac{p'(z)}{p(z)} z^j \, dz.$$

This idea originates in [2] and was refined by [8] to produce error bounds, i.e., to bound Q such that the s_j's can be computed by the discretization

$$s_j^\star = \frac{1}{Q}\sum_{q=0}^{Q-1} \omega^{(i+1)q}\frac{p'(\omega^q)}{p(\omega^q)}.$$

The value of Q depends on a lower bound for $|p(z)|$ on the unit circle, which in turns is related to a bound on the distance from this circle to the closest root of p, hence the need for a circle "not too close" to the roots in Step 1 of the algorithm.

Efficiency is attained at the price of quite technical developments [8]. If the closest root to the circle is at distance $O(1/n)$, a value of Q of order $O(n^2)$ is used[1] and the corresponding $p'(\omega^q)$ and $p(\omega^q)$ are computed by a discrete Fourier transform. From there, the sums $s_j^\star$ for $j = 0, \ldots, K$ are computed by DFT, K being the smallest power of 2 larger than $k = s_0$. An approximation of the factor $p_k(x)$ can then be recovered efficiently by a variant of Newton-Hensel's lifting (see [1, p. 34]). Then the other factor is obtained by division. In order to reach the right level of complexity, it is necessary to compute only $O(n)$ bits for these steps and then refine the factorization by another Newton like algorithm as follows. Starting from the approximate factorization

$$\|p(x) - p_k^{(0)}(x)p_{n-k}^{(0)}(x)\| \le \epsilon,$$

where $p_k^{(0)}$ has degree k, the aim is to find a refinement $p_k^{(1)} = p_k^{(0)} + q_k$, $p_{n-k}^{(1)} = p_{n-k}^{(0)} + q_{n-k}$ with $\deg q_i < i$, improving the error. Since

$$p - p_k^{(1)}p_{n-k}^{(1)} = (p - p_k^{(0)}p_{n-k}^{(0)}) - p_k^{(1)}p_{n-k}^{(0)} - p_k^{(0)}p_{n-k}^{(1)} - p_k^{(1)}p_{n-k}^{(1)},$$

the Newton iteration is obtained by satisfying

$$(p - p_k^{(0)}p_{n-k}^{(0)}) = p_k^{(1)}p_{n-k}^{(0)} + p_k^{(0)}p_{n-k}^{(1)}, \tag{5}$$

which determines $p_k^{(1)}$ and $p_{n-k}^{(1)}$ uniquely. These polynomials could be found by Euclid's algorithm, but this is too expensive. Instead, one also computes an inverse $q^{(i)}$ of $p_{n-k}^{(i)}$ modulo $p_k^{(i)}$ by a second, parallel, Newton iteration and then $p_k^{(i)}$ is given by $q^{(i)}p = q^{(i)}(p - p_k^{(i)}p_{n-k}^{(i)}) \bmod p_k^{(i)}$. A similar formula gives $p_{n-k}^{(i)}$. Then the required precision is obtained after a few iteration at a cost bounded by $O(n \log \epsilon^\star)$.

4. **Finding Splitting Circles**

The basic technique to find discs containing a known number of roots of a polynomial is the iteration of Graeffe's method (see [3]). Starting from $p(x)$ of degree n, one performs the following iteration:

$$p_{i+1}(x^2) = (-1)^n p_i(x)p_i(-x),$$

which transforms the polynomial $p_i(x)$ into a polynomial $p_{i+1}(x)$ whose roots are the squares of the roots of $p_i(x)$. This process emphasizes the differences of moduli between the roots. The coefficients of these iterates are Newton sums from which precise information about the different moduli of the roots of the original polynomial can be recovered at a low cost. More precisely, one gets the following lemma.

[1] More precise values are given in [8, p. 35].

Lemma 3. *Let $z_1, \ldots, z_n$ be the roots of $p(x)$, satisfying $|z_1| \leq \cdots \leq |z_n| \leq 1$. Given $c > 0$ and $d \geq 0$, it is possible to compute $\underline{r}_1, \overline{r}_1, \ldots, \underline{r}_n, \overline{r}_n$ such that $\underline{r}_k \leq |z_k| \leq \overline{r}_k = (1 + c/n^d) r_k$, $k = 1, \ldots, n$ with $\tilde{O}(n)$ arithmetic operations.*

This iteration is applied after having first shifted the origin to the center of gravity of the roots, which is given by the first two coefficients of the polynomial. When it follows from this computation that there is a $k = \alpha n$, α in a fixed interval $(\rho, 1-\rho)$, with some $\rho < 1$ such that $|z_{k+1}|/|z_k| \geq 1 + c/n$ for some c fixed in advance, then this yields a splitting circle and the factoring algorithm of the previous section can be applied.

It is when no such circle can be found that progress has been made by Victor Pan recently. In this case, there is an annulus centered at 0 which contains most of the roots of the polynomial. Now the idea is to shift the origin to each of $r' = 2\overline{r}_{11n/12}$ and ir', and apply the same method. Then either a good splitting circle is found, or there is a small circle which is easily computed and contains the intersection of these three annuli, itself containing an important cluster of zeros (at least half of the zeros of p if $c = 1/100$). In this case, the idea is that one of the zeros of a derivative of p of high order (for instance, one can take $p^{(\lfloor n/2 \rfloor + 1)}$) is either the center of a good splitting circle or makes it possible to isolate a *massive cluster* of zeros, where more than half of the zeros of p are at distance less than the desired accuracy 2^{-b}. In both cases, the polynomial can then be factored numerically and the computation proceeds on those factors that do not correspond to a massive cluster. Many refinements are given in [5], in particular it is shown that it is not necessary to compute all the zeros of $p^{(\lfloor n/2 \rfloor + 1)}$.

Conclusion

This summary is a very rough sketch of a very detailed study given in [5]. For practical polynomial solving, other algorithms are known to perform extremely well, but their complexity analysis has yet to be done.

The talk also mentioned extensions to the multivariate case, this is described in [4].

Bibliography

[1] Bini (Dario) and Pan (Victor Y.). – *Polynomial and matrix computations. Vol. 1.* – Birkhäuser Boston Inc., Boston, MA, 1994, *Progress in Theoretical Computer Science*, xvi+415p. Fundamental algorithms.

[2] Delves (L. M.) and Lyness (J. N.). – A numerical method for locating the zeros of an analytic function. *Mathematics of Computation*, vol. 21, 1967, pp. 543–560.

[3] Henrici (Peter). – *Applied and computational complex analysis.* – Wiley-Interscience, New York, 1974, *Pure and Applied Mathematics*, vol. Volume 1: Power series, integration, conformal mapping, location of zeros, xv+682p.

[4] Mourrain (Bernard) and Pan (Victor Y.). – Asymptotic acceleration of solving multivariate polynomial systems of equations. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. pp. 488–496. – ACM Press, 1998.

[5] Pan (V. Y.). – Optimal and nearly optimal algorithms for approximating polynomial zeros. *Computers & Mathematics with Applications*, vol. 31, n° 12, 1996, pp. 97–138.

[6] Pan (Victor). – Solving polynomials with computers. *American Scientist*, vol. 86, 1998, pp. 62–69.

[7] Pan (Victor Y.). – Optimal (up to polylog factors) sequential and parallel algorithms for approximating complex polynomial zeros. In *SIAM Journal on Computing*. pp. 741–750. – ACM Press, 1995.

[8] Schönhage (Arnold). – *The fundamental theorem of algebra in terms of computational complexity.* – Technical report, Mathematisches Institut der Universität Tübingen, 1982. Preliminary report.

[9] Schönhage (Arnold). – Equation solving in terms of computational complexity. In *Proceedings of the International Congress of Mathematicians*, pp. 131–153. – 1987. Berkeley, California, 1986.