MATHEMATICS OF COMPUTATION, VOLUME 32, NUMBER 142
APRIL 1978, PAGES 391–397

The Comparison of Numerical Methods for Solving Polynomial Equations

By Aurél Galántai

Abstract. In this paper we compare the Turán process [5]–[6] with the Lehmer-Schur method [2]. We prove that the latter is better.

1. **The Algorithms.** We first describe the Turán process [5]–[6] which can be considered as an improvement of Graeffe's method. For the complex polynomial

$$(1.1)\qquad p_0(z) \equiv \sum_{j=0}^{n} a_{j0} z^j = 0 \qquad (a_{j0} \in \mathbf{C},\ a_{00}a_{n0} \neq 0),$$

the method can be formulated as follows.

Let

$$(1.2)\qquad p_j(z) \equiv p_{j-1}(\sqrt{z})p_{j-1}(-\sqrt{z}) \equiv \sum_{k=0}^{n} a_{kj} z^j \qquad (j = 1, 2, \dots)$$

be the jth Graeffe transformation and let

$$(1.3)\qquad M[p_0(z), m_0] = \left[\max_{1 \leqslant k \leqslant n} \left|\frac{\sigma_k}{n}\right|^{\mu_0/k}\right]^{-1},$$

where $\mu_0 = 2^{-m_0}$, $\sigma_0 = 0$,

$$(1.4)\qquad \sigma_k = \left[k a_{km_0} - \sum_{j=1}^{k-1} a_{jm_0}\sigma_{k-j}\right] \Big/ a_{0m_0} \qquad (k = 1, \dots, n)$$

and $m_0 \geqslant 1$ is fixed.

Let the constants α_{m_0}, l be defined by the inequalities

$$0.5 < \alpha_{m_0} < 5^{-\mu_0}, \qquad l > \pi\left[\arccos \frac{2.5 + \alpha_{m_0}}{2 + 2\alpha_{m_0}}\right]^{-1} - 1, \qquad m_0 \geqslant 2.$$

Then with the notations

$$(1.5)\qquad M^{(0)} = M[p_0(z), m_0], \qquad S^{(0)} = 0,$$

the dth step of the algorithm is the following:

1. *Algorithm* (*T*). (i) Let

$$S_j^{(d+1)} = S^{(d)} + 0.5(1 + \alpha_{m_0})M^{(d)}\exp\left(j\,\frac{2\pi i}{l+1}\right),$$

where $j = 0, 1, \dots, l$ and $i = \sqrt{-1}$.

Received January 14, 1976; revised August 18, 1976.
AMS (MOS) subject classifications (1970). Primary 65H05.

(ii) If there exists an index j such that $p_0(S_j^{(d+1)}) = 0$, then we get a root and the process terminates.

(iii) Let us compute the quantities

$$M_j^{(d+1)} = M[p_0(z + S_j^{(d+1)}), m_0] \qquad (j = 0, 1, \ldots, l)$$

and let

$$M^{(d+1)} = \min_j M_j^{(d+1)} = M_{j(d)}^{(d+1)}, \qquad S^{(d+1)} = S_{j(d)}^{(d+1)}.$$

Turán [5] proved that $S^{(d)}$ tends to a root of $p_0(z)$, and the convergence is linear. Turán [5] also proved that the number of iterations needed to achieve an arbitrary relative error ϵ (> 0) is independent of $p_0(z)$ and depends on degree $p_0(z)$ only.

Our purpose is to answer the remarks of the last section of [6]. For this reason we compare the Turán process with the Lehmer-Schur method which is often applied in practice ([2], [3], [4]). This algorithm can be described as follows.

Let

(1.6) $$T[p_0(z)] = \sum_{j=0}^{n-1} (\bar{a}_{00}a_{j0} - a_{n0}\bar{a}_{n-j,0})z^j$$

and

(1.7) $$T^j[p_0(z)] = T\{T^{j-1}[p_0(z)]\} \qquad (j = 2, \ldots).$$

Let us compute the numbers $c_j = T^j[p_0(0)]$, $(j = 1, \ldots, k)$, where

(1.8) $$k = \min\{m \in \mathbf{N} | c_m = 0\}.$$

Here, $\mathbf{N}$ denotes the set of nonnegative integers. With the aid of the sequence $\{c_j\}_{j=1}^k$ we define the function $N[p_0(z)]$ as follows

$$N[p_0(z)] = \begin{cases} 1 & \text{if } \exists j \in \{1, \ldots, k-1\} \text{ such that } c_j < 0, \\ 0 & \text{if } c_j > 0 \ (j = 1, \ldots, k-1) \text{ and degree } T^{k-1}[p_0(z)] = 0, \\ -1 & \text{otherwise.} \end{cases}$$

Lehmer [2] proved that if $N[p_0(z)] = 1$ then the polynomial $p_0(z)$ has a root in $\{z \in \mathbf{C} | |z| \leqslant 1\}$, if $N[p_0(z)] = 0$ then $p_0(z)$ has no roots in this set. We shall return to the case $N[p_0(z)] = -1$.

Let us introduce the notations

(1.9) $$\alpha_j^{(d)} = \begin{cases} 0.5\gamma_0^{(d)}R^{(d-1)} & (j = 0), \\ 0.4\gamma_j^{(d)}R^{(d-1)} & (j = 1, \ldots, 8), \end{cases}$$

and

(1.10) $$\beta_j^{(d)} = \begin{cases} z^{(d-1)} & (j = 0), \\ z^{(d-1)} + \dfrac{0.75R^{(d-1)}}{\cos\frac{\pi}{8}} \exp\left(\dfrac{2\pi i(j-1)}{8}\right) & (j = 1, \ldots, 8), \end{cases}$$

where the sequences $\{R^{(d)}\}$, $\{z^{(d)}\}$ and $\{\gamma_j^{(d)}\}$ are defined by the dth step of the Lehmer-Schur method ($d = 1, \ldots$). Let $\widetilde{p}_0(z) = p_0(z)/\psi$ ($\psi > 0$) and

$$z^{(0)} = 0; \qquad R^{(0)} = 1 + \max_j \left| \frac{a_{j0}}{a_{n0}} \right| . \tag{1.11}$$

Then the dth step can be written as follows.

2. *Algorithm* (L). (i) If there exists an index j such that $p_0(\beta_j^{(d)}) = 0$, then we get a root and the process terminates.

(ii) We choose the index $j \in \{0, 1, \ldots, 8\}$ such that

$$N[\widetilde{p}_0(\alpha_j^{(d)} z + \beta_j^{(d)})] = 1$$

and let

$$z^{(d)} = \beta_j^{(d)}, \qquad R^{(d)} = \alpha_j^{(d)}.$$

The numbers $\gamma_j^{(d)} \in [1, 1 + \delta]$, ($\delta \leqslant 0.5$) are chosen such that $N[\widetilde{p}_0(\alpha_j^{(d)} z + p_j^{(d)})] \geqslant 0$ will be satisfied (except in unusual circumstances $\gamma_j^{(d)} = 1$ can be chosen). Lehmer [2] proved that process converges linearly. The number of iteration steps needed to achieve an arbitrary absolute error ϵ (> 0) depends on $p_0(z)$.

2. **The Limitations of the Algorithms.** Denote by $\mathbf{Z}$ the set of integers and let $\mathbf{P}_n$ be the set of complex polynomials of degree n.

A numerical method M (iterative process) for solving $p_0(z) = 0$ where $p_0(z) \in \mathbf{P}_n$ can be identified with the sequence $\{b_k\} \subset \mathbf{C}$ which rises from the computation. This sequence depends on $p_0(z)$ and will be denoted by $\{Mp_0\} = \{b_k\}$. There exists a subsequence $\{b_{k_j}\}$ of $\{b_k\}$ such that

$$z^* = \lim_{j\to\infty} b_{k_j} \quad \text{and} \quad p_0(z^*) = 0. \tag{2.1}$$

A digital computer can perform elementary (complex) operations only over the finite set

$$S[0, K] \cap \mathbf{C}_\delta, \tag{2.2}$$

where $S[0, K] = \{z \in \mathbf{C} \mid |z| \leqslant K\}$ and

$$C_\delta = \{z \in \mathbf{C} | z = k\delta + j\delta i\colon k, j \in \mathbf{Z}\} \quad (\delta > 0). \tag{2.3}$$

If there exists an element b_{k_0} in the sequence $\{b_k\}$ such that $|b_{k_0}| > K$, then the algorithm M cannot continue to run because of overflow.

In order to study the overflow we introduce the class of polynomials

$$\mathbf{P}_M(a, K, K^*) = \{p_0(z) \in \mathbf{P}(a, K^*) \mid \{Mp_0\} \subset S[0, K], |\{Mp_0\}| = \infty\}, \tag{2.4}$$

where

$$\mathbf{P}(a, K^*) = \{p_0(z) \in \mathbf{P}_n \mid 0 < |z_j| \leqslant a \ (j = 1, \ldots, n), \|p_0(z)\| \leqslant K^*\} \tag{2.5}$$

and

$$\|p_0(z)\| = \max_j |a_{j0}|. \tag{2.6}$$

Here $|\{Mp_0\}|$ denotes the cardinality of $\{b_k\}$, and z_j is the jth zero of $p_0(z)$.

The set $\mathbf{P}_M(a, K, K^*)$ represents the class of all polynomials which can be solved by M in a bounded set.

The following statements are valid.

THEOREM 2.1. *The set* $\mathbf{P}_T(a, K, K^*)$ *defined by Algorithm* 1 *is empty for every* $a, K, K^* > 0$.

Proof. If the roots of $p_0(z)$ are arranged so that

$$|z_1| \geqslant |z_2| \geqslant \ldots \geqslant |z_n|, \tag{2.7}$$

then the estimate

$$5^{-\mu_0} \leqslant \frac{|z_n|}{M[p_0(z), m_0]} \leqslant 1 \tag{2.8}$$

is valid (see [5]–[6]). For this reason the convergence of Algorithm 1 is identical with

$$|z_n^{(d)}| \leqslant cq^d \qquad (c > 0,\ 0 < q < 1), \tag{2.9}$$

where $z_n^{(d)}$ is the zero of $p_0(z + S^{(d)})$, $(d = 0, 1, \ldots)$ of minimal absolute value. Using the inequality (2.8), we have

$$\frac{n}{5c'}\left(\frac{1}{q}\right)^{d/\mu_0} \leqslant \frac{n}{5|z_n^{(d)}|^{1/\mu_0}} \leqslant |\sigma_{k(d)}^{(d)}| \qquad (d \geqslant d') \tag{2.10}$$

where $k(d) \in \{1, \ldots, n\}$ is the index of the maximal element in (1.3) and

$$\frac{n}{5c'}\left(\frac{1}{q}\right)^{d'/\mu_0} > 1. \tag{2.11}$$

Since $|\sigma_{k(d)}^{(d)}| = O(w^d)$, where $w = (1/q)^{1/\mu_0}$, therefore for a large index d_0

$$|\sigma_{k(d)}^{(d)}| > K \qquad (d \geqslant d_0) \tag{2.12}$$

is satisfied. Thus the theorem is proved.

THEOREM 2.2. *If* $K \geqslant K^* 2^{n+1}(1 + a^n 2^n)^{n+1} + 1$, *then*

$$\mathbf{P}_L(a, K, K^*) = \mathbf{P}(a, K^*) \tag{2.13}$$

is satisfied for Algorithm 2.

Proof. It is easy to see that the quantities recurring in the algorithm satisfy the inequalities

$$|p_0(\beta_j^{(d)})| \leqslant \begin{cases} \|p_0\| 2^{n+1} & (a < 0.5), \\ \|p_0\|(1 + 2^n a^n)^{n+1} & (a \geqslant 0.5), \end{cases} \tag{2.14}$$

$$\| T^j[p_0(z)] \| \leqslant ½(2\|p_0(z)\|)^{2^j} \qquad (j = 1, \ldots, n) \tag{2.15}$$

and

$$\|p_0(\alpha_j^{(d)}z + \beta_j^{(d)})\| \leqslant \|p_0(z)\|(2 + 2^{n+1}a^n)^n \qquad (j = 0, 1, \ldots, 8), \tag{2.16}$$

for $d = 0, 1, \ldots$. With the notation

$$\delta = \|p_0(z)\|(2 + 2^{n+1}a^n)^n,$$

and by using (2.15)–(2.16), we have

$$\|T^k[p_0(\alpha_j^{(d)}z + \beta_j^{(d)})]\| \leqslant \tfrac{1}{2}(2\delta)^{2^k} \qquad (k = 1, \ldots, n). \tag{2.17}$$

Since K is greater than the right side of (2.14) and (2.16), using $\psi > 2\delta$ we can get $\widetilde{\delta} < 0.5$ which proves the theorem.

The difference between Algorithms 1 and 2 is caused by the fact that Algorithm 1 is based on the inequality (2.8) while Algorithm 2 is based on the characteristic function $N[p_0(z)]$ which is invariant for the mapping $p_0(z) \to p_0(z)/\psi$, $(\psi > 0)$.

We remark that Algorithm 1 modified by the mappings

$$p_0(z) \to p_0(z)/\psi, \qquad p_0(z) \to p_0(z/\psi) \qquad (0 < \psi \leqslant K)$$

also has a $\mathbf{P}_T(a, K, K^*)$ empty for every $a, K, K^* > 0$.

3. The Study of Cost Functions. In the previous section it was proved that Algorithm 1 is unapplicable. Since an approximate solution with a given error $\epsilon > 0$ can be computed in the bounded set $S[0, \widetilde{K}]$, where $\widetilde{K}$ depends on $p_0(z)$, ϵ, and the method M, further analysis of the algorithms is necessary.

The cost function of the jth algorithm ($j = 1, 2$) is defined by the number of additions and multiplications per step and denoted by K_a^j and K_m^j.

Assuming that the computing time of the kth root can be characterized by three additions and three multiplications (which is a rough underestimate), the cost function of Algorithm 1 is

$$K_m^1 = (l + 1)(m_0 + 4)\frac{n^2}{2} + (l + 1)(m_0 + 8)\frac{n}{4} + O(1), \tag{3.1}$$

$$K_a^2 = (l + 1)(m_0 + 4)\frac{n^2}{4} + (2l + 3)n + O(1). \tag{3.2}$$

For the cost function of Algorithm 2 the inequalities

$$K_m^2 \leqslant 27n^2 - 18n, \tag{3.3}$$

$$K_a^2 \leqslant 9n^2 + 36n, \tag{3.4}$$

hold.

If we identify the bounds (3.3)–(3.4) with the cost of one step, then the speed of Algorithm 2 is

$$|z^{(d)} - z^*| \leqslant c_2(2/5)^d \qquad (d = 0, 1, \ldots). \tag{3.5}$$

The speed of Algorithm 1 is

$$|S^{(d)} - z^*| \leqslant c_1[q(\alpha_{m_0}, m_0, l)]^d \qquad (d = 0, 1, \ldots), \tag{3.6}$$

where

$$(3.7)\quad q(\alpha_{m_0}, m_0, l) = \left[1 + 0.25(1 + \alpha_{m_0})^2 - (1 + \alpha_{m_0})\cos\frac{\pi}{l+1}\right]^{1/2}\alpha_{m_0}^{-1}.$$

If $\delta = (m_0 + 4)(l + 1)/54 > 1$ and $n \geqslant n'$, then

$$(3.8)\quad K_m^1 \geqslant \delta K_m^2 \quad \text{and} \quad K_a^1 > \delta K_a^2.$$

THEOREM 3.1. *If $l \geqslant l'$, then*

$$(3.9)\quad q(\alpha_{m_0}, m_0, l) > (2/5)^\delta.$$

Proof. For a large l'

$$(3.10)\quad q(\alpha_{m_0}, m_0, l)^2 \geqslant \frac{1 - (\cos\pi/(l+1))^2}{\alpha_{m_0}^2} > \frac{9\alpha_{m_0}^{-2}}{(l+1)^2} \qquad (l \geqslant l')$$

and

$$(3.11)\quad (5/2)^\delta > l + 1.$$

From this fact the theorem immediately follows.

If $l \geqslant l'$, then the cost of d steps of Algorithm 1 gives $[\delta d]$ steps using the Lehmer-Schur method. By Theorem 3.1 we have

$$(3.12)\quad c^*[q(\alpha_{m_0}, m_0, l)]^d > (2/5)^{[\delta d]} \qquad (c^* > 0, d \geqslant d_0),$$

which proves that *the Lehmer-Schur process is faster than the Turán process.* For the parameters $m_0 = 4$, $\alpha_4 = 0.9$, $l = 11$, (see [5]–[6]) the relation (3.12) is also satisfied. This can be verified easily by (3.10) and (3.11).

In the paper [6] there is a reference to the infinite precision integer arithmetics [1] for the sake of application of Algorithm 1. It is known [1] that the computing time of the multiplication is at most

$$(3.13)\quad l(x)^{1+\tau} \qquad (1 \geqslant \tau > 0)$$

units of time ($l(x)$ denotes the length of x in the binary system). Since Algorithm 1 has to use numbers of length at least $2^{m_0-2}l(x)$ where $l(x)$ is needed by Algorithm 2, for the cost functions in the measure of computing time,

$$(3.14)\quad K_m^1(t) \geqslant (\delta 2^{m_0-2})^{1+\tau} K_m^2(t)$$

is satisfied. As a simple corollary, in (3.12) we can write $\delta 2^{m_0-2}$ instead of δ. This fact increases the relative convergence speed of the Lehmer-Schur process.

Department for Numerical Mathematics and Computing
Eötvös Loránd University
Budapest, Hungary

1. G. COLLINS, "Computer algebra of polynomials and rational functions," *Amer. Math. Monthly*, v. 80, 1973, pp. 725–755.

2. D. H. LEHMER, "A machine method for solving polynomial equations," *J. Assoc. Comput. Mach.*, v. 8, 1961, pp. 151–163.

3. A. RALSTON, *A First Course in Numerical Analysis,* McGraw-Hill, New York, 1965.

4. F. SZIDAROVSZKY, *Introduction to Numerical Methods* (in Hungarian), Közgazdásági és Jogi Könyvkiadó, Budapest, 1974.

5. P. TURÁN, "On the numerical solution of algebraic equations" (in Hungarian), MTA III, *Osztály Közleményei,* v. 18, 1968, pp. 223–235.

6. P. TURÁN, "Power sum method and the approximative solution of algebraic equations," *Math. Comp.,* v. 29, 1975, pp. 311–318.