MATHEMATICS OF COMPUTATION, VOLUME 29, NUMBER 129
JANUARY 1975, PAGES 311–318

Power Sum Method and the Approximative Solution of Algebraic Equations

By Paul Turán

To D. H. Lehmer on the occasion of his 70th birthday

Abstract. A universal algorithm is given for computing a root of an arbitrary polynomial with complex coefficients in a number of operations dependent only on the order n and relative error ϵ. This paper is a brief English exposition of the author's earlier "Algebrai egyenletek közelitö megoldásáról," Vol. XVIII of the *Communications of the* III *Class of the Hungarian Academy of Sciences* (1968), pp. 223–235.

1. This paper was partly prompted by the thought-provoking paper of Garrett Birkhoff entitled "Current trends in algebra" in the *Amer. Math. Monthly's* Aug.–Sept. 1973 issue. Without entering into a discussion of the paper as a whole, of its merits, or of the wide scope of its presentation, I would like to emphasize one point only. Sturm's and related theorems and more generally the numerical treatment of algebraic equations were still the subject of books of algebra *per definitionem* in 1928, in O. Haupt's *Algebra* and partly even in van der Waerden's book. But then, all of a sudden, King Lear's fate reached it; his daughters threw him out of his kingdom of algebra entirely; afterwards, in order not to be considered old fashioned, no respectable algebraist (except perhaps Krull) felt the need to behave even a bit like Cordelia. It seemed for a long time that the exile would be permanent. But as often happens new points of view emerged and resulted in a spiral-like escalation of the subject. So the exile now seems to be over. One can suspect that in time a palace or even a small county in the realm of algebra will be given back to him. Birkhoff, after mentioning some books of the new numerical algebra states,"... every forward-looking young algebraist should at least be cognizant of their contents." Such a sentence should heal all the wounds of Lear's soul caused by Regan and Goneril.

2. What I would actually like to discuss are the following lines of Birkhoff's paper (p. 776). "There are many other interesting new areas of research in (real and complex) numerical algebra. I shall just mention three of the most important; references to activity in them may be found in many review journals:

(a) Finding the roots of polynomial equations of degrees up to 100.

(b) "Unconstrained" minimization of functions of many variables.

Received July 12, 1974.
AMS (MOS) subject classifications (1970). Primary 65H05, 65H10.

(c) Linear programming and other techniques for finding minima of functions subjected to "constraints" by equations and inequalities."
Our discussion will deal exclusively with (a).

First of all it is puzzling in what sense (a) can be considered as a "new area of research" after D. Bernoulli. There are several possible interpretations. Keeping to one possible interpretation the problem can be considered as solved by the paper "A machine method for solving polynomial equations" by D. H. Lehmer published in *J. Assoc. Comput. Mach.* in 1961. To consider perhaps the widest theoretical interpretation one needs some preliminaries. By elementary operations on complex numbers we mean

$$\text{(2.1)} \qquad \begin{cases} \text{I. Addition, subtraction, multiplication, division,} \\ \text{II. The positive value of } k\text{th root of a positive number.} \end{cases}$$

Let an algorithm Π be a finite sequence of these elementary operations; the length of the algorithm Π is the number of elementary operations in Π. Let us consider the algebraic equation

$$\text{(2.2)} \qquad a_0 + a_1 z + \cdots + a_n z^n = 0$$

with arbitrary complex coefficients satisfying the normalization

$$\text{(2.3)} \qquad \min(|a_0|, |a_n|) = 1,$$

with its zeros being $z_1, z_2, \cdots, z_n$, where

$$\text{(2.4)} \qquad |z_1| \geqslant |z_2| \geqslant \cdots \geqslant |z_n|.$$

We shall say that the equation (2.2)–(2.3) "can be approximately completely solved" if for an arbitrarily small positive ϵ one can devise a method consisting of n algorithms $\Pi_1^*, \Pi_2^*, \cdots, \Pi_n^*$ which applied to the coefficients in (2.2) give values $z_1^*, z_2^*, \cdots, z_n^*$, respectively, so that

$$\text{(2.5)} \qquad |(z_j/z_j^*) - 1| \leqslant \epsilon, \qquad j = 1, 2, \cdots, n.$$

In principle these algorithms can depend not only on n and ϵ but also on the coefficients, and hence their lengths can be unbounded if ϵ is fixed and the coefficients a_ν are varying. If this is the case, the method is theoretically as well as practically less valuable. We say further that the equation (2.2)–(2.3) "can be approximately solved" if for an arbitrarily small positive ϵ one can devise *one* algorithm Π^{**} which applied to the coefficients in (2.2) gives a complex number α so that for *some* index j we have

$$\text{(2.6)} \qquad |(z_j/\alpha) - 1| \leqslant \epsilon.$$

Again Π^{**} can depend in principle on the coefficients and its length can be unbounded in the previous sense. The following further requirement on the Π's is important mainly for practical reasons. The equation to be solved is often given as a

characteristic equation of a matrix A. Hence, the further requirement is reasonable that the algorithms should be "elegantly rephrasable" (and reasonably programmable) for eigenvalues of matrices too. This requirement we can call requirement G. Theoretically as well as practically important is the problem of whether or not the Π-algorithms can be devised so that they do *not* depend on the coefficients a_ν and thus so that their lengths depend only on n and ϵ (even if requirement G is not fulfilled). The obvious practical significance of a universal solution of this problem is that one obtains a universal bound on machine time depending *only* on n and ϵ for *all* equations (2.2)–(2.3). The theoretical significance of the problem will be clear if we would suppose for a moment that–contrary to Ruffini-Abel's theorem–the general equation (2.2)–(2.3) is algebraically solvable for all n's. This would mean that one could devise an algorithm Π^{***} with length depending only upon n which applied to all equations (2.2)–(2.3) would give the exact value of a root of the equation. So a universal solution of (2.6) or (2.5) would mean briefly speaking that the analogue of Ruffini-Abel's theorem for the *approximative* solution of algebraic equations does not hold.

3. Whether or not the general equation (2.2)–(2.3) can be approximately completely solved in the sense (2.5) so that the Π_j^*-algorithms do not depend on the coefficients is according to the best of knowledge unknown. The matter is different however if we turn to approximate solvability in the sense (2.6). In my paper on approximative solution of algebraic equations ("Algebrai egyenletek közelitö megoldásáról," Vol. XVIII of the *Communications of the* III *Class of the Hungarian Academy of Sciences* (1968), pp. 223–235, in Hungarian) I not only proved the existence of the required Π^{**} algorithm with a length depending only on n and ϵ but described explicitly such algorithms which moreover also satisfy the G-requirement. Since my original paper is not easily accessible I shall describe here one of my algorithms *in extenso* and give (implicitly) its form for eigenvalues of matrices, in order not to make the paper too long. I remark also that there is an analogous theory for more general equations where the elements of the $n \times n$ matrix are polynomials in λ of lth degree at most.

4. I shall not enter here into the details of the proofs. Their basic idea consists in reducing them to some extremal problems concerning power sums of complex numbers which can be considered as a generalization of the theory of diophantine approximations. Such ideas have also proved to be very useful in function theory, in the theory of ordinary differential equations and in the treatment of various questions of the analytic number theory.* The problems relevant here refer to the determination or estimation of

*See a forthcoming English edition of my book *Eine neue Methode in Analysis und deren Anwendungen* in the Interscience Tracts series (with which I have been threatening the world for 15 years in various lectures and papers).

$$\min_{\xi_j} \max_{\nu=1,2,\cdots,n} |\xi_1^\nu + \xi_2^\nu + \cdots + \xi_n^\nu| \tag{4.1}$$

when the ξ_j-variables are normalized

$$\text{(a) by } \min_j |\xi_j| = 1 \tag{4.2}$$

resp.

$$\text{(b) by } \max_j |\xi_j| = 1. \tag{4.3}$$

Denoting the respective minimax by $\bar{M}_n$ resp. $\bar{\bar{M}}_n$, I found that

$$\bar{M}_n = 1 \tag{4.4}$$

(equality "essentially" only for the system

$$\xi_j = e^{2\pi ij/(n+1)}, \quad j = 1, 2, \cdots, n) \tag{4.5}$$

and

$$\bar{\bar{M}}_n > \log 2/\log(n+1), \tag{4.6}$$

which was improved by Atkinson to**

$$\bar{\bar{M}}_n > 1/6. \tag{4.7}$$

A further important inequality of this type was found by Buchholtz, who proved under the normalization (4.3) the inequality

$$\max_{\nu=1,2,\cdots,n} n^{-1/\nu}|\xi_1^\nu + \cdots + \xi_n^\nu|^{1/\nu} > 1/5. \tag{4.8}$$

5. Before turning to the discussion of the algorithm I shall make two remarks. First, that owing to the identity

$$\tfrac{1}{2}(c+d) + \sqrt{((c-d)/2)^2} = \max(c, d) \tag{5.1}$$

(with c, d real, and where the square root denotes the positive value), the operation $\max(d_1, d_2, \cdots, d_r)$ can be included in the domain of operations (2.1).***

The second remark refers to the adaptability of the algorithm to machines. Not being an expert in machines I could not embark on a proper estimation of the machine implementation; of course, it would be desirable if it were done. For the same reason I am sure that several details of the algorithm can be modified by a machine expert to make it more adaptable to a computer. A few observations on this will be made in Section 10.

6. We shall need what I call the first rule. Let

** The best possible value instead of 1/6 is not known.

*** And also $\min(d_1, d_2, \cdots, d_r)$.

$$a_0 + a_1 z + \cdots + a_n z^n \overset{\text{def}}{=} a_{00} + a_{10} z + \cdots + a_{n0} z^n \tag{6.1}$$

and we form the so-called Graeffe transforms

$$f_\nu(z) = a_{0\nu} + a_{1\nu} z + \cdots + a_{n\nu} z^n \quad (\nu = 0, 1, \cdots) \tag{6.2}$$

defined by

$$f_{\nu+1}(z) = f_\nu(\sqrt{z}) f_\nu(-\sqrt{z}). \tag{6.3}$$

Furthermore, let m be an arbitrary positive integer and with the mth Graeffe transform we calculate the numbers $\sigma_1, \sigma_2, \cdots, \sigma_n$ successively by the formulas

$$\begin{aligned} a_{0m}\sigma_1 + a_{1m} &= 0 \\ a_{0m}\sigma_2 + a_{1m}\sigma_1 + 2a_{2m} &= 0 \\ &\vdots \\ a_{0m}\sigma_n + a_{1m}\sigma_{n-1} + \cdots + n a_{nm} &= 0. \end{aligned} \tag{6.4}$$

Then we assert the

First Rule. With the notation (2.4) and

$$M = 1 \Big/ \max_{\nu=1,2,\cdots,n} \left| \frac{\sigma_\nu}{n} \right|^{1/(\nu \cdot 2^m)} \tag{6.5}$$

the inequality

$$5^{-1/2^m} \leqslant |z_n| / M \leqslant 1 \tag{6.6}$$

holds.

7. Let A be a nonsingular $n \times n$ matrix with complex entries and $\lambda_1, \lambda_2, \cdots, \lambda_n$ its eigenvalues, where

$$|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_n|. \tag{7.1}$$

If A^{-1} is the inverse matrix, we form by successive squarings the matrix

$$(A^{-1})^{2^m} \overset{\text{def}}{=} B.$$

Then we assert the

Second Rule. With the notation (7.1) and

$$N = 1 \Big/ \max_{\nu=1,2,\cdots,n} \left| \frac{1}{n} \operatorname{trace} B^\nu \right|^{1/(\nu \cdot 2^m)},$$

the inequality

$$5^{-1/2^m} \leqslant |\lambda_n| / N \leqslant 1$$

holds.

As one can see, the formulation of the second rule is even more elegant than that of the first (and is also easily programmable). This already indicates that the requirement G will be fulfilled by our algorithms. So it will be enough to give the algorithm for the equation (2.2) only.

Because of (5.1) the operations performed in these rules belong to the domain in (2.1).

8. Now the algorithm for (2.2), i.e., for (6.1), runs as follows.†

0th Step. We apply the first rule with $m = 4$. The M-value in (6.5) will be denoted by $M^{(0)}$ and we let

$$\xi^{(0)} \stackrel{\text{def}}{=} 0.$$

First Step. We consider the twelve numbers

$$\xi_j^{(1)} = \xi^{(0)} + \frac{19}{20} M^{(0)} e^{j\pi i/6}, \qquad j = 0, 1, 2, \cdots, 11.$$

If $f_0(\xi_j^{(1)}) = 0$ for some j we are finished. If not, we form the twelve polynomials

$$f_0(\xi_j^{(1)} + w) \quad (j = 0, 1, \cdots, 11), \tag{8.1}$$

rearrange them in the form (6.1) and apply the first rule with $m = 4$ to each. Then we get twelve numbers $M_j^{(1)}$ and one can determine an index μ_1 so that $\min_j M_j^{(1)} = M_{\mu_1}^{(1)}$. Let us abbreviate $M_{\mu_1}^{(1)} = M^{(1)}$ and define

$$\xi_{\mu_1}^{(1)} = \xi^{(1)}.$$

Second Step. We consider the twelve numbers

$$\xi_j^{(2)} = \xi^{(1)} + \frac{19}{20} M^{(1)} e^{j\pi i/6}, \quad j = 0, 1, \cdots, 11.$$

If $f_0(\xi_j^{(2)}) = 0$ for some j, we are finished. If not, we form the twelve polynomials

$$f_0(\xi_j^{(2)} + w) \quad (j = 0, 1, \cdots, 11), \tag{8.2}$$

rearrange them in the form (6.1) and apply the first rule with $m = 4$ to each. Then we get twelve numbers $M_j^{(2)}$ and one can determine an index μ_2 so that $\min_j M_j^{(2)} = M_{\mu_2}^{(2)}$. As before, we abbreviate $M_{\mu_2}^{(2)} = M^{(2)}$ and define

$$\xi_{\mu_2}^{(2)} = \xi^{(2)}.$$

Then the second step is finished.

These steps can be continued and will terminate after finitely many steps only in a zero of $f_0(z)$. At every step we have to form twelve numbers and one polynomial for each, apply the first rule to each with $m = 4$ and thus get twelve new M-numbers; we then have to select from them a minimal one. One can then prove that the sequence $\xi^{(d)}$ is such that for *each* $d \geqslant 2$ there is a zero $z^{(d)}$ of our equation, so that the inequality

† Our algorithm shows some formal similarities to the previously mentioned algorithm of D. H. Lehmer, l.c. (which is, as far as I can see, an unbounded one and I do not see whether or not the requirement G in 2. is fulfilled for it).

$$\left|\frac{z^{(d)}}{\xi^{(d)}} - 1\right| < 2\left(\frac{9}{28}\right)^d \tag{8.3}$$

holds.

9. The first and second rules have, of course, analogues for $|z_1|$ resp. $|\lambda_1|$. Although they are not necessary for our algorithms, they are obviously of independent importance, theoretically as well as practically; in order to emphasize the extent to which the G-requirement is fulfilled by our methods, I shall write down only the analogue of the second rule as the

Third Rule. Let m be an arbitrary positive integer and form by successive squaring the matrix $A^{2^m} \overset{\text{def}}{=} C$. Putting

$$L = \max_{\nu=1,\cdots,n} \left|\frac{1}{n} \operatorname{trace} C^{\nu}\right|^{1/(\nu \cdot 2^m)},$$

the inequality $1 \leqslant |\lambda_1|/L \leqslant 5^{1/2^m}$ holds.

10. It is clear that the length of our algorithm indeed depends only upon n and ϵ; also that choosing the parameters of the algorithm properly (regular 12-gon, etc.), it can be adapted to various optimizations. I have chosen the regular 12-gon only to make rearrangements in (8.1), (8.2) easier. As (8.3) shows, in order to attain 7% relative exactitude, $d = 3$ can be chosen, i.e., three steps are already enough, independent of n! This indicates intuitively that in performing the algorithm the machine time depends *not so much on* n but more on the prescribed relative exactitude required.

One can recognize that basically it deals with a reorganization of Graeffe's method which however is an essential one. It is rather remarkable since for the original form

(a) it is false without requiring $|z_1| > |z_2|$;

(b) no algorithms are known at present for deciding whether or not $|z_1| > |z_2|$ is fulfilled;

(c) if $|z_1| > |z_2|$ is fulfilled, then the algorithm is still "unbounded".

To be a bit personal, I often argued with numerical analysts on the practicality of my algorithm. Their main objection was—which I could never believe but I did not have the means to object—that very large numbers necessarily occur, which soon "overflow". So after a while I considered the whole matter as a nice pure mathematical theorem with some applied flavour. Now on the second page of the very interesting paper of G. E. Collins entitled "Computer algebra of polynomials and rational functions" in the same issue of the *Amer. Math. Monthly* one can read the following lines: "...Throughout we make provisions for polynomials with arbitrarily large coefficients. This is natural since computers are now fast enough that any restrictions imposed by the word length of the computer are artificial and

unnecessary. It is also important since only the most trivial algebraic operations on polynomials can be performed without generating integer coefficients which are 100 or more decimal digits in length. Frequently the final result will have coefficients of modest size, but obtaining this result requires the generation of polynomials with very much larger coefficients..." This is exactly the present situation too. The proper conclusions are left for the reader to draw.

Mathematical Institute
Hungarian Academy of Sciences
Budapest, Hungary