Finding Zeros of a Polynomial By the Q-D Algorithm

P. HENRICI
University of California, Los Angeles, and Eidgenössische Technische Hochschule, Zurich, Switzerland*

BRUCE O. WATKINS
Utah State University,† Logan, Utah

A method which finds simultaneously all the zeros of a polynomial, developed by H. Rutishauser, has been tested on a number of polynomials with real coefficients. This slowly converging method (the Quotient-Difference (Q-D) algorithm) provides starting values for a Newton or a Bairstow algorithm for more rapid convergence.

Necessary and sufficient conditions for the existence of the Q-D scheme are not completely known; however, failure may occur when zeros have equal, or nearly equal magnitudes. Success was achieved, in most of the cases tried, with the failures usually traceable to the equal magnitude difficulty. In some cases, computer roundoff may result in errors which spoil the scheme. Even if the Q-D algorithm does not give all the zeros, it will usually find a majority of them.

1. Introduction

Many well-known methods for determining the zeros of a polynomial (such as the Muller method [1], Laguerre's method [2], the Newton method or its variant for pairs of complex conjugate zeros, the Bairstow method [3]) converge very rapidly if good first approximations to the zeros are known. If no such approximations are known, the convergence may be much slower or, in certain cases, may not take place at all [4]. Furthermore, the methods mentioned furnish only one zero at a time; as soon as a zero is found, it is divided out, and the method is started afresh with a new random start.

Other methods (such as the ones ascribed to Bernoulli and Graeffe) do not require any first approximations, but then the convergence is usually slower and again only one (or two) zeros are found at a time.

In this note we relate some experimental results covering a technique for finding zeros based on the Quotient-Difference (Q-D) algorithm. This method was introduced by H. Rutishauser [5], and its theoretical aspects have since been discussed in a number of papers [6, 7, 8, 9]. Unlike any of the methods mentioned above, the Q-D algorithm has the advantage of furnishing simultaneous approximations to all zeros of a given polynomial, using no information other than the coefficients of the given polynomial. Theoretically the method could be used to determine the zeros of a polynomial with arbitrary accuracy; however, since the method converges at least as slowly as the Bernoulli method, and since it is subject to propagation of rounding error, we use the Q-D method only to determine first approximations to the zeros, which then are determined more accurately by the Newton or the Newton-Bairstow procedure.

The present paper is concerned solely with the computational aspects of the algorithm. For the theory we infer to the references given previously.

2. The Algorithm

Let

$$p(z) = a_0 z^N + a_1 z^{N-1} + \cdots + a_N \qquad (1)$$

be a polynomial whose coefficients (real or complex) are all different from zero. The Quotient-Difference Algorithm consists in constructing a two-dimensional array of numbers, whose elements are arranged as indicated below[1] in the special case $N=4$:

$$\begin{array}{ccccccccc} & q_0^{(1)} & & q_0^{(2)} & & q_0^{(3)} & & q_0^{(4)} & \\ e_0^{(0)} & & e_0^{(1)} & & e_0^{(2} & & e_0^{(3)} & & e_0^{(4)} \\ & q_1^{(1)} & & q_1^{(2} & & q_1^{(3)} & & q_1^{(4)} & \\ e_1^{(0)} & & e_1^{(1)} & & e_1^{(2)} & & e_1^{(3)} & & e_1^{(4)} \end{array}$$

etc.

The preparation of this paper was sponsored in part by the Office of Naval Research and the U.S. Army Research Office (Durham). Reproduction in whole or in part is permitted for any purpose of the United States Government. Part of the computing time was supplied by the Western Data Processing Center, UCLA, Los Angeles. Zeros were checked by a double precision program developed by General Motors Corp., Allison Div., Indianapolis, Indiana.

* Numerical Analysis Research.

† Department of Electrical Engineering.

[1] Our notation is adapted to the situation on hand; it does not agree with the standard notation in [3] or [5].

The entries in this scheme are connected by the relations

$$q_{n+1}^{(k)} = q_n^{(k)} + (e_n^{(k)} - e_n^{(k-1)}) \quad (k = 1, 2, \cdots, N), \tag{2}$$

$$e_{n+1}^{(k)} = e_n^{(k)} \frac{q_{n+1}^{(k+1)}}{q_{n+1}^{(k)}} \quad (k = 1, 2, \cdots, N-1), \tag{3}$$

where $n = 0, 1, 2, \cdots$.

Since each of these relations involves four adjacent entries of the scheme, they can be used to generate the scheme row after row as soon as the first two rows and the first and last column of the scheme are known. The first two rows are determined from the coefficients of the polynomial, as follows:

$$q_0^{(1)} = -\frac{a_1}{a_0} \qquad (q_0^{(k)} = 0) \qquad (k = 2, 3, \cdots, N), \tag{4}$$

$$e_0^{(k)} = \frac{a_{k+1}}{a_k} \quad (k = 1, 2, \cdots, N-1). \tag{5}$$

(Here we use the assumption that the coefficients of $p(z)$ do not vanish.) The first and the last columns of the method consist entirely of zeros:

$$e_n^{(0)} = e_n^{(N)} = 0 \quad (n = 0, 1, 2, \cdots). \tag{6}$$

Thus for instance for the polynomial

$$p(z) = 128z^4 - 256z^3 + 160z^2 - 32z + 1,$$

the first six rows of the scheme are as follows:

e^0	$q_1^{(1)}$	$e^{(1)}$	$q^{(2)}$	$e^{(2)}$	$q^{(3)}$	$e^{(3)}$	$q^{(4)}$	$e^{(4)}$
	2.000000		0.000000		0.000000		0.000000	
0		−0.625000		−0.200000		−0.031250		0
	1.375000		0.425000		0.168750		0.031250	
0		−0.193182		−0.079412		−0.005787		0
	1.181818		0.538770		0.242375		0.037037	
0		−0.088068		−0.035725		−0.000884		0

(n rows)

The q_n^k converge to the zeros of the polynomial as n increases. To five significant figures the zeros for this case are 0.96194, 0.69134, 0.30866, 0.03806, and are obtained with n about 15.

3. Convergence Properties

Clearly, the method as described above may fail to exist, since one of the divisors $q_{n+1}^{(k)}$ in (3) may turn out to be zero. Necessary and sufficient conditions for the existence of the scheme can be given [3, 5], but they are too involved to be of much use in practice. Suffice it to say that the method, to the extent to which it is necessary to compute it, existed in the great majority of cases which have been tried.

If the scheme exists, it has some remarkable convergence properties. We number the zeros of p in descending magnitude.

$$|z_1| \geq |z_2| \geq \cdots \geq |z_N|$$

and set $|z_0| = \infty$, $|z_{N+1}| = 0$ for convenience. The following convergence theorems then hold.

(i) *For every index k such that $|z_{k-1}| > |z_k| > |z_{k+1}|$,*

$$\lim_{k\to\infty} q_n^{(k)} = z_k. \tag{7}$$

Thus, if z_k is the only zero of modulus $|z_k|$, it can be found as the limit of the k-th q-column. Thus in particular, if the moduli of all zeros are distinct (which happens, for instance, if all zeros are real, positive, and simple), then the limits of all q-columns exist and are just the zeros, in descending order.

(ii) *For every k such that $|z_k| > |z_{k+1}|$,*

$$\lim_{k\to\infty} e_n^{(k)} = 0.$$

Thus, if one or several e-columns do not converge to zero, this is an indication that several zeros have the same modulus. Most frequently this occurs when a real polynomial has a pair of complex conjugate zeros. Such zeros, too, can be recovered from the scheme, as follows:

(iii) *For every k such that*

$$|z_{k-1}| > |z_k| = |z_{k+1}| > |z_{k+2}|,$$

the limits

$$\lim_{n\to\infty} [q_n^{(k)} + q_n^{(k+1)}] = A_k \tag{8a}$$

$$\lim_{n\to\infty} q_{n-1}^{(k)} q_n^{(k+1)} = B_k \tag{8b}$$

exist and have the values $z_k + z_{k+1}$ and $z_k z_{k+1}$, respectively; thus $z^2 - A_k z + B_k$ is the factor of $p(z)$ with the zeros z_k and z_{k+1}.

Relations similar to (8) also exist when there are more than two zeros of equal modulus, but they are more complicated [3].

As computational checks, two relations may be used:

$$q_n^{(1)} + q_n^{(2)} + \cdots + q_n^{(N)} = -\frac{a_{N-1}}{a_N} \quad (n = 0, 1, 2\cdots) \tag{9a}$$

$$q_{n+1}^{(1)} q_{n+2}^{(2)} \cdots q_{n+N}^{(N)} = (-1)^N \frac{a_0}{a_n} \quad (n = 0, 1, 2\cdots). \tag{9b}$$

4. Experimental Results

As indicated above, the convergence of the Q-D scheme is only linear and therefore slow. Furthermore, since the scheme is not self-checking, it is subject to propagation of rounding error. For these reasons it would be unwise to rely solely on the Q-D scheme for the determination of the zeros. Following a suggestion [3], we have written a routine which uses the scheme to produce good starting values for the Newton or Bairstow methods, which are applied to the complete (undisturbed) polynomial.

It is of interest to compare this procedure with the very elaborate programs described in [4]. There the zeros (or

Problem	Degree	Coefficient[1]	Zeros or Factors[2]	Number of Zeros Found[3]
TABLE 1. Polynomial Coefficients, Zeros and Q-D Program Results				
1	3	1, −1, 2, 5	$-1.1325,\ 1.0662 \pm i1.8106$	3
2	3	1, 0, 3, 1	$-0.32219,\ 1.6109 \pm i1.7544$	3
3	3*	1, 1.0004, −1.0002, −1.0006	−1.00000, 1.00010, 1.00050	3
4	3*	1, 3.00006, 3.000120, 1.00006	−1.00000, −1.00001, 1.00005	3
5	3	1, 0, −3, 1	−1.8794, 1.5321, 0.34730	3
6	3*	1, 8, 21, 18	−3, −3, −2	3
7	4	128, −256, 160, −32, 1	0.96194, 0.69134, 0.30866, 0.03806	4
8	4	1, −8, 39, −62, 50	$3 \pm i4,\ 1 \pm i1$	4
9	4*	1, 7, 13, −3, −18	1, −2, −3, −3	4
10	4*	1, −6, 9, 4, −12	2, 2, 3, −1	4
11	4*	1, −5, 6, 4, −8	2, 2, 2, −1	4
12	4*	1, −8, 24, −32, 16	2, 2, 2, 2	4
13	4	1, −3, −14, 52, −48	$4, -4.1273, 1.5637 \pm i0.68002$	4
14	4*	1, −3, −12, 52, −48	2, 2, 3, −4	4
15	4*	1, 3.05, 4.101, 3.10205, 1.05105	$z^2 + z + 1.001,\ z + 1,\ z + 1.05$	4
16	4	1, −3, 0, 52, −48	$-3.2635,\ 0.95756,\ 2.6530 \pm i2.8848$	4
17	4	1, −3, −14, 0, −48	$5.7098,\ -3.1068,\ 0.19852 \pm i1.6329$	4
18	5	1, −6, 14, −16, −7, −30	$3.682,\ -0.5 \pm i0.868,\ 1.65 \pm i2.23$	5
19	6	1, 0, 0, 0, 0, 1	$1, -1, 0.33333 \pm i0.86667, -0.33333 \pm i0.86667$	6
20	6*	1, −9, 45, 85, 34, 74, −100	$-1, 2, 3 \pm i4, 1 \pm i1$	6
21	6*	1, −7, 25, 25, −246, +422, −300	$2, -3, 3 \pm i4, 1 \pm i1$	6
22	6*	1, −13, 85, −305, 594, −622, 300	$2, 3, 3 \pm i4, 1 \pm i1$	6
23	6*	1, −4, −5, 190, −666, 944, −600	$2, -6, 3 \pm i4, 1 \pm i1$	6
24	6*	1, −11, 65, −195, 314, −274, 100	$2, 1, 3 \pm i4, 1 \pm i1$	6
25	7*	1, 4.870, −0.670, −0.15430003, −0.4265, −1.02113, −2.48608, −6.2771496	$z^2 + z + 1.01,\ z^2 - z + 1.10,\ z + 1.0,\ z - 1.13,\ z + 5$	7
26	7*	1, 507.9, −2079.899 9865.5087, −8504.8717, −2072.1159, 9864.8527, −9020.211	$z^2 - 3z + 16,\ z^2 - z + 1.001,\ z + 1,\ z - 1.1,\ z + 512$	7
27	10*	1, −2.5, −460.8, −9133.4, −50761.8, −88653.098, −53510.4, −37313, −197170, −364800, −198000	$z^2 - 2z + 2,\ z^2 + 2z + 2.2,\ z^2 + 20z + 200,\ z + 1,\ z + 1.5,\ z + 5,\ z - 3$	10
28	12*	1, 1288, 271787, 1337920, −1.5021189E8, 1.2664523E9, −5.2810183E9, −1.9582773E10, 9.621956E9, 1.9383320E10, −3.1504466E11, −8.6167780E11, −5.4975580E11	$z^2 + 3z + 4,\ z^2 - 4z + 8,\ z^2 - 10z + 64, z + 1,\ z + 2,\ z - 16,\ z + 32,\ z + 256,\ z + 1024$	12
29	13*	1, 128901, 273087, 1612424.9, −1.488606E8, 1.1147382E9, −4.0019013E9, −2.49166E10, −1.0156643E10, 2.9101496E10, −2.954675E11, −1.1798729E12, −1.4200503E12, −5.5525336E11	$z^2 + 3z + 4,\ z^2 - 4z + 8,\ z^2 - 10z + 64,\ z + 1,\ z + 2,\ z - 16,\ z + 32,\ z + 256,\ z + 1024,\ z + 1.01$	13
30	15*	1, 2, 334, −592, 36352, 60716, −1486310, −2191720, 23210431, 30731586, −169919420, −134375270, 1.634846E9, 2.1045721E9, −5.7149106E9, −6.2270208E9	$z^2 + 4z + 8,\ z^2 - 4z + 9,\ z + 1,\ z - 2,\ z + 3,\ z + 4,\ z - 5,\ z + 6,\ z + 7,\ z + 10,\ z - 11,\ z - 12,\ z + 13$	15
31	18	1, 5, 10, 9, 50, 40, 30, 105, 98, 15, 81, 72, 60, 48, 36, 25, 130, 100, 520	$-2.9028 \pm i1.8927,\ 0.73184 \pm i1.2325,\ 0.50380 \pm i1.6390,\ -1.2458 \pm i0.31540,\ 0.89442 \pm i0.88178,\ -1.0386 \pm i0.67193,\ 0.05192 \pm i1.1350,\ -0.51543 \pm i0.96378,\ 1.0210 \pm i0.35144$	16
32	19	1, 5, 10, 9, 50, 40, 30, 105, 98, 15, 81, 72, 60, 48, 36, 25, 130, 100, 520, 672	$-1.0507,\ -2.9028 \pm i1.8927,\ 1.0770 \pm i0.32915,\ -0.41890 \pm i1.0476,\ 0.50824 \pm i1.6396,\ 0.95083 \pm i0.85507,\ 0.76888 \pm i1.2373,\ -1.1915 \pm i0.41270,\ 0.16098 \pm i1.1971,\ -0.92737 \pm i0.72873$	15
TABLE 2. Random Polynomial Coefficients of 5 Digits, Zeros and Q-D Program Results				
1	6	53.081, −1429.7, 0.07721, −2.0143, 56.131, −561.12, 23.451	$0.54839 \pm i0.54964,\ 0.04197,\ -0.56935 \pm i0.56935,$ 26.934	6
2	6	−2.9041, −6.9933, −8438.8, 23.804, −47.947, 36.539, 0.68540	$0.15886,\ -0.018289,\ -0.068874 \pm i0.15235,$ $-1.2054 \pm i53.892$	6
3	7	−48.773, 3.8121, −74.419, −116.96, 4.2614, −9915.8, −7.1330, −4.3250	$-0.00035978 \pm i0.020882,\ -0.79461 \pm i2.8190,$ $2.2513 \pm i1.8147,\ -2.8345$	7

(continued)

Problem	Degree	Coefficients[1]	Zeros or Factors[2]	Number of Zeros Found[3]
4	7	3.8121, −74.419, −116.96, 4.2614, −9915.8, −0.07133, −0.04325, −4.3936	0.038108 ± *i*0.066060, 0.076218, 1.7322 ± *i*4.5447, −5.1782, 21.236	7
5	8	1, 2.9742, −6.7676, 6.2608, −54.215, −97.167, 7.0080, 7.7130, −15.806	−4.90436, 2.77582, −1.18832, −0.690732, 0.198214 ± *i*2.37181, 0.318483 ± *i*0.385059	8
6	8	4.3936, −8.3619, 7.7182, −5545.9, −288.08, 3803.4, 105.40, −54.646, 8191.9	1.2106, 0.35283 ± *i*0.91569, −0.98342 ± *i*0.54629, 11.420, −4.7333 ± *i*9.3942	8
7	8	7.7182, −5545.9, −288.08, 3803.4, 105.40, −54.646, 8191.9, 8.7029, −167.00	0.14221, 1.2085, −0.14332, 0.35080 ± *i*0.92018, −0.97998 ± *i*0.54791, 718.60	8
8	9	1, −993.56, −21.125, 122.20, 2.6288, −7.7351, 7.2159, −2184.9, 61.422, −31.350	0.014036 ± *i*0.11897, 0.99662 ± *i*0.56000, −0.0080556 ± *i*1.1198, −1.0132 ± *i*0.56143, 993.58	9
9	9	8.7029, −167.00, 463.33, 1126.1, 76.241, −7.0508, −4085.4, −1036.1, −99.729, −54.649	−0.31157, 0.029598 ± *i*0.20480, 1.4716, −0.39604 ± *i*1.3425, −1.9311, 5.6018, 15.091	9
10	9	463.33, 1126.1, 76.241, −7.0508, −4085.4, −1036.1, −99.729, −54.649, −2.9041, −6.9933	0.16456 ± *i*0.18925, −0.36540, −0.10821 ± *i*0.25013, 0.13911, −0.40776 ± *i*1.4637, −2.7533	9
11	15	1, 39.247, −20.573, −8.3243, 22.834, −0.78440, −4.2754, 504.15, −21.134, 72.874, 2.9240, −94.501, 5.5945, 4.0532, 2549.3, 21.129	−0.0082883, −1.2152, 1.1806 ± *i*0.53560, 0.36708 ± *i*1.2673, −0.70578 ± *i*0.96858, 0.022019 ± *i*1.4604, 1.3624 ± *i*0.76961, −1.3588 ± *i*0.74633, −39.759	15
12	18	1, −2.6027, 693.57, −6.5842, 1.6240, −6113.5, 21.652, 4.5896, −0.71410, −7.8138, 8.3005, 37.504, −64.678, 4.9990, −6.8560, 45.808, 337.62, −14.910, −6093.8	0.74344 ± *i*0.66073, −0.98881, 0.97248 ± *i*0.24402, −0.58013 ± *i*0.82796, −0.11371 ± *i*0.99865, 0.35904 ± *i*0.92990, −0.88517 ± *i*0.45229, 2.0683, −1.0373 ± *i*1.7847, 1.3029 ± *i*26.304	12
13	36	−9265.3, 6468.0, −42.015, 70.311, 3072.4, 2.9530, 5.6163, 870.73, −7.9141, −74.110, −22.964, 9.2252, −2.4987, −39.063, 6.5810, −6.8461, −7.8867, −32.151, −34.637, 67.916, −390.57, 60.247, 265.74, −453.86, −7015.6, −309.67, −2.0574, −85.581, −99.394, −20.775, 49.225, 3924.5, −0.083830, 73.941, 0.049060, 88.312, −993.56	0.81276 ± *i*0.061480, 0.58720 ± *i*0.48197, 0.22083 ± *i*0.700000, −0.88460 ± *i*0.30152, 0.63308 ± *i*0.69314, −0.20948 ± *i*0.89996, 1.0530 ± *i*0.087703, −0.83729 ± *i*0.39630, −0.072582 ± *i*0.99088, 0.57888 ± *i*0.77201, −0.583111 ± *i*0.78522, 0.15853 ± *i*1.0003, 0.82646 ± *i*0.57735, −0.98285 ± *i*0.11248, 0.39262 ± *i*0.92500, 0.95385 ± *i*0.36071, −0.74665 ± *i*0.60412, −0.37723 ± 0.89807	22

* Polynomials made up from selected zeros or factors. Otherwise polynomials made up from selected coefficients.

[1] Coefficients are listed in order of descending powers of the polynomial.

[2] Zeros and coefficients are at least six significant figures for * polynomials. Otherwise these values are to five significant figures. Factors are indicated in the form $(z+1)$, etc., otherwise the values given are zeros.

[3] See discussion on equal zeros.

pairs of zeros) are found by the Newton-Bairstow technique using unsophisticated first guesses, and each zero (or each quadratic factor) is divided out as soon as found. As a final step, when all the zeros have been found, the original polynomial is used to produce improved zeros by iteration. It would seem that, considered as a method for finding first approximations, the Q-D scheme is more economical.

Several methods for checking the convergence of the $e^{(k)}$-columns in the Q-D method were tried. Two successful test expressions, either of which may be used, follow:

$$(2 \mid e_n^{(k)} \mid /(\mid q_n^{(k)} + q_n^{(k+1)} \mid + \mid q_{n-1}^{(k)} + q_{n-1}^{(k+1)} \mid)) - \text{EPS} \mid e_n^{(k)} \mid / \mid e_{25}^{(k)} \mid - \text{EPS}.$$

EPS must be chosen sufficiently small so that the test expression used will remain positive until the kth e-column has effectively converged. To further ensure such convergence, a count of five or more passages of the test set an indicator $I(k)$. This count was set back to zero on failure to pass the test if the count was less than three; otherwise the count was reduced by one on failure to pass the test. The number of iterations, n, was an input value. Experience showed that an EPS value of about 0.01 and an n value of about 150 gave satisfactory results for most polynomials.

Besides trying the method on a number of fairly simple polynomials (Table 1) a number of other polynomials were investigated (Table 2). These later polynomials were made up of coefficients formed by using random numbers of five digits from [10]. The decimal was placed by spinning a pointer with six possible stopping positions, two positions being used to move the decimal to the right one place, two positions being used to move the decimal to the right two places, one position three places, and one position four places. The sign was established by a coin flip. It is of interest to note that the magnitude of many of the zeros of these polynomials tend to cluster about the value one. For example, in one 18th degree polynomial, 13 zeros have magnitudes differing from one by less than 1.2 per

cent. Similar clustering was observed on most of the other polynomials.

Two principal difficulties were encountered in using the Q-D algorithm. The first consisted of obtaining a value of zero in relation (2) due to computer roundoff. The second consisted in the difficulty of separating roots of almost equal magnitude. Due to the clustering effect previously discussed, the second difficulty was more serious. The technique used to overcome both difficulties was to make a shift of the variable along the real axis. Due to the strong influence of coefficient variation on the zeros, as pointed out by Wilkinson [4], this shift was made in double precision arithmetic. A similar shift was used in case some coefficient was zero.

The technique used to obtain the results indicated in Table 2 was as follows:

(A) Q-D method (single precision) to find starting values for the zeros. From 100 to 500 iterations were used, depending on the polynomial degree.

(B) Newton-Bairstow method (single precision) to find more accurate values (always using the original coefficients).

(C) Check root count, and if root count not equal to the degree of the polynomial: (1) Reduce $p(z)$ by dividing out the zeros that have been located. (2) Shift the variable in the reduced polynomial on the real axis (see [3, p. 173]).

(D) Return to (A) with a reduced polynomial to find new starting values.

As can be seen from the tabular results on roots found, this method was quite successful for polynomials of ninth degree or lower. Shifting the variable greatly assisted in separating zeros close in magnitude, which otherwise could not have been located. The failures are traceable, we believe, to the fundamental problems of finding zeros in single precision arithmetic, as discussed by Wilkinson [4].

The largest magnitude zeros are produced first, and thence in order of descending magnitude. Hence, by varying the amount of the axis shift mentioned in C(2) above, different zeros will be output, and if two or three different shift values are used, all zeros may be obtained in even the most difficult cases, as the greatest computational errors occur in the later stages of the process outlined above. This feature might be automated, but this was not done in this study.

In Table 1, zeros of exactly equal magnitude were obtained in an earlier program which did not involve the axis shift feature. The later program used for Table 2 deliberately suppressed zeros previously found, since in such cases it could be assumed that the Newton process was incorrectly converging to a previously found zero. (This could be avoided if a reduced polynomial were used in the Newton-Bairstow scheme.) However, it was generally true that exactly equal zeros did not appear as equal in the results due to computer rounding. Hence, equal real roots, at least, will usually be found in any case, showing up as slightly unequal values. A fairly simple program for equal real zeros of orders three and four can be devised from reference [3]. However, in actual practice, exactly equal zeros are highly unlikely.

Another recently developed program proceeds as follows:

(A) Q-D method (single precision) to find starting values for the zeros.

(B) Newton-Bairstow method (double precision) to find more accurate values.

(C) In (B) reduce $p(z)$ by dividing out the zeros z_k as they are located, after a test of the remainder of the polynomial evaluated at $z = z_k$.

(D) Check zero count, and if count not equal to the degree of polynomial, shift the variable in the reduced polynomial on the real axis (double precision).

(E) Return to (A) with reduced polynomial.

This program found all the zeros in all problems shown in Table 2, thus demonstrating that the difficulty in the previously discussed program lay in the fact that the zeros were found in single precision arithmetic.

5. Conclusions

The Q-D algorithm appears to be a useful and powerful tool in obtaining the zeros of a polynomial with real coefficients since (1) all computations are with real numbers; (2) a complete set of starting values for rapidly converging methods are obtained at one time, rather than one value per iteration. The method of shifting the variable along the real axis solves the inherent problem of separating zeros of almost equal magnitude.

RECEIVED MARCH, 1965; REVISED JUNE, 1965

REFERENCES

1. MULLER, D. A method for solving algebraic equations using an automatic computer. *Math. Tab. Wash. 10* (1956), 208–215.
2. MAEHLY, HANS J. Zur iterativen auflosung algebraischer gleichungen. *Z. Angew. Math. und Physik. 5* (1954), 260–263.
3. HENRICI, P. *Elements of Numerical Analysis.* John Wiley and Sons, New York, 1964.
4. WILKINSON, J. H. The evaluation of the zeros of ill-conditioned polynomials III. *Numerische Mathememtik 1* (1959), 150–180.
5. RUTISHAUSER, H. Der quotienten-differenzen-algorithmus. *Z. Angew. Math. Physik. 5* (1954), 233–251.
6. ——. Anwendungen des quotienten-differenzen-algorithmus. *Z. Angew. Math. Physik. 5* (1954), 496–507.
7. ——. Bestimmung der eigenvektoren und eigenwerte einer matrix. *Z. Angew. Math. Physik. 6* (1955), 387–401.
8. ——. Eine formel von Wronski and ihre bedautung für den quotienten - differenzen - algorithmus. *Z. Angew. Math. Physik. 7* (1956), 164–165.
9. HENRICI, P. The quotient-difference algorithm. Nat. Bur. Standards Applied Mathematics Series 49, U. S. Government Printing Off., Washington, D. C., 1958, 23–46.
10. The RAND Corp. *A Million Random Digits.* Free Press, Chicago, 1955.