Mathematica Balkanica

Mathematical Society of South-Eastern Europe
A quarterly published by
the Bulgarian Academy of Sciences – National Committee for Mathematics

Mathematica Balkanica

New Series Vol.8, 1994, Fasc.4

Iteration Methods for Simultaneous Finding All Roots of Generalized Polynomial Equations

Khr. Semerdzhiev

Presented by Bl. Sendov

Iteration methods for simultaneous finding all roots of algebraic, trigonometric, exponential and generalized polynomials are developed. Two essential iteration formulae by the Davidenco's approach are derived. As particular cases, analogues of Newton's, Chebyshev's and Obreshkoff's methods are obtained. Using generalized divided differences with multiple knots, methods for simultaneous finding all roots with given multiplicities of generalized polynomials are developed. Convergence theorems and numerical examples for every method are supplied. Some explanation of the better behaviour of the methods for simultanious finding zeros of polynomials in comparison with the corresponding methods for individual zero search is given. An algorithm for the determination of the zero multiplicities of polynomials is presented.

1. Introduction

Many problems in mathematics, other natural sciences and technics reduce themselves to determining all roots of generalized polynomial equations. The problem of polynomial roots finding has been generally formulated in an elementary way, but for its solution one has to overcome a number of difficulties which essentially arise from its nonlinearity.

The polynomial equations roots can be searched individually (sequentially) i.e.one after the other. For this purpose a variety of numerical methods have been created. The iteration methods for simultaneous finding all roots (SFAR) of generalized polynomials are of particular interest in this respect, since these computational schemes have a wider region of convergence for it (the convergence) depends less upon the choice of the initial approximations. For example, the well-known method of Weierstrass-Iliev-Durand-Dochev-Kerner (WIDDK) has been tested for 4000 equations with randomly chosen initial approximations, this method beeng convergent except for 4 cases. However, after slight changes in the initial approximations the process becomes a convergent one.

This paper deals with a number of effective methods of SFAR of algebraic, trigonometric, exponential and generalized polynomial equations with simple or multiple roots. 16 computational schemes have been described here among which the WIDDK method and the method of Ehrlich only have been familiar earlier. The remaining methods have been constructed and investigated by us and are published as preliminary communications of JINR Dubna and papers in "Comptes rendus" of the Bulgarian Academy of Sciences. Other modifications of the method for SFAR of an algebraic polynomial (accelerated convergence, two-side approximations, etc.) have been adduced in the monograph [11].

Now we shall formulate the problem itself in its general case. Let the generalized polynomial (G - polynomial) $f(x)$ developed upon the Chebyshev system of basic functions $\{\varphi_k(x)\}_0^n$ be given and for it it is necessary to find all its zeros $\{x_k\}_1^n$. At this stage we consider only the case when the zeros of $f(x)$ are simple (real or complex). Let $\{x_k^{[0]}\}_1^n$ be initial approximations of the roots. Suppose can construct a G -polynomial $Q_{[0]}(x)$ upon the system considered whose zeros are $\{x_k^{[0]}\}_1^n$. We shall use Davidenco's approach [4]. Let us consider one-parameter family of polynomials P of the kind $P(x;t) = (1-t)Q_{[0]}(x) + tf(x)$, where $t \in [0,1]$. The zeros $\{x_i(t)\}_1^n$ of the polynomial P are functions of the parameter t. Differentiating the identities $P(x_i(t);t) \equiv 0$, $i = \overline{1,n}$ we obtain the system of ordinary differential equations

$$x_i'(t) = \frac{Q_{[0]}(x_i(t)) - f(x_i(t))}{(1-t)Q'_{[0]}(x_i(t)) + tf'(x_i(t))},\ i = \overline{1,n}. \tag{1}$$

This system has to be solved with initial approximations $x_i(0) = x_i^{[0]}$, $i = \overline{1,n}$. Evidently $x_i(1) = x_i$, $i = \overline{1,n}$. For calculating $x_i(1)$, $i = \overline{1,n}$, the system (1) can be integrated by using some of the numerical methods for solving the Cauchy problem for ordinary differential equations, moving along the parameter t from 0 to 1, keeping a certain step. Instead of that we shall find approximatively $x_i(1)$ by applying the approximative formula $x_i(1) = x_i(0) + x_i'(0)$ i.e.

$$x_i^{[1]} = x_i^{[0]} - f(x_i(0))/Q'_{[0]}(x_i(0)),\ i = \overline{1,n},$$

where $x_i^{[1]} \equiv x_i(1)$. In this way we obtain the iteration formula

$$x_i^{[k+1]} = x_i^{[k]} - f(x_i^{[k]})/Q'_{[k]}(x_i^{[k]}),\ i = \overline{1,n}, k = 0,1,..., \tag{2}$$

where $Q_{[k]}(x)$ is a polynomial upon the given Chebyshev system and has as roots $\{x_i^{[k]}\}_{i=1}^n$. The method (2) is a modification of the well-known method of Newton.

However, the more exact Taylor's formula $x_i(1) = x_i(0) + x_i'(0) + \frac{1}{2}x_i''(0)$ can be applied for integrating the system (1) as well, and, analogously to (2) we find the modification of the method of Chebyshev

$$x_i^{[k+1]} = x_i^{[k]} - f(x_i^{[k]})[2Q'_{[k]}(x_i^{[k]}) - f'(x_i^{[k]}) + f(x_i^{[k]}) \frac{Q''_{[k]}(x_i^{[k]})}{2\,Q'_{[k]}(x_i^{[k]})} / Q'^2_{[k]}(x_i^{[k]}), \; i = \overline{1,n}, \; k = 0,1,.... \tag{3}$$

In (2) and (3) $x_i^{[k]}$ denotes the k^{th} approximation (the k^{th} iteration) to the i^{th} root of $f(x)$.

Remark 1.1. Furthermore $T(0,k,r)$ denotes the following statement:If $|x_i^{[0]} - x_i| \leq cq$ at some c and q, then $|x_i^{[k]} - x_i| \leq cq^{r^k}$, $i = \overline{1,n}$, $k = 1,2,....$

2. Methods for SFAR of algebraic, trigonometric and exponential polynomials with quadratic rate of convergence.

Let us set in (2) $f(x)$ to be an algebraic polynomial (A- polynomial)

$$A_N(x) \equiv x^N + a_1x^{N-1} + ... + a_N \tag{4}$$

and let us choose $Q_{[k]}(x) = \prod_{i=1}^N (x - x_j^{[k]})$. Then from (2) we obtain the following method

$$x_i^{[k+1]} = x_i^{[k]} - A_N(x_i^{[k]}) / \prod_{j=1, j\neq i}^{N} (x_i^{[k]} - x_j^{[k]}), \tag{5}$$

$$i = \overline{1,n}, \; k = 0,1,....$$

This method has been developed by many authors. The historical survey has pointed out that in its essence the formula (5) has been used as early as Weierstrass (1901) used it in proving the essential theorem of algebra. Later Ilieff (1948) showed that using the information about all roots of equation $A_N(x) = 0$ one can enlarge the convergence region of Newton's iteration. This fact prompted to Dochev the greater possibilities of the method for SFAR and in [5,6,12] he elaborated and investigated the method (5) again. The method (5) has then appeared [7] simultaneously in an implicit form. This same method

(5) has been considered in [8]. Thus, the method (5) can with all good reason be named WIDDK. The quadratic convergence of (5) is established by the theorem:

Theorem 2.1 [2,32]. *Let* $0 < q < 1$, $d = \min_{i\neq j} |x_i - x_j|$ *and* c *be sufficiently small and such that* $d-2c > 0$ *and* $2^N c/(d-2c) < 1$. *Then* $T(0,k,2)$ *holds true.* ∎

Proposition 2.2. *Irrespective of the choice of initial approximations* $\{x_i^{[0]}\}_1^N$ *to the roots* $\{x_i\}_1^N$ *of (4), the successive approximations, obtained by (5) satisfy the correlation:*

$$\sum_{i=1}^{N} x_i^{[k+1]} = \sum_{i=1}^{N} x_i, \; k = 0, 1, ... \tag{6}$$

Proof. If we sum both sides of (5) with respect to i from 1 to N, then we obtain

$$\sum_{i=1}^{N} x_i^{[k+1]} = \sum_{i=1}^{N} x_i^{[k]} - D_A(x_1^{[k]}, ..., x_N^{[k]}; A_N(x)), \tag{7}$$

where

$$D_A(x_1^{[k]}, ..., x_N^{[k]}; A_N(x)) = \sum_{i=1}^{N} A_N(x_i^{[k]})) \,/ \prod_{j=1, j\neq i}^{N} (x_i^{[k]} - x_j^{[k]})$$

is the divided difference of an algebraic type [2] from $N-1$ order of the polynomial $A_N(x)$ in the knots $\{x_i^{[k]}\}_{i=1}^N$.

Keeping in mind the properties of D_A we have

$$D_A(x_1^{[k]}, ..., x_N^{[k]}; A_N(x)) = x_1^{[k]} + ... + x_N^{[k]} + \alpha_1. \tag{8}$$

On the other hand, from Villette's formulae $\sum_{i=1}^N x_i = -\alpha_1$ holds true. If we place the last expression from (8) in (7) we obtain (6). ∎

The Proposition 2.2 points out an important property, a characteristic pequliarity of method (5). Namely, that same method has advantages compared with the ordinary method of Newton

$$x_i^{[k+1]} = x_i^{[k]} - A_N(x_i^{[k]})/A'_N(x_i^{[k]}). \tag{9}$$

One of them consists of the following: In Newton's method such initial approximation $x_i^{[0]}$ can be chosen so that $x_i^{[1]} = \pm\infty$. Using (5), however, this can never occur, because of (6) for no initial approximation. The equality (6) can be used as a condition for stoping the iteration process (5) (stop condition).

Example 2.3. The equation

$$A_3(x) \equiv x^3 - 8x^2 - 23x + 30 = 0 \tag{10}$$

has as roots -3, 1 and 10. If we choose $x_1^{[0]} = -4$ $x_2^{[0]} = 2$ and $x_3^{[0]} = 9$, then by the method (5) we get $x_1^{[4]} = -3.(6*0)6$, $x_2^{[4]} = 1.(6*0)6$ and $x_3^{[4]} = 9.(8*9)6$.

Here and further on we use a shortened way of writing of the following kind: $-3.(6*0)6 \equiv -3.0000006$, $9.(8*9)6 \equiv 9.999999996$.

Let us get over to the case when $f(x) = T_N(x)$ is a trigonometric polynomial (T-polynomial). The basic system here is $(1, \sin x, \cos x, ..., \sin Nx, \cos Nx)$. Setting in (2) $f(x) = T_N(x)$,

$$Q_{[k]}(x) = B_k^{-1} \prod_{j=1}^{2N} \sin((x - x_j^{[k]})/2),$$

where $B_k = [T_N(y)]^{-1} \prod_{j=1}^{2N} \sin((y - x_j^{[k]})/2)$ and y is an arbitrary number from the interval $[-\pi, \pi]$ but different from the roots of $T_N(x)$, we obtain the following iteration formula for SFAR of $T_N(x)$, lying in the strip, bounded by the parallel lines $x = \pm\pi$:

$$x_i^{[k+1]} = x_i^{[k]} - 2B_k T_N(x_i^{[k]}) \Big/ \prod_{j=1, j\neq i}^{2N} \sin((x_i^{[k]} - x_j^{[k]})/2) \tag{11}$$

$$i = \overline{1, 2N}, \quad k = 0, 1, \ldots.$$

Theorem 2.4 [29]. *Let* $0 < q < 1$, $d = \min_{i\neq j} |x_i - x_j|$, $d_1 = \min |y - x_i|$, $r = \min\{2|\sin(d/2 - c)|, 2|\sin(d_1/2)|\}$ *and* $c > 0$ *is such that* $d - 2c > 0$ *and* $c2^{4N} < r$. *Then* $T(0, k, 2)$ *holds true.*

Example 2.5. For the T-polynomial $T_2(x) = \prod_{i=1}^{4} \sin((x - x_i)/2)$ the method (11) was tested at different values of x_1, x_2, x_3 and x_4. In the case when $x_1 = -1.7$, $x_2 = 0.3$, $x_3 = 0.5$, $x_4 = 1.7$, $x_1^{[0]} = -1.5$, $x_2^{[0]} = 0$, $x_3^{[0]} = 0.7$ and $x_4^{[0]} = 1.4$ we obtain $x_1^{[7]} = -1.7(14*0)$, $x_2^{[7]} = 0.3(14*0)6$, $x_3^{[7]} = 0.5(14*0)3$ and $x_4^{[7]} = 1.7(14*0)$.

Remark 2.6. The numerical experiments and the proof of Theorem 2.4 as well show that the additional insertion into (11) of the normalizing constants B_k is very significant. Without this normalization of the iteration procedure, it is divergent. In the algebraic case the normalization is carried out in a natural way, since the leading coefficiet (in front of x^N) can always be made equal to one.

Remark 2.7. If $T_N(x)$ is written in the form $a_0 + \sum_{k=1}^{N}(a_k \cos kx + b \sin kx)$, then it can easily be transformed into an A-polynomial $A_{2N}(x)$ with complex coefficients and for $A_{2N}(x)$ the method (4) can be used. However, the method (11) can be used directly. Moreover, in this case, for more economical calculation of the values of $T_N(x)$ it is preferable to use the scheme below: $c_N = a_N$, $s_N = b_N$, $p = \sin x$, $q = \cos x$, $(c_{k-1} = a_{k-1} + qc_k + ps_k$, $s_{k-1} = b_{k-1} + qs_k - pc_k)$, $k = N, ..., 3, 2$; $T_N(x) = a_0 + qc_1 + ps_1$. This algorithm requires only one call to the standard functions sinx and cosx. If, however, $T_N(x)$ is written in some other form, then it is sufficient to determine only N and work by the method (11). Such is, for instance, the case $T_N(x) \equiv (\cos x + \sin 3x)$. It is clear that $N = 60$ but the writing of $T_N(x)$ upon the base $(1, \sin x, \cos x, ..., \sin 60x, \cos 60x)$ is labour-consuming.

In the case when $f(x) = E_N(x)$ is an exponential polynomial (E-polynomial) (the basic system is $\{1, \mathrm{sh}x, \mathrm{sh}x, ..., \mathrm{sh}Nx, \mathrm{sh}Nx\}$ or $\{1, e^x, e^{-x}, ..., e^{Nx}, e^{-Nx}\}$), then the analogue

$$x_i^{[k+1]} = x_i^{[k]} - 2C_k E_N(x_i^{[k]}) \Big/ \prod_{j=1, j\neq i}^{2N} \mathrm{sh}((x_i^{[k]} - x_j^{[k]})/2), \tag{12}$$

$$i = \overline{1, 2N}, \ k = 0, 1, ...$$

of (11) can be used.

The normalizing constants C_k are chosen as follows:

$$C_k = [E_N(z)]^{-1} \prod_{j=1}^{2N} \mathrm{sh}((z - x_j^{[k]})/2),$$

where z is an arbitrary number, for which $E_N(z) \neq 0$.

Theorem 2.8. *Let $0 < q < 1$, $d = \min_{i\neq j} |x_i - x_j|$, where $\{x_i\}_1^{2N}$ are the roots of $E_N(x)$, $d_1 = \min |z - x_i|$, $p = \max_{i\neq j} |x_i - x_j|$, $p_1 = \max |z - x_i|$, $r = \min\{|\mathrm{sh}(d/2 - c)|, |\mathrm{sh}(d_1/2)|\}$, $R = \max\{2|\mathrm{sh}(2p_1 + c)/4|, 2|\mathrm{sh}(2p + 3c)/4|\}$, $L = \max\{1, R/r\}$ and c is sufficiently small, so that $0 < c < 1$, $d - 2c > 0$ and $cL^{4N+1} < 1$. Then $T(0, k, 2)$ holds true.*

Example 2.9. The E-polynomial

$$E_2(x) = a_0 + a_1 e^{-x} + b_1 e^x + a_2 e^{-2x} + b_2 e^{2x} \tag{13}$$

where $a_0 = (e^3 + e^{-3} + pq)/16$, $a_1 = -(e^{7/2}p + +e^{1/2}q)/16$, $b_1 = -(e^{-7/2}p + e^{-1/2}q)/16$, $a_2 = e^4/16$, $b_2 = e^{-4}/16$, $p = 2\mathrm{sh}(3/2)$, $q = 2\mathrm{sh}(0.5)$ has as roots the following numbers $(-1, 2, 3, 4)$. At $x_1^{[0]} = -1.2$, $x_2^{[0]} = 1.7$, $x_3^{[0]} = 2.8$, $x_4^{[0]} = 3.7$ and $z = 0$ by the method (12) we obtain $x_1^{[5]} = -0.(16 * 9)$, $x_2^{[5]} = 2.(15 * 0)$, $x_3^{[5]} = 3.(15 * 0)$ and $x_4^{[5]} = 3.(15 * 9)$. We shall adduce the results of another computation by the method (12) with the help of which we point out some advantage of (12) over the ordinary method of Newton. When $x_1^{[0]} = -0.2$, $x_2^{[0]} = 1$, $x_3^{[0]} = 2.5$ and $x_4^{[0]} = 5$ are taken then the method (12) has a normal behaviour and implies $x_1^{[9]} = -1.(14 * 0)$, $x_2^{[9]} = 2.(14 * 0)$, $x_3^{[9]} = 3.(14 * 0)$ and $x_4^{[9]} = 4.(14 * 0)$, while the method of Newton for individual searching of the roots gives $x_1^{[12]} = x_3^{[12]} = x_4^{[12]} = 4.(14 * 0)$ and $x_2^{[12]} = 2.(14 * 0)$.

3. Iteration methods with cubic rate of convergence for SFAR of A-, T- and E-polynomials.

Let us consider again the polynomoal (4) and for finding its roots apply the method (3), where we set $f(x) = A_N(x)$ and $Q_{[k]}(x) = \prod_{i=1}^{N}(x - x_i^{[k]})$. We get [27] the following method

$$(14) \qquad x_i^{[k+1]} = x_i^{[k]} - A_N(x_i^{[k]})[2y_i^{[k]} - A_N'(x_i^{[k]}) + A_N(x_i^{[k]})z_i^{[k]}]/(y_i^{[k]})^2,$$

$$i = \overline{1, N}, \ k = 0, 1, 2, ...,$$

where

$$y_i^{[k]} = \prod_{j=1, j\neq i}^{N} (x_i^{[k]} - x_j^{[k]}), \quad z_i^{[k]} = \sum_{j=1, j\neq i}^{N} (x_i^{[k]} - x_j^{[k]}).$$

Theorem 3.1. *Let q be a fixed number $0 < q < 1$ and $d = \min_{i\neq j} |x_i - x_j|$. Let c be sufficiently small so that the inequalities $d - 2c > 0$ and $2^{3N}c^2/(d - 2c)^2 < 1$. Then $T(0, k, 3)$ holds true.*

Example 3.2. Let us compare the method (14) and the usual method of Chebyshev for individual finding roots of equation (10). With the same initial approximations as in Example 2.3, Chebyshev's method implies $x_1^{[3]} = -3.(13 * 9)7$, $x_2^{[3]} = 0.(9 * 9)3$, $x_3^{[3]} = 9.(12 * 9)6$, while the method (14) results in $x_1^{[3]} = -3.(17 * 0)$ $x_2^{[3]} = 1.(17 * 0)$ and $x_3^{[3]} = 10.(16 * 0)$.

R e m a r k 3.3. When programming the method (14) it is advisable to use the generalized scheme of Horner [14] for simultaneous calculation of the value of the polynomial and its derivatives.

Analogously, from (9) the iteration computational schemes for SFAR of T - and E-polynomials can be derived. Following the procedure of deriving method (11) from (9) we obtain [17] :

$$(15)\ x_i^{[k+1]} = x_i^{[k]} - 4B_kT_N(x_i^{[k]})[t_i^{[k]} - B_kT_N'(x_i^{[k]}) + B_kT_N(x_i^{[k]})u_i^{[k]}]/(t_i^{[k]})^2,$$

$$i = \overline{1, 2N},\ k = 0, 1, 2, ...$$

where

$$t_i^{[k]} = \prod_{j=1, j\neq i}^{2N} \sin((x_i^{[k]} - x_j^{[k]})/2), \quad u_i^{[k]} = \frac{1}{2} \sum_{j=1, j\neq i}^{2N} \operatorname{cotg}((x_i^{[k]} - x_j^{[k]})/2).$$

Theorem 3.4. *Let q, d, d_1, r be as in Theorem 2.4 and c is sufficiently small, so that $d - 2c > 0$ and $\sqrt{3}2^{5N}c/r < 1$. Then $T(0, k, 3)$ holds true.*

E x a m p l e 3.5 We again consider the T-polynomial from Example 2.5. At $x_1^{[0]} = -1.5$, $x_2^{[0]} = 0.2$, $x_3^{[0]} = 0.4$ and $x_4^{[0]} = 1.5$ the method (15) results in $x_1^{[4]} = -1.7(14*0)$, $x_2^{[4]} = 0.2(15*9)$, $x_3^{[4]} = 0.5(15*0)9$ and $x_4^{[4]} = 1.7(14*0)$, while the ordinary method of Chebyshev for the third root is divergent.

In the case of the E-polynomial $E_N(x)$ from (9) we obtain [34]

$$(16) x_i^{[k+1]} = x_i^{[k]} - 4C_kE_N(x_i^{[k]})[v_i^{[k]} - C_kE_N'(x_i^{[k]}) + C_kE_N(x_i^{[k]})w_i^{[k]}]/(v_i^{[k]})^2,$$

$$i = \overline{1, 2N},\ k = 0, 1, 2, ...,$$

where

$$v_i^{[k]} = \prod_{j=1, j\neq i}^{2N} \operatorname{sh}((x_i^{[k]} - x_j^{[k]})/2), \quad w_i^{[k]} = \frac{1}{2} \sum_{j=1, j\neq i}^{2N} \operatorname{cth}((x_i^{[k]} - x_j^{[k]})/2).$$

The normalizing factors C_k are chosen as in (12).

Theorem 3.6. *Let q, d, d_1, p, p_1, r, R, and L be as in Theorem 2.8. Let the number c be sufficiently small so that the conditions $0 < c < 1$, $d - 2c > 0$ and $2^{4N}c^2(2^{4N+2}r^2L^2 + N) < r^2$ are satisfied. Then $T(0, k, 3)$ holds true.*

E x a m p l e 3.7. We consider the equation from Example 2.9 with initial approximations $x_1^{[0]} = -0.5$, $x_2^{[0]} = 1.7$, $x_3^{[0]} = 2.6$ and $x_4^{[k]} = 4.3$ by method (16) at the forth iteration we achieve the exact roots, while the method of Chebyshev for the first and the third roots is divergent.

4. Methods for simultaneous finding all zeros with known multiplicities.

The methods considered in sections 2 and 3 require all roots of the polynomial to be simple. The methods from section 2 with quadratic convergence appeared to be simpler.Thus the question arises can't we generalize the methods (4), (11) and (12) so that to preserve their quadratic convergence and refine simultaneously all roots of A-, T-, and E-polynomials with their help both in the cases when the multiplicities are arbitrary but given.

It occurs that we can answer positively to the question, which has thus arisen. The underlying idea [28] of our further considerations consists in the important fact, found out by us that the correlation (7) holds true. Namely, (7) prompts that if the polynomial (4) has n different roots $\{x_i\}_1^n$ with multiplicities $\{\alpha_i\}_1^n$ respectively $\alpha_i \geq 1$, $\sum_{i=1}^n \alpha_i = N$, then it is possible in (7) to try to use D_A for the polynomial $A_N(x)$ with multiple knots

$$(17) \qquad \sum_{i=1}^{n} \alpha_i x_i^{[k+1]} = \sum_{i=1}^{n} \alpha_i x_i^{[k]} - D_A[\underbrace{x_1^{[k]}, ..., x_1^{[k]}}_{\alpha_1 - times}; \dots \underbrace{x_n^{[k]}, ..., x_n^{[k]}}_{\alpha_n - times}; A_N(x)].$$

The formula (17) is an analogue of (7). If we use the expression [1] for the divided differences with multiple knots we can transform (17) into the form

$$(18) \qquad \sum_{i=1}^{n} \alpha_i x_i^{[k+1]} = \sum_{i=1}^{n} \alpha_i x_i^{[k]} - \sum_{i=1}^{n} \sum_{j=0}^{\alpha_i - 1} \frac{A_N^{(j)}(x_i^{[k]})}{j!(\alpha_i - 1 - j)!} \times$$

$$\left[\frac{(x - x_i^{[k]})^{\alpha_i}}{\Omega_{n,A}^{[k]}(x)} \right]_{x=x_i}^{(\alpha_i - 1 - j)},$$

where $\Omega_{n,A}^{[k]}(x) = \prod_{l=1}^{n} (x - x_l^{[k]})^{\alpha_l}$. Then

$$(19)\qquad x_i^{[k+1]} = x_i^{[k]} - \frac{1}{\alpha_i}\sum_{j=0}^{\alpha_i-1}\frac{A_N^{(j)}(x_i^{[k]})}{j!(\alpha_i-1-j)!}\left[\frac{x-x_i^{[k]})^{\alpha_i}}{\Omega_{n,A}^{[k]}(x)}\right]_{x=x_i^{[k]}}^{(\alpha_i-1-j)}$$

$$i=\overline{1,n},\ k=0,1,2,...,$$

will correspond to the method (5). The method (19) is a natural generalization of method WIDDK (5).

Theorem 4.1 [18]. *Let the numbers $\{x_i\}_1^n$ be roots of $A_N(x)$ and have multiplicities $\{\alpha_i\}_1^n$ respectively, $\alpha_i \geq 1$, $\sum_{i=1}^n \alpha_i = N$. Let $d = \min_{s\neq i}|x_s - x_i|$ and $b = \max_{s\neq i}|x_s - x_i|$. Suppose there exist numbers q and c such that $c > 0$, $0 < q < 1$, $d - 2c > 0$ and*

$$c\left\{\frac{2^N}{d-2c} + \max_{i=\overline{1,n}}\frac{1}{\alpha_i}\sum_{\substack{l+m=\alpha_i-1\\ l\geq 0,\ m>0}}\binom{\alpha_i}{l}(d-2c)^{\alpha_i-m-N}\times\right.$$

$$\left.\sum_{\substack{\nu_1+\ldots+\nu_{i-1}+\nu_{i+1}+\ldots+\nu_n=m\\ \nu_1\geq 0,\ldots,\nu_n\geq 0}}\ \prod_{s=1,s\neq i}^{n}\sum_{p=0}^{\alpha_s-\nu_s}\sum_{r=0}^{p}\binom{\alpha_s}{p}\binom{p}{r}b^r(cq)^{\alpha_s-r}\right\} < 1.$$

Then $T(0,k,2)$ holds true.

Example 4.2. Consider the equation

$$A_6(x) \equiv x^6 - 15x^4 - 14x^3 + 36x^2 + 24x - 32 = 0$$

whose roots $x_1 = 1$, $x_2 = -2$, and $x_3 = 4$ have multiplicities $\alpha_1 = 2$, $\alpha_2 = 3$ and $\alpha_3 = 1$ respectively. At $x_1^{[0]} = 3$, $x_2^{[0]} = -5$ and $x_3^{[0]} = 7$ by method (19) one gets $x_1^{[8]} = 1.(28*0)$, $x_2^{[8]} = -2.(28*0)$, and $x_3^{[8]} = 4.(28*0)$.

Proposition 4.3. *The successive approximations obtained by method (19) satisfy the correlation*

$$\sum_{i=1}^n \alpha_i x_i^{[k+1]} = \sum_{i=1}^n \alpha_i x_i,\ k=0,1,2,...,$$

independently of the choice of initial approximations $\{x_i^{[0]}\}_1^n$.

The Proposition 4.3 can be proved as Proposition 2.2.

Remark 4.4. Some investigations concerning the development of method for SFAR of A-polynomial with arbitrary given multiplicities are published in [16] after our paper [28].

In paper [18] we have introduced D_T and D_E with multiple knots. Here we shall show how they can be used when constructing methods for refining multiple roots of T- and E-polynomials.

When we sum with respect to i all equalities in (11) we receive

$$\sum_{i=1}^{2N} \alpha_i x_i^{[k+1]} = \sum_{i=1}^{2N} \alpha_i x_i^{[k]} - 2B_k D_t(x_1^{[k]}, ..., x_n^{[k]}; T_N(x)).$$

Then

$$(20) \quad \sum_{i=1}^{n} \alpha_i x_i^{[k+1]} = \sum_{i=1}^{n} \alpha_i x_i^{[k]} - 2B_k D_T(\underbrace{x_1^{[k]}, ..., x_1^{[k]}}_{\alpha_1-times}; ... \underbrace{x_n^{[k]}, ..., x_n^{[k]}}_{\alpha_n-times}; T_N(x)),$$

where $\{\alpha_i\}_1^n$ are the corresponding multiplicities of zeros $\{x_i\}_1^n$ of polynomial $T_N(x)$, $\sum_{i=1}^n \alpha_i = 2N$, will be an analogue of the above formula in the case of multiple zeros.

Using the expression for D_T with multiple knots [18] the equality (20) is rewritten in the form

$$\sum_{i=1}^{n} \alpha_i x_i^{[k+1]} = \sum_{i=1}^{n} \alpha_i x_i^{[k]} - 2B_k \sum_{i=1}^{n} 2^{\alpha_i - 1} \sum_{j=0}^{\alpha_i - 1} \frac{T_N^{(j)}(x_i^{[k]})}{j!(\alpha_i - 1 - j)!} \times$$

$$\left[\frac{(\sin((x - x_i^{[k]})/2))^{\alpha_i}}{\Omega_{n,T}^{[k]}(x)} \right]_{x = x_i^{[k]}}^{(\alpha_i - 1 - j)},$$

and the iteration method

$$(21) \qquad x_i^{[k+1]} = x_i^{[k]} - \frac{2^{\alpha_i} B_k}{\alpha_i} \sum_{j=0}^{\alpha_i - 1} \frac{T_N^{(j)}(x_i^{[k]})}{j!(\alpha_i - 1 - j)!} \times$$

$$\left[\frac{(\sin((x - x_i^{[k]})/2))^{\alpha_i}}{\Omega_{n,T}^{[k]}(x)} \right]_{x = x_i^{[k]}}^{(\alpha_i - 1 - j)} \quad i = \overline{1, n}, \ k = 0, 1, 2, ...,$$

where $B_k = [T_N(y)]^{-1} \prod_{j=1}^{n} (\sin((y - x_j^{[k]})/2))^{\alpha_j}$, $\Omega_{n,T}^{[k]}(x) = \prod_{l=1}^{n}(\sin((x - x_l^{[k]})/2))^{\alpha_l}$, y is an arbitrary point, $y \in [-\pi, \pi]$, $T_N(y) \neq 0$, will correspond to the method (11).

Example 4.5. The equation

$$T_3(x) \equiv \left(\sin\frac{x-1}{2}\right)^3 \left(\sin\frac{x-2}{2}\right)^2 \sin\frac{x-2.5}{2} = 0$$

has been solved by the method (21) at initial approximations $x_1^{[0]} = 0.8$, $x_2^{[0]} = 1.7$, $x_3^{[0]} = 2.3$ and $y = 0$ and we achieve $x_1^{[7]} = 1.(15*0)$, $x_2^{[7]} = 2.(15*0)$ and $x_3^{[7]} = 2.5(14*0)$.

In the case of E-polynomial $E_N(x)$ analogously to (21) we have

$$(22)\quad x_i^{[k+1]} = x_i^{[k]} - \frac{2^{\alpha_i} C_k}{\alpha_i} \sum_{j=0}^{\alpha_i - 1} \frac{E_N^{(j)}(x_i^{[k]})}{j!(\alpha_i - 1 - j)!} \left[\frac{(\mathrm{sh}((x - x_i^{[k]})/2))^{\alpha_i}}{\Omega_{n,E}^{[k]}(x)}\right]_{x=x_i^{[k]}}^{(\alpha_i - 1 - j)}$$

$$i = \overline{1,n},\ k = 0, 1, 2, \ldots,$$

where $C_k = [E_N(z)]^{-1} \prod_{l=1}^{n}(\mathrm{sh}((z - x_l^{[k]})/2))^{\alpha_l}$, $\Omega_{n,E}^{[k]}(x) = \prod_{l=1}^{n}(\mathrm{sh}((x - x_l^{[k]})/2))^{\alpha_l}$, z is an arbitrary number, $E(z) \neq 0$.

Example 4.6. The iteration formula (22) has been applied in solving

$$E_2(x) \equiv a_0 + a_1 e^{-x} + b_1 e^x + a_2 e^{-2x} + b_2 e^{2x} = 0,$$

where

$$a_0 = (4 + e^5 + e^{-5})/16,\ a_1 = -(e^{-2} + e^{-3})/8,$$
$$a_2 = e/16, b_1 = -(e^2 + e^{-3})/8,\ b_2 = (16e)^{-1}.$$

This equation has double roots $x_1 = -2$ and $x_2 = 3$. At $x_1^{[0]} = -1.5$, $x_2^{[0]} = 3.4$ and $z = 0$ by method (22) we get $x_1^{[6]} = -2.(13*0)$ and $x_2^{[6]} = 3.(13*0)$.

Methods (21) and (22) have quadratic rate of convergence. The theorems which justify that fact are similar to Theorem 4.1.

5. Algebraic, trigonometric and exponential analogues of Obreshkoff's method.

The iteration methods for SFAR of A-, T-and E-polynomials in §3 have a cubic convergence rate but the convergence depends essentially on the normalizing factors B_k and C_k introduced. Less labour-consuming methods with a cubic rate of convergence without normalizing factors can be derived. To achieve this aim we can take instead, as a basis, Obreshkoff's method [24,25,9] for individual finding some zero of the function $f(x)$

$$(23)\ x^{[k+1]} = x^{[k]} - f(x^{[k]})/[f(x^{[k]}) - \frac{1}{2}f(x^{[k]})f''(x^{[k]})/f'(x^{[k]})],\ k = 0, 1, \ldots$$

In the case when $f(x)$ is an A-polynomial a rather efficient analogue of method (23) for SFAR of $A_N(x)$ under the condition that all its zeros are simple has been proposed by Ehrlich [8]:

$$(24) \qquad x_i^{[k+1]} = x_i^{[k]} - A_N(x_i^{[k]})/[A'_N(x_i^{[k]}) - A_N(x^{[k]}) \sum_{j=1,j\neq i}^{N} (x_i^{[k]} - x_j^{[k]})^{-1}],$$

$$k = \overline{1, N},\ k = 0, 1, \ldots.$$

Ehrlich has proved that the method (24) is convergent to the zeros of the polynomial $A_N(x)$ with a cubic convergence. It can be seen from (24) that the expressions $\sum_{j=1,j\neq i}^{N}(x_i^{[k]} - x_j^{[k]})^{-1}]$, $i = \overline{1, N}$ depend only on N and on the successive approximations $\{x_i^{[k]}\}_{i=1}^{N}$ and do not depend on the polynomial $A_N(x)$. If we again introduce the auxiliary A-polynomial $Q_{[k]}(x)$ as in §2 then it can easily be seen that

$$Q''_{[k]}(x_i^{[k]})/Q'_{[k]}(x_i^{[k]}) = \sum_{j=1,j\neq i}^{N} (x_i^{[k]} - x_j^{[k]})^{-1}].$$

Consequently, method (24) can be written in the following way:

$$(25) \qquad x_i^{[k+1]} = x_i^{[k]} - A_N(x_i^{[k]})/[A'_N(x_i^{[k]}) - \frac{1}{2}A_N(x_i^{[k]}) \times Q''_{[k]}(x_i^{[k]})/Q'_{[k]}(x_i^{[k]})].$$

Formula (25) points out the connection of Ehrlich's method (24) with Obreshkoff's method (23). Formula (25) suggests as well that if $f(x)$ is a generalized polynomial of the order N developed upon some Chebyshev's system of basic functions $\{\varphi_k(x)\}_0^N$, then for finding its roots $\{x_i\}$ under the condition that they all are simple the method

$$(26) \qquad x_i^{[k+1]} = x_i^{[k]} - f(x_i^{[k]})/[f'(x_i^{[k]}) - \frac{1}{2}f(x_i^{[k]})\times$$

$$Q''_{[k]}(x_i^{[k]})/Q'_{[k]}(x_i^{[k]})],\ i = \overline{1, N}, \quad k = 0, 1, \ldots,$$

where $Q_{[k]}(x)$ is a G - polynomial of order N developed upon the same system $\{\varphi(x)\}_0^N$ and having zeros at points $\{x_i^{[k]}\}_{i=1}^N$ can be tested.

Evidently method (26) is a generalization of (25).In applying method (26) the difficulty lies in the choice of proper G-polynomials $Q_{[k]}(x)$, corresponding to the G- polynomial $f(x)$.

Let us now consider the particular case when $f(x)$ is a T-polynomial $T_N(x)$. In this case the polynomial $Q_{[k]}(x)$ can be chosen so that $Q_{[k]}(x) = \prod_{j=1}^{2N} \sin((x - x_j^{[k]})/2)$, from where it follows that

$$(27) \qquad Q''_{[k]}(x_i^{[k]})/Q'_{[k]}(x_i^{[k]}) = \sum_{j=1, j\neq i}^{2N} \operatorname{cotg}((x_i^{[k]} - x_j^{[k]})/2).$$

Thus from (26) and (27) we get [19] the method

$$(28) \qquad x_i^{[k+1]} = x_i^{[k]} - T_N(x_i^{[k]})/[T'_N(x_i^{[k]}) - \frac{1}{2}T_N(x_i^{[k]})\times$$
$$\sum_{j=1, j\neq i}^{2N} \operatorname{cotg}((x_i^{[k]} - x_j^{[k]})/2)], \ i = \overline{1, 2N}, \ k = 0, 1, ...,$$

for SFAR of T-polynomial $T_N(x)$, lying in the strip between the parallel straight lines $x = \pm\pi$. The cubic covergence of method (28) is justified by the following theorem:

Theorem 5.1. *Let $0 < q < 1$, the zeros $\{x_i\}_1^{2N}$ of $T_N(x)$ be simple, $d = \min_{i\neq j} |x_i - x_j|$, $r = |\sin(d/2 - c)|$, where c is such a positive number for which the inequalities $d - 2c > 0$ and $2c^2((2N-1)/r^2 + 1) < 1$ are valid. Then $T(0, k, 3)$ holds true.*

Example 5.2. The roots of $T_2(x)$ from Example 2.5. have been found by method (28). At initial approximations $x_1^{[0]} = -1.5$, $x_2^{[0]} = 0.1$, $x_3^{[0]} = 0.7$ and $x_4^{[0]} = 1.4$ we obtain $x_1^{[4]} = -1.7(14*0)$, $x_2^{[4]} = 0.2(14*9)9$, $x_3^{[4]} = 0.5(14*0)3$ and $x_4^{[4]} = 1.7(14*0)$. In the case when $f(x)$ is an E-polynomial $E_N(x)$ and $Q_{[k]}(x) = \prod_{j=1}^{2N} \operatorname{sh}((x - x_j^{[k]})/2)$ then from (26) we receive the method

$$(29) \qquad x_i^{[k+1]} = x_i^{[k]} - E_N(x_i^{[k]})/[E'_N(x_i^{[k]}) - \frac{1}{2}E_N(x_i^{[k]})\times$$
$$\sum_{j=1, j\neq i}^{2N} \operatorname{cth}((x_i^{[k]} - x_j^{[k]})/2)], \quad i = \overline{1, 2N}, \ k = 0, 1, ...$$

Theorem 5.3. *Let $0 < q < 1$ and the zeros $\{x\}_1^{2N}$ of the polynomial $E_N(x)$ be simple and let us denote $d = \min_{i\neq j}|x_i - x_j|$, $r = |\mathrm{sh}(d/2 - c)|$ where c is sufficiently small positive number so that the inequalities $d - 2c > 0$ and $4c^2((2N-1)\mathrm{sh}(c/r^2+3) < 1$ are satisfied. Then $T(0,k,3)$ is valid.*

Example 5.4. Method (29) has been experimented on Example 2.9. At $x_1^{[0]} = -1.2$, $x_2^{[0]} = 1.7$, $x_3^{[0]} = 2.8$ and $x_4^{[0]} = 3.7$ through the method (29) we get $x_1^{[4]} = -0.(15*9)7$, $x_2^{[4]} = 2.(15*0)$, $x_3^{[4]} = 3.(14*0)1$ and $x_4^{[4]} = 3.(15*9)$.

Method (26) enables us to obtain an iteration scheme for SFAR of a given G- polynomial $P_N(x)$ upon some Chebyshev system $\{\varphi(x)\}_0^N$ when the zeros $\{x_i\}_1^N$ are simple. One of the possible ways [21] of choosing the auxiliary polynomial $Q_{[k]}(x)$ is the following :

$$Q_{[k]}(x) = \det\begin{bmatrix} \varphi_0(x) & \varphi_1(x) & \dots & \varphi_N(x) \\ \varphi_0(x_1^{[k]}) & \varphi_1(x_1^{[k]}) & \dots & \varphi_N(x_1^{[k]}) \\ \dots & \dots & \dots & \dots \\ \varphi_0(x_N^{[k]}) & \varphi_1(x_N^{[k]}) & \dots & \varphi_N(x_N^{[k]}) \end{bmatrix}.$$

Furthermore in §7 we shall consider a more general method for the same case but when zeros have arbitrary given multiplicities.

6. A method with a quadratic rate of convergence for determining all arbitrary multiplicity roots of a given G-polynomial.

Let the G - polynomial

$$(30) \qquad P_N(x) \equiv \varphi_N(x) + \sum_{k=0}^{N-1} \alpha_k\varphi_k(x)$$

where $\{\varphi_k(x)\}_0^N$ is a Chebyshev system with respect to the interval $[a,b]$ be given.

Without loss of generality we may accept that the coefficient in front of $\varphi_N(x)$ is equal to one. In [20] a method for SFAR $\{x_i\}_1^N$ of (30) has been derived and investigated when the zeros are simple and when the system $\{\varphi_k(x)\}_0^N$ is an arbitrary system of continuously differentiable functions.

It occurs, however, that when $\{\varphi_k(x)\}_0^N$ are sufficiently smooth a more general method can be obtained both for the case when the multiplicities $\{\alpha_i\}_1^m$ of the zeros $\{x_i\}_1^m$ are arbitrary $\sum_{i=1}^m \alpha_i = N$. That is why here we shall present only the general case [30]. The scheme proposed below is based on (12).

For the sake of simplicity we introduce the notation :

$$(31)\ M\begin{bmatrix}\varphi_0 & \varphi_1 \dots \varphi_N \\ x & z_1 \dots z_m \\ 1 & \alpha_1 \dots \alpha_m\end{bmatrix} = \begin{bmatrix} \varphi_0(x) & \varphi_1(x) & \dots & \varphi_N(x) \\ \varphi_0(z_1) & \varphi_1(z_1) & \dots & \varphi_N(z_1) \\ \varphi_0'(z_1) & \varphi_1'(z_1) & \dots & \varphi_N'(z_1) \\ \dots & \dots & \dots & \dots \\ \varphi_0^{(\alpha_1-1)}(z_1) & \varphi_1^{(\alpha_1-1)}(z_1) & \dots & \varphi_N^{(\alpha_1-1)}(z_1) \\ \dots & \dots & \dots & \dots \\ \varphi_0(z_m) & \varphi_1(z_m) & \dots & \varphi_N(z_m) \\ \varphi_0'(z_m) & \varphi_1'(z_m) & \dots & \varphi_N'(z_m) \\ \dots & \dots & \dots & \dots \\ \varphi_0^{(\alpha_m-1)}(z_m) & \varphi_1^{(\alpha_m-1)}(z_m) & \dots & \varphi_N^{(\alpha_m-1)}(z_m) \end{bmatrix}$$

We consider the iteration process

$$(32)\qquad x_i^{[k+1]} = x_i^{[k]} - (-1)^N P_N^{(\alpha_i-1)}(x_i^{[k]})/Q_{[k]}^{(\alpha_i)}(x_i^{[k]}),$$

$$i = \overline{1,m},\ k = 0, 1, \dots,$$

where

$$Q_{[k]}(x) = \det M\begin{bmatrix}\varphi_0 & \varphi_1 & \dots & \varphi_N \\ x & x_1^{[k]} & \dots & x_m^{[k]} \\ 1 & \alpha_1 & \dots & \alpha_m\end{bmatrix} / \det M\begin{bmatrix}\varphi_0 & \varphi_1 & \dots & \varphi_N \\ x_1^{[k]} & x_2^{[k]} & \dots & x_m^{[k]} \\ \alpha_1 & \alpha_2 & \dots & \alpha_m\end{bmatrix}$$

For finding the α^{th} derivative of $Q_{[k]}(x)$ it is sufficient to differentiate α_i times the first row of the determinant in the numerator of $Q_{[k]}(x)$.

Theorem 6.1. *Let* $0 < c < 1$, $0 < q < 1$ *and*

$$L(c) = \min_{i=1,m}\ \inf_{|y_i - x_i|\le cq} \left| \frac{d^{\alpha_i}}{dx^{\alpha_i}} \left\{ \det M\begin{bmatrix}\varphi_0 & \varphi_1 & \dots & \varphi_N \\ x & y_1 & \dots & y_m \\ 1 & \alpha_1 & \dots & \alpha_m\end{bmatrix}\right\}_{x=y_i} \right|$$

Let the derivatives $\varphi_s^{\alpha_i+1}(x)$, $i = \overline{1,m}$ *and constants* M_{sr} *exist such that* $|\varphi_s^{(r)}(x)| \le M_{sr}$, $\forall x \in [a,b]$, $s = \overline{0,N}$, $r = \overline{0, max_{1=\overline{1,m}}(\alpha_i+1)}$. *Let* c *be a number chosen sufficiently small so that the inequality*

$$c\left\{\prod_{r=0}^{N-1}\left(M_{r,\alpha_i}^2 + \sum_{s=1}^{m}\sum_{j=0}^{\alpha_s-1} M_{r,j}^2\right)\left[\left(M_{N,\alpha_i+1} + \sum_{j=0}^{N-1}|a_j|M_{j,\alpha_i+1}\right)^2 + \right.\right.$$

$$\sum_{s=1}^{m}\sum_{j=1}^{\alpha_s}\left(M_{N,j}+\sum_{l=o}^{N-1}|a_l|M_{l,j}\right)^2\Bigg]\Bigg\}^{1/2} < L(c),\ i=\overline{1,m},$$

is satisfied. Then $T(0,k,2)$ *holds true.*

Remark 6.2. In the particular case $\alpha_1=\alpha_2=\ldots=\alpha_m=1$ from (32) our method [20] can be obtained. In the still more particular case $\varphi_k(x)=x^k$, $k=\overline{0,N}$ the determinants in (32) are. determinants of Van der Mond and from (32) the method of WIDDK (22) is obtained.

The conditions of Theorem 6.1. indicate that the convergence of method (32) is local. In the following example we shall show that this method is convergent both at non-local choice of initial approximations.

Example 6.3. The equation from Example 4.2. has been solved through method (32) with computer accuracy of 7 decimal points. At $x_1^{[0]}=3$, $x_2^{[0]}=-5$ and $x_3^{[0]}=7$ we get $x_1^{[8]}=1.(7*0)$ $x_2^{[8]}=-2.(7*0)$ and $x_3^{[8]}=4.(7*0)$.

Other numerical experiments have been presented in [30]. In particular, the comparison between (32) and the modified Newton's method for individual determining multiple roots of the G - polynomials $P_N(x)$:

$$x_i^{[k+1]}=x_i^{[k]}-\alpha_i P_N(x_i^{[k]})/P_N'(x_i^{[k]}),\ i=\overline{1,m},\ k=0,1,...,$$

has been made. 50 iterations without any further improvement of numerical results have been necessary for root refining to the 5 digit of the equation from Example 6.3. with the same initial approximations.

7. A method with a cubic convergence for synchronous determination of all zeros with arbitrary multiplicities of a given G-polynomial.

We shall again consider the G-polynomial

$$P_N(x)=\sum_{j=0}^{N}a_j\varphi_j(x), \tag{33}$$

where $\{\varphi_j(x)\}_0^N$ is an arbitrary system of Chebyshev and $\{a_j\}_0^N$ are arbitrary real or complex coefficients. In contrast to §6 we here require only $a_N\neq 0$. Let (33) have real or complex zeros $\{z_i\}_1^m$ with multiplicities $\{\alpha_i\}_1^m$, $\sum_{j=1}^m\alpha_j=N$. We shall here consider a generalization of method (26) which, in its computational labour-consumption, is commensurable with the method described in §6, since in its application a renormalization of polynomial (33) is not necessary, but, in addition, it has higher (cubic) rate of convergence. We shall use the notations from §6.

So let us consider the following generalisation [22] of method (26):

$$(34)\qquad x_i^{[k+1]} = x_i^{[k]} - P_N^{(\alpha_i-1)}(x_i^{[k]}) \Big/ \Big[P_N^{(\alpha_i)}(x_i^{[k]}) - \frac{1}{2}P_N^{(\alpha_1-1)}(x_i^{[k]})\times$$

$$Q_{[k]}^{(\alpha_i+1)}(x_i^{[k]})/Q_{[k]}^{(\alpha_i)}(x_i^{[k]})\Big], \quad i=\overline{1,m},\ k=0,1,2,\dots,$$

where $Q_{[k]}(x)$ is a G-polynomial of order N upon the same system $\{\varphi_j(x)\}_0^N$, represented in the form:

$$Q_{[k]}(x) = \det M \begin{bmatrix} \varphi_0 & \varphi_1 & \dots & \varphi_N \\ x & x_1^{[k]} & \dots & x_m^{[k]} \\ 1 & \alpha_1 & \dots & \alpha_m \end{bmatrix}$$

Evidently $Q_{[k]}(x)$ has as zeros the numbers $\{x_i^{[k]}\}_{i=1}^m$ with multiplicities $\{\alpha_i\}_1^m$ respectievely.

Theorem 7.1. *Let $a = \max_{i=\overline{1,m}} \alpha_i$ and the Chebyshev system $\{\varphi_j(x)\}_0^N$ consists of sufficiently smooth functions. Let $|\varphi_j^{(l)}(x)| \le M_{jl}$, $l = \overline{0, a+2}$. Let the numbers q and c be such that $0<q<1$, $0<c<1$ and $L(c)$ is as in Theorem 6.1. and*

$$K(c) = L^2(c) - c\left\{\left(\sum_{r=0}^N |a_r| M_{r\alpha}\right)\left[\prod_{s=0}^N\left(M_{s,\alpha+1}^2 + \sum_{i=1}^m\sum_{k=0}^{\alpha_i-1} M_{sk}^2\right)\right]^{1/2}\right\} > 0,$$

$$c^2\left\{\prod_{s=0}^N\left(M_{s,\alpha+1}^2 + M_{s\alpha}^2 + \sum_{i=1}^m\sum_{k=0}^{\alpha_i-1} M_{sk}^2\right)\left[\left(\sum_{r=0}^N |a_r| M_{r\alpha+r}\right)^2 + \right.\right.$$

$$\left.\left.\left(\sum_{r=0}^N |a_r| M_{r,\alpha+1}\right)^2 \sum_{i=1}^m\sum_{k=0}^{\alpha_i-1}\left(\sum_{r=0}^N |a_r| M_{rk}\right)^2\right]\right\}^{1/2} < K(c).$$

Then $T(0,k,3)$ holds true.

Example 7.2. For the A-polynomial

$$P_6(x) = x^6 - 6x^5 + 50x^3 - 45x^2 - 108x + 108$$

having zeros $x_1 = -2$, $x_2 = 1$ and $x_3 = 3$ with multiplicities $\alpha_1 = 2$, $\alpha_2 = 1$ and $\alpha_3 = 3$ respectively, at initial approximations $x_1^{[0]} = -3$, $x_2^{[0]} = 0.1$ and $x_3^{[0]} = 4$ the method (34) yields $x_1^{[4]} = -2.(15*0)$, $x_2^{[4]} = 1.(15*0)$ and $x_3^{[4]} = 3.(14*0)1$.

Example 7.3. A G-polynomial with double zeros upon the basic system $\{1, x^2, \sin 3x, e^{-x}, (1+x^2)^{-1}\}$ has been constructed. The zeros of this polynomial through the method (34) have been refined at the forth iteration with 16 digits after the decimal point.

Other computational experiments have been carried out in [22].

Remark 7.4. At $\alpha_1 = ... = \alpha_m = 1$ from (34) our method [21] is obtained. In the still more particular case when the system of basic functions is $\{x^k\}_0^N$, Ehrlich's method (24) is obtained.

8. A note on the better behaviour of the methods for SFAR of polynomials.

The iteration methods developed in this paper have quadratic or cubic convergence, but the statements of the theorems point out that the convergence is a local one. To ensure the convergence it is necessary to choose the initial approximations sufficiently close to the zeros of the corresponding polynomial. Althogh the methods for zero localization of a polynomial (particularly for A-polynomials) are well-developed the iteration methods for zero determination at non-local choice of initial approximations are of great interest. A great many computational experiments with the methods, considered in this paper, show their advantage with respect to the classic methods of Newton, Chebyshev and Obreshkoff. They are convergent in many other cases as well when the initial approximations are chosen far from the zeros. One of the reasons for such a favourable behaviour of the iteration processes under consideration for SFAR of polynomials consists in the simultaneity and synchronization of root finding. For this examination we shall further use the continious analogues of the methods developed. The continious analogues treat the approximations of the roots as a system of material points, moving along the real axis. For simplicity's sake, we shall consider A-polynomials, but our reasoning will hold for T- and E-polynomials as well.

Let the equation (21) have single real roots $\{x_i\}_1^N$. Let the numbers $\{x_i^{[0]}\}_0^N$ be the initial approximations to the roots of (21). Consider the method

(22) and the classic Newton's method (??). Their continious analogues are:

$$(35)\qquad x_i'(t) = -A_N(x_i(t))/\prod_{j=1,j\neq i}^{N}(x_i(t)-x_j(t)),\ i=\overline{1,N},$$

$$(36)\qquad x_i'(t) = -A_N(x_i(t))/A_N'(x_i(t)),\ i=\overline{1,N},\ x_i(0)=x_i^{[0]}.$$

We take into account that the differential equations (35) and (36) govern the motion of some system of material points $\{x_i(t)\}_1^N$ that is in position $\{x_i^{[0]}\}_1^N$ at $t=0$.

Let us concentrate our attention to the root x_i of equation (21). We shall interpret the continious processes (35) and (36) in the neibourhood of the root x_i. Let us denote by A and B the projection on the X-axis of the local extrema in the neibourhood of x. We choose the initial approximation $x_i^{[0]}$ to the root x_i in one of the intervals (A, x_i) and (x_i, B); for example $x_i^{[0]} \in (A, x)$, which yields

$$D = A_N(x_i(0))/\prod_{j=1,j\neq i}^{N}(x_i(0)-x_j(0)) < 0$$

$$F = A_N(x_i(0))/A_N'(x_i(0)) < 0$$

It follows from formulae (35) and (36) that in both cases the derivative $x_i'(t)$ at the point $t=0$ is positive, i.e. we shall have "a motion" of the point $x_i(t)$ towards the root x_i. Things are different, however, if we choose the initial approximation in one of the intervals (x_{i-1}, A) or (B, x_{i+1}). For example, if $x_i^{[0]} \in (B, x_{i+1})$ we obtain $D > 0$ and $F < 0$. According to formula (36), $x_i'(0) > 0$ i.e. "a motion" of $x_i(t)$ occurs in a counter-rootwise direction. Consequently, the continious process does not converge to the root x_i from the given choice of the initial approximation $x_i^{[0]}$. In this case, however, formula (35) is suitable, i.e. $x_i'(0) < 0$ which shows that $x_i(t) \to x_i$.

The wider convergence domain of the continious process (35) compared with that of the continious process (36) suggests a similar advantage of formula (22) over formula (??). The experiments that have been carried out confirm convincingly this suggestion.

If we replace the derivatives $x_i'(t)$, $i=\overline{1,N}$, in the continious process (35) by their difference quotients $(x_i(t+h)-x_i(t))/h$, $i=\overline{1,N}$, we obtain the formula :

$$(37)\qquad x_i(t+h) = x_i(t) - hA_N(x_i(t))/\prod_{j=1,j\neq i}^{N}(x_i(t)-x_j(t)),\ i=\overline{1,N},$$

Formula (37) explains the fact that in the numerical realization of (35), "the motion" along the parameter t should sometimes be performed with a small step h, in order to preserve the disposition of the material system $\{x_i(t)\}_1^N$ with respect to the roots of (21). This way, however, is impractical if the realization of (35) consumes too much computer time.For this reason, it is often best to use the step $h = 1$, which gives the iteration (22).

9. On the determination of the zero multiplicities of polynomials.

In the methods for SFAR of A-, T-, $E-$ and G- polynomials in the derivation of the corresponding iteration processes as well, as in the proof of their convergence, it is assumed that the zero multiplicities are given. In this way the question of determining the multiplicities of the roots of the given polynomial arises. This same question has been considered by many authors.

In [23] a method of approximative determination of root multiplicities has bee presented, considering a version of Bernoulli's method, making use of Newton's sums.

In [15] an iteration method for approximative calculation of multiplicities has been constructed.

In [3,26] the multiplicities has been determined approximatively as well. The common characteristic of methods [3,15,23,26] is the fact that they are approximative and preliminary information on the zeros of the polynomial, considered, is necessary for their application.

The question of the determination of the zero multiplicities of some G-polynomial upon an arbitrary Chebishev system is still open.

This paragraph is devoted to a method for exact determination of the multiplicities of all roots of a given A-polynomial with real coefficients. In this connection, information only about the polynomial coefficients has been used. This question has been reduced to solving some integer triangular system of linear algebraic equations. The method is simple and easily realizable on computer.

So, let an A-polynomial [31]

$$P_N(x) = \sum_{i=0}^{N} a_i x^{N-i} \tag{38}$$

with real coefficients $\{a_i\}_0^N$, having real or complex zeros $\{x_i\}_1^m$ with multiplicities $\{\alpha_i\}_1^m$ respectively: $\sum_{i=1}^m \alpha_i = N, m \leq N$ be given. Then the representation

$$(39) \qquad P_N(x) = \alpha_0 \prod_{j=1}^{m} (x - x_j)^{\alpha_j}$$

is valid. In practice, it is very difficult to obtain the factorised form (39) of the polynomial $P_N(x)$, since for this purpose it is necessary to find all its roots and their multiplicities. However, using Euclid's algorithm for the determination of greatist common diviser (GCD) of two polynomials and Hermite's method [24], the polynomial (38) can be factorised in the following way

$$(40) \qquad P_N(x) = X_1 X_2^2 X_3^3 ... X_N^N,$$

where $X_\alpha(x) = \prod_{i=1}^{K_\alpha}(x - x_i), \ \alpha = \overline{1, N}$.

The polynomial $X_\alpha(x)$ is a product of all binomial factors of polynomial $P_N(x)$, and K_α is the number of all zeros with the multiplicity a of the polynomial (38), i.e. $\sum_{\alpha=1}^N \alpha K_\alpha = \sum_{i=1}^m \alpha_i = N$.

The GCD of the polynomial $P_N(x)$ and its derivative $P_N'(x)$ will be

$$(41) \qquad D_1(P_N, P_N') = X_2 X_3^2 X_4^3 ... X_N^{N-1}.$$

Let S_1 be the degree of $D_1(P_N, P_N')$. Analogously, we determine the GCD of polynomial $D_1(P_N, P_N')$ and its derivative $D_1'(P_N, P_N')$, i.e.

$$(42) \qquad D_2(D_1, D_1') = X_3 X_4^2 X_5^3 ... X_N^{N-2}.$$

Let S_2 be the degree of $D_2(D_1, D_1')$. Continuing in this way we reach GCD

$$(43) \qquad D_{N-1}(D_{N-2}, D_{N-2}') = X_N,$$

which already be mutually prime with its derivative and its degree will be equal to the degree of the factor $X_N(x)$, i.e. $S_{N-1} = K_N$.

Now, we find the degrees in the right sides of (40)–(43) and setting them equal to N, S_1, ..., S_{N-1} respectievely, we find the following triangular integer system with respect to the unknowns $\{K_i\}_1^N$:

$$(44) \qquad \left| \begin{array}{l} K_1 + 2K_2 + 3K_3 + ... + NK_N = N \\ K_2 + 2K_3 + 3K_4 + ... + (N-1)K_N = S_1 \\ K_3 + 2K_4 + 3K_5 + ... + (N-2)K_N = S_2 \\ .... \\ K_{N-1} + 2K_N = S_{N-2} \\ K_N = S_{N-1} \end{array} \right.$$

The system (44) is solved in an obvious way. The method represented here, has been realized by us on computer and in a variety of tests it has proved its efficiency.

References

1. I.S. Beriozin, E.N. Zhidkov. Computational Methods, **1**, Moscow, Nauka 1966 (in Russian).
2. B. Boyanov, Khr. Semerdzhiev. Numerical Methods. Publishing house of Plovdiv University. Plovdiv 1982 (in Bulgarian).
3. M.N. Channabasappa. A note on the computation of multiple zeros of polynomials by Newton's method. *BIT* **19**, 1979, 134–135.
4. D.F. Davidenko. On a new method for numerical solving system of nonlinear equations. *Dokl. AN. USSR* **88**, 1953, 601–603 (in Russian).
5. K. Dochev. An alternative method of Newton for simultaneous calculation of all the roots of a given algebraic equation. *Phys. Math. J., Bulg. Acad. Sci.* **5**, 1962, 136-139 (in Bulgarian).
6. K. Dočev. Über Newtonsche Iterationen. *C.R. Acad. Bulg. Sci.* **15**, 1962, 699–701.
7. E.Durand. Solution numérique des équation algébraique, tome 1. Paris. Masson 1960.
8. L.M. Ehrlich. A modified Newton's method for polynomials. *Comm. ACM* **10**, 1967, 107–108.
9. J. Heinz. Polynom-Nulstellen mit dem Rechenstab. *Prax. Math* **5**, 1963, 97–99.
10. L. Ilieff. On the approximations of Newton. *Annual of the Sofia University* **46**, 1948-1950, 167–171 (in Bulgarian).
11. L. Ilieff. Laguerre entire functions. Sofia, Publishing house of the Bulgarian Academy of Sciences, 1987.
12. L.Ilieff, K. Dočev. Über Newtonsche Iterationen. *Wiss. Z. Techn. Univ. Dresden*, **12**, 1963, 117–118.
13. I.O. Kerner. Ein Gesamtschrittverfahren zur Berechnung der Nullstellen von Polynomen. *Numer. Math.* **8**, 1966, 290–294.
14. D.E. Knuth. The art of computer programming. **2**. Seminumerical Algorithms, London Don Mills Ontario: Addison- Wesley Publishing Company, 1969.
15. J.-L. Lagouanell. Sur une methode de calcule de l'ordre de multiplicité des zéros d'un polynome. *C.R. Acad. Sci.***AB 262**. Paris, A626-A627, 1966.
16. G. Loizou. Une note sur le procédé itératif de M Marica D. Preic. *C.R. Acad. Sci.* Paris, **295**, 1982, 707–710.

17. I.V. Makrelov. On a numerical method for simultaneous finding all zeros of a given trigonometric polynomial. Travaux scientifiques -mathematiques, Universite de Plovdiv **17**, 1979, 205–218 (in Bulgarian).

18. I.V. Makrelov, Khr. Semerdzhiev. Methods for simultaneous finding all roots of algebraic, trigonometric and exponential equations if their multiplicities are given. *J. of comp. math. and math. phys.* **24**, 1984, 1443–1453 (in Russian).

19. I.V. Makrelov, Khr. Semerdzhiev. On two analogues of Ehrlich's method for simultaneous finding all zeros of trigonometric and exponential polynomials. *C.R. of Bulg. Acad. of Sci.* **36**, 1983, 879–882 (in Russian).

20. I.V. Makrelov, Khr. Semerdzhiev. Dochev's method for a generalized polynomial over an arbitrary Chebyshev system. *C.R. Bulg. Acad. of Sci.* **38**, 1985, 1323–1326 (in Russian).

21. I.V. Makrelov, Khr. Semerdzhiev, S.G. Tamburov. On a generalization of Ehrlich's method. *C.R. of Bulg. Acad. of Sci.* **39**, 1986, 43–46 (in Russian).

22. I.V. Makrelov, Khr. Semerdzhiev, S.G. Tamburov. A method for simultaneous finding all zeros of a given generalized polynomial over a Chebyshev system. *Comm. of JINR*, Dubna, P11-85-932, 1985 (in Russian).

23. D.G. Moursund. Examination of multiple roots and root clusters of a polynomial using the Bernoulli procedure. *J. of ACM* **12**, 1965, 169–174.

24. N. Obreshkoff. Higher Algebra. Sofia, Nauka i izkustvo 1958 (in Bulgarian).

25. N. Obreshkoff. On the numerical solving equations. *Annual of the Sofia University* **56**, 1963, 73–83 (in Bulgarian).

26. P.O. Oya. On the method of Newton in the case of multiple solutions. *Sci. works of Tartu University* **633**, 1983, 88–94.

27. Khr. Semerdzhiev. Methods for simultaneous approximative determining all roots of a given algebraic equation. *Comm. of JINR*, Dubna, P5-12485, 1979 (in Russian).

28. Khr. Semerdzhiev. A method for simultaneous finding all roots of an algebraic equation if their multiplicities are given. *C.R. Acad. Sci.* **35**, 1982, 1057–1060 (in Russian).

29. Khr. Semerdzhiev, E. Angelova. On a numerical method for simultaneous finding all zeros of a given trigonometric polynomial. Collection of works of young scientific workers and students, University of Plovdiv 6, 1978, 127–136, (in Bulgarian).

30. Khr. Semerdzhiev, S.G. Tamburov. A method for determining all zeros of a generalized polynomial over an arbitrary Chebyshev system. *Comm. of JINR*, Dubna, P11-85-931 1985 (in Russian).

31. Khr. Semerdzhiev, S.G. Tamburov. On a method for determining the multiplicities of the zeros of an algebraic polynomial. *C.R. Bulg. Acad. Sci.* **37**, 1984, 1143–1145 (in Russian).

32. Bl. Sendov, V. Popov. Numerical Methods. Sofia, Nauka i izkustvo 1976 (in Bulgarian).

33. K. Weierstrass. Neuer Bewweis des Satzes, dass jede ganze rationale Funktion einer Veranderlichen dargestellt werden kann als Produkt aus linearen Funktionen derselben Veranderlichen, Ges. Werke 3, 1901, 251–269.

34. E.P. Zhidkov, I.V. Makrelov, Khr. Semerdzhiev. Two methods for simultaneous finding all roots of exponential equations. *Comm. of JINR*, Dubna, 1983 (in Russian).

Faculty of Mathematics and Informatics
University of Plovdiv
Plovdiv 4000
Tzar Assen str. 24
BULGARIA

Received 28.12.1992