Aplikace matematiky

Josef Dvorčuk

Factorization of a polynomial into quadratic factors by Newton method

Aplikace matematiky, Vol. 14 (1969), No. 1, 54–80

Persistent URL: http://dml.cz/dmlcz/103207

Terms of use:

© Institute of Mathematics AS CR, 1969

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.

This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* http://dml.cz

FACTORIZATION OF A POLYNOMIAL INTO QUADRATIC FACTORS BY NEWTON METHOD

JOSEF DVORČUK

(Received December 27, 1967)

INTRODUCTION

In this paper the method for simultaneous finding of all the roots of a polynomial is derived. The method is based on the factorization of a polynomial into quadratic factors.

The method is designed for the polynomial with real coefficients. It is derived by using the Newton method.

The quadratic convergence of this method is proved for the given good guesses of the roots and for the polynomial with distinct roots.

Algorithm of the method is described in Algol 60.

1. Description of the method. The method performs the factorization of the polynomial

$$f(z) = \sum_{i=0}^{n} a_i z^{n-i} \quad (a_0 \neq 0) \tag{1.1}$$

into quadratic factors i.e.

$$f(z) = a_0 \prod_{j=1}^{m} (z^2 + p_j z + q_j)\,, \tag{1.2}$$

where a_i are real and n is even i.e.

$$n = 2m\,. \tag{1.3}$$

The factorization (1.2) with real coefficients p_j, q_j $(j = 1, 2, \ldots, m)$ exists for real a_i $(i = 0, 1, 2, \ldots, n)$. Hence for polynomials with real coefficients we can perform the calculation in the real arithmetics only.

The method is based on formulas

$$(1.4)\qquad p_i^{(k+1)} = p_i^{(k)} + \sum_{j=0}^{n} a_j u_i^{(n-j)}, \quad q_i^{(k+1)} = q_i^{(k)} + \sum_{j=0}^{n} a_j v_i^{(n-j)} \quad i = 1, 2, \ldots, m$$

for $k = 0, 1, 2, \ldots$ with given approximations

$$(1.5)\qquad p_i^{(0)}, q_i^{(0)} \quad i = 1, 2, \ldots, m,$$

where for $u_i^{(j)}, v_{(j)}^i$ relations hold as follows

$$(1.6)\qquad \begin{aligned} u_i^{(0)} &= -D_i r_i^{(m)} \\ u_i^{(1)} &= D_i s_i^{(m)} \\ u_i^{(j+2)} &= -p_i^{(k)} u_i^{(j+1)} - q_i^{(k)} u_i^{(j)} \quad j = 0, 1, \ldots, n-2 \end{aligned}$$

$$(1.7)\qquad \begin{aligned} v_i^{(0)} &= D_i(s_i^{(m)} - p_i^{(k)} r_i^{(m)}) \\ v_i^{(1)} &= D_i q_i^{(k)} r_i^{(m)} \\ v_i^{(j+2)} &= -p_i^{(k)} v_i^{(j+1)} - q_i^{(k)} v_i^{(j)} \quad j = 0, 1, \ldots, n-2, \end{aligned}$$

where

$$(1.8)\qquad D_i = a_0^{-1}\left[(s_i^{(m)})^2 - p_i^{(k)} r_i^{(m)} s_i^{(m)} + q_i^{(k)}(r_i^{(m)})^2\right]^{-1}$$

and $r_i^{(m)}, s_i^{(m)}$ are obtained by recurrent formulas

$$r_i^{(0)} = 0, \quad s_i^{(0)} = 1$$

$$(1.9)\qquad r_i^{(j)} = \begin{cases} \left[q_j^{(k)} - q_i^{(k)} - p_i^{(k)} \cdot (p_j^{(k)} - p_i^{(k)})\right] r_i^{(j-1)} + (p_j^{(k)} - p_i^{(k)})\, s_i^{(j-1)} \text{ for } j \neq i \\ r_i^{(j-1)} \text{ for } j = i \end{cases}$$

$$s_i^{(j)} = \begin{cases} -q_i^{(k)} \cdot (p_j^{(k)} - p_i^{(k)})\, r_i^{(j-1)} + (q_j^{(k)} - q_i^{(k)})\, s_i^{(j-1)} \text{ for } j \neq i \\ s_i^{(j-1)} \text{ for } j = i \end{cases}$$

for $j = 1, 2, \ldots, m$.

2. DERIVATION OF THE METHOD

The method is derived by using the Newton method for the system of equations expressing the relationship between coefficients p_j, q_j $(j = 1, 2, \ldots, m)$ and a_i $(i = 0, 1, \ldots, n)$ in the factorization (1.2).

We shall consider (1.2) in the form

$$(2.1)\qquad \sum_{i=0}^{n} a_i z^{n-i} = a_0 \prod_{j=1}^{m} (z^2 + x_j z + y_j) =$$

$$= a_0\left(z^n + \sum_{i=1}^{n} h_i(x_1, y_1, x_2, \ldots, y_m)\, z^{n-i}\right)$$

where h_i are functions of variables x_j, y_j $(j = 1, 2, \ldots, m)$. Hence follows the system of equations, which will be written in vector form:

$$a_0\mathbf{h}(x_1, y_1, \ldots, y_m) - \mathbf{a} = \mathbf{0}\,, \tag{2.2}$$

where

$$\mathbf{h} = \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ y_1 \\ \vdots \\ y_m \end{pmatrix}, \quad \mathbf{0} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{2.3}$$

Using the matrices

$$\mathbf{R}_i^{(r)} = \underbrace{\left.\begin{bmatrix} y_i & x_i & 1 & 0 & \ldots & 0 \\ 0 & y_i & x_i & 1 & \ldots & 0 \\ 0 & 0 & y_i & x_i & \ldots & 0 \\ \multicolumn{6}{c}{\dotfill} \\ 0 & 0 & 0 & 0 & \ldots & y_i \end{bmatrix}\right\} r \text{ rows}}_{r \text{ columns}} \tag{2.4}$$

i.e. $r \times r$ matrices with y_i on the diagonal and with x_i and 1 above, (2.3) can be rewritten as follows

$$\mathbf{h} = \prod_{j=1}^{m} \mathbf{R}_j^{(n)} \mathbf{e}_n\,, \tag{2.5}$$

where $\mathbf{e}_n$ is the last column of the $n \times n$ unit matrix.

Newton method gives

$$\delta\mathbf{x} = -\mathbf{A}^{-1}(a_0\mathbf{h} - \mathbf{a})\,, \tag{2.6}$$

where

$$\mathbf{A} = a_0 \begin{bmatrix} \dfrac{\partial h_1}{\partial x_1} & \dfrac{\partial h_1}{\partial y_1} & \ldots & \dfrac{\partial h_1}{\partial y_m} \\ \multicolumn{4}{c}{\dotfill} \\ \dfrac{\partial h_n}{\partial x_1} & \dfrac{\partial h_n}{\partial y_1} & \ldots & \dfrac{\partial h_n}{\partial y_m} \end{bmatrix}, \quad \delta\mathbf{x} = \begin{pmatrix} \delta x_1 \\ \delta y_1 \\ \vdots \\ \delta y_m \end{pmatrix}, \tag{2.7}$$

According to

$$\mathbf{R}_i^{(r)}\mathbf{R}_j^{(r)} = \mathbf{R}_j^{(r)}\mathbf{R}_i^{(r)}$$

for any i and j we get for $k = 1, 2, \ldots, m$

$$(2.8)\qquad \begin{cases} \dfrac{\partial \mathbf{h}}{\partial y_k} = \prod\limits_{\substack{j=1 \\ j \neq k}}^{m} \mathbf{R}_j^{(n)} \mathbf{e}_n \\ \dfrac{\partial \mathbf{h}}{\partial x_k} = \left(\prod\limits_{\substack{j=1 \\ j \neq k}}^{m} \mathbf{R}_j^{(n)}\right) \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & & 0 \end{pmatrix} \mathbf{e}_n = \prod\limits_{\substack{j=1 \\ j \neq k}}^{m} \mathbf{R}_j^{(n)} \bar{\mathbf{e}}_n , \end{cases}$$

where $\bar{\mathbf{e}}_n$ is the last but one column of $n \times n$ unit matrix.

We determine the inverse matrix $\mathbf{A}^{-1}$ from the equation

$$\mathbf{A}^{-1} . \mathbf{A} = \mathbf{I} .$$

According to (2.8) we have to distinguish the even and odd rows of the matrix $\mathbf{A}^{-1}$; thus we get the systems of equations for the i-th odd row:

$$(2.9)\qquad \mathbf{w}_{2i-1}^T = (u_i^{(n-1)}, u_i^{(n-2)}, \ldots, u_i^{(0)})$$

in this form (the upper index T means transposition)

$$(2.10)\qquad \begin{cases} a_0 \bar{\mathbf{e}}_n^T \left(\prod\limits_{\substack{j=1 \\ j \neq k}}^{m} \mathbf{R}_j^{(n)}\right)^T \mathbf{w}_{2i-1} = 0 \\ a_0 \mathbf{e}_n^T \left(\prod\limits_{\substack{j=1 \\ j \neq k}}^{m} \mathbf{R}_j^{(n)}\right)^T \mathbf{w}_{2i-1} = 0 \end{cases} k = 1, 2, \ldots, m, \; k \neq i$$

$$a_0 \bar{\mathbf{e}}_n^T \left(\prod_{\substack{j=1 \\ j \neq i}}^{m} \mathbf{R}_j^{(n)}\right)^T \mathbf{w}_{2i-1} = 1$$

$$a_0 \mathbf{e}_n^T \left(\prod_{\substack{j=1 \\ j \neq i}}^{m} \mathbf{R}_j^{(n)}\right)^T \mathbf{w}_{2i-1} = 0 .$$

If we denote the matrix of the system (2.10) by $\mathbf{B}_i$ i.e.

$$(2.11)\qquad \mathbf{B}_i \mathbf{w}_{2i-1} = \bar{\mathbf{e}}_n$$

then analogously for the i-th even row

$$(2.12)\qquad \mathbf{w}_{2i}^T = (v_i^{(n-1)}, v_i^{(n-2)}, \ldots, v_i^{(0)})$$

we get the system in the form

$$(2.13)\qquad \mathbf{B}_i \mathbf{w}_{2i} = \mathbf{e}_n$$

Lemma 2.1. *The solution of the system of linear algebraic equations*

$$\mathbf{C}_i \mathbf{w}_{2i-1} = \bar{\mathbf{e}}_n \tag{2.14}$$

$$(\mathbf{C}_i \mathbf{w}_{2i} = \mathbf{e}_n \text{ respectively}) \tag{2.15}$$

is the solution of (2.11), (2.13) *respectively, the* $(n-2)$ *upper rows of matrix* $\mathbf{C}_i$ *being formed by* $(n-2) \times n$ *matrix*

$$\bar{\mathbf{C}}_i = a_0 \begin{Bmatrix} 1 & x_i & y_i & 0 & \dots & 0 \\ 0 & 1 & x_i & y_i & \dots & 0 \\ 0 & 0 & 1 & x_i & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & y_i \end{Bmatrix} \tag{2.16}$$

and the last two rows of $\mathbf{C}_i$ *being identical with the last two rows of matrix* $\mathbf{B}_i$

Proof. Let us denote the part of matrix $\mathbf{B}_i$ without the last two rows by $\bar{\mathbf{B}}_i$. Then it holds

$$\bar{\mathbf{B}}_i = \frac{1}{a_0} \mathbf{A}_i^T \bar{\mathbf{C}}_i , \tag{2.17}$$

where matrices $\mathbf{A}_i$ have an analogous structure as $\mathbf{A}$ (see (2.7) and (2.8)); however, they are $(n-2) \times (n-2)$ matrices and are generated by matrices

$$\mathbf{R}_j^{(n-2)}, \quad j = 1, 2, \dots, m, \quad j \neq i$$

i.e. (2.17) implies that the rows of matrix $\bar{\mathbf{B}}_i$ are linear combinations of rows of $\bar{\mathbf{C}}_i$ which proves Lemma 2.1.

Remark 2.1. The solution of the system (2.14) ((2.15) respectively) forms the i-th odd (even respectively) column of matrix $(\mathbf{A}^{-1})^T$ and so the sufficient condition of the existence of the solution of systems (2.14) and (2.15) for $i = 1, 2, \dots, m$ is a sufficient condition of the existence of $\mathbf{A}^{-1}$.

Lemma 2.2. *For the solution of systems* (2.14) ((2.15) respectively) *the following relationships hold*

$$u_i^{(j+2)} = -x_i u_i^{(j+1)} - y_i u_i^{(j)} \tag{2.18}$$

$$(v_i^{(j+2)} = -x_i v_i^{(j+1)} - y_i v_i^{(j)}, \quad \text{respectively}) \quad j = 0, 1, \dots, n-2$$

$$u_i^{(1)} = \Delta_i^{-1} s_i^{(m)}, \quad u_i^{(0)} = -\Delta_i^{-1} r_i^{(m)}$$

$$(v_i^{(1)} = \Delta_i^{-1} y_i r_i^{(m)}, \quad v_i^{(0)} = \Delta_i^{-1}(s_i^{(m)} - x_i r_i^{(m)}) \quad \text{respectively})$$

where

$$\Delta_i = a_0[(s_i^{(m)})^2 - x_i s_i^{(m)} r_i^{(m)} y_i (r_i^{(m)})^2] \tag{2.19}$$

and where $r_i^{(m)}$, $s_i^{(m)}$ *are obtained by a recurrent process for* $j = 1, 2, \ldots, m$

$$r_i^{(0)} = 0\,, \quad s_i^{(0)} = 1$$

$$r_i^{(j)} = \begin{cases} [y_j - y_i - x_i(x_j - x_i)]\, r_i^{(j-1)} + (x_j - x_i)\, s_i^{(j-1)} & \text{for } j \neq i \\ r_i^{(j-1)} & \text{for } j = i \end{cases} \tag{2.20}$$

$$s_i^{(j)} = \begin{cases} -y_i(x_j - x_i)\, r_i^{(j-1)} + (y_j - y_i)\, s_i^{(j-1)} & \text{for } j \neq i \\ s_i^{(j-1)} & \text{for } j = i \end{cases}$$

Proof. Systems (2.14), (2.15) are solved by the method of Gauss elimination.

The last two equations are changed by substraction of the multiples of the upper equations of the equations

$$a_0 \begin{pmatrix} r_i & s_i & 0 \\ 0 & r_i & s_i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \left(\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \text{respectively} \right). \tag{2.21}$$

Let us note that for example the last equation of (2.14) is formed by the vector

$$a_0 \mathbf{e}_n^T \Big(\prod_{\substack{j=1 \\ j \neq i}}^{m} \mathbf{R}_j^{(n)} \Big)^T$$

i.e. by the coefficients of polynomial

$$g_i(z) = a_0 \prod_{\substack{j=1 \\ j \neq i}}^{m} (z^2 + x_j z + y_j)\,. \tag{2.22}$$

Thus dividing the polynomial $g_i(z)$ by $a_0(z^2 + x_i z + y_i)$ we get for r_i, s_i the relationship

$$g_i(z) = a_0[t(z)\,(z^2 + x_1 z + y_i) + r_i z + s_i]\,. \tag{2.23}$$

From (2.23) follows the recurrent process (2.20) which will be proved by induction.

If $1 \neq i$ then it obviously follows

$$a_0(z^2 + x_1 z + y_1) = a_0[z^2 + x_i z + y_i + (x_1 - x_i)\, z + (y_1 - y_i)]\,.$$

Further let $j \neq i$ and let

$$a_0 \prod_{\substack{k=1 \\ k \neq i}}^{j-1} (z^2 + x_k z + y_k) = a_0[t_{j-1}(z)\,.\,(z^2 + x_i z + y_i) + r_i^{(j-1)} z + s_i^{(j-1)}]\,.$$

If the last identity is multiplied by $z^2 + x_j z + y_j$ and slightly modified, then we get

$$a_0 \prod_{\substack{k=1 \\ k \neq i}}^{j} (z^2 + x_k z + y_k) = a_0 \{ [(z^2 + x_j z + y_j)\, t_{j-1}(z) + r_i^{(j-1)} z + s_i^{(j-1)} + \\ + (x_j - x_i)\, r_i^{(j-1)}]\, (z^2 + x_i z + y_i) + \\ + \{[y_j - y_i - x_i(x_j - x_i)]\, r_i^{(j-1)} + (x_j - x_i)\, s_i^{(j-1)}\}\, z + \\ + [-y_i(x_j - x_i)\, r_i^{(j-1)} + (y_j - y_i)\, s_i^{(j-1)}]\}$$

which after the comparison with the equation (2.23) gives the relationship (2.20) and so it holds

$$r_i = r_i^{(m)}, \quad s_i = s_i^{(m)}.$$

The process of calculation of coefficients of the system (2.21) is derived.

We complete the system (2.21) to

$$a_0 \begin{pmatrix} 1 & x_i & y_i \\ r_i & s_i & 0 \\ 0 & r_i & s_i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \left(\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \text{respectively} \right). \tag{2.24}$$

After solving (2.24) and due to the fact that other equations of the system (2.14), (2.15) are

$$u_i^{(j+2)} + x_i u_i^{(j+1)} + y_i u_i^{(j)} = 0, \quad v_i^{(j+2)} + x_i v_i^{(j+1)} + y_i v_i^{(j)} = 0 \text{ respectively.} \tag{2.25}$$

Thus the proof of lemma 2.2 is completed.

In (2.6) there is the expression

$$-a_0 \mathbf{A}^{-1} \mathbf{h}. \tag{2.26}$$

The calculation of this term is solved by lemma 2.3.

Lemma 2.3. *For the solution of system* (2.14), (2.15) *by means of the expressions* (2.9), (2.12) *respectively it holds*

$$\mathbf{w}_{2i-1}^T \mathbf{h} = x_i u_i^{(n-1)} + y_i u_i^{(n-2)}, \quad \mathbf{w}_{2i}^T \mathbf{h} = x_i v_i^{(n-1)} + y_i v_i^{(n-2)} \textit{ respectively.} \tag{2.27}$$

Proof. Let us denote

$$u_i^{(n)} = -x_i u_i^{(n-1)} - y_i u_i^{(n-2)}, \quad v_i^{(n)} = -x_i v_i^{(n-1)} - y_i v_i^{(n-2)} \tag{2.28}$$

and

$$\overline{\mathbf{w}}_{2i-1}^T = (u_i^{(n)}, u_i^{(n-1)}, \ldots, u_i^{(0)}), \quad \overline{\mathbf{w}}_{2i}^T = (v_i^{(n)}, v_i^{(n-1)}, \ldots, v_i^{(0)});$$

then it holds

$$\overline{\mathbf{w}}_{2i-1}^T\Big(\prod_{k=1}^{m}\mathbf{R}_k^{(n+1)}\Big)\,\mathbf{e}_{n+1}=0\,,\quad \overline{\mathbf{w}}_{2i}^T\Big(\prod_{k=1}^{m}\mathbf{R}_k^{(n+1)}\Big)\,\mathbf{e}_{n+1}=0 \tag{2.29}$$

because with regard to (2.25) and (2.28)

$$\begin{aligned}\overline{\mathbf{w}}_{2i-1}^T\mathbf{R}_i^{(n+1)} &= (r_1, r_2, 0, 0, \ldots, 0)\\ \overline{\mathbf{w}}_{2i}^T\mathbf{R}_i^{(n+i)} &= (r_3, r_4, 0, 0, \ldots, 0)\end{aligned} \tag{2.30}$$

i.e. in the product (2.30) not more than the first two components differ from zero.

If this row vector is multiplied again from the right side by the matrix $\mathbf{R}_j^{(n+1)}$ then the number of nonzero components does not increase more than twice. After multiplying by all matrices from (2.29), at most the first n components are different from zero. However, when multiplying from the right side by vector $\mathbf{e}_{n+1}$ only the $(n+1)$-th element of vector is essential which gives zero result.

Let us realize that there is one difference only between the vectors

$$\prod_{k=1}^{m}\mathbf{R}_k^{(n+1)}\mathbf{e}_{n+1}\,,\quad \prod_{k=1}^{m}\mathbf{R}_k^{(n)}\mathbf{e}_n\,.$$

Namely, the upper component of the left vector equals 1, and the other n components of this vector being identical with all n components of the right vector.

It further follows

$$\begin{aligned}\overline{\mathbf{w}}_{2i-1}^T\prod_{k=1}^{m}\mathbf{R}_k^{(n+1)}\mathbf{e}_{n+1} &= u_i^{(n)} + \mathbf{w}_{2i-1}^T\prod_{k=1}^{m}\mathbf{R}_k^{(n)}\mathbf{e}_n\,,\\ \overline{\mathbf{w}}_{2i}^T\prod_{k=1}^{m}\mathbf{R}_k^{(n+1)}\mathbf{e}_{n+1} &= v_i^{(n)} + \mathbf{w}_{2i}^T\prod_{k=1}^{m}\mathbf{R}_k^{(n)}\mathbf{e}_n\,.\end{aligned}$$

The assertion of lemma 2.3 follows according to (2.5), (2.29) and (2.28)

Remark 2.2. If solutions exist of all systems (2.14) and (2.15) for $i = 1, 2, \ldots, m$, then there exists an inverse matrix $\mathbf{A}^{-1}$. Formula (2.6) assumes the form (1.4) by substituting

$$x_i = p_i^{(k)}\,,\quad y_i = q_i^{(k)}\,.$$

Lemma 2.3 simplifies the calculation considerably. The following lemma 2.4 will deal with the existence of solutions of all the systems (2.14) and (2.15).

First of all let us denote

$$\lambda_j, \mu_j$$

the roots of polynomials

$$z^2 + x_j z + y_j \quad j = 1, 2, \ldots, m$$

Lemma 2.4. *There is one and only one solution of systems* (2.14) *and* (2.15) *if and only if*

$$\lambda_1 \neq \lambda_j, \mu_j\,, \quad \mu_j \neq \lambda_j, \mu_j\,, \quad j = 1, 2, \ldots, m, j \neq i$$

Proof. The necessary and sufficient condition of the existence of the only solution of systems (2.14) and (2.15) is

$$\det \mathbf{C}_i \neq 0\,.$$

According to the considerations of the proof of lemma 2.2 it follows that

$$\det \mathbf{C}_i = \Delta_i a_0^{n-1}\,.$$

If function $g_i(z)$ (see (2.22)) and a relationship (2.23) is used, we get

$$g_i(\lambda_i) = a_0(r_i\lambda_i + s_i)\,, \quad g_i(\mu_i) = a_0(r_i\mu_i + s_i) \tag{2.31}$$

and further from (2.19)

$$\begin{aligned} \Delta_i &= a_0(s_i^2 + (\lambda_i + \mu_i)\, s_i r_i + \lambda_i\mu_i r_i^2) = \\ &= a_0(r_i\lambda_i + s_i)\,(r_i\mu_i + s_i) = a_0^{-1}\, g_i(\lambda_i)\, g_i(\mu_i)\,. \end{aligned} \tag{2.32}$$

With regard to

$$g_i(z) = a_0 \prod_{\substack{j=1 \\ j\neq i}}^{m} (z - \lambda_i)\,(z - \mu_j) \tag{2.33}$$

we get

$$\det \mathbf{C}_i = a_0^n \prod_{\substack{j=1 \\ j\neq i}}^{m} (\lambda_i - \lambda_j)\,(\lambda_i - \mu_j)\,(\mu_i - \lambda_j)\,(\mu_i - \mu_j)$$

Hence the assertion of lemma 2.4 follows.

Theorem 2.1. *The inverse matrix* $\mathbf{A}^{-1}$ *from* (2.6) *exists if and only if*

$$\lambda_i \neq \lambda_j, \mu_j\,, \quad \mu_i \neq \lambda_j, \mu_j \quad j = 1, 2, \ldots, m, \; j \neq i$$

for $i = 1, 2, \ldots, m$.

Proof. The sufficient condition follows from Lemma 2.4, 2.1 and Remark 2.1.

It is sufficient to prove the necessary condition. For the existence of $\mathbf{A}^{-1}$ it is necessary that

$$\det \mathbf{A} \neq 0\,.$$

The rows of the matrix $\mathbf{A}$ have to be linearly independent. Therefore all the rows

of the matrix $\bar{\mathbf{B}}_i$ (see (2.17)) have to be linearly independent. So there exists an $(n-2) \times (n-2)$ matrix $\tilde{\mathbf{B}}_i$ formed by omitting two columns of $\bar{\mathbf{B}}_i$, such that

$$\det \tilde{\mathbf{B}}_i \neq 0 . \tag{2.34}$$

The matrix $\tilde{\mathbf{B}}_i$ could be expressed in the form (see (2.17))

$$\tilde{\mathbf{B}}_i = \frac{1}{a_0} \mathbf{A}_i^T \tilde{\mathbf{C}}_i \tag{2.35}$$

where $\tilde{\mathbf{C}}_i$ is formed from $\mathbf{C}_i$ in the same way as $\tilde{\mathbf{B}}_i$ from $\bar{\mathbf{B}}_i$. From (2.34) and (2.35) we get, however, that necessarily

$$\det \mathbf{A}_i \neq 0 , \quad i = 1, 2, \ldots, m .$$

If we go on decreasing the order of matrices we get

$$D_{ij} = \begin{vmatrix} y_i & 0 & y_j & 0 \\ x_i & y_i & x_j & y_j \\ 1 & x_i & 1 & x_j \\ 0 & 1 & 0 & 1 \end{vmatrix} \neq 0$$

for any j and i as well as

$$\begin{aligned} D_{ij} &= (y_j - y_i)^2 + x_i(y_j - y_i)(x_i - x_j) + y_i(x_i - x_j)^2 = \\ &= [y_j - y_i - \lambda_i(x_i - x_j)][y_j - y_i - \mu_i(x_i - x_j)] = \\ &= (\lambda_i - \lambda_j)(\lambda_i - \mu_j)(\mu_i - \lambda_j)(\mu_i - \mu_j) \end{aligned}$$

for $j \neq i$ and hence theorem 2.1. is proved.

Remark 2.3. Let us consider that the approximations $p_i^{(k)}, q_i^{(k)}$, $i = 1, 2, \ldots, m$ are given for the factorisation and let us denote $\lambda_i^{(k)}, \mu_i^{(k)}$ the roots of polynomials

$$z^2 + p_i^{(k)} z + q_i^{(k)} , \quad i = 1, 2, \ldots, m .$$

The following theorem will show the possibility of the existence of the next approximations

$$p_i^{(k+1)}, q_i^{(k+1)}$$

constructed by use of Lemma 2.2.

Theorem 2.2. *Let the approximations*

$$p_i^{(k)}, q_i^{(k)} , \quad i = 1, 2, \ldots, m$$

be such that the roots

$$\lambda_1^{(k)}, \mu_1^{(k)}, \lambda_2^{(k)}, \ldots, \mu_m^{(k)}$$

are distinct.

Than the Newton method in the form (2.6) *gives the following approximations*

$$p_i^{(k+1)} = p_i^{(k)} + \delta p_i^{(k)}, \quad q_i^{(k+1)} = q_i^{(k)} + \delta q_i^{(k)}, \quad i = 1, 2, \ldots, m$$

determined by formulas (1.4), (1.6), (1.7), (1.8) *and* (1.9).

Proof. The theorem follows from the above mentioned lemmas by the substitution

$$x_i = p_i^{(k)}, \quad y_i = q_i^{(k)}, \quad i = 1, 2, \ldots, m.$$

According to the presumptions of Theorem 2.2 and Lemma 2.4 the inverse matrix in formula (2.6) of the Newton method exists. Using Lemma 2.2 and 2.3 we get that

$$\delta p_i^{(k)}, \delta q_i^{(k)}$$

exist and are expressed in the form

$$\delta p_i^{(k)} = \sum_{j=0}^{n} a_j u_i^{(n-j)}, \quad \delta q_i^{(k)} = \sum_{j=0}^{n} a_j v_i^{(n-j)}$$

This proves theorem 2.2.

Remark 2.4. Let us realize that Theorem 2.2 does not assume the distinction of the roots of $f(z)$. It requires the distinction of the approximation of those roots only.

Further we have to realize that Theorem 2.2 confirms only the existence of the next approximations

$$p_i^{(k+1)}, q_i^{(k+1)}$$

To obtain further approximations $p_i^{(k+2)}$, $q_i^{(k+2)}$ we need some other applications of this theorem.

3. CONVERGENCE OF THE METHOD

First of all we shall derive formulas equivalent to (2. 8), (2.19) and (2.20).

We define

$$\lambda_{m+i} = \mu_i \quad \text{for} \quad i = 1, 2, \ldots, m. \tag{3.1}$$

Lemma 3.1. *If* $\lambda_i \neq \lambda_{i+m}$ *and if the solutions of all the systems* (2.14) *and* (2.15) *exist then we get from* (2.6)

$$\delta x_i = \frac{f(\lambda_i)}{a_0 \prod_{\substack{j=1 \\ j \neq i}}^{n} (\lambda_i - \lambda_j)} + \frac{f(\lambda_{i+m})}{a_0 \prod_{\substack{j=1 \\ j \neq i+m}}^{n} (\lambda_{i+m} - \lambda_j)} \tag{3.2}$$

$$\delta y_i = -\frac{\lambda_{i+m} f(\lambda_i)}{a_0 \prod_{\substack{j=0 \\ j \neq i}}^{n} (\lambda_i - \lambda_j)} - \frac{\lambda_i f(\lambda_{i+m})}{a_0 \prod_{\substack{j=1 \\ j \neq i+m}}^{n} (\lambda_{i+m} - \lambda_j)} \tag{3.3}$$

Proof.

With regard to

$$u_i^{(j+2)} + x_i u_i^{(j+1)} + y_i u_i^{(j)} = 0\,, \quad v_i^{(j+2)} + x_i v_i^{(j+1)} + y_i v_i^{(j)} = 0$$

$u_i^{(j)}$, $v_i^{(j)}$ are the solutions of the following difference equation

$$u_{k+1} + x_i u_k + y_i u_{k-1} = 0$$

with the coefficients x_i, y_i, and the initial values

$$u_i^{(0)}, u_i^{(1)} \,(v_i^{(0)}, v_i^{(1)} \text{ respectively})\,.$$

However, it means that for $\lambda_i \neq \lambda_{i+m}$

$$u_i^{(j)} = b_1 \lambda_i^j + c_1 \lambda_{i+m}^j\,, \quad v_i^{(j)} = b_2 \lambda_i^j + c_2 \lambda_{i+m}^j\,, \quad j = 0, 1, \ldots, n$$

the coefficients b_1, c_1, b_2, c_2 being obtained from the initial values.

From (2.31) we get

$$r_i = \frac{g_i(\lambda_i) - g_i(\lambda_{i+m})}{a_0(\lambda_i - \lambda_{i+m})}$$

$$s_i = \frac{\lambda_{i+m} g_i(\lambda_i) - \lambda_i g_i(\lambda_{i+m})}{a_0(\lambda_{i+m} - \lambda_i)}\,,$$

from (2.18) using (2.32)

$$b_1 = \frac{1}{(\lambda_i - \lambda_{i+m}) g_i(\lambda_i)}\,, \quad c_1 = \frac{1}{(\lambda_{i+m} - \lambda_i) g_i(\lambda_{i+m})}$$

$$b_2 = -\frac{\lambda_{i+m}}{(\lambda_i - \lambda_{i+m}) g_i(\lambda_i)}\,, \quad c_2 = -\frac{\lambda_i}{(\lambda_{i+m} - \lambda_i) g_i(\lambda_{i+m})}$$

and if we use (2.33) we derive (3.2) and (3.3).

Remark 3.1. We shall not consider the case $\lambda_i = \lambda_{i+m}$ because it would require a modification of the method. This case will not be considered in the algorithm either.

Theorem 3.1. *Let the polynomial* (1.1) *have distinct roots* $z_1, z_2, \ldots, z_n$. *Let us denote* $d = \min_{i \neq j} |z_i - z_j|$. *Let the approximations* $p_i^{(0)}, q_i^{(0)}$, $i = 1, 2, \ldots, m$ *of the coefficients in the factorization* (1.2) *be given such that the roots* $\lambda_1^{(0)}, \lambda_2^{(0)}, \ldots, \lambda_n^{(0)}$ *of the polynomials*

$$z^2 + p_i^{(0)} z + q_i^{(0)}, \quad i = 1, 2, \ldots, m$$

fulfil

$$|\lambda_i^{(0)} - z_i| \leq \frac{1 - (1 + q)^{1/(n-1)}}{1 - 2(1 + q)^{1/(n-1)}} d, \tag{3.4}$$

where

$$0 < q < (8 \cdot 2^{1/(n-1)} - 7)^{-1}. \tag{3.5}$$

Then the sequences

$$\{p_i^{(k)}\}_{k=0}^{\infty} \{q_i^{(k)}\}_{k=0}^{\infty} \quad i = 1, 2, \ldots, m$$

determined by (1.4) *are quadratically convergent.*

Proof.

We shall perform the numbering of the roots $\lambda_i^{(0)}, \lambda_{i+m}^{(0)}$ of the polynomials

$$z^2 + p_i^{(0)} z + q_i^{(0)}, \quad i = 1, 2, \ldots, m.$$

We denote $\lambda_i^{(k)}, \lambda_{i+m}^{(k)}$ the roots of $z^2 + p_i^{(k)} z + q_i^{(k)}$,

$$F_i^{(k)} = a_0^{-1} f(\lambda_i^{(k)}) / \Big(\prod_{\substack{j=1 \\ j \neq i}}^{n} (\lambda_i^{(k)} - \lambda_j^{(k)}) \Big) \tag{3.6}$$

and

$$z_i^{(k)} = \lambda_i^{(k-1)} - F_i^{(k-1)}.$$

Thus

$$z_i^{(k)} + z_{i+m}^{(k)} = -p_i^{(k)} = \lambda_i^{(k)} + \lambda_{i+m}^{(k)}$$

$$z_i^{(k)} z_{i+m}^{(k)} - F_i^{(k-1)} F_{i+m}^{(k-1)} = q_i^{(k)} = \lambda_i^{(k)} \lambda_{i+m}^{(k)}$$

and

$$\lambda_i^{(k)} = 0{\cdot}5\{z_i^{(k)} + z_{i+m}^{(k)} + \sqrt{[(z_i^{(k)} - z_{i+m}^{(k)})^2 + 4F_i^{(k-1)} F_{i+m}^{(k-1)}]}\}$$

$$\lambda_{i+m}^{(k)} = 0{\cdot}5\{z_i^{(k)} + z_{i+m}^{(k)} - \sqrt{[(z_i^{(k)} - z_{i+m}^{(k)})^2 + 4F_i^{(k-1)} F_{i+m}^{(k-1)}]}\}.$$

Hence if we choose the branch of $\sqrt{}$ such that

$$\sqrt{[(z_i^{(k)} - z_{i+m}^{(k)})^2]} = z_i^{(k)} - z_{i+m}^{(k)}$$

we get

$$(3.7) \qquad |\lambda_i^{(k)} - z_i^{(k)}| = 0{\cdot}5|z_i^{(k)} - z_{i+m}^{(k)}| \left| \sqrt{\left[1 + \frac{4F_i^{(k-1)} F_{i+m}^{(k-1)}}{(z_i^{(k)} - z_{i+m}^{(k)})^2}\right]} - 1 \right| \leqq$$

$$\leqq 2 \frac{|F_i^{(k-1)} F_{i+m}^{(k-1)}|}{|z_i^{(k)} - z_{i+m}^{(k)}|} .$$

If we denote

$$(3.8) \qquad b = \frac{1 - (1 + q)^{1/(n-1)}}{1 - 2(1 + q)^{1/(n-1)}}$$

then it is proved in [1] that

$$(3.9) \qquad \begin{aligned} &|z_i^{(1)} - z_i| \leqq b \,.\, q \,.\, d \,, \\ &|F_i^{(0)}| \leqq (1 + q)\, b \,.\, d \,, \\ &|z_i^{(1)} - z_{i+m}^{(1)}| \geqq (1 - 2bq)\, d \,. \end{aligned}$$

According to (3.5) we have $(8.2^{1/(n-1)} - 7)\, q < 1$ and consequently $(1 + q)^{1/(n-1)}\, 8q < < 1 + 7q$.

Hence

$$[(1 + q)^{1/(n-1)} - 1]\, 2(1 + 3q) < (2(1 + q)^{1/(n-1)} - 1)\,(1 - q)$$

gives

$$b < \frac{1 - q}{2(1 + 3q)} .$$

Therefore q_1 exists such that $0 < q < q_1 < 1$ and

$$(3.10) \qquad b < 0{\cdot}5(q_1 - q)/(1 + 2q + qq_1) .$$

We shall perform the estimation by using (3.7) and (3.10):

$$|z_i^{(1)} - \lambda_i^{(1)}| \leqq \frac{2b^2(1 + q)^2}{(1 - 2qb)} d < \frac{(q_1 - q)\, b(1 + q)^2\, d}{(1 + 2q + qq_1)\,(1 - q(q_1 - q)/(1 + 2q + qq_1))} =$$

$$= (q_1 - q)\, b \,.\, d$$

and so

$$|\lambda_i^{(1)} - z_i| \leqq |z_i^{(1)} - \lambda_i^{(1)}| + |z_i^{(1)} - z_i| < q_1 bd \,.$$

Analogously we estimate

$$|\lambda_{i+m}^{(1)} - z_{i+m}| < q_1 bd$$

Analogous considerations can be performed for k if we assume that

$$|\lambda_i^{(k-1)} - z_i| \leqq bdq_1^{2^{k-1}}, \quad i = 1, 2, \ldots, n$$

because (3.9) assumes the following form for k

$$|z_i^{(k)} - z_i| \leqq b \,.\, q_1^{2^{k-1}} d$$
$$|F_i^{(k-1)}| \leqq (1 + q)\, b \,.\, q_1^{2^{k-1}-1} d$$

Further we get

$$|z_i^{(k)} - \lambda_i^{(k)}| \leqq \frac{2b^2(1 + q)^2 q_1^{2^{k-2}}}{(1 - 2b\, qq_1^{2^{k-2}})}\, d$$

and using (3.10)

$$|z_i^{(k)} - \lambda_i^{(k)}| < \frac{(q_1 - q)\, b(1 + q)^2 q_1^{2^{k-2}} d}{(1 + q)^2 + q(1 - q_1^{2^{k-2}})(q_1 - q)}$$

hence

$$|z_i^{(k)} - \lambda_i^{(k)}| < (q_1 - q)\, bq_1^{2^{k-2}} d \,.$$

and

$$|\lambda_i^{(k)} - z_i| < b \,.\, q_1^{2^{k-2}} d \,.$$

The quadratic convergence of the sequences $\{\lambda_i^{(k)}\}_{k=0}^{\infty}$ is proved.

The quadratic convergence of the sequences

$$\{p_i^{(k)}\}_{k=0}^{\infty}, \ \{q_i^{(k)}\}_{k=0}^{\infty} \quad i = 1, 2, \ldots, m$$

is proved as well.

4. PROPERTIES OF THE METHOD

Theorem 4.1. For $p_i^{(k)}$ determined by (1.4) and for $k = 1, 2, \ldots$ it holds

$$p_1^{(k)} + p_2^{(k)} + \ldots + p_m^{(k)} = a_1 a_0^{-1} \tag{4.1}$$

Proof. We rewrite (2.6) in the form

$$\mathbf{A}\, \delta\mathbf{x} = \mathbf{a} - a_0\mathbf{h} \tag{4.2}$$

According to (2.1),

$$h_1 = x_1 + x_2 + \ldots + x_m .$$

Therefore

$$\frac{\partial h_1}{\partial x_k} = 1 , \quad \frac{\partial h_1}{\partial y_k} = 0 , \quad k = 1, 2, \ldots, m . \tag{4.3}$$

From (4.2) and (4.3) we get

$$a_0 \sum_{i=1}^{m} \delta x_i = a_1 - a_0 \sum_{i=1}^{m} x_i .$$

After substituting $x_i = p_i^{(k)}$ we get

$$a_0 \sum_{i=1}^{m} \delta p_i^{(k)} = a_1 - a_0 \sum_{i=1}^{m} p_i^{(k)}$$

for $k = 0, 1, 2, \ldots$ from (4.2). Further

$$a_0 \sum_{i=1}^{m} \delta p_1^{(k)} = a_1 - a_0 \sum_{i=1}^{m} \left(p_i^{(k-1)} + \delta p_i^{(k-1)}\right) =$$

$$= a_1 - a_0 \sum_{i=1}^{m} p_i^{(k-1)} - \left(a_1 - a_0 \sum_{i=1}^{m} p_i^{(k-1)}\right) = 0$$

for $k = 1, 2, \ldots$.

Now we have

$$a_1 - a_0 \sum_{i=1}^{m} p_i^{(k)} = 0$$

for $k = 1, 2, \ldots$. This proves Theorem 4.1.

Remark 4.1. This method can be modified similarly to the case of the modification of Jacobi method into Gauss-Seidl method. Namely, calculating $p_i^{(k+1)}$, $q_i^{(k+1)}$

$$p_j^{(k)}, q_j^{(k)} \quad \text{for} \quad j < i$$

can be replaced in (1.9) by

$$p_j^{(k+1)}, q_j^{(k+1)} .$$

For this modification theorems 3.1 and 3.2 hold.

5. ALGORITHM OF THE METHOD

Procedure **sifac** requires given initial approximations

$$p_i^{(0)}, q_i^{(0)}, \quad i = 1, 2, \ldots, m. \tag{5.1}$$

The results of the procedure **sifac** are approximations

$$p_i^{(k)}, q_i^{(k)}, \quad i = 1, 2, \ldots, m \tag{5.2}$$

for the first k satisfying any criterion (see below) of the stopping of the iterative process.

The initial values for procedure **sifac** can be formed by the procedure **initial** according to the following formulas for $i = 1, 2, \ldots, m$

$$q_i^{(0)} = 1{\cdot}2r^2(1 - 0{\cdot}4/n)^{i-1}, \quad p_i^{(0)} = 2r(1 + 4/(3n + 6) - 4i/(n + 2)),$$

where r is obtained by procedure **radius**, which is a modification of Bernoulli method for finding the maximal modul of the root of a polynomial.

The results of the procedure **sifac** can be used for the reconstruction of the coefficients of the polynomial $f(z)$ by using procedure **coef** and can be used for finding of all roots of the polynomial $f(z)$ by using procedure **roots**.

Procedure **sifac** performs the calculation according to Lemma 2.2. for

$$|q_i^{(k)}| < 1$$

and for

$$|q_i^{(k)}| \geqq 1$$

the following Lemma 5.1 is used

Lemma 5.1. *If* $y_i \neq 0$ *then we get for the solution of the system* (2.14) (2.15)

$$u_i^{(j)} = -\frac{x_i}{y_i} u_i^{(j+1)} - \frac{1}{y_i} u_i^{(j+2)},$$

$$v_i^{(j)} = -\frac{x_i}{y_i} v_i^{(j+1)} - \frac{1}{y_i} v_i^{(j+2)}, \quad j = n - 2, n - 3, \ldots, 0$$

respectively,

$$u_i^{(n-1)} = \bar{\Delta}_i^{-1}\left(\bar{s}_i^{(m)} - \frac{x_i}{y_i}\bar{r}_i^{(m)}\right), \quad u_i^{(n)} = -x_i u_i^{(n-1)} - \bar{\Delta}_i^{-1}\bar{r}_i^{(m)}$$

$$v_i^{(n-1)} = -\bar{\Delta}_i^{-1}\bar{r}_i^{(m)}, \quad v_i^{(n)} = -x_i v_i^{(n-1)} - y_i\bar{\Delta}_i^{-1}\bar{s}_i^{(m)},$$

respectively where

$$\bar{\Delta}_i = a_0\left[(\bar{s}_i^{(m)})^2 - \frac{x_i}{y_i}\bar{r}_i^{(m)}\bar{s}_i^{(m)} + \frac{1}{y_i}(\bar{r}_i^{(m)})^2\right]$$

and where $\bar{r}_i^{(m)}$, $\bar{s}_i^{(m)}$ *are obtained by using the following recurrent formula for* $j = 1, 2, \ldots, m$

$$\bar{r}_i^{(0)} = 0\,, \quad \bar{s}_i^{(0)} = 1\,;$$

$$\bar{r}_i^{(j)} = \begin{cases}\left[1 - \frac{y_j}{y_i} - \left(x_j - \frac{y_j x_i}{y_i}\right)\frac{x_i}{y_i}\right]\bar{r}_i^{(j-1)} + \left(x_j - \frac{y_j x_i}{y_i}\right)\bar{s}_i^{(j-1)} & \text{for } j \neq i \\ \bar{r}_i^{(j-1)} & \text{for } j = i\end{cases}$$

$$\bar{s}_i^{(j)} = \begin{cases}-\left(x_j - \frac{y_j x_i}{y_i}\right)\frac{1}{y_i}\bar{r}_i^{(j-1)} + \left(1 - \frac{y_j}{y_i}\right)\bar{s}_i^{(j-1)} & \text{for } j \neq i \\ \bar{s}_i^{(j-1)} & \text{for } j = i\,.\end{cases}$$

Proof is analogous to that of Lemma 2.2. However, the Gauss elimination is performed from right to left. Thus obtained $u_i^{(n-1)}$, $u_i^{(n-2)}$, $v_i^{(n-1)}$, $v_i^{(n-2)}$ are used in Lemma 2.3 and formulas for $u_i^{(n)}$, $v_i^{(n)}$ complete the proof of Lemma 5.1.

Stopping of the iterative process. We define the following sequences

$$\Delta_k = \max_{1 \leq i \leq m}\left[\min\left(|\delta p_i^{(k)}| + |\delta q_i^{(k)}|, \frac{|\delta p_i^{(k)}| + |\delta q_i^{(k)}|}{|p_i^{(k)}| + |q_i^{(k)}|}\right)\right] \tag{5.4}$$

$$\varphi_k = \begin{cases}\varphi_{k-1} & \text{for } \Delta_k < \Delta_{k-1} \\ \varphi_{k-1} + 1 & \text{for } \Delta_k \geq \Delta_{k-1}\end{cases}, \tag{5.5}$$

where

$$\varphi_0 = 0\,.$$

We consider two criteria of the stopping of the iterative process:

Criterion A

$$\Delta_k < eps \tag{5.6}$$

for the given eps with the following results:

$$\begin{aligned} iter &= k \\ c &= \textbf{true} \\ rel &= \Delta_k \end{aligned} \tag{5.7}$$

Criterion B.

(5.8) $$\varphi_k > max$$

for the given max with the following results

(5.9)
$$iter = k$$
$$c = \textbf{false}$$
$$rel = \Delta_k$$

Formal parameters

a) **quantities to be given**

- n – the order of the polynomial –
  – it must be even
- b – array $b[0 : n]$ containing coefficients a_i beginning with a_0 (see (1.1))
- eps – real value for (5.6)
- max – integer value for (5.8)
- $b1$ – array $b1[1 : n]$ containing initial values (see (5.1)) for the procedure **sifac** ordered in the following way
  $p_1^{(0)}, q_1^{(0)}, p_2^{(0)}, \ldots, q_m^{(0)}$
  and after ending of the iterative process containing the results
  $p_1^{(k)}, q_1^{(k)}, p_2^{(k)}, \ldots, q_m^{(k)}$

b) **results produced by the procedures**

- c – Boolean variable containing **true** if the process was ended by the criterion A (see (5.6) and (5.7)) or **false** if the process was ended by the criterion B (see (5.8) and (5.9))
- $iter$ – number of performed iterations (see (5.7) or (5.9))
- rel – values of Δ_k (see (5.4), (5.7) and (5.9))
- r – value for (5.3)
- $b2$ – array $b2\,[1 : n]$ containing results of the procedure initial
- $b3$ – array $b3\,[1 : n]$ containing intermediate results of the procedure radius
- $b4$ – array $b4\,[1 : n]$ containing the next values of the approximations in procedure sifac
- $b5$ – array $b5\,[1 : n]$ containing real parts of the roots after calling of the procedure roots
- $b6$ – array $b6\,[1 : n]$ containing imaginary parts of the roots after calling of the procedure roots

b7 – array *b7* [0 : *n* + 2] containing the coefficients of the reconstructed polynomial after calling of the procedure coef i.e. *b7* [*i*] approximates *b*[*i*] for *i* = 0, 1, ..., *n*.

err- estimation

$$err = \max_{1 \leqq i \leqq n} \frac{b7[i] - b[i]}{b[i]}$$

if *b*[*i*] = 0 then the denominator equals 1.

c) **an example of application**

```
begin integer n, n1, max; real eps;
  inreal (1, eps);
  ininteger (1, n1);
  ininteger (1, max);
  n := 2 × entier ((n1 + 1)/2);
begin real r, rel, err; integer i1, iter;
  Boolean c; real array a[0 : n], a1,
  a2[1 : n], a3[0 : n + 2];
  comment the declaration of the procedures must be placed here;
  for i1 := 0 step 1 until n1 do inreal (1, a[i1]);
  if n ≠ n1 then a[n] := 0;
  radius (n, r, a, a1);
  initial (n, r, a, a1);
  sifac (n, eps, max, a, a1, a2, iter, c, rel);
  coef (n, a, a1, a3, err);
  roots (n, a1, a1, a2);
  outresults (n, a, a1, a2, a3, iter, rel, err)
end body;
end application;
```

The statement

```
sifac (n, eps, max, a, a1, a2, iter, c, rel);
```

can be replaced by

```
sifac (n, eps, max, a, a1, a1, iter, c, rel);
```

for the modification of the method analogous to Gauss-Seidl variation of the Jacobi method (see remark 4.1).

If initial values (5.1) are given then the statements

radius (*n*, *r*, *a*, *a*1);
initial (*n*, *r*, *a*, *a*1);

will be replaced by

for *i*1 := 1 **step** 1 **until** *n* **do** *inreal* (1, *a*1 [*i*1]);

Declaration of the procedures

```
procedure sifac (n, eps, max, b, b1, b4, iter, c, rel);
  value n, eps, max; integer n, max, iter;
  real eps, rel; Boolean c; real array b, b1, b4:
  begin integer i, j, k, fi; real p, p1, p2, rep, r1, s1, v, x1, x2;
    iter := fi := 0;
      rep := 1;
Ls1 : iter := iter + 1;
    rel := 0;
    for i := 2 step 2 until n do
      begin real procedure delta (u, v); value u, v; real u, v;
        begin real p, q;
          p := b[if c then 0 else n] × v;
          for k := 1 step 1 until n - 1 do
            begin
            p := b[if c then k else n - k] × u + p;
            q := - p1 × u - p2 × v;
            v := u ;
            u := q
            end k;
          delta := b[if c then n else 0] × u + p
        end delta;
    Ls2 : r1 := 0;
        s1 := 1;
        c := abs (b1[i]) ≧ 1;
        p2 := if c then 1/b1[i] else b1[i];
        p1 := if c then b1[i - 1] × p2 else b1[i - 1] ;
        p := abs (b1[i]) + ab(b1[i - 1]);
```

```
    for j := 2 step 2 until n do if i ≠ j then
        begin
        x2 := if c then b1[j] × p1 − b1[j − 1] else p1 − b1]j − 1];
        x1 := if c then 1 − b1[j] × p2 else b1[j] − p2;
        v := (x2 × p1 + x1) × r1 − x2 × s1;
        s1 := x2 × p2 × r1 + x1 × s1;
        r1 := v
        end j;
    v := (r1 × p2 − s1 × p1) × r1 + s1 × s1;
    if abs (v) < p ×10 − 10 then
        begin
        b1[i] := if b1[i] = 0 then 10− 8 else b1[i] × 1.1;
        b1[i − 1] := if b1[i − 1] = 0 then 10− 8 else b1[i − 1] × 1.1;
        go to Ls2
        end;
    v := 1/(v × b[0]);
    if c then begin
        x1 := delta((s1 − r1 × p1) × v, (b1[i − 1] × (r1 × p1 − s1) − r1) × v);
        x2 := delta(−r1 × v, (b1[i − 1] × r1 − b1[i] × s1) × v)
        end
          else begin
        x1 := delta (s1 × v, −r1 × v);
        x2 := delta (r1 × p2 × v, (s1 − r1 × p1) × v)
        end;
    v := if p > 1 then (abs (x1) + abs (x2))/p else abs (x1) + abs (x2);
    if v > rel then rel := v;
        b4[i] := b1[i] + x2;
        b4[i − 1] := b1[i − 1] + x1;
        comment for output p_i^(k), q_i^(k) we can perform output of i, b1[i − 1],
        b1[i] and v;
    end i;
    for k := 1 step 1 until n do b1[k] := b4[k];
    if abs (rel/rep) ≧ 1 then fi := fi + 1;
    c := rel < eps;
    if ¬ (c ∨ fi ≧ max) then go to Ls1
end sifac;
```

```
procedure initial (n, r, b, b2); integer n; real r; real array b, b2;
  begin real p, q, s; integer j;
    p := -8 × r/(n + 2);
    q := r + r - p/3;
    s := r × r × 1.2;
    for j := 2 step 2 until n do begin
      s := b2[j] := s × (1 - 0.4/n];
      b2[j - 1] := q := p + q
      end j;
  end initial;

procedure radius (n, r, b, b3); integer n; real r; real array b, b3;
    begin real r1, s, v, x1, x2, y1, y2; integer i, j, k, p; Boolean c;
    v := -1/b[0];
    j := k := p := 1;
    r := r1 := y2 := b3[1] := 1;
    c := false;

Lr1: x2 := 0;
    for i := 1 step 1 until j do x2 := b[i] × b3[j - i + 1] + x2;
    s  := x2 × v;
    x2 := abs (s);
    x1 := x2 ↑ (1/p);
    go to if abs (x2) > 1 then Lr8
      else if c then Lr3
        else if abs (y2) < abs (x2) then Lr2
          else if p < n then Lr4 else Lr6;

Lr2: c := true;
Lr3: if abs(x2) < abs(y2) then go to Lr6;
Lr4: y2 := x2;
    y1 := x1;
Lr5: p  := p + 1;
    x2 := r;
    go to Lr9;
Lr6: go to if abs (y2) < 0·95 then Lr7
    else if k > 16 then Lr 10 else Lr5;
```

```
Lr7: x1 := y1;
     x2 := y2 × y1;
Lr8: c := false;
     y2 := 1; p := 1;
      r := x1 × r;
     x2 := r × x2;
Lr9: k := k + 1;
     x2 := 1/x2;
     if j < n then
                    begin
                    for i := 1 step 1 until j do
                      b3[i] := b3[i] × x2;
                    r1 := r1 × x2;
                    b3[1] := b3[1] + r1;
                    j := j + 1
                    end
               else for i := 2 step l until n do
                        b3[i − 1] := b3[i] × x2;
     b3[j] := s × x2;
     if k < 5 × n then go to Lr1;
Lr10: end radius;

procedure coef (n, b, b1, b7 err); integer n; real err; real array b, b1, b7;

  begin integer j, k; real p;
    for j := 0 step 1 until n + 2 do b7 [j] := 0;
    b7[n] := b[0];
    for k := 2 step 2 until n do
        for j := n − k step 1 until n do
        b7[j] := b7[j] × b1[k] + b7[j + 1] × b1[k − 1] + b7[j + 2];
    err := 0;
    for j := 1 step 1 until n do
        begin
        p := b7[j] − b[j];
        if b[j] ≠ 0 then p := p/b[j];
        if abs [p] > abs (err) then err := p
        end;
  end coef;
```

```
procedure roots (n, b1, b5, b6); integer n; real array b1, b5, b6;
  begin integer k; real p, q;
    for k := 2 step 2 until n do
      begin
      p := -0.5 × b1[k - 1];
      q := p × p - b1[k];
      if q < 0 then begin
                      b5[k - 1] := b5[k] := p;
                      b6[k] := sqrt (-q);
                      b6[k - 1] := -b6[k]
                      end
               else begin
                      b5[k] := p + sqrt (q);
                      b5[k - 1] := -b1[k - 1] - b5[k];
                      b6[k - 1] := b6[k] := 0
                      end;
  end k; .
end roots;
```

Numerical examples. The method is illustrated by two examples:

$$f_1(z) = z^{20} - 1\,, \quad f_2(z) = (z-1)^4 \,.\, (z-2)^3 \,.\, (z-3)^2 \,.\, (z-4)$$

Table 1

Initial values and results for $z^{20} - 1$

j	initial values		results	
	$p_j^{(0)}$	$q_j^{(0)}$	$p_j^{(30)}$	$q_j^{(30)}$
1	1·75757575756	1·176	1·17557050458	1
2	1·39393939392	1·15248	0·61803398875	1
3	1·03030303029	1·1294304	1·61803398874	1
4	0·66666666666	1·106841792	−1·90211303259	1
5	0·30303030302	1·08470495614	1·90211303259	1
6	−0·06060606061	1·06301085702		−1
7	−0·42424242425	1·04175063988		1
8	−0·78787878789	1·02091562709	−0·61803398875	1
9	−1·15151515152	1·00049731454	−1·17557050458	1
10	−1·51515151515	0·98048736825	−1·61803398875	1

Table 2

Initial values and results for $f_2(z)$

j	Initial values		results	
	$p_j^{(0)}$	$q_j^{(0)}$	$p_j^{(45)}$	$q_j^{(45)}$
1	6·24903434608	18·5911925932	−7·00127215639	12·0053638741
2	2·57087676919	17·9475448895	−2.92672894585	1·8112962497
3	0·89271919231	17·1336430938	−2·08416132007	1·0856580055
4	−1·78543838456	16·4482973701	−3·98890729233	2·9677317228
5	−4·46359596145	15·7903654752	−3·99999551239	3·9998973799

Table 3

Obtained roots

j	roots of $z^{20} - 1$	roots of $f_2(z)$
1	$-0{\cdot}587785252293 \pm 0{\cdot}809016994378i$	3·00154783161
2	$-0{\cdot}309016994377 \pm 0{\cdot}951056516296i$	3·99972432477
3	$-0{\cdot}809016994371 \pm 0{\cdot}587785252293i$	0·88878694904
4	$0{\cdot}951056516296 \pm 0{\cdot}309016994380i$	2·03794199680
5	$-0{\cdot}951056516296 \pm 0{\cdot}309016994362i$	1·02552480006
6	$\pm 0{\cdot}999999999996$	1·05863652000
7	$\pm 0{\cdot}999999999996i$	0·98940955560
8	$0{\cdot}309016994373 \pm 0{\cdot}951056516296i$	2·99949773673
9	$0{\cdot}587787252293 \pm 0{\cdot}809016994371i$	1·39032072785
10	$0{\cdot}809016994378 \pm 0{\cdot}587785252293i$	2·00967478453

For $f_1(z)$ it was substituted

$$eps = 10^{-9}, \quad max = 50$$

and for $f_2(z)$

$$eps = 10^{-6}, \quad max = 20$$

Results are (see (5.7) and (5.9)).

$$iter = 20, \; c = \textbf{true}, \; rel = 9.54_{10} - 12 \text{ for } f_1(z)$$

$$iter = 45, \; c = \textbf{false}, \; rel = 1.6_{10} - 1 \text{ for } f_2(z)$$

Initial values and results are placed in the tables 1, 2, 3.

These examples illustrate the algorithm of the method. More detailed numerical explanation and the comparison of this method and of other modifications of the Newton method used for simultaneous finding of all the roots of the polynomial will be published in the next paper.

References

[1] *Дочев Кирил*: Видоизменен метод на Нютон за единовременно приблизително пресмятане на всички корени на дадено алгебраично уравнение.
Физико математическо списаеи Болг. акад. на науките, том 5 (*38*), 1962, pp. 136—139.

Souhrn

ROZKLAD POLYNOMU NA KVADRATICKE SOUČINITELE NEWTONOVOU METODOU

JOSEF DVORČUK

V článku je odvozena metoda rozkladu polynomu sudého stupně na kvadratické součinitele tj.

$$f(z) = \sum_{i=0}^{n} a_i z^{n-i} = a_0 \prod_{j=1}^{0,5n} (z^2 + p_j z + q_j)$$

konstruováním posloupností

$$\{p_i^{(k)}\}_{k=0}^{\infty}, \{q_i^{(k)}\}_{k=0}^{\infty} \quad i = 1, 2, \ldots, 0{\cdot}5n$$

podle vzorců (1.4), (1.6), (1.7) a (1.9).

Metoda je určena pro vyhledání současně všech kořenů polynomu.

Pro polynomy s reálnými koeficienty metoda vyžaduje pouze reálnou aritmetiku.

Jsou-li zadána dobrá počáteční přiblížení $p_i^{(0)}$, $q_i^{(0)}$ pro polynom s různými kořeny, pak metoda konverguje kvadraticky (věta 3.1). V dalších větách jsou formulovány ještě některé vlastnosti metody.

V algoritmu je navržen způsob volby počátečních přiblížení.

Algoritmus metody je popsán v Algole 60.

Numerické výsledky ilustrují algoritmus. Podrobnější rozbor numerických výsledků a porovnání odvozené metody s dalšími variantami Newtonovy metody používané pro tento účel bude publikován v dalším článku.

Author's address: Josef Dvorčuk, Ústav výpočtové techniky ČSAV a ČVUT, Horská 3, Praha 2.