LINEAR INTERPOLATION AND PARALLEL ITERATION FOR SPLITTING FACTORS OF POLYNOMIALS*1)

ZHENG SHI-MING (郑士明)

(Hangzhou University, Hangzhou, China)

§ 1. Introduction

The Bairstow method is a well-known iteration for determining a real quadratic factor of a polynomial with real coefficients

$$F(x)=\sum_{i=0}^{N}a_ix^{N-i}. \tag{1}$$

The method stemmed from applying the Newton method to a system of equations with two variables. Its advantages are that the computational program is simple and that the convergence is quadratic if there are only simple roots or real double roots in the polynomial (see [1]).

In designing filters the computational accuracy is an important problem. In computations we have to find all quadratic factors of the "product polynomial"

$$F(x)=P(x)+K\cdot Q(x), \tag{2}$$

where $P(x)$, $Q(x)$ take the form

$$\prod_{i=1}^{n}(x-r_i) \tag{3}$$

or

$$\prod_{i=1}^{n}(x^2-v_{i1}x-v_{i2}) \tag{4}$$

and K, r_i, v_{i1}, v_{i2} are real numbers (see [2]). In general, the polynomial (2) is transformed into the form (1). When a quadratic factor has been found by the Bairstow method, the polynomial is divided by the factor and the iteration is continued with the quotient polynomial. In this way, all quadratic factors can be found (see [3]). However, in the transformation of (2) into (1) and in the deflation there are accumulations of rounding errors in the coefficients of the polynomial. Wilkinson[4] showed that for the polynomial with clustered roots very small perturbations in coefficients will make comparatively large errors in the roots. The polynomials mentioned in the design of filters are just so. To avoid the accumulation of errors in deflation we can make purification as given in [4], i.e. we can use the factors obtained by deflation as initial approximations for iteration in the original polynomial (1). However, numerical practice shows that there is a danger

* Received February 4, 1985.

1) Projects Supported by the Science Fund of the Chinese Academy of Sciences.

in purification. Although a factor obtained by deflation is closer to some factor of (1), the iteration may converge to another factor. Thus, some of the factors obtained in purification may be repeated or be a new combination of factors and the others disappear. If we use parallel iteration to find all factors of a polynomial simultaneously, then the deflation is not necessary and the danger may be avoided (see examples in Section 5). Moreover, that iteration is suitable for vector computers.

In this paper, from the viewpoint of linear interpolation we give a general rule for constructing a method finding quadratic factors of a polynomial, from which the Bairstow method may be derived and a family of parallel iterations for finding all quadratic factors of a polynomial simultaneously is given too. The latter method is applicable in principle to all polynomials given in any form, provided that their linear interpolation polynomials can be computed in certain ways. In particular, if we apply it directly to (2), the computational program is still simple. We show that its convergence is of order $q+1$ and displays good behavior. For comparison, some simple numerical examples are presented.

§ 2. The Linear Interpolations for Polynomials and Rational Functions

We denote by $\mathbb{R}^n$ the real n-dimensional space. Let $\mathbb{P}$ be a set of polynomials with real coefficients, $\mathbb{P}^n=\{f\in\mathbb{P}\mid$ the degree of f is not greater than $n\}$, $\mathbb{F}=\{f/g\mid f, g\in\mathbb{P}\}$. For $\boldsymbol{u}=(u_1, u_2)^T\in\mathbb{R}^2$, we write

$$Q(\boldsymbol{u})=Q(\boldsymbol{u}, x)=x^2-u_1x-u_2 \tag{5}$$

and denote by

$$L(f)=L(f; \boldsymbol{u}, c; x)=l_1(f; \boldsymbol{u}, c)(x-c)+l_2(f; \boldsymbol{u}, c) \tag{6}$$

the linear interpolation polynomial for $f\in\mathbb{F}$ with nodes α_1, α_2, where α_1, α_2 are the roots of $Q(\boldsymbol{u}, x)$ and $c\in\mathbb{R}^1$ is a number independent of f. $L(f)$ is determined uniquely in spite of c, but a suitable choice of c may reduce the computations (see below). In this section we consider the computation of such linear interpolations for the polynomials given by (1) or (2) and for the rational functions.

It is clear that finding $L(f)$ is equivalent to finding

$$\boldsymbol{l}(f)=\boldsymbol{l}(f; \boldsymbol{u}, c)=(l_1(f; \boldsymbol{u}, c), l_2(f; \boldsymbol{u}, c))^T. \tag{7}$$

Clearly,

$$\begin{cases} L(af+bg)=aL(f)+bL(g), \\ \boldsymbol{l}(af+bg)=a\boldsymbol{l}(f)+b\boldsymbol{l}(g), \quad \forall a, b\in\mathbb{R}^1, f, g\in\mathbb{F}, \end{cases} \tag{8}$$

$$\begin{cases} L(Q(\boldsymbol{u}); \boldsymbol{u}, c; x)\equiv 0, \\ \boldsymbol{l}(Q(\boldsymbol{u}); \boldsymbol{u}, c)=(0, 0)^T. \end{cases} \tag{9}$$

For any $\boldsymbol{v}=(v_1, v_2)^T\in\mathbb{R}^2$,

$$Q(\boldsymbol{v}, x)=x^2-v_1x-v_2=Q(\boldsymbol{u}, x)+(u_1-v_1)x+u_2-v_2.$$

Therefore,

$$\begin{cases} L(Q(\boldsymbol{v}); \boldsymbol{u}, c; x) = (u_1 - v_1)(x-c) + u_2 - v_2 + c(u_1 - v_1), \\ \boldsymbol{l}(Q(\boldsymbol{v}); \boldsymbol{u}, c) = (u_1 - v_1, u_2 - v_2 + c(u_1 - v_1))^T. \end{cases} \tag{10}$$

Thus, from

$$\begin{aligned} L(fg; \boldsymbol{u}, c; x) &= L(L(f) \cdot L(g)) \\ &= L([l_1(f)(x-c) + l_2(f)][l_1(g)(x-c) + l_2(g)]) \\ &= l_1(f)l_1(g)L(x^2 - 2cx + c^2) + [l_1(f)l_2(g) + l_2(f)l_1(g)](x-c) + l_2(f)l_2(g) \\ &= \{[(u_1 - 2c)l_1(f) + l_2(f)]l_1(g) + l_1(f)l_2(g)\}(x-c) \\ &\quad + (u_2 + u_1 c - c^2)l_1(f)l_1(g) + l_2(f)l_2(g), \end{aligned}$$

we see that

$$\boldsymbol{l}(fg; \boldsymbol{u}, c) = A(f; \boldsymbol{u}, c)\boldsymbol{l}(f; \boldsymbol{u}, c), \tag{11}$$

where

$$A(f; \boldsymbol{u}, c) = \begin{pmatrix} (u_1 - 2c)l_1(f; \boldsymbol{u}, c) + l_2(f; \boldsymbol{u}, c) & l_1(f; \boldsymbol{u}, c) \\ (u_2 + u_1 c - c^2)l_1(f; \boldsymbol{u}, c) & l_2(f; \boldsymbol{u}, c) \end{pmatrix}. \tag{12}$$

Using

$$f(\alpha_i) = l_1(f)(\alpha_i - c) + l_2(f), \quad i = 1, 2, \ u_1 = \alpha_1 + \alpha_2, \ u_2 = -\alpha_1\alpha_2,$$

we can easily verify that

$$\det A(f; \boldsymbol{u}, c) = f(\alpha_1)f(\alpha_2). \tag{13}$$

Hence, if $g(\alpha_1)g(\alpha_2) \neq 0$, applying (11) to $f = g(f/g)$, we have

$$\boldsymbol{l}(f/g; \boldsymbol{u}, c) = A(g; \boldsymbol{u}, c)^{-1}\boldsymbol{l}(f; \boldsymbol{u}, c). \tag{14}$$

Thus, it is not difficult to compute the linear interpolations for the rational functions by using that for the polynomials. Using (8) and (11) we can conveniently compute the linear interpolations for the polynomials with form (1) or (2).

If $F \in \mathbb{P}^N$ is given by (1), writing $F_0(x) = a_0$, $F_i(x) = xF_{i-1}(x) + a_i$, $i = 1, \cdots, N$ and choosing $c = u_1$, we have $\boldsymbol{l}(x; \boldsymbol{u}, u_1) = (1, u_1)^T$,

$$A(x; \boldsymbol{u}, u_1) = \begin{pmatrix} 0 & 1 \\ u_2 & u_1 \end{pmatrix}.$$

From (8) and (11), we obtain

$$\begin{aligned} \boldsymbol{l}(F_0) &= (0, a_0)^T, \\ \boldsymbol{l}(F_i) &= A(x; \boldsymbol{u}, u_1)\boldsymbol{l}(F_{i-1}) + (0, a_i)^T \\ &= (l_2(F_{i-1}), a_i + u_1 l_2(F_{i-1}) + u_2 l_2(F_{i-2}))^T, \quad i = 1, \cdots, N. \end{aligned}$$

Let $b_i = l_2(F_i)$. We have

Algorithm A. $\boldsymbol{l}\left(\sum_{i=0}^{N} a_i x^{N-i}; \boldsymbol{u}, u_1\right) = (b_{N-1}, b_N)^T$, where b_j are given by recursion

$$\begin{cases} b_{-1} = 0, \ b_0 = a_0, \\ b_j = a_j + u_1 b_{j-1} + u_2 b_{j-2}, \quad j = 1, \cdots, N. \end{cases}$$

If $G \in \mathbb{P}$ is given by

$$G(x) = a_0 \prod_{i=1}^{n} g_i(x), \quad g_i \in \mathbb{P},$$

writing $G_0(x) = a_0$, $G_i(x) = g_i(x)G_{i-1}(x)$, $i = 1, \cdots, n$, then from (11) we have

Algorithm B. $l\left(a_0\prod_{i=1}^{n}g_i(x);\ u,\ c\right)=l_n$ is given by recursion

$$\begin{cases} l_0=(0,\ a_0)^T,\\ l_i=A(g_i;\ u,\ c)l_{i-1}, \quad i=1,\ \cdots,\ n. \end{cases}$$

In particular, if $G(x)$ is a polynomial of form (3) or (4), then the corresponding $g_i(x)$ are $x-r_i$ or $x^2-v_{i1}x-v_{i2}$, respectively. From (8) and (10), the corresponding $l(g_i;\ u,\ c)$ are $(1,\ c-r_i)^T$ or $(u_1-v_{i1},\ u_2-v_{i2}+c(u_1-v_{i1}))^T$, respectively. Then we place them into (12) to obtain $A(g_i;\ u,\ c)$. From (12) we see that we may choose $c=0$, $u_1/2$ or u_1 in order to reduce the computations in the recursion. If the roots α_1, α_2 of $x^2-u_1x-u_2$ are real numbers, then we may also choose $c=\alpha_1$ or $c=\alpha_2$. Some $A(g_i;\ u,\ c)$, corresponding to $c=0$, or $c=\alpha_1$, are listed in Table 1.

Table 1 The forms of partial $A(g_i;\ u,\ c)$

c \ g_i	$x-r_i$	$Q(v_i,\ x)$
0	$\begin{pmatrix} u_1-r_i & 1 \\ u_2 & -r_i \end{pmatrix}$	$\begin{pmatrix} u_1(u_1-v_{i1})+u_2-v_{i2} & u_1-v_{i1} \\ u_2(u_1-v_{i1}) & u_2-v_{i2} \end{pmatrix}$
α_1	$\begin{pmatrix} \alpha_2-r_i & 1 \\ 0 & \alpha_1-r_i \end{pmatrix}$	$\begin{pmatrix} \alpha_2(u_1-v_{i1})+u_2-v_{i2} & u_1-v_{i1} \\ 0 & \alpha_1(u_1-v_{i1})+u_2-u_{i2} \end{pmatrix}$

§ 3. Linear Interpolation and Splitting Factors of Polynomials

It is clear that $l(f;\ u,\ 0)=l(f;\ u,\ c)-c(0,\ l_1(f;\ u,\ c))^T$. Therefore, we always suppose $c=0$ in the following and write

$$L(f;\ u;\ x)=L(f;\ u,\ c;\ x),\quad l(f;\ u)=l(f;\ u,\ 0),\quad A(f;\ u)=A(f;\ u,\ 0).$$

Suppose $p_i=(p_{i1},\ p_{i2})^T\in\mathbb{R}^2$, $Q(p_i)=Q(p_i,\ x)=x^2-p_{i1}x-p_{i2}$ is the i-th factor of $F(x)$:

$$F(x)=Q(p_i,\ x)F_i(x). \tag{15}$$

If $u_i=(u_{i1},\ u_{i2})^T\in\mathbb{R}^2$ is an approximation of p_i and α_{i1}, α_{i2} are the roots of $Q(u_i,\ x)=x^2-u_{i1}x-u_{i2}$, $F_i(\alpha_{i1})F_i(\alpha_{i2})\neq0$, then we obtain from (10) $l(Q(p_i);\ u_i)=u_i-p_i$. But $Q(p_i)=F(x)/F_i(x)$, so

$$p_i=u_i-l(F/F_i;\ u_i). \tag{16}$$

If an approximate polynomial $G(x)$ of $F_i(x)$ can be given in one way or another, $G(\alpha_{i1})G(\alpha_{i2})\neq0$, and if $l(F;\ u_i)$ and $l(G;\ u_i)$ can be computed by some schemes, then we obtain from (16) and (14) a new approximation of p_i;

$$u_i'=u_i-l(F/G;\ u_i)=u_i-A(G;\ u_i)^{-1}l(F;\ u_i), \tag{17}$$

and an iteration is constructed.

For example, if $F(x)$ is given by (1), comparing the coefficients, we see that for any $u=(u_1,\ u_2)^T\in\mathbb{R}^2$,

$$F(x)=(x^2-u_1x-u_2)G(x)+b_{N-1}(x-u_1)+b_N,$$

where $G(x)=\sum_{j=0}^{N-2} b_j x^{N-2-j}$, b_j $(j=0, 1, \cdots, N)$ are determined by Algorithm A. If $\boldsymbol{u}$ is a good approximation of $\boldsymbol{p}_i$, then b_{N-1}, b_N are close to 0, and we may approximate $F_i(x)$ by $G(x)$ in (15) and $\boldsymbol{l}(G; \boldsymbol{u})$ can be again obtained by Algorithm A. This is just the Bairstow method.

In the following we suppose that $F\in \mathbb{R}^{2n}$ (otherwise, we consider $xF(x)$ instead of $F(x)$). Then there are $\boldsymbol{p}_i=(p_{i1}, p_{i2})^T\in\mathbb{R}^2$, $i=1, \cdots, n$, such that

$$F(x)=a_0\prod_{j=1}^{n} Q(\boldsymbol{p}_j, x)=a_0\prod_{j=1}^{n}(x^2-p_{j1}x-p_{j2})=Q(\boldsymbol{p}_i, x)F_i(x)$$
$$=Q(\boldsymbol{p}_i, x)\cdot a_0\prod_{\substack{j=1\\ j\neq i}}^{n} Q(\boldsymbol{p}_j, x). \tag{18}$$

Except when otherwise stated we always suppose that parameter q is a natural number, and that subscripts i, j, k are evaluated $1, 2, \cdots, n$ in order; and we denote by $m=0, 1, \cdots$ the numbers of iteration steps and by $\mu=1, \cdots, q$ the numbers of substeps from m-th to $(m+1)$-th step. Let $\boldsymbol{u}_j^{\left(m+\frac{\mu-1}{q}\right)}$ be the $\left(m+\frac{\mu-1}{q}\right)$-th approximation of $\boldsymbol{p}_j$. To obtain the $\left(m+\frac{\mu}{q}\right)$-th of $\boldsymbol{p}_j$, we approximate $F_i(x)$ in (15) by

$$G_i^{\left(m+\frac{\mu-1}{q}\right)}(x)=a_0\prod_{j\neq i} Q\left(\boldsymbol{u}_j^{\left(m+\frac{\mu-1}{q}\right)}, x\right). \tag{19}$$

Then $\boldsymbol{l}\left(G_i^{\left(m+\frac{\mu-1}{q}\right)}; \boldsymbol{u}_i\right)$ and $A\left(G_i^{\left(m+\frac{\mu-1}{q}\right)}; \boldsymbol{u}_i\right)$ can be easily computed by Algorithm B. Therefore we have

Parallel Iteration $P(q)$

$$\boldsymbol{u}_i^{\left(m+\frac{\mu}{q}\right)}=\boldsymbol{u}_i^{(m)}-\boldsymbol{l}\left(F/G_i^{\left(m+\frac{\mu-1}{q}\right)}; \boldsymbol{u}_i^{(m)}\right)$$
$$=\boldsymbol{u}_i^{(m)}-A\left(G_i^{\left(m+\frac{\mu-1}{q}\right)}; \boldsymbol{u}_i^{(m)}\right)^{-1}\boldsymbol{l}(F; \boldsymbol{u}_i^{(m)}),\quad m=0, 1, \cdots, \mu=1, \cdots, q. \tag{20}$$

§ 4. The Convergence of Parallel Iteration $P(q)$

For $\boldsymbol{u}=(u_1, u_2)^T\in\mathbb{R}^2$ we write

$$\|\boldsymbol{u}\|=\max_{i=1,2}|u_i|.$$

First we prove two lemmas.

Lemma 1. *Suppose that* $\boldsymbol{u}$, $\boldsymbol{u}_i$, $\boldsymbol{v}_i\in\mathbb{R}^2$, $\|\boldsymbol{u}_i-\boldsymbol{v}_i\|<\delta$, $i=1, \cdots, n$,

$$U_n(x)=\prod_{j=1}^{n} Q(\boldsymbol{u}_j, x);\quad V_n(x)=\prod_{j=1}^{n} Q(\boldsymbol{v}_j, x).$$

Then

$$\|\boldsymbol{l}(U_n-V_n; \boldsymbol{u})\|=O(\delta)\quad as\ \delta\to 0. \tag{21}$$

Proof. By (8), (10), $\boldsymbol{l}(Q(\boldsymbol{u}_i)-Q(\boldsymbol{v}_i); \boldsymbol{u})=\boldsymbol{v}_i-\boldsymbol{u}_i$, (21) holds clearly when $n=1$. If (21) holds for $n-1$, from (8), (11) and

$$U_n-V_n=(Q(\boldsymbol{u}_n)-Q(\boldsymbol{v}_n))U_{n-1}+Q(\boldsymbol{v}_n)(U_{n-1}-V_{n-1})$$

we see that (21) holds also. Lemma 1 is proved.

Lemma 2. *If* $F\in\mathbb{P}^{2n}$, $\boldsymbol{u}_i\in\mathbb{R}^2$, $i=1, \cdots, n$, *then*

$$F(x)=\sum_{j=1}^{n}L(F/G_j;\ u_j;\ x)G_j(x)+a_0\prod_{j=1}^{n}Q(u_j,\ x),\tag{22}$$

where a_0 is the coefficient of x^{2n}, $G_j(x)=a_0\prod_{k\neq j}Q(u_k,\ x)$.

Proof. We denote by $L_{2n-1}(x)$ the first sum in (22). It is clear that $L_{2n-1}\in \mathbf{P}^{2n-1}$. From (9) we see that $L(G_j;\ u_i;\ x)\equiv 0,\ j\neq i$. Therefore we obtain from (8)

$$L(L_{2n-1};\ u_i;\ x)=L(L(F/G_i)\cdot L(G_i)).$$

Since $L(F;\ u_i;\ x)=L(L(F/G_i)\cdot G_i)=L(L(F/G_i)\cdot L(G_i))$, so

$$L_{2n-1}(\alpha_{ij})=L(L_{2n-1};\ u_i;\ \alpha_{ij})=L(F;\ u_i;\ \alpha_{ij})=F(\alpha_{ij}),\quad j=1,\ 2;\ i=1,\ \cdots,\ n,$$

where α_{i1}, α_{i2} denote again the roots of $Q(u_i,\ x)$. Using the uniqueness of the interpolation polynomial if $2n$ points α_{ij} are different from each other, otherwise using also the continuity of this polynomial with respect to the nodes, we see that $L_{2n-1}(x)$ is the interpolation polynomial of degree $2n-1$ for $F(x)$ with nodes α_{ij}. Then (22) is obtained by the Lagrange interpolation formula. Lemma 2 is proved.

Theorem 1. *If the roots of $Q(p_j;\ x)$ are different from those of $Q(p_i;\ x)$, $j\neq i$, then starting from suitable approximations $u_i^{(0)}$ of p_i, the parallel iteration $P(q)$ is convergent with order $q+1$.*

Proof. Let $\delta^{\left(m+\frac{\mu-1}{q}\right)}=\max\limits_{1\le i\le n}\|u_i^{\left(m+\frac{\mu-1}{q}\right)}-p_i\|$ and let $u_i=u_i^{(m)}$ in (16). Subtracting (16) from (20) and writing $G_i=G_i^{\left(m+\frac{\mu-1}{q}\right)}$, we obtain from (8), (11), (14)

$$\begin{aligned}u_i^{\left(m+\frac{\mu}{q}\right)}-p_i&=l(F/F_i-F/G_i;\ u_i^{(m)})\\&=l\left(\frac{(F/F_i)(G_i-F_i)}{G_i};\ u_i^{(m)}\right)=A(G_i;\ u_i^{(m)})^{-1}A(F/F_i;\ u_i^{(m)})l(G_i-F_i;\ u_i^{(m)}).\end{aligned}\tag{23}$$

Using (16), (12) and Lemma 1, we have

$$l(F/F_i;\ u_i^{(m)})=u_i^{(m)}-p_i,$$

$$\|A(F/F_i;\ u_i^{(m)})\|=O(\delta^{(m)}),\tag{24}$$

$$\|l(G_i-F_i;\ u_i^{(m)})\|=O\left(\delta^{\left(m+\frac{\mu-1}{q}\right)}\right).\tag{25}$$

According to the condition of the theorem and (13), $A(F_i;\ p_i)$ is invertible and there is a number $\beta\in\mathbf{R}^1$ such that $\|A(F_i;\ p_i)^{-1}\|\le\beta$. Clearly, if $\delta^{(m)}\to 0$, $\delta^{\left(m+\frac{\mu-1}{q}\right)}\to 0$, then $A(G_i;\ u_i^{(m)})\to A(F_i;\ p_i)$. Therefore, by the perturbation lemma (see [5]), $A(G_i;\ u_i^{(m)})$ is invertible too and there is a number $\beta_1\in\mathbf{R}^1$ such that $\|A(G_i;\ u_i^{(m)})^{-1}\|\le\beta_1$. Then $\|u_i^{\left(m+\frac{\mu}{q}\right)}-p_i\|=O\left(\delta^{\left(m+\frac{\mu-1}{q}\right)}\delta^{(m)}\right)$ from (23)—(25). Thus, by using mathematical induction for μ we have

$$\delta^{\left(m+\frac{\mu}{q}\right)}=O((\delta^{(m)})^{\mu+1}),\quad \mu=1,\ \cdots,\ q.$$

In particular, $\delta^{(m+1)}=O((\delta^{(m)})^{q+1})$. The theorem is proved.

From (20), (19) and (13) we see that a necessary condition to ensure that the iteration $P(q)$ can proceed is that the initial approximations $u_i^{(0)}$ are chosen such that the roots of $Q(u_j^{(0)},\ x)$ are different from those of $Q(u_i^{(0)},\ x)$, $j\neq i$. Hence, we choose generally

$$\alpha_{ik}^{(0)}=\frac{i}{2}+(-1)^k\sqrt{-1},\quad i=1,\cdots,n,\ k=1,2,$$

i.e.,

$$u_i^{(0)}=\left(i,\ -\left(\frac{i}{2}\right)^2-1\right)^T.$$

It seems that few failures appear in numerical practice for iteration $P(1)$, starting from such initial approximations. The following theorem shows that the arithmatic mean of all $2n$ roots of the approximate quadratic factors obtained after one iteration $P(1)$ is equal to that of exact factors, no matter how we choose $u_i^{(0)}$. This provides an explanation for the above phenomenon in a sense.

Theorem 2. *No matter how we choose the initial approximations $u_i^{(0)}$, the equation*

$$\sum_{i=1}^n u_{i1}^{\left(m+\frac{1}{q}\right)}=\sum_{i=1}^n p_{i1}=-a_1/a_0,\quad m=0,1,\cdots \tag{26}$$

holds for parallel iteration $P(q)$.

Proof. Let $\mu=1$ in (20). We obtain

$$l(F/G_i^{(m)};\ u_i^{(m)})=u_i^{(m)}-u_i^{\left(m+\frac{1}{q}\right)}$$

and

$$L(F/G_i^{(m)};\ u_i^{(m)};\ x)=\left(u_{i1}^{(m)}-u_{i1}^{\left(m+\frac{1}{q}\right)}\right)x+u_{i2}^{(m)}-u_{i2}^{\left(m+\frac{1}{q}\right)}.$$

Substituting $L(F/G_i^{(m)};\ u_i^{(m)};\ x)$ into Lemma 2, we have

$$F(x)=a_0\sum_{j=0}^n\left[\left(u_{j1}^{(m)}-u_{j1}^{\left(m+\frac{1}{q}\right)}\right)x+\left(u_{j2}^{(m)}-u_{j2}^{\left(m+\frac{1}{q}\right)}\right)\right]\prod_{k\neq j}(x^2-u_{k1}^{(m)}x-u_{k2}^{(m)})$$

$$+a_0\prod_{j=1}^n(x^2-u_{j1}^{(m)}x-u_{j2}^{(m)}). \tag{27}$$

Comparing the coefficients of x^{2n-1} in (27) with those in (18) and (1), we obtain (26). The theorem is proved.

§ 5. Numerical Examples

The numerical results in finding in four ways all quadratic factors $Q(p_i;\ x)$ $(i=1,2,3)$ of the polynomial

$$F(x)=\prod_{k=1}^6(x-r_k) \tag{28}$$

are listed in Table 2. The ways (I), (II), (III) start from the same initial approximations $u_i^{(0)}$ and all iterations are stopped when

$$\frac{|u_{ij}^{(m+1)}-u_{ij}^{(m)}|}{|u_{ij}^{(m+1)}|^{\delta_{ij}}}\leqslant 0.00001,\quad i=1,2,3,\ j=1,2,$$

where

$$\delta_{ij}=\begin{cases}0, & |u_{ij}^{(m+1)}|\leqslant 1,\\ 1, & |u_{ij}^{(m+1)}|>1.\end{cases}$$

Table 2 The numerical results in finding all quadratic factors of $\prod_{k=1}^{6}(x-r_k)$ in four ways

j	(I) $P(1)$ is applied directly to (28).			(II) (28) is transformed into (1) and $P(1)$ is applied to (1).			(III) (28) is transformed into (1). Bairstow method is applied to (1) and is continued with deflating polynomial.			(IV) The results of (III) are purified in (1).		
	m	Roots of $Q(u_j^{(m)}, x)$		m	Roots of $Q(u_j^{(m)}, x)$*		m	Roots of $Q(u_j^{(m)}, x)$		m	Roots of $Q(u_j^{(m)}, x)$	
	r_k: 1.0, 2.0, 3.0, 3.0, 2.0, 1.0						$u_{j1}^{(0)}=0.99p_{j1}$, $u_{j2}^{(0)}=1.01p_{j2}$					
1		0.9996548	1.000345		0.9989114	1.001095	3	0.9999931±0.001600937i		1	0.9994534	1.000545
2	5	2.000000	2.000000	200	1.999985±0.0067658244i		3	1.999951±0.007888425i		3	1.999997±0.003521046i	
3		2.998618	3.001380		3.000066±0.0077512254i		1	3.000056±0.008052940i		5	2.995190	3.004758
	r_k: 0.11, 0.12, 0.13, 0.14, 0.15, 0.16						$u_j^{(0)}=(j, -(j/2)^2-1)^T$					
1		0.1399998	0.1499999		0.1453043±0.0020173944i		279	0.1100322	0.1605745	14	0.1097232	0.1610183
2	28	0.1200003	0.1299998	245	0.1196649	0.1303117	28	0.1203899	0.1464660	24	0.1097183	0.1611029
3		0.1100000	0.1599999		0.1102714	0.1598955	1	0.1283793	0.1441581	31	0.1103406	0.1598999

Then we accept $u_i^{(m+1)}$ as p_i. For comparison with the exact roots of $Q(p_i, x)$, the results listed in the table are the roots of $Q(u_i^{(m+1)}, x)$. The computations were completed on a microcomputer CROMEMCO.

References

[1] G. Schmeißer, H. Schirmeier, Praktische Mathematik, Walterde, de Greyter, 1976.

[2] 王斯雷,郑士明,赵彦达,杭州大学学报(自然科学版), 1980, 1, 38—62.

[3] 渡部和,伝送回路网の理論と設計,オーム社, 1968.

[4] J. H. Wilkinson, Rounding Errors in Algebraic Processes, Her Majesty's Stationary Office, London, 1963.

[5] J. M. Ortega, W. C. Reinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, 1970.