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

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

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, $\mathbb{P}^n=\{f\in\mathbb{P}\mid \text{the degree of } f \text{ is not greater than } n\}$, $\mathbb{F}=\{f/g\mid f, g\in\mathbb{P}\}$. 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,