

Divided Differences, Shift Transformations and Larkin's Root Finding Method

By A. Neumaier and A. Schäfer

Abstract. For a one-dimensional complex-valued function f this paper deals with iterative root finding methods using divided differences of f . Assuming that f is given in a Newtonian representation we show how Horner-like transformations ("shift transformations") yield the divided differences needed in each iteration step. In particular, we consider an iteration method given by Larkin [5] and derive an equivalent version of this method fitting into this context. Monotonic convergence to real roots of real polynomials is investigated. Both "shift-" and "nonshift versions" of several root finding methods are tested and compared with respect to their numerical behavior.

1. Introduction. In some region D of the complex plane \mathbb{C} we consider a meromorphic function $f: D \rightarrow \mathbb{C}$ of the variable $x \in D$. Proceeding from an appropriate Taylor expansion of f , one can obtain several well-known root finding methods using a set of derivative values. Examples are Newton's method, Euler's method, etc. Several corresponding methods (for example, the secant rule, Muller's method) which avoid the evaluation of derivatives and use instead divided differences can be derived similarly from a Newtonian expansion of f . It has been emphasized frequently that such methods are in a sense more economical (see, e.g., Traub [6]). Conditions on which a function f can be represented in a unique Newtonian form are specified in Section 2. We also give some special Horner-like algorithms ("shift algorithms") transforming f into suitable forms. An appropriate sequence of these transformations will then provide all the divided differences needed during an iteration process (see Section 3).

In [5] Larkin suggests an iterative root finding method which is derived from approximating f by rational interpolation and can be carried out by constructing divided differences of the function $\bar{f} := 1/f$. Again considering special Newtonian expansions of f and \bar{f} , we obtain an equivalent version of this method which uses divided differences of the original function f . In case f is a real polynomial having only real roots, monotonic convergence to the largest (respectively, smallest) root is proved in Section 4.

In Section 5 we apply shift transformations to determine all the roots of a given polynomial f . Convergence to a certain root automatically yields f in deflated form. In practice it is somewhat dangerous to compute new representations of f iteratively since accumulated rounding errors may cause the roots to be changed gradually.

Received October 25, 1983; revised May 22, 1984.
1980 *Mathematics Subject Classification*. Primary 65H05.

©1985 American Mathematical Society
0025-5718/85 \$1.00 + \$.25 per page

However, an appropriate choice of expansion points will reduce the sensitivity of some roots with respect to perturbations of f ; thus, we can hope that later transformations will not affect the zeros too much. Numerical examples show that for some common initial representations of the given polynomial our approach yields rather satisfactory results.

2. The Newtonian Form. In the sequel let m and j denote nonnegative integers. For any m complex numbers x_k , $k = 1(1)m$, we define

$$\omega(x; x_1, x_2, \dots, x_m) := \prod_{k=1}^m (x - x_k), \quad x \in \mathbb{C}.$$

For a finite sequence of points $z_k \in \mathbb{C}$, $k \in \mathbb{Z}$, we also use the notation

$$\omega_{ij}(x; z) := \begin{cases} \omega(x; z_i, z_{i+1}, \dots, z_j), & \text{if } i \leq j, \\ 1, & \text{if } i = j + 1. \end{cases}$$

The degree of a polynomial p is written as $\deg p$, and the space of polynomials p with $\deg p \leq n$ is denoted by P_n .

First we give a lemma which is fundamental for all following considerations:

LEMMA 1. *If the function $f: D \rightarrow \mathbb{C}$ is analytic at $z_k \in D$, $k = j - m(1)j$, then there exist both a function $r: D \rightarrow \mathbb{C}$ which is analytic at the points z_k and a polynomial*

$$(1) \quad p_m(x) = \sum_{i=j-m}^j f_{ij} \omega_{i+1,j}(x; z) \in P_m$$

such that

$$f(x) = p_m(x) + r(x) \omega_{j-m,j}(x; z).$$

The functions p_m and r are unique.

Proof. With $r_j(x) := f(x)$ we define for $i = j(-1)j - m$ recursively

$$(2) \quad r_{i-1}(x) := \begin{cases} \frac{r_i(x) - r_i(z_i)}{x - z_i}, & \text{if } x \neq z_i, \\ r_i(z_i), & \text{if } x = z_i. \end{cases}$$

With f , all the functions r_i constructed by (2) are analytic at z_k , $k = j - m(1)j$.

Setting

$$(3) \quad \begin{aligned} f_{ij} &:= r_i(z_i), \quad i = j - m(1)j, \\ r(x) &:= r_{j-m-1}(x), \end{aligned}$$

we have $r_i(z_i) = r_i(x) - r_{i-1}(x) \cdot (x - z_i)$ and

$$\sum_{i=j-m}^j r_i(z_i) \cdot \omega_{i+1,j}(x; z) + r_{j-m-1}(x) \cdot \omega_{j-m,j}(x; z) = r_j(x) = f(x).$$

Thus, from definition (3), we obtain f in the required form (1). The uniqueness of p_m and r follows by antithesis.

Let both $f = p_m + r \cdot \omega_{j-m,j}$ and $f = \hat{p}_m + \hat{r} \cdot \omega_{j-m,j}$ be representations of f according to (1). We assume $p_m \neq \hat{p}_m$. Since r and \hat{r} are analytic at z_k , $k = j - m(1)j$,

the same holds for the difference

$$r - \hat{r} = (p_m - \hat{p}_m) / \omega_{j-m, j}.$$

However, we have $0 \neq p_m - \hat{p}_m \in P_m$ and $\deg \omega_{j-m, j} = m + 1$. Therefore the function $r - \hat{r}$ has at least one singularity. This contradiction yields $p_m \equiv \hat{p}_m$ and, hence, $r \equiv \hat{r}$. \square

We say that the function f given by (1) is "represented in Newtonian form with respect to the base points $z_k, k = j - m(1)j$ ".

Remarks. 1. In case the points z_k are pairwise distinct, p_m is the unique Newtonian interpolation polynomial coinciding with f at $z_k, k = j - m(1)j$. Using the well-known notation for divided differences (usually defined according to (2) and (3) (see, e.g., Traub [6])), we have

$$(4) \quad f_{ij} = f[z_i, z_{i+1}, \dots, z_j], \quad r(x) = f[x, z_{j-m}, z_{j-m+1}, \dots, z_j].$$

It is evident that for fixed i and j the value f_{ij} is independent of the size of $m \geq j - i$, so f_{ij} does not change if f is expanded in Newtonian form (1) with respect to additional points $z_{j-m-1}, z_{j-m-2}, \dots$.

2. If $z_k = \hat{z}$ for $k = j - m(1)j$, then (1) corresponds to the Taylor expansion of f at the point \hat{z} . The coefficients then are

$$f_{ij} = \frac{1}{(j-i)!} f^{(j-i)}(\hat{z}).$$

3. For $f \in P_n$ and $m = n$ the remainder function vanishes identically:

$$r(x) \equiv 0.$$

In this case we have $f = p_n$ with p_n given by (1). If, moreover, $z_k = 0$ for $k = j - m(1)j$, then $f = p_n$ has the standard form

$$p_n(x) = \sum_{i=j-m}^j f_{ij} x^{j-i}.$$

LEMMA 2. Let two functions $f, \varphi: D \rightarrow \mathbf{C}$ be given in the form

$$f(x) = \sum_{i=j-m}^j f_{ij} \omega(x; z_{i+1}, z_{i+2}, \dots, z_j) + r(x) \omega(x; z_{j-m}, z_{j-m+1}, \dots, z_j)$$

and

$$\varphi(x) = \sum_{i=j-m+1}^{j+1} \hat{\varphi}_{i, j+1} \omega(x; z_{i+1}, \dots, z_j, \lambda) + \hat{r}(x) \omega(x; z_{j-m+1}, \dots, z_j, \lambda),$$

where $\lambda \in \mathbf{C}$. If r and \hat{r} are analytic at $z_{j-m}, z_{j-m+1}, \dots, z_j$, then f and φ represent the same function if and only if

$$(5) \quad \hat{\varphi}_{i+1, j+1} = f_{ij} + (\lambda - z_i) \hat{\varphi}_{i, j+1}, \quad i = j - m + 1(1)j, \quad \text{and} \\ \hat{\varphi}_{j-m+1, j+1} = f_{j-m, j} + r(x)(x - z_{j-m}) - \hat{r}(x)(x - \lambda).$$

Proof. (a) Let $f \equiv \varphi$. We write

$$f(x) = p_m(x) + r(x)(x - z_{j-m}) \omega_{j-m+1, j}(x; z), \\ \varphi(x) = \pi_m(x) + \hat{r}(x)(x - \lambda) \cdot \omega_{j-m+1, j}(x; z)$$

and set

$$d(x) := r(x)(x - z_{j-m}) - \hat{r}(x)(x - \lambda) = \frac{p_m(x) - \pi_m(x)}{\omega_{j-m+1,j}(x; z)}.$$

Since $d(x)$ is analytic at the roots $z_{j-m+1}, z_{j-m+2}, \dots, z_j$ of $\omega_{j-m+1,j}$, the polynomial $p_m - \pi_m$ must vanish at these points, and $\deg(p_m - \pi_m) \leq \deg \omega_{j-m+1,j}$ implies that $d := d(x)$ is a constant. Thus we may define

$$\hat{f}_{j-m,j} := \hat{\phi}_{j-m+1,j+1} - d \quad \text{and} \quad \hat{f}_{ij} := \hat{\phi}_{i+1,j+1} - (\lambda - z_i) \cdot \hat{\phi}_{i,j+1}$$

for $i = j - m + 1(1)j$, obtaining

$$\begin{aligned} \varphi(x) &= [\hat{f}_{j-m,j} + r(x)(x - z_{j-m}) - \hat{r}(x)(x - \lambda)] \omega_{j-m+1,j}(x; z) \\ &\quad + \sum_{i=j-m+2}^{j+1} \hat{f}_{i-1,j-1} \cdot \omega_{i,j}(x; z) + \hat{r}(x)(x - \lambda) \cdot \omega_{j-m+1,j}(x; z) \\ &= \sum_{i=j-m}^j \hat{f}_{ij} \omega_{i+1,j}(x; z) + r(x)(x - z_{j-m}) \omega_{j-m+1,j}(x; z). \end{aligned}$$

This is a representation of f in Newtonian form (1); Lemma 1 guarantees that the coefficients \hat{f}_{ij} are unique. Hence we have $f_{ij} = \hat{f}_{ij}$ for $i = j - m(1)j$ in accordance with statement (5).

(b) Conversely the assertion $f \equiv \varphi$ follows from (5) by direct calculations analogously to part (a) of this proof. \square

Remarks. 1. The analyticity of r and \hat{r} at $x = \lambda$ is not required.

2. If in addition \hat{r} is analytic at λ , then $f \equiv \varphi$ yields $r(\lambda) = \hat{r}(z_{j-m})$.

The relations (5) can be used to transform the Newtonian representation

$$f(x) = \sum_{i=j-m}^j f_{ij} \omega_{i+1,j}(x; z) + r^{(j)}(x) \omega_{j-m,j}(x; z)$$

into

$$(6) \quad f(x) = \sum_{i=j-m+1}^{j+1} f_{i,j+1} \omega_{i+1,j+1}(x; z) + r^{(j+1)}(x) \omega_{j-m+1,j+1}(x; z),$$

where the base points are "shifted" by deleting z_{j-m} and adding the new point z_{j+1} . We call this the "shift transformation" of f (given by (1)) with shift z_{j+1} .

From (5) we get the following Horner-like scheme for computing the new coefficients of the polynomial part of f .

Shift algorithm:

$$(7_m) \quad \begin{aligned} f_{j-m,j+1} &:= r^{(j)}(z_{j+1}), \\ f_{i+1,j+1} &:= f_{ij} + f_{i,j+1}(z_{j+1} - z_i), \quad i = j - m(1)j. \end{aligned}$$

The special case of $f \in P_n$. Let $f \in P_n$, $n \geq 0$, be completely represented in Newtonian form, i.e.

$$f(x) = \sum_{i=j-m}^j f_{ij} \omega_{i+1,j}(x; z) + r^{(j)}(x) \omega_{j-m,j}(x; z),$$

with the remainder function

$$r^{(j)}(x) = \sum_{i=-n}^{-m-1} \hat{f}_{ij} \omega_{i+1, -m-1}(x; z)$$

expanded with respect to base points $z_{-n}, z_{-n+1}, \dots, z_{-m-1}$. With notation (4), the coefficients of $r^{(j)}$ are

$$\hat{f}_{ij} = f[z_i, z_{i+1}, \dots, z_{-m-1}, z_{j-m}, z_{j-m+1}, \dots, z_j], \quad i = -n(1) - m - 1.$$

1. For $m = n$ (which implies $r^{(j)} \equiv 0$), a shift transformation leads to new coefficients $\hat{f}_{i,j+1}$, $i = j - n + 1(1)j + 1$, which are completely determined by algorithm (7_{m=n}). In particular, we have the same leading coefficient in both representations: $\hat{f}_{j-n+1, j+1} = \hat{f}_{j-n, j}$. If $z_k = 0$, $k = j - n(1)j$, then (7_n) is the usual Horner algorithm for the evaluation of f at z_{j+1} .

2. For $m < n$, we may keep the base points $z_{-n}, z_{-n+1}, \dots, z_{-m-1}$ fixed, whereas $z_{j-m}, z_{j-m+1}, \dots, z_j$ are shifted as prescribed above in order to transform f into (6). With the notation $\hat{f}_{-m, j} := \hat{f}_{j-m, j}$ the new coefficients

$$\hat{f}_{i, j+1} := f[z_i, z_{i+1}, \dots, z_{-m-1}, z_{j-m+1}, \dots, z_{j+1}], \quad i = -n(1) - m - 1,$$

and the coefficients $\hat{f}_{i, j+1}$, $i = j - m + 1(1)j + 1$, can be computed by the *partial shift algorithm*

$$(8) \quad \begin{aligned} \hat{f}_{-n, j+1} &:= \hat{f}_{-n, j}, \\ \hat{f}_{i+1, j+1} &:= \hat{f}_{i+1, j} + \hat{f}_{i, j+1}(z_{j+1} - z_i) + \hat{f}_{i, j}(z_i - z_{j-m}), \quad i = -n(1) - m - 1, \\ \hat{f}_{j-m+1, j+1} &:= \hat{f}_{-m, j+1}, \\ \hat{f}_{i+1, j+1} &:= \hat{f}_{ij} + \hat{f}_{i, j+1}(z_{j+1} - z_i), \quad i = j - m + 1(1)j. \end{aligned}$$

3. A transformation of

$$f(x) = p_n(x) := \sum_{i=-n}^0 f_{i0} \omega_{i+1, 0}(x; z)$$

into a new expansion

$$\varphi(x) = \sum_{i=-n}^0 \varphi_{i0} \omega_{i+1, 0}(x; \xi)$$

with $n + 1$ arbitrarily chosen points $\xi_{-n}, \xi_{-n+1}, \dots, \xi_0$, may be carried out in the following way:

We expand $a^{(0)}(x) := f(x)$ with respect to $z_{-n+1}, z_{-n+2}, \dots, z_0, \xi_0$, i.e., we apply a shift transformation (7_n) with shift ξ_0 . Renaming the new representation by $a^{(1)}$ we have $\varphi_{00} = a^{(1)}(\xi_0)$ and rewrite $a^{(1)}$ in the form

$$f(x) = \varphi_{00} + a^{(2)}(x) \cdot (x - \xi_0), \quad a^{(2)} \in P_{n-1}.$$

Application of (7_{n-1}) to the polynomial $a^{(2)}(x)$ with shift ξ_{-1} yields $\varphi_{-1, 0} = a^{(2)}(\xi_{-1})$ and $f(x) = \varphi_{00} + [\varphi_{-1, 0} + a^{(3)}(x)(x - \xi_{-1})](x - \xi_0)$. Iteration of this leads to the

general Horner algorithm (see Werner [7]):

$$(9) \left. \begin{aligned} a_k^{(0)} &:= f_{-k,0}, & k &= 0(1)n \\ a_n^{(i+1)} &:= a_n^{(i)}, \\ a_k^{(i+1)} &:= a_k^{(i)} + (\xi_{-i} - z_{-k+i})a_{k+1}^{(i)}, & k &= n-1(-1)i \\ \varphi_{-i,0} &:= a_i^{(i+1)}, \end{aligned} \right\} \quad i = 0(1)n.$$

3. Application to Root Finding Methods. In this section we describe how shift transformations may be applied to approximate zeros of a polynomial $f \in P_n$.

Many well-known iterative methods use divided differences of f : Starting with a number of appropriate estimates $z_{-m}, z_{-m+1}, \dots, z_0$, the j th iteration step of an m th order method consists in

- (1) the computation of values $f_{ij} = f[z_i, z_{i+1}, \dots, z_j]$ for $j - m \leq i \leq j$,
- (2) the construction of a new approximation z_{j+1} (according to the method employed).

Subsequently, the point z_{j-m} is deleted, so that the $m+1$ latest approximation points are always used. A large number of such methods is described by Traub [6], who introduced so-called "one-point iteration functions with memory".

We assume that $f \in P_n$ is given in (or can be transformed into) Newtonian form (1) with $j=0$. The divided differences needed for the construction of z_1 are the coefficients f_{i0} , $i = -m(1)0$. A transformation of f by "shifting" the expansion points $z_{-m}, z_{-m+1}, \dots, z_0$ to $z_{-m+1}, \dots, z_0, z_1$ yields the new values f_{i1} for $i = 1 - m(1)1$. A proper algorithm is given by (7). Continuation yields all the divided differences needed in the iteration process. In particular, we may permit the cases $z_{j-m} = z_{j-m+1} = \dots = z_j$ and define $z_{j+m+1} := z_{j+m} := \dots := z_{j+1}$ applying $m+1$ transformations, each one with the same shift. We then obtain derivative values in place of divided differences. Therefore, Newton's method, Euler's method, etc., are covered by our approach. Note that no division is necessary for a transformation. This prevents amplification of rounding errors which might occur in case of division of small differences.

Several root finding methods may be derived by considering the Newtonian interpolation polynomial

$$p_m(x) := \sum_{i=j-m}^j f_{ij} \omega_{i+1,j}(x; z)$$

which coincides with the given function $f(x) = p_m(x) + r(x)\omega_{j-m,j}(x; z)$ at the points $z_{j-m}, z_{j-m+1}, \dots, z_j$. The new estimate z_{j+1} is defined as a solution β of the equation

$$(10) \quad p_m(x) = 0$$

or, if this is not practicable, at least as a reasonable approximation to β .

In case $m=1$ Eq. (10) reads

$$f_{jj} + f_{j-1,j}(x - z_j) = 0$$

and this leads to a regula falsi step

$$(11) \quad z_{j+1} := z_j - f_{jj}/f_{j-1,j}$$

with convergence order 1.62 (we always indicate the order of convergence only for the case of simple roots).

If $z_{j-1} = z_j$ holds in (11) then $f_{j-1,1} = f'(z_j)$ and we have one step of Newton's method (convergence order 2).

In case $m = 2$ we obtain Muller's method (see, e.g., Traub [6], where the derivation of Muller's iteration function is based on Newtonian interpolation of f): Starting with given points $z_{-2}, z_{-1}, z_0 \in D$ the iteration is defined by

$$(12) \quad \begin{aligned} w_j &:= f_{j-1,j} + f_{j-2,j}(z_j - z_{j-1}) \quad \text{and} \\ z_{j+1} &:= z_j - \frac{2f_{jj}}{w_j \pm \sqrt{w_j^2 - 4f_{jj}f_{j-2,j}}} \end{aligned}$$

for $j = 0, 1, 2, \dots$, where the sign of the square root is that of w_j . The order of convergence is 1.84. If $z_{j-2} = z_{j-1} = z_j$, then the divided differences $f_{j-1,j}$ and $f_{j-2,j}$ coincide with the derivatives $f'(z_j)$ and $\frac{1}{2}f''(z_j)$, respectively. Then formula (12) corresponds to Euler's method (convergence order 3).

However, in general, the cost for calculating $f'(z_j)$ or $f''(z_j)$ is as high as for $f(z_j)$ itself; in our case these values might be obtained by several shift transformations of f , each one with the same shift z_j . For example, two transformations are needed in order to perform one Newton step (convergence order 2). With a similar number of arithmetical operations two regula falsi steps can be carried out and yield convergence order $(1.62)^2 = 2.62$. Similarly, in case $m = 2$ three Muller steps (convergence order $(1.84)^3 = 6.22$) will lead to a better result than one Euler step. A variant of (12) (again with convergence order 1.84) which avoids the calculation of a square root is suggested by Traub [6]:

$$(13) \quad z_{j+1} := z_j - \frac{f_{j-1,j}}{f_{j-1,j}^2 - f_{j-1,j-1}f_{j-2,j}} f_{jj}.$$

A generalization of (12) results from modifying a family of root finding methods described by Hansen and Patrick [2]. With w_j from (12) and $\alpha \in \mathbf{R}$ we define

$$(14) \quad z_{j+1} := \begin{cases} z_j - \frac{(\alpha + 1)f_{jj}}{\alpha w_j \pm \sqrt{w_j^2 - 2(\alpha + 1)f_{jj}f_{j-2,j}}} & \text{if } \alpha \neq -1, \\ z_j - f_{jj}w_j / (w_j^2 - f_{jj}f_{j-2,j}) & \text{if } \alpha = -1. \end{cases}$$

It can be shown that the order of convergence is 1.84 independent of α .

The case $z_{j-2} = z_{j-1} = z_j$ corresponds to a Hansen/Patrick step. As remarked in [2] some special values of α lead to well-known iteration methods; e.g., we have for

| | |
|-----------------------------|--------------------------------------|
| $\alpha = 1$ | Euler's method, |
| $\alpha = \frac{1}{n-1}$ | Laguerre's method (if $f \in P_n$), |
| $\alpha = 0$ | Ostrowski's square root iteration, |
| $\alpha = -1$ | Halley's method, |
| $\alpha \rightarrow \infty$ | Newton's method. |

For finite values of α the Hansen/Patrick methods have convergence order 3.

In (14) the case $\alpha = 1$ is equivalent to a Muller step (12). If we replace w_j in (14) by $f_{j-1,j}$ and set $\alpha := -1$ we obtain Traub's formula (13).

Since rounding errors in the coefficients of the Newtonian form usually will influence the accuracy of the computed roots, we end this section with a result from perturbation theory. For simplicity, we confine ourselves to the case of simple roots of a polynomial

$$f(x) := \sum_{i=j-n}^j f_{ij} \omega_{i+1,j}(x; z) \in P_n.$$

From numerical computations we may obtain the polynomial

$$\tilde{f}(x) := \sum_{i=j-n}^j \tilde{f}_{ij} \omega_{i+1,j}(x; z),$$

where the coefficients \tilde{f}_{ij} differ from the exact data f_{ij} by

$$\Delta f_{ij} := f_{ij} - \tilde{f}_{ij}, \quad i = j - n(1)j.$$

If some bounds are known for Δf_{ij} , $i = j - n(1)j$, then the following lemma may help to answer the question whether a root $\tilde{\lambda}$ of \tilde{f} is still a "good approximation" to a root of f .

LEMMA 3. *If $\tilde{\lambda}$ is a simple root of \tilde{f} , and if $f'(\tilde{\lambda}) \neq 0$ then there exists a root λ of f with*

$$|\lambda - \tilde{\lambda}| \leq \frac{n}{|f'(\tilde{\lambda})|} \left| \sum_{i=j-n}^j (\Delta f_{ij}) \omega_{i+1,j}(\tilde{\lambda}; z) \right|.$$

Proof. This follows immediately from Laguerre's Theorem (see, e.g., [3]), which implies

$$|\lambda - \tilde{\lambda}| \leq n \cdot |f(\tilde{\lambda})/f'(\tilde{\lambda})|,$$

and from $f(\tilde{\lambda}) = f(\tilde{\lambda}) - \tilde{f}(\tilde{\lambda}) = \sum_{i=j-n}^j (\Delta f_{ij}) \omega_{i+1,j}(\tilde{\lambda}; z)$. \square

Remark. For a polynomial f of exact degree n the following condition number of a simple root $\lambda \neq 0$ is suggested by Gautschi [1]:

$$\text{cond}(\lambda) := \frac{1}{|\lambda| |f'(\lambda)|} \sum_{i=j-n+1}^j |f_{ij}| \cdot |\omega_{i+1,j}(\lambda; z)|.$$

It indicates the sensitivity of λ with respect to small perturbations Δf_{ij} in the coefficients f_{ij} , $i = j - n + 1(1)j$. The error $\Delta f_{j-n,j}$ is assumed to be zero. We see that the size of $\text{cond}(\lambda)$ depends on the location of the base points $z_{j-n+1}, z_{j-n+2}, \dots, z_j$ and becomes small, for instance, if z_j is close to λ . If there are bounds for Δf_{ij} of the form

$$|\Delta f_{ij}| \leq \delta \cdot |f_{ij}|, \quad i = j - n + 1(1)j, \quad \delta \geq 0,$$

then we get from Lemma 3:

$$(15) \quad \frac{|\lambda - \tilde{\lambda}|}{|\lambda|} \leq n \cdot \delta \cdot \text{cond}(\lambda) + O(\delta^2).$$

4. On Monotonic Convergence of Larkin's Method. In [5], Larkin suggests a class of methods (from rational interpolation) to approximate simple zeros of meromorphic functions $f: D \rightarrow \mathbb{C}$. The rational function interpolating f at the $m + 1$ points $z_{j-m}, z_{j-m+1}, \dots, z_j$ has the form

$$(16) \quad (x - \beta)/Q(x),$$

where $\beta \in \mathbb{C}$ and $Q \in P_{m-1}$, $Q(\beta) \neq 0$. The new approximation z_{j+1} to the required zero of f is defined as the unique zero β of (16). Henceforth, we exclude the trivial case that $f(z_k) = 0$ for some k , $j - m \leq k \leq j$. Larkin gives a simple scheme for the computation of z_{j+1} in terms of the function

$$\bar{f}(x) := \frac{1}{f(x)}$$

and its divided differences $\bar{f}_{j-m,k} = \bar{f}[z_{j-m}, z_{j-m+1}, \dots, z_k]$, $k = j - m(1)j$.

In our terms we can write the function \bar{f} in Newtonian form similar to (1) but with the base points in "reverse order":

$$(17) \quad \bar{f}(x) = \sum_{i=j-m}^j \bar{f}_{j-m,i} \omega_{j-m,i-1}(x; z) + \bar{r}(x) \cdot \omega_{j-m,j}(x; z).$$

For $\bar{f}_{j-m,j} \neq 0$ a "Larkin step of order m " is defined by

$$(18) \quad z_{j+1} := z_j + \bar{f}_{j-m,j-1}/\bar{f}_{j-m,j};$$

as is shown in [5] z_{j+1} is the unique zero of (16).

An iteration method based on formula (18) may consist of Larkin steps of various order, e.g., one can start with initial points z_{-1}, z_0 ($m = 1$) and add the new estimate z_{j+1} in each step without deleting z_{j-m} ($m \rightarrow m + 1$). Thus all the previous information is used during the iteration. Asymptotic quadratic convergence for this procedure is proved in [4]. For fixed m the order of convergence is given by the positive root of $t^m - \sum_{i=0}^{m-1} t^i = 0$.

The following special cases show that this class of methods contains iteration functions similar to those in Section 3. Moreover, we obtain from Lemma 4 below, that a Larkin step can easily be carried out applying divided differences of the function f directly.

Special cases. If $m = 1$, then formula (18) results in a regula falsi step (11) applied to the function f .

The case $m = 2$ leads to

$$z_{j+1} := z_j - \frac{f_{j-2,j-1}}{f_{j-2,j-1}f_{j-1,j} - f_{j-1,j-1}f_{j-2,j}} f_{jj},$$

which is quite similar to (13).

The next lemma shows how the coefficients f_{ij} of the Newtonian form (1) are related to $\bar{f}_{j-m,k}$ of (17).

LEMMA 4. Let the functions $f, \bar{f}: D \rightarrow \mathbb{C}$ with $\bar{f} = 1/f$ be analytic at the points $z_{j-m}, z_{j-m+1}, \dots, z_j$. If f and \bar{f} are given in Newtonian form (1) and (17), respectively, then

$$\sum_{i=j-m}^j f_{ij} \bar{f}_{j-m,i} = 0$$

for $m > 0$.

Proof. Writing $f = p_m + r \cdot \omega_{j-m,j}$ and $\tilde{f} = \bar{p}_m + \bar{r} \cdot \omega_{j-m,j}$, we have

$$\begin{aligned} p_m(x) \bar{p}_m(x) &= \sum_{i,l=j-m}^j f_{ij} \tilde{f}_{j-m,i} \omega_{j-m,i-1}(x; z) \omega_{i+1,j}(x; z) \\ &= \hat{p}_m(x) + \hat{r}(x) \omega_{j-m,j}(x; z), \end{aligned}$$

where

$$\hat{p}_m(x) = \sum_{\substack{l \leq i \leq j \\ j-l \leq m}} f_{ij} \tilde{f}_{j-m,i} \omega_{j-m,i-1}(x; z) \omega_{i+1,j}(x; z) \in P_m$$

and

$$\hat{r}(x) = \sum_{\substack{i < l \leq j \\ j-l \leq m}} f_{ij} \tilde{f}_{j-m,i} \omega_{i+1,i-1}(x; z).$$

Rewriting \hat{p}_m in the form (1) with respect to the points $z_{j-m}, z_{j-m+1}, \dots, z_j$, we obtain

$$\hat{p}_m(x) = \hat{p}_{m-1}(x) + \omega_{j-m+1,j}(x; z) \sum_{i=j-m}^j f_{ij} \tilde{f}_{j-m,i}$$

with a unique polynomial $\hat{p}_{m-1} \in P_{m-1}$. Thus we have

$$f\tilde{f} = \hat{p}_{m-1} + \omega_{j-m+1,j} \cdot \sum_{i=j-m}^j f_{ij} \tilde{f}_{j-m,i} + (\hat{r} + p_m \bar{r} + \bar{p}_m r + r \bar{r} \omega_{j-m,j}) \omega_{j-m,j},$$

which is a unique Newtonian representation (1) of the product $f\tilde{f}$. Since $f\tilde{f} = 1$, the remainder function in parentheses must be zero just as all the coefficients of the polynomial part except for the constant term. In particular, we obtain

$$\sum_{i=j-m}^j f_{ij} \tilde{f}_{j-m,i} = 0. \quad \square$$

If the divided differences $f_{ik} = f[z_i, z_{i+1}, \dots, z_k]$ are given for $k = j - m + 1(1)j$ and $i = j - m(1)k$, then one step of Larkin's method can be performed as follows:

With arbitrary $f_{j-m,j-m}^* \neq 0$ we compute the values $\tilde{f}_{j-m,k}^*$ for $k = j - m + 1(1)j$ recursively from

$$(19) \quad \sum_{i=j-m}^k f_{ik} \tilde{f}_{j-m,i}^* = 0$$

and define

$$z_{j+1} := z_j + \tilde{f}_{j-m,j-1}^* / \tilde{f}_{j-m,j}^*.$$

This is equivalent to (18) since the values $\tilde{f}_{j-m,i}^*$ differ from $\tilde{f}_{j-m,i}$ only by a common factor depending on the initial choice of $\tilde{f}_{j-m,j-m}^*$.

Remark. For fixed $j \geq 0$ and $k = j - m + 1(1)j$ we define the points

$$z_{k+1}^* := z_k + \tilde{f}_{j-m,k-1}^* / \tilde{f}_{j-m,k}^*$$

according to Larkin steps of order $k + m - j$. In particular, we have $z_{j+1}^* = z_{j+1}$.

Writing equality (19) as

$$\sum_{i=j-m}^k f_{ik} \prod_{l=i+1}^k \frac{\tilde{f}_{j-m,l-1}^*}{\tilde{f}_{j-m,l}^*} = 0$$

we obtain

$$(20) \quad \sum_{i=j-m}^k f_{ik} \prod_{l=i+1}^k (z_{l+1}^* - z_l) = 0$$

for $k = j - m + 1(1)j$. Note that (20) yields another algorithm for the construction of $z_{j+1} = z_{j+1}^*$ by computing the differences $z_{k+1}^* - z_k$ from linear equations. It shows that one Larkin step of order m is composed of m steps of lower order.

We now assume that $f \in P_n$ has real coefficients and only real roots $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. If one starts Larkin's method with initial points $z_{-m}, z_{-m+1}, \dots, z_0 \in \mathbf{R}$, then all the approximations z_j constructed by (18) will be real. Our goal is to prove that if all initial points $z_{-m}, z_{-m+1}, \dots, z_0$ exceed λ_n , then we have monotonic convergence to the largest root λ_n of f . (Of course we then also have monotonic convergence from below in the "dual" case $z_k \leq \lambda_1, k = -m(1)0$.)

It is necessary for monotonicity that the correction terms $\tilde{f}_{j-m,j-1}^*/\tilde{f}_{j-m,j}^*$ always have the same sign. This property will be shown in Lemma 6 by induction with respect to the polynomial degree n . We write f in the form

$$f(x) = (x - \lambda)g(x)$$

or, equivalently,

$$\bar{g}(x) = (x - \lambda)\tilde{f}(x),$$

where $\bar{g} := 1/g, \tilde{f} := 1/f$, and $x \neq \lambda$. The next lemma gives a general relation between the coefficient of \bar{g} and \tilde{f} in case both functions are represented in the form (17).

LEMMA 5. Let the function $\tilde{f}: D \rightarrow \mathbf{C}$ be analytic at $z_{j-m}, z_{j-m+1}, \dots, z_j$, and let $\bar{g}: D \rightarrow \mathbf{C}$ be defined by

$$\bar{g}(x) := (x - \lambda)\tilde{f}(x), \quad \lambda \in \mathbf{C}.$$

Then \tilde{f} and \bar{g} can be represented in Newton form (17) with respect to $z_k, k = j - m(1)j$, and

$$(21) \quad \bar{g}_{j-m,i} = \tilde{f}_{j-m,i-1} + (z_i - \lambda)\tilde{f}_{j-m,i}$$

for $i = j - m + 1(1)j$.

Proof. We write $\tilde{f}(x) = \bar{p}_m(x) + r_f(x)\omega_{j-m,j}(x; z)$ with

$$\bar{p}_m(x) = \sum_{i=j-m}^j \tilde{f}_{j-m,i}\omega_{j-m,i-1}(x; z).$$

The function \bar{g} (which is also analytic at $z_k, k = j - m(1)j$) is uniquely represented in Newton form (17) by

$$\begin{aligned} \bar{g}(x) &= \sum_{i=j-m}^j \bar{g}_{j-m,i} \omega_{j-m,i-1}(x; z) + r_{\bar{g}}(x) \omega_{j-m,j}(x; z) \\ &= \sum_{i=j-m}^j \bar{f}_{j-m,i}(x - \lambda) \omega_{j-m,i-1}(x; z) + r_f(x)(x - \lambda) \omega_{j-m,j}(x; z) \\ &= \sum_{i=j-m+1}^j \bar{f}_{j-m,i-1} \omega(x; z_{j-m}, z_{j-m+1}, \dots, z_{j-2}, \lambda) \\ &\quad + (r_f(x)(x - z_j) + \bar{f}_{j-m,j}) \omega(x; z_{j-m}, z_{j-m+1}, \dots, z_{j-1}, \lambda). \end{aligned}$$

From Lemma 2 we get

$$\bar{f}_{j-m,i-1} = \bar{g}_{j-m,i} + (\lambda - z_i) \bar{f}_{j-m,i}, \quad i = j - m + 1(1)j. \quad \square$$

Remark. As in Lemma 2 we do not require \bar{g} to be analytic at $x = \lambda$.

LEMMA 6. Let $f \in P_n$ be a real polynomial with real roots $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. Let the function $\bar{f} = 1/f$ be represented in Newton form (17) with respect to the points $z_{j-m}, z_{j-m+1}, \dots, z_j$. If

$$(22) \quad z_k > \lambda_n$$

holds for $k = j - m(1)j$, then

$$(23) \quad (-1)^{i-(j-m)} \bar{f}_{j-m,i} \cdot \bar{f}_{j-m,j-m} \geq 0$$

for $i = j - m(1)j$.

Proof. (a) Inequality (23) obviously holds for $n = 0$: In case $f \equiv 1$ we have $(-1)^0 \bar{f}_{j-m,i-m}^2 = 1 > 0$ and $\bar{f}_{j-m,i} = 0$ for $j - m + 1 \leq i \leq j$.

(b) We assume that (23) is valid for all polynomials $g \in P_{n-1}$, i.e., with $\bar{g} = 1/g$ we have $(-1)^{i-(j-m)} \bar{g}_{j-m,i} \cdot \bar{g}_{j-m,j-m} \geq 0$ for $m \geq 1$ arbitrary and $i = j - m(1)j$. We set $f(x) := (x - \lambda)g(x) \in P_n$ with $\lambda \in \mathbf{R}, \lambda < z_k$ for $k = j - m(1)j$. Then $\bar{f} = 1/f$ is analytic at the points z_k , and since $\bar{g}(x) = (x - \lambda)\bar{f}(x)$ we get from Lemma 5:

$$\begin{aligned} &(-1)^{i-(j-m)} \bar{f}_{j-m,i} \bar{f}_{j-m,j-m} \\ &= \frac{1}{z_i - \lambda} \left((+1)^{i-j-m} \bar{g}_{j-m,i} \frac{\bar{g}_{j-m,j-m}}{z_{j-m} - \lambda} + (-1)^{i-1-j+m} \bar{f}_{j-m,i-1} \bar{f}_{j-m,j-m} \right). \end{aligned}$$

From $\bar{f}_{j-m,j-m}^2 \geq 0$ it follows (by induction with respect to i) that the above expression is nonnegative for $i = j - m(1)j$. \square

Combining Lemma 5 and Lemma 6 we obtain the announced monotonicity property of Larkin's method:

THEOREM. Let $f \in P_n$ be a real polynomial with real roots $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. If $z_k > \lambda_n$ holds for all initial points $z_k, k = -m(1)0$, then a sequence defined by (18) is monotonically decreasing and converges to λ_n .

Proof. We shall show that the assumption (22)

$$\lambda_n < z_k \quad \text{for } k = j - m(1)j$$

(which holds in case $j = 0$) implies $\lambda_n \leq z_{j+1} \leq z_j$, so that monotonic convergence follows by induction. Writing f in the form $f(x) = (x - \lambda_n)g(x)$ we have from relation (21) and definition (18):

$$\lambda_n + \frac{\bar{g}_{j-m,j}}{\bar{f}_{j-m,j}} = z_j + \frac{\bar{f}_{j-m,j-1}}{\bar{f}_{j-m,j}} = z_{j+1}.$$

Inequality (23) yields $\bar{f}_{j-m,j-1}/\bar{f}_{j-m,j} \leq 0$ and, again from (23), we obtain $\bar{g}_{j-m,j}/\bar{f}_{j-m,j} \geq 0$, since \bar{f} and \bar{g} have the same sign at z_{j-m} . This implies $\lambda_n \leq z_{j+1} \leq z_j$. The monotonically decreasing and bounded iteration sequence tends to a limit $\lambda^* \geq \lambda_n$. Thus $\lim_{j \rightarrow \infty} (z_{j+1} - z_j) = 0$ and setting $k = m$ in relation (20) we have $\lim_{j \rightarrow \infty} f(z_j) = 0$. The continuity of f implies $f(\lambda^*) = 0$. Hence $\lambda^* = \lambda_n$, which completes the proof. \square

It can be shown that Larkin's method also converges monotonically to intermediate roots of real polynomials with only real roots provided that we choose the order m even, and the starting points between two roots such that special initial divided differences have the same sign. One can use this property to determine the most sensitive roots first.

5. Numerical Tests. We consider always real polynomials $f \in P_n$ with real roots $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. The monotonic behavior of an approximation sequence $\{z_j\}$, $j \geq 0$, yields a simple criterion for the termination of the iterative approximation of a root λ of f . For example, if $\lambda < z_{j+1} < z_j$ holds theoretically, then we stop iterating as soon as the computed values satisfy $z_{j+1} \geq z_j$. Then we may regard $\bar{\lambda} := z_j$ as an acceptable approximation to λ with respect to the machine accuracy. If shift transformations (7_n) are applied to f during the iteration, then we obtain f in the form

$$(24) \quad f(x) = f(\bar{\lambda}) + (x - \bar{\lambda})g(x)$$

with the deflated polynomial

$$g(x) = \sum_{i=j-n}^{j-1} f_{ij} \omega_{i+1,j-1}(x; z).$$

By deleting $f(\bar{\lambda}) \approx 0$ we can immediately continue with $g(x)$ to approximate another root of f . If, for example, λ_n is approximated from above, then the points $z_{j-m-1}, z_{j-m}, \dots, z_{j-1}$ exceed the largest root λ_{n-1} of g . If we choose these iterates as new starting values, the monotonicity property guarantees convergence to λ_{n-1} . In this way all the roots can be approximated in decreasing order.

In finite-precision arithmetic the deflation process will generally change the remaining zeros: A transformation into the form (24) is affected by rounding errors, and further perturbations occur by deleting the small nonzero value $f(\bar{\lambda})$, if $\lambda \neq \bar{\lambda}$.

Generally, the remaining roots can be regarded as exact roots of a perturbed polynomial $\bar{f} \in P_n$. As indicated in Section 3, the sensitivity of the roots with respect to perturbations in the polynomial coefficients is determined by their condition number (see (15)) which depends on the representation of f . Thus we have to take care that transformations do not destroy a well-conditioned form of f , especially if the base points of f are shifted step by step to compute the divided differences.

For example, if we already have base points z_k close to roots $\mu_k \neq \lambda$ for some k , then we can keep them fixed during the approximation of λ . This is possible by

using partial shift transformations (8). Conversely, if $f \in P_n$ is given in ill-conditioned form, then appropriate transformations shifting all base points to approximations of the roots will improve the condition number of the roots. In this case we can hope that later transformations and deflation steps will not change these roots too much. This is confirmed by Example 1 below.

Example 1. We consider Larkin's method given by (18) and choose $m = 2$ (convergence order 1.84). Two versions of the method are compared with each other:

(i) The usual definition according to (2) and (3) is used to compute the divided differences f_{ij} and \tilde{f}_{ij} for $i = j(-1)j - m$. Always, the same given representation of $f \in P_n$ (or of the computed deflated polynomials of smaller degree) is evaluated. We speak of the "nonshift version" of the iteration method.

(ii) A "shift version" computes the divided differences f_{ij} , $i = j - m(1)j$, applying algorithm (7_n) ("complete shifting") or algorithm (8) ("partial shifting"). The shift version of Larkin's method uses (19) or (20).

Let f_1 be a polynomial of degree $n = 16$ with the roots $\lambda_i = i + 0.1$, $i = 1(1)16$, given in standard form (all expansion points 0, leading coefficient 1). To compute all the roots of f_1 the first starting values were chosen near $x = 0$; after a deflation step the iteration was restarted with some of the previous approximation points z_{j-1} , z_{j-2}, \dots . For this choice our theorem guarantees monotonic convergence from below. Therefore, the zeros are determined in increasing order, which usually keeps the deflation stable (see Wilkinson [8], Jenkins and Traub [9]). An iteration was terminated if monotonicity was violated. The computations were performed in double precision (60 binary digit mantissa) on a UNIVAC 1100/82.

We consider the maximum relative error

$$e := \max_{i=1(1)16} \frac{|\lambda_i - \tilde{\lambda}_i|}{|\lambda_i|},$$

where $\tilde{\lambda}_i$ denotes the numerical approximation to the root λ_i . Both versions were tested several times with slightly changed initial points. As average values for e we obtained

$$e = 4.8 \cdot 10^{-9} \quad \text{for the nonshift version,}$$

$$e = 8.5 \cdot 10^{-9} \quad \text{for the shift version.}$$

In many cases the relative error became maximal for the approximation of $\lambda_{12} = 12.1$. In fact, λ_{12} is one of the most sensitive roots of f_1 as indicated by its condition number ($\approx 3 \cdot 10^{10}$). Working with 60 binary digit precision we can hardly expect more than 9 correct decimal figures in the computed $\tilde{\lambda}_{12}$, so that the results are quite satisfactory with respect to the limiting accuracy. This was confirmed when we tried to improve the computed values $\tilde{\lambda}_i$ by choosing them as starting values for a series of additional approximation steps. In this procedure we always iterated in the original polynomial f_1 in order to prevent error propagation arising from shift transformations or deflation steps. For a particular root one or two regula falsi steps (11) were applied. The relative error in λ_{12} , for example, could be reduced only to $2 \cdot 10^{-9}$. This shows that the deflation process did not diminish the attainable accuracy too much. Moreover, a moderate number of shift transformations seems to be numerically not dangerous. In this example, we had an average number of 10 iteration steps per root, which implies a total of roughly 150 shift transformations.

Remark. In case of Larkin's nonshift version the computation of the divided differences $\tilde{f}_{j-m,i}$, $i = j - m(1)j$, requires $O(m)$ operations, but we need $O(m^2)$ operations to obtain the values \tilde{f}_{ij}^* in (19) or the difference $z_{k+1}^* - z_k$ in (20) solving a triangular system. Thus it is cheaper to use the original (nonshift) version of Larkin's method for large m . Yet, if we confine ourselves to order $m \leq 3$ the shift version is more economical.

Example 2. We consider the iteration method defined by formula (14) and Larkin's method with $m = 2$ in order to test their behavior near both simple and multiple roots. Shift transformations are used to determine the roots of

$$f_2(x) := (x + 1)^2(x - 1)^2(x - 3).$$

We set $\alpha := 0.25$ in (14). The results are listed in Table 1.

TABLE 1
Approximation of the simple root $\lambda_5 = 3$ and the double root $\lambda_3 = \lambda_4 = 1$.

| Iteration index | Method defined by (14), $\alpha = 0.25$ | Larkin's method ($m = 2$) |
|-----------------|---|-----------------------------|
| 0 | 3.3 | 3.3 |
| 1 | 3.0006114 | 3.0203753 |
| 2 | 2.9999442 | 3.0016795 |
| 3 | 3.0000000 | 3.0000117 |
| 4 | | 3.0000000 |
| 0 | 1.1 | 1.1 |
| 1 | 1.0269207 | 1.0344681 |
| 2 | 0.9936201 | 1.0208248 |
| 3 | 0.9967069 | 1.0116232 |
| ⋮ | ⋮ | ⋮ |
| 14 | 1.0000000 | 1.0000141 |

The approximation of the double root $\lambda_1 = \lambda_2 = -1$ also shows a superiority of formula (14). Similar results have been obtained for other values $\alpha \in [0, 1]$, e.g., for $\alpha = 1$ (Muller's method). For $\alpha < 0$ the speed of convergence decreases.

Example 3. In case $z_j = z_{j-1} = z_{j-2}$, formula (14) with $\alpha := 1/(n - 1)$ corresponds to Laguerre's method (for $f \in P_n$), which converges very fast for large starting values $|z_0| \gg |\lambda_i|$, $i = 1(1)n$. If z_j , z_{j-1} , and z_{j-2} are large and distinct, we do not know the best choice for α in general. Considering the polynomial $f(z) := z^n$ and, for example, $z_j := cz_{j-1} := c^2z_{j-2}$, $0 < c \leq 1$, one calculates that the optimal value of α (for which formula (14) yields $z_{j+1} = \lambda = 0$) lies between $1/(n - 1)$ and 1, and α tends to $1/(n - 1)$ for $c \rightarrow 1$. Since numerical tests with (14) show that the average number of iterations is about the same for all $\alpha \in [0, 1]$, it is reasonable to iterate always with $\alpha := 1/(n - 1)$. With this choice we tested the behavior of (14) and compare it again with Larkin's method (see Table 2). Since the operation count for the evaluation of (14) is as high as for a Larkin step of order 4, we choose $m = 4$ in (18). The considered polynomial is f_2 from Example 2. To keep the iterates real we

applied a regula falsi step (11) if the square root in (14) became imaginary (this case occurred only once in the example).

TABLE 2

Start with large $|z_0|$. The notation "(r.f.)" indicates a regula falsi step (see text).

| Iteration index | Method defined by (14), $\alpha = 0.25$ | Larkin's method ($m = 4$) |
|-----------------|---|-----------------------------|
| 0 | 100 | 100 |
| 1 | 3.5431 | 50.3 |
| 2 | 3.5431 | 39.6 |
| 3 | 3.2172 (r.f.) | 30.7 |
| 4 | 2.9160 | 23.6 |
| ⋮ | ⋮ | ⋮ |
| 8 | 3.0000000 | 8.17 |

To approximate the root $\lambda_5 = 3$ up to 8 decimal digits Larkin's method needs 17 iteration steps, more than twice as much as formula (14).

Added in proof. In the meantime we noticed that Larkin's rootfinding method has been discovered already in 1958 by G. Opitz, *Z. Angew. Math. Mech.*, v. 38, 1958, pp. 276–277; cf. also G. Opitz, *Z. Angew. Math. Mech.*, v. 41, 1961, pp. T48–T50.

Institut für Angewandte Mathematik
Universität Freiburg i. Br.
D-7800 Freiburg i. Br.
West Germany

1. W. GAUTSCHI, "On the condition of algebraic equations," *Numer. Math.*, v. 21, 1973, pp. 405–424.
2. E. HANSEN & M. PATRICK, "A family of root finding methods," *Numer. Math.*, v. 27, 1977, pp. 257–269.
3. P. HENRICI, *Applied and Computational Complex Analysis*, Wiley, New York, 1974.
4. F. M. LARKIN, "Root-finding by fitting rational functions," *Math. Comp.*, v. 35, 1980, pp. 803–816.
5. F. M. LARKIN, "Root-finding by divided differences," *Numer. Math.*, v. 37, 1981, pp. 93–104.
6. J. F. TRAUB, *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, N.J., 1964.
7. W. WERNER, *On the Simultaneous Determination of Polynomial Roots*, Lecture Notes in Math., Vol. 953, Springer-Verlag, Berlin and New York, 1982, pp. 188–202.
8. J. H. WILKINSON, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, N.J., 1963.
9. M. A. JENKINS & J. F. TRAUB, "A three-stage algorithm for real polynomials using quadratic iteration," *SIAM J. Numer. Anal.*, v. 7, 1970, pp. 545–566.