SIAM J. NUMER. ANAL.
Vol. 7, No. 4, December 1970

A THREE-STAGE ALGORITHM FOR REAL POLYNOMIALS USING QUADRATIC ITERATION*

M. A. JENKINS† AND J. F. TRAUB‡

Abstract. We introduce a new three-stage process for calculating the zeros of a polynomial with real coefficients. The algorithm finds either a linear or quadratic factor, working completely in real arithmetic. In the third stage the algorithm uses one of two variable-shift iterations corresponding to the linear or quadratic case. The iteration for a linear factor is a real arithmetic version of the third stage of the algorithm for complex polynomials which we studied in an earlier paper. A new variable-shift iteration is introduced in this paper which is suitable for quadratic factors. If the complex algorithm and the new real algorithm are applied to the same real polynomial, then the real algorithm is about four times as fast.

We prove that the mathematical algorithm always converges and show that the rate of convergence of the third stage is faster than second order.

The problem and algorithm may be recast into matrix form. The third stage is a quadratic form of shifted inverse powering and a quadratic form of generalized Rayleigh iteration.

The results of extensive testing are summarized. For an ALGOL W program run on an IBM 360/67 we found that for polynomials ranging in degree from 20 to 50, the time required to calculate all zeros averaged $2n^2$ milliseconds.

An ALGOL 60 implementation of the algorithm and a program which calculates a posteriori bounds on the zeros may be found in Jenkins' 1969 Stanford dissertation [2].

TABLE OF CONTENTS

1. Introduction.

We introduce a new three-stage algorithm for calculating the zeros of a *real* polynomial P,

$$P(z) = \sum_{i=0}^{n} a_i z^{n-i}, \qquad a_0 = 1, \quad a_n \neq 0,$$

$$P(z) = \prod_{i=1}^{j} (z - \rho_i)^{m_i}.$$

* Received by the editors August 26, 1969, and in revised form December 26, 1969.

† Queen's University, Kingston, Ontario, Canada, and IBM Scientific Center Complex, Yorktown Heights, New York 10598.

‡ Bell Telephone Laboratories, Incorporated, Murray Hill, New Jersey. Now at Computer Science Department, University of Washington, Seattle, Washington 98105.

The condition $a_0 = 1$ is for convenience only. The algorithm uses only real arithmetic, even when calculating complex conjugate zeros.

In Jenkins and Traub [3] we studied an algorithm for polynomials with complex coefficients which is similar in spirit to two-stage algorithms proposed by Traub [6], [7], [8]. The results of [3] apply, of course, to the special case of real coefficients. However, the complex algorithm uses iteration in the complex plane. If the complex algorithm and the new real algorithm are applied to the same real polynomial, then the real algorithm is about four times as fast.

The real algorithm has the following desirable characteristics.

(i) The mathematical algorithm always converges to a linear or quadratic factor.

(ii) Zeros are calculated in roughly increasing order of modulus; this avoids the instability which occurs when the polynomial is deflated with a large zero. (However, this ordering of the deflation is not sufficient to ensure deflation stability for all polynomials. A discussion may be found in § 10.)

(iii) Few critical decisions have to be made by the program which implements the algorithm.

(iv) Only real arithmetic is used. Complex conjugate zeros are found as quadratic factors.

Certain sequences of polynomials called H polynomials play a basic role in the complex algorithm. In this paper a different sequence of polynomials plays a basic role. We call these K polynomials.

We summarize the contents of this paper. The main properties of fixed-shift K polynomials are given in § 2. Variable-shift K polynomials are introduced in § 3 and are used to define a variable-shift iteration for a quadratic factor. Sufficient conditions for convergence of this iteration are given in § 4 and the quadratic character of convergence is established in § 5. The three-stage algorithm is given in § 6. The third stage consists of either a real arithmetic version of the variable-shift iteration given in [3] to obtain a linear factor or the variable-shift iteration for a quadratic factor given in § 3. Global convergence of the three-stage algorithm for an arbitrary distribution of zeros is proved in § 7.

In § 8 we recast the problem and algorithm in matrix form and prove that the variable-shift iteration for a quadratic factor may be viewed as an efficient process for carrying out a quadratic form of shifted inverse powering and a quadratic form of generalized Rayleigh iteration.

Our focus in this paper is on the mathematical algorithm and its properties. The program implementing the algorithm, the results of extensive testing, and a program which clusters the zeros and provides a posteriori error bounds may be found in Jenkins [2]. In § 9 we do discuss how the programs make the critical decisions and how certain calculations can be carried out efficiently. In the final section we summarize the results of extensive testing and give a small numerical example.

2. Fixed-shift K polynomials. We introduce fixed-shift K polynomials and prove a number of their properties. Let $\sigma(z) = z^2 + uz + v$ be a real quadratic polynomial with zeros s_1 and s_2 such that $s_1 \neq s_2$ and $P(s_1)P(s_2) \neq 0$. Let $K^{(0)}(z)$

be a polynomial of degree at most $n - 1$. Define the sequence

$$K^{(\lambda+1)}(z) = \frac{1}{\sigma(z)}[K^{(\lambda)}(z) + (Az + B)P(z)], \qquad \lambda = 0, 1, \cdots,$$

where A and B are chosen so that $\sigma(z)$ exactly divides the expression

$$[K^{(\lambda)}(z) + (Az + B)P(z)].$$

Then

$$K^{(\lambda+1)}(z) = \frac{1}{\sigma(z)}\left[K^{(\lambda)}(z) - \left[\frac{z - s_2}{s_1 - s_2}\frac{K^{(\lambda)}(s_1)}{P(s_1)} + \frac{z - s_1}{s_2 - s_1}\frac{K^{(\lambda)}(s_2)}{P(s_2)}\right]P(z)\right], \tag{2.1}$$

and $K^{(\lambda+1)}(z)$ is of degree at most $n - 1$. We can rewrite (2.1) as

$$K^{(\lambda+1)}(z) = \frac{1}{\sigma(z)}\left[K^{(\lambda)}(z) + \frac{\begin{vmatrix} P(s_1) & P(s_2) \\ K^{(\lambda)}(s_1) & K^{(\lambda)}(s_2) \end{vmatrix} z + \begin{vmatrix} K^{(\lambda)}(s_1) & K^{(\lambda)}(s_2) \\ s_1P(s_1) & s_2P(s_2) \end{vmatrix}}{\begin{vmatrix} s_1P(s_1) & s_2P(s_2) \\ P(s_1) & P(s_2) \end{vmatrix}} P(z)\right]. \tag{2.2}$$

Since $\sigma(z) = (z - s_1)(z - s_2)$ is real, s_1 and s_2 are either both real or a complex conjugate pair. In either case, the linear factor multiplying $P(z)$ is real.

An alternative derivation of (2.2) is to consider a double-shift technique similar to that used in the QR algorithm. If we define

$$\begin{aligned} K^{(\lambda+1/2)}(z) &= \frac{1}{z - s_1}\left[K^{(\lambda)}(z) - \frac{K^{(\lambda)}(s_1)}{P(s_1)}P(z)\right], \\ K^{(\lambda+1)}(z) &= \frac{1}{z - s_2}\left[K^{(\lambda+1/2)}(z) - \frac{K^{(\lambda+1/2)}(s_2)}{P(s_2)}P(z)\right], \end{aligned} \tag{2.3}$$

then one may easily verify that this is equivalent to (2.2). Hence we may draw on our knowledge of the single-shift recurrence to prove properties of the K polynomial sequence.

The properties of $K^{(\lambda)}(z)$ follow from the following lemma which is easily proved by induction.

LEMMA 2.1. *Let*

$$K^{(0)}(z) = \sum_{i=1}^{j} c_i^{(0)}P_i(z), \qquad P_i(z) = \frac{P(z)}{z - \rho_i}.$$

Let

$$\sigma_i = \sigma(\rho_i) = (\rho_i - s_1)(\rho_i - s_2).$$

Then for all λ,

$$K^{(\lambda)}(z) = \sum_{i=1}^{j} c_i^{(0)}\sigma_i^{-\lambda}P_i(z). \tag{2.4}$$

We introduce some additional notation. Let

$$K_0^{(\lambda)} = K^{(\lambda)}(z),$$
$$K_{v+1}^{(\lambda)}(z) = \frac{1}{z}\left[K_v^{(\lambda)}(z) - \frac{K_v^{(\lambda)}(0)}{P(0)}P(z)\right], \qquad v = 0, 1. \tag{2.5}$$

These polynomials are the first three members of a no-shift H polynomial sequence started with $K^{(\lambda)}(z)$. One can easily show that for $v = 0, 1, 2,$

$$K_v^{(\lambda)}(z) = \sum_{i=1}^{j} c_i^{(\lambda)} \rho_i^{-v} P_i(z), \qquad c_1^{(\lambda)} = c_i^{(0)} \sigma_i^{-\lambda}. \tag{2.6}$$

We define $\bar{K}^{(\lambda)}(z)$ to be $K^{(\lambda)}(z)$ divided by its leading coefficient. Let

$$\sigma^{(\lambda)}(z) = \frac{\begin{vmatrix} K_0^{(\lambda)}(s_1) & K_0^{(\lambda)}(s_2) & z^2 \\ K_1^{(\lambda)}(s_1) & K_1^{(\lambda)}(s_2) & z \\ K_2^{(\lambda)}(s_1) & K_2^{(\lambda)}(s_2) & 1 \end{vmatrix}}{\begin{vmatrix} K_1^{(\lambda)}(s_1) & K_1^{(\lambda)}(s_2) \\ K_2^{(\lambda)}(s_1) & K_2^{(\lambda)}(s_2) \end{vmatrix}}. \tag{2.7}$$

We remind the reader that

$$\sigma(z) = (z - s_1)(z - s_2), \qquad \sigma_i = \sigma(\rho_i).$$

Our interest in K polynomials follows from the following two theorems.

THEOREM 2.1 (Convergence to a linear factor). *Let* $c_1^{(0)} \neq 0$ *and*

$$|\sigma_1| < |\sigma_i|, \qquad i = 2, \cdots, j. \tag{2.8}$$

Then ρ_1 *is real and for all finite* z,

$$\lim_{\lambda \to \infty} \frac{P(z)}{\bar{K}^{(\lambda)}(z)} = z - \rho_1. \tag{2.9}$$

Note. The zero labeled ρ_1 depends on the choice of $\sigma(z)$.

Proof. If (2.8) holds, then ρ_1 must be real since otherwise $(\bar{\rho}_1 - s_1)(\bar{\rho}_1 - s_2) = \sigma_i$, for some $i \geqq 2$, contradicting (2.8). From (2.4) we have

$$\bar{K}^{(\lambda)}(z) = \frac{P_1(z) + \sum_{i=2}^{j} \frac{c_i^{(0)}}{c_1^{(0)}} \left(\frac{\sigma_i}{\sigma_1}\right)^{-\lambda} P_i(z)}{1 + \sum_{i=2}^{j} \frac{c_i^{(0)}}{c_1^{(0)}} \left(\frac{\sigma_i}{\sigma_1}\right)^{-\lambda}},$$

which exists for λ sufficiently large, and the conclusion follows easily.

THEOREM 2.2 (Convergence to a quadratic factor). *Let* $c_1^{(0)} c_2^{(0)} \neq 0$ *and*

$$|\sigma_1| = |\sigma_2| < |\sigma_i|, \qquad i = 3, \cdots, j. \tag{2.10}$$

Then for all finite z,

$$\lim_{\lambda \to \infty} \sigma^{(\lambda)}(z) = (z - \rho_1)(z - \rho_2). \tag{2.11}$$

Note. The zeros labeled ρ_1 and ρ_2 depend on the choice of $\sigma(z)$.

Proof. Define

$$B_{\mu\nu} = \begin{vmatrix} K_\mu^{(\lambda)}(s_1) & K_\mu^{(\lambda)}(s_2) \\ K_\nu^{(\lambda)}(s_1) & K_\nu^{(\lambda)}(s_2) \end{vmatrix}.$$

We may rewrite (2.7) as

$$\sigma^{(\lambda)}(z) = \frac{1}{B_{12}}[B_{12}z^2 - B_{02}z + B_{01}]. \tag{2.12}$$

From (2.6),

$$B_{\mu\nu} = \begin{vmatrix} \sum_{i=1}^{j} c_i^{(\lambda)}\rho_i^{-\mu}P_i(s_1) & \sum_{i=1}^{j} c_i^{(\lambda)}\rho_i^{-\mu}P_i(s_2) \\ \sum_{i=1}^{j} c_i^{(\lambda)}\rho_i^{-\nu}P_i(s_1) & \sum_{i=1}^{j} c_i^{(\lambda)}\rho_i^{-\nu}P_i(s_2) \end{vmatrix}$$

$$= P(s_1)P(s_2)\left[\sum_{i,k} \frac{\rho_i^{-\mu}\rho_k^{-\nu}c_i^{(\lambda)}c_k^{(\lambda)}}{(s_1 - \rho_i)(s_2 - \rho_k)} - \sum_{i,k} \frac{\rho_i^{-\nu}\rho_k^{-\mu}c_i^{(\lambda)}c_k^{(\lambda)}}{(s_2 - \rho_i)(s_1 - \rho_k)}\right].$$

After some simplification this reduces to

$$B_{\mu\nu} = (s_2 - s_1)P(s_1)P(s_2) \sum_{k>i} (\rho_i^{-\mu}\rho_k^{-\nu} - \rho_i^{-\nu}\rho_k^{-\mu}) \frac{c_i^{(\lambda)}c_k^{(\lambda)}}{\sigma_i\sigma_k}(\rho_i - \rho_k). \tag{2.13}$$

Using (2.13) in (2.12) we have

$$\sigma^{(\lambda)}(z) = \frac{\sum_{k>i} \dfrac{c_i^{(\lambda)}c_k^{(\lambda)}}{\sigma_i\sigma_k}\left(\dfrac{\rho_i - \rho_k}{\rho_i\rho_k}\right)^2 (z - \rho_i)(z - \rho_k)}{\sum_{k>i} \dfrac{c_i^{(\lambda)}c_k^{(\lambda)}}{\sigma_i\sigma_k}\left(\dfrac{\rho_i - \rho_k}{\rho_i\rho_k}\right)^2}.$$

We define

$$e_{ik} = \left(\frac{\rho_i - \rho_k}{\rho_1 - \rho_2}\right)\left(\frac{\rho_1\rho_2}{\rho_i\rho_k}\right). \tag{2.14}$$

Then from (2.6),

$$\sigma^{(\lambda)}(z) = \frac{(z - \rho_1)(z - \rho_2) + \sum_{\substack{k>i \\ k\geq 3}} e_{ik}^2 \dfrac{c_i^{(0)}c_k^{(0)}}{c_1^{(0)}c_2^{(0)}}\left(\dfrac{\sigma_1\sigma_2}{\sigma_i\sigma_k}\right)^{\lambda+1} (z - \rho_i)(z - \rho_k)}{1 + \sum_{\substack{k>i \\ k\geq 3}} e_{ik}^2 \dfrac{c_i^{(0)}c_k^{(0)}}{c_1^{(0)}c_2^{(0)}}\left(\dfrac{\sigma_1\sigma_2}{\sigma_i\sigma_k}\right)^{\lambda+1}}, \tag{2.15}$$

which exists for λ sufficiently large and (2.11) follows easily.

These theorems show that the K polynomials may be used to find either a linear or quadratic factor of P. If (2.8) holds, the K polynomials have properties similar to the H polynomials and we use the variable-shift iteration described in Jenkins and Traub [3, § 4]. In contrast to the case studied in [3], we are now assured of iterating on the real line. See § 6 for details.

If (2.10) holds, the rate of convergence of the quadratic polynomial $\sigma^{(\lambda)}(z)$ to the quadratic factor $(z-\rho_1)(z-\rho_2)$ is linear, depending on $\max_{i\geq 3}|\sigma_1/\sigma_i|$. Once $\sigma^{(\lambda)}(z)$ has begun to converge, we would expect to hasten convergence by replacing $\sigma(z)$ in (2.2) by the best available approximation to $(z-\rho_1)(z-\rho_2)$, namely, $\sigma^{(\lambda)}(z)$. This leads to the idea of variable-shift K polynomials.

3. A variable-shift iteration for a quadratic factor. Let $K^{(0)}(z)$ be a real polynomial of degree at most $n-1$. Let $\sigma^{(0)}(z)$ be a real quadratic polynomial with zeros $s_1^{(0)}$ and $s_2^{(0)}$ such that $s_2^{(0)} \neq s_1^{(0)}$ and $P(s_1^{(0)})P(s_2^{(0)}) \neq 0$. For $\lambda = 0, 1, \cdots$, define the sequence

$$K^{(\lambda+1)}(z) = \frac{1}{\sigma^{(\lambda)}(z)}\left[K^{(\lambda)}(z) + \frac{\begin{vmatrix} P(s_1^{(\lambda)}) & P(s_2^{(\lambda)}) \\ K^{(\lambda)}(s_1^{(\lambda)}) & K^{(\lambda)}(s_2^{(\lambda)}) \end{vmatrix} z + \begin{vmatrix} K^{(\lambda)}(s_1^{(\lambda)}) & K^{(\lambda)}(s_2^{(\lambda)}) \\ s_1^{(\lambda)}P(s_1^{(\lambda)}) & s_2^{(\lambda)}P(s_2^{(\lambda)}) \end{vmatrix}}{\begin{vmatrix} s_1^{(\lambda)}P(s_1^{(\lambda)}) & s_2^{(\lambda)}P(s_2^{(\lambda)}) \\ P(s_1^{(\lambda)}) & P(s_2^{(\lambda)}) \end{vmatrix}} P(z)\right], \tag{3.1}$$

$$\sigma^{(\lambda+1)}(z) = \frac{\begin{vmatrix} K_0^{(\lambda+1)}(s_1^{(\lambda)}) & K_0^{(\lambda+1)}(s_2^{(\lambda)}) & z^2 \\ K_1^{(\lambda+1)}(s_1^{(\lambda)}) & K_1^{(\lambda+1)}(s_2^{(\lambda)}) & z \\ K_2^{(\lambda+1)}(s_1^{(\lambda)}) & K_2^{(\lambda+1)}(s_2^{(\lambda)}) & 1 \end{vmatrix}}{\begin{vmatrix} K_1^{(\lambda+1)}(s_1^{(\lambda)}) & K_1^{(\lambda+1)}(s_2^{(\lambda)}) \\ K_2^{(\lambda+1)}(s_1^{(\lambda)}) & K_2^{(\lambda+1)}(s_2^{(\lambda)}) \end{vmatrix}}, \tag{3.2}$$

where $s_1^{(\lambda)}$ and $s_2^{(\lambda)}$ are the zeros of $\sigma^{(\lambda)}(z)$, $K_0^{(\lambda+1)}(z) = K^{(\lambda+1)}(z)$,

$$K_{v+1}^{(\lambda+1)}(z) = \frac{1}{z}\left[K_v^{(\lambda+1)}(z) - \frac{K_v^{(\lambda+1)}(0)}{P(0)}P(z)\right], \qquad v = 0, 1. \tag{3.3}$$

If $P(s_1^{(\lambda)}) = 0$ or $P(s_2^{(\lambda)}) = 0$, terminate the calculation. The $K^{(\lambda)}(z)$ are polynomials of degree at most $n-1$. There should be no confusion from using the same symbol for the sequences generated by (2.2) and (3.1). The calculation of the coefficients of $K_v^{(\lambda+1)}(z)$, $v = 1, 2$, is not done explicitly as substitution of (3.3) into (3.2) yields a form involving only $K^{(\lambda+1)}(z)$, $P(z)$ and $\sigma^{(\lambda)}(z)$. The following lemma is easily verified.

LEMMA 3.1. *Let* $c_1^{(0)}c_2^{(0)} \neq 0$ *and assume*

$$K^{(0)}(z) = \sum_{i=1}^{j} c_i^{(0)}P_i(z).$$

Then for all λ,

$$K_v(z) = \sum_{i=1}^{j} \rho_i^{-v}c_i^{(\lambda)}P_i(z), \qquad v = 0, 1, 2,$$

$$c_i^{(\lambda)} = c_i^{(0)}\prod_{t=0}^{\lambda-1}[\sigma_i^{(t)}]^{-1}, \qquad \sigma_i^{(t)} = \sigma^{(t)}(\rho_i).$$

4. Convergence of the variable-shift iteration for a quadratic factor. We investigate the convergence of the variable-shift iteration defined by (3.1) and (3.2).

We first prove a useful lemma. Define

$$R = \tfrac{1}{2}|\rho_1 - \rho_2|, \tag{4.1}$$

$$R_1 = \min_{\substack{k \geqq 3 \\ i=1,2}} |\rho_i - \rho_k|, \tag{4.2}$$

$$R_2 = \min_{k \geqq 3} |\rho_k - \rho_1|\,|\rho_k - \rho_2|, \tag{4.3}$$

$$r_{ik}^{(\lambda)} = \frac{\sigma_1^{(\lambda)}\sigma_2^{(\lambda)}}{\sigma_i^{(\lambda)}\sigma_k^{(\lambda)}}, \qquad k > i, \quad k \geqq 3, \quad \sigma_i^{(\lambda)} = \sigma^{(\lambda)}(\rho_i),$$

$$d_{ik}^{(\lambda)} = \frac{c_i^{(\lambda)}c_k^{(\lambda)}}{c_1^{(\lambda)}c_2^{(\lambda)}}e_{ik}^2, \qquad e_{ik} = \left(\frac{\rho_i - \rho_k}{\rho_1 - \rho_2}\right)\left(\frac{\rho_1\rho_2}{\rho_i\rho_k}\right), \qquad k > i, \quad k \geqq 3.$$

LEMMA 4.1. *Let*

$$N_1 = \tfrac{1}{4}\min(R^2, R_2). \tag{4.4}$$

If

$$|\sigma_1^{(\lambda)}| < N_1, \qquad |\sigma_2^{(\lambda)}| < N_1,$$

then

$$|r_{ik}^{(\lambda)}| < 1, \qquad k > i, \quad k \geqq 3, \tag{4.5}$$

and

$$\left|\sigma_2^{(\lambda)}\frac{(\rho_1 - \rho_i)(\rho_1 - \rho_k)}{\sigma_i^{(\lambda)}\sigma_k^{(\lambda)}}\right| < 20\left(\frac{R}{R_1}\right)^2, \qquad k > i \geqq 2. \tag{4.6}$$

Proof. Let $s_1^{(\lambda)}$ be the zero of $\sigma^{(\lambda)}(z)$ which is closest to ρ_1 (or either zero if they are equidistant from ρ_1). Then from (4.4) we have

$$|\rho_1 - s_1^{(\lambda)}|^2 \leqq |\rho_1 - s_1^{(\lambda)}|\,|\rho_1 - s_2^{(\lambda)}| = |\sigma_1^{(\lambda)}| < R^2/4$$

which implies

$$|\rho_1 - s_1^{(\lambda)}| < R/2. \tag{4.7}$$

By symmetry,

$$|\rho_2 - s_2^{(\lambda)}| < R/2. \tag{4.8}$$

We now show that

$$|\rho_1 - s_1^{(\lambda)}| < R_1/2. \tag{4.9}$$

Assume that for some $i \geqq 3$,

$$|\rho_1 - s_1^{(\lambda)}| \geqq \tfrac{1}{2}|\rho_1 - \rho_i|. \tag{4.10}$$

Using (4.7) we have

$$|\rho_i - \rho_2| \leqq |\rho_i - \rho_1| + 2R \leqq 2|\rho_1 - s_1^{(\lambda)}| + 2R < 3R. \tag{4.11}$$

Using (4.7), (4.8), (4.10) and (4.11) we have

$$\begin{aligned} |\sigma_1^{(\lambda)}| = |\rho_1 - s_1^{(\lambda)}|\,|\rho_1 - s_2^{(\lambda)}| &\geqq \tfrac{1}{2}|\rho_1 - \rho_i|(2R - |\rho_2 - s_2^{(\lambda)}|) \\ &\geqq \tfrac{1}{2}|\rho_1 - \rho_i|\tfrac{3}{2}R \\ &\geqq \tfrac{1}{4}|\rho_1 - \rho_i|\,|\rho_2 - \rho_i| \geqq \tfrac{1}{4}R_2. \end{aligned}$$

This contradicts the assumption $|\sigma_1| < N_1$. Hence (4.9) holds. By a symmetrical argument,

$$|\rho_2 - s_2^{(\lambda)}| < R_1/2. \tag{4.12}$$

From (4.4), (4.9) and (4.12) we have

$$\begin{aligned} \left|\frac{\sigma_1^{(\lambda)}}{\sigma_i^{(\lambda)}}\right| &\leqq \frac{|\rho_i - \rho_1|\,|\rho_i - \rho_2|/4}{(|\rho_i - \rho_1| - |\rho_1 - s_1^{(\lambda)}|)(|\rho_i - \rho_2| - |\rho_2 - s_2^{(\lambda)}|)} \\ &\leqq \frac{1/4}{(1 - |\rho_1 - s_1^{(\lambda)}|/|\rho_i - \rho_1|)(1 - |\rho_2 - s_2^{(\lambda)}|/|\rho_i - \rho_2|)} < 1, \qquad i \geqq 3. \end{aligned} \tag{4.13}$$

By symmetry,

$$\left|\frac{\sigma_2^{(\lambda)}}{\sigma_i^{(\lambda)}}\right| < 1, \qquad i \geqq 3,$$

and we have proved (4.5).

We can write

$$\left|\frac{\sigma_2^{(\lambda)}(\rho_1 - \rho_i)(\rho_1 - \rho_k)}{\sigma_i^{(\lambda)}\sigma_k^{(\lambda)}}\right| = \left(\frac{|\rho_1 - \rho_i|\,|\rho_1 - \rho_k|}{|\rho_i - s_1^{(\lambda)}|\,|\rho_k - s_1^{(\lambda)}|}\right)\left(\frac{|\rho_2 - s_1^{(\lambda)}|\,|\rho_2 - s_2^{(\lambda)}|}{|\rho_i - s_2^{(\lambda)}|\,|\rho_k - s_2^{(\lambda)}|}\right).$$

The first factor on the right is less than 4 by a proof essentially the same as the proof of (4.13). Using (4.7), (4.8) and (4.12) we have

$$\begin{aligned} \frac{|\rho_2 - s_1^{(\lambda)}|\,|\rho_2 - s_2^{(\lambda)}|}{|\rho_i - s_2^{(\lambda)}|\,|\rho_k - s_2^{(\lambda)}|} &\leqq \frac{(2R + |\rho_1 - s_1^{(\lambda)}|)|\rho_2 - s_2^{(\lambda)}|}{(|\rho_i - \rho_2| - |\rho_2 - s_2^{(\lambda)}|)(|\rho_k - \rho_2| - |\rho_2 - s_2^{(\lambda)}|)} \\ &\leqq \frac{(5R/2)(R/2)}{R_1^2(1 - |\rho_2 - s_2^{(\lambda)}|/|\rho_i - \rho_2|)(1 - |\rho_2 - s_2^{(\lambda)}|/|\rho_k - \rho_2|)} \\ &\leqq 5\left(\frac{R}{R_1}\right)^2 \end{aligned}$$

and the proof of the lemma is complete.

We now give sufficient conditions for convergence of the variable-shift iteration for a quadratic factor.

THEOREM 4.1. *Let*

$$N_2 = \left(1 + 20\left(\frac{R}{R_1}\right)^2\right)^{-1}.$$

Assume

$$\text{(i)}\ |\sigma_1^{(0)}| < N_1, \qquad |\sigma_2^{(0)}| < N_1,$$

$$\text{(ii)}\ c_1^{(0)} c_2^{(0)} \neq 0,$$

$$\text{(iii)}\ \sum_{\substack{k>i \\ k\geq 3}} |d_{ik}^{(0)}| < N_2.$$

Then

$$\sigma^{(\lambda)}(z) \to (z-\rho_1)(z-\rho_2).$$

Proof. We defer, for the moment, the question as to whether $\sigma^{(\lambda+1)}(z)$ defined by (3.2) exists for all λ. As in the proof of Theorem 2.2, we can write

$$\sigma^{(\lambda+1)}(z) = \frac{(z-\rho_1)(z-\rho_2) + \sum_{\substack{k>i \\ k\geq 3}} e_{ik}^2 \dfrac{c_i^{(\lambda+1)} c_k^{(\lambda+1)}}{c_1^{(\lambda+1)} c_2^{(\lambda+1)}} \dfrac{\sigma_1^{(\lambda)} \sigma_2^{(\lambda)}}{\sigma_i^{(\lambda)} \sigma_k^{(\lambda)}} (z-\rho_i)(z-\rho_k)}{1 + \sum_{\substack{k>i \\ k\geq 3}} e_{ik}^2 \dfrac{c_i^{(\lambda+1)} c_k^{(\lambda+1)}}{c_1^{(\lambda+1)} c_2^{(\lambda+1)}} \dfrac{\sigma_1^{(\lambda)} \sigma_2^{(\lambda)}}{\sigma_i^{(\lambda)} \sigma_k^{(\lambda)}}}. \tag{4.14}$$

We have defined

$$r_{ik}^{(\lambda)} = \frac{\sigma_1^{(\lambda)} \sigma_2^{(\lambda)}}{\sigma_i^{(\lambda)} \sigma_k^{(\lambda)}}, \qquad d_{ik}^{(\lambda)} = \frac{c_i^{(\lambda)} c_k^{(\lambda)}}{c_1^{(\lambda)} c_2^{(\lambda)}} e_{ik}^2. \tag{4.15}$$

Note that

$$d_{ik}^{(\lambda+1)} = r_{ik}^{(\lambda)} d_{ik}^{(\lambda)}. \tag{4.16}$$

Setting z to p_1 in (4.14) and using (4.15) and (4.16) we have

$$\frac{\sigma_1^{(\lambda+1)}}{\sigma_1^{(\lambda)}} = \frac{\sum_{k>i\geq 2} r_{ik}^{(\lambda)} d_{ik}^{(\lambda)} \dfrac{(\rho_1-\rho_i)(\rho_1-\rho_k)\sigma_2^{(\lambda)}}{\sigma_i^{(\lambda)} \sigma_k^{(\lambda)}}}{1 + \sum_{\substack{k>i \\ k\geq 3}} [r_{ik}^{(\lambda)}]^2 d_{ik}^{(\lambda)}}. \tag{4.17}$$

By hypothesis (i), the previous lemma holds when λ is 0 and from (4.5), (4.8) and (4.17) we have

$$\left|\frac{\sigma_1^{(1)}}{\sigma_1^{(0)}}\right| \leqq \frac{20(R/R_1)^2 \sum_{k>i\geq 2} |d_{ik}^{(0)}|}{1 - \sum_{\substack{k>i \\ k\geq 3}} |d_{ik}^{(0)}|} = \tau_0. \tag{4.18}$$

Using (iii) in (4.18) we have

$$\left|\frac{\sigma_1^{(1)}}{\sigma_1^{(0)}}\right| \leqq \tau_0 < 1.$$

By a symmetrical argument we have

$$\left|\frac{\sigma_2^{(1)}}{\sigma_2^{(0)}}\right| \leqq \tau_0 < 1.$$

Define

$$T_i^{(\lambda)} = \left|\frac{\sigma_i^{(\lambda+1)}}{\sigma_i^{(\lambda)}}\right|, \qquad i = 1,2, \quad \lambda = 0,1,\cdots.$$

We prove $T_1^{(\lambda)} \leqq \tau_0 < 1$ and $T_2^{(\lambda)} \leqq \tau_0 < 1$ for all λ. The proof is by induction. Assume

$$T_1^{(0)}, T_1^{(1)}, \cdots, T_1^{(\lambda-1)}, T_2^{(0)}, T_2^{(1)}, \cdots, T_2^{(\lambda-1)} \leqq \tau_0 < 1.$$

Then

$$|\sigma_i^{(\lambda)}| < |\sigma_i^{(\lambda-1)}| < \cdots < |\sigma_i^{(0)}| < N_1, \qquad i = 1,2,$$

and we can apply the previous lemma repeatedly to obtain

$$\sum_{\substack{k>i \\ k\geqq 3}} |d_{ik}^{(\lambda)}| = \sum_{\substack{k>i \\ k\geqq 3}} \prod_{t=0}^{\lambda-1} |r_{ik}^{(t)}|\,|d_{ik}^{(0)}| < \sum_{\substack{k>i \\ k\geqq 3}} |d_{ik}^{(0)}| < N_2. \tag{4.19}$$

Using (4.5), (4.6), (4.19) in (4.17) we obtain

$$T_1^{(\lambda)} = \left|\frac{\sigma_1^{(\lambda+1)}}{\sigma_1^{(\lambda)}}\right| < \tau_0.$$

From an expression similar to (4.17) for $\sigma_2^{(\lambda+1)}/\sigma_2^{(\lambda)}$ we may show

$$T_2^{(\lambda)} = \left|\frac{\sigma_2^{(\lambda+1)}}{\sigma_2^{(\lambda)}}\right| < \tau_0.$$

Hence we have shown that as $\lambda \to \infty$, $\sigma_1^{(\lambda)} \to 0$ and $\sigma_2^{(\lambda)} \to 0$, which is sufficient to imply that $\sigma^{(\lambda)}(z) \to (z-\rho_1)(z-\rho_2)$.

To prove the iteration is defined for all λ, we note that from (4.19),

$$\sum |d_{ik}^{(\lambda+1)}| < N_2 < 1,$$

which with (4.5) and (4.14) implies that $\sigma^{(\lambda+1)}(z)$ is defined for all λ, and the proof of the theorem is complete.

The theorem can be paraphrased as follows. If $K^{(0)}(z)$ is close to being a linear combination of $P(z)/(z-\rho_1)$ and $P(z)/(z-\rho_2)$ and $\sigma^{(0)}(z)$ is a close approximation to $(z-\rho_1)(z-\rho_2)$, then $\sigma^{(\lambda)}(z) \to (z-\rho_1)(z-\rho_2)$.

5. Rate of convergence. We investigate the rate of convergence of the variable-shift iteration for a quadratic factor. Let

$$C_i(\lambda) = \frac{|\sigma_i^{(\lambda+1)}|}{|\sigma_i^{(\lambda)}|^2}, \qquad i = 1,2.$$

In Theorem 4.1, we proved the existence of a number τ_0 such that for $\lambda \geqq 0$,

$$\frac{|\sigma_i^{(\lambda+1)}|}{|\sigma_i^{(\lambda)}|} = T_i^{(\lambda)} \leqq \tau_0 < 1, \qquad i = 1, 2.$$

The rate of convergence is governed by the following theorem.

THEOREM 5.1. *Let the hypotheses of Theorem* 4.1 *hold. Then*

$$C_i(\lambda) \leqq \frac{4}{R_1^2}\tau_0^{\lambda(\lambda-1)/2}, \qquad i = 1, 2.$$

Proof. From (4.17),

$$\frac{\sigma_1^{(\lambda+1)}}{[\sigma_1^{(\lambda)}]^2} = \frac{\sum_{k>i\geqq 2} \dfrac{\sigma_2^{(\lambda)}}{\sigma_i^{(\lambda)}\sigma_k^{(\lambda)}} \dfrac{\sigma_2^{(\lambda)}(\rho_1-\rho_i)(\rho_1-\rho_k)}{\sigma_i^{(\lambda)}\sigma_k^{(\lambda)}} d_{ik}^{(\lambda)}}{1 + \sum_{\substack{k>i \\ k\geqq 3}} [r_{ik}^{(\lambda)}]^2 d_{ik}^{(\lambda)}}. \tag{5.1}$$

One may verify that for all λ,

$$\left|\frac{\sigma_2^{(\lambda)}}{\sigma_i^{(\lambda)}\sigma_k^{(\lambda)}}\right| \leqq \frac{1}{|\sigma_k^{(\lambda)}|} \leqq \frac{4}{R_1^2}, \qquad k > i \geqq 2,$$

$$|r_{ik}^{(\lambda)}| \leqq \tau_0^{\lambda}, \qquad k > i, \quad k \geqq 3,$$

and

$$\sum_{\substack{k>i \\ k\geqq 3}} |d_{ik}^{(\lambda)}| \leqq \frac{\tau_0^{\lambda(\lambda-1)/2}}{1 + 20(R/R_1)^2}.$$

Substituting these bounds into (5.1) establishes the theorem.

Thus the variable-shift iteration for a quadratic factor is second order with a $C_i(\lambda)$ which approaches zero.

6. The three-stage algorithm. We motivate the three-stage algorithm for real polynomials described below. In Stage 1 we calculate a sequence of no-shift polynomials. The purpose of this is to make the smaller zeros stand out. We terminate Stage 1 after a small number of steps and enter Stage 2 where we calculate a sequence of fixed-shift K polynomials. In Stage 2 we use a quadratic factor $\sigma(z) = (z - s_1)(z - s_2)$, where s_1 is a complex number with modulus less than or equal to $\min_i |\rho_i|$ and random amplitude and where $s_2 = \bar{s}_1$. We let ρ_1 designate a zero which satisfies

$$|\sigma_1| = \min |\sigma_i|, \qquad \sigma_i = (\rho_i - s_1)(\rho_i - s_2).$$

We assume that either

$$|\sigma_1| < |\sigma_i|, \qquad i = 2, \cdots, j, \tag{6.1}$$

or

$$|\sigma_1| = |\sigma_2| < |\sigma_i|, \qquad i = 3, \cdots, j, \tag{6.2}$$

holds.

If (6.1) holds, then by Theorem 2.1, the sequence

$$t_\lambda = s_1 - \frac{P(s_1)}{\bar{K}^{(\lambda)}(s_1)}$$

converges to ρ_1. *We remind the reader that ρ_1 must be real and that the zero labeled ρ_1 depends on the choice of $\sigma(z)$, that is, on the choice of s_1 and s_2.*

If (6.2) holds, then by Theorem 2.2, the sequence of quadratics $\sigma^{(\lambda)}(z)$ defined by (2.7) converges to $(z - \rho_1)(z - \rho_2)$. *The zeros ρ_1 and ρ_2 may be either real or complex conjugate and the zeros labeled ρ_1 and ρ_2 depend on the choice of s_1 and s_2.*

As soon as either $\{t_\lambda\}$ or $\{\sigma^{(\lambda)}(z)\}$ passes the convergence test we are ready to enter Stage 3. In Stage 3 we use one of two variable-shift iterations. If $\{t_\lambda\}$ passes the test first, then we use a real arithmetic version of the variable-shift iteration for a linear factor defined in Jenkins and Traub [3, § 3]. If $\{\sigma^{(\lambda)}(z)\}$ is the first to pass the test, we use the variable-shift iteration for a quadratic factor defined in § 3.

We now state the three-stage algorithm. The algorithm is used to calculate a zero or a pair of zeros of P. The polynomial is deflated after each zero or pair of zeros is found and then the algorithm is applied to the deflated polynomial. Hence P represents either the original polynomial or a polynomial obtained by deflation.

The above algorithm is incomplete in the sense that we do not discuss how to terminate the three stages or how to decide whether (6.1) or (6.2) holds. A discussion of these decisions can be found in § 9.

Stage 1 (No-shift process).

$$K^{(0)}(z) = P'(z),$$

$$K^{(\lambda+1)}(z) = \frac{1}{z}\left[K^{(\lambda)}(z) - \frac{K^{(\lambda)}(0)}{P(0)}P(z)\right], \qquad \lambda = 0, 1, \cdots, M-1.$$

Stage 2 (Fixed-shift process). Take β to be a positive number such that $\beta \leqq \min|\rho_i|$ and let $\sigma(z)$ be a real quadratic polynomial whose zeros s_1, s_2 satisfy $|s_1| = |s_2| = \beta$, $s_1 \neq s_2$ and $P(s_1)P(s_2) \neq 0$, and such that either (6.1) or (6.2) holds. Let

$$K^{(\lambda+1)}(z) = \frac{1}{\sigma(z)}\left[K^{(\lambda)}(z) + \frac{\begin{vmatrix} P(s_1) & P(s_2) \\ K^{(\lambda)}(s_1) & K^{(\lambda)}(s_2) \end{vmatrix} z + \begin{vmatrix} K^{(\lambda)}(s_1) & K^{(\lambda)}(s_2) \\ s_1P(s_1) & s_2P(s_2) \end{vmatrix}}{\begin{vmatrix} s_1P(s_1) & s_2P(s_2) \\ P(s_1) & P(s_2) \end{vmatrix}} P(z)\right], \tag{6.3}$$

$$\lambda = M, M+1, \cdots, L-1.$$

Stage 3 (Variable-shift process).

Iteration for a linear factor. If (6.1) holds, then take

$$s^{(L)} = \operatorname{Re}\left(s_1 - \frac{P(s_1)}{\bar{K}^{(L)}(s_1)}\right)$$

and let

$$K^{(\lambda+1)}(z) = \frac{1}{z - s^{(\lambda)}}\left[K^{(\lambda)}(z) - \frac{K^{(\lambda)}(s^{(\lambda)})}{P(s^{(\lambda)})}P(z)\right],$$

$$s^{(\lambda+1)} = s^{(\lambda)} - \frac{P(s^{(\lambda)})}{\overline{K}^{(\lambda+1)}(s^{(\lambda)})}, \qquad \lambda = L, L+1, \cdots. \tag{6.4}$$

Iteration for a quadratic factor. If (6.2) holds, then take

$$\sigma^{(L)}(z) = \frac{\begin{vmatrix} K_0^{(L)}(s_1) & K_0^{(L)}(s_2) & z^2 \\ K_1^{(L)}(s_1) & K_1^{(L)}(s_2) & z \\ K_2^{(L)}(s_1) & K_2^{(L)}(s_2) & 1 \end{vmatrix}}{\begin{vmatrix} K_1^{(L)}(s_1) & K_1^{(L)}(s_2) \\ K_2^{(L)}(s_1) & K_2^{(L)}(s_2) \end{vmatrix}}, \tag{6.5}$$

and for $\lambda = L, L+1, \cdots$, let

(6.6)

$$K^{(\lambda+1)}(z) = \frac{1}{\sigma^{(\lambda)}(z)}\left[K^{(\lambda)}(z) + \frac{\begin{vmatrix} P(s_1^{(\lambda)}) & P(s_2^{(\lambda)}) \\ K^{(\lambda)}(s_1^{(\lambda)}) & K^{(\lambda)}(s_2^{(\lambda)}) \end{vmatrix} z + \begin{vmatrix} K^{(\lambda)}(s_1^{(\lambda)}) & K^{(\lambda)}(s_2^{(\lambda)}) \\ s_1^{(\lambda)}P(s_1^{(\lambda)}) & s_2^{(\lambda)}P(s_2^{(\lambda)}) \end{vmatrix}}{\begin{vmatrix} s_1^{(\lambda)}P(s_1^{(\lambda)}) & s_2^{(\lambda)}P(s_2^{(\lambda)}) \\ P(s_1^{(\lambda)}) & P(s_2^{(\lambda)}) \end{vmatrix}} P(z)\right],$$

$$\sigma^{(\lambda+1)}(z) = \frac{\begin{vmatrix} K_0^{(\lambda+1)}(s_1^{(\lambda)}) & K_0^{(\lambda+1)}(s_2^{(\lambda)}) & z^2 \\ K_1^{(\lambda+1)}(s_1^{(\lambda)}) & K_1^{(\lambda+1)}(s_2^{(\lambda)}) & z \\ K_2^{(\lambda+1)}(s_1^{(\lambda)}) & K_2^{(\lambda+1)}(s_2^{(\lambda)}) & 1 \end{vmatrix}}{\begin{vmatrix} K_1^{(\lambda+1)}(s_1^{(\lambda)}) & K_2^{(\lambda+1)}(s_2^{(\lambda)}) \\ K_2^{(\lambda+1)}(s_1^{(\lambda)}) & K_2^{(\lambda+1)}(s_2^{(\lambda)}) \end{vmatrix}}, \tag{6.7}$$

where $s_1^{(\lambda)}$ and $s_2^{(\lambda)}$ are the zeros of $\sigma^{(\lambda)}(z)$, $K_0^{(\lambda)}(z) = K^{(\lambda)}(z)$, and

$$K_{v+1}^{(\lambda)}(z) = \frac{1}{z}\left[K_v^{(\lambda)}(z) - \frac{K_v^{(\lambda)}(0)}{P(0)}P(z)\right], \qquad v = 0, 1.$$

7. Proof of global convergence of the three-stage algorithm. We investigate the convergence of the three-stage algorithm for real polynomials. The major result of this paper is given by the following theorem.

THEOREM 7.1. *Assume that $\sigma(z)$ is chosen so that either* (6.1) *or* (6.2) *holds. Let L be sufficiently large and fixed. Then if* (6.1) *holds, $s^{(\lambda)}$ generated by* (6.4) *converges to ρ_1, while if* (6.2) *holds, the sequence $\sigma^{(\lambda)}(z)$ generated by* (6.6) *and* (6.7) *converges to $(z - \rho_1)(z - \rho_2)$.*

Proof. One may easily verify that

$$K^{(M)}(z) = \sum_{i=1}^{j} m_i \rho_i^{-M} P_i(z).$$

From (2.4),

$$K^{(L)}(z) = \sum_{i=1}^{j} c_i^{(L)} P_i(z), \qquad c_i^{(L)} = m_i \rho_i^{-M} \sigma_i^{-(L-M)}.$$

Proof for linear factor iteration. Assume (6.1) holds. We have $c_1^{(L)} \neq 0$. Let $D_L = \sum_{i=2}^{j} |c_i^{(L)}/c_1^{(L)}|$. Then

$$D_L = \sum_{i=2}^{j} \left| \frac{m_i}{m_1} \left(\frac{\rho_1}{\rho_i} \right)^M \left(\frac{\sigma_1}{\sigma_i} \right)^{L-M} \right|.$$

Fix M. Then by choosing L sufficiently large we can make $D_L < \frac{1}{3}$. By Theorem 2.1 we can choose L so large that

$$s^{(L)} = \operatorname{Re}\left(s_1 - \frac{P(s_1)}{\bar{K}^{(L)}(s_1)} \right)$$

satisfies

$$|s^{(L)} - \rho_1| < \tfrac{1}{2} \min_{i \geq 2} |\rho_1 - \rho_i|.$$

Choose L large enough so that both conditions hold.

Jenkins and Traub [3, § 5] have proved the following lemma.

LEMMA 7.1. *If*

(i) $$|s^{(L)} - \rho_1| < \tfrac{1}{2} \min_{i \geq 2} |\rho_1 - \rho_i|,$$

(ii) $$c_1^{(L)} \neq 0,$$

(iii) $$D_L = \sum_{i=2}^{j} \frac{|c_i^{(L)}|}{|c_1^{(L)}|} < \frac{1}{3},$$

then $s^{(\lambda)} \to \rho_1$.

An application of this lemma enables us to conclude that $s^{(\lambda)} \to \rho_1$.

Proof for quadratic factor iteration. Assume (6.2) holds. We have $c_1^{(L)} c_2^{(L)} \neq 0$. Let

$$E_L = \sum_{\substack{k>i \\ k \geq 3}} \left| \frac{c_i^{(L)} c_k^{(L)}}{c_1^{(L)} c_2^{(L)}} \right|.$$

Then

$$E_L = \sum_{\substack{k>i \\ k \geq 3}} \left| \frac{m_i}{m_1} \frac{m_k}{m_2} \left(\frac{\rho_i \rho_k}{\rho_1 \rho_2} \right)^M \left(\frac{\sigma_i \sigma_k}{\sigma_1 \sigma_2} \right)^{L-M} \right|.$$

Fix M. Then by choosing L sufficiently large we can make $E_L < N_2$. By Theorem 2.2, we can choose L so large that $\sigma^{(L)}(z)$ defined by (6.5) satisfies

$$|\sigma_1^{(L)}| < N_1, \qquad |\sigma_2^{(L)}| < N_1.$$

Choose L large enough that both conditions hold. Then by Theorem 4.1,

$$\sigma^{(\lambda)}(z) \to (z - \rho_1)(z - \rho_2).$$

This completes the proof.

8. Variable-shift iteration for a quadratic factor is quadratic generalized Rayleigh iteration. We now give a matrix formulation of the iteration defined by (6.6) and (6.7). We show that the variable-shift iteration for a quadratic factor is a quadratic form of shifted inverse powering and a quadratic analogue of generalized Rayleigh iteration.

Let

$$A = \begin{pmatrix} 0 & 0 & \cdots & 0 & -a_n \\ 1 & 0 & \cdots & 0 & -a_{n-1} \\ 0 & 1 & \cdots & 0 & -a_{n-2} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \cdot & \cdot & \cdots & 1 & -a_1 \end{pmatrix}$$

be the companion matrix of P. Let

$$K^{(\lambda)}(z) = \sum_{i=0}^{n-1} k_i^{(\lambda)} z^{n-1-i}$$

and

$$[\mathbf{k}^{(\lambda)}]^{\mathsf{T}} = (k_{n-1}^{(\lambda)}, \cdots, k_0^{(\lambda)}).$$

Let

$$P_i(z) = \frac{P(z)}{z - \rho_i} = \sum_{j=0}^{n-1} p_{ij} z^{n-1-j}.$$

Define

$$\mathbf{p}_i^{\mathsf{T}} = (p_{i\,n-1}, \cdots, p_{i\,0}), \qquad \mathbf{q}_i^{\mathsf{T}} = (1, \cdots, \rho_i^{n-1}).$$

One may easily verify that for the eigenvalue ρ_i, the right and left eigenvectors are $\mathbf{p}_i$ and $\mathbf{q}_i^{\mathsf{T}}$, respectively.

Using the form of the fixed-shift recurrence given by (2.3) one can verify that it is equivalent to

$$\mathbf{k}^{(\lambda+1)} = (A - s_1 I)^{-1}(A - s_2 I)^{-1}\mathbf{k}^{(\lambda)} = [\sigma(A)]^{-1}\mathbf{k}^{(\lambda)}. \tag{8.1}$$

The variable-shift recurrence is equivalent to

$$\mathbf{k}^{(\lambda+1)} = [\sigma^{(\lambda)}(A)]^{-1}\mathbf{k}^{(\lambda)}. \tag{8.2}$$

Equations (8.1) and (8.2) exhibit the processes as a quadratic form of inverse powering.

Let

$$\sigma^{(\lambda)}(z) = z^2 + u_\lambda z + v_\lambda,$$
$$[\mathbf{s}_i^{(\lambda)}]^{\mathsf{T}} = (1, s_i^{(\lambda)}, \cdots, [s_i^{(\lambda)}]^{n-1}), \qquad i = 1, 2.$$

We show that the definition of $\sigma^{(\lambda+1)}(z)$ given by (6.7) is equivalent to solving the pair of equations

$$\begin{aligned} [\mathbf{s}_1^{(\lambda)}][I + u_{\lambda+1}A^{-1} + v_{\lambda+1}A^{-2}]\mathbf{k}^{(\lambda+1)} &= 0, \\ [\mathbf{s}_2^{(\lambda)}][I + u_{\lambda+1}A^{-1} + v_{\lambda+1}A^{-2}]\mathbf{k}^{(\lambda+1)} &= 0 \end{aligned} \tag{8.3}$$

for $u_{\lambda+1}$ and $v_{\lambda+1}$ and forming $\sigma^{(\lambda+1)}(z)$.

It may be verified that the polynomial which corresponds to $A^{-1}\mathbf{k}^{(\lambda+1)}$ is

$$\frac{1}{z}\left[K^{(\lambda+1)}(z) - \frac{K^{(\lambda+1)}(0)}{P(0)}P(z)\right]$$

which we have called $K_1^{(\lambda+1)}(z)$. Thus

$$[\mathbf{s}_1^{(\lambda)}]A^{-\nu}\mathbf{k}^{(\lambda+1)} = K_\nu^{(\lambda+1)}(s_1^{(\lambda)}), \qquad \nu = 0, 1, 2,$$

and hence (8.3) is equivalent to

$$\begin{bmatrix} K_1^{(\lambda+1)}(s_1^{(\lambda)}) & K_2^{(\lambda+1)}(s_1^{(\lambda)}) \\ K_1^{(\lambda+1)}(s_2^{(\lambda)}) & K_2^{(\lambda+1)}(s_2^{(\lambda)}) \end{bmatrix}\begin{bmatrix} u_{\lambda+1} \\ v_{\lambda+1} \end{bmatrix} = -\begin{bmatrix} K_0^{(\lambda+1)}(s_1^{(\lambda)}) \\ K_0^{(\lambda+1)}(s_2^{(\lambda)}) \end{bmatrix}. \tag{8.4}$$

One may easily verify that (8.4) is equivalent to (6.7).

We summarize this result.

THEOREM 8.1. *The variable-shift iteration for a quadratic factor is equivalent to*

$$\mathbf{k}^{(\lambda+1)} = [\sigma^{(\lambda)}(A)]^{-1}\mathbf{k}^{(\lambda)},$$

and $\sigma^{(\lambda+1)}(z)$ *determined from the pair of equations*

$$[\mathbf{s}_1^{(\lambda)}]^{\mathsf{T}}(I + u_{\lambda+1}A^{-1} + v_{\lambda+1}A^{-2})\mathbf{k}^{(\lambda+1)} = 0,$$
$$[\mathbf{s}_2^{(\lambda)}]^{\mathsf{T}}(I + u_{\lambda+1}A^{-1} + v_{\lambda+1}A^{-2})\mathbf{k}^{(\lambda+1)} = 0.$$

Observe that $[\mathbf{s}_i^{(\lambda)}]^{\mathsf{T}} \to \mathbf{q}_i^{\mathsf{T}}$, $i = 1, 2$. Although $\mathbf{k}^{(\lambda)}$ does not converge, it becomes restricted to the space spanned by the right eigenvectors, $\mathbf{p}_1$ and $\mathbf{p}_2$. Hence we have a quadratic analogue of generalized Rayleigh iteration appropriate for non-Hermitian matrices. (Discussion of generalized Rayleigh iteration may be found in Wilkinson [9, p. 179], Ostrowski [5].)

However we are in a very favorable position as compared with the usual situation when inverse iteration and generalized Rayleigh iteration is applied to a non-Hermitian matrix. The reader is referred to Jenkins and Traub [3, § 7] for a discussion of this.

Observe that (8.3) is not a direct generalization of the matrix form of the variable-shift iteration

$$s_{\lambda+1} = \frac{[\mathbf{s}^{(\lambda)}]^{\mathsf{T}}A\mathbf{k}^{(\lambda+1)}}{[\mathbf{s}^{(\lambda)}]^{\mathsf{T}}\mathbf{k}^{(\lambda+1)}} \tag{8.5}$$

used by Jenkins and Traub [3] to approximate a smallest zero. Equation (8.3) can be rewritten as

$$[\mathbf{s}^{(\lambda)}]^{\mathsf{T}}(A - s_{\lambda+1}I)\mathbf{k}^{(\lambda+1)} = 0,$$

and the natural generalization to an iteration for a quadratic factor is

$$[\mathbf{s}_1^{(\lambda)}]^{\mathsf{T}}(A^2 + u_{\lambda+1}A + v_{\lambda+1}I)\mathbf{k}^{(\lambda+1)} = 0,$$
$$[\mathbf{s}_2^{(\lambda)}]^{\mathsf{T}}(A^2 + u_{\lambda+1}A + v_{\lambda+1}I)\mathbf{k}^{(\lambda+1)} = 0. \tag{8.6}$$

However (8.6) proves unsatisfactory computationally if the ratio of the smallest zero to the largest zero (in modulus) is less than ε, where ε is the relative error of

the arithmetic used for the computation. We illustrate this with an example. Let $P(z)$ be a cubic polynomial such that

$$|\rho_1| = |\rho_2| \ll |\rho_3|.$$

Suppose $K^{(\lambda+1)}(z) = P_1(z) + P_2(z) + \varepsilon_3 P_3(z)$ with $|\varepsilon_3| = \varepsilon$. This is as good a linear combination of P_1 and P_2 as we can expect in the presence of roundoff error. Let $|s_1^{(\lambda)} - \rho_1| = \varepsilon|\rho_1|$ and $|s_2^{(\lambda)} - \rho_2| = \varepsilon|\rho_2|$. Then treating $K^{(\lambda+1)}(z)$ as a vector, (8.6) is equivalent, after some manipulation, to

$$\begin{aligned} &\rho_1^2 + u_{\lambda+1}\rho_1 + v_{\lambda+1}(1 + O(\varepsilon^2)) + \varepsilon_3(\rho_3^2 + u_{\lambda+1}\rho_3 + v_{\lambda+1})O\left(\frac{\varepsilon}{\rho_3}\right) = 0, \\ &\rho_2^2 + u_{\lambda+1}\rho_2 + v_{\lambda+1}(1 + O(\varepsilon^2)) + \varepsilon_3(\rho_3^2 + u_{\lambda+1}\rho_3 + v_{\lambda+1})O\left(\frac{\varepsilon}{\rho_3}\right) = 0, \end{aligned} \tag{8.7}$$

where we use $O(t)$ to represent a quantity x, which satisfies $|x/t| < K_t$ for some constant K_t. Considering (8.7) as a system of equations in $u_{\lambda+1}, v_{\lambda+1}$, and neglecting small terms, we may solve $u_{\lambda+1}$ obtaining

$$u_{\lambda+1} = -\left[\rho_1 + \rho_2 + \varepsilon_3 \frac{\rho_3^2}{\rho_1 - \rho_2} O\left(\frac{\varepsilon}{\rho_3}\right)\right].$$

Thus if $|\rho_1/\rho_3| = k\varepsilon$ with $k \ll 1$, then $u_{\lambda+1}$ will not be as close to $-(\rho_1 + \rho_2)$ as u_λ, even though $|s_\lambda - \rho_1|$ is small and $K^{(\lambda+1)}(z)$ is close to being a linear combination of $P_1(z)$ and $P_2(z)$.

If we use (8.3) on this example, we find that

$$u_{\lambda+1} = -\left[\rho_1 + \rho_2 + \varepsilon_3 \frac{\rho_1^2}{\rho_1 - \rho_2} O\left(\frac{\varepsilon}{\rho_3}\right)\right],$$

which is satisfactory.

9. Computer implementation of the algorithm. The program implementing the algorithm, the results of extensive testing, and a program which clusters the zeros and provides a posteriori error bounds may be found in Jenkins [2]. Here we confine ourselves to a description of how the program makes its major decisions and how the $K^{(\lambda)}(z)$ and $\sigma^{(\lambda)}(z)$ may be efficiently calculated.

The termination of Stage 1, that is, the choice of M, is not crucial. The function of Stage 1 is to accentuate the smaller zeros. In the implementation, M is set at 5, a number arrived at by numerical experience.

The following four major decisions have to be made by the program:

(i) Selection of the initial quadratic polynomial $\sigma(z)$.

(ii) Is

$$|\sigma_1| < |\sigma_i|, \qquad i = 2, \cdots, j, \tag{9.1}$$

or

$$|\sigma_1| = |\sigma_2| < |\sigma_i|, \qquad i = 3, \cdots, j. \tag{9.2}$$

(iii) Termination of Stage 2, that is, the choice of L.

(iv) Termination of Stage 3.

We indicate how these four decisions are made.

(i) *Selection of* $\sigma(z)$. Choose a complex number s_1 so that $|s_1| = \beta, \beta \leqq \min|\rho_i|$, $i = 1, 2, \cdots, j$, and so that

$$|s_1 - \rho_1| < |s_1 - \rho_i|, \qquad i = 2, \cdots, j. \tag{9.3}$$

A lower bound on the moduli of the zeros due to Cauchy (Marden [4, Example 1, p. 98]) is given by the unique positive zero, β, of the polynomial

$$z^n + |a_1|z^{n-1} + \cdots + |a_{n-1}|z - |a_n|.$$

This number is easily calculated by Newton–Raphson iteration. The value of s_1 is then chosen by using random numbers from a uniform distribution to pick a point on the circle of radius β.

We then set $\sigma(z) = (z - s_1)(z - \bar{s}_1)$. It is highly likely that $\sigma(z)$ chosen in this manner will satisfy either (9.1) or (9.2). If neither condition is satisfied, the test for choosing L described below may not be passed, in which case a new value of s_1 will be chosen.

(ii) *Does* (9.1) *or* (9.2) *hold*? This decision is made jointly with decision (iii).

(iii) *Termination of Stage* 2. If (9.1) holds, then the sequence

$$t_\lambda = s_1 - \frac{P(s_1)}{\bar{K}^{(\lambda)}(s_1)}$$

converges to ρ_1. If (9.2) holds, then the sequence $\sigma^{(\lambda)}(z)$ defined by

$$\sigma^{(\lambda)}(z) = \frac{\begin{vmatrix} K_0^{(\lambda)}(s_1) & K_0^{(\lambda)}(s_2) & z^2 \\ K_1^{(\lambda)}(s_1) & K_1^{(\lambda)}(s_2) & z \\ K_2^{(\lambda)}(s_1) & K_2^{(\lambda)}(s_2) & 1 \end{vmatrix}}{\begin{vmatrix} K_1^{(\lambda)}(s_1) & K_1^{(\lambda)}(s_2) \\ K_2^{(\lambda)}(s_1) & K_2^{(\lambda)}(s_2) \end{vmatrix}} \tag{9.4}$$

converges to $(z - \rho_1)(z - \rho_2)$. (The zero labeled ρ_1 in this discussion depends on the choice of s_1.) We monitor both sequences and when one begins to converge we decide λ is large enough and that the corresponding condition holds. If neither test is passed by the time λ reaches a certain value, which is increased as additional shifts are tried (see Jenkins [2]), a new value of s_1 and hence a new $\sigma(z)$ is selected. The test for convergence of the sequence $\{t_\lambda\}$ is as follows. Experience has shown that it is efficient to terminate Stage 2 after only a very weak test for convergence has been passed. If

$$|t_{\lambda+1} - t_\lambda| \leqq \tfrac{1}{2}|t_\lambda|, \qquad |t_{\lambda+2} - t_{\lambda+1}| \leqq \tfrac{1}{2}|t_{\lambda+1}|,$$

then Stage 2 is terminated. The sequence $\{\sigma^{(\lambda)}(z)\}$ is monitored by applying the same test to $\{v_\lambda\}$, where $\sigma^{(\lambda)}(z) = z^2 + u_\lambda z + v_\lambda$.

(iv) *Termination of Stage* 3. As in the complex algorithm we terminate Stage 3 when the computed value of the polynomial is less than or equal to a bound on the roundoff error in evaluating it. For a complex conjugate pair of zeros we use the test derived by Adams [1]. For a single real zero or a pair of real zeros we use the original test due to Kahan and described by Adams [1].

We now describe how the process for computing the K polynomials in either Stage 2 or Stage 3 is actually carried out. We describe the Stage 2 process here;

the Stage 3 process differs only by having a variable $\sigma^{(\lambda)}(z)$ in place of $\sigma(z)$. We continue to use the notation

$$\sigma(z) = z^2 + uz + v \quad \text{and} \quad \sigma^{(\lambda)}(z) = z^2 + u_\lambda z + u_\lambda.$$

Rather than computing the K polynomials by (2.2), the "scaled recurrence"

$$K^{(0)}(z) = \frac{1}{n}P'(z),$$

(9.5)

$$K^{(\lambda+1)}(z) = \frac{1}{\sigma(z)}\left[\begin{vmatrix} s_1P(s_1) & s_2P(s_2) \\ P(s_1) & P(s_2) \end{vmatrix} K^{(\lambda)}(z) + \left(z + \frac{\begin{vmatrix} K^{(\lambda)}(s_1) & K^{(\lambda)}(s_2) \\ s_1P(s_1) & s_2P(s_2) \end{vmatrix}}{\begin{vmatrix} P(s_1) & P(s_2) \\ K^{(\lambda)}(s_1) & K^{(\lambda)}(s_2) \end{vmatrix}}\right)P(z)\right]$$

is used. This generates a sequence of monic polynomials and avoids the overflow and underflow problems which would occur if (2.2) were used. If $P(s_1)K^{(\lambda)}(s_2) = P(s_2)K^{(\lambda)}(s_1)$, then $K^{(\lambda+1)}(z)$ is not defined in (9.5). However we can use (2.2) in this case.

The computation of $K^{(\lambda+1)}(z)$ by (9.5) requires roughly $8n$ real multiplications and additions. This may be reduced to $6n$ by the following observation. Let

(9.6)
$$\begin{aligned} P(z) &= Q_P(z)\sigma(z) + b(z + u) + a, \\ K^{(\lambda)}(z) &= Q_K^{(\lambda)}(z)\sigma(z) + d(z + u) + c. \end{aligned}$$

Then

(9.7)
$$\begin{aligned} P(s_1) &= a - bs_2, & P(s_2) &= a - bs_1, \\ K^{(\lambda)}(s_1) &= c - ds_2, & K^{(\lambda)}(s_2) &= c - ds_1. \end{aligned}$$

Substituting these quantities in (9.5) and simplifying we obtain

$$K^{(\lambda+1)}(z) = \left(\frac{a^2 + uab + vb^2}{bc - ad}\right)Q_K^{(\lambda)}(z) + \left(z - \left(\frac{ac + uad + vbd}{bc - ad}\right)\right)Q_P(z) + b. \tag{9.8}$$

If we calculate a, b and c, d by the usual generalization of the Horner recurrence to quadratic factors, then $Q_P(z)$ and $Q_K^{(\lambda)}(z)$ are generated as a by-product. Notice that (9.8) can be carried out entirely in real arithmetic and s_1 and s_2 do not appear. In Stage 2, $\sigma(z)$ is fixed, a, b and $Q_P(z)$ are formed just once and hence, whereas $6n$ real multiplications and additions are required for the first step, only $4n$ are required thereafter.

$\sigma^{(\lambda)}(z)$ defined by (9.4) is calculated at each step of Stage 2 to monitor the convergence of $\{\sigma^{(\lambda)}(z)\}$. We do not actually form $K_1^{(\lambda+1)}(z)$, $K_2^{(\lambda+1)}(z)$ but use their definition to deduce formulas for u_λ and v_λ involving only u, v, a, b, c and d. The formulas and their derivation may be found in Jenkins [2]. The formulas involve only real quantities and have some factors in common with (9.8). Very little extra work is done in evaluating (9.4) at every step of Stage 2 as most of the work is in the calculation of a, b, c and d which must be done for the next step of the K polynomial recurrence.

The Stage 3 calculation involves the use of the K polynomial recurrence (6.6) to calculate $K^{(\lambda+1)}(z)$ and the calculation of the coefficients of $\sigma^{(\lambda+1)}(z)$ defined by

(6.7) at each step. These calculations are very similar to those in Stage 2 and can be carried out by procedures which are shared by both stages.

10. Numerical results. Extensive testing has been performed on a computer program which implements the algorithm described in this paper. The program was developed in ALGOL W, an extension of ALGOL 60, at Stanford University and run on an IBM 360/67. It was translated to ALGOL 60 for the IBM 360 to make it portable, and run on an IBM 360/91 at the IBM Thomas J. Watson Research Center.

The ALGOL 60 program may be found in Jenkins [2] where the results of extensive testing on the IBM 360/91 are reported. A second program which appears in [2] calculates a posteriori error bounds on the zeros.

We summarize some of the results from [2]. The program was tested on some well-known examples from the literature, very difficult polynomials, and randomly generated polynomials. The program always succeeded in finding the zeros one or two at a time, except when overflow interfered with the calculation.

A design objective of the program was that *all* zeros of a polynomial were to be calculated about as accurately as one might expect for the precision used. (The accuracy one might expect depends on the condition of the polynomial.) The test results indicated that the program was able to meet this objective with the exception of one particular kind of deflation instability which is discussed below.

We found that for polynomials of degree $\geqq 20$, the timing was fairly independent of the configuration of zeros. Thus for the ALGOL W examples run on the IBM 360/67 at Stanford we have, for polynomials of degree $\geqq 20$:

The real algorithm takes about $2n^2$ milliseconds to calculate all the zeros of an n-th degree real polynomial.

This may be contrasted with the results of using the complex algorithm (Jenkins and Traub [3]) on real polynomials which takes about $8n^2$ milliseconds. Observe that a FORTRAN program would be much faster.

Zeros are calculated in roughly increasing order of modulus; this avoids the instability which occurs when the polynomial is deflated with a large zero. Another type of deflation instability can occur. It is possible for a polynomial to be well-conditioned but to contain factors which are ill-conditioned. For example, the zeros of $P(z) = z^n - 1$ are well-conditioned. However the polynomial consisting of those nth roots of unity lying only on a half-circle is ill-conditioned if n is large. This has been pointed out by Wilkinson [10, p. 64].

Our program has no difficulty with $P(z) = z^n - 1$, as the random shift avoids finding most of the zeros on one half-circle first. However a polynomial with all its zeros lying on two half-circles of differing radii will suffer deterioration of condition during deflation as the program tends to remove most of the zeros from the smaller half-circle first. For any fixed precision we can, for a sufficiently high degree polynomial of this form, suffer deflation instability. P. Businger (private communication) has constructed polynomials of this form which produce deflation instability. For example, in double precision (14 hexadecimal digits) on the IBM 360, Businger's polynomial of degree 60 causes severe deflation instability.

To illustrate the algorithm we exhibit a low degree numerical example. This

example was run on a 360 using the ALGOL 60 programs appearing in Jenkins' dissertation [2].

The example has a complex conjugate pair of zeros, three equimodular zeros, two of which form a multiple pair and a pair of zeros which are nearly a multiple pair.

$$P(z) = z^7 - 6.01z^6 + 12.54z^5 - 8.545z^4 - 5.505z^3 + 12.545z^2 - 8.035z + 2.01$$
$$= (z - .5 + .5i)(z - .5 - .5i)(z - 1)^2(z + 1)(z - 2)(z - 2.01).$$

In calculating each of the zeros below, five no-shift steps were taken ($M = 5$). In Table 1 we give the quadratic factor used in Stage 2, the number of Stage 2 steps ($L - M$), the starting value for the stage three iteration (either $s^{(L)}$ or $\sigma^{(L)}(z)$ depending on which iteration is used) and the iterates used in Stage 3.

TABLE 1
A numerical example

Zeros (1) and (2)	$\sigma(z) = z^2 - .084038z + .035497$, $L - M = 3$, $\sigma^{(L)}(z) = z^2 - .98340z + .49994$
j	$\sigma^{(L+j)}$
1	$z^2 - .999992340057738z + .499996453164521$
2	$z^2 - 1.00000000000001z + .499999999999998$
Zero (3)	$\sigma(z) = z^2 + .056097z + .16360$, $L - M = 3$, $s^{(L)} = .20394$
j	$s^{(L+j)}$
1	1.82484162310780
2	.934209438070405
3	.999696240037581
4	1.00000000025743
Zero (4)	$\sigma(z) = z^2 - 1.0643z + .31594$, $L - M = 3$, $s^{(L)} = .99935$
j	$s^{(L+j)}$
1	1.00000000002948
2	.999999999742563
Zero (5)	$\sigma(z) = z^2 + 1.4910z + 1.0000$, $L - M = 3$, $s^{(L)} = -.999998$
j	$s^{(L+j)}$
1	$-.999999999999998$
Zeros (6) and (7)	solved directly from quadratic
	2.00000000000011
	2.00999999999989

Acknowledgment. We express our appreciation to P. A. Businger, Bell Telephone Laboratories, for performing tests and making suggestions which led to important improvements in the program.

REFERENCES

[1] D. A. ADAMS, *A stopping criterion for polynomial root finding*, Comm. ACM, 10 (1967), pp. 655–658.

[2] M. A. JENKINS, *Three-stage variable-shift iterations for the solution of polynomial equations with a posteriori error bounds for the zeros*, Dissertation, Stanford University, Stanford, Calif., 1969. Available as Rep. CS 138, Computer Science Department, Stanford University.

[3] M. A. JENKINS AND J. F. TRAUB, *A three-stage variable-shift iteration for polynomial zeros and its relation to generalized Rayleigh iteration*, Numer. Math, 14 (1970), pp. 252–261.

[4] M. MARDEN, *The Geometry of the Zeros of a Polynomial in a Complex Variable*, American Mathematical Society, Providence, R.I., 1949.

[5] A. M. OSTROWSKI, *On the convergence of the Rayleigh quotient iteration for the computation of the characteristic roots and vectors. IV: Generalized Rayleigh quotient for nonlinear elementary divisors*, Arch. Rational Mech. Anal., 3 (1959), pp. 341–347.

[6] J. F. TRAUB, *A class of globally convergent iteration functions for the solution of polynomial equations*, Math. Comp., 20 (1966), pp. 113–138.

[7] ———, *Proof of global convergence of an iterative method for calculating complex zeros of a polynomial*, Notices Amer. Math. Soc., 13 (1966), p. 117.

[8] ———, *The calculation of zeros of polynomials and analytic functions*, Mathematical Aspects of Computer Science, Proc. Symp. Applied Mathematics, vol. 19, American Mathematical Society, Providence, R.I., 1967, pp. 138–152. Also available as Rep. CS. 36, Computer Science Department, Stanford University, Stanford, Calif., 1966.

[9] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

[10] ———, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, N.J., 1963.