On Bairstow's Method
for the Solution of Polynomial Equations

By Kenneth W. Brodlie

Abstract. We show that Bairstow's method is just one member of a family of similar algorithms for determining a quadratic factor of a polynomial. We suggest a way of choosing an appropriate member of this family for a particular problem. Numerical results indicate that our strategy compares favorably with the classical Bairstow algorithm.

1. **Introduction.** Let $P(z) = a_n z^n + a_{n-1} z^{n-1} + \ldots + a_0$ be a polynomial with real coefficients. The complex roots of P occur as complex conjugate pairs, and so P has real quadratic factors. Suppose we wish to determine such a factor, say $(z^2 + p^* z + q^*)$, given an approximation $(z^2 + p^{(0)} z + q^{(0)})$, $p^{(0)}$, $q^{(0)}$ real.

Bairstow [3] suggested the following method which involves only real arithmetic. Dividing P by the real quadratic $(z^2 + pz + q)$, we can write

(1.1) $$P(z) = (z^2 + pz + q)(b_{n-2} z^{n-2} + \ldots + b_0) + uz + v$$

where $(uz + v)$ is the remainder. On equating coefficients we have

(1.2) $$b_i = a_{i+2} - pb_{i+1} - qb_{i+2} \quad (i = n-2, \ldots, 0), \qquad b_{n-1} = b_n = 0,$$

and so the coefficients b_i can be regarded as uniquely defined functions of p and q. Furthermore, u and v are also functions of p and q and are defined, again from equating coefficients, by

(1.3) $$u(p, q) = a_1 - pb_0 - qb_1 = b_{-1}, \qquad v(p, q) = a_0 - qb_0,$$

where b_{-1} is defined naturally by extending (1.2). Solution of the equations

(1.4) $$u(p, q) = 0, \qquad v(p, q) = 0$$

yields p^* and q^* such that $(z^2 + p^* z + q^*)$ is a real quadratic factor of P.

Bairstow suggested that Eqs. (1.4) be solved by Newton's method. Thus a sequence $(p^{(k)}, q^{(k)})$ is generated by

(1.5) $$\begin{bmatrix} p^{(k+1)} \\ q^{(k+1)} \end{bmatrix} = \begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix} - \begin{bmatrix} \partial u/\partial p & \partial u/\partial q \\ \partial v/\partial p & \partial v/\partial q \end{bmatrix}^{-1} \begin{bmatrix} u \\ v \end{bmatrix}_{p=p^{(k)}; q=q^{(k)}}, \quad k = 0, 1, 2, \ldots.$$

Differentiation of Eqs. (1.2) yields a recurrence relation for the derivatives of b_i. If we define $d_i = \partial b_i / \partial p$, then

(1.6)
$$d_i = -b_{i+1} - pd_{i+1} - qd_{i+2} \quad (i = n-3, \ldots, 0, -1),$$
$$d_{n-2} = d_{n-1} = 0.$$

Moreover, it is readily shown that $\partial b_{i-1}/\partial q = \partial b_i/\partial p$, and so derivatives with respect to q are generated by the same recurrence. Thus the iteration (1.5) becomes

$$\begin{bmatrix} p^{(k+1)} \\ q^{(k+1)} \end{bmatrix} = \begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix} - \begin{bmatrix} d_{-1} & d_0 \\ -qd_0 & -b_0 - qd_1 \end{bmatrix}^{-1} \begin{bmatrix} b_{-1} \\ a_0 - qb_0 \end{bmatrix}_{p=p^{(k)}; q=q^{(k)}}$$

that is,

(1.7)
$$\begin{bmatrix} p^{(k+1)} \\ q^{(k+1)} \end{bmatrix} = \begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix} - \frac{1}{J} \begin{bmatrix} -b_{-1}(qd_1 + b_0) - d_0(a_0 - qb_0) \\ qd_0 b_{-1} + d_{-1}(a_0 - qb_0) \end{bmatrix}_{p=p^{(k)}; q=q^{(k)}}$$

where $J = qd_0^2 - d_{-1}(b_0 + qd_1)$, the determinant of the Jacobian of Eqs. (1.4).

Let us define the value of this determinant at the solution (p^*, q^*) by J^*. Then, if $J^* \neq 0$, the theory of Newton's method tells us that the iteration (1.7) will converge quadratically to (p^*, q^*), provided the initial guess $(p^{(0)}, q^{(0)})$ is sufficiently close. Thus today Bairstow's method is regularly used in conjunction with some other method which has a larger domain of convergence but a slower rate, this other method being used to set up a good initial approximation from which Bairstow's method can be put into operation.

If $J^* = 0$, Bairstow's method as it stands is unsatisfactory. The iteration (1.7) may fail to converge from an arbitrarily close starting point (see, for example, [10, p. 190]); if it does converge the rate of convergence may be extremely slow; and in any case the method is likely to fail in practice due to division by a near-zero denominator. It is therefore important to isolate the cases where $J^* = 0$. Henrici [6, p. 114] and Carrano [5] have between them proved the following result: if α_1, α_2 are the roots of $(z^2 + p^*z + q^*)$, then $J^* \neq 0$ if and only if α_1, α_2 are simple, distinct roots of $P(z)$, or real, equal roots of multiplicity two. Thus, in particular, if $(z^2 + p^*z + q^*)$ is a quadratic factor of multiplicity greater than one, then $J^* = 0$. For this case Arthur [2] and Carrano [5] have independently suggested a modification which retains the property of quadratic convergence from a sufficiently close starting point, at some computational expense.

In this paper we take a fresh look at Bairstow's method. In particular, in Section 2 we notice that Eq. (1.1) represents only one of a number of ways in which $P(z)$ may be divided by $(z^2 + pz + q)$. These different divisions give rise to a family of similar methods of which Bairstow's method is one member.

In Section 3 we discuss how to choose the "best" method in a particular situation and the results of Section 4 show this choice to be quite effective. Indeed we are able to recommend a method which seems definitely superior to Bairstow's method and only a little harder to implement.

In Section 5 we discuss the related field of polynomial deflation, and there we acknowledge that the ideas of other authors in that field have contributed greatly to the

work presented here. We mention in particular the papers of Bingham [4] and Peters and Wilkinson [9].

2. Bairstow's Method from a Different Viewpoint. It is helpful to consider Bairstow's method from a different angle. Suppose we ask that

$$P(z) = (z^2 + pz + q)(b_{n-2} z^{n-2} + \ldots + b_0)$$

for given p and q. This requires that the $(n-1)$ unknowns b_i satisfy an overdetermined system of $(n+1)$ linear equations

$$(2.1) \qquad \begin{bmatrix} a_n \\ a_{n-1} \\ a_{n-2} \\ \cdot \\ \cdot \\ \cdot \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ p & \ddots & & \\ q & \ddots & & \\ & \ddots & & \\ & & 1 & \\ & & p & \\ & & q & \end{bmatrix} \begin{bmatrix} b_{n-2} \\ b_{n-3} \\ \cdot \\ \cdot \\ \cdot \\ b_0 \end{bmatrix}.$$

These equations are only consistent when $p = p^*$, $q = q^*$. Suppose, however, that we introduce two new variables u and v; let us add u to the right-hand side of the last but one equation and v to the right-hand side of the last equation. The system of linear equations is now square:

$$\begin{bmatrix} a_n \\ a_{n-1} \\ a_{n-2} \\ \cdot \\ \cdot \\ \cdot \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ p & \ddots & & \\ q & \ddots & & \\ & \ddots & & & \\ & & 1 & & \\ & & p & 1 & \\ & & q & & 1 \end{bmatrix} \begin{bmatrix} b_{n-2} \\ b_{n-3} \\ \cdot \\ \cdot \\ \cdot \\ b_0 \\ \hline u \\ v \end{bmatrix}.$$

Furthermore the unknowns b_i, u, v are explicitly available as functions of p and q by forward substitution. (The b_i are generated by (1.2), and u and v are given by (1.3).) If we now solve $u(p, q) = 0$, $v(p, q) = 0$ for p^*, q^*, then this is precisely the strategy of Bairstow's method.

Now it is clear that the new variables may be added to *any* pair of equations in (2.1) to make the system square. Then, provided the matrix of the set of linear equations is nonsingular, the variables u and v can be found in terms of p and q. Thus it could be that Bairstow's method is only one member of a family of similar algorithms, and it is this idea which we pursue in this paper.

First of all we remark that if u and v are added to a pair of *consecutive* equations then we are assured that the matrix of the resulting set of linear equations is nonsingu-

lar; indeed we shall see later that the solution for the unknown b_i, u, v is then explicitly available. Moreover it is not difficult to construct an example in which u and v are added to a pair of equations that are not consecutive and the resulting matrix is singular. Therefore in what follows we shall always suppose that the pair of equations are consecutive.

For example, we could add the new variables u, v to the right-hand side of the first two equations of the system (2.1). This corresponds to writing

$$(2.2) \quad P(z) = (q + pz + z^2)(c_0 + c_1 z + \ldots + c_{n-2} z^{n-2}) + uz^n + vz^{n-1},$$

that is, dividing "backwards". We have written the quotient polynomial with coefficients c_i, since these will differ from the coefficients of the quotient polynomial in (1.1) except when $p = p^*$, $q = q^*$. The system (2.1) is extended to:

$$
\begin{bmatrix} a_n \\ a_{n-1} \\ a_{n-2} \\ \cdot \\ \cdot \\ \cdot \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}
=
\begin{bmatrix} 1 & & 1 & & & \\ & 1 & p & \diagdown & & \\ & & q & \diagdown & & \\ & & & \diagdown & & \\ & & & & \diagdown & \\ & & & & 1 & \\ & & & & p & \\ & & & & q & \end{bmatrix}
\begin{bmatrix} u \\ v \\ \overline{c_{n-2}} \\ \cdot \\ \cdot \\ \cdot \\ c_1 \\ c_0 \end{bmatrix}
$$

showing that the unknowns c_i, v and u are generated as functions of p and q by back substitution. We have

$$
(2.3) \quad
\begin{cases}
c_i = (a_i - pc_{i-1} - c_{i-2})/q \quad (i = 0, \ldots, n-2), \quad c_{-1} = c_{-2} = 0, \\
u = a_n - c_{n-2}, \\
v = a_{n-1} - pc_{n-2} - c_{n-3} = qc_{n-1}.
\end{cases}
$$

Solution of the equations $u(p, q) = 0$, $v(p, q) = 0$ again yields p^* and q^* such that $(z^2 + p^*z + q^*)$ is a real quadratic factor of P.

The general case where variables u, v are added to the right-hand side of any consecutive pair of equations from (2.1) can be described as follows. Suppose we add new variables u_r and v_r to the $(n-r)$th and $(n-r+1)$st equations, respectively, for some r, $0 \leqslant r \leqslant n-1$. This is equivalent to leaving a remainder of the form $(u_r z^{r+1} + v_r z^r)$ on dividing P by $(z^2 + pz + q)$. In this case the quotient polynomial has coefficients which are either coefficients of the quotient polynomial of (1.1), or coefficients of the quotient polynomial of (2.2). To be precise we have

$$
\begin{aligned}
P(z) = (z^2 + pz + q)(b_{n-2} z^{n-2} + \ldots + b_r z^r + c_{r-1} z^{r-1} + \ldots + c_0) \\
(2.4) \\
+ u_r z^{r+1} + v_r z^r.
\end{aligned}
$$

The system (2.1) is extended to

$$
\begin{bmatrix} a_n \\ a_{n-1} \\ a_{n-2} \\ \cdot \\ \cdot \\ \cdot \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}
=
\begin{bmatrix}
1 & & & & & \\
p & \ddots & & & & \\
q & & 1 & & & \\
& \ddots & p & 1 & & \\
& & q & & 1 & \\
& & & 1 & p & \ddots \\
& & & q & & 1 \\
& & & & p & \ddots \\
& & & & q &
\end{bmatrix}
\begin{bmatrix} b_{n-2} \\ \cdot \\ \cdot \\ b_r \\ \hline u_r \\ v_r \\ \hline c_{r-1} \\ \cdot \\ \cdot \\ c_0 \end{bmatrix}
$$

from which it is clear that the coefficients $b_{n-2}, \ldots, b_r$ are generated by forward substitution (1.2), and the coefficients $c_0, \ldots, c_{r-1}$ by backward substitution (2.3). Furthermore u_r and v_r are found as

(2.5) $$u_r(p, q) = b_{r-1} - c_{r-1}, \qquad v_r(p, q) = q(c_r - b_r)$$

with, of course, b_{r-1} being generated by (1.2) and c_r by (2.3). Solution of the equations

(2.6) $$u_r(p, q) = 0, \qquad v_r(p, q) = 0$$

yields the coefficients of a quadratic factor of P.

Thus we have a choice of n pairs of simultaneous nonlinear equations that we could solve for p^* and q^*, corresponding to values of $r = 0, 1, \ldots, n - 1$. The question of selecting the "best" pair to solve is considered in Section 3. Notice, however, that the case $r = 0$ yields the classical Bairstow method, while the case $r = n - 1$ corresponds to the "backward division" method described earlier in this section.

We carry out the solution of Eqs. (2.6) by Newton's method. An iteration of the method can be written as

(2.7) $$
\begin{bmatrix} p^{(k+1)} \\ q^{(k+1)} \end{bmatrix}
=
\begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix}
-
\begin{bmatrix} \partial u_r/\partial p & \partial u_r/\partial q \\ \partial v_r/\partial p & \partial v_r/\partial q \end{bmatrix}^{-1}
\begin{bmatrix} u_r \\ v_r \end{bmatrix}_{p=p^{(k)};\, q=q^{(k)}}
$$

The partial derivatives of b_i are given by the recurrence (1.6). Partial derivatives of c_i are obtained by differentiating Eqs. (2.3). If we define $e_i = \partial c_i/\partial p$, then

$$e_i = (-c_{i-1} - pe_{i-1} - e_{i-2})/q \quad (i = 1, 2, \ldots, n), \qquad e_0 = e_{-1} = 0.$$

Since $\partial c_{i-1}/\partial q = \partial c_i/\partial p$, one recurrence again generates derivatives with respect to both p and q.

Thus the iteration (2.7) becomes

$$
\begin{bmatrix} p^{(k+1)} \\ q^{(k+1)} \end{bmatrix}
=
\begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix}
-
\begin{bmatrix} d_{r-1} - e_{r-1} & d_r - e_r \\ q(e_r - d_r) & (c_r - b_r) + q(e_{r+1} - d_{r+1}) \end{bmatrix}^{-1}
\begin{bmatrix} u_r \\ v_r \end{bmatrix}_{p=p^{(k)};\, q=q^{(k)}}
$$

Writing $y_i = d_i - e_i$, and noting that $v_r = -qu_{r+1}$, we have

(2.8) $$
\begin{bmatrix} p^{(k+1)} \\ q^{(k+1)} \end{bmatrix}
=
\begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix}
-
\frac{1}{J_r}
\begin{bmatrix} qu_{r+1}y_r - u_r(u_{r+1} + qy_{r+1}) \\ q(u_r y_r - u_{r+1} y_{r-1}) \end{bmatrix}_{p=p^{(k)};\, q=q^{(k)}}
$$

where $J_r = qy_r^2 - y_{r-1}(u_{r+1} + qy_{r+1})$, the determinant of the Jacobian of Eqs. (2.6).

Let us define the value of J_r at the solution (p^*, q^*) by J_r^*. If $J_r^* \neq 0$, the iteration (2.8) will converge quadratically provided $(p^{(0)}, q^{(0)})$ is sufficiently close to (p^*, q^*). If $J_r^* = 0$, the iteration will very likely fail. Thus we are interested in isolating the circumstances under which J_r^* is zero. Let $P(z) = (z^2 + p^*z + q^*)Q(z)$ and let the roots of $(z^2 + p^*z + q^*)$ be α_1, α_2. Then some algebra shows $J_r^* = Q(\alpha_1)Q(\alpha_2)/(q^*)^r$. Thus J_r^* is nonzero if and only if α_1, α_2 are simple distinct roots of $P(z)$, or real equal roots of multiplicity two. Notice that this result is independent of r, and so each algorithm of the family behaves similarly in this respect. Presumably an extension of the family of algorithms, similar to the Arthur-Carrano extension of the classical algorithm, could be devised to handle multiple quadratic factors. However, this is not considered here, and from now on we suppose that $J_r^* \neq 0$.

If the value of r is given, the computational effort required to carry out the iteration (2.8) is on a par with that required to carry out the iteration (1.7) of the classical Bairstow algorithm. In fact an operations count shows that $(4n + 3)$ multiplications/divisions are required to carry out either iteration. Thus if we can find an effective, and computationally inexpensive, criterion for choosing r, we should hope to improve the Bairstow algorithm—perhaps by decreasing the number of iterations required to achieve convergence, or perhaps by increasing the likelihood of the iteration converging at all from a given starting point.

3. Choice of Which Equations to Solve. In this section we turn to the task of determining which pair of the possible n pairs of nonlinear equations we should solve. Notice that the particular choice of equations can be varied as the iteration (2.8) proceeds; that is, the value of r need not be fixed ab initio, but may depend on k if desired.

To begin with we shall ignore any consideration of the computational effort required to make our choice. Consider the equations $u_r(p, q) = 0$, $v_r(p, q) = 0$. Solution of these equations by Newton's method is effected by making first-order approximations to u_r and v_r at the current point $(p^{(k)}, q^{(k)})$, and then solving the resulting linear system. Some measure of the error of these approximations is given by the second-order terms in the expansions of u_r and v_r at $(p^{(k)}, q^{(k)})$. So let us define by $\rho_{u_r}(p^{(k)}, q^{(k)})$ the spectral norm of the matrix of second derivatives of u_r evaluated at $(p^{(k)}, q^{(k)})$, with a similar definition of $\rho_{v_r}(p^{(k)}, q^{(k)})$. Then an appropriate choice of equations to solve might be given by the value of r which minimizes the expression,

$$\{B_r \rho_{u_r}(p^{(k)}, q^{(k)})\}^2 + \{C_r \rho_{v_r}(p^{(k)}, q^{(k)})\}^2,$$

for $r = 0, 1, \ldots, n - 1$. Here B_r, C_r are weighting factors, appropriate choices of which will be discussed later.

Unfortunately, however, such a selection scheme requires computation of the order of five or six times the computation required to carry out the iteration itself once r has been selected. Indeed this scheme will not be considered further since it was found in practice to be no more effective than the following more simple selection scheme.

In this next scheme we choose r as that value which minimizes in some sense the residuals of the equations at the current point, in particular, that value which minimizes the expression,

$$\{B_r |u_r(p^{(k)}, q^{(k)})| + C_r |v_r(p^{(k)}, q^{(k)})|\},$$

for $r = 0, 1, \ldots, n-1$, where B_r and C_r are again positive weighting factors. The following argument suggests an appropriate choice of B_r and C_r. Rearranging Eq. (2.4), we have

$$(z^2 + pz + q)(b_{n-2} z^{n-2} + \ldots + b_r z^r + c_{r-1} z^{r-1} + \ldots + c_0)$$

$$= a_n z^n + \ldots + a_{r+2} z^{r+2} + (a_{r+1} - u_r)z^{r+1} + (a_r - v_r)z^r$$

$$+ a_{r-1} z^{r-1} + \ldots + a_0$$

which can be regarded as a perturbation of the original polynomial $P(z)$. So, if u_r and v_r are regarded as perturbations to a_{r+1}, a_r, then appropriate values of the scaling factors are

$$B_r = \begin{cases} \dfrac{1}{|a_{r+1}|}, & a_{r+1} \neq 0, \\ \infty, & a_r = 0, \end{cases} \qquad C_r = \begin{cases} \dfrac{1}{|a_r|}, & a_r \neq 0, \\ \infty, & a_r = 0. \end{cases}$$

This yields the test expression,

$$\sigma(r) = \begin{cases} |u_r(p^{(k)}, q^{(k)})/a_{r+1}| + |v_r(p^{(k)}, q^{(k)})/a_r|, & a_r \neq 0, a_{r+1} \neq 0, \\ \infty, & \text{otherwise,} \end{cases}$$

defined for $r = 0, 1, \ldots, n-1$. This choice of scaling factors ensures that comparisons of $\sigma(r)$ for different values of r may justifiably be made.

If r is selected as that value which minimizes $\sigma(r)$, then the iteration (2.8) requires $(8n + 3)$ multiplications/divisions—roughly twice that for the classical Bairstow algorithm. Hence even this simple selection scheme seems rather expensive in terms of computer operations. Thus it may only be practicable to select r on this basis at the first iteration, and keep its value fixed thereafter. In the next section we report on some numerical testing of these ideas.

4. Numerical Results. In this section we examine the relative performance of three methods: first, the classical Bairstow algorithm (Method 1); second, an algorithm based on iteration (2.8) which selects r at each iteration on the basis of the test expression $\sigma(r)$ (Method 2); and third, an algorithm which selects r on the basis of $\sigma(r)$ at the first iteration only, and then keeps r fixed thereafter (Method 3). This third algorithm involves twice the computation of the classical algorithm at the first iteration, but no extra in later iterations.

Six polynomials were used in the numerical experiments. The first four polynomials, I–IV, of degrees 4, 6, 10 and 18, were used as test problems by Henrici and Watkins [7]. Respectively, they are Problems 8, 23, 27 of Table 1 and Problem 12 of Table 2 in that paper (see also [11]). Polynomial IV is notable in having 13 of its 18 roots with moduli in the interval [0.98, 1.03].

Polynomial V was chosen as one with roots of widely differing magnitude, namely the polynomial,

$$P(z) = z^6 + 11.1z^5 + 112.11z^4 + 121.21z^3 + 112.11z^2 + 11.1z + 1$$

$$= (z^2 + 10z + 100)(z^2 + z + 1)(z^2 + 0.1z + 0.01).$$

Polynomial VI is of some historical interest. In his investigations into the stability of the aeroplane in 1914, Bairstow [3] required to find roots of the polynomial,

$$P(z) = z^8 + 20.4z^7 + 151.3z^6 + 490z^5 + 687z^4 + 719z^3 + 150z^2 + 109z + 6.87.$$

Bairstow remarked that "the solution presented some difficulties".

The three methods were asked to find each quadratic factor of the six polynomials, starting from three different initial guesses. For the first starting point, $p^{(0)}$ and $q^{(0)}$ were taken to have an error of 5%; for the second the error was 10%; and for the third it was 20%.

Convergence was assumed when both

$$\left| \frac{p^{(k)} - p^*}{p^*} \right| \leqslant 10^{-6} \quad \text{and} \quad \left| \frac{q^{(k)} - q^*}{q^*} \right| \leqslant 10^{-6}.$$

Obviously this is not a suitable criterion for practical purposes (one would use instead the stopping criterion of Adams [1], e.g.) but it is adequate for comparison of different methods. A method was supposed to have failed if convergence was not achieved in 12 iterations.

The following tables summarize the performance of the algorithms on the test examples. In Table 1 we show the number of problems successfully solved by the different methods. The figures are also broken down by the different initial guesses (Table 2). The first figure denotes the number of problems successfully solved by the particular method; any figure which follows in parentheses indicates the number of further occasions on which the method converged, but to a factor which was not that closest to the initial guess.

Method 1	Method 2	Method 3
58(2)	71	72

TABLE 1. *Number of problems successfully solved out of 78 attempted*

Error in initial guesses	Method 1	Method 2	Method 3
5%	24	25	26
10%	19(1)	24	24
20%	15(1)	22	22

TABLE 2. *Number of problems successfully solved out of 26 attempted for each error in initial guess*

In Table 3 below we show the average number of iterations taken to achieve convergence for the different methods. So that a fair comparison may be made, the aver-

age is taken over the 58 test problems on which all three methods were successful. Again these figures are broken down by the different initial guesses (Table 4).

Method 1	Method 2	Method 3
4.66	3.43	3.38

TABLE 3. *Average number of iterations required to achieve convergence on problems on which all methods were successful*

Error in initial guesses	Method 1	Method 2	Method 3
5%	4.04	2.83	2.92
10%	4.63	3.37	3.37
20%	5.67	4.47	4.13

TABLE 4. *Average number of iterations required to achieve convergence on problems on which all methods were successful—breakdown by different errors in initial guesses*

The results show clearly that the new methods, Methods 2 and 3, improve the likelihood of convergence from a given initial guess. It is encouraging that the inexpensive Method 3 is just as successful in this respect as Method 2; apparently a careful initial selection of r is quite adequate. A point worth mentioning is that any problem which caused failure of either of the new methods also caused failure of Bairstow's method, Method 1.

When all three methods converge, the new methods tend to converge rather faster than the classical Bairstow algorithm. In the case of Method 2, this is more than offset by the extra computation required. However, for Method 3 the position is more favorable. On the average the classical algorithm required 4.66 iterations for convergence, while Method 3 required 3.38 iterations, a saving of 1.28 iterations. If we allow one iteration for the extra computation required by Method 3 to select r at the first iteration, then Method 3 still has an advantage of approximately one quarter of an iteration on the average over the classical algorithm. Moreover, this status is retained for each of the different initial guesses. Again it is encouraging that Method 3 converges just as quickly as Method 2.

It was consistently observed that the classical algorithm found difficulty with roots of large modulus, while the new methods were quite successful. This is akin to the situation in the related subject of polynomial deflation, a point which is taken up in the next section.

5. **Polynomial Deflation.** The idea of dividing a polynomial $P(z)$ by a quadratic so as to leave a remainder of the form $(u_r z^{r+1} + v_r z^r)$, where r may take any value between 0 and $n - 1$ and not just 0, is by no means new. It was suggested by Bingham [4], and also by Peters and Wilkinson [9], in the context of polynomial deflation. There we have found a factor $(z^2 + pz + q)$ to maximum possible accuracy and wish to determine the quotient polynomial of $P(z)$ and this factor, so that further factors of

P may be found. Thus p and q are assumed to be very close to p^* and q^*, and any remainder on division will generally be small. This is in contrast to our situation here where $(z^2 + pz + q)$ is often a rather crude estimate of a factor, and remainders on division can be large.

It would appear, however, that the relative behavior of the different division algorithms is fairly independent of how close p and q are to p^* and q^*, in the following sense. Wilkinson [12, pp. 66–67] has pointed out that "forward deflation", that is deflation based on the forward division algorithm (1.2) and leaving a remainder $u_0 z + v_0$, can be disastrous if the roots of the quadratic factor are large in modulus compared with the remaining roots of P. Now forward division is the basis of the classical Bairstow method, and it was quite noticeable in our numerical tests that the method found great difficulty with roots of large modulus.

Both Bingham and Peters and Wilkinson suggested a "composite deflation" method in which the division leaves a remainder of the form $(u_r z^{r+1} + v_r z^r)$, for some r, $0 \leqslant r \leqslant n - 1$. A good choice of r is again one which makes $(u_r z^{r+1} + v_r z^r)$ a small perturbation to the original polynomial (see (2.4)). The criterion of choosing r to minimize $\sigma(r)$ thus seems appropriate also to polynomial deflation, and indeed a similar criterion was suggested by Bingham. Composite deflation is then quite successful irrespective of the order of magnitude in which the roots are determined. In a similar way we find that our new algorithms perform equally well for roots of any size.

Moreover, it is interesting that Method 3 is just as successful as Method 2. This suggests that the initial choice of r, that is the initial choice of the division algorithm, generally remains suitable throughout the computation. Thus when using Method 3 we might have some confidence in basing our deflation process on the same division algorithm as used by the method, in which case we would accept as our deflated polynomial the final quotient polynomial naturally generated by Method 3. It might be safer, though, to apply, after the convergence test has been satisfied, one further step of the method in which the value of r is reselected on the basis of $\sigma(r)$. This should yield a more accurate deflated polynomial and may further improve our estimate of the quadratic factor. Note that we would certainly not feel confident in accepting as our deflated polynomial the final quotient polynomial generated by Bairstow's method, unless the roots are determined in increasing order of magnitude.

6. **Conclusions and Extensions.** The results of Section 4 indicate that the new method, Method 3, is superior to the classical Bairstow method. There seems a much greater chance of convergence from a given starting point; and moreover when the classical method does converge, the new method seems to converge slightly quicker even allowing for the extra computation required. As we mentioned in the introduction, Bairstow's method is commonly used in conjunction with some slow but reliable method which sets up a good initial approximation. The use of Method 3, rather than Bairstow's method, not only increases the likelihood of convergence from such an approximation, thus reducing the chance of having to switch back to the slower method in case of failure, but moreover may allow us to switch from the slower method at an earlier state in the computation.

It is not necessary to solve Eqs. (2.6) by Newton's method; in principle any method for solving a pair of nonlinear equations could be used. Indeed it seems attractive to use some method with a much wider domain of convergence, such as the Levenberg-Marquardt method [8].

Bairstow's method can be extended in an obvious way to find a real factor of $P(z)$ of degree $s > 2$. An approximation to the factor is divided into $P(z)$, using forward division, to leave a remainder $(A_{s-1}^{(0)} z^{s-1} + \ldots + A_0^{(0)})$, where $A_{s-1}^{(0)}, \ldots, A_0^{(0)}$ are functions of the s coefficients of the divisor. Solution of the nonlinear equations $A_i^{(0)} = 0$ $(i = s - 1, \ldots, 0)$ yields the coefficients of the factor itself. The forward division can be replaced by a compositie division which leaves a remainder of the form $(A_{s-1}^{(r)} z^{r+s-1} + \ldots + A_0^{(r)} z^r)$, for any r, $0 \leqslant r \leqslant n - s + 1$. Thus we have the choice of solving the equations $A_i^{(r)} = 0$ $(i = s - 1, \ldots, 0)$ for any r, $0 \leqslant r \leqslant n - s + 1$. Hence a family of algorithms is again available to us, and similar criteria to those described here for the case $s = 2$ should yield good choices of r.

Exactly the same technique can be used to determine complex factors, though the system of equations to be solved is then complex.

Finally, we mention that Bairstow's method is itself an extension of Newton's method for finding a single root, or linear factor of a polynomial. Again Newton's method is based on forward division, and it is possible to replace this by a composite division. This could well lead to a superior algorithm, and we hope to report on this at a later date.

Computer Laboratory
University of Leicester
Leicester, England

1. D. A. ADAMS, "A stopping criterion for polynomial root-finding," *Comm. ACM*, v. 10, 1967, pp. 655–658. MR 39 #2314.

2. D. W. ARTHUR, "Extension of Bairstow's method for multiple quadratic factors," *J. Inst. Math. Appl.*, v. 9, 1972, pp. 194–197.

3. L. BAIRSTOW, *Investigations Relating to the Stability of the Aeroplane*, Reports and Memoranda #154, Advisory Committee for Aeronautics, October 1914, pp. 51–64.

4. J. A. C. BINGHAM, "An improvement to iterative methods of polynomial factorization," *Comm. ACM*, v. 10, 1967, pp. 57–60.

5. F. M. CARRANO, "A modified Bairstow method for multiple zeros of a polynomial," *Math. Comp.*, v. 27, 1973, pp. 781–792.

6. P. HENRICI, *Elements of Numerical Analysis*, Wiley, New York, 1964. MR 29 # 4173.

7. P. HENRICI & B. O. WATKINS, "Finding zeros of a polynomial by the QD algorithm," *Comm. ACM*, v. 8, 1965, pp. 570–574. MR 31 #4172.

8. D. W. MARQUARDT, "An algorithm for least-squares estimation of nonlinear parameters," *J. Soc. Indust. Appl. Math.*, v. 11, 1963, pp. 431–441. MR 27 #3040.

9. G. PETERS & J. H. WILKINSON, "Practical problems arising in the solution of polynomial equations," *J. Inst. Math. Appl.*, v. 8, 1971, pp. 16–35. MR 45 #7980.

10. L. B. RALL, *Computational Solution of Nonlinear Operator Equations*, Wiley, New York, 1969. MR 39 #2289.

11. R. F. THOMAS, "Corrections to numerical data on QD algorithm," *Comm. ACM*, v. 9, 1966, pp. 322–323.

12. J. H. WILKINSON, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, N. J., 1963. MR 28 #4661.