develop the other two guesses. Of course, the algorithm can also be programmed to accommodate three guesses. For languages like Fortran, the code will find complex roots if the proper variables are declared as complex.

7.5 BAIRSTOW'S METHOD

Bairstow's method is an iterative approach related loosely to both the Müller and Newton-Raphson methods. Before launching into a mathematical description of the technique, recall the factored form of the polynomial,

$$f_5(x) = (x+1)(x-4)(x-5)(x+3)(x-2) \tag{7.28}$$

If we divided by a factor that is not a root (for example, $x + 6$), the quotient would be a fourth-order polynomial. However, for this case, a remainder would result.

On the basis of the above, we can elaborate on an algorithm for determining a root of a polynomial: (1) guess a value for the root $x = t$, (2) divide the polynomial by the factor $x - t$, and (3) determine whether there is a remainder. If not, the guess was perfect and the root is equal to t. If there is a remainder, the guess can be systematically adjusted and the procedure repeated until the remainder disappears and a root is located. After this is accomplished, the entire procedure can be repeated for the quotient to locate another root.

Bairstow's method is generally based on this approach. Consequently, it hinges on the mathematical process of dividing a polynomial by a factor. Recall from our discussion of polynomial deflation (Sec. 7.2.2) that synthetic division involves dividing a polynomial by a factor $x - t$. For example, the general polynomial [Eq. (7.1)]

$$f_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \tag{7.29}$$

can be divided by the factor $x - t$ to yield a second polynomial that is one order lower,

$$f_{n-1}(x) = b_1 + b_2 x + b_3 x^2 + \cdots + b_n x^{n-1} \tag{7.30}$$

with a remainder $R = b_0$, where the coefficients can be calculated by the recurrence relationship

$$b_n = a_n$$
$$b_i = a_i + b_{i+1} t \qquad \text{for } i = n-1 \text{ to } 0$$

Note that if t were a root of the original polynomial, the remainder b_0 would equal zero.

To permit the evaluation of complex roots, Bairstow's method divides the polynomial by a quadratic factor $x^2 - rx - s$. If this is done to Eq. (7.29), the result is a new polynomial

$$f_{n-2}(x) = b_2 + b_3 x + \cdots + b_{n-1} x^{n-3} + b_n x^{n-2}$$

with a remainder

$$R = b_1(x - r) + b_0 \tag{7.31}$$

As with normal synthetic division, a simple recurrence relationship can be used to perform the division by the quadratic factor:

$$b_n = a_n \tag{7.32a}$$

$$b_{n-1} = a_{n-1} + rb_n \tag{7.32b}$$

$$b_i = a_i + rb_{i+1} + sb_{i+2} \qquad \text{for } i = n-2 \text{ to } 0 \tag{7.32c}$$

The quadratic factor is introduced to allow the determination of complex roots. This relates to the fact that, if the coefficients of the original polynomial are real, the complex roots occur in conjugate pairs. If $x^2 - rx - s$ is an exact divisor of the polynomial, complex roots can be determined by the quadratic formula. Thus, the method reduces to determining the values of r and s that make the quadratic factor an exact divisor. In other words, we seek the values that make the remainder term equal to zero.

Inspection of Eq. (7.31) leads us to conclude that for the remainder to be zero, b_0 and b_1 must be zero. Because it is unlikely that our initial guesses at the values of r and s will lead to this result, we must determine a systematic way to modify our guesses so that b_0 and b_1 approach zero. To do this, Bairstow's method uses a strategy similar to the Newton-Raphson approach. Because both b_0 and b_1 are functions of both r and s, they can be expanded using a Taylor series, as in [recall Eq. (4.26)]

$$b_1(r + \Delta r, s + \Delta s) = b_1 + \frac{\partial b_1}{\partial r}\Delta r + \frac{\partial b_1}{\partial s}\Delta s$$

$$b_0(r + \Delta r, s + \Delta s) = b_0 + \frac{\partial b_0}{\partial r}\Delta r + \frac{\partial b_0}{\partial s}\Delta s \tag{7.33}$$

where the values on the right-hand side are all evaluated at r and s. Notice that second- and higher-order terms have been neglected. This represents an implicit assumption that $-r$ and $-s$ are small enough that the higher-order terms are negligible. Another way of expressing this assumption is to say that the initial guesses are adequately close to the values of r and s at the roots.

The changes, Δr and Δs, needed to improve our guesses can be estimated by setting Eq. (7.33) equal to zero to give

$$\frac{\partial b_1}{\partial r}\Delta r + \frac{\partial b_1}{\partial s}\Delta s = -b_1 \tag{7.34}$$

$$\frac{\partial b_0}{\partial r}\Delta r + \frac{\partial b_0}{\partial s}\Delta s = -b_0 \tag{7.35}$$

If the partial derivatives of the b's can be determined, these are a system of two equations that can be solved simultaneously for the two unknowns, Δr and Δs. Bairstow showed that the partial derivatives can be obtained by a synthetic division of the b's in a fashion similar to the way in which the b's themselves were derived:

$$c_n = b_n \tag{7.36a}$$

$$c_{n-1} = b_{n-1} + rc_n \tag{7.36b}$$

$$c_i = b_i + rc_{i+1} + sc_{i+2} \qquad \text{for } i = n-2 \text{ to } 1 \tag{7.36c}$$

where $\partial b_0/\partial r = c_1$, $\partial b_0/\partial s = \partial b_1/\partial r = c_2$, and $\partial b_1/\partial s = c_3$. Thus, the partial derivatives are obtained by synthetic division of the b's. Then the partial derivatives can be substituted into Eqs. (7.34) and (7.35) along with the b's to give

$$c_2\Delta r + c_3\Delta s = -b_1$$

$$c_1\Delta r + c_2\Delta s = -b_0$$

These equations can be solved for Δr and Δs, which can in turn be employed to improve the initial guesses of r and s. At each step, an approximate error in r and s can be estimated, as in

$$|\varepsilon_{a,r}| = \left|\frac{\Delta r}{r}\right| 100\% \tag{7.37}$$

and

$$|\varepsilon_{a,s}| = \left|\frac{\Delta s}{s}\right| 100\% \tag{7.38}$$

When both of these error estimates fall below a prespecified stopping criterion ε_s, the values of the roots can be determined by

$$x = \frac{r \pm \sqrt{r^2 + 4s}}{2} \tag{7.39}$$

At this point, three possibilities exist:

1. *The quotient is a third-order polynomial or greater.* For this case, Bairstow's method would be applied to the quotient to evaluate new values for r and s. The previous values of r and s can serve as the starting guesses for this application.
2. *The quotient is a quadratic.* For this case, the remaining two roots could be evaluated directly with Eq. (7.39).
3. *The quotient is a first-order polynomial.* For this case, the remaining single root can be evaluated simply as

$$x = -\frac{s}{r} \tag{7.40}$$

EXAMPLE 7.3

Bairstow's Method

Problem Statement. Employ Bairstow's method to determine the roots of the polynomial

$$f_5(x) = x^5 - 3.5x^4 + 2.75x^3 + 2.125x^2 - 3.875x + 1.25$$

Use initial guesses of $r = s = -1$ and iterate to a level of $\varepsilon_s = 1\%$.

Solution. Equations (7.32) and (7.36) can be applied to compute

$$b_5 = 1 \qquad b_4 = -4.5 \qquad b_3 = 6.25 \qquad b_2 = 0.375 \qquad b_1 = -10.5$$
$$b_0 = 11.375$$

$$c_5 = 1 \qquad c_4 = -5.5 \qquad c_3 = 10.75 \qquad c_2 = -4.875 \qquad c_1 = -16.375$$

Thus, the simultaneous equations to solve for Δr and Δs are

$$-4.875\Delta r + 10.75\Delta s = 10.5$$
$$-16.375\Delta r - 4.875\Delta s = -11.375$$

which can be solved for $\Delta r = 0.3558$ and $\Delta s = 1.1381$. Therefore, our original guesses can be corrected to

$$r = -1 + 0.3558 = -0.6442$$
$$s = -1 + 1.1381 = 0.1381$$

and the approximate errors can be evaluated by Eqs. (7.37) and (7.38),

$$|\varepsilon_{a,r}| = \left|\frac{0.3558}{-0.6442}\right| 100\% = 55.23\% \qquad |\varepsilon_{a,s}| = \left|\frac{1.1381}{0.1381}\right| 100\% = 824.1\%$$

Next, the computation is repeated using the revised values for r and s. Applying Eqs. (7.32) and (7.36) yields

$$b_5 = 1 \qquad b_4 = -4.1442 \qquad b_3 = 5.5578 \qquad b_2 = -2.0276 \qquad b_1 = -1.8013$$
$$b_0 = 2.1304$$
$$c_5 = 1 \qquad c_4 = -4.7884 \qquad c_3 = 8.7806 \qquad c_2 = -8.3454 \qquad c_1 = 4.7874$$

Therefore, we must solve

$$\begin{aligned} -8.3454\Delta r + 8.7806\Delta s &= 1.8013 \\ 4.7874\Delta r - 8.3454\Delta s &= -2.1304 \end{aligned}$$

for $\Delta r = 0.1331$ and $\Delta s = 0.3316$, which can be used to correct the root estimates as

$$r = -0.6442 + 0.1331 = -0.5111 \qquad |\varepsilon_{a,r}| = 26.0\%$$
$$s = 0.1381 + 0.3316 = 0.4697 \qquad |\varepsilon_{a,s}| = 70.6\%$$

The computation can be continued, with the result that after four iterations the method converges on values of $r = -0.5$ ($|\varepsilon_{a,r}| = 0.063\%$) and $s = 0.5$ ($|\varepsilon_{a,s}| = 0.040\%$). Equation (7.39) can then be employed to evaluate the roots as

$$x = \frac{-0.5 \pm \sqrt{(-0.5)^2 + 4(0.5)}}{2} = 0.5, -1.0$$

At this point, the quotient is the cubic equation

$$f(x) = x^3 - 4x^2 + 5.25x - 2.5$$

Bairstow's method can be applied to this polynomial using the results of the previous step, $r = -0.5$ and $s = 0.5$, as starting guesses. Five iterations yield estimates of $r = 2$ and $s = -1.249$, which can be used to compute

$$x = \frac{2 \pm \sqrt{2^2 + 4(-1.249)}}{2} = 1 \pm 0.499i$$

At this point, the quotient is a first-order polynomial that can be directly evaluated by Eq. (7.40) to determine the fifth root: 2.

Note that the heart of Bairstow's method is the evaluation of the b's and c's via Eqs. (7.32) and (7.36). One of the primary strengths of the method is the concise way in which these recurrence relationships can be programmed.

Figure 7.5 lists pseudocode to implement Bairstow's method. The heart of the algorithm consists of the loop to evaluate the b's and c's. Also notice that the code to solve the simultaneous equations checks to prevent division by zero. If this is the case, the values of r and s are perturbed slightly and the procedure is begun again. In addition, the algorithm places a user-defined upper limit on the number of iterations (MAXIT) and should be designed to avoid division by zero while calculating the error estimates. Finally, the algorithm requires initial guesses for r and s (rr and ss in the code). If no prior knowledge of the roots exist, they can be set to zero in the calling program.

(*a*) Bairstow Algorithm

```
SUB Bairstow (a,nn,es,rr,ss,maxit,re,im,ier)
DIMENSION b(nn), c(nn)
r = rr; s = ss; n = nn
ier = 0; ea1 = 1; ea2 = 1
DO
  IF n < 3 OR iter ≥ maxit EXIT
  iter = 0
  DO
    iter = iter + 1
    b(n) = a(n)
    b(n − 1) = a(n − 1) + r * b(n)
    c(n) = b(n)
    c(n − 1) = b(n − 1) + r * c(n)
    DO i = n − 2, 0, −1
      b(i) = a(i) + r * b(i+1) + s * b(i + 2)
      c(i) = b(i) + r * c(i+1) + s * c(i + 2)
    END DO
    det = c(2) * c(2) − c(3) * c(1)
    IF det ≠ 0 THEN
       dr = (−b(1) * c(2) + b(0) * c(3))/det
       ds = (−b(0) * c(2) + b(1) * c(1))/det
       r = r + dr
       s = s + ds
       IF r≠0 THEN ea1 = ABS(dr/r) * 100
       IF s≠0 THEN ea2 = ABS(ds/s) * 100
    ELSE
      r = r + 1
      s = s + 1
      iter = 0
    END IF
    IF ea1 ≤ es AND ea2 ≤ es OR iter ≥ maxit EXIT
  END DO
  CALL Quadroot(r,s,r1,i1,r2,i2)
  re(n) = r1
  im(n) = i1
  re(n − 1) = r2
  im(n − 1) = i2
  n = n − 2
  DO  i = 0, n
    a(i) = b(i + 2)
  END DO
END DO
IF iter < maxit  THEN
   IF n = 2  THEN
      r = −a(1)/a(2)
      s = −a(0)/a(2)
      CALL Quadroot(r,s,r1,i1,r2,i2)
      re(n) = r1
      im(n) = i1
      re(n − 1) = r2
      im(n − 1) = i2
   ELSE
      re(n) = −a(0)/a(1)
      im(n) = 0
   END IF
ELSE
   ier = 1
END IF
END Bairstow
```

(*b*) Roots of Quadratic Algorithm

```
SUB Quadroot(r,s,r1,i1,r2,i2)
disc = r ^ 2 + 4 * s
IF disc > 0  THEN
   r1 = (r + SQRT(disc))/2
   r2 = (r − SQRT(disc))/2
   i1 = 0
   i2 = 0
ELSE
   r1 = r/2
   r2 = r1
   i1 = SQRT(ABS(disc))/2
   i2 = −i1
END IF
END QuadRoot
```

FIGURE 7.5
(*a*) Algorithm for implementing Bairstow's method, along with (*b*) an algorithm to determine the roots of a quadratic.