Bairstow's Method

- **Theory:** In order to extract a real or complex root of a polynomial, Bairstow's method given in the year 1920, attempts to extract a quadratic factor of the form $x^2 + px + q$ from the polynomial P(x). In general if P(x) is divided by $x^2 + px + q$ we obtain a quotient of degree (n−2) of the form,

$$b_0\, x^{(n-2)} + b_1\, x^{(n-3)} + \cdots + b_{n-3}\, x + b_{n-2}$$

and a linear remainder of the form Rx+ S . Our problem then is to find p and q such that

$$R(p, q) = 0\ ; \quad S(p, q) = 0$$

Starting with a guess of p and q, we find corrections Δ p and Δ q , so that

$$R(p+\Delta p, q+\Delta q) = 0\ ; \quad S(p+\Delta p, q+\Delta q) = 0$$

Expanding in Taylor's series and truncating after the first-order terms, as in Newton's method, we get

$$R(p, q) + \frac{\partial R}{\partial p}\Delta p + \frac{\partial R}{\partial q}\Delta q = 0\ ;$$

$$S(p, q) + \frac{\partial S}{\partial p}\Delta p + \frac{\partial S}{\partial q}\Delta q = 0$$

The pair of equations yield expressions for Δp and Δq. Supposing that Δp and Δq have been computed, the procedure is repeated for the corrected values p+Δp and q +Δq.

In order to compute the coefficients b_0 , b_1, b_2,⋯, b_{n-2} , R and S , we use the identity

$$a_0x^n + a_1 x^{n-1} + \cdots + a_n \equiv (x^2 + px + q)(b_0\, x^{n-2} + b_1\, x^{n-3} + \cdots + b_{n-2}) + Rx + S$$

Comparing like powers of x on the two sides, we obtain

$$a_0 = b_0$$

$$a_1 = b_1 + pb_0$$

$$a_2 = b_2 + pb_1 + qb_0$$

… ……

$$a_k = b_k + pb_{k-1} + qb_{k-2}$$

………

$$a_{n-2} = b_{n-2} + pb_{n-3} + qb_{n-4}$$

$$a_{n-1} = R + pb_{n-2} + qb_{n-3}$$

$$a_n = S + qb_{n-2}$$

The quantities $b_0, b_1 \cdots, b_n$, R and S can be obtained recursively from these equations. Defining

$$b_{-2} := b_{-1} := 0;$$

$$b_{n-1} =: R;$$

$$b_n + pb_{n-1} =: S$$

The recursive solution is then,

$$b_k = a_k - pb_{k-1} - qb_{k-2}, \quad (k = 0, 1, 2, \cdots, n) \quad \ldots (1)$$

If we insert the newly defined expressions for R and S in the equations for Δp and Δq, we obtain

$$\left(\frac{\partial bn-1}{\partial p}\right)\Delta p + \left(\frac{\partial bn-1}{\partial q}\right)\Delta q + b_{n-1} = 0$$

$$\left(\frac{\partial bn}{\partial p} + p\frac{\partial bn-1}{\partial p} + b_{n-1}\right)\Delta p + \left(\frac{\partial bn}{\partial q} + p\frac{\partial bn-1}{\partial q}\right)\Delta q + b_n + pb_{n-1} = 0$$

The second equation reduces owing to the first, leading to the pair

$$\frac{\partial bn-1}{\partial p}\Delta p + \frac{\partial bn-1}{\partial q}\Delta q + b_{n-1} = 0$$

And

$$\left(\frac{\partial bn}{\partial p} + b_{n-1}\right)\Delta p + \frac{\partial bn}{\partial q}\Delta q + b_n = 0$$

The partial derivatives of b_k are given by,

$$\frac{\partial bk}{\partial p} = -b_{k-1} - p\frac{\partial bk-1}{\partial p} - q\frac{\partial bk-2}{\partial p};$$

$$(\partial b_0/\partial p) = (\partial b_{-1}/\partial p) = (\partial b_{-2}/\partial p) = 0$$

And

$$\partial b_k/\partial q = -p\,(\partial b_{k-1}/\partial q) - b_{k-2} - q\,(\partial b_{k-2}/\partial q)\;;$$

$$(\partial b_0/\partial q) = (\partial b_{-1}/\partial q) = (\partial b_{-2}/\partial q) = 0$$

Eliminating b_{k-1} from the above two equations,

$$(\partial b_k/\partial p) + p(\partial b_{k-1}/\partial p) + q(\partial b_{k-2}/\partial p) = \partial(b_{k+1}/\partial q) + p(\partial b_k/\partial q) + q(\partial b_{k-1}/\partial q)$$

This relation holds for arbitrary p and q and so equating the coefficients one must have

$$(\partial b_k/\partial q) = (\partial b_{k-1}/\partial p) = -c_{k-2}, \qquad (k = 0,1,2,\cdots,n)$$

With the introduction of quantities c_k , the two equations for the partial derivatives of b_k pass in to the forms

$$c_{k-1} = b_{k-1} - pc_{k-2} - qc_{k-3}; \qquad c_{k-2} = b_{k-2} - pc_{k-3} - qc_{k-4}$$

or equivalently,

$$c_k = b_k - pc_{k-1} - qc_{k-2}, \qquad (k = 0, 1, 2,\cdots, n) \qquad \ldots.(1a)$$

with $c_{-2} = c_{-1} = 0$

Equation (1.a) has a form similar to Eq. (1) and c_k is computed from b_k in exactly the same way as b_k from a_k . With the partial derivatives of b_k determined in this manner, the pair of equations for Δp and Δq become

$$c_{n-2}\Delta p + c_{n-3}\Delta q = b_{n-1}$$

$$(c_{n-1} - b_{n-1})\Delta p + c_{n-2}\Delta q = b_n$$

Solving the two equations we finally obtain the corrections

$$\Delta p = b_{n-1}\, c_{n-2} - b_n\, c_{n-3}/c^2{}_{n-2} - c_{n-3}(c_{n-1} - b_{n-1})$$

$$\Delta q = b_n\, c_{n-2} - b_{n-1}(c_{n-1} - b_{n-1})/\, c^2{}_{n-2} - c_{n-3}(c_{n-1} - b_{n-1}) \quad \ldots.(1.b)$$

It is easy to program the computation of b_k and c_k following Eqs. (1) and (1.a). If the starting values of p and q are not too bad the iterations will converge quadratically as in the Newton's method. The termination of iterations can be based on the smallness of |R| and |S |,or that of $\sqrt{(\mathbf{b^2_{n-1} + b^2_n})}$.If nothing is known about the starting values of p and q, a search operation can be performed over range of values and select the pair that makes $\sqrt{(b^2_{n-1} + b^2_n)}$ the smallest.

On successfully separating the quadratic factor, the deflated polynomial $\mathbf{b_0 x_{n-2} + b_1\, x_{n-3} + \cdots + b_{n-2} = 0}$ can similarly be treated for finding all the roots of the original equation P(x) =0.

- **Algorithm:**

1. Enter the degree of polynomial n.

2. Enter the coefficients of the polynomial from the max power of x.

3. Enter the initial values of U & V of the quadratic factor X^2+UX+V.

4. If n> 2 then call the subroutine BAIRSTOW.

Set D0=1, D1= -U, D2= -V

5. Call subroutine ROOT.

Calculate the roots of the polynomial.

6. If n=2 then call subroutine ROOT for finding the roots of the quadratic equation.

Else find the root of a linear equation in case of n=1.

7. Display all the roots.

8. Stop the program.

- **Programming:**

! BAIRSTOW METHOD

```
      REAL U0,V0,U,V,A,B,D0,D1,D2,R1,R2,X
      INTEGER N
      DIMENSION A(10),B(10)
      WRITE(*,*) 'ENTER DEGREE OF POLYNOMIAL'
      READ(*,*) N
      WRITE(*,*) 'ENTER POLYNOMIAL COEFF.'
      READ(*,*)(A(I),I=1,N+1)
      WRITE(*,*)'ENTER INITIAL VALUES OF U & V'
      READ(*,*)U0,V0
      55 IF(N.GT.2) THEN
      CALL BAIRSTOW(N,A,B,U0,V0,U,V)
      D0=1
      D1=-U
      D2=-V

      WRITE(*,*)'The Roots are'

      CALL ROOT(D0,D1,D2,R1,R2)
      N=N-2
      DO 120 I=1,N+1
      A(I)=B(I+2)
120   CONTINUE
      U0=U
```

```
      V0=V
      GOTO 55
      ENDIF
      IF(N.EQ.2) THEN
      CALL ROOT(A(3),A(2),A(1),R1,R2)
      ELSE
      X=-A(1)/A(2)
      WRITE(*,*)X
      ENDIF
      STOP
      END

      SUBROUTINE BAIRSTOW(N,A,B,U0,V0,U,V)
      INTEGER N
      REAL A,B,U0,V0,U,V,DU,DV
      DIMENSION A(10), B(10), C(10)
100   B(N+1)=A(N+1)
      B(N)=A(N)+U0*B(N+1)
      DO 25 I=N-1,1,-1
      B(I)=A(I)+U0*B(I+1)+V0*B(I+2)
25    CONTINUE
      C(N+1)=0
      C(N)=B(N+1)
      DO 20 I=N-1,1,-1
      C(I)=B(I+1)+U0*C(I+1)+V0*C(I+2)
20    CONTINUE
```

```
      D=C(2)*C(2)-C(1)*C(3)
      DU=-(B(2)*C(2)-B(1)*C(3))/D
      DV=-(B(1)*C(2)-B(2)*C(1))/D
      U=U0+DU
      V=V0+DV
      IF(ABS(DU).GT.0.0000005 .AND. ABS(DV).GT.0.0000005) THEN
      U0=U
      V0=V
      GOTO 100
      ENDIF
      RETURN
      END

      SUBROUTINE ROOT(A,B,C,R1,R2)
      REAL A,B,C,R1,R2,CHECK
      CHECK=B*B-4*A*C
      IF(CHECK.LT.0) THEN
      R1= -B/(2*A)
      R2=SQRT(ABS(CHECK))/(2*A)
      WRITE(*,1)R1,R2
1     FORMAT(1X,F7.4,'+',F6.3,'i')
      WRITE(*,2)R1,R2
2     FORMAT(1X,F7.4,'-',F6.3,'i')
      ELSE IF(CHECK.EQ.0) THEN
      R1=-B/(2*A)
      R2=R1
```

```
WRITE(*,*)R1,R2
ELSE
R1=(-B+SQRT(CHECK))/(2*A)
R2=(-B-SQRT(CHECK))/(2*A)
WRITE(*,*)R1,R2
ENDIF
RETURN
END
```

- **Output:**

```
ENTER DEGREE OF POLYNOMIAL
5
ENTER POLYNOMIAL COEFF.
1
-1
1
-1
-12
12
ENTER INITIAL VALUES OF U & V
0
4
The Roots are
    1.425335      -1.425335
The Roots are
 .0000+ 1.979i
 .0000- 1.979i
    1.000000
Stop - Program terminated.

Press any key to continue
```

- ***ADVANTAGES OF BAIRSTOW METHOD:***

    1. The major advantage is that it has the capabilities of returning of both real and complex roots, and it may take a long time, but it doesn't fail.

    2. An advantage of the method is that it uses real arithmetic only.

    3. Since it is a 2nd order method, convergence is relatively fast.

- ***DISADVANTAGES OF BAIRSTOW METHOD:***

    1. The farther the starting values from the roots, the longer it takes to converge.

    2. Only works for polynomial functions. Really hard to implement and understand.