ALGORITHM 493
Zeros of a Real Polynomial [C2]

M.A. JENKINS

Queen's University

Key Words and Phrases: roots, zeros of a polynomial
CR Categories: 5.15
Language: Fortran

DESCRIPTION

The subroutine RPOLY is a Fortran program to find all the zeros of a real polynomial. The parameters are:

OP	double precision vector of coefficients in order of decreasing powers of the variable
DEGREE	integer degree of the polynomial
ZEROR, ZEROI	double precision vectors of real and imaginary parts of the zeros found by the algorithm
FAIL	logical parameter which is true only if the leading coefficient is zero or if RPOLY has found fewer than degree zeros; in the latter case the degree is reset to the number of zeros found.

The routine as written solves polynomials of degree up to 100; however, this can be modified by systematic changing of the declarations in the routines.

The program is based on the three-stage algorithm described in Jenkins and Traub [1]. The algorithm generates a sequence of polynomials of degree one less than the degree of the given polynomial from which an approximation to a zero or a quadratic factor can be extracted. The first stage is linearly convergent and involves no shift of origin. It is used primarily to bias the decision making process in the second stage in favor of the zeros of small magnitude. The second stage is also linearly convergent and involves a double shift to a complex point and its conjugate. The shift point is chosen arbitrarily on a circle whose radius is less than the magnitude of all the zeros. In most cases, either the shift is closest to a real zero, or the pair of shift points are equidistant and closest to a pair of zeros. In the former case the second stage yields an approximation to the real zero and in the latter case

Received 11 Feb. 1974 and 16 Oct. 1974.

Author's address: Department of Computing and Information Science, Queen's University, Kingston, Ont., Canada K7L 3N6.

it yields an approximation to the real quadratic factor. The third stage involves one of two variable-shift iterations, where the latest approximation(s) is used as the shift point(s) for the generation of the next polynomial. The choice of iteration is based on which case is observed in stage two. The convergence is superquadratic and usually requires only a few steps. The decision processes are made in a fail-safe manner. If the third stage is entered prematurely or the incorrect iteration is chosen, the second stage is resumed. If no convergence is observed in the second stage after a fixed number of steps, a new shift is chosen and the second stage is restarted with a higher fixed limit. The third-stage iterations are terminated when a stopping criterion based on roundoff error analysis has been satisfied. The real zero or quadratic factor is removed by polynomial deflation and the algorithm is repeated on the reduced polynomial.

The first statements of RPOLY set the following four constants which describe the floating-point arithmetic of the computer being used:

ETA	maximum relative representation error, which can be described as the smallest positive floating-point number such that $1 + \text{ETA} > 1$ in floating-point arithmetic
INFIN	large floating-point number near the top of the range
SMALNO	small positive floating-point number near zero
BASE	exponent base for the floating-point number system.

The program is written in a portable subset of standard Fortran. It has been successfully used on the Burroughs B6700 and the IBM 360/50.

The program has been tested on a large number of polynomials, some chosen to test weaknesses common to zerofinding routines, others randomly generated by a number of techniques.

REFERENCES

1. JENKINS, M.A., AND TRAUB, J.F. A three-stage algorithm for real polynomials using quadratic iteration. *SIAM J. Numer. Anal. 7* (1970), 545–566.
2. JENKINS, M.A., AND TRAUB, J.F. Principles for testing polynomial zerofinding programs. *ACM TOMS 1*, 1 (March 1975), 26–34.

ALGORITHM

```
      SUBROUTINE RPOLY(OP, DEGREE, ZEROR, ZEROI,                    RPO   10
     * FAIL)                                                        RPO   20
C FINDS THE ZEROS OF A REAL POLYNOMIAL                              RPO   30
C OP  - DOUBLE PRECISION VECTOR OF COEFFICIENTS IN                  RPO   40
C       ORDER OF DECREASING POWERS.                                 RPO   50
C DEGREE   - INTEGER DEGREE OF POLYNOMIAL.                          RPO   60
C ZEROR, ZEROI - OUTPUT DOUBLE PRECISION VECTORS OF                 RPO   70
C                REAL AND IMAGINARY PARTS OF THE                    RPO   80
C                ZEROS.                                             RPO   90
C FAIL  - OUTPUT LOGICAL PARAMETER, TRUE ONLY IF                    RPO  100
C         LEADING COEFFICIENT IS ZERO OR IF RPOLY                   RPO  110
C         HAS FOUND FEWER THAN DEGREE ZEROS.                        RPO  120
C         IN THE LATTER CASE DEGREE IS RESET TO                     RPO  130
C         THE NUMBER OF ZEROS FOUND.                                RPO  140
C TO CHANGE THE SIZE OF POLYNOMIALS WHICH CAN BE                    RPO  150
C SOLVED, RESET THE DIMENSIONS OF THE ARRAYS IN THE                 RPO  160
C COMMON AREA AND IN THE FOLLOWING DECLARATIONS.                    RPO  170
C THE SUBROUTINE USES SINGLE PRECISION CALCULATIONS                 RPO  180
C FOR SCALING, BOUNDS AND ERROR CALCULATIONS. ALL                   RPO  190
```

```
C CALCULATIONS FOR THE ITERATIONS ARE DONE IN DOUBLE                    RPO  200
C PRECISION.                                                            RPO  210
      COMMON /GLOBAL/ P, QP, K, QK, SVK, SR, SI, U,                     RPO  220
     * V, A, B, C, D, A1, A2, A3, A6, A7, E, F, G,                      RPO  230
     * H, SZR, SZI, LZR, LZI, ETA, ARE, MRE, N, NN                      RPO  240
      DOUBLE PRECISION P(101), QP(101), K(101),                         RPO  250
     * QK(101), SVK(101), SR, SI, U, V, A, B, C, D,                     RPO  260
     * A1, A2, A3, A6, A7, E, F, G, H, SZR, SZI,                        RPO  270
     * LZR, LZI                                                         RPO  280
      REAL ETA, ARE, MRE                                                RPO  290
      INTEGER N, NN                                                     RPO  300
      DOUBLE PRECISION OP(101), TEMP(101),                              RPO  310
     * ZEROR(100), ZEROI(100), T, AA, BB, CC, DABS,                     RPO  320
     * FACTOR                                                           RPO  330
      REAL PT(101), LO, MAX, MIN, XX, YY, COSR,                         RPO  340
     * SINR, XXX, X, SC, BND, XM, FF, DF, DX, INFIN,                    RPO  350
     * SMALNO, BASE                                                     RPO  360
      INTEGER DEGREE, CNT, NZ, I, J, JJ, NM1                            RPO  370
      LOGICAL FAIL, ZEROK                                               RPO  380
C THE FOLLOWING STATEMENTS SET MACHINE CONSTANTS USED                   RPO  390
C IN VARIOUS PARTS OF THE PROGRAM. THE MEANING OF THE                   RPO  400
C FOUR CONSTANTS ARE...                                                 RPO  410
C ETA      THE MAXIMUM RELATIVE REPRESENTATION ERROR                    RPO  420
C          WHICH CAN BE DESCRIBED AS THE SMALLEST                       RPO  430
C          POSITIVE FLOATING POINT NUMBER SUCH THAT                     RPO  440
C          1.D0+ETA IS GREATER THAN 1.                                  RPO  450
C INFINY   THE LARGEST FLOATING-POINT NUMBER.                           RPO  460
C SMALNO   THE SMALLEST POSITIVE FLOATING-POINT NUMBER                  RPO  470
C          IF THE EXPONENT RANGE DIFFERS IN SINGLE AND                  RPO  480
C          DOUBLE PRECISION THEN SMALNO AND INFIN                       RPO  490
C          SHOULD INDICATE THE SMALLER RANGE.                           RPO  500
C BASE     THE BASE OF THE FLOATING-POINT NUMBER                        RPO  510
C          SYSTEM USED.                                                 RPO  520
C THE VALUES BELOW CORRESPOND TO THE BURROUGHS B6700                    RPO  530
      BASE = 8.                                                         RPO  540
      ETA = .5*BASE**(1-26)                                             RPO  550
      INFIN = 4.3E68                                                    RPO  560
      SMALNO = 1.0E-45                                                  RPO  570
C ARE AND MRE REFER TO THE UNIT ERROR IN + AND *                        RPO  580
C RESPECTIVELY. THEY ARE ASSUMED TO BE THE SAME AS                      RPO  590
C ETA.                                                                  RPO  600
      ARE = ETA                                                         RPO  610
      MRE = ETA                                                         RPO  620
      LO = SMALNO/ETA                                                   RPO  630
C INITIALIZATION OF CONSTANTS FOR SHIFT ROTATION                        RPO  640
      XX = .70710678                                                    RPO  650
      YY = -XX                                                          RPO  660
      COSR = -.069756474                                                RPO  670
      SINR = .99756405                                                  RPO  680
      FAIL = .FALSE.                                                    RPO  690
      N = DEGREE                                                        RPO  700
      NN = N + 1                                                        RPO  710
C ALGORITHM FAILS IF THE LEADING COEFFICIENT IS ZERO.                   RPO  720
      IF (OP(1).NE.0.D0) GO TO 10                                       RPO  730
      FAIL = .TRUE.                                                     RPO  740
      DEGREE = 0                                                        RPO  750
      RETURN                                                            RPO  760
C REMOVE THE ZEROS AT THE ORIGIN IF ANY                                 RPO  770
   10 IF (OP(NN).NE.0.0D0) GO TO 20                                     RPO  780
      J = DEGREE - N + 1                                                RPO  790
      ZEROR(J) = 0.D0                                                   RPO  800
      ZEROI(J) = 0.D0                                                   RPO  810
      NN = NN - 1                                                       RPO  820
      N = N - 1                                                         RPO  830
      GO TO 10                                                          RPO  840
C MAKE A COPY OF THE COEFFICIENTS                                       RPO  850
   20 DO 30 I=1,NN                                                      RPO  860
         P(I) = OP(I)                                                   RPO  870
   30 CONTINUE                                                          RPO  880
C START THE ALGORITHM FOR ONE ZERO                                      RPO  890
   40 IF (N.GT.2) GO TO 60                                              RPO  900
      IF (N.LT.1) RETURN                                                RPO  910
C CALCULATE THE FINAL ZERO OR PAIR OF ZEROS                             RPO  920
      IF (N.EQ.2) GO TO 50                                              RPO  930
      ZEROR(DEGREE) = -P(2)/P(1)                                        RPO  940
```

```
      ZEROI(DEGREE) = 0.0D0                                          RPO  950
      RETURN                                                         RPO  960
   50 CALL QUAD(P(1), P(2), P(3), ZEROR(DEGREE-1),                   RPO  970
     * ZEROI(DEGREE-1), ZEROR(DEGREE), ZEROI(DEGREE))                RPO  980
      RETURN                                                         RPO  990
C FIND LARGEST AND SMALLEST MODULI OF COEFFICIENTS.                  RPO 1000
   60 MAX = 0.                                                       RPO 1010
      MIN = INFIN                                                    RPO 1020
      DO 70 I=1,NN                                                   RPO 1030
        X = ABS(SNGL(P(I)))                                          RPO 1040
        IF (X.GT.MAX) MAX = X                                        RPO 1050
        IF (X.NE.0. .AND. X.LT.MIN) MIN = X                          RPO 1060
   70 CONTINUE                                                       RPO 1070
C SCALE IF THERE ARE LARGE OR VERY SMALL COEFFICIENTS                RPO 1080
C COMPUTES A SCALE FACTOR TO MULTIPLY THE                            RPO 1090
C COEFFICIENTS OF THE POLYNOMIAL. THE SCALING IS DONE                RPO 1100
C TO AVOID OVERFLOW AND TO AVOID UNDETECTED UNDERFLOW                RPO 1110
C INTERFERING WITH THE CONVERGENCE CRITERION.                        RPO 1120
C THE FACTOR IS A POWER OF THE BASE                                  RPO 1130
      SC = LO/MIN                                                    RPO 1140
      IF (SC.GT.1.0) GO TO 80                                        RPO 1150
      IF (MAX.LT.10.) GO TO 110                                      RPO 1160
      IF (SC.EQ.0.) SC = SMALNO                                      RPO 1170
      GO TO 90                                                       RPO 1180
   80 IF (INFIN/SC.LT.MAX) GO TO 110                                 RPO 1190
   90 L = ALOG(SC)/ALOG(BASE) + .5                                   RPO 1200
      FACTOR = (BASE*1.0D0)**L                                       RPO 1210
      IF (FACTOR.EQ.1.D0) GO TO 110                                  RPO 1220
      DO 100 I=1,NN                                                  RPO 1230
        P(I) = FACTOR*P(I)                                           RPO 1240
  100 CONTINUE                                                       RPO 1250
C COMPUTE LOWER BOUND ON MODULI OF ZEROS.                            RPO 1260
  110 DO 120 I=1,NN                                                  RPO 1270
        PT(I) = ABS(SNGL(P(I)))                                      RPO 1280
  120 CONTINUE                                                       RPO 1290
      PT(NN) = -PT(NN)                                               RPO 1300
C COMPUTE UPPER ESTIMATE OF BOUND                                    RPO 1310
      X = EXP((ALOG(-PT(NN))-ALOG(PT(1)))/FLOAT(N))                  RPO 1320
      IF (PT(N).EQ.0.) GO TO 130                                     RPO 1330
C IF NEWTON STEP AT THE ORIGIN IS BETTER, USE IT.                    RPO 1340
      XM = -PT(NN)/PT(N)                                             RPO 1350
      IF (XM.LT.X) X = XM                                            RPO 1360
C CHOP THE INTERVAL (0,X) UNTIL FF .LE. 0                            RPO 1370
  130 XM = X*.1                                                      RPO 1380
      FF = PT(1)                                                     RPO 1390
      DO 140 I=2,NN                                                  RPO 1400
        FF = FF*XM + PT(I)                                           RPO 1410
  140 CONTINUE                                                       RPO 1420
      IF (FF.LE.0.) GO TO 150                                        RPO 1430
      X = XM                                                         RPO 1440
      GO TO 130                                                      RPO 1450
  150 DX = X                                                         RPO 1460
C DO NEWTON ITERATION UNTIL X CONVERGES TO TWO                       RPO 1470
C DECIMAL PLACES                                                     RPO 1480
  160 IF (ABS(DX/X).LE..005) GO TO 180                               RPO 1490
      FF = PT(1)                                                     RPO 1500
      DF = FF                                                        RPO 1510
      DO 170 I=2,N                                                   RPO 1520
        FF = FF*X + PT(I)                                            RPO 1530
        DF = DF*X + FF                                               RPO 1540
  170 CONTINUE                                                       RPO 1550
      FF = FF*X + PT(NN)                                             RPO 1560
      DX = FF/DF                                                     RPO 1570
      X = X - DX                                                     RPO 1580
      GO TO 160                                                      RPO 1590
  180 BND = X                                                        RPO 1600
C COMPUTE THE DERIVATIVE AS THE INTIAL K POLYNOMIAL                  RPO 1610
C AND DO 5 STEPS WITH NO SHIFT                                       RPO 1620
      NM1 = N - 1                                                    RPO 1630
      DO 190 I=2,N                                                   RPO 1640
        K(I) = FLOAT(NN-I)*P(I)/FLOAT(N)                             RPO 1650
  190 CONTINUE                                                       RPO 1660
      K(1) = P(1)                                                    RPO 1670
      AA = P(NN)                                                     RPO 1680
      BB = P(N)                                                      RPO 1690
      ZEROK = K(N).EQ.0.D0                                           RPO 1700
```

```
      DO 230 JJ=1,5                                                    RPO 1710
        CC = K(N)                                                      RPO 1720
        IF (ZEROK) GO TO 210                                           RPO 1730
C USE SCALED FORM OF RECURRENCE IF VALUE OF K AT 0 IS                  RPO 1740
C NONZERO                                                              RPO 1750
        T = -AA/CC                                                     RPO 1760
        DO 200 I=1,NM1                                                 RPO 1770
          J = NN - I                                                   RPO 1780
          K(J) = T*K(J-1) + P(J)                                       RPO 1790
  200   CONTINUE                                                       RPO 1800
        K(1) = P(1)                                                    RPO 1810
        ZEROK = DABS(K(N)).LE.DABS(BB)*ETA*10.                         RPO 1820
        GO TO 230                                                      RPO 1830
C USE UNSCALED FORM OF RECURRENCE                                      RPO 1840
  210   DO 220 I=1,NM1                                                 RPO 1850
          J = NN - I                                                   RPO 1860
          K(J) = K(J-1)                                                RPO 1870
  220   CONTINUE                                                       RPO 1880
        K(1) = 0.D0                                                    RPO 1890
        ZEROK = K(N).EQ.0.D0                                           RPO 1900
  230 CONTINUE                                                         RPO 1910
C SAVE K FOR RESTARTS WITH NEW SHIFTS                                  RPO 1920
      DO 240 I=1,N                                                     RPO 1930
        TEMP(I) = K(I)                                                 RPO 1940
  240 CONTINUE                                                         RPO 1950
C LOOP TO SELECT THE QUADRATIC  CORRESPONDING TO EACH                  RPO 1960
C NEW SHIFT                                                            RPO 1970
      DO 280 CNT=1,20                                                  RPO 1980
C QUADRATIC CORRESPONDS TO A DOUBLE SHIFT TO A                         RPO 1990
C NON-REAL POINT AND ITS COMPLEX CONJUGATE. THE POINT                  RPO 2000
C HAS MODULUS BND AND AMPLITUDE ROTATED BY 94 DEGREES                  RPO 2010
C FROM THE PREVIOUS SHIFT                                              RPO 2020
        XXX = COSR*XX - SINR*YY                                        RPO 2030
        YY = SINR*XX + COSR*YY                                         RPO 2040
        XX = XXX                                                       RPO 2050
        SR = BND*XX                                                    RPO 2060
        SI = BND*YY                                                    RPO 2070
        U = -2.0D0*SR                                                  RPO 2080
        V = BND                                                        RPO 2090
C SECOND STAGE CALCULATION, FIXED QUADRATIC                            RPO 2100
        CALL FXSHFR(20*CNT, NZ)                                        RPO 2110
        IF (NZ.EQ.0) GO TO 260                                         RPO 2120
C THE SECOND STAGE JUMPS DIRECTLY TO ONE OF THE THIRD                  RPO 2130
C STAGE ITERATIONS AND RETURNS HERE IF SUCCESSFUL.                     RPO 2140
C DEFLATE THE POLYNOMIAL, STORE THE ZERO OR ZEROS AND                  RPO 2150
C RETURN TO THE MAIN ALGORITHM.                                        RPO 2160
        J = DEGREE - N + 1                                             RPO 2170
        ZEROR(J) = SZR                                                 RPO 2180
        ZEROI(J) = SZI                                                 RPO 2190
        NN = NN - NZ                                                   RPO 2200
        N = NN - 1                                                     RPO 2210
        DO 250 I=1,NN                                                  RPO 2220
          P(I) = QP(I)                                                 RPO 2230
  250   CONTINUE                                                       RPO 2240
        IF (NZ.EQ.1) GO TO 40                                          RPO 2250
        ZEROR(J+1) = LZR                                               RPO 2260
        ZEROI(J+1) = LZI                                               RPO 2270
        GO TO 40                                                       RPO 2280
C IF THE ITERATION IS UNSUCCESSFUL ANOTHER QUADRATIC                   RPO 2290
C IS CHOSEN AFTER RESTORING K                                          RPO 2300
  260   DO 270 I=1,N                                                   RPO 2310
          K(I) = TEMP(I)                                               RPO 2320
  270   CONTINUE                                                       RPO 2330
  280 CONTINUE                                                         RPO 2340
C RETURN WITH FAILURE IF NO CONVERGENCE WITH 20                        RPO 2350
C SHIFTS                                                               RPO 2360
      FAIL = .TRUE.                                                    RPO 2370
      DEGREE = DEGREE - N                                              RPO 2380
      RETURN                                                           RPO 2390
      END                                                              RPO 2400



      SUBROUTINE FXSHFR(L2, NZ)                                        FXS   10
C COMPUTES UP TO  L2  FIXED SHIFT K-POLYNOMIALS,                       FXS   20
C TESTING FOR CONVERGENCE IN THE LINEAR OR QUADRATIC                   FXS   30
```

```
C CASE. INITIATES ONE OF THE VARIABLE SHIFT                          FXS   40
C ITERATIONS AND RETURNS WITH THE NUMBER OF ZEROS                    FXS   50
C FOUND.                                                             FXS   60
C L2 - LIMIT OF FIXED SHIFT STEPS                                    FXS   70
C NZ - NUMBER OF ZEROS FOUND                                         FXS   80
      COMMON /GLOBAL/ P, QP, K, QK, SVK, SR, SI, U,                  FXS   90
     * V, A, B, C, D, A1, A2, A3, A6, A7, E, F, G,                   FXS  100
     * H, SZR, SZI, LZR, LZI, ETA, ARE, MRE, N, NN                   FXS  110
      DOUBLE PRECISION P(101), QP(101), K(101),                      FXS  120
     * QK(101), SVK(101), SR, SI, U, V, A, B, C, D,                  FXS  130
     * A1, A2, A3, A6, A7, E, F, G, H, SZR, SZI,                     FXS  140
     * LZR, LZI                                                      FXS  150
      REAL ETA, ARE, MRE                                             FXS  160
      INTEGER N, NN                                                  FXS  170
      DOUBLE PRECISION SVU, SVV, UI, VI, S                           FXS  180
      REAL BETAS, BETAV, OSS, OVV, SS, VV, TS, TV,                   FXS  190
     * OTS. OTV, TVV, TSS                                            FXS  200
      INTEGER L2, NZ, TYPE, I, J, IFLAG                              FXS  210
      LOGICAL VPASS, SPASS, VTRY, STRY                               FXS  220
      NZ = 0                                                         FXS  230
      BETAV = .25                                                    FXS  240
      BETAS = .25                                                    FXS  250
      OSS = SR                                                       FXS  260
      OVV = V                                                        FXS  270
C EVALUATE POLYNOMIAL BY SYNTHETIC DIVISION                          FXS  280
      CALL QUADSD(NN, U, V, P, QP, A, B)                             FXS  290
      CALL CALCSC(TYPE)                                              FXS  300
      DO 80 J=1,L2                                                   FXS  310
C CALCULATE NEXT K POLYNOMIAL AND ESTIMATE V                         FXS  320
        CALL NEXTK(TYPE)                                             FXS  330
        CALL CALCSC(TYPE)                                            FXS  340
        CALL NEWEST(TYPE, UI, VI)                                    FXS  350
        VV = VI                                                      FXS  360
C ESTIMATE S                                                         FXS  370
        SS = 0.                                                      FXS  380
        IF (K(N).NE.0.D0) SS = -P(NN)/K(N)                           FXS  390
        TV = 1.                                                      FXS  400
        TS = 1.                                                      FXS  410
        IF (J.EQ.1 .OR. TYPE.EQ.3) GO TO 70                          FXS  420
C COMPUTE RELATIVE MEASURES OF CONVERGENCE OF S AND V                FXS  430
C SEQUENCES                                                          FXS  440
        IF (VV.NE.0.) TV = ABS((VV-OVV)/VV)                          FXS  450
        IF (SS.NE.0.) TS = ABS((SS-OSS)/SS)                          FXS  460
C IF DECREASING, MULTIPLY TWO MOST RECENT                            FXS  470
C CONVERGENCE MEASURES                                               FXS  480
        TVV = 1.                                                     FXS  490
        IF (TV.LT.OTV) TVV = TV*OTV                                  FXS  500
        TSS = 1.                                                     FXS  510
        IF (TS.LT.OTS) TSS = TS*OTS                                  FXS  520
C COMPARE WITH CONVERGENCE CRITERIA                                  FXS  530
        VPASS = TVV.LT.BETAV                                         FXS  540
        SPASS = TSS.LT.BETAS                                         FXS  550
        IF (.NOT.(SPASS .OR. VPASS)) GO TO 70                        FXS  560
C AT LEAST ONE SEQUENCE HAS PASSED THE CONVERGENCE                   FXS  570
C TEST. STORE VARIABLES BEFORE ITERATING                             FXS  580
        SVU = U                                                      FXS  590
        SVV = V                                                      FXS  600
        DO 10 I=1,N                                                  FXS  610
          SVK(I) = K(I)                                              FXS  620
   10   CONTINUE                                                     FXS  630
        S = SS                                                       FXS  640
C CHOOSE ITERATION ACCORDING TO THE FASTEST                          FXS  650
C CONVERGING SEQUENCE                                                FXS  660
        VTRY = .FALSE.                                               FXS  670
        STRY = .FALSE.                                               FXS  680
        IF (SPASS .AND. ((.NOT.VPASS) .OR.                           FXS  690
     *   TSS.LT.TVV)) GO TO 40                                       FXS  700
   20   CALL QUADIT(UI, VI, NZ)                                      FXS  710
        IF (NZ.GT.0) RETURN                                          FXS  720
C QUADRATIC ITERATION HAS FAILED. FLAG THAT IT HAS                   FXS  730
C BEEN TRIED AND DECREASE THE CONVERGENCE CRITERION.                 FXS  740
        VTRY = .TRUE.                                                FXS  750
        BETAV = BETAV*.25                                            FXS  760
C TRY LINEAR ITERATION IF IT HAS NOT BEEN TRIED AND                  FXS  770
C THE S SEQUENCE IS CONVERGING                                       FXS  780
        IF (STRY .OR. (.NOT.SPASS)) GO TO 50                         FXS  790
```

```
        DO 30 I=1,N                                                    FXS  800
          K(I) = SVK(I)                                                FXS  810
   30   CONTINUE                                                       FXS  820
   40   CALL REALIT(S, NZ, IFLAG)                                      FXS  830
        IF (NZ.GT.0) RETURN                                            FXS  840
C LINEAR ITERATION HAS FAILED. FLAG THAT IT HAS BEEN                   FXS  850
C TRIED AND DECREASE THE CONVERGENCE CRITERION                         FXS  860
        STRY = .TRUE.                                                  FXS  870
        BETAS = BETAS*.25                                              FXS  880
        IF (IFLAG.EQ.0) GO TO 50                                       FXS  890
C IF LINEAR ITERATION SIGNALS AN ALMOST DOUBLE REAL                    FXS  900
C ZERO ATTEMPT QUADRATIC INTERATION                                    FXS  910
        UI = -(S+S)                                                    FXS  920
        VI = S*S                                                       FXS  930
        GO TO 20                                                       FXS  940
C RESTORE VARIABLES                                                    FXS  950
   50   U = SVU                                                        FXS  960
        V = SVV                                                        FXS  970
        DO 60 I=1,N                                                    FXS  980
          K(I) = SVK(I)                                                FXS  990
   60   CONTINUE                                                       FXS 1000
C TRY QUADRATIC ITERATION IF IT HAS NOT BEEN TRIED                     FXS 1010
C AND THE V SEQUENCE IS CONVERGING                                     FXS 1020
        IF (VPASS .AND. (.NOT.VTRY)) GO TO 20                          FXS 1030
C RECOMPUTE QP AND SCALAR VALUES TO CONTINUE THE                       FXS 1040
C SECOND STAGE                                                         FXS 1050
        CALL QUADSD(NN, U, V, P, QP, A, B)                             FXS 1060
        CALL CALCSC(TYPE)                                              FXS 1070
   70   OVV = VV                                                       FXS 1080
        OSS = SS                                                       FXS 1090
        OTV = TV                                                       FXS 1100
        OTS = TS                                                       FXS 1110
   80 CONTINUE                                                         FXS 1120
      RETURN                                                           FXS 1130
      END                                                              FXS 1140



      SUBROUTINE QUADIT(UU, VV, NZ)                                    QUA   10
C VARIABLE-SHIFT K-POLYNOMIAL ITERATION FOR A                          QUA   20
C QUADRATIC FACTOR CONVERGES ONLY IF THE ZEROS ARE                     QUA   30
C EQUIMODULAR OR NEARLY SO.                                            QUA   40
C UU,VV - COEFFICIENTS OF STARTING QUADRATIC                           QUA   50
C NZ - NUMBER OF ZERO FOUND                                            QUA   60
      COMMON /GLOBAL/ P, QP, K, QK, SVK, SR, SI, U,                    QUA   70
     * V, A, B, C, D, A1, A2, A3, A6, A7, E, F, G,                     QUA   80
     * H, SZR, SZI, LZR, LZI, ETA, ARE, MRE, N, NN                     QUA   90
      DOUBLE PRECISION P(101), QP(101), K(101),                        QUA  100
     * QK(101), SVK(101), SR, SI, U, V, A, B, C, D,                    QUA  110
     * A1, A2, A3, A6, A7, E, F, G, H, SZR, SZI,                       QUA  120
     * LZR, LZI                                                        QUA  130
      REAL ETA, ARE, MRE                                               QUA  140
      INTEGER N, NN                                                    QUA  150
      DOUBLE PRECISION UI, VI, UU, VV, DABS                            QUA  160
      REAL MS, MP, OMP, EE, RELSTP, T, ZM                              QUA  170
      INTEGER NZ, TYPE, I, J                                           QUA  180
      LOGICAL TRIED                                                    QUA  190
      NZ = 0                                                           QUA  200
      TRIED = .FALSE.                                                  QUA  210
      U = UU                                                           QUA  220
      V = VV                                                           QUA  230
      J = 0                                                            QUA  240
C MAIN LOOP                                                            QUA  250
   10 CALL QUAD(1.D0, U, V, SZR, SZI, LZR, LZI)                        QUA  260
C RETURN IF ROOTS OF THE QUADRATIC ARE REAL AND NOT                    QUA  270
C CLOSE TO MULTIPLE OR NEARLY EQUAL AND  OF OPPOSITE                   QUA  280
C SIGN                                                                 QUA  290
      IF (DABS(DABS(SZR)-DABS(LZR)).GT..01D0*                          QUA  300
     * DABS(LZR)) RETURN                                              QUA  310
C EVALUATE POLYNOMIAL BY QUADRATIC SYNTHETIC DIVISION                  QUA  320
      CALL QUADSD(NN, U, V, P, QP, A, B)                               QUA  330
      MP = DABS(A-SZR*B) + DABS(SZI*B)                                 QUA  340
C COMPUTE A RIGOROUS  BOUND ON THE ROUNDING ERROR IN                   QUA  350
```

```
C EVALUTING P                                                          QUA  360
      ZM = SQRT(ABS(SNGL(V)))                                          QUA  370
      EE = 2.*ABS(SNGL(QP(1)))                                         QUA  380
      T = -SZR*B                                                       QUA  390
      DO 20 I=2,N                                                      QUA  400
        EE = EE*ZM + ABS(SNGL(QP(I)))                                  QUA  410
   20 CONTINUE                                                         QUA  420
      EE = EE*ZM + ABS(SNGL(A)+T)                                      QUA  430
      EE = (5.*MRE+4.*ARE)*EE - (5.*MRE+2.*ARE)*                       QUA  440
     * (ABS(SNGL(A)+T)+ABS(SNGL(B))*ZM) +                             QUA  450
     * 2.*ARE*ABS(T)                                                  QUA  460
C ITERATION HAS CONVERGED SUFFICIENTLY IF THE                          QUA  470
C POLYNOMIAL VALUE IS LESS THAN 20 TIMES THIS BOUND                    QUA  480
      IF (MP.GT.20.*EE) GO TO 30                                       QUA  490
      NZ = 2                                                           QUA  500
      RETURN                                                           QUA  510
   30 J = J + 1                                                        QUA  520
C STOP ITERATION AFTER 20 STEPS                                        QUA  530
      IF (J.GT.20) RETURN                                              QUA  540
      IF (J.LT.2) GO TO 50                                             QUA  550
      IF (RELSTP.GT..01 .OR. MP.LT.OMP .OR. TRIED)                     QUA  560
     * GO TO 50                                                        QUA  570
C A CLUSTER APPEARS TO BE STALLING THE CONVERGENCE.                    QUA  580
C FIVE FIXED SHIFT STEPS ARE TAKEN WITH A U,V CLOSE                    QUA  590
C TO THE CLUSTER                                                       QUA  600
      IF (RELSTP.LT.ETA) RELSTP = ETA                                  QUA  610
      RELSTP = SQRT(RELSTP)                                            QUA  620
      U = U - U*RELSTP                                                 QUA  630
      V = V + V*RELSTP                                                 QUA  640
      CALL QUADSD(NN, U, V, P, QP, A, B)                               QUA  650
      DO 40 I=1,5                                                      QUA  660
        CALL CALCSC(TYPE)                                              QUA  670
        CALL NEXTK(TYPE)                                               QUA  680
   40 CONTINUE                                                         QUA  690
      TRIED = .TRUE.                                                   QUA  700
      J = 0                                                            QUA  710
   50 OMP = MP                                                         QUA  720
C CALCULATE NEXT K POLYNOMIAL AND NEW U AND V                          QUA  730
      CALL CALCSC(TYPE)                                                QUA  740
      CALL NEXTK(TYPE)                                                 QUA  750
      CALL CALCSC(TYPE)                                                QUA  760
      CALL NEWEST(TYPE, UI, VI)                                        QUA  770
C IF VI IS ZERO THE ITERATION IS NOT CONVERGING                        QUA  780
      IF (VI.EQ.0.D0) RETURN                                           QUA  790
      RELSTP = DABS((VI-V)/VI)                                         QUA  800
      U = UI                                                           QUA  810
      V = VI                                                           QUA  820
      GO TO 10                                                         QUA  830
      END                                                              QUA  840



      SUBROUTINE REALIT(SSS, NZ, IFLAG)                                REA   10
C VARIABLE-SHIFT H POLYNOMIAL ITERATION FOR A REAL                     REA   20
C ZERO.                                                                REA   30
C SSS   - STARTING ITERATE                                             REA   40
C NZ    - NUMBER OF ZERO FOUND                                         REA   50
C IFLAG - FLAG TO INDICATE A PAIR OF ZEROS NEAR REAL                   REA   60
C         AXIS.                                                        REA   70
      COMMON /GLOBAL/ P, QP, K, QK, SVK, SR, SI, U,                    REA   80
     * V, A, B, C, D, A1, A2, A3, A6, A7, E, F, G,                    REA   90
     * H, SZR, SZI, LZR, LZI, ETA, ARE, MRE, N, NN                    REA  100
      DOUBLE PRECISION P(101), QP(101), K(101),                        REA  110
     * QK(101), SVK(101), SR, SI, U, V, A, B, C, D,                    REA  120
     * A1, A2, A3, A6, A7, E, F, G, H, SZR, SZI,                       REA  130
     * LZR, LZI                                                        REA  140
      REAL ETA, ARE, MRE                                               REA  150
      INTEGER N, NN                                                    REA  160
      DOUBLE PRECISION PV, KV, T, S, SSS, DABS                         REA  170
      REAL MS, MP, OMP, EE                                             REA  180
      INTEGER NZ, IFLAG, I, J, NM1                                     REA  190
      NM1 = N - 1                                                      REA  200
      NZ = 0                                                           REA  210
      S = SSS                                                          REA  220
      IFLAG = 0                                                        REA  230
      J = 0                                                            REA  240
```

```
C MAIN LOOP                                                          REA  250
   10 PV = P(1)                                                      REA  260
C EVALUATE P AT S                                                    REA  270
      QP(1) = PV                                                     REA  280
      DO 20 I=2,NN                                                   REA  290
        PV = PV*S + P(I)                                             REA  300
        QP(I) = PV                                                   REA  310
   20 CONTINUE                                                       REA  320
      MP = DABS(PV)                                                  REA  330
C COMPUTE A RIGOROUS BOUND ON THE ERROR IN EVALUATING                REA  340
C P                                                                  REA  350
      MS = DABS(S)                                                   REA  360
      EE = (MRE/(ARE+MRE))*ABS(SNGL(QP(1)))                          REA  370
      DO 30 I=2,NN                                                   REA  380
        EE = EE*MS + ABS(SNGL(QP(I)))                                REA  390
   30 CONTINUE                                                       REA  400
C ITERATION HAS CONVERGED SUFFICIENTLY IF THE                        REA  410
C POLYNOMIAL VALUE IS LESS THAN 20 TIMES THIS BOUND                  REA  420
      IF (MP.GT.20.*((ARE+MRE)*EE-MRE*MP)) GO TO 40                  REA  430
      NZ = 1                                                         REA  440
      SZR = S                                                        REA  450
      SZI = 0.D0                                                     REA  460
      RETURN                                                         REA  470
   40 J = J + 1                                                      REA  480
C STOP ITERATION AFTER 10 STEPS                                      REA  490
      IF (J.GT.10) RETURN                                            REA  500
      IF (J.LT.2) GO TO 50                                           REA  510
      IF (DABS(T).GT..001*DABS(S-T) .OR. MP.LE.OMP)                  REA  520
     * GO TO 50                                                      REA  530
C A CLUSTER OF ZEROS NEAR THE REAL AXIS HAS BEEN                     REA  540
C ENCOUNTERED RETURN WITH IFLAG SET TO INITIATE A                    REA  550
C QUADRATIC ITERATION                                                REA  560
      IFLAG = 1                                                      REA  570
      SSS = S                                                        REA  580
      RETURN                                                         REA  590
C RETURN IF THE POLYNOMIAL VALUE HAS INCREASED                       REA  600
C SIGNIFICANTLY                                                      REA  610
   50 OMP = MP                                                       REA  620
C COMPUTE T, THE NEXT POLYNOMIAL, AND THE NEW ITERATE                REA  630
      KV = K(1)                                                      REA  640
      QK(1) = KV                                                     REA  650
      DO 60 I=2,N                                                    REA  660
        KV = KV*S + K(I)                                             REA  670
        QK(I) = KV                                                   REA  680
   60 CONTINUE                                                       REA  690
      IF (DABS(KV).LE.DABS(K(N))*10.*ETA) GO TO 80                   REA  700
C USE THE SCALED FORM OF THE RECURRENCE IF THE VALUE                 REA  710
C OF K AT S IS NONZERO                                               REA  720
      T = -PV/KV                                                     REA  730
      K(1) = QP(1)                                                   REA  740
      DO 70 I=2,N                                                    REA  750
        K(I) = T*QK(I-1) + QP(I)                                     REA  760
   70 CONTINUE                                                       REA  770
      GO TO 100                                                      REA  780
C USE UNSCALED FORM                                                  REA  790
   80 K(1) = 0.0D0                                                   REA  800
      DO 90 I=2,N                                                    REA  810  \
        K(I) = QK(I-1)                                               REA  820
   90 CONTINUE                                                       REA  830
  100 KV = K(1)                                                      REA  840
      DO 110 I=2,N                                                   REA  850
        KV = KV*S + K(I)                                             REA  860
  110 CONTINUE                                                       REA  870
      T = 0.D0                                                       REA  880
      IF (DABS(KV).GT.DABS(K(N))*10.*ETA) T = -PV/KV                 REA  890
      S = S + T                                                      REA  900
      GO TO 10                                                       REA  910
      END                                                            REA  920



      SUBROUTINE CALCSC(TYPE)                                        CAL   10
C THIS ROUTINE CALCULATES SCALAR QUANTITIES USED TO                  CAL   20
C COMPUTE THE NEXT K POLYNOMIAL AND NEW ESTIMATES OF                 CAL   30
```

```
C THE QUADRATIC COEFFICIENTS.                                          CAL   40
C TYPE - INTEGER VARIABLE SET HERE INDICATING HOW THE                  CAL   50
C CALCULATIONS ARE NORMALIZED TO AVOID OVERFLOW                        CAL   60
      COMMON /GLOBAL/ P, QP, K, QK, SVK, SR, SI, U,                    CAL   70
     * V, A, B, C, D, A1, A2, A3, A6, A7, E, F, G,                     CAL   80
     * H, SZR, SZI, LZR, LZI, ETA, ARE, MRE, N, NN                     CAL   90
      DOUBLE PRECISION P(101), QP(101), K(101),                        CAL  100
     * QK(101), SVK(101), SR, SI, U, V, A, B, C, D,                    CAL  110
     * A1, A2, A3, A6, A7, E, F, G, H, SZR, SZI,                       CAL  120
     * LZR, LZI                                                        CAL  130
      REAL ETA, ARE, MRE                                               CAL  140
      INTEGER N, NN                                                    CAL  150
      DOUBLE PRECISION DABS                                            CAL  160
      INTEGER TYPE                                                     CAL  170
C SYNTHETIC DIVISION OF K BY THE QUADRATIC 1,U,V                       CAL  180
      CALL QUADSD(N, U, V, K, QK, C, D)                                CAL  190
      IF (DABS(C).GT.DABS(K(N))*100.*ETA) GO TO 10                     CAL  200
      IF (DABS(D).GT.DABS(K(N-1))*100.*ETA) GO TO 10                   CAL  210
      TYPE = 3                                                         CAL  220
C TYPE=3 INDICATES THE QUADRATIC IS ALMOST A FACTOR                    CAL  230
C OF K                                                                 CAL  240
      RETURN                                                           CAL  250
   10 IF (DABS(D).LT.DABS(C)) GO TO 20                                 CAL  260
      TYPE = 2                                                         CAL  270
C TYPE=2 INDICATES THAT ALL FORMULAS ARE DIVIDED BY D                  CAL  280
      E = A/D                                                          CAL  290
      F = C/D                                                          CAL  300
      G = U*B                                                          CAL  310
      H = V*B                                                          CAL  320
      A3 = (A+G)*E + H*(B/D)                                           CAL  330
      A1 = B*F - A                                                     CAL  340
      A7 = (F+U)*A + H                                                 CAL  350
      RETURN                                                           CAL  360
   20 TYPE = 1                                                         CAL  370
C TYPE=1 INDICATES THAT ALL FORMULAS ARE DIVIDED BY C                  CAL  380
      E = A/C                                                          CAL  390
      F = D/C                                                          CAL  400
      G = U*E                                                          CAL  410
      H = V*B                                                          CAL  420
      A3 = A*E + (H/C+G)*B                                             CAL  430
      A1 = B - A*(D/C)                                                 CAL  440
      A7 = A + G*D + H*F                                               CAL  450
      RETURN                                                           CAL  460
      END                                                              CAL  470



      SUBROUTINE NEXTK(TYPE)                                           NEX   10
C COMPUTES THE NEXT K POLYNOMIALS USING SCALARS                        NEX   20
C COMPUTED IN CALCSC                                                   NEX   30
      COMMON /GLOBAL/ P, QP, K, QK, SVK, SR, SI, U,                    NEX   40
     * V, A, B, C, D, A1, A2, A3, A6, A7, E, F, G,                     NEX   50
     * H, SZR, SZI, LZR, LZI, ETA, ARE, MRE, N, NN                     NEX   60
      DOUBLE PRECISION P(101), QP(101), K(101),                        NEX   70
     * QK(101), SVK(101), SR, SI, U, V, A, B, C, D,                    NEX   80
     * A1, A2, A3, A6, A7, E, F, G, H, SZR, SZI,                       NEX   90
     * LZR, LZI                                                        NEX  100
      REAL ETA, ARE, MRE                                               NEX  110
      INTEGER N, NN                                                    NEX  120
      DOUBLE PRECISION TEMP, DABS                                      NEX  130
      INTEGER TYPE                                                     NEX  140
      IF (TYPE.EQ.3) GO TO 40                                          NEX  150
      TEMP = A                                                         NEX  160
      IF (TYPE.EQ.1) TEMP = B                                          NEX  170
      IF (DABS(A1).GT.DABS(TEMP)*ETA*10.) GO TO 20                     NEX  180
C IF A1 IS NEARLY ZERO THEN USE A SPECIAL FORM OF THE                  NEX  190
C RECURRENCE                                                           NEX  200
      K(1) = 0.D0                                                      NEX  210
      K(2) = -A7*QP(1)                                                 NEX  220
      DO 10 I=3,N                                                      NEX  230
        K(I) = A3*QK(I-2) - A7*QP(I-1)                                 NEX  240
   10 CONTINUE                                                         NEX  250
      RETURN                                                           NEX  260
C USE SCALED FORM OF THE RECURRENCE                                    NEX  270
   20 A7 = A7/A1                                                       NEX  280
      A3 = A3/A1                                                       NEX  290
```

```
      K(1) = QP(1)                                                    NEX  300
      K(2) = QP(2) - A7*QP(1)                                         NEX  310
      DO 30 I=3,N                                                     NEX  320
        K(I) = A3*QK(I-2) - A7*QP(I-1) + QP(I)                        NEX  330
   30 CONTINUE                                                        NEX  340
      RETURN                                                          NEX  350
C USE UNSCALED FORM OF THE RECURRENCE IF TYPE IS 3                    NEX  360
   40 K(1) = 0.D0                                                     NEX  370
      K(2) = 0.D0                                                     NEX  380
      DO 50 I=3,N                                                     NEX  390
        K(I) = QK(I-2)                                                NEX  400
   50 CONTINUE                                                        NEX  410
      RETURN                                                          NEX  420
      END                                                             NEX  430


      SUBROUTINE NEWEST(TYPE, UU, VV)                                 NEW   10
C COMPUTE NEW ESTIMATES OF THE QUADRATIC COEFFICIENTS                 NEW   20
C USING THE SCALARS COMPUTED IN CALCSC.                               NEW   30
      COMMON /GLOBAL/ P, QP, K, QK, SVK, SR, SI, U,                   NEW   40
     * V, A, B, C, D, A1, A2, A3, A6, A7, E, F, G,                    NEW   50
     * H, SZR, SZI, LZR, LZI, ETA, ARE, MRE, N, NN                    NEW   60
      DOUBLE PRECISION P(101), QP(101), K(101),                       NEW   70
     * QK(101), SVK(101), SR, SI, U, V, A, B, C, D,                   NEW   80
     * A1, A2, A3, A6, A7, E, F, G, H, SZR, SZI,                      NEW   90
     * LZR, LZI                                                       NEW  100
      REAL ETA, ARE, MRE                                              NEW  110
      INTEGER N, NN                                                   NEW  120
      DOUBLE PRECISION A4, A5, B1, B2, C1, C2, C3,                    NEW  130
     * C4, TEMP, UU, VV                                               NEW  140
      INTEGER TYPE                                                    NEW  150
C USE FORMULAS APPROPRIATE TO SETTING OF TYPE.                        NEW  160
      IF (TYPE.EQ.3) GO TO 30                                         NEW  170
      IF (TYPE.EQ.2) GO TO 10                                         NEW  180
      A4 = A + U*B + H*F                                              NEW  190
      A5 = C + (U+V*F)*D                                              NEW  200
      GO TO 20                                                        NEW  210
   10 A4 = (A+G)*F + H                                                NEW  220
      A5 = (F+U)*C + V*D                                              NEW  230
C EVALUATE NEW QUADRATIC COEFFICIENTS.                                NEW  240
   20 B1 = -K(N)/P(NN)                                                NEW  250
      B2 = -(K(N-1)+B1*P(N))/P(NN)                                    NEW  260
      C1 = V*B2*A1                                                    NEW  270
      C2 = B1*A7                                                      NEW  280
      C3 = B1*B1*A3                                                   NEW  290
      C4 = C1 - C2 - C3                                               NEW  300
      TEMP = A5 + B1*A4 - C4                                          NEW  310
      IF (TEMP.EQ.0.D0) GO TO 30                                      NEW  320
      UU = U - (U*(C3+C2)+V*(B1*A1+B2*A7))/TEMP                       NEW  330
      VV = V*(1.+C4/TEMP)                                             NEW  340
      RETURN                                                          NEW  350
C IF TYPE=3 THE QUADRATIC IS ZEROED                                   NEW  360
   30 UU = 0.D0                                                       NEW  370
      VV = 0.D0                                                       NEW  380
      RETURN                                                          NEW  390
      END                                                             NEW  400


      SUBROUTINE QUADSD(NN, U, V, P, Q, A, B)                         QUA   10
C DIVIDES P BY THE QUADRATIC  1,U,V  PLACING THE                      QUA   20
C QUOTIENT IN Q AND THE REMAINDER IN A,B                              QUA   30
      DOUBLE PRECISION P(NN), Q(NN), U, V, A, B, C                    QUA   40
      INTEGER I                                                       QUA   50
      B = P(1)                                                        QUA   60
      Q(1) = B                                                        QUA   70
      A = P(2) - U*B                                                  QUA   80
      Q(2) = A                                                        QUA   90
      DO 10 I=3,NN                                                    QUA  100
        C = P(I) - U*A - V*B                                          QUA  110
        Q(I) = C                                                      QUA  120
        B = A                                                         QUA  130
        A = C                                                         QUA  140
   10 CONTINUE                                                        QUA  150
      RETURN                                                          QUA  160
      END                                                             QUA  170
```

```
      SUBROUTINE QUAD(A, B1, C, SR, SI, LR, LI)                    QUA   10
C CALCULATE THE ZEROS OF THE QUADRATIC A*Z**2+B1*Z+C.              QUA   20
C THE QUADRATIC FORMULA, MODIFIED TO AVOID                         QUA   30
C OVERFLOW, IS USED TO FIND THE LARGER ZERO IF THE                 QUA   40
C ZEROS ARE REAL AND BOTH ZEROS ARE COMPLEX.                       QUA   50
C THE SMALLER REAL ZERO IS FOUND DIRECTLY FROM THE                 QUA   60
C PRODUCT OF THE ZEROS C/A.                                        QUA   70
      DOUBLE PRECISION A, B1, C, SR, SI, LR, LI, B,                QUA   80
     * D, E, DABS, DSQRT                                          QUA   90
      IF (A.NE.0.D0) GO TO 20                                      QUA  100
      SR = 0.D0                                                    QUA  110
      IF (B1.NE.0.D0) SR = -C/B1                                   QUA  120
      LR = 0.D0                                                    QUA  130
   10 SI = 0.D0                                                    QUA  140
      LI = 0.D0                                                    QUA  150
      RETURN                                                       QUA  160
   20 IF (C.NE.0.D0) GO TO 30                                      QUA  170
      SR = 0.D0                                                    QUA  180
      LR = -B1/A                                                   QUA  190
      GO TO 10                                                     QUA  200
C COMPUTE DISCRIMINANT AVOIDING OVERFLOW                           QUA  210
   30 B = B1/2.D0                                                  QUA  220
      IF (DABS(B).LT.DABS(C)) GO TO 40                             QUA  230
      E = 1.D0 - (A/B)*(C/B)                                       QUA  240
      D = DSQRT(DABS(E))*DABS(B)                                   QUA  250
      GO TO 50                                                     QUA  260
   40 E = A                                                        QUA  270
      IF (C.LT.0.D0) E = -A                                        QUA  280
      E = B*(B/DABS(C)) - E                                        QUA  290
      D = DSQRT(DABS(E))*DSQRT(DABS(C))                            QUA  300
   50 IF (E.LT.0.D0) GO TO 60                                      QUA  310
C REAL ZEROS                                                       QUA  320
      IF (B.GE.0.D0) D = -D                                        QUA  330
      LR = (-B+D)/A                                                QUA  340
      SR = 0.D0                                                    QUA  350
      IF (LR.NE.0.D0) SR = (C/LR)/A                                QUA  360
      GO TO 10                                                     QUA  370
C COMPLEX CONJUGATE ZEROS                                          QUA  380
   60 SR = -B/A                                                    QUA  390
      LR = SR                                                      QUA  400
      SI = DABS(D/A)                                               QUA  410
      LI = -SI                                                     QUA  420
      RETURN                                                       QUA  430
      END                                                          QUA  440
```