AD 676108

BRL TN 1699

BRL

AD

TECHNICAL NOTE NO. 1699

NUMERICAL SOLUTIONS OF POLYNOMIAL EQUATIONS

by

Tadeusz Leser
Henry Wisniewski

August 1968

This document has been approved for public release and sale; its distribution is unlimited.

U.S. ARMY ABERDEEN RESEARCH AND DEVELOPMENT CENTER
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

CLEARINGHOUSE

Destroy this report when it is no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

BALLISTIC RESEARCH LABORATORIES

TECHNICAL NOTE NO. 1699

AUGUST 1968

NUMERICAL SOLUTIONS OF POLYNOMIAL EQUATIONS

Tadeusz Leser
Henry Wisniewski

Applied Mathematics Division

This document has been approved for public release and sale;
its distribution is unlimited.

RDT&E Project No. 1T061102A14B

ABERDEEN PROVING GROUND, MARYLAND

B A L L I S T I C R E S E A R C H L A B O R A T O R I E S

TECHNICAL NOTE NO. 1699

TLeser/HWisniewski/blk
Aberdeen Proving Ground, Md.
August 1968

NUMERICAL SOLUTIONS OF POLYNOMIAL EQUATIONS

ABSTRACT

This report describes the FORTRAN Subroutine POLYR and a related complete program BRL-RSSR for finding all roots (real and complex) of a real polynomial equation

$$P(x) = \sum_{i=0}^{N} A_i x^{N-i} = 0.$$

The method which is used combines the root squaring and the subresultant (extracting quadratic factors) procedures for calculating all roots, even multiple roots, of real polynomials. Reconstruction of the coefficients from the product

$$\prod_{i=1}^{N} (x - \mathrm{Root}_i),$$

which is included, serves as a means of checking.

The described subroutine and program were adapted from a similar program in double precision described in Report ANL 6987, Argonne National Laboratory by Erwin H. Bareiss and Ronald Hamelink.

ROOTS OF POLYNOMIALS

Subroutine POLYR and BRL-RSSR Program

General Description. A FORTRAN subroutine and a related complete program for finding all roots (real and complex) of a real polynomial equation

$$P(x) = \sum_{i=0}^{N} A_i x^{N-i} = 0$$

are available for use on the BRLESC computer.

Method. The root squaring and subresultant (extracting quadratic factors) procedure "RSSR" for finding real, complex and repeated roots of real polynomials and checking them by reconstructing the coefficients from the product

$$\prod_{i=1}^{N} (x-ROOT_i)$$

is described in detail in

Report ANL 6987
Argonne National Laboratory
by Edwin H. Bareiss and Ronald Hamelink .

Authors. The routine under the name RSSR was written in FORTRAN by the above authors for the IBM-704 and CDC-3600 computers using double precision arithmetic. The modifications and additions necessary to make it run on BRLESC in single precision were performed by Henry Wisniewski of the Computing Laboratory of BRL.

Modifications to the Subroutine POLYR. When compared with the original FORTRAN ANL subroutine (pp 49-73) the BRL subroutine POLYR differs as follows:

a) The Main Program, which does input and output and calls RSSR, has been entirely replaced (see tabulation).

b) Subroutine RSSR

No changes.

c) Subroutine ROOTSQ

Changes:

LINE	STATEMENT
1+6	IF (KM) 11, 4, 11
11	DO 4 L=1, K M

d) Subroutine REALROOT

Changes:

LINE	STATEMENT
11+1	T=2**M*I

Reason: Mixed type arithmetic is not allowed in BRLESC FORTRAN.

e) Subroutine COMPROOT

No changes.

f) Subroutine TEST

Changes:

LINE	STATEMENT
11	IF (IB(3) - 74) 100, 100, 12
100	IF (IB(3) + 74) 12, 101, 101
18	IF (IB(2) - 74) 102, 102, 12
103	IF (IB(2) + 74) 12, 103, 103
103+1	IF (IB(3) - 74) 104, 104, 12
104	IF (IB(3) + 74) 12, 105, 105

g, h, i) Subroutines SUBRES, RECON, QUADRIV

No changes.

j) Subroutine DOUBLOG

Changes:

LINE	STATEMENT
1	PRINT 2
3	To = Ix
3+1	y = ALOG(x) + To*ALOG(T)

Reason: Mixed type arithmetic is not allowed.

k) Subroutine DOUBLEXP

Changes:

LINE	STATEMENT
First	Z = EXP(X) $IZ = 0

l) Subroutine ADD

Changes:

LINE	STATEMENT
10	IF (IDIFF) 11, 12, 11
12	D0 11 I = 1, IDIFF

m, n) Subroutines SUBTRACT and SCALE

No changes.

o) Subroutine UNSCALE

Changes:

LINE	STATEMENT
First + 1	IF(IX + 84) 3, 4, 4
4	IF(IX - 84) 5, 5, 6

Reason: Maximum size of IX on BRLESC is 84

6	X = 1.0E + 153

Reason: Maximum number on BRLESC is 1.34078E154

6+2	PRINT 7
7	FORMAT (25H0EXP. OVERFLOW IN UNSCALE)

<u>Operating Instructions for the Subroutine POLYR</u>. They are replaced by:

Entry is made by the statement

CALL POLYR (N, COEFF, ROOTS, D)

where

N = Degree of the polynomial. (A fixed point integer.)

COEFF = One-dimensional array of length (N+1) which contains the locations of the coefficients.

ROOTS = Two-dimensional array of length (2xN) which contains the roots.

D = One-dimensional array of length (N+1) which contains the reconstructed polynomial from the roots and the first coefficients.

The coefficient of the highest degree is stored first, followed by the remaining coefficients in decreasing order of degree.

Storage	
COEFF (1)	A_o
COEFF (2)	A_1
COEFF (3)	A_2
...	...
COEFF (N+1)	A_N

The results are:

ROOTS (1, 1)	Real Part	ROOTS (2, 1)	Imaginary Part
ROOTS (1, 2)	Real Part	ROOTS (2, 2)	Imaginary Part
ROOTS (1, 3)	Real Part	ROOTS (2, 3)	Imaginary Part
... ...	... ...	... ...	... ...
ROOTS (1, N)	Real Part	ROOTS (2, N)	Imaginary Part

The coefficients of the reconstructed polynomial.

D(1)	A_0
D(2)	A_1
D(3)	A_2
...	...
D(N+1)	A_N

This subroutine is available on cards only. (See Appendix A)

Modifications to the Program RSSR. The BRL-RSSR Program contains the same modifications as POLYR with the exception of a Main Program which is retained as shown in Report ANL 6987 with the following changes:

LINE	STATEMENT
2+1 and 3	Cancelled
4	PRINT 2, TITLE
9+3	Cancelled
10+1	"
14+1 through 18	"
19+2	"
30+2	END
31 through 37+2	Cancelled

Operating Instructions for the BRL-RSSR Program. As in Report ANL 6987, p. 73, with the following changes:

a. No change

b. No change

c. Cancelled

d. The first line is changed as follows:

A_i. Format (E 18.10)

The remaining lines are not changed. (See Appendix B)

Example

The input and output of the BRLESC computations of the roots of the very badly conditioned polynomial

$$P(x) = 7056.x^6 - 51972.x^5 + 22559.36x^4 + 661360.64x^3 - 2101084.2x^2 + 2490368.x - 1048576. = 0$$

are given in Appendix C.

Availability. Copies of the punched card decks for POLYR and RSSR may be obtained from Systems Programming Branch, Computer Support Division.

BIBLIOGRAPHY

1. Bareiss, E.H. and Hamelink, R. , RSSR Routine, A Root-Squaring and Subresultant Procedure for Finding Zeros of Real Polynomials, Argonne National Laboratory - 6987, 1965.

2. Bareiss, E.H., The Numerical Solution of Polynomial Equations and the Resultant Procedures , Mathematical Methods for Digital Computers, Vol. 2, 1967.

APPENDIX A

```
      SUBROUTINE POLYR(N,COEFF,ROOTS,D)
      DIMENSION A(51,3),IA(51,3),ROOTS(2,N ),D(1),COEFF(1)
      TYPE INTEGER DEGREE
      DEGREE=N
      N1=DEGREE+1
      M=10
      MMAX=15
      DELTA=0.0001
      EPSILON=0.000001
      DO 14 I=1,N1
      A(I,1)=COEFF(I)
      IA(I,1)=0
      CALL SCALE(A(I,1),IA(I,1))
   14 CONTINUE
      CALL RSSR(A,IA,ROOTS,DEGREE,M,MMAX,DELTA,EPSILON,D)
      IF(N1-(DEGREE+1))29,29,26
   26 RETURN
   29 PRINT 30
   30 FORMAT(21HOSOME ROOTS NOT FOUND)
      RETURN
      END

      SUBROUTINE RSSR (A,IA,ROOTS,DEGREE,M,MMAX,DELTA,EPSILON,D)
      DIMENSION A(51,3),IA(51,3),ROOTS(2,50),D(51),ROMOD(50),MROMOD(50),
     1NONRT(50),MNONRT(50)
      TYPE INTEGER DEGREE
      N=DEGREE
      IF (N) 1,1,3
    1 DEGREE=NCUR
    2 RETURN
    3 N1=N+1
      N2=N1+1
      DO 5 I=1,N
      K=N2-I
      IF (A(K,1)) 6,4,6
    4 J=N1-I
      ROOTS(1,J)=0.0
      ROOTS(2,J)=0.0
    5 CONTINUE
      DEGREE=0
      GO TO 2
    6 N1=K
      N=K-1
      NCUR=N
      NL=N
    7 CALL ROOTSQ (A,IA,NCUR,M)
      CALL REALROOT (A,IA,M,NCUR,DELTA,EPSILON,ROMOD,MROMOD,
     1NONRT,MNONRT,NCO,ROOTS)
      IF (NCO) 12,12,8
    8 N1=NCUR+1
      CALL COMPROOT (A,IA,ROMOD,ROOTS,M,MNONRT,NONRT,
     1MROMOD,NCO,DELTA,EPSILON,NCUR)
      IF (NCUR) 12,12,9
    9 IF (NL-NCUR) 11,11,10
   10 NL=NCUR
      GO TO 7
   11 M=M+1
      IF (MMAX-M) 1,7,7
   12 CALL RECON (ROOTS,A(1,1),IA(1,1),D,DEGREE)
      GO TO 1
```

```
      END

      SUBROUTINE ROOTSQ (A,IA,NCUR,MM)
      DIMENSION A(51,3),IA(51,3)
      N1=NCUR+1
      DO 1 J=1,N1
      A(J,2)=A(J,1)
      IA(J,2)=IA(J,1)
      A(J,3)=0.0
      IA(J,3)=0
    1 CONTINUE
      DO 9 M=1,MM
      DO 6 J=1,N1
      K1=N1-J
      K2=J-1
      KM=XMINOF(K1,K2)
      IF(KM)  11,4,11
   11 DO 4 L=1,KM
      LR=XMODF(L,2)
      JL=J-L
      JLP=J+L
      IF (LR) 2,2,3
    2 X=A(JL,2)*A(JLP,2)
      IX=IA(JL,2)+IA(JLP,2)
      CALL SCALE (X,IX)
      CALL ADD (A(J,3),IA(J,3),X,IX,A(J,3),IA(J,3))
      GO TO 4
    3 X=A(JL,2)*A(JLP,2)
      IX=IA(JL,2)+IA(JLP,2)
      CALL SCALE (X,IX)
      CALL SUBTRACT (A(J,3),IA(J,3),X,IX,A(J,3),IA(J,3))
    4 CONTINUE
      A(J,3)=2.0*A(J,3)
      CALL SCALE (A(J,3),IA(J,3))
      X=A(J,2)**2
      IX=IA(J,2)+IA(J,2)
      CALL SCALE (X,IX)
      CALL ADD (A(J,3),IA(J,3),X,IX,A(J,3),IA(J,3))
      JR=XMODF (J,2)
      IF (JR) 5,5,6
    5 A(J,3)=-A(J,3)
    6 CONTINUE
      IF (MM-M) 9,9,7
    7 DO 8 J=1,N1
      A(J,2)=A(J,3)
      IA(J,2)=IA(J,3)
      A(J,3)=0.0
      IA(J,3)=0
    8 CONTINUE
    9 CONTINUE
      RETURN
      END

      SUBROUTINE REALROOT (A,IA,M,NCUR,DELTA,EPSILON,ROMOD,MROMOD,
     1NONRT,MNONRT,NCO,ROOTS)
      DIMENSION A(51,3),IA(51,3),ROOTS(2,50),ROMOD(50),MROMOD(50),
     1NONRT(50),MNONRT(50),RATIO(51),IPIV(51),ARED(50),IARED(50)
      RATIO(1)=1.0
      DO 4 I=2,NCUR
      I1=XMODF(I,2)
```

```
      IF (A(I,3)) 2,1,2
    1 RATIO(I)=0.0
      GO TO 4
    2 T=A(I,2)*A(I,2)
      IT=IA(I,2)+IA(I,2)
      CALL SCALE (T,IT)
      T=T/A(I,3)
      IT=IT-IA(I,3)
      IF (IT-2) 50,50,1
   50 IF (IT+2) 1,51,51
   51 CALL UNSCALE (T,IT)
      RATIO(I)=T
      IF (I1) 3,3,4
    3 RATIO(I)=-RATIO(I)
    4 CONTINUE
      RATIO(NCUR+1)=1.0
      IPIV(1)=1
      IPIV(NCUR+1)=1
      DO 7 I=2,NCUR
      X=ABSF(RATIO(I)-1.0)
      IF (X-DELTA) 5,6,6
    5 IPIV(I)=1
      GO TO 7
    6 IPIV(I)=0
    7 CONTINUE
      NCUR1=NCUR+1
      I1=0
      MULT=0
      I=1
      I4=1
    8 I1=I1+1
      I2=I1+I
      MULT=MULT+1
      IF (IPIV(I2)) 8,8,9
    9 ROMOD(I4)=A(I2,3)/A(I,3)
      IROMOD=IA(I2,3)-IA(I,3)
      CALL SCALE (ROMOD(I4),IROMOD)
      IF (ROMOD(I4)) 10,11,11
   10 ROMOD(I4)=-ROMOD(I4)
   11 CALL DOUBLOG (ROMOD(I4),IROMOD,XN,IXN)
      T=2  **M*I1
      XN=XN/T
      CALL SCALE (XN,IXN)
      CALL DOUBLEXP (XN,IXN,ROMOD(I4),IROMOD)
      IF(IROMOD-74 ) 100,100,101
  100 IF(IROMOD+74 ) 101,101,102
  101 ROMOD(I4)=0.0
      IROMOD=0
      GO TO 103
  102 CALL UNSCALE (ROMOD(I4),IROMOD)
  103 MROMOD(I4)=MULT
      IF (NCUR+1-I2) 13,13,12
   12 I=I2
      I4=I4+1
      MULT=0
      I1=0
      GO TO 8
   13 Q=0.0
      NCO=0
      DO 22 I=1,I4
```

```
      KL=I4+1-I
      W=-ROMOD(KL)
      I5=MROMOD(KL)
      DO 20 J=1,I5
      J=J
   14 CALL TEST (A,IA,W,Q,NCUR,ROMOD(KL),EPSILON,K)
      IF (K) 17,17,15
   15 ROOTS(1,NCUR)=-W
      ROOTS(2,NCUR)=0.0
      ARED(1)=A(1,1)
      IARED(1)=IA(1,1)
      DO 16 L=2,NCUR
      Y=ARED(L-1)*W
      IY=IARED(L-1)
      CALL SCALE (Y,IY)
      CALL SUBTRACT (A(L,1),IA(L,1),Y,IY,ARED(L),IARED(L))
      A(L,1)=ARED(L)
      IA(L,1)=IARED(L)
   16 CONTINUE
      GO TO 19
   17 IF (W) 18,21,21
   18 W=-W
      GO TO 14
   19 NCUR=NCUR-1
   20 CONTINUE
      GO TO 22
   21 NCO=NCO+1
      NONRT(NCO)=KL
      MNONRT(NCO)=I5+1-J
   22 CONTINUE
      RETURN
      END

      SUBROUTINE COMPROOT (A,IA,ROMOD,ROOTS,M,MNONRT,NONRT,
     1MROMOD,NCO,DELTA,EPSILON,NCUR)
      DIMENSION A(51,3),IA(51,3),ROMOD(50),ROOTS(2,50),SR(51,3),ISR(51,3
     1),SROMOD(50),SROOTS(2,50),MNONRT(50),NONRT(50),MSROMOD(50),
     2NSONRT(49),MSNORT(49),MROMOD(50),D(2),R(2),B(49)
      DO 23 I=1,NCO
      JA=NONRT(I)
      I1=MNONRT(I)
      I1=I1/2
      IF (I1) 1,1,2
    1 I1=1
    2 IF (ROMOD(JA)) 3,23,3
    3 Q=ROMOD(JA)
      DO 22 J=1,I1
  300 CALL SUBRES (A,IA,NCUR,SR,ISR,Q)
      IF (NCUR-4) 4,100,5
  100 NSCUR=2
      GO TO 101
    4 NSCUR=1
      J2=1
      GO TO 6
    5 NSCUR=NCUR-3
  101 J2=NSCUR
    6 LL=NSCUR+1
      IF (NSCUR-1) 7,7,9
    7 IF (SR(1,1)) 8,10,8
    8 X=SR(1,1) $ IX=ISR(1,1)
```

```
      Y=SR(2,1) $ IY=ISR(2,1)
      CALL UNSCALE (X,IX)
      CALL UNSCALE (Y,IY)
      SROOTS(1,1)=-Y/X
      NSCUR=0
      GO TO 11
    9 CALL ROOTSQ (SR,ISR,NSCUR,M)
      CALL REALROOT (SR,ISR,M,NSCUR,DELTA,EPSILON,SROMOD,MSROMOD,
     1NSONRT,MSNORT,NSCO,SROOTS)
      IF (J2-NSCUR) 10,10,11
   10 SROOTS(1,J2)=0.0
   11 SROOTS(1,J2)=SROOTS(1,J2)*ROMOD(JA)
      T=ROMOD(JA)+ROMOD(JA)
      IF (SROOTS(1,J2)-T) 12,16,16
   12 W=SROOTS(1,J2)
      WE=ROMOD(JA)*ROMOD(JA)
      CALL TEST (A,IA,W,WE,NCUR,ROMOD(JA),EPSILON,K)
      IF (K) 15,15,13
   13 ROOTS(1,NCUR)=-W/2.0
      T=4.0*WE
      U=W*W
      T=T-U
      IF (T) 201,201,202
  201 T=-T
      U=DSQRT (T)
      ROOTS(1,NCUR)=ROOTS(1,NCUR)-U/2.0
      ROOTS(1,NCUR-1)=-(W-U)/2.0
      ROOTS(2,NCUR)=0.0
      ROOTS(2,NCUR-1)=0.0
      GO TO 204
  202 U=DSQRT(T)
  203 ROOTS(2,NCUR)=U/2.0
      ROOTS(1,NCUR-1)=ROOTS(1,NCUR)
      ROOTS(2,NCUR-1)=-ROOTS(2,NCUR)
  204 D(1)=W
      D(2)=WE
      CALL QUADIV (NCUR,A,IA,R,D,B)
      JX=NCUR-1
      DO 14 JY=1,JX
      A(JY,1)=B(JY)
      IA(JY,1)=0
      CALL SCALE (A(JY,1),IA(JY,1))
   14 CONTINUE
      NCUR=NCUR-2
      GO TO 22
   15 W=-W
      CALL TEST (A,IA,W,WE,NCUR,ROMOD(JA),EPSILON,K)
      IF (K) 16,16,13
   16 IF (J2-(NSCUR+1)) 23,17,19
   17 IF (J2-1) 23,23,18
   18 J2=J2-1
      SROOTS(1,J2)=0.0
      GO TO 12
   19 IF (SROOTS(1,J2)-SROOTS(1,J2-1)) 20,21,20
   20 J2=J2-1
      GO TO 11
   21 J2=J2-1
      GO TO 16
   22 CONTINUE
   23 CONTINUE
```

```
      RETURN
      END

      SUBROUTINE TEST (A,IA,W,Q,N,ROMOD,EPSILON,K)
      DIMENSION A(51,3),IA(51,3),B(3),IB(3),T(2),E(2),C(51)
      B(1)=0.0$IX=0$IW=0
      IB(1)=0
      B(2)=A(1,1)
      IB(2)=IA(1,1)
      DO 2 I=1,N
      X=W*B(2)
      IX=IB(2)
      CALL SCALE (X,IX)
      Y=Q*B(1)
      IY=IB(1)
      CALL SCALE (Y,IY)
      CALL ADD (X,IX,Y,IY,Z,IZ)
      CALL SUBTRACT (A(I+1,1),IA(I+1,1),Z,IZ,B(3),IB(3))
      IF (N-I) 2,2,1
    1 B(1)=B(2)
      IB(1)=IB(2)
      B(2)=B(3)
      IB(2)=IB(3)
    2 CONTINUE
    3 KOUNT=1
      CEPSIL=EPSILON
      T(1)=0.0$T(2)=0.0
      N1=N+1
      X=2.0*EPSILON
      Y=X*ROMOD
      E(1)=ROMOD+Y
      E(2)=ROMOD+CEPSIL*ROMOD
      DO 7 I=1,N1
      IF (A(I,1)) 4,5,5
    4 C(I)=-A(I,1)$IC=IA(I,1)
      GO TO 6
    5 C(I)=A(I,1)$IC=IA(I,1)
    6 CALL UNSCALE (C(I),IC)
      T(1)=T(1)*E(1)+C(I)
      T(2)=T(2)*E(2)+C(I)
    7 CONTINUE
      DIF=T(1)-T(2)
    8 IF (Q) 18,9,18
    9 IF (B(3)) 10,11,11
   10 B(3)=-B(3)
   11 IF (IB(3)-74 ) 100,100,12
  100 IF (IB(3)+74 ) 12,101,101
  101 CALL UNSCALE (B(3),IB(3))
      IF (DIF-B(3)) 12,12,17
   12 K=0
      IF (KOUNT-2) 13,16,16
   13 IF (Q) 15,14,15
   14 IF (W) 15,16,16
   15 SENSE LIGHT 2
      KOUNT=KOUNT+1
   16 RETURN
   17 K=1
      GO TO 16
   18 IF (IB(2)-74 ) 102,102,12
  102 IF (IB(2)+74 ) 12,103,103
```

```
103 CALL UNSCALE (B(2),IB(2))
    IF (IB(3)-74 ) 104,104,12
104 IF (IB(3)+74 ) 12,105,105
105 CALL UNSCALE (B(3),IB(3))
    X=Q*B(2)*B(2)
    Y=W*B(2)*B(3)
    Z=B(3)*B(3)
    Y=X-Y+Z
    IF (Y) 19,17,20
 19 Y=-Y
 20 DIF=DIF*DIF
    IF (DIF-Y) 12,17,17
    END

    SUBROUTINE SUBRES (A,IA,N,SR,ISR,ROMOD)
    DIMENSION A(51,3),IA(51,3),SR(51,3),ISR(51,3),C(51),B(50,3)
    N1=N+1
    T=1.0
    DO 1 I=1,N
    J=N1-I
    T=T*ROMOD
    C(J)=A(J,1)*T
    IC=IA(J,1)
    CALL UNSCALE (C(J),IC)
  1 CONTINUE
    C(N1)=A(N1,1)
    IC=IA(N1,1)
    CALL UNSCALE (C(N1),IC)
    IF (N-2) 12,12,2
  2 N2=N-2
    DO 3 I=1,N2
    B(I,1)=0.0
    B(I,2)=0.0
  3 CONTINUE
    I=2
    B(1,2)=C(1)
  4 B(1,3)=C(I)-C(1,1)
    DO 5 J=2,N2
    B(J,3)=-B(J-1,2)-B(J,1)
  5 CONTINUE
    IF (N-(3+I)) 100,6,6
  6 I=I+1
    DO 7 J=1,N2
    B(J,1)=B(J,2)
    B(J,2)=B(J,3)
  7 CONTINUE
    GO TO 4
100 IF (N-4)  14,101,8
101 IF (N-(2+I)) 104,102,102
102 I=I+1
    DO 103 J=1,2
    B(J,1)=B(J,2)
    B(J,2)=B(J,3)
103 CONTINUE
    GO TO 4
104 B(3,3)=-B(2,2)
    SR(3,1)=-C(5)+B(1,3) $ ISR(3,1)=0
    SR(2,1)=B(2,3) $ ISR(2,1)=0
    SR(1,1)=B(3,3) $ ISR(1,1)=0
    CALL SCALE  (SR(1,1),ISR(1,1))
```

```
      CALL SCALE  (SR(2,1),ISR(2,1))
      CALL SCALE  (SR(3,1),ISR(3,1))
      GO TO 11
    8 SR(N2,1)=C(N)-B(1,3) $ ISR(N2,1)=0
      SR(N2-1,1)=-C(N1)-B(2,3) $ ISR(N2-1,1)=0
      CALL SCALE (SR(N2,1),ISR(N2,1))
      CALL SCALE (SR(N2-1,1),ISR(N2-1,1))
      IF (N2-2) 11,11,9
    9 DO 10 J=3,N2
      K=N2+1-J
      SR(K,1)=-B(J,3) $ ISR(K,1)=0
      CALL SCALE (SR(K,1),ISR(K,1))
   10 CONTINUE
   11 RETURN
   12 SR(1,1)=C(1) $ ISR(1,1)=0
      SR(2,1)=-C(2) $ ISR(2,1)=0
   13 CALL SCALE (SR(1,1),ISR(1,1))
      CALL SCALE (SR(2,1),ISR(2,1))
      GO TO 11
   14 SR(1,1)=-C(4)
      SR(2,1)=C(3)-C(1)
      ISR(1,1)=0
      ISR(2,1)=0
      GO TO 13
      END

      SUBROUTINE RECON (ROOTS,A,IA,D,N)
      DIMENSION ROOTS(2,50),D(51)
      X=A$IX=IA
      CALL UNSCALE (X,IX)
      DO 1 I=1,N
      D(I)=0.0
    1 CONTINUE
      D(N+1)=1.0
      I=1
      NL=N-1
    2 IF (ROOTS(2,I)) 3,7,3
    3 T=ROOTS(1,I)*ROOTS(1,I)
      U=ROOTS(2,I)*ROOTS(2,I)
      T=T+U
      U=2.0*ROOTS(1,I)
      DO 5 J=1,NL
      IF (I+J-N) 5,4,4
    4 D(J)=D(J+2)+T*D(J)
      D(J)=D(J)-U*D(J+1)
    5 CONTINUE
      D(N)=T*D(N)
      D(N)=D(N)-U*D(N+1)
      D(N+1)=T*D(N+1)
      I=I+2
    6 IF (N-I) 10,2,2
    7 DO 9 J=1,N
      IF (J+I-N) 9,9,8
    8 D(J)=D(J+1)-D(J)*ROOTS(1,I)
    9 CONTINUE
      D(N+1)=-D(N+1)*ROOTS(1,I)
      I=I+1
      GO TO 6
   10 NS=N+1
      DO 11 II=1,NS
```

```
      D(II)=D(II)*X
   11 CONTINUE
      RETURN
      END

      SUBROUTINE QUADIV (N,A,IA,R,D,B)
      DIMENSION A(51,3),IA(51,3),R(2),D(2),B(4))
      B(1)=A(1,1)$IB=IA(1,1)
      CALL UNSCALE (B(1),IB)
      IF (N-2) 4,4,1
    1 AA=A(2,1)$IAA=IA(2,1)
      CALL UNSCALE (AA,IAA)
      B(2)=AA-B(1)*D(1)
      IF (N-3) 4,4,2
    2 NT=N-1
      DO 3 I=3,NT
      XN=B(I-1)*D(1)
      YN=B(I-2)*D(2)
      AA=A(I,1)$IAA=IA(I,1)
      CALL UNSCALE (AA,IAA)
      B(I)=AA-(XN+YN)
    3 CONTINUE
    4 XN=B(N-1)*D(1)
      YN=B(N-2)*D(2)
      AA=A(N,1)$IAA=IA(N,1)
      CALL UNSCALE (AA,IAA)
      R(1)=AA-(XN+YN)
      AA=A(N+1,1)$IAA=IA(N+1,1)
      CALL UNSCALE (AA,IAA)
      R(2)=AA-B(N-1)*D(2)
      RETURN
      END

      SUBROUTINE DOUBLOG (X,IX,Y,IY)
      T=64.0
      IF (X)1,5,3
    1 PRINT  2
    2 FORMAT (46HOTHE LOG OF A NON-POSITIVE NUMBER IS REQUESTED)
    5 Y=0.0
      IY=0
      GO TO 4
3     TO=IX$
      Y=ALOG(X)+TO*ALOG(T)
      IY=0
      CALL SCALE (Y,IY)
    4 RETURN
      END

      SUBROUTINE DOUBLEXP (X,IX,Z,IZ)
      Z= EXP(X)$IZ=0
      IF (IX) 1,9,6
    1 I=-IX$II=6*I
      DO 4 J=1,II
      K=XMODF(IZ,2)
      IF (K) 2,3,2
    2 IZ=IZ-1
      Z=64.0*Z
    3 IZ=IZ/2
      Z=DSQRT(Z)
      CALL SCALE (Z,IZ)
```

```
 4 CONTINUE
 5 RETURN
 6 I=6*IX
   DO 7 J=1,I
   Z=Z*Z$IZ=IZ+IZ
   CALL SCALE (Z,IZ)
 7 CONTINUE
   GO TO 5
 8 CALL SCALE (Z,IZ)
   GO TO 5
   END

   SUBROUTINE ADD (X,IX,Y,IY,Z,IZ)
   IF (X) 3,1,3
 1 Z=Y
   IZ=IY
 2 RETURN
 3 IF (Y) 5,4,5
 4 Z=X
   IZ=IX
   GO TO 2
 5 IDIFF=IX-IY
   IF (IDIFF) 6,7,7
 6 IA=IY
   A=Y
   B=X
   IDIFF=-IDIFF
   GO TO 8
 7 IA=IX
   A=X
   B=Y
 8 IF (16-IDIFF) 9,9,10
 9 Z=A
   IZ=IA
   GO TO 2
10 IF(IDIFF) 11,12,11
11 DO 12 I=1,IDIFF
   B=B/64.0
12 CONTINUE
   Z=A+B
   IZ=IA
   CALL SCALE (Z,IZ)
   GO TO 2
   END

   SUBROUTINE SUBTRACT (X,IX,Y,IY,Z,IZ)
   W=-Y
   CALL ADD (X,IX,W,IY,Z,IZ)
   RETURN
   END

   SUBROUTINE SCALE (X,IX)
   TYPE DOUBLE X,Y
   REC64=1.0/64.0
   IF (X) 1,11,2
 1 Y=-X
   GO TO 3
 2 Y=X
 3 IF (64.0-Y) 4,5,5
 4 Y=Y/64.0
```

```
      IX=IX+1
      GO TO 3
    5 IF (Y-REC64) 6,7,7
    6 Y=Y*64.0
      IX=IX-1
      GO TO 5
    7 IF (X) 8,9,9
    8 X=-Y
      GO TO 10
    9 X=Y
   10 RETURN
   11 IX=0
      GO TO 10
      END

      SUBROUTINE UNSCALE (X,IX)
      IF (IX+84 ) 3,4,4
    3 X=0.0
      IX=0
      GO TO 2
    4 IF (IX-84 ) 5,5,6
    6 X=1.0E+153
      IX=0
      PRINT    7
    7 FORMAT (25H0EXP. OVERFLOW IN UNSCALE)
      GO TO 2
    5 IF (IX) 1,2,1
    1 X=X*64.0**IX
    8 IX=0
    2 RETURN
      END
E
```

APPENDIX B

```
*     C-907  FINDING ZEROS OF REAL POLYNOMIALS
C     MAIN PROGRAM
C     DOES INPUT AND OUTPUT, AND CALLS RSSR
      DIMENSION A(51,3),IA(51,3),ROOTS(2,50),TITLE(10),D(51)
      TYPE INTEGER DEGREE
    1 READ 2, TITLE
    2 FORMAT (10A8)
    4 PRINT 2, TITLE
    5 FORMAT (1H1,10A8)
      READ     6, DEGREE,M,MMAX,DELTA,EPSILON
    6 FORMAT (3I6,2E12.4)
      IF (M) 7,7,8
    7 M=10
      MMAX=15
      DELTA=0.0001
      EPSILON=0.000001
    8 PRINT     9, DEGREE,M,MMAX,DELTA,EPSILON
    9 FORMAT (8H0DEGREE=I3,17H  NO. OF ROOT SQ=I3,7H  MMAX=I3,/
     17H DELTA=E12.3,10H  EPSILON=E12.3)
      N1=DEGREE+1
      PRINT  10
   10 FORMAT (19H0INPUT COEFFICIENTS)
   11 DO 14 I=1,N1
      READ     12, X
   12 FORMAT (E18.10)
      A(I,1)=X
      IA(I,1)=0
      J=I-1
      PRINT     13, J,X
   13 FORMAT (3HCA(I2,3H)= 1PE18.10)
      CALL SCALE (A(I,1),IA(I,1))
   14 CONTINUE
   19 N2=DEGREE
      CALL RSSR (A,IA,ROOTS,DEGREE,M,MMAX,DELTA,EPSILON,D)
   20 IF (N1-(DEGREE+1)) 29,29,21
   21 PRINT  22
   22 FORMAT (24HCROOTS OF THE POLYNOMIAL)
      PRINT  23
   23 FORMAT (42H0REAL PART                    IMAGINARY PART)
      K=DEGREE+1
      DO 25 I=K,N2
      X=ROOTS(1,I)
      Y=ROOTS(2,I)
      PRINT  24,X,Y
   24 FORMAT (1H01PE24.15,1PE26.15)
   25 CONTINUE
      IF (DEGREE) 26,26,29
   26 PRINT  27
   27 FORMAT (25H0RECONSTRUCTED POLYNOMIAL)
      DO 28 I=1,N1
      X=D(I)
      J=I-1
      PRINT     13, J,X
   28 CONTINUE
      GO TO 1
   29 PRINT  30
   30 FORMAT (21H0SOME ROOTS NOT FOUND)
      GO TO 1
      END
      SUBROUTINE RSSR (A,IA,ROOTS,DEGREE,M,MMAX,DELTA,EPSILON,D)
```

```
      DIMENSION A(51,3),IA(51,3),ROOTS(2,50),D(51),ROMOD(50),MROMOD(50),
     1NONRT(50),MNONRT(50)
      TYPE INTEGER DEGREE
      N=DEGREE
      IF (N) 1,1,3
    1 DEGREE=NCUR
    2 RETURN
    3 N1=N+1
      N2=N1+1
      DO 5 I=1,N
      K=N2-I
      IF (A(K,1)) 6,4,6
    4 J=N1-I
      ROOTS(1,J)=0.0
      ROOTS(2,J)=0.0
    5 CONTINUE
      DEGREE=0
      GO TO 2
    6 N1=K
      N=K-1
      NCUR=N
      NL=N
    7 CALL ROOTSQ (A,IA,NCUR,M)
      CALL REALROOT (A,IA,M,NCUR,DELTA,EPSILON,ROMOD,MROMOD,
     1NONRT,MNONRT,NCO,ROOTS)
      IF (NCO) 12,12,8
    8 N1=NCUR+1
      CALL COMPROOT (A,IA,ROMOD,ROOTS,M,MNONRT,NONRT,
     1MROMOD,NCO,DELTA,EPSILON,NCUR)
      IF (NCUR) 12,12,9
    9 IF (NL-NCUR) 11,11,10
   10 NL=NCUR
      GO TO 7
   11 M=M+1
      IF (MMAX-M) 1,7,7
   12 CALL RECON (ROOTS,A(1,1),IA(1,1),D,DEGREE)
      GO TO 1
      END

      SUBROUTINE ROOTSQ (A,IA,NCUR,MM)
      DIMENSION A(51,3),IA(51,3)
      N1=NCUR+1
      DO 1 J=1,N1
      A(J,2)=A(J,1)
      IA(J,2)=IA(J,1)
      A(J,3)=0.0
      IA(J,3)=0
    1 CONTINUE
      DO 9 M=1,MM
      DO 6 J=1,N1
      K1=N1-J
      K2=J-1
      KM=XMINOF(K1,K2)
      IF(KM)  11,4,11
   11 DO 4 L=1,KM
      LR=XMODF(L,2)
      JL=J-L
      JLP=J+L
      IF (LR) 2,2,3
    2 X=A(JL,2)*A(JLP,2)
```

```
      IX=IA(JL,2)+IA(JLP,2)
      CALL SCALE (X,IX)
      CALL ADD (A(J,3),IA(J,3),X,IX,A(J,3),IA(J,3))
      GO TO 4
    3 X=A(JL,2)*A(JLP,2)
      IX=IA(JL,2)+IA(JLP,2)
      CALL SCALE (X,IX)
      CALL SUBTRACT (A(J,3),IA(J,3),X,IX,A(J,3),IA(J,3))
    4 CONTINUE
      A(J,3)=2.0*A(J,3)
      CALL SCALE (A(J,3),IA(J,3))
      X=A(J,2)**2
      IX=IA(J,2)+IA(J,2)
      CALL SCALE (X,IX)
      CALL ADD (A(J,3),IA(J,3),X,IX,A(J,3),IA(J,3))
      JR=XMODF (J,2)
      IF (JR) 3,5,6
    5 A(J,3)=-A(J,3)
    6 CONTINUE
      IF (MM-M) 9,9,7
    7 DO 8 J=1,N1
      A(J,2)=A(J,3)
      IA(J,2)=IA(J,3)
      A(J,3)=0.0
      IA(J,3)=0
    8 CONTINUE
    9 CONTINUE
      RETURN
      END

      SUBROUTINE REALROOT (A,IA,M,NCUR,DELTA,EPSILON,ROMOD,MROMOD,
     1NONRT,MNONRT,NCO,ROOTS)
      DIMENSION A(51,3),IA(51,3),ROOTS(2,50),ROMOD(50),MROMOD(50),
     1NONRT(50),MNONRT(50),RATIO(51),IPIV(51),ARED(50),IARED(50)
      RATIO(1)=1.0
      DO 4 I=2,NCUR
      I1=XMODF(I,2)
      IF (A(I,3)) 2,1,2
    1 RATIO(I)=0.0
      GO TO 4
    2 T=A(I,2)*A(I,2)
      IT=IA(I,2)+IA(I,2)
      CALL SCALE (T,IT)
      T=T/A(I,3)
      IT=IT-IA(I,3)
      IF (IT-2) 50,50,1
   50 IF (IT+2) 1,51,51
   51 CALL UNSCALE (T,IT)
      RATIO(I)=T
      IF (I1) 3,3,4
    3 RATIO(I)=-RATIO(I)
    4 CONTINUE
      RATIO(NCUR+1)=1.0
      IPIV(1)=1
      IPIV(NCUR+1)=1
      DO 7 I=2,NCUR
      X=ABSF(RATIO(I)-1.0)
      IF (X-DELTA) 5,6,6
    5 IPIV(I)=1
      GO TO 7
```

```
  6 IPIV(I)=0
  7 CONTINUE
    NCUR1 NCUR+1
    I1=0
    MULT=0
    I=1
    I4=1
  8 I1=I1+1
    I2=I1+I
    MULT=MULT+1
    IF (IPIV(I2)) 8,8,9
  9 ROMOD(I4)=A(I2,3)/A(I,3)
    IROMOD=IA(I2,3)-IA(I,3)
    CALL SCALE (ROMOD(I4),IROMOD)
    IF (ROMOD(I4)) 10,11,11
 10 ROMOD(I4)=-ROMOD(I4)
 11 CALL DOUBLOG (ROMOD(I4),IROMOD,XN,IXN)
    T=2  **M*I1
    XN=XN/T
    CALL SCALE (XN,IXN)
    CALL DOUBLEXP (XN,IXN,ROMOD(I4),IROMOD)
    IF(IROMOD-74 ) 100,100,101
100 IF(IROMOD+74 ) 101,101,102
101 ROMOD(I4)=0.0
    IROMOD=0
    GO TO 103
102 CALL UNSCALE (ROMOD(I4),IROMOD)
103 MROMOD(I4)=MULT
    IF (NCUR+1-I2) 13,13,12
 12 I=I2
    I4=I4+1
    MULT=0
    I1=0
    GO TO 8
 13 Q=0.0
    NCO=0
    DO 22 I=1,I4
    KL=I4+1-I
    W=-ROMOD(KL)
    I5=MROMOD(KL)
    DO 20 J=1,I5
    J=J
 14 CALL TEST (A,IA,W,Q,NCUR,ROMOD(KL),EPSILON,K)
    IF (K) 17,17,15
 15 ROOTS(1,NCUR)=-W
    ROOTS(2,NCUR)=0.0
    ARED(1)=A(1,1)
    IARED(1)=IA(1,1)
    DO 16 L=2,NCUR
    Y=ARED(L-1)*W
    IY=IARED(L-1)
    CALL SCALE (Y,IY)
    CALL SUBTRACT (A(L,1),IA(L,1),Y,IY,ARED(L),IARED(L))
    A(L,1)=ARED(L)
    IA(L,1)=IARE (L)
 16 CONTINUE
    GO TO 19
 17 IF (W) 18,21,21
 18 W=-W
    GO TO 14
```

```
 19 NCUR=NCUR-1
 20 CONTINUE
    GO TO 22
 21 NCO=NCO+1
    NONRT(NCO)=KL
    MNONRT(NCO)=I5+1-J
 22 CONTINUE
    RETURN
    END

    SUBROUTINE COMPROOT (A,IA,ROMOD,ROOTS,M,MNONRT,NONRT,
   1MROMOD,NCO,DELTA,EPSILON,NCUR)
    DIMENSION A(51,3),IA(51,3),ROMOD(50),ROOTS(2,50),SR(51,3),ISR(51,3
   1),SROMOD(50),SROOTS(2,50),MNONRT(50),NONRT(50),MSROMOD(50),
   2NSONRT(49),MSNORT(49),MROMOD(50),D(2),R(2),B(49)
    DO 23 I=1,NCO
    JA=NONRT(I)
    I1=MNONRT(I)
    I1=I1/2
    IF (I1) 1,1,2
  1 I1=1
  2 IF (ROMOD(JA)) 3,23,3
  3 Q=ROMOD(JA)
    DO 22 J=1,I1
300 CALL SUBRES (A,IA,NCUR,SR,ISR,Q)
    IF (NCUR-4) 4,100,5
100 NSCUR=2
    GO TO 101
  4 NSCUR=1
    J2=1
    GO TO 6
  5 NSCUR=NCUR-3
101 J2=NSCUR
  6 LL=NSCUR+1
    IF (NSCUR-1) 7,7,9
  7 IF (SR(1,1)) 8,10,8
  8 X=SR(1,1) $ IX=ISR(1,1)
    Y=SR(2,1) $ IY=ISR(2,1)
    CALL UNSCALE (X,IX)
    CALL UNSCALE (Y,IY)
    SROOTS(1,1)=-Y/X
    NSCUR=0
    GO TO 11
  9 CALL ROOTSQ (SR,ISR,NSCUR,M)
    CALL REALROOT (SR,ISR,M,NSCUR,DELTA,EPSILON,SROMOD,MSROMOD,
   1NSONRT,MSNORT,NSCO,SROOTS)
    IF (J2-NSCUR) 10,10,11
 10 SROOTS(1,J2)=0.0
 11 SROOTS(1,J2)=SROOTS(1,J2)*ROMOD(JA)
    T=ROMOD(JA)+ROMOD(JA)
    IF (SROOTS(1,J2)-T) 12,16,16
 12 W=SROOTS(1,J2)
    WE=ROMOD(JA)*ROMOD(JA)
    CALL TEST (A,IA,W,WE,NCUR,ROMOD(JA),EPSILON,K)
    IF (K) 15,15,13
 13 ROOTS(1,NCUR)=-W/2.0
    T=4.0*WE
    U=W*W
    T=T-U
    IF (T) 201,201,202
```

```
201 T=-T
    U=DSQRT (T)
    ROOTS(1,NCUR)=ROOTS(1,NCUR)-U/2.0
    ROOTS(1,NCUR-1)=-(W-U)/2.0
    ROOTS(2,NCUR)=0.0
    ROOTS(2,NCUR-1)=0.0
    GO TO 204
202 U=DSQRT(T)
203 ROOTS(2,NCUR)=U/2.0
    ROOTS(1,NCUR-1)=ROOTS(1,NCUR)
    ROOTS(2,NCUR-1)=-ROOTS(2,NCUR)
204 D(1)=W
    D(2)=WE
    CALL QUADIV (NCUR,A,IA,R,D,B)
    JX=NCUR-1
    DO 14 JY=1,JX
    A(JY,1)=B(JY)
    IA(JY,1)=0
    CALL SCALE (A(JY,1),IA(JY,1))
 14 CONTINUE
    NCUR=NCUR-2
    GO TO 22
 15 W=-W
    CALL TEST (A,IA,W,WE,NCUR,ROMOD(JA),EPSILON,K)
    IF (K) 16,16,13
 16 IF (J2-(NSCUR+1)) 23,17,19
 17 IF (J2-1) 23,23,18
 18 J2=J2-1
    SROOTS(1,J2)=0.0
    GO TO 12
 19 IF (SROOTS(1,J2)-SROOTS(1,J2-1)) 20,21,20
 20 J2=J2-1
    GO TO 11
 21 J2=J2-1
    GO TO 16
 22 CONTINUE
 23 CONTINUE
    RETURN
    END

    SUBROUTINE TEST (A,IA,W,Q,N,ROMOD,EPSILON,K)
    DIMENSION A(51,3),IA(51,3),B(3),IB(3),T(2),E(2),C(51)
    B(1)=0.0$IX=0$IW=0
    IB(1)=0
    B(2)=A(1,1)
    IB(2)=IA(1,1)
    DO 2 I=1,N
    X=W*B(2)
    IX=IB(2)
    CALL SCALE (X,IX)
    Y=Q*B(1)
    IY=IB(1)
    CALL SCALE (Y,IY)
    CALL ADD (X,IX,Y,IY,Z,IZ)
    CALL SUBTRACT (A(I+1,1),IA(I+1,1),Z,IZ,B(3),IB(3))
    IF (N-I) 2,2,1
  1 B(1)=B(2)
    IB(1)=IB(2)
    B(2)=B(3)
    IB(2)=IB(3)
```

```
  2 CONTINUE
  3 KOUNT=1
    CEPSIL=EPSILON
    T(1)=0.0$T(2)=0.0
    N1=N+1
    X=2.0*EPSILON
    Y=X*ROMOD
    E(1)=ROMOD+Y
    E(2)=ROMOD+CEPSIL*ROMOD
    DO 7 I=1,N1
    IF (A(I,1)) 4,5,5
  4 C(I)=-A(I,1)$IC=IA(I,1)
    GO TO 6
  5 C(I)=A(I,1)$IC=IA(I,1)
  6 CALL UNSCALE (C(I),IC)
    T(1)=T(1)*E(1)+C(I)
    T(2)=T(2)*E(2)+C(I)
  7 CONTINUE
    DIF=T(1)-T(2)
  8 IF (Q) 18,9,18
  9 IF (B(3)) 10,11,11
 10 B(3)=-B(3)
 11 IF (IB(3)-74 ) 100,100,12
100 IF (IB(3)+74 ) 12,101,101
101 CALL UNSCALE (B(3),IB(3))
    IF (DIF-B(3)) 12,12,17
 12 K=0
    IF (KOUNT-2) 13,16,16
 13 IF (Q) 15,14,15
 14 IF (W) 15,16,16
 15 SENSE LIGHT 2
    KOUNT=KOUNT+1
 16 RETURN
 17 K=1
    GO TO 16
 18 IF (IB(2)-74 ) 102,102,12
102 IF (IB(2)+74 ) 12,103,103
103 CALL UNSCALE (B(2),IB(2))
    IF (IB(3)-74 ) 104,104,12
104 IF (IB(3)+74 ) 12,105,105
105 CALL UNSCALE (B(3),IB(3))
    X=Q*B(2)*B(2)
    Y=W*B(2)*B(3)
    Z=B(3)*B(3)
    Y=X-Y+Z
    IF (Y) 19,17,20
 19 Y=-Y
 20 DIF=DIF*DIF
    IF (DIF-Y) 12,17,17
    END

    SUBROUTINE SUBRES (A,IA,N,SR,ISR,ROMOD)
    DIMENSION A(51,3),IA(51,3),SR(51,3),ISR(51,3),C(51),B(50,3)
    N1=N+1
    T=1.0
    DO 1 I=2,N
    J=N1-I
    T=T*ROMOD
    C(J)=A(J,1)*T
    IC=IA(J,1)
```

```
      CALL UNSCALE (C(J),IC)
    1 CONTINUE
      C(N1)=A(N1,1)
      IC=IA(N1,1)
      CALL UNSCALE (C(N1),IC)
      IF (N-2) 12,12,2
    2 N2=N-2
      DO 3 I=1,N2
      B(I,1)=0.0
      B(I,2)=0.0
    3 CONTINUE
      I=2
      B(1,2)=C(1)
    4 B(1,3)=C(I)-B(1,1)
      DO 5 J=2,N2
      B(J,3)=-B(J-1,2)-B(J,1)
    5 CONTINUE
      IF (N-(3+I)) 100,6,6
    6 I=I+1
      DO 7 J=1,N2
      B(J,1)=B(J,2)
      B(J,2)=B(J,3)
    7 CONTINUE
      GO TO 4
  100 IF (N-4)  14,101,8
  101 IF (N-(2+I)) 104,102,102
  102 I=I+1
      DO 103 J=1,2
      B(J,1)=B(J,2)
      B(J,2)=B(J,3)
  103 CONTINUE
      GO TO 4
  104 B(3,3)=-B(2,2)
      SR(3,1)=-C(5)+B(1,3) $ ISR(3,1)=0
      SR(2,1)=B(2,3) $ ISR(2,1)=0
      SR(1,1)=B(3,3) $ ISR(1,1)=0
      CALL SCALE  (SR(1,1),ISR(1,1))
      CALL SCALE  (SR(2,1),ISR(2,1))
      CALL SCALE  (SR(3,1),ISR(3,1))
      GO TO 11
    8 SR(N2,1)=C(N)-B(2,3) $ ISR(N2,1)=0
      SR(N2-1,1)=-C(N1)-B(2,3) $ ISR(N2-1,1)=0
      CALL SCALE (SR(N2,1),ISR(N2,1))
      CALL SCALE (SR(N2-1,1),ISR(N2-1,1))
      IF (N2-2) 11,11,9
    9 DO 10 J=3,N2
      K=N2+1-J
      SR(K,1)=-B(J,3) $ ISR(K,1)=0
      CALL SCALE (SR(K,1),ISR(K,1))
   10 CONTINUE
   11 RETURN
   12 SR(1,1)=C(1) $ ISR(1,1)=0
      SR(2,1)=-C(2) $ ISR(2,1)=0
   13 CALL SCALE (SR(1,1),ISR(1,1))
      CALL SCALE (SR(2,1),ISR(2,1))
      GO TO 11
   14 SR(1,1)=-C(4)
      SR(2,1)=C(3)-C(1)
      ISR(1,1)=0
      ISR(2,1)=0
```

```
      GO TO 13
      END

      SUBROUTINE RECON (ROOTS,A,IA,D,N)
      DIMENSION ROOTS(2,50),D(51)
      X=A$IX=IA
      CALL UNSCALE (X,IX)
      DO 1 I=1,N
      D(I)=0.0
    1 CONTINUE
      D(N+1)=1.0
      I=1
      NL=N-1
    2 IF (ROOTS(2,I)) 3,7,3
    3 T=ROOTS(1,I)*ROOTS(1,I)
      U=ROOTS(2,I)*ROOTS(2,I)
      T=T+U
      U=2.0*ROOTS(1,I)
      DO 5 J=1,NL
      IF (I+J-N) 5,4,4
    4 D(J)=D(J+2)+T*D(J)
      D(J)=D(J)-U*D(J+1)
    5 CONTINUE
      D(N)=T*D(N)
      D(N)=D(N)-U*D(N+1)
      D(N+1)=T*D(N+1)
      I=I+2
    6 IF (N-I) 10,2,2
    7 DO 9 J=1,N
      IF (J+I-N) 9,9,8
    8 D(J)=D(J+1)-D(J)*ROOTS(1,I)
    9 CONTINUE
      D(N+1)=-D(N+1)*ROOTS(1,I)
      I=I+1
      GO TO 6
   10 NS=N+1
      DO 11 II=1,NS
      D(II)=D(II)*X
   11 CONTINUE
      RETURN
      END

      SUBROUTINE QUADIV (N,A,IA,R,D,B)
      DIMENSION A(51,3),IA(51,3),R(2),D(2),B(49)
      B(1)=A(1,1)$IB=IA(1,1)
      CALL UNSCALE (B(1),IB)
      IF (N-2) 4,4,1
    1 AA=A(2,1)$IAA=IA(2,1)
      CALL UNSCALE (AA,IAA)
      B(2)=AA-B(1)*D(1)
      IF (N-3) 4,4,2
    2 NT=N-1
      DO 3 I=3,NT
      XN=B(I-1)*D(1)
      YN=B(I-2)*D(2)
      AA=A(I,1)$IAA=IA(I,1)
      CALL UNSCALE (AA,IAA)
      B(I)=AA-(XN+YN)
    3 CONTINUE
    4 XN=B(N-1)*D(1)
```

```
      YN=B(N-2)*D(2)
      AA=A(N,1)$IAA=IA(N,1)
      CALL UNSCALE (AA,IAA)
      R(1)=AA-(XN+YN)
      AA=A(N+1,1)$IAA=IA(N+1,1)
      CALL UNSCALE (AA,IAA)
      R(2)=AA-B(N-1)*D(2)
      RETURN
      END

      SUBROUTINE DOUBLOG (X,IX,Y,IY)
      T=64.0
      IF (X)1,5,3
    1 PRINT  2
    2 FORMAT (46HOTHE LOG OF A NON-POSITIVE NUMBER IS REQUESTED)
    5 Y=0.0
      IY=0
      GO TO 4
3     TO=IX$
      Y=ALOG(X)+TO*ALOG(T)
      IY=0
      CALL SCALE (Y,IY)
    4 RETURN
      END

      SUBROUTINE DOUBLEXP (X,IX,Z,IZ)
      Z= EXP(X)$IZ=0
      IF (IX) 1,5,6
    1 I=-IX$II=6*I
      DO 4 J=1,II
      K=XMODF(IZ,2)
      IF (K) 2,3,2
    2 IZ=IZ-1
      Z=64.0*Z
    3 IZ=IZ/2
      Z=DSQRT(Z)
      CALL SCALE (Z,IZ)
    4 CONTINUE
    5 RETURN
    6 I=6*IX
      DO 7 J=1,I
      Z=Z*Z$IZ=IZ+IZ
      CALL SCALE (Z,IZ)
    7 CONTINUE
      GO TO 5
    8 CALL SCALE (Z,IZ)
      GO TO 5
      END

      SUBROUTINE ADD (X,IX,Y,IY,Z,IZ)
      IF (X) 3,1,3
    1 Z=Y
      IZ=IY
    2 RETURN
    3 IF (Y) 5,4,5
    4 Z=X
      IZ=IX
      GO TO 2
    5 IDIFF=IX-IY
      IF (IDIFF) 6,7,7
```

```
 6 IA=IY
   A=Y
   B=X
   IDIFF=-IDIFF
   GO TO 8
 7 IA=IX
   A=X
   B=Y
 8 IF (16-IDIFF) 9,9,10
 9 Z=A
   IZ=IA
   GO TO 2
10 IF(IDIFF) 11,12,11
11 DO 12 I=1,IDIFF
   B=B/64.0
12 CONTINUE
   Z=A+B
   IZ=IA
   CALL SCALE (Z,IZ)
   GO TO 2
   END

   SUBROUTINE SUBTRACT (X,IX,Y,IY,Z,IZ)
   W=-Y
   CALL ADD (X,IX,W,IY,Z,IZ)
   RETURN
   END

   SUBROUTINE SCALE (X,IX)
   TYPE DOUBLE X,Y
   REC64=1.0/64.0
   IF (X) 1,11,2
 1 Y=-X
   GO TO 3
 2 Y=X
 3 IF (64.0-Y) 4,5,5
 4 Y=Y/64.0
   IX=IX+1
   GO TO 3
 5 IF (Y-REC64) 6,7,7
 6 Y=Y*64.0
   IX=IX-1
   GO TO 5
 7 IF (X) 8,9,9
 8 X=-Y
   GO TO 10
 9 X=Y
10 RETURN
11 IX=0
   GO TO 10
   END

   SUBROUTINE UNSCALE (X,IX)
   IF (IX+84 ) 3,4,4
 3 X=0.0
   IX=0
   GO TO 2
 4 IF (IX-84 ) 5,5,6
 6 X=1.0E+153
   IX=0
```

```
      PRINT    7
    7 FORMAT (25H␣EXP. OVERFLOW IN UNSCALE)
      GO TO 2
    5 IF (IX) 1,2,1
    1 X=X*64.0**IX
    8 IX=0
    2 RETURN
      END
E
```

APPENDIX C

INPUT

```
EXAMPLE 1
     6     20     20 1.000E-4     1.000E-6
 7056.
-51072.
 22559.36
 661360.64
-2101084.2
 2490368.
-1048576.
```

OUTPUT

```
EXAMPLE 1
DEGREE=  6  NO. OF ROOT SQ= 20  MMAX= 20
DELTA=   0.100E-03  EPSILON=   0.100E-05

INPUT COEFFICIENTS

A( 0)=   7.0560000000E 03

A( 1)=  -5.1072000000E 04

A( 2)=   2.2559360000E 04

A( 3)=   6.6136064000E 05

A( 4)=  -2.1010842000E 06

A( 5)=   2.4903680000E 06

A( 6)=  -1.0485760000E 06

ROOTS OF THE POLYNOMIAL

REAL PART                    IMAGINARY PART

  -3.999687510094593E 00      0.000000000000000E 00

   3.998749957314208E 00      0.000000000000000E 00

   2.285716169209813E 00      0.000000000000000E 00

   2.285716169209813E 00      0.000000000000000E 00

   1.357365320213065E 00      0.000000000000000E 00

   1.310236461141304E 00      0.000000000000000E 00
```

OUTPUT

RECONSTRUCTED POLYNOMIAL

```
A( 0)=   7.0560000000E 03

A( 1)=  -5.1072009377E 04

A( 2)=   2.2559385004E 04

A( 3)=   6.6136077332E 05

A( 4)=  -2.1010846001E 06

A( 5)=   2.4903682667E 06

A( 6)=  -1.0485760000E 06
```

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
U.S.Army Aberdeen Research and Development Center, Ballistic Research Laboratories
Aberdeen Proving Ground, Maryland

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3. REPORT TITLE
NUMERICAL SOLUTIONS OF POLYNOMIAL EQUATIONS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)
Tadeusz Leser and Henry Wisniewski

6. REPORT DATE
August 1968

7a. TOTAL NO. OF PAGES

7b. NO. OF REFS

8a. CONTRACT OR GRANT NO.

b. PROJECT NO. RDT&E 1T061101A14B

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
Technical Note No. 1699

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
U.S. Army Materiel Command
Washington, D.C.

13. ABSTRACT

This report describes the FORTRAN Subroutine POLYR and a related complete program BRL-RSSR for finding all roots (real and complex) of a real polynomial equation

$$P(x) = \sum_{i=0}^{N} A_i x^{N-i} = 0.$$

The method which is used combines the root squaring and the subresultant (extracting quadratic factors) procedures for calculating all roots, even multiple roots, of real polynomials. Reconstruction of the coefficients from the product

$$\prod_{i=1}^{N} (x - \mathrm{Root}_i) \quad ,$$

which is included, serves as a means of checking.

The described subroutine and program were adapted from a similar program in double precision described in Report ANL 6987, Argonne National Laboratory by Erwin H. Bareiss and Ronald Hamelink.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified

Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Roots of Polynomials						
Polynomials						
Roots						

Unclassified

Security Classification