63-

406 436

BRL

MEMORANDUM REPORT NO. 1467
APRIL 1963

ROUTINE FOR FINDING ROOTS OF POLYNOMIALS WITH REAL COEFFICIENTS

Tadeusz Leser

RDT & E Project No. 1M010501A003

BALLISTIC RESEARCH LABORATORIES

ABERDEEN PROVING GROUND, MARYLAN

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

MEMORANDUM REPORT NO. 1467

APRIL 1963

ROUTINE FOR FINDING ROOTS OF POLYNOMIALS WITH REAL COEFFICIENTS

Tadeusz Leser

Computing Laboratory

RDT & E Project No. 1M010501A003

A B E R D E E N   P R O V I N G   G R O U N D,   M A R Y L A N D

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

MEMORANDUM REPORT NO. 1467

TLeser/bj
Aberdeen Proving Ground, Md.
April 1963

ROUTINE FOR FINDING ROOTS OF POLYNOMIALS WITH REAL COEFFICIENTS

ABSTRACT

A routine has been developed which computes the real and the complex roots of polynomials with real coefficients up to the tenth degree. In case of an even polynomial it computes its quadratic factors and solves each factor by the quadratic formula. In case of an odd polynomial it computes one real root by locating it and refining by Newton's algorithm. Then it removes the computed root from the polynomial reducing its degree by one. The reduced polynomial is of even degree and it is solved by quadratic factors. The routine fails when (a) the real roots are of even multiplicity, converging then to wrong values, (b) the real roots are of odd multiplicity, not converging at all, (c) the polynomial is badly conditioned (very small changes in coefficents cause large changes in the values of the roots) converging then to wrong values.

The routine is planned to compute frequencies in vibrations problems which involve complex roots. Statistical evidence seems to indicate that only polynomials with real roots can be badly conditioned. If this were the case the handicap (c) would be of minor importance only.

A polynomial of n-th degree can be written as

$$f(x) = x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n = \sum_{i=o}^{n} a_i x^{n-i}$$

where $a_o = 1$, and a_i are real numbers. The roots of $f(x)$ will be denoted by

$$\bar{x}_1, \bar{x}_2, \dots\dots\dots \bar{x}_n.$$

The coefficient a_o of x^n has been set equal to one for convenience.

If n is even all the roots may be complex numbers. If n is odd there must be at least one real root, in which case we start by finding this real root.

(a) Locating One Real Root for an Odd Polynomial.

We shall establish first the bound for all roots, which is given by the theorem

$$\text{UB(upper bound)} = \text{Max } |a_i| + 1,$$

Where Max $|a_i|$ is the greatest coefficient a_i, $(i = 1,2,\dots,n)$ in absolute value. Thus

$$\bar{x}_1 < \text{UB}.$$

Let us find two consecutive values of $f(x)$ for real values of x, say $f_1 = f(x_1)$ and $f_2 = f(x_2)$, where $x_1 = -\text{UB}+\Delta x$; $x_2 = -\text{UB}+2\Delta x$, and Δx is an assigned step size of x, chosen sufficiently small that there is likely to be only one real root between $-\text{UB} + k\Delta x$ and $-\text{UB} + (k + 1)\Delta x$ for any k. If $f_1 < 0$ and $f_2 > 0$ then the real root is between x_1 and x_2. If f_1 and f_2 have the same sign, then compute f_3 and compare the signs of f_2 and f_3 and so on. Suppose that at some value x_{k+1}, f changes sign, that is

$$f_k < 0 \text{ and } f_{k+1} > 0.$$

Then the real root $\bar{x}_1$ is located between x_k and x_{k+1},

$$x_k < \bar{x}_1 < x_{k+1}.$$

(b) Refining the Located Root by Newton's Method.

Let us call the first approximation to the root $\bar{x}_1$

$$x^{(1)} = x_{k+1}.$$

The next approximation is then

$$x^{(2)} = x^{(1)} - f(x^{(1)})/f'(x^{(1)}),\ f'(x^{(1)}) = \left(\frac{d\ f(x)}{d\ x}\right)_{x^{(1)}}$$

and generally

$$x^{(j+1)} = x^{(j)} - f(x^{(j)})/f'(x^{(j)}).$$

The iteration continues until two consecutive values of $x^{(j)}$ becomes equal to each other within some small preassigned number. This establishes the first real root $\bar{x}_1 = x^{(j)}$.

(c) Removing from f(x) the Known Real Root.

The next step is to remove this real root by synthetic decision, and find the coefficients b_i of the depressed polynomial. $f(x)$ may be written

$$f(x) = \sum_{i=o}^{n} a_i\ x^{n-i} = (x - \bar{x}_1) \sum_{i=o}^{n-1} b_i\ x^{n-1-i}$$

Equating the coefficients of the same powers in both members of the above identity gives

$$b_o = 1.$$

$$b_r = a_r + b_{r-1}\,\bar{x}_1,\ (r = 1,2,\ldots,n-1).$$

The depressed polynomial $g_1(x) = \sum_{i=o}^{n-1} b_i\ x^{n-1-i}$ is of even degree.

(d) Computing Quadratic Factors of Even Polynomials.

An even polynomial can always be factored into real quadratic factors. Suppose that

$$x^2 + p\,x + q$$

is a real quadratic factor of $f(x)$. Then $f(x)$ may be written

$$f(x) = \sum_{i=o}^{n} a_i x^{n-i} = (x^2 + p x + q) \sum_{i=o}^{n-2} b_i x^{n-2-i}$$

Equating the coefficients of the same powers in both members of the above identity gives the coefficients b_i of the reduced polynomial.

$$b_o = 1; \; b_{-1} = 0.$$

$$b_r = a_r - pb_{r-1} - qb_{r-2}; \; (r = 1,2,\ldots,n-2).$$

In the case when the quadratic

$$x^2 + p x + q$$

is not an exact factor of $f(x)$, the factoring gives

$$\sum_{i=o}^{n} a_i x^{n-i} = (x^2 + p x + q) \sum_{i=o}^{n-2} b_i x^{n-2-i} + R$$

where

$$R = S_1 x + S_2.$$

Equating the coefficients of x gives

$$S_1 = b_{n-1}; \; S_2 = b_n + pb_{n-1}.$$

Bairstow devised an iteration scheme for improving the values of p and q, in such a way that the coefficients S_1 and S_2 and the remainder term R approach zero.

<u>Bairstow Algorithm</u>. Let $\overline{p}$ and $\overline{q}$ be the exact values of the coefficients which will make S_1 and S_2 vanish. The approximate coefficients S_1 and S_2 are functions of p and q and a Taylor series expansion (nonlinear terms being neglected) with $\overline{p} = p + \Delta p$; $\overline{q} = q + \Delta q$ yields

$$\begin{aligned} S_1(\overline{p}, \overline{q}) &= 0 = S_1(p, q) + \Delta p(\partial S_1/\partial p) + \Delta q(\partial S_1/\partial q) \\ S_2(\overline{p}, \overline{q}) &= 0 = S_2(p, q) + \Delta p(\partial S_2/\partial p) + \Delta q(\partial S_2/\partial q) \end{aligned} \qquad (1)$$

Since $S_1 = b_{n-1}$ and $S_2 = b_n + pb_{n-1}$

we have

$$\partial S_1/\partial p = \partial b_{n-1}/\partial p; \; \partial S_2/\partial p = \partial b_n/\partial p + p(\partial b_{n-1}/\partial p) + b_{n-1}$$

$$\partial S_1/\partial q = \partial b_{n-1}/\partial q; \; \partial S_2/\partial q = \partial b_n/\partial q + p(\partial b_{n-1}/\partial q)$$

Upon using the relation

$$b_r = a_r - pb_{r-1} - qb_{r-2}$$

and introducing the notation

$$-(\partial b_r/\partial p) = d_{r-1}$$

we obtain the recursion formula for $\frac{\partial b_r}{\partial p}$

$$d_r = b_r - pd_{r-1} - qd_{r-1}; \; d_{-1} = 0; \; d_o = 1; \; (r = 1, 2, \ldots., n-2)$$

A similar recursion formula for $\frac{\partial b_r}{\partial q}$ can be obtained, giving $-(\partial b_r/\partial q) = d_{r-2}$

Hence

$$\partial S_1/\partial p = - d_{n-2}; \; \partial S_1/\partial q = - d_{n-3}$$
$$\partial S_2/\partial p = - d_{n-1} + pd_{n-2} + b_{n-1}; \; \partial S_2/\partial q = - d_{n-2} - pd_{n-3} \tag{2}$$

Substituting (2) in (1) yields

$$(\Delta p) \; d_{n-2} + (\Delta q) \; d_{n-3} = b_{n-1}$$

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

which, solved by Cramer's rule for Δp and Δq gives

$$\Delta p = \frac{\begin{vmatrix} b_{n+1} & d_{n-3} \\ b_n & d_{n-2} \end{vmatrix}}{\text{Den}} \qquad \Delta q = \frac{\begin{vmatrix} d_{n-2} & b_{n-1} \\ (d_{n-1} - b_{n-1}) & b_n \end{vmatrix}}{\text{Den}}$$

where

$$\text{Den} = \begin{vmatrix} d_{n-2} & d_{n-3} \\ (d_{n-1} - b_{n-1}) & d_{n-2} \end{vmatrix}$$

Since the non-linear terms in the Taylor expansion have been neglected the new values of p and q

$$\bar{p} = p + \Delta p; \qquad \bar{q} = q + \Delta q$$

will not represent the true coefficients of the quadratic factor. We expect them to be better approximations than p and q in the sense that they will make S_1 and S_2 smaller. This procedure can be repeated until S_1 and S_2 will vanish within some prescribed small number.

Initial Approximations. The convergence of the Bairstow algorithm depends on the initial approximation being close enough to the true value. We have selected the following initial approximation to p and q:

$$p = 2(|a_{n-1}|/n)^{\frac{1}{n-1}}, \quad q = |a_n|^{\frac{2}{n}}$$

The above approximations are averages derived from the well known properties of the polynomial coefficients,

$$a_n = \prod_{i=1}^{n} \bar{x}_i; \qquad a_{n-1} = \sum_{i=1}^{n} \frac{1}{\bar{x}_i} \prod_{j=1}^{n} \bar{x}_j$$

This initial approximation gave convergence in very many widely different cases which were tested. The routine diverged or converged to wrong values for polynomials with multiple real roots. Some of the 9th degree polynomials with all real roots were badly conditioned and converged to wrong values. Polynomials with all or some complex roots converged in all cases to the correct values.

It has been found that computation of quadratic factors of odd polynomials by the Bairstow algorithm, which we use in our routine, diverged in all cases which were tested, even in those where the initial approximations were very close to the true values. This interesting fact has not to my knowledge been registered in the existing literature on the subject.

High degree polynomials with coefficients of the order 10^3 or higher must have the roots reduced by a constant factor, say 10^{-k}, in order to make the coefficients smaller. Otherwise the intermediate products arising in the computations may exceed in magnitude the capacity of a machine.

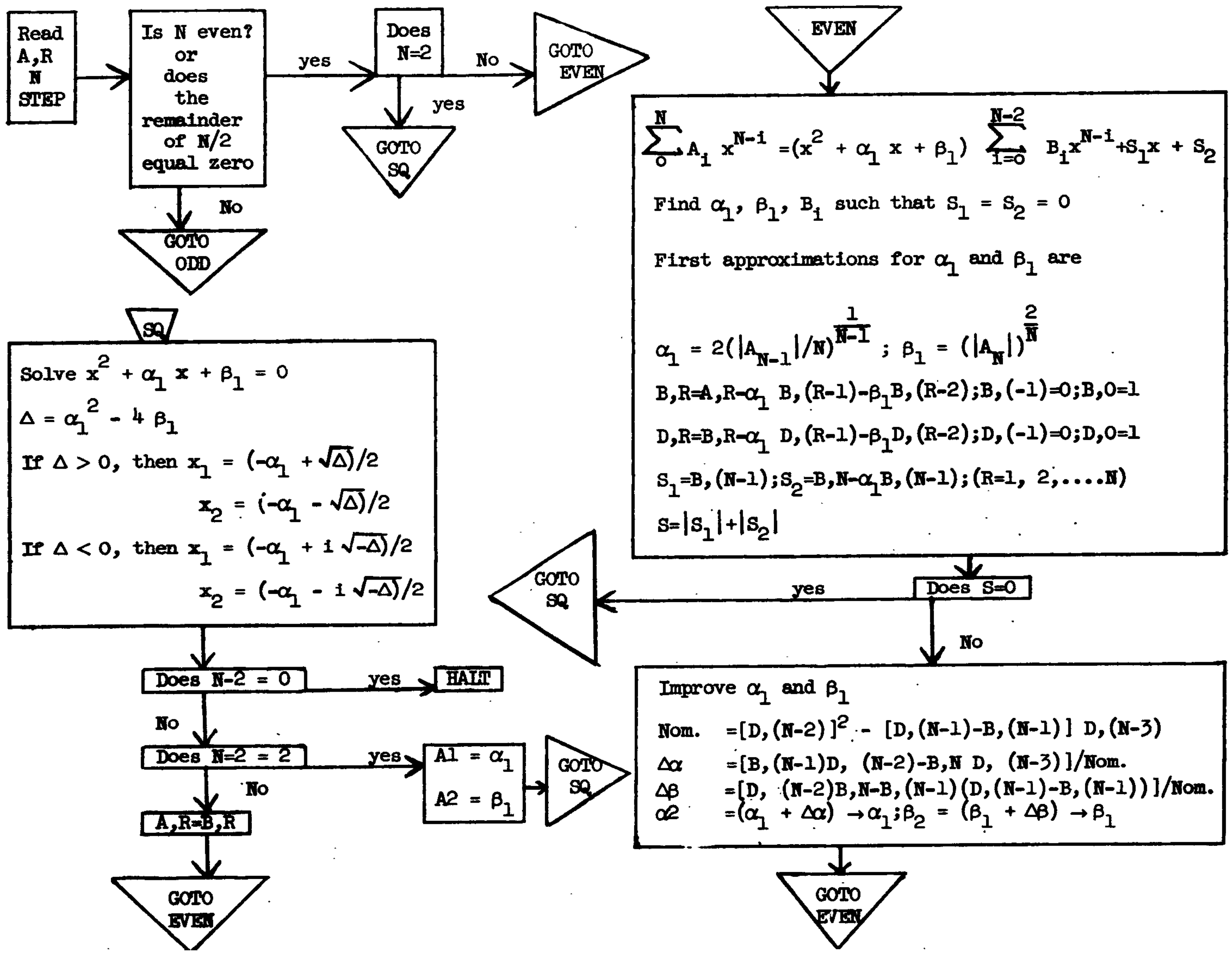
The appended flow chart and the program in the FORAST language explain the computational procedure.

Tadeusz Leser

TADEUSZ LESER

APPENDIX I

FLOW CHART



Read A,R N STEP
Is N even? or does the remainder of N/2 equal zero
yes
No
Does N=2
No
yes
GOTO EVEN
GOTO SQ
GOTO ODD
EVEN
$\sum_{o}^{N} A_i x^{N-i} = (x^2 + \alpha_1 x + \beta_1) \sum_{i=o}^{N-2} B_i x^{N-i} + S_1 x + S_2$
Find α_1, β_1, B_i such that $S_1 = S_2 = 0$
First approximations for α_1 and β_1 are
$\alpha_1 = 2(|A_{N-1}|/N)^{\frac{1}{N-1}}$; $\beta_1 = (|A_N|)^{\frac{2}{N}}$
B,R=A,R-α_1 B,(R-1)-β_1B,(R-2);B,(-1)=0;B,0=1
D,R=B,R-α_1 D,(R-1)-β_1D,(R-2);D,(-1)=0;D,0=1
S_1=B,(N-1);S_2=B,N-α_1B,(N-1);(R=1, 2,....N)
S=$|S_1|+|S_2|$
Does S=0
yes
No
GOTO SQ
SQ
Solve $x^2 + \alpha_1 x + \beta_1 = 0$
$\Delta = \alpha_1^2 - 4\beta_1$
If $\Delta > 0$, then $x_1 = (-\alpha_1 + \sqrt{\Delta})/2$
$x_2 = (-\alpha_1 - \sqrt{\Delta})/2$
If $\Delta < 0$, then $x_1 = (-\alpha_1 + i\sqrt{-\Delta})/2$
$x_2 = (-\alpha_1 - i\sqrt{-\Delta})/2$
Does N-2 = 0
yes
HALT
No
Does N-2 = 2
yes
A1 = α_1
A2 = β_1
GOTO SQ
No
A,R=B,R
GOTO EVEN
Improve α_1 and β_1
Nom. =[D,(N-2)]^2 - [D,(N-1)-B,(N-1)] D,(N-3)
$\Delta\alpha$ =[B,(N-1)D, (N-2)-B,N D, (N-3)]/Nom.
$\Delta\beta$ =[D, (N-2)B,N-B,(N-1)(D,(N-1)-B,(N-1))]/Nom.
α2 =(α_1 + $\Delta\alpha$) $\rightarrow \alpha_1$;β_2 = (β_1 + $\Delta\beta$) $\rightarrow \beta_1$
GOTO EVEN

FLOW CHART (Cont'd)

ODD

Compute A_{max} = Max $(|A_1|, |A_2| \ldots\ldots, |A_n|)$

Compute Upper bound = UB = $A_{max} + 1$.

Compute consecutive values $f_k = f(x_k)$; $k = 1,2,\ldots.$

(Initial value $x_o = ((K)(Step) - UB)$.

Compare two consecutive values f_k and f_{k+1}

If f_k and f_{k+1} have opposite sign then the root $\bar{x}_1$ is located, $x_k < \bar{x}_1 < x_{k+1}$

Find coefficient of the derivative polynomial $\sum_{i=o}^{N-1} C_i x^{N-1-i}$.

$$(f'(x) = \sum_{n=o}^{N-1} C_i x^{N-1-i} = \sum_{i=o}^{N-1} A_i (N-i)\, x^{N-1-i})$$

$C_i = A_i(N-i), (i=1,2\ldots n-1)$.

Refine the value of $\bar{x}_1$ by Newton's method

Taking as initial approximation $x^{(o)} = x_{k+1}$

$$x^{(i+1)} = x^{(i)} - f(x_i)/f'(x_i);$$

When $x^{(k+1)} = x^{(k)}$ within small number then

$x^{(k)} = \bar{x}_1$; Print $\bar{x}_1$.

Does N-1=2 — yes → GOTO SQ.

NO

Find the coefficient of the reduced polynomial $\sum_{i=o}^{n-1} T_i x^{n-i-1}$

$$(f(x) = (x - \bar{x}_1) \sum_{i=o}^{n-1} T_i x^{n-i-1});$$

$T_i = A_i + \bar{x}_1\ (i-1)$; Replace A_i by T_i.

GOTO EVEN

APPENDIX II

PROGRAM IN FORAST LANGUAGE

```
           BLOC(A-A10)B-B10)D-D10)C-C10)S-S11)T-T10)%
START         READ(NF)STEP)%
              READ(10)NOS.AT(A)%
              H°NF/2% ENTER(WH.FRA)H)WHN)FRAN)%
              IF(FRAN°0.5)WITHIN(.00001)GOTO(ODD)%
              IF-NOT(NF-2°0) WITHIN(.00001) GOTO(EVEN)%
              ALP1°A1% BET1°A2% GOTO(SQ)%
ODD           ENTER(CVFTOI)NF)N)%
              SET(R°1)% AM°ABS(A1)%
1.            IF-ABS(A,(R+1) %AM)GOTO(2.)%
              COUNT(N)IN(R)GOTO(1.)% GOTO(3.)
2.            AM°ABS(A,(R+1))%
              COUNT(N)IN(R)GOTO(1.)%
3.            UB°1+AM%
              SUM°1% SET(I°1)%
3.1           SUM°SUM*(-UB)+A,I%
              COUNT(N+1)IN(I)GOTO(3.1)%
              SUM1°SUM% SET(K°1)%
3.11                    ENTER(CVITOF)K)KF)%
              EX1°KF*STEP-UB
              S1°1% SET(I°1)%
GPS           S,(I+1)°S,I*EX1 +A,I%
              COUNT(N+1)IN(I) GOTO(GPS)%
              SUM2°S,(N+1)%
              IF(ABS(SUM2-SUM1)%°ABS(SUM2+SUM1))GOTO(3.2)%
              SUM1°SUM2% INT(K°K+1)% GOTO(3.11)%
3.2           SET(I°1)% CO°NF%
4.            ENTER(CVITOF)I)IF)%
              C,I°A,I*(NF-IF)%
              COUNT(N)IN(I) GOTO(4.)%
              SET(J°1)% SD1°CO%
GPSD          SD,(J+1)°SD,J*EX1 +C,J%
              COUNT(N)IN(J) GOTO(GPSD)%
              SUMD°SD,N%
4.1           EX°EX1 -(SUM2/SUMD)%
              IF(EX°EX1 )WITHIN(.00001)GOTO(ROOT1)
              EX1 °EX% S1°1% SET(I°1)% SD1°CO%
5.            S,(I+1)°S,I*EX+A,I% SD,(I+1)°SD,I*EX+C,I%
              COUNT(N+1)IN(I) GOTO(5.)%
              SUM2°S,(N+1)% SUMD°SD,N% GOTO(4.1)%
ROOT1         PRINT*RROOT%(EX)%
              TO°1% SET(I°1)%
6.            T,I°A,I+EX*T,(I-1)%
              COUNT(N)IN(I)GOTO(6.)%
              MOVE(N)NOS.FROM(T)TO(A)%
              IF-NOT(NF-1°2)WITHIN(.00001)GOTO(7.1)
              ALP1°A1%BET1°A2%NF°NF-1%GOTO(SQ)%
7.1           INT(N°N-1)% NF°NF-1% GOTO(EVEN)
EVEN          ENTER(CVFTOI)NF)N)%
              P°ABS(A,(N-1))/NF%Q°ABS(A,N)%
              Y°1/(NF-1)% ENTER(POWER)P)Y)ALP)% ALP1°2*ALP%
              X°2/NF% ENTER(POWER)Q)X)BET1)%
BE1           INT(N°N+1)% SET(R°2)% B0°1% D0°1%
              B1°A1-ALP1% D1°B1-ALP1%
BE            B,R°A,R-ALP1*B,(R-1)-BET1*B,(R-2)%
              D,R°B,R-ALP1*D,(R-1)-BET1*D,(R-2)%
              COUNT(N)IN(R)GOTO(BE)%
              INT(N°N-1)% S1°B,(N-1)% S2°B,N+ALP1*B,(N-1)%
              S°ABS(S1)+ABS(S2)%
              IF(S°0)WITHIN(.0001)GOTO(SQ)%
              DEL°D,(N-2)**2-D,(N-3)*(D,(N-1)-B,(N-1))%
              DELAL°(B,(N-1)*D,(N-2)-B,N*D,(N-3))/DEL%
              DELBE°(D,(N-2)*B,N-B,(N-1)*(D,(N-1)-B,(N-1)))/DEL%
              ALP2°ALP1+DELAL% BET2°BET1+DELBE%
              ALP1°ALP2% BET1°BET2% GOTO(BE1)%
SQ            DELTA°ALP1**2-4*BET1%
              IF(DELTA*0)GOTO(8.)%
              ALPH°-ALP1/2% BET°SQRT(DELTA)/2%
              X1°ALPH+BET% X2°ALPH-BET%
ROOTR         PRINT*RROOTS%(X1)(X2)%
              PRINT*COEFS%(ALP1)(BET1)%
              IF-NOT(NF-2°0)WITHIN(.00001)GOTO(9.)
              GOTO(START)%
8.            ALPH°-ALP1/2% BET°SQRT(-DELTA)/2%
ROOTI         PRINT*IROOTS%(ALPH)(BET)%
              PRINT*COEFS%(ALP1)(BET1)%
              IF-NOT(NF-2°0)WITHIN(.00001)GOTO(9.)%
              GOTO(START)%
9.            IF-NOT(NF-2°2)WITHIN(.00001)GOTO(9.1)%
              INT(N°N-2)% NF°NF-2% ALP1°B1% BET1°B2% GOTO(SQ)%
9.1           NF°NF-2% INT(N°N-2)%
              MOVE(N)NOS.FROM(B1)TO(A1)%
              GOTO(EVEN)%
          END GOTO(START)%
```

APPENDIX III

TYPICAL INPUT

Each line corresponds to one card.

1st line   degree of the polynomial N and stepsize STEP

2nd line   The first six coefficients of the polynomial, A0,A1,A2,A3,A4,A5,

3rd line   the seventh coefficient A6.

```
60000000 01 50000000 01
10000000 00 16033508 02 48171359 02 31974650 01 42487209 00 23977863-01
29523451-03
```

TYPICAL OUTPUT

1st line two real roots $\bar{x}_1, \bar{x}_2$

3rd line real and imaginary parts of two complex roots $\bar{x}_3, \bar{x}_4$

5th line two real roots $\bar{x}_5, \bar{x}_6$

2nd, 4th, 6th lines give coefficients of the quadratic trinomials corresponding to the roots above. The first coefficient in all cases is 1 and it is not printed.

```
RROOTS-16876462-01-46551241-01                          7
COEFS 63427703-01 78562026-03                           8
IROOTS-70631108-03 89263459-01                          9
COEFS 14126222-02 79684641-02                          10
RROOTS-39072656  1-12061402  2                         11
COEFS 15968668  2 47127102  2                          12
```

DISTRIBUTION LIST

No. of Copies	Organization
10	Commander Armed Services Technical Information Agency ATTN: TIPCR Arlington Hall Station Arlington 12, Virginia
1	Commanding General U. S. Army Materiel Command ATTN: AMCRD-RS-PE-Bal, Mr. Stetson Research and Development Directorate Washington 25, D. C.
1	Commanding Officer Harry Diamond Laboratories ATTN: Technical Information Office, Branch 012 Washington 25, D. C.
1	Commanding General U. S. Army Missile Command Redstone Arsenal, Alabama
1	Commanding General White Sands Missile Range New Mexico
3	Commanding Officer U. S. Army Research Office (Durham) Box CM, Duke Station Durham, North Carolina
1	Army Research Office 3045 Arlington Pike Arlington Hall, Virginia
1	Operations Research Office Department of the Army 6935 Arlington Road Bethesda, Maryland Washington 14, D. C.
3	Chief, Bureau of Naval Weapons ATTN: DIS-33 Department of the Navy Washington 25, D. C.
1	Commander U. S. Naval Weapons Laboratory Dahlgren, Virginia
1	Commander U. S. Naval Ordnance Test Statio[n] ATTN: Technical Library China Lake, California
2	Commander Naval Ordnance Laboratory White Oak Silver Spring 19, Maryland
1	Commanding Officer and Director David W. Taylor Model Basin Washington 7, D. C.
1	Director, Project RAND Department of the Air Force 1700 Main Street Santa Monica, California
1	ASD (ASRSEE, Mr. Gard) Wright-Patterson Air Force Base Ohio
1	Director National Aeronautics & Space Administration 1520 H Street, N. W. Washington 25, D. C.
2	U. S. Atomic Energy Commission Los Alamos Scientific Laborator[y] ATTN: Computation Laboratory P. O. Box 1663 Los Alamos, New Mexico

DISTRIBUTION LIST

No. of Copies	Organization
1	Oak Ridge National Laboratory P. O. Box P Oak Ridge, Tennessee
1	Chief, Numerical Analysis Section National Bureau of Standards Division 11.1 Washington 25, D. C. Of Interest to: Dr. Mittleman
1	Harvard University Computation Laboratory ATTN: Dr. H. Aiken Cambridge, Massachusetts
1	Massachusetts Institute of Technology Digital Computer Laboratory 211 Massachusetts Avenue Cambridge 39, Massachusetts
1	Oregon State College Department of Mathematics ATTN: Dr. W. E. Milne Corvallis, Oregon
1	Army Mathematics Center University of Wisconsin Madison, Wisconsin
1	Republic Aviation Corporation ATTN: Gershinsky - Digital Computing & Data Processing Division Farmingdale, Long Island, New York
1	Michigan College of Mining & Technology Houghton Michigan
1	Professor A. A. Bennett Department of Mathematics Brown University Providence 12, Rhode Island
1	Professor John W. Cell School of Physical Sciences & Applied Math Department of Mathematics North Carolina State College Raleigh, North Carolina
1	Professor Saul Gorn Moore School of Electrical Engineering University of Pennsylvania Philadelphia 4, Pennsylvania
1	Professor Thomas J. Higgins Department of Electrical Engineering Engineering Building University of Madison Madison 6, Wisconsin
1	Professor R. S. Lehman Department of Mathematics University of California Berkeley 4, California
1	Professor A. J. Perlis Computation Center Carnegie Institute of Technology Pittsburgh 13, Pennsylvania
10	The Scientific Information Officer Defence Research Staff British Embassy 3100 Massachusetts Avenue, N. W. Washington 8, D. C.
4	Defence Research Member Canadian Joint Staff 2450 Massachusetts Avenue, N. W. Washington 8, D. C.