Mathematics Notes

Note I

15 October 1966

BESSEL

A Subroutine for the Generation of Bessel Functions with Real or Complex Arguments

Richard C. Lindberg

The Dikewood Corporation

ABSTRACT

A subroutine for the generation of Bessel, Neumann, and Hankel functions is described from both the operational and the algorithmic points of view. Methods of accuracy verification are discussed, and sample error printouts are presented. The use of the subroutine to generate tables of Bessel functions and their derivatives is outlined.

BESSEL

A Subroutine for the Generation of
Bessel Functions with Real or Complex Arguments

INTRODUCTION

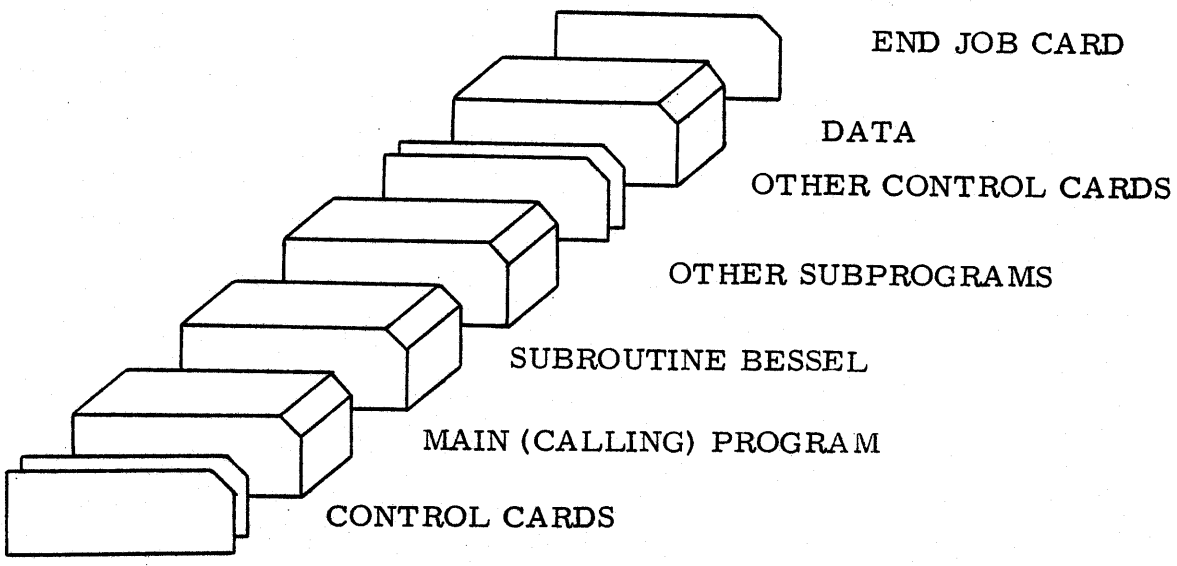
This note describes a computer subroutine entitled BESSEL, written in FORTRAN 66 for the CDC 6600 computer. The subroutine accepts as inputs the order (n) and argument (z) of the function, and returns values for $J_n(z)$, $Y_n(z)$, $H_n^{(2)}(z)$, $J'_n(z)$, $Y'_n(z)$, and $H_n^{(2)\prime}(z)$. The order must be TYPE INTEGER in the range $0 \leq |n| \leq 100$ and the argument must be TYPE COMPLEX in the range $0.001 \leq |z| \leq 100$. Values outside these ranges will be accepted, but the accuracy of the results is not guaranteed. All returned parameters must be TYPE COMPLEX.

There are a number of error checks contained within the subroutine, and a validity check parameter (TYPE INTEGER) is returned along with calculated answers to enable a qualitative evaluation of the answers to be made. This check is normally zero. If it is not, then either one or more answers do not meet the accuracy requirements or one or more answers are outside the range of the computer ($\pm 10^{150}$).

OPERATION

General

BESSEL must be used in conjunction with a main routine which assigns values to the order and argument, calls BESSEL, and accepts the results from the subroutine. A typical program production deck incorporating the BESSEL subroutine is illustrated below:



Production Deck Organization Using BESSEL

Calling Sequence

BESSEL is called by the main program with a FORTRAN 66 CALL statement. The format of this statement is

```
CALL BESSEL (N, Z, J, Y, H2, JPRIME, YPRIME, H2PRIME,
IVALCHK, IPRINT)
```

The parameters N, Z and IPRINT are supplied to the subroutine by the main program; they must have been previously assigned before execution of the CALL statement. All other parameters are returned by the subroutine.

Parameters

1. N — TYPE INTEGER. This is the order of the Bessel functions to be calculated. For stated accuracy, the range of N must be $0 \leq |N| \leq 100$.

2. Z — TYPE COMPLEX. This is the argument of the Bessel functions to be calculated. For stated accuracy, Z must be in the range $0.001 \leq |Z| \leq 100$, and $|\arg Z| < \pi$. Zero is an acceptable argument.

3. J — TYPE COMPLEX. This is the value of the Bessel function of the first kind of order N and argument Z, returned by the subroutine.

4. Y — TYPE COMPLEX. This is the value of the Bessel function of the second kind (Neumann function) of order N and argument Z, returned by the subroutine.

5. H2 — TYPE COMPLEX. This is the value of the Hankel function of the second kind of order N and argument Z, returned by the subroutine.

6. JPRIME — TYPE COMPLEX. This is the value of the derivative of J, returned by the subroutine.

7. YPRIME — TYPE COMPLEX. This is the value of the derivative of Y, returned by the subroutine.

8. H2PRIME — TYPE COMPLEX. This is the value of the derivative of H2, returned by the subroutine.

9. IVALCHK — TYPE INTEGER. This is a validity check parameter returned by the subroutine. If the calculated values for J, Y, H2, JPRIME, YPRIME and H2PRIME are within the range of the computer and meet the accuracy requirements, then the value of IVALCHK is zero. If any one of the calculated values is outside the range or does not meet the accuracy requirement, then IVALCHK = 1.

10. IPRINT — TYPE INTEGER. This parameter is supplied by the main program and has the value 0 or 1. If IPRINT = 0, the subroutine provides all values as stated above and nothing more. If IPRINT = 1, in addition to all calculated values the subroutine prints all appropriate error messages. These messages are primarily of a diagnostic nature and their selection is not under control of the programmer. The value of IPRINT is normally 0.

The names given to the parameters were chosen for illustrative purposes, and are formal parameters in the subroutine. The actual parameters used must agree in type and number, and need not be identical in name.

ERROR MESSAGES

Validity Check

The parameter IVALCHK contains a returned value of either zero or one. If the value is one, then either one or more answers are out of range of the computer or one or more answers do not meet the accuracy requirements. This parameter is always returned, and provides a means for quick, qualitative evaluation of the calculated results.

Error Printouts

If the value of IPRINT is equal to one, then if any criterion as stated below is not met, a printout occurs. These various printouts originate at various points in the main BESSEL subroutine, and at various points within certain subroutines which BESSEL itself uses. Each printout, its origin and the criterion is discussed below, in conjunction with a brief description of the subroutine or subprogram containing it.

1. FUNCTION ZERO -- This subprogram provides an independent check of the values generated by BESSEL by verifying that these calculated values are solutions to a differential equation having Bessel functions as its solution. The differential equation used is

$$z[a_{n+1}(z)+a_n'(z)] - na_n(z) = 0$$

where $a_n(z)$ is any Bessel function of order n and argument z. If the left side of the equation above is greater in magnitude than 10^{-8}, and IPRINT $\neq$ 0, the following error message is printed:

ZERO PRINT	zero	z
$a_n(z)$	$a_{n+1}(z)$	$a_n'(z)$
factor 1	factor 2	

Zero is the calculated difference from zero as computed by the following:

$$\text{zero} = 1 - \left| \frac{z[a_{n+1}(z) + a_n'(z)]}{na_n(z)} \right|$$

z is the value of the argument; $a_n(z)$, $a_{n+1}(z)$ are given values of the Bessel functions of orders n and n+1, respectively, and of argument z; and $a_n'(z)$ is the given value of the derivative of the Bessel function of order n and argument z. Factor 1 and factor 2 are values of the numerator and denominator, respectively, of the ratio contained in the above formula.

2. SUBROUTINE CBESS -- This subroutine computes the Bessel functions of orders 0 and 1 which are used as starting values in BESSEL. A Wronskian check of computed values is contained in CBESS. If the check indicates that errors greater than 10^{-8} have occurred, and if IPRINT $\neq 0$, a printout similar to the following is made:

    WRONSKIAN CHECK FOR BESSELS, N = 0, 1, SHOWS AGREEMENT TO 10** -6. Z = 1.0000000 E+00 - 1.0000000 E+00

    The Wronskian relation used is

$$J_1(z)\,Y_0(z) - J_0(z)\,Y_1(z) = \frac{2}{\pi z} \quad .$$

3. SUBROUTINE BESSEL -- The printouts occurring in BESSEL itself are a direct result of the returned value of zero as calculated by the FUNCTION ZERO subprogram. These printouts are similar to the following:

    DIFF. EQN CHECK FOR BESSELS SHOWS AGREEMENT TO 10** -4. N = 53, Z = 7.0710678E-03 - 7.0710678E-03

ORGANIZATION

BESSEL, as a program, utilizes other subroutines and subprograms. These additional programs are listed below for reference, along with a brief description of their function and calling sequence. All of the routines are contained in the BESSEL subroutine deck or are contained in the CDC 6600 function library.

Subroutines:

CBESS	This subroutine computes the Bessel functions of orders 0 and 1 for any complex argument. Its calling sequence is

CALL CBESS (Z, JZ, J1, YZ, Y1, H2Z,
H21, IVALCHK, IPRINT)

The formal parameters Z, JZ, J1, YZ, Y1, H2Z, and H21 are all TYPE COMPLEX. Their definitions are obvious from the context. The formal parameters IVALCHK and IPRINT are TYPE INTEGER and have the same meaning as in BESSEL.

BKWRD This is a backward recursion routine for Bessel functions. The calling sequence is

CALL BKWRD (Z, RATIO, IDIM)

where Z is the argument, IDIM is the maximum beginning value of n, and RATIO is the returned value. The formula used is

$$\text{Ratio}(n) = 1/[(2n)/z - \text{Ratio}(n+1)] \quad .$$

Z and RATIO are TYPE COMPLEX, while IDIM is TYPE INTEGER. See Section V, ALGORITHMS, for a further description.

FRWRD This is a forward recursion routine for Bessel functions. The calling sequence is

CALL FRWRD (Z, RATIO, IDIM, R1)

Where Z, RATIO and IDIM have the same definitions as in BKWRD, and R1 is the starting value of the ratio, and is TYPE COMPLEX. The formula used is

$$\text{Ratio}(n+1) = \frac{(2n)\,[\text{Ratio}(n)]/z - 1}{\text{Ratio}(n)}$$

See Section V, ALGORITHMS, for a further description.

Functions:

ZERO This is a function which verifies solutions to the differential equation explained in Section III above. Its calling sequence is

ZERO (Z, N, A, AADD, APRIME, IPRINT)

The parameters have the following definitions:

Z	TYPE COMPLEX. This is the argument of the Bessel function.
N	TYPE INTEGER. This is the order of the Bessel function.
A	TYPE COMPLEX. This is the value of $a_n(z)$ as explained in Section III above.
AADD	TYPE COMPLEX. This is the value of $a_{n+1}(z)$ as explained in Section III above.
APRIME	TYPE COMPLEX. This is the value of $a_n'(z)$ as explained in Section III above.
IPRINT	TYPE INTEGER. This parameter has the same definition as that contained in BESSEL.

ARG This is a function which computes the value of the argument of a complex number $\underline{z}$ through use of the formula

$$\arg(z) = 2 \arctan \frac{y}{\sqrt{x^2+y^2} + x}$$

$$z = x + iy$$

Its calling sequence is ARG(Z) where Z is TYPE COMPLEX.

CEXP This is a TYPE COMPLEX function which computes the value of the exponential function for a TYPE COMPLEX argument Z. The calling sequence is CEXP(Z), and the formula used is

$$\exp(z) = \exp(x) [\cos y + i \sin y] \quad .$$

CCONJ This is a TYPE COMPLEX function which computes the complex conjugate of a TYPE COMPLEX argument Z. Its calling sequence is CCONJ(Z), and the formula used is

$$\bar{z} = x - iy \quad .$$

CCOS — This is a TYPE COMPLEX function which computes the value of the cosine function of a TYPE COMPLEX argument Z through use of the formula

$$\cos(z) = \frac{e^{iz} + e^{-iz}}{2} \quad .$$

CSIN — This is a TYPE COMPLEX function which computes the value of the sine function of a TYPE COMPLEX argument Z through use of the formula

$$\sin(z) = \frac{e^{iz} - e^{-iz}}{2i} \quad .$$

CLOG — This is a TYPE COMPLEX function which computes the logarithm of a TYPE COMPLEX argument Z through use of the formula

$$\ln(z) = \ln|z| + i \arg(z) \quad .$$

CSQRT — This is a TYPE COMPLEX function which computes the square root of a TYPE COMPLEX argument Z through use of the formula

$$\sqrt{z} = \sqrt{x^2 + y^2} \left(\cos \frac{\arg(z)}{2} + i \sin \frac{\arg(z)}{2} \right) \quad .$$

ALGORITHMS

The starting values for BESSEL are generated by the subroutine CBESS. For arguments $|z| \leq 6$, the Bessel functions of orders 0 and 1 are computed by the formulae:

$$J_n(z) = (\tfrac{1}{2}z)^n \sum_{k=0}^{\infty} \frac{(-\frac{1}{4}z^2)^k}{k!\,\Gamma(n+k+1)}$$

$$Y_n(z) = -\frac{(\frac{1}{2}z)^{-n}}{\pi}\sum_{k=0}^{n-1}\frac{(n-k-1)!}{k!}(\tfrac{1}{4}z^2)^k + \frac{2}{\pi}\ln(\tfrac{1}{2}z)\,J_n(z)$$

$$-\frac{(\frac{1}{2}z)^{n}}{\pi}\sum_{k=0}^{\infty}\psi(k+1)+\psi(k+n+1)\frac{(-\frac{1}{4}z^2)^k}{k!\,(n+k)!}$$

$$H_n^{(2)}(z) = J_n(z) - iY_n(z)$$

where $$\psi(n) = -\gamma + \sum_{k=1}^{n-1} k^{-1}$$

$$\gamma = 0.5772156649 \qquad \text{(Euler's Constant)}$$

$$\Gamma(n+1) = n!\ ,$$

$$n = 0, 1\ .$$

If $|z| > 6$ and $|\arg z| < \pi$, an asymptotically converging series approach is used. The formulae are:

$$J_n(z) = \sqrt{\frac{2}{\pi z}}\left\{P(n,z)\cos X - Q(n,z)\sin X\right\}$$

$$Y_n(z) = \sqrt{\frac{2}{\pi z}}\left\{P(n,z)\sin X + Q(n,z)\cos X\right\}$$

$$H_n^{(2)}(z) = \sqrt{\frac{2}{\pi z}}\left\{P(n,z) - iQ(n,z)\right\}\exp(-iX)$$

where $$P(n,z) = 1 - \frac{(\mu-1)(\mu-9)}{2!\,(8z)^2} + \frac{(\mu-1)(\mu-9)(\mu-25)(\mu-49)}{4!\,(8z)^4} - \cdots$$

$$Q(n,z) = \frac{\mu-1}{8z} - \frac{(\mu-1)(\mu-9)(\mu-25)}{3!\,(8z)^3} + \cdots$$

$$\mu = 4n^2$$

$$X = z - (\tfrac{1}{2}n + \tfrac{1}{4})\pi \quad ,$$

$$n = 0, 1$$

The resulting values for $J_0(z)$, $Y_0(z)$, $J_1(z)$ and $Y_1(z)$ are then checked for accuracy through use of the Wronskian relation

$$J_1(z)\ Y_0(z) - Y_0(z)\ J_1(z) = \frac{2}{\pi z} \quad .$$

The agreement must be less than or equal to 10^{-8} in magnitude. If this criterion is not met, the quantity IVALCHK is set equal to <u>one</u>, and if IPRINT is equal to <u>one</u>, an error printout is made.

The various derivatives are calculated through use of the formula

$$Z_n'(z) = \frac{n}{z}\ Z_n(z) - Z_{n+1}(z)$$

where $Z_n(z)$ is any of the functions $J_n(z)$, $Y_n(z)$ or $H_n^{(2)}(z)$.

The calculated values are then checked for accuracy of one part in 10^8 or better by verifying that the differential equation

$$z\,[Z_{n+1}(z) + Z_n'(z)] - nZ_n(z) = 0$$

is satisfied. If this criterion is not met, IVALCHK is set equal to one, and if IPRINT $\neq 0$, an error printout is made.

Subroutine BESSEL uses the starting values obtained from CBESS in forward and backward recursion formulae to generate Bessel functions of large orders. These relationships are developed as follows: For Bessel functions of the first kind, the recurrence relation

$$J_{n+1}(z) = \frac{2n}{z} J_n(z) - J_{n-1}(z)$$

may be rewritten as

$$B_n(z) = \left[\frac{2n}{z} - B_{n+1}(z) \right]^{-1}$$

where $B_n(z)$ is a ratio of Bessel functions:

$$B_n(z) = \frac{J_n(z)}{J_{n-1}(z)} .$$

The subroutine BKWRD begins by setting $B_{n+1}(z) = 0$, and proceeds through the recursion from $n \geq 20 * N$, where N is the given value of the order, to $n = 1$. Each calculated value of $B_n(z)$ is placed in an array which is in labeled COMMON between BESSEL and BKWRD.

Upon return to the subroutine BESSEL, the recurrence relation

$$J_{n+1}(z) = B_{n+1}(z) \cdot J_n(z)$$

is then used to obtain the desired Bessel function.

For Bessel functions of the second kind, the recurrence relation

$$Y_{n+1}(z) = \frac{2n}{z} - Y_{n-1}(z)$$

may be rewritten as

$$F_{n+1}(z) = \frac{\frac{2n}{z} F_n(z) - 1}{F_n(z)}$$

where

$$F_n(z) = \frac{Y_n(z)}{Y_{n-1}(z)} .$$

The subroutine FRWRD uses as a starting value the ratio of Neumann functions

$$F_1(z) = \frac{Y_1(z)}{Y_0(z)}$$

obtained from BESSEL, and calculates values of the forward ratios from $n = 1$ to $n = N+2$, where N is the given value of the order. Each value of $F_n(z)$ as calculated is placed in an array which is in labeled COMMON between BESSEL and FRWRD.

Upon return to the subroutine BESSEL, the recurrence relation

$$Y_{n+1}(z) = F_{n+1}(z) \cdot Y_n(z)$$

is used to obtain the desired Neumann function.

For real arguments only, the Hankel functions of the second kind are calculated as

$$H_n^{(2)}(z) = J_n(z) - iY_n(z) \quad .$$

If the argument is <u>not</u> real, and $|\arg z| < \pi$, the subroutine FRWRD is used with a starting value

$$F_1(z) = \frac{H_1^{(2)}(z)}{H_0^{(2)}(z)}$$

obtained from BESSEL. Returned values are contained in an array in labeled COMMON between BESSEL and FRWRD.

EXTENDED USE IN TABLE GENERATION

Because of the method used to calculate the Bessel, Neumann, and Hankel functions, it will be noted that these functions are calculated for <u>all</u> integer values of the order from 0 to N, where N is the given value of the order. If it is desired to produce an array of Bessel, Neumann and/or Hankel functions,

the recurrence relations as previously described may be used by the main routine along with the ratio values as contained in labeled COMMON. The labeled COMMON is

```
COMMON/ RATIO/ B(2000), FY(1000), FH(1000)
COMPLEX B, FY, FH
```

where B contains the Bessel ratios, FY contains the Neumann ratios, and FH contains the Hankel ratios.

SUMMARY

The subroutine BESSEL, providing values for the Bessel, Neumann, and Hankel functions and their derivatives, of order n and argument z, uses approximately 2750 computer words. In addition, a COMMON block length of 8000 words is required for full utilization of the capabilities of the subroutine. On the average, approximately 15 msec/call is required. These figures are based on the use of a CDC 6600 computer. Accuracy has been verified through use of the tables contained in Handbook of Mathematical Functions, AMS 55, M. Abramowitz and I. A. Stegun, eds; National Bureau of Standards, November 1964 (with corrections) and Tables of the Bessel-Kelvin Functions ber, bei, ker, kei, and Their Derivatives for the Argument and Range 0(0.01) 107.50, H. H. Lowell, National Aeronautics and Space Administration, TR R-32, 1959.

ACKNOWLEDGEMENTS

The routine CBESS, which generates the starting values used in BESSEL, was originally written by R. W. Murray and J. W. Garner, both of The Dikewood Corporation. In addition, the assistance of D. E. Brannon in the proper use of the recursion formulae is gratefully acknowledged.

APPENDIX: PROGRAM LISTING

```
C
C
C
      SUBROUTINE BESSEL (N,Z,J,Y,H2,JPRIME,YPRIME,H2PRIME,IVALCHK,
     $IPRINT)
C
      EQUIVALENCE (JNZ,J),(YNZ,Y),(HNZ,H2),(JNZPRM,JPRIME),(YPRIME,
     $YNZPRM),(H2PRIME,HNZPRM),(IVALCHK,VALCHK),(IPRINT,PRINT)
      COMPLEX J,Y,H2,JPRIME,YPRIME,H2PRIME
      COMMON/RATIO/B(2000),FY(1000),FH(1000)
      TYPE COMPLEX       CSQRT,CLOG,CCOS,CSIN,CEXP,CCONJ
      COMPLEX Z,JNZ,YNZ,HNZ,JNZPRM,YNZPRM,HNZPRM,J2,J1,Y2,Y1,H2,H1,FY,
     $FH,CONST,PI,WRONSK,JNZADD,YNZADD,HNZADD,HNONEZ,HNONEA
      REAL MAX
      INTEGER PRINT,CHECK,VALCHK
C
C                  -PI.LT.ARG (Z).LT.+PI FOR PROPER OPERATION
C                  WILL ACCEPT NEGATIVE N,ZERO Z
C
      VALCHK=0
      MAX=SQRT(2.0)*(1.0E150)
      IF (N.LT.0) GO TO 5
      IRETURN=1
      IF (N.GT.1) GO TO 3
      CALL CRESS (Z,J2,J1,Y2,Y1,H2,H1,CHECK,PRINT)
      IF (CHECK.NE.0) VALCHK=1
      IF (N.EQ.1) GO TO 1
      JNZ=J2
      YNZ=Y2
      HNZ=H2
      JNZPRM=-J1
      YNZPRM=-Y1
      HNZPRM=-H1
      RETURN
1     JNZ=J1
      YNZ=Y1
      HNZ=H1
      IF (Z.EQ.(0.,0.)) GO TO 2
      JNZPRM=J2-(1./Z)*J1
      YNZPRM=Y2-(1./Z)*Y1
      HNZPRM=H2-(1./Z)*H1
      RETURN
2     JNZPRM=0.
      YNZPRM=0.
      HNZPRM=0.
      RETURN
3     IF (Z.EQ.(0.,0.)) GO TO 5
      CALL CRESS (Z,J2,J1,Y2,Y1,H2,H1,CHECK,PRINT)
      IF (CHECK.NE.0) VALCHK=1
      IDIM=N*20
      IF (IDIM.LT.200) IDIM=200
      IF (IDIM.GT.2000) IDIM=2000
```

```
      CALL BKWRD(Z,F,IDIM)
      JNZ=JZ
      DO 4 I=1,N
      JNZ=JNZ*B(I)
4     CONTINUE
      JNZPRM=(N/Z)*JNZ+(-1.,0.)*(JNZ*B(N+1))
      R1=Y1/YZ
      IDIM=N+2
      IF (IDIM.GT.1000) IDIM=1000
      CALL FRWRD(Z,FY,IDIM,R1)
      YNZ=YZ
      DO 9 I=1,N
      YNZ=YNZ*FY(I)
9     CONTINUE
      YNZPRM=(N/Z)*YNZ-YNZ*FY(N+1)
      JNZADD=JNZ*B(N+1)
      YNZADD=YNZ*FY(N+1)
      IF (AIMAG(Z).EQ.0.) GO TO 11
      R1=H1/HZ
      IDIM=N+2
      IF (IDIM.GT.1000) IDIM=1000
      CALL FRWRD(Z,FH,IDIM,R1)
      HNZ=HZ
      DO 10 I=1,N
      HNZ=HNZ*FH(I)
10    CONTINUE
      HNZPRM=(N/Z)*HNZ-HNZ*FH(N+1)
      GO TO 30
11    HNZ=JNZ+(0.,-1.)*YNZ
      HNZPRM=(N/Z)*HNZ-(JNZ*B(N+1)+(0.,-1.)*(YNZ*FY(N+1)))
20    HNZADD=(JNZ*B(N+1))+(0.,-1.)*(YNZ*FY(N+1))
      GO TO 32
30    HNZADD=HNZ*FH(N+1)
C
C                    --DIFFERENTIAL EQUATION CHECKS--
C                    THE EQUATION USED IS Z*(A(N+1)+APRIME(N))-N*A(N)=0,
C                    WHERE N=ORDER, Z=ARGUMENT, AND A(N)=J(N,Z) OR Y(N,Z)
C                    OR H2(N,Z).
C
32    DIFF= ABS(ZERO(Z,N,JNZ,JNZADD,JNZPRM,PRINT))
      IF (DIFF.GT.1.0E-8) GO TO 12
50    DIFF= ABS(ZERO(Z,N,YNZ,YNZADD,YNZPRM,PRINT))
      IF (DIFF.GT.1.0E-8) GO TO 50
900   DIFF= ABS(ZERO(Z,N,HNZ,HNZADD,HNZPRM,PRINT))
      IF (DIFF.GT.1.0E-8) GO TO 33
      GO TO (8,7),IRETURN
33    IF (PRINT.EQ.0) GO TO (8,7),IRETURN
      IF (DIFF.GT.1.0E-6) GO TO 35
      PRINT 34,N,Z
34    FORMAT (5HNODIFF. FOR CHECK FOR HANKELS SHOWS AGREEMENT TO 10**-8.
     $ N=I3,4H, Z=2E16.7)
      GO TO (8,7),IRETURN
```

```
35     IF (DIFF.GT.1.0E-4) GO TO 37
       PRINT 36,N,Z
36     FORMAT (58HODIFF. EQN CHECK FOR HANKELS SHOWS AGREEMENT TO 10**-4.
     $ N=I3,4H, Z=2E16.7)
       GO TO (8,7),IRETURN
37     IF (DIFF.GT.1.0E-2) GO TO 39
       PRINT 38,N,Z
38     FORMAT (58HODIFF. EQN CHECK FOR HANKELS SHOWS AGREEMENT TO 10**-2.
     $ N=I3,4H, Z=2E16.7)
       GO TO (8,7),IRETURN
39     PRINT 40,N,Z
40     FORMAT (66HODIFF. EQN CHECK FOR HANKELS DOES NOT SHOW AGREEMENT T
     $ 10**-2. N=I3,4H, Z=2E16.7/)
       GO TO (8,7),IRETURN
12     IF(PRINT.EQ.0) GO TO 90
       IF (DIFF.GT.1.0E-6) GO TO 14
       PRINT 13,N,Z
13     FORMAT (58HODIFF. EQN CHECK FOR BESSELS SHOWS AGREEMENT TO 10**-6.
     $ N=I3,4H, Z=2E16.7)
       GO TO 90
14     IF (DIFF.GT.1.0E-4) GO TO 16
       PRINT 15,N,Z
15     FORMAT (58HODIFF. EQN CHECK FOR BESSELS SHOWS AGREEMENT TO 10**-4.
     $ N=I3,4H, Z=2E16.7)
       GO TO 90
16     IF (DIFF.GT.1.0E-2) GO TO 18
       PRINT 17,N,Z
17     FORMAT (58HODIFF. EQN CHECK FOR BESSELS SHOWS AGREEMENT TO 10**-2.
     $ N=I3,4H, Z=2E16.7)
       GO TO 90
18     PRINT 19,N,Z
19     FORMAT (66HODIFF. EQN CHECK FOR BESSELS DOES NOT SHOW AGREEMENT TO
     $ 10**-2. N=I3,4H, Z=2E16.7/)
       GO TO 90
50     IF (PRINT.EQ.0) GO TO 900
       IF (DIFF.GT.1.0E-6) GO TO 52
       PRINT 51,N,Z
51     FORMAT (69HODIFF. EQN. CHECK FOR NEUMANN FUNCTIONS SHOWS AGREEMENT
     $ TO 10**-6. N=I3,4H, Z=2E16.7)
       GO TO 900
52     IF (DIFF.GT.1.0E-4) GO TO 54
       PRINT 53,N,Z
53     FORMAT (69HODIFF. EQN. CHECK FOR NEUMANN FUNCTIONS SHOWS AGREEMENT
     $ TO 10**-4. N=I3,4H, Z=2E16.7)
       GO TO 900
54     IF (DIFF.GT.1.0E-2) GO TO 56
       PRINT 55,N,Z
55     FORMAT (69HODIFF. EQN. CHECK FOR NEUMANN FUNCTIONS SHOWS AGREEMENT
     $ TO 10**-2. N=I3,4H, Z=2E16.7)
       GO TO 900
56     PRINT 57,N,Z
57     FORMAT (77HODIFF. EQN. CHECK FOR NEUMANN FUNCTIONS DOES NOT SHOW A
```

```
     $GREEMENT TO 10**-2. N=T3,4H, Z=2E16.7/)
      GO TO 900
5     JNZ=(0.,0.)
      YNZ=(-1.E300,0.)
      HNZ=(1.E300,0.)
      JNZPRM=(0.,0.)
      YNZPRM=(1.E300,0.)
      HNZPRM=(-1.E300,0.)
      GO TO (8,7),IRETURN
6     IRETURN=2
      N=-N
      GO TO 3
7     IF ((CABS(JNZ).GT.MAX).OR.(CABS(YNZ).GT.MAX).OR.(CABS(HNZ).GT.MAX)
     $.OR.(CABS(JNZPRM).GT.MAX).OR.(CABS(YNZPRM).GT.MAX).OR.(CABS(HNZPRM
     $).GT.MAX)) VALCHK=1
      IF(((N-(N/2))-(N/2)).EQ.0) RETURN
      JNZ=-JNZ
      YNZ=-YNZ
      HNZ=-HNZ
      JNZPRM=(N/Z)*JNZ+(JNZ*B(N+1))
      YNZPRM=(N/Z)*YNZ+(YNZ*FY(N+1))
      HNZPRM=JNZPRM+(0.,-1.)*YNZPRM
      RETURN
8     IF ((CABS(JNZ).GT.MAX).OR.(CABS(YNZ).GT.MAX).OR.(CABS(HNZ).GT.MAX)
     $.OR.(CABS(JNZPRM).GT.MAX).OR.(CABS(YNZPRM).GT.MAX).OR.(CABS(HNZPRM
     $).GT.MAX)) VALCHK=1
      RETURN
      END
C
C
C
      FUNCTION ZERO (Z,N,A,AADD,APRIME,IPRINT)
C
      EQUIVALENCE (IPRINT,PRINT)
      INTEGER PRINT
      COMPLEX Z,A,AADD,APRIME,FACT1,FACT2,RATIO
      FACT1=Z*(AADD+APRIME)
      FACT2=N*A
      RATIO=FACT1/FACT2
      ZERO=1.0000C0000001-CABS(RATIO)
      IF (ABS(ZERO).LT.1.0E-08) RETURN
      RATIO=REAL(FACT1)/REAL(FACT2)+(0.,1.)*(AIMAG(FACT1)/AIMAG(FACT2))
      ZERO=1.000000000001-CABS(RATIO)
      IF (ABS(ZERO).LT.1.0E-08) RETURN
      IF (PRINT.EQ.0) GO TO 5
      PRINT 1, ZERO,Z,A,AADD,APRIME
1     FORMAT (11HOZERO PRINT,3E20.10/6E20.10)
      PRINT 2, FACT1,FACT2
2     FORMAT (4E20.10)
5     FACT1R=REAL(FACT1)
      FACT1I=AIMAG(FACT1)
      FACT2R=REAL(FACT2)
```

```
      FACT2I=AIMAG(FACT2)
      L1RE=FACT1R      .AND.03777000000000000000
      L1IE=FACT1I       .AND.03777000000000000000
      L2RE=FACT2R      .AND.03777000000000000000
      L2IE=FACT2I       .AND.03777000000000000000
      L1RM=FACT1R      .AND.07777777777777777
      L1IM=FACT1I       .AND.07777777777777777
      L2RM=FACT2R      .AND.07777777777777777
      L2IM=FACT2I       .AND.07777777777777777
      IF ((L1RE.NE.L2RE).OR.(L1IE.NE.L2IE)) GO TO 3
      ZERO=CABS((L1RM-L2RM)+(0.0,1.0)*(L1IM-L2IM))*1.0E-9
      IF (ABS(ZERO).LT.1.0E-08) RETURN
3     IF (PRINT.EQ.0) RETURN
      PRINT 4,FACT1R,L1RE,L1RM,FACT2R,L2RE,L2RM,FACT1I,L1IE,L1IM,
     $FACT2I,L2IE,L2IM
4     FORMAT (8H FACT1R=3(020,3X)/8H FACT2R=3(020,3X)//8H FACT1I=3(020,
     $3X)/8H FACT2I=3(020,3X))
      RETURN
      END
C
C
C
      SUBROUTINE BKWRD (Z,RATIO,IDIM)
C
      DIMENSION RATIO(IDIM)
      COMPLEX RATIO,KONST,DENOM,R1,Z
C                   COMPUTE CONSTANTS, PRESET ARRAY
      KONST(I)=2.*I/Z
      DO 1 J=1,IDIM
      RATIO(J)=(0.,0.)
1     CONTINUE
      I=IDIM-1
      RATIO(IDIM)=Z/(2.*IDIM)
C                   COMPUTE RATIOS
2     DENOM=KONST(I)-RATIO(I+1)
      IF (DENOM) 3,4,3
3     RATIO(I)=1./DENOM
      GO TO 5
4     RATIO(I)=1.0E300
5     I=I-1
      IF (I) 6,6,2
6     RETURN
      END
C
C
C
      SUBROUTINE FRWRD (Z,RATIO,IDIM,R1)
C
      DIMENSION RATIO(IDIM)
      COMPLEX RATIO,R1,Z,KONST
      KONST(I)=2.*I/Z
      IMAX=IDIM-1
```

```
      RATIO(1)=R1
      DO 7 I=1,IMAX
      RATIO(I+1)=(RATIO(I)*KONST(I)-(1.,0.))/RATIO(I)
7     CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE CBESS (Z,JZ,J1,YZ,Y1,H2Z,H21,IVALCHK,IPRINT)
C
      EQUIVALENCE (JZERO,JZ),(JONE,J1),(YZERO,YZ),(H2ZERO,H2Z),(H2ONE,
     $H21),(IVALCHK,CHECK),(IPRINT,PRINT)
      COMPLEX JZ,J1,YZ,Y1,H2Z,H21
      INTEGER PRINT,CHECK
           COMPLEX Z,JZERO,JONE,YZERO,YONE,ZSQ,FACT,ZFACT,JZADD,J1ADD
     $,EZ,P,Q,COSP,SINP,H2ZERO,H2ONE,CONST,WRONSK
      TYPE COMPLEX       CSQRT,CLOG,CCOS,CSIN,CEXP
      REAL MAX
      MAX=SQRT(2.0)*(1.0E150)
      CHECK=0
      IF(CABS(Z)- 6.)1,1,10
1     JZERO=(1.,0.)
      JONE=(1.,0.)
      YZERO=(0.,0.)
      YONE=(1.,0.)
      FK=1.
      FKFACT=1.
      ZSQ=-Z*Z*.25
      ZFACT=(1.,0.)
      SKINV=1.
2     ZFACT=ZFACT*ZSQ
      FACT=ZFACT/FKFACT
      JZADD=FACT/FKFACT
      FK=FK+1.
      FKFACT=FKFACT*FK
      J1ADD=FACT/FKFACT
      JZERO=JZERO+JZADD
      JONE=JONE+J1ADD
      YZERO=YZERO+JZADD*SKINV
      YONE=YONE+J1ADD*(SKINV+SKINV+1./FK)
      SKINV=SKINV+1./FK
      IF (CABS(JZADD/JZERO).GT.1.0E-25) GO TO 2
      IF (CABS(J1ADD/JONE).GT.1.0E-25) GO TO 2
      JONE=JONE*Z*0.5
      IF (Z.EQ.(0.,0.)) GO TO 5
      YZERO=((.5772156649+CLOG(Z*.5))*JZERO-YZERO)/1.570796326795
      YONE=((.5772156649+CLOG(Z*.5))*JONE-(1.-ZSQ*YONE)/Z)/1.570796326794
     15
      H2ZERO=JZERO+(0.,-1.)*YZERO
      H2ONE =JONE +(0.,-1.)*YONE
      GO TO 6
```

```
5       YZERO=(-1.E300,0.)
        YONE=(-1.E300,0.)
        H2ZERO=(1.E300,0.)
        H2ONE=(1.E300,0.)
        CHECK=1
        RETURN
C                       CABS(Z).GT.6  **  -PI.LT.ARG(Z).LT.PI FOR PROPER OP.
10      FACT=3.141592653589*Z
        FACT=CSQRT(FACT)
        COSP=CCOS(Z)/FACT
        SINP=CSIN(Z)/FACT
        ZFACT=(1.,1.)*CEXP((0.,-1.)*Z)/FACT
        U=0.
        EZ=8.0*Z
200     FN=1.
        FK=1.
        P=1.
        Q=(U-1.)/EZ
        FACT=Q
15      FN=FN+2.0
        FK=FK+1.
        FACT=-FACT*(U-FN*FN)/EZ/FK
        P=P+FACT
        FN=FN+2.
        FK=FK+1.
        FACT=FACT*(U-FN*FN)/EZ/FK
        Q=Q+FACT
        IF (CABS(FACT/Q).LT.1.0E-8) GO TO 40
        IF (FK.LT.21.0) GO TO 15
40      IF(U) 300,300,400
300     JZERO=(P+Q)*COSP+(P-Q)*SINP
        YZERO=(P+Q)*SINP-(P-Q)*COSP
        H2ZERO=ZFACT*(P+(0.,-1.)*Q)
        U=4.
        GO TO 200
400     JONE=(P+Q)*SINP-(P-Q)*COSP
        YONE=-(P+Q)*COSP-(P-Q)*SINP
        H2ONE=ZFACT*(Q+(0., 1.)*P)
C
C                       WRONSKIAN CHECK
C
6       CONST=(2.,0.)/(3.1415926535897 93*Z)
        IF ((CABS(JZERO).GT.MAX).OR.(CABS(JONE).GT.MAX).OR.(CABS(YZERO).GT
       $.MAX).OR.(CABS(YONE).GT.MAX)) CHECK=1
        IF (PRINT.EQ.0) RETURN
        P=JONE*YZERO
        Q=JZERO*YONE+CONST
        WRONSK=P/Q
        DIFF=1.000000000001-CABS(WRONSK)
        IF ( ABS(DIFF).LT.1.0E-8) RETURN
        IF ( ABS(DIFF).GT.1.0E-6) GO TO 8
        PRINT 7,Z
```

```
7       FORMAT (66HOWRONSKIAN CHECK FOR BESSELS, N=0,1, SHOWS AGREEMENT TO
     $ 10**-6. Z=2E16.7)
        RETURN
8       IF ( ABS(DIFF).GT.1.0E-4) GO TO 11
        PRINT 9,Z
9       FORMAT (66HOWRONSKIAN CHECK FOR BESSELS, N=0,1, SHOWS AGREEMENT TO
     $ 10**-4. Z=2E16.7)
        RETURN
11      IF ( ABS(DIFF).GT.1.0E-2) GO TO 13
        PRINT 12,Z
12      FORMAT (66HOWRONSKIAN CHECK FOR BESSELS, N=0,1, SHOWS AGREEMENT TO
     $ 10**-2. Z=2E16.7)
        RETURN
13      PRINT 14,Z
14      FORMAT (74HOWRONSKIAN CHECK FOR BESSELS, N=0,1, DOES NOT SHOW AGREE
     $MENT TO 10**-2. Z=2E16.7/)
        RETURN
        END
C
C
C
        FUNCTION ARG(Z)
C
        TYPE COMPLEX Z
        X=REAL(Z)
        Y=AIMAG(Z)
        IF(Y) 1,2,1
2       IF(X) 3,4,4
3       ARG=3.1415926539
        RETURN
4       ARG=0.0
        RETURN
1       ARG=2.0*ATAN(Y/(SQRT(X*X+Y*Y)+X))
        RETURN
        END
C
C
C
        COMPLEX FUNCTION CEXP(Z)
C
        TYPE COMPLEX Z,I
        I=(0.,1.)
        X=REAL(Z)
        Y=AIMAG(Z)
        IF (Y) 1,2,1
1       CEXP=EXP(X)*(COS(Y)+I*SIN(Y))
        RETURN
2       CEXP=EXP(X)
        RETURN
        END
C
C
```

```
C
      COMPLEX FUNCTION CCONJ(Z)
C
      TYPE COMPLEX Z,I
      I=(0.,1.)
      X=Z
      Y=Z*I
      CCONJ=X+I*Y
      RETURN
      END
C
C
C
      COMPLEX FUNCTION CCOS(Z)
C
      TYPE COMPLEX Z,I,EIZ
      TYPE COMPLEX CEXP
      IF (AIMAG(Z)) 2,1,2
1     CCOS=COS(REAL(Z))
      RETURN
2     I=(0.,1.)
      EIZ=CEXP(I*Z)
      CCOS=(EIZ+1./EIZ)/2.
      RETURN
      END
C
C
C
      COMPLEX FUNCTION CSIN(Z)
C
      TYPE COMPLEX Z,I,EIZ
      TYPE COMPLEX CEXP
      IF (AIMAG(Z)) 2,1,2
1     CSIN=SIN(REAL(Z))
      RETURN
2     I=(0.,1.)
      EIZ=CEXP(I*Z)
      CSIN=(EIZ-1./EIZ)/(2.*I)
      RETURN
      END
C
C
C
      COMPLEX FUNCTION CLOG(Z)
C
      TYPE COMPLEX Z,I
      I=(0.,1.)
      CLOG=ALOG(CABS(Z))+I*ARG(Z)
      RETURN
      END
C
C
```

```
C
      COMPLEX FUNCTION CSQRT(Z)
C
      TYPE COMPLEX Z,I
      I=(0.,1.)
      THETA=ARG(Z)*0.5
      CSQRT=SQRT(CABS(Z))*(COS(THETA)+I*SIN(THETA))
      RETURN
      END
C
C
C
```