Mathematics Notes

Note 4

7 October 1969

SBF

A Subroutine for the Generation of Spherical Bessel Functions With Real Arguments and Positive Integer Orders

Joe P. Martinez

The Dikewood Corporation

ABSTRACT

This note describes a computer subroutine which generates the spherical Bessel functions $j_n(x)$, $y_n(x)$, $h_n^{(1)}(x)$, and $h_n^{(2)}(x)$ as well as their first derivatives. The use of the subroutine in generating tables of values of the functions and their derivatives is outlined. Various methods of generating the functions and the problems encountered are discussed.

I. INTRODUCTION

SBF is a computer subroutine written in FORTRAN IV for the Control Data Corporation 6600 computer in the Air Force Weapons Laboratory at Kirtland Air Force Base. The spherical Bessel functions $j_n(x)$ and $y_n(x)$ are generated from recurrence relationships, then the Hankel functions $h_n^{(1)}(x)$ and $h_n^{(2)}(x)$ are calculated from the relationship between the j and y functions. A separate subroutine computes the first derivatives of the functions. An option is provided whereby another subroutine is called for an accuracy check; this subroutine computes the Wronskian relationship between $j_n(x)$ and $y_n(x)$.

In developing a technique to generate the spherical Bessel functions several methods were tried, but computer round-off error was encountered with most methods. A section of various algorithms which were tested is included.

II. OPERATION

General

SBF is used in conjunction with a main program or subroutine which assigns values to the order and argument of the functions and calls the subroutine. Arrays of values for the functions and their derivatives are returned by SBF to the calling routine through the formal parameter list and must be typed and dimensioned in the calling routine. The value of the order n must be greater than or equal to 1; if the functions of order 0 are desired they may be obtained through the use of the labeled COMMON block ZERO, to be described later. The argument x must be type REAL and its absolute value be greater than or equal to .0001.

Calling Sequence

To use Subroutine SBF the calling routine must supply the standard FORTRAN IV CALL statement,

CALL SBF (N,X,J,Y,JP,YP,H1,H2,H1P,H2P,IW)

The parameters N, X, and IW are supplied to SBF by the calling routine and the rest of the parameters are returned to the calling routine by SBF.

Parameters

1. N — TYPE INTEGER. This is the highest order desired for the spherical Bessel functions and their first derivatives. The subroutine returns all the functions between 1 and N. N must be ≥ 1.

2. X — TYPE REAL. X is the argument of the spherical Bessel functions and it must be in the range $-.0001 \geq x \geq .0001$.

The next eight parameters are arrays of values for the functions and their derivatives which are returned to the calling routine by SBF. They must all be dimensioned by the calling routine. Since the arrays will return N

values of the functions, from n = 1 to n = N, they must each be dimensioned for at least N storage locations. The arrays must also be of the same type, that is, REAL or COMPLEX.

3. J — TYPE REAL. This is an array of values for $j_n(x)$, the spherical Bessel functions of the first kind.

4. Y — TYPE REAL. This array returns values for the spherical Bessel functions of the second kind, $y_n(x)$.

5. JP — TYPE REAL. This is an array of values for $dj_n(x)/dx$, the first derivative of the j functions.

6. YP — TYPE REAL. This is an array of values for $dy_n(x)/dx$, returned by SBF.

7. H1 — TYPE COMPLEX. This is an array of values for the spherical Bessel functions of the third kind, $h_n^{(1)}(x)$, also known as the spherical Hankel functions of the first kind.

8. H2 — TYPE COMPLEX. This array returns values for the spherical Bessel functions of the fourth kind, $h_n^{(2)}(x)$. This function is also the spherical Hankel functions of the second kind.

9. H1P — TYPE COMPLEX. This is an array of values for $dh_n^{(1)}(x)/dx$.

10. H2P — TYPE COMPLEX. This is an array of values for $dh_n^{(2)}(x)/dx$.

11. IW — TYPE INTEGER. This parameter is a flag used to determine whether a separate subroutine is to be called from SBF. The subroutine computes the Wronskian for each pair of $j_n(x)$ and $y_n(x)$ values, to be used as

an accuracy criteria. If IW = 0 the subroutine will not be called. Any non-zero integer number will cause the Wronskian subroutine to be called.

The names given to the above parameters were chosen for illustrative purposes and are formal parameters in the subroutine. The actual parameters used by the calling routine need not agree in name, although they must agree in type and number.

III. ORGANIZATION

Subroutine SBF uses two additional subroutines which are called from SBF, as well as several computer system functions.

Subroutines

PRIME — This subroutine is always called from SBF in order to compute the first derivatives of the spherical Bessel functions of all four kinds. Communications between PRIME and SBF is through a parameter list as well as through the labeled COMMON block, ZERO, which contains the functions of zero order. PRIME uses as inputs N, X, J, Y, H1, and H2 and returns JP, YP, H1P, and H2P. The parameters are the same as described for Subroutine SBF. The calling sequence is:

```
CALL PRIME (N,X,J,Y,JP,YP,H1,H2,H1P,H2P)
```

WRON — If the parameter IW in the argument list for SBF is set to any non-zero integer number this subroutine will be called. WRON calculates the Wronskian relationship between the j and y functions and their derivatives, from values computed by SBF. After the Wronskian is calculated it is divided by the analytic value of the Wronskian. This allows a comparison of this function against 1, the desired value. The largest error is selected from the calculations and a message similar to the following is printed:

```
WITH THE ARGUMENT = 100.0000, AND ORDER N = 1 THROUGH
N = 1150, THE WRONSKIAN CHECK SHOWS A MAXIMUM RELATIVE
ERROR OF 4.61853E-13, OCCURRING AT N = 98.
```

This message gives the relative error of the normalized Wronskian against 1, and tells where the maximum error occurs. Communication between WRON and SBF is through a parameter list. The calling sequence is:

```
CALL WRON (N,X,J,Y,JP,YP)
```

The parameters in the argument list are all inputs to WRON; they are the same as described for Subroutine SBF.

Functions

The following is a list of the external and intrinsic functions used by SBF and its associated subroutines. These functions are supplied by the CDC 6600 computer system itself.

SIN	FLOAT
COS	CMPLX
ABS	IFIX
MAXO	MOD

Input/Output Files

Only an output file is required by SBF, due to the printed message in WRON.

IV. GENERAL

Timing

The computation speed of this subroutine is very fast when used with the CDC 6600 computer. For example, the entire set of tables for the spherical Bessel functions contained in the Handbook of Mathematical Functions, AMS 55, was calculated in 2.4 seconds. In a typical run it computed the arrays of values for one argument for n = 1 to n = 1000 in approximately .1 seconds. In one call of the subroutine, therefore, each order requires about 10^{-4} seconds of central processor time. This timing will probably differ with other machines.

Accuracy

The functions generated by the routine were compared against tables contained in the references at the end of this note and no discrepancies were noted for any of the values listed in the tables. For the argument x = 10000.0 and n ranging from n = 1 to n = 1150, the Wronskian check shows a maximum error of .02 percent in the Wronskian relationship.

Range

The range of the argument x is limited for small absolute values. Subroutine SBF was tested at $x = 10^{-4}$ with good results, but smaller values may cause problems due to the large absolute values of the y functions at small arguments. Zero may not be used as an argument because division by the argument occurs in the subroutine and this would cause computer overflow problems. Large values of the argument were also tested with good results. Values tested included $x = 10^{4}$; values larger than this were not used, so the accuracy is undetermined for the larger values. The order n was taken to n = 1150 with good results and values larger than this should produce good results also. It should be mentioned here that at $n \geq |x|$ the absolute value of the j functions start decreasing and the absolute value of the

y functions start increasing; after this point the functions are calculated only to the machine limits. That is, the j functions are calculated to the point where $|j_n(x)| \cong 10^{-308}$ and then the rest of the functions, to n = N, are set to 0.0. Similarly, the y functions are calculated to the point where $|y_n(x)| \cong 10^{308}$ and the rest of the functions are set to $\pm 10^{308}$, depending on whether x is negative and the order is even or odd. These machine limits, $\pm 10^{308}$ and $\pm 10^{-308}$, are based on the CDC 6600 computer, but may be different for other computers. The value of n at which point the functions are set to 0 or machine infinity is stored in the variable ISAVE and may be used, if desired, in the calling routine by use of the labeled COMMON block RR.

Functions of Zero Order

Communication between SBF and PRIME takes place through a parameter list and through the labeled COMMON block ZERO. This COMMON block may be used by the calling program after SBF has been called if the functions of zero order and their derivatives are desired. The elements of the COMMON block are as follows, in order.

1. JO — TYPE REAL. This is $j_o(x)$.
2. YO — TYPE REAL. This is $y_o(x)$.
3. JPO — TYPE REAL. This is $dj_o(x)/dx$.
4. YPO — TYPE REAL. This is $dy_o(x)/dx$.
5. H1O — TYPE COMPLEX. This is $h_o^{(1)}(x)$.
6. H2O — TYPE COMPLEX. This is $h_o^{(2)}(x)$.
7. H1PO — TYPE COMPLEX. This is $dh_o^{(1)}(x)/dx$.
8. H2PO — TYPE COMPLEX. This is $dh_o^{(2)}(x)/dx$.

Storage Requirements

The storage requirements needed to execute SBF and its associated subroutines with a main program depend primarily on the value of N, and the user will have to determine this. On the CDC 6600 the number of storage locations required for the three subroutines is approximately 1000_8 locations.

V. ALGORITHMS

Algorithm for the $j_n(x)$ Function

The spherical Bessel functions of the first kind were the most difficult to generate. This is because the absolute value of the functions decrease rapidly after $n \geq |x|$. Forward recurrence techniques introduced large machine round-off errors after this point. Several methods of generating the $j_n(x)$ were attempted. A section later in this note will summarize these methods. The method which was used by SBF, finally, was a backward recurrence formula. This method produced excellent results over a very large range of n and x. In initializing the recurrence relationship, the subroutine determines the larger of the two quantities, $|x| + 50$ and ISAVE + 10. ISAVE is the value of n, either at n = N or at the point where the $y_n(x)$ function goes to machine infinity. If N is smaller than the order at which $y_n(x)$ goes infinite then ISAVE will take on the value for N. If N is larger than the order for which $y_n(x)$ goes infinite, ISAVE will take on that value, as explained earlier. Once the larger of the two quantities mentioned above is determined, the value is stored in the variable IM. Then the function $j_{IM}(x)$ is set to 0, and the function $j_{IM-1}(x)$ is set to 10^{-75}. The backward recurrence formula

$$j_n(x) = \frac{(2n + 3)}{x} j_{n+1}(x) - j_{n+2}(x) \qquad (1)$$

is then carried out to n = 1. The calculated value for n = 1 is divided by the analytic value of $j_1(x)$ and this ratio is used to multiply all the calculated values to n = ISAVE. The proper relationship between the functions is established by the backward recurrence formula and the multiplication by the ratio between the artificially calculated n = 1 function and its exact value produces the actual functions.

Algorithm for the $y_n(x)$ Function

The spherical Bessel functions of the second kind are calculated from a forward recurrence formula. First, the $y_1(x)$ and $y_2(x)$ are found from analytic formulas and then the recurrence relation

$$f_n(x) = \frac{(2n - 1)}{x} f_{n-1}(x) - f_{n-2}(x), \tag{2}$$

where

$$f_n(x) = y_n(x),$$

is applied for the rest of the functions until n = N or the function approaches the machine limit. These functions are calculated by SBF prior to the calculation of the $j_n(x)$.

Algorithm for the Hankel Functions

The spherical Hankel functions are established as follows

$$h_n^{(1)}(x) = j_n(x) + iy_n(x) \tag{3}$$

and

$$h_n^{(2)}(x) = j_n(x) - iy_n(x) \tag{4}$$

Algorithm for the Derivatives of the Functions

The first derivatives for the spherical Bessel functions of all four kinds are computed from the following relationship

$$f'_n(x) = f_{n-1}(x) - \frac{(n + 1)}{x} f_n(x) \tag{5}$$

where $f_n(x)$ may be any of the four kinds of functions.

All of the above relationships are taken from Reference 1, the National Bureau of Standards, AMS 55, Handbook of Mathematical Functions.

The Wronskian Check

When the Subroutine WRON is called, the following relationship is calculated

$$W = j_n(x)\, y'_n(x) - j'_n(x)\, y_n(x) . \qquad (6)$$

The analytic value of the Wronskian is x^{-2}, so W is multiplied by x^2 in order to normalize the calculated value. The normalized value should be exactly 1.0, so this provides a convenient check for the accuracy of the functions and their derivatives.

VI. OTHER ALGORITHMS

Initially the calculation of the functions was done by solving for $h_1^{(1)}(x)$ and $h_2^{(1)}(x)$ analytically and applying the recurrence formula of Eq. (2) where $f_n(x) = h_n^{(1)}(x)$. The initial values used were

$$h_1^{(1)}(x) = \frac{\sin(x)}{x^2} - \frac{\cos(x)}{x} - i\left[\frac{\cos(x)}{x^2} + \frac{\sin(x)}{x}\right] \tag{7}$$

and

$$h_2^{(1)}(x) = \left(\frac{3}{x^3} - \frac{1}{x}\right)\sin(x) - \frac{3}{x^2}\cos(x) + i\left[\left(\frac{-3}{x^3} + \frac{1}{x}\right)\cos(x) - \frac{3}{x^2}\sin(x)\right] \tag{8}$$

The idea was to separate the real and imaginary parts of the Hankel functions to obtain the j and y functions in one process. The imaginary part, or y functions, agreed exactly with tables found in books, but the real part, the j functions, were not correct in many instances. Upon further investigation it was noted that the error was introduced at $n \geq |x|$ because the j functions start decreasing in absolute magnitude after this point and computer round-off error becomes significant. After several orders are computed beyond this point the round-off error causes the functions to be off by several orders of magnitude.

The recurrence formula of Eq. (2) was tried again, but this time the initial values which were used were

$$h_1^{(1)}(x) = -\left(\frac{x + i}{x^2}\right)e^{ix} \tag{9}$$

and

$$h_2^{(1)}(x) = -\left(\frac{-ix^2 + 3x + 3i}{x^3}\right)e^{ix} \tag{10}$$

The results were the same as with the case above. The $j_n(x)$ were also generated alone with Eq. (2), with similar results. The conclusion drawn from the above was that the $j_n(x)$ could be generated by a forward recurrence technique only to the point where $n \cong |x|$; beyond this, for larger n, this method was useless.

Clearly, other methods were needed for the generation of the $j_n(x)$. The Handbook of Mathematical Functions, AMS 55 (Reference 1), contains various formulas which were used and are outlined below.

Equation 10.1.2 of Reference 1 is

$$j_n(x) = \frac{x^n}{1 \cdot 3 \cdot 5 \cdots (2n+1)} \left[1 - \frac{\frac{1}{2}x^2}{1!(2n+3)} + \frac{\left(\frac{1}{2}x^2\right)^2}{2!(2n+3)(2n+5)} - \cdots \right] \quad (11)$$

This formula works very well when $n > |x|$; consequently, only small arguments are practical. When $|x|$ is larger than n, round-off error is introduced in the infinite sum inside the brackets of Eq. (11). This method is also very time consumming, as far as computer time is concerned. The companion formula, 10.1.3, for $y_n(x)$,

$$y_n(x) = -\frac{1 \cdot 3 \cdot 5 \cdots (2n-1)}{x^{n+1}} \left[1 - \frac{\frac{1}{2}x^2}{1!(1-2n)} + \frac{\left(\frac{1}{2}x^2\right)^2}{2!(1-2n)(3-2n)} - \cdots \right] \quad (12)$$

was tested also. This behaved similar to Eq. (11), in that round-off error is significant if $|x|$ is large in comparison with n.

Equations 10.1.8 and 10.1.9 in Reference 1 are the formulas for $j_n(x)$ and $y_n(x)$, respectively, expressed as representations by elementary functions. A subroutine based on this method (Reference 4) was tested in order to compare the numerical results. In testing it, results were similar to those obtained by the forward recurrence technique. That is, for the $j_n(x)$ at $n \geq |x|$ round-off error is introduced. The $y_n(x)$ generated by this method are accurate, however. Another formula tested was Eq. 10.1.13 in

AMS 55. This is the Poisson Integral representation,

$$j_n(x) = \frac{x^n}{2^{n+1}\, n!} \int_0^{\pi} \cos(x \cos\theta) \sin^{2n+1}\theta \, d\theta . \qquad (13)$$

The numerical results were excellent for this method over the range of n and x that were tested. But, the computer time necessary for the computations was quite large when compared with other methods.

Before developing the technique used by SBF, one more method was tested. This was the formula of Eq. 10.1.16 in AMS 55. Here the $j_n(x)$ and $y_n(x)$ were to be calculated simultaneously by calculating $h_n^{(1)}(x)$ and separating the real and imaginary parts.

$$h_n^{(1)}(x) = \frac{i^{-n-1}}{x} e^{ix} \sum_{k=0}^{n} (-2ix)^{-k} \frac{(n+k)!}{k!\,\Gamma(n-k+1)} \qquad (14)$$

This equation produced similar results to most of the other methods; that is, the round-off error was significant for the $j_n(x)$ beyond the point where $n \geq |x|$.

The backward recurrence method of generating $j_n(x)$ proved to be the most accurate of all the methods that were tested. In most cases it was also the one which took less time to compute on the CDC 6600 computer.

VII. SUMMARY

Subroutine SBF can be used to generate the spherical Bessel functions of all four kinds and their derivatives with a high degree of accuracy. The accuracy has been verified through the use of tables in the references below and by the Wronskian relationship. The order n is limited to positive values greater than or equal to 1, and the argument x must be in the range $-.0001 \geq x \geq .0001$. All of the formulas used were directly from or derived from AMS 55.

ACKNOWLEDGEMENTS

The suggestions of Capt. Carl E. Baum of the Air Force Weapons Laboratory in the use of the recurrence relationships and other algorithms is gratefully acknowledged.

REFERENCES

(1) Handbook of Mathematical Functions, AMS 55, M. Abramowitz and I. A. Stegun, Editors, National Bureau of Standards, 1964.

(2) Methods of Theoretical Physics, P. Morse and H. Feshbach, McGraw-Hill, New York, 1953.

(3) Standard Mathematical Tables, S. M. Selby, Editor, The Chemical Rubber Company, Cleveland, Ohio, 1967.

(4) Personal Correspondence, J. N. Brittingham, LRL, Livermore, Computer Information Center, April 18, 1969.

APPENDIX

PROGRAM LISTING

```
      SUBROUTINE SBF (N,X,J,Y,JP,YP,H1,H2,H1P,H2P,IW)                    SBF    1
C                                                                        SBF    2
C     CALCULATE THE SPHERICAL BESSEL FUNCTIONS.                          SBF    3
C     THE J FUNCTIONS ARE CALCULATED FROM A BACKWARD RECURRENCE RELATIONSBF    4
C     THE Y FUNCTIONS ARE CALCULATED FROM A FORWARD RECURRENCE RELATION  SBF    5
C                                                                        SBF    6
      REAL J(N),Y(N),JP(N),YP(N),J0,J1,JP0,JT1,JT2,JT                    SBF    7
      COMPLEX H1(N),H2(N),H1P(N),H2P(N),H10,H20,H1P0,H2P0                SBF    8
      COMMON /ZERO/ J0,Y0,JP0,YP0,H10,H20,H1P0,H2P0                      SBF    9
      COMMON /RR/ ISAVE                                                  SBF   10
      Z=X                                                                SBF   11
      S=SIN(Z)                                                           SBF   12
      C=COS(Z)                                                           SBF   13
      RZ=1./Z                                                            SBF   14
C     CALCULATE FUNCTIONS FOR THE N=0 AND N=1 ORDERS.                    SBF   15
      J0=RZ*S                                                            SBF   16
      Y0=-RZ*C                                                           SBF   17
      J(1)=RZ*(J0-C)                                                     SBF   18
      Y(1)=RZ*(Y0-S)                                                     SBF   19
      IF (N.LT.2) GO TO 80                                               SBF   20
C     CALCULATE FUNCTIONS FOR THE N=2 ORDER.                             SBF   21
      J(2)=3.*RZ*J(1)-RZ*S                                               SBF   22
      Y(2)=3.*RZ*Y(1)+RZ*C                                               SBF   23
      IF (N.LT.3) GO TO 80                                               SBF   24
C     CALCULATE THE Y FUNCTIONS.                                         SBF   25
      DO 10 I=3,N                                                        SBF   26
      Y(I)=FLOAT(2*I-1)*RZ*Y(I-1)-Y(I-2)                                 SBF   27
      ISAVE=I                                                            SBF   28
C     CHECK Y AGAINST MACHINE LIMITS.                                    SBF   29
      IF (ABS(Y(I))-1.E308) 10,20,20                                     SBF   30
10    CONTINUE                                                           SBF   31
      GO TO 40                                                           SBF   32
20    DO 30 K=ISAVE,N                                                    SBF   33
      J(K)=0.                                                            SBF   34
      Y(K)=-1.E308                                                       SBF   35
30    IF (Z.LT.0..AND.(MOD(K,2).EQ.0)) Y(K)= 1.E308                      SBF   35A
C     INITIALIZE BACKWARD RECURRENCE FOR J FUNCTIONS.                    SBF   36
40    J1=J(1)                                                            SBF   37
      N1=ISAVE+1                                                         SBF   38
      IM=MAX0(IFIX(ABS(Z))+50,ISAVE+10)                                  SBF   39
      JT1=0.                                                             SBF   40
      JT2=1.E-75                                                         SBF   41
      IND=IM-ISAVE+1                                                     SBF   42
      DO 50 I=2,IND                                                      SBF   43
      K=IM-I                                                             SBF   44
      JT=FLOAT(2*K+3)*RZ*JT2-JT1                                         SBF   45
      JT1=JT2                                                            SBF   46
50    JT2=JT                                                             SBF   47
      J(ISAVE)=JT1                                                       SBF   48
      J(ISAVE-1)=JT2                                                     SBF   49
C     DO LOOP TO OBTAIN BACKWARD RECURRENCE FUNCTIONS.                   SBF   50
      DO 60 I=3,ISAVE                                                    SBF   51
      K=N1-I                                                             SBF   52
```

```
60    J(K)=FLOAT(2*K+3)*RZ*J(K+1)-J(K+2)
C     FIND RATIO OF EXACT VALUE AT N=1 TO CALCULATED VALUE.
      R=J1/J(1)
C     MULTIPLY CALCULATED VALUES BY RATIO R.
      DO 70 I=1,ISAVE
70    J(I)=J(I)*R
C     HANKEL FUNCTIONS OF FIRST AND SECOND KINDS FOR ALL ORDERS.
80    H10=CMPLX(J0,Y0)
      H20=CMPLX(J0,-Y0)
      DO 90 I=1,N
      H1(I)=CMPLX(J(I),Y(I))
90    H2(I)=CMPLX(J(I),-Y(I))
C     CALL SUBROUTINE TO CALCULATE FIRST DERIVATIVES.
      CALL PRIME (N,Z,J,Y,JP,YP,H1,H2,H1P,H2P)
C     CALL SUBROUTINE TO CALCULATE WRONSKIANS.
      IF (IW.NE.0) CALL WRON (N,Z,J,Y,JP,YP)
      RETURN
      END
```

```
SBF   53
SBF   54
SBF   55
SBF   56
SBF   57
SBF   58
SBF   59
SBF   60
SBF   61
SBF   62
SBF   63
SBF   64
SBF   65
SBF   66
SBF   67
SBF   68
SBF   69
SBF   70-
```

```
      SUBROUTINE PRIME (N,Z,J,Y,JP,YP,H1,H2,H1P,H2P)                    PRI    1
C                                                                       PRI    2
C     THIS SUBROUTINE CALCULATES THE FIRST DERIVATIVES OF THE SPHERICAL PRI    3
C     BESSEL FUNCTIONS.                                                 PRI    4
C                                                                       PRI    5
      REAL J(N),Y(N),JP(N),YP(N),J0,JP0                                 PRI    6
      COMPLEX H1(N),H2(N),H1P(N),H2P(N),H10,H20,H1P0,H2P0                PRI    7
      COMMON /ZERO/ J0,Y0,JP0,YP0,H10,H20,H1P0,H2P0                     PRI    8
      RZ=1./Z                                                           PRI    9
      TRZ=2.*RZ                                                         PRI   10
C     CALCULATE DERIVATIVES FOR N=0 AND N=1 ORDERS.                     PRI   11
      JP0=-J(1)                                                         PRI   12
      YP0=-Y(1)                                                         PRI   13
      H1P0=-H1(1)                                                       PRI   14
      H2P0=-H2(1)                                                       PRI   15
      YP(1)=Y0-TRZ*Y(1)                                                 PRI   16
      JP(1)=J0-TRZ*J(1)                                                 PRI   17
      H1P(1)=H10-TRZ*H1(1)                                              PRI   18
      H2P(1)=H20-TRZ*H2(1)                                              PRI   19
      IF (N.LT.2) RETURN                                                PRI   20
C     DO LOOP OVER THE ORDERS.  DERIVATIVES ARE CALCULATED BY A         PRI   21
C     RECURRENCE RELATIONSHIP.                                          PRI   22
      DO 10 I=2,N                                                       PRI   23
      II=I-1                                                            PRI   24
      FRZ=FLOAT(I+1)*RZ                                                 PRI   25
      YP(I)=Y(II)-FRZ*Y(I)                                              PRI   26
      JP(I)=J(II)-FRZ*J(I)                                              PRI   27
      H1P(I)=H1(II)-FRZ*H1(I)                                           PRI   28
10    H2P(I)=H2(II)-FRZ*H2(I)                                           PRI   29
      RETURN                                                            PRI   30
      END                                                               PRI   31-
```

```
      SUBROUTINE WRON (N,Z,J,Y,JP,YP)                                   WRO    1
C                                                                       WRO    2
C     THIS SUBROUTINE COMPUTES THE WRONSKIAN RELATIONSHIP BETWEEN THE   WRO    3
C     J AND Y FUNCTIONS, AND COMPUTES A RELATIVE ERROR.                 WRO    4
C                                                                       WRO    5
      REAL J(N),Y(N),JP(N),YP(N)                                        WRO    6
      SAVE=0.                                                           WRO    7
      Z2=Z*Z                                                            WRO    8
C     DO LOOP OVER THE ORDERS.                                          WRO    9
      DO 20 M=1,N                                                       WRO   10
      IF (J(M).EQ.0.) GO TO 30                                          WRO   11
C     COMPUTE THE WRONSKIAN.                                            WRO   12
      W=J(M)*YP(M)-JP(M)*Y(M)                                           WRO   13
C     NORMALIZE BY THE ANALYTICAL VALUE OF THE WRONSKIAN.               WRO   14
      W=W*Z2                                                            WRO   15
C     COMPARE AGAINST 1.0.                                              WRO   16
      AERR=ABS(W-1.)                                                    WRO   17
C     SELECT LARGEST ERROR IN ARRAY.                                    WRO   18
      IF (AERR-SAVE) 20,20,10                                           WRO   19
10    SAVE=AERR                                                         WRO   20
      MSAVE=M                                                           WRO   21
20    CONTINUE                                                          WRO   22
30    PRINT 40, Z,N,AERR,MSAVE                                          WRO   23
      RETURN                                                            WRO   24
C                                                                       WRO   25
40    FORMAT (20H WITH THE ARGUMENT =,F12.5,29H, AND ORDER N = 1 THROUGHWRO   26
     1 N =,I5,21H, THE WRONSKIAN CHECK/34H SHOWS A MAXIMUM RELATIVE ERROWRO   27
     2R OF,E12.5,19H, OCCURRING AT N = ,I5)                             WRO   28
      END                                                               WRO   29-
```