Mathematics Notes

Note 6

October 1969

SPHBSLR

A Subroutine to Generate Spherical Bessel Functions for Real Arguments

J. N. Brittingham and P. L. Petronelli

Lawrence Radiation Laboratory

ABSTRACT

This note presents a subroutine to generate spherical Bessel function of first and second kind for real arguments. The algorithm and checking procedure are described. A subroutine to check the error in the Wronskian of the spherical Bessel function is also presented.

SPHBSLR

A Subroutine to Generate Spherical Bessel Functions for Real Arguments*

INTRODUCTION:

The present note describes a computer subroutine SPHBSLR that generates spherical Bessel functions of the first and second kind, for real arguments, namely j_n (X) and y_n (X). (i.e., $j_n(X) = (\frac{\pi}{2X})^{\frac{1}{2}} J_{n+\frac{1}{2}}(X)$ and $y_n(X) = (\frac{\pi}{2X})^{\frac{1}{2}} Y_{n+\frac{1}{2}}(X)$ where n = 0,1,2,3,...) The subroutine is written in Fortran IV for the CDC 6600 computer. The range of arguments for which the subroutine is accurate depends upon the value of n. For n such that $0 \le n \le 50$, the argument must be $10^{-4} \le X \le 5.0 \cdot 10^{6}$. For n such that $50 < n \le 60$ the, argument must be $8.0 \cdot 10^{-4} \le X \le 5.0 \cdot 10^{6}$. This subroutine accepts values outside the above range of n and X, but no assurance of accuracy is given to the numbers returned by the subroutine.

A subroutine, CHECK, is also presented in this note. CHECK is a subroutine to calculate the numerical accuracy of the Wronskian of the spherical Bessel functions for any given X and n. Checking the accuracy of the Wronskian is a prime method in evaluating SPHBSLR.

USE OF SPHBSLR:

The subroutine SPHBSLR is used in conjunction with a main program. The main program initiates SPHBSLR by using a Fortran IV CALL statement. The format for this statement is

CALL SPHBSLR (X, N, JJ, YY)

Where

X - a floating point variable, is the argument of the spherical Bessel functions. The numerical value of this variable is assigned by the main program prior to the use of the CALL statement.

N - an integer variable, is the order of the spherical Bessel functions. The numerical value of this integer is assigned by the main program prior to the use of the CALL statement.

* Work performed under the auspices of the U. S. Atomic Energy Commission.

JJ - a floating point variable, is the variable name under which the spherical Bessel function of the first kind returns to the main program.

YY - a floating point variable, is the variable name under which the spherical Bessel function of the second kind returns to the main program.

The names given the parameters in the above CALL statement are used for demonstration purposes only. The actual parameter names used by the main program must agree in type and need not agree in name.

RANGE:

SPHBSLR generates spherical Bessel functions with an error less than 10^{-8} in the following range:

when $0 \leq n \leq 50$ then $10^{-4} \leq X \leq 2.0 \cdot 10^{6}$,

and $50 < n \leq 60$ then $8.0 \cdot 10^{-4} \leq X \leq 2.0 \cdot 10^{6}$.

It should be noted that a machine of world length comparable to the CDC 6600 is needed to obtain similar accuracy.

The accuracy and range of SPHBSLR was established from three tests. First, the Wronskian error was examined for accuracies less than 10^{-8} (see CHECK). Second, data from this subroutine was compared with data tabulated in reference [2]. Third, the functions generated were tested to assure that their numerical values were not outside the range of the machine.

EXTERNAL CALLS:

SPHBSLR makes use of two standard subroutines. These routines are contained in the CDC 6600 function library.

SINF - This routine computes the value of the sine function of the given argument.

COSF - This routine computes the value of the cosine function of the given argument.

METHOD:

The spherical Bessel function of first and second kind can be written as a finite series and as an infinite series.

The finite series are

$$j_n(X) = \frac{1}{X}\left[P(n+\tfrac{1}{2}, X)\sin\left(X-\frac{n\pi}{2}\right) + Q(n+\tfrac{1}{2}, X)\cos\left(X-\frac{n\pi}{2}\right)\right]$$

and

$$y_n(X) = (-1)^{n+1} X^{-1}\left[P(n+\tfrac{1}{2}, X)\cos\left(X+\frac{n\pi}{2}\right) - Q(n+\tfrac{1}{2}, X)\sin\left(X+\frac{n\pi}{2}\right)\right]$$

$$(n = 0,1,2,3,\ldots) \quad .$$

Where

$$P(n+\tfrac{1}{2}, X) = \sum_{K=0}^{\{\frac{1}{2}n\}} (-1)^K \frac{(n+2K)!}{(2K)!\,\Gamma(n-2k+1)} (2X)^{-2K}$$

and

$$Q(n+\tfrac{1}{2}, X) = \sum_{K=0}^{\{\frac{1}{2}(n-1)\}} (-1)^K \frac{[n+(2K+1)]!}{(2K+1)!\,\Gamma(n-2k)} (2X)^{-2K-1}$$

$$(n = 0,1,2,3,\ldots) \quad .$$

The $\{\alpha\}$ represents the largest integer contained in α. Γ represents the gamma function and ! represents the factorial of the preceding number.

The infinite series for the spherical Bessel functions are

$$j_n(X) = \frac{X^n}{1\cdot3\cdot5\ldots(2n+1)}\left[1 - \frac{\frac{1}{2}X^2}{1!\,(2n+3)} + \frac{(\frac{1}{2}X^2)^2}{2!\,(2n+3)(2n+5)} - \ldots\right]$$

and

$$y_n(X) = \frac{-1\cdot3\cdot5\ldots(2n-1)}{X^{n+1}}\left[1 - \frac{\frac{1}{2}X^2}{1!\,(1-2n)} + \frac{(\frac{1}{2}X^2)^2}{2!\,(1-2n)(3-2n)} - \ldots\right]$$

$$(n = 0,1,2,3,\ldots) \quad .$$

SPHBSLR mates the finite and infinite series solutions to obtain the range of n and X. The finite series are used when

$1 \le X$ for $n = 0,1$

and $52n \le 60X$ for $n = 2,3,4,...$.

One hundred terms of the infinite series are used when

$1 > X$ for $n = 0,1$

and $52n > 60X$ for $n = 2,3,4,...$.

USE OF CHECK:

CHECK is a subroutine written to calculate the accuracy of the Wronskian of the spherical Bessel functions for a given argument and order. CHECK is used in conjunction with a main program. The main program must use a Fortran IV CALL statement to initiate CHECK. The format for this CALL statement is

```
CALL CHECK (X, N, ERROR)
```

Where

X - a floating point variable, is the argument of the spherical Bessel functions used in the Wronskian check. The main program must assign a numerical value to this variable prior to using the CALL statement.

N - an integer variable, is the order of the spherical Bessel functions used in the Wronskian check. The main program must assign a numerical value to this variable prior to using the CALL statement.

ERROR - a floating point variable, is the variable name under which the numerical value from the Wronskian check is returned to the main program.

The names given the parameters in the above CALL statement are used for demonstration purposes only. The actual parameter names used by the main program must agree in type and need not agree in name.

METHOD:

The Wronskian of the spherical Bessel functions for a fixed argument and order is equal to X^{-2}. Symbolically

$$j_n(X) \cdot \frac{dy_n(X)}{dX} - y_n(X) \cdot \frac{dj_n}{dX} = \frac{1}{X^2} .$$

The derivative of the spherical Bessel functions are representable by

$$\frac{dT_n(X)}{dX} = \frac{n}{2n+1} T_{n-1}(X) - \frac{n+1}{2n+1} T_{n+1}(X),$$

when $T_n(X)$ is either $j_n(X)$ or $y_n(X)$.
A recurrence relation for the spherical Bessel functions is

$$\frac{2n+1}{X} T_n(X) = T_{n-1}(X) + T_{n+1}(X) ,$$

when $T_n(X)$ is as above. The recurrence relation and the derivative expression reduce the Wronskian relation to

$$y_n(X) \cdot j_{n+1}(X) - y_{n+1}(X) \quad j_n(X) = \frac{1}{X^2} .$$

The error in the Wronskian, $\varepsilon_n(X)$, is found by obtaining $j_n(X)$, $y_n(X)$, $j_{n+1}(X)$, and $y_{n+1}(X)$ from SPHBSLR for a fixed X and n and substituting into the relation

$$\varepsilon_n(X) = \frac{[y_n(X) \cdot j_{n+1}(X) - y_{n+1}(X) \cdot j_n(X) - \frac{1}{X^2}]}{\frac{1}{X^2}} .$$

The above expression for $\varepsilon_n(X)$ is used to calculate the accuracy of the Wronskian in CHECK. The numerical value of $\varepsilon_n(X)$ is returned to the main program under the name ERROR.

SUMMARY:

The subroutine SPHBSLR calculates the spherical Bessel functions of the first and second kind when given a real argument and integer order.

The elapse time for execution is between 2 and 50 milliseconds. The computer memory size is 1134 octal words. A subroutine CHECK is given as a means to indicate the accuracy of SPHBSLR.

REFERENCES:

[1] Methods of Theoretical Physics, Vol. I and II, P. Morse and H. Feshback, Mc Graw-Hill Book Company, New York, New York, 1953.

[2] Handbook of Mathematical Functions, M. Abramowitz and I. Stegum, A.M.S. #55, National Bureau of Standards, 1964.

APPENDIX: PROGRAM LISTING

```
      SUBROUTINE SPHBSLR(XX,N,SJ,SY)                                    SPB00001
      PARAMETER(PIE=3.14159265358979,P2=PIE/2.)                         SPB00002
      COMMON/DOUBLES/SFYN,SFJN                                          SPB00003
C
C     COMMON BLOCK /DOUBLES/ PROVIDES ACCESS TO DOUBLE PRECISION VALUES
C
      DOUBLE X,FN,F,FG,SQRZ,QRRZ,SQRZI,ZXI,SINS,COSS,SINP,COSP          SPB00004
      DOUBLE SFYN,SFJN,QS,PS,AJXI,SUM,SLAT,FABE,FX,FXF,ABDF,ABGG        SPB00005
      DOUBLE ARCH,AIR,BIR,GATTR,DID,ABGG,ABDG,DI                        SPB00006
      DOUBLE CON,ZCON,TWO,T2                                            SPB00007
      DOUBLE SXT                                                        SPB00008
      DATA(NL=100)                                                      SPB00009
      DATA(TWO=2.0)                                                     SPB00010
      DATA(T2=.5)                                                       SPB00011
C
C     CONVERT ARGUMENTS TO DOUBLE PRECISION
C
      X=XX                                                              SPB00012
      SFN=N                                                             SPB00013
      FN=SFN                                                            SPB00014
      XN=N                                                              SPB00015
C
C     DETERMINE TYPE OF SERIES
C
      IF(60.*XX-52.*XN)1,,                                              SPB00016
      IF(XX-1.0),100,100                                                SPB00017
C
C     INFINITE SERIES
C
1     CONTINUE                                                          SPB00018
      QRRZ=X*X*T2                                                       SPB00019
      ZXI=1.0/X                                                         SPB00020
      CON=1.0                                                           SPB00021
      ZCON=-1.0                                                         SPB00022
      FABE=ZXI                                                          SPB00023
      SLAT=1.0                                                          SPB00024
      DO 5 I=1,N+1                                                      SPB00025
      FABE=FABE*X/CON                                                   SPB00026
      SLAT=SLAT*ZXI*ZCON                                                SPB00027
      CON=CON+TWO                                                       SPB00028
    5 ZCON=ZCON+TWO                                                     SPB00029
      SUM=1.0                                                           SPB00030
      SUPU=1.0                                                          SPB00031
      CON=2.0*FN+1.0                                                    SPB00032
      ZCON=-1.0-2.0*FN                                                  SPB00033
      ABGG=1.0                                                          SPB00034
      ABDF=1.0                                                          SPB00035
      DO 7 I=1,NL                                                       SPB00036
      CON=CON+TWO                                                       SPB00037
      ZCON=ZCON+TWO                                                     SPB00038
      XOS=I                                                             SPB00039
      FG=XOS                                                            SPB00040
      ABGG=-ABGG*QRRZ/(FG*CON)                                          SPB00041
```

```
      ABDF=-ABDF*QRRZ/(FG*ZCON)                                         SPB00042
      SUM=SUM+ABGG                                                      SPB00043
 7    SUPU=SUPU+ABDF                                                    SPB00044
      SFJN=FABE*SUM                                                     SPB00045
      SFYN=SLAT*SUPU                                                    SPB00046
      GO TO 1010                                                        SPB00047
C
C     FINITE SERIES
C
 100  NP=N/2                                                            SPB00048
      FG=FN*0.5                                                         SPB00049
      F=(FN-1.0)*0.5                                                    SPB00050
      NQ=(N-1)/2                                                        SPB00051
      SQRZ=X+X                                                          SPB00052
      QRRZ=1.0/SQRZ                                                     SPB00053
      SQRZI=QRRZ*QRRZ                                                   SPB00054
      ZXI=1.0/X                                                         SPB00055
      ZARG=X-(FN*P2)                                                    SPB00056
      SINS=SINF(ZARG)                                                   SPB00057
      COSS=COSF(ZARG)                                                   SPB00058
      ZARG=X+(FN*P2)                                                    SPB00059
      SINP=SINF(ZARG)                                                   SPB00060
      COSP=COSF(ZARG)                                                   SPB00061
      IF(N-1)  ,162,182                                                 SPB00062
      SFJN=ZXI*SINS                                                     SPB00063
      SFYN=-ZXI*COSS                                                    SPB00064
      GO TO 1010                                                        SPB00065
 162  CONTINUE                                                          SPB00066
      SFJN=ZXI*(SINS+ZXI*COSS)                                          SPB00067
      SFYN=ZXI*(COSP-(ZXI*SINP))                                        SPB00068
      GO TO 1010                                                        SPB00069
 182  PS=1.0                                                            SPB00070
      L=N/2                                                             SPB00071
      AJXI=1.0                                                          SPB00072
      IF(N-(2*L))ODD,,ODD                                               SPB00073
      AJXI=-1.0                                                         SPB00074
 ODD  CONTINUE                                                          SPB00075
      SUM=1.0                                                           SPB00076
      SLAT=1.0                                                          SPB00077
      FABE=1.0                                                          SPB00078
      DO 700 NSN=1,NP                                                   SPB00079
      SFN=NSN                                                           SPB00080
      FX=SFN                                                            SPB00081
      FXF=FX+FX                                                         SPB00082
      ABDF=FXF-1.0                                                      SPB00083
      ABGG=FXF-2.0                                                      SPB00084
      FABE=-(FN+FXF)*(FN+ABDF)*(FN-ABDF)*(FN-ABGG)/(FXF*ABDF)           SPB00085
      SXT=SQRZI*FABE                                                    SPB00086
      SLAT=SLAT*SXT                                                     SPB00087
 700  SUM=SUM+SLAT                                                      SPB00088
      PS=SUM                                                            SPB00089
      IF(NQ-1) ,720,720                                                 SPB00090
      QS=6.0*QRRZ                                                       SPB00091
      GO TO 800                                                         SPB00092
```

```
SUBROUTINE CHECK(Z,N,ERR)
ZONE=1.0/(Z*Z)
CALL SPHBSLR(Z,N+1,SJ,SY)
CALL SPHBSLR(Z,N,SSJ,SSY)
SW=SJ*SSY-SY*SSJ
ERR=(SW-ZONE)/ZONE
RETURN
END
```

```
  720 SLAT=(FN+1.0)*FN*QRRZ                                      SPB00093
      SUM=SLAT                                                   SPB00094
      DO 730 NSN=1,NQ                                            SPB00095
      SFN=NSN                                                    SPB00096
      FX=SFN                                                     SPB00097
      FXF=FX+FX                                                  SPB00098
      ABDF=FXF                                                   SPB00099
      ABGG=FXF+1.0                                               SPB00100
      AIR=FN+ABDF                                                SPB00101
      BIR=FN+ABGG                                                SPB00102
      GATTR=ABDF-1.0                                             SPB00103
      ARCH=-(AIR*BIR)*(FN-ABDF)*(FN-GATTR)/(ABDF*ABGG)           SPB00104
      SXT=ARCH*SQRZI                                             SPB00105
      SLAT=SLAT*SXT                                              SPB00106
  730 SUM=SUM+SLAT                                               SPB00107
      QS=SUM                                                     SPB00108
  800 SFJN=ZXI*((PS*SINS)+(QS*COSS))                             SPB00109
      SFYN=AJXI*ZXI*((PS*COSP)-(QS*SINP))                        SPB00110
 1010 CONTINUE                                                   SPB00111
C
C     CONVERT VALUES TO SINGLE PRECISION
C
      SY=SFYN                                                    SPB00112
      SJ=SFJN                                                    SPB00113
      RETURN                                                     SPB00114
      END                                                        SPB00115
```