Mathematics Notes

Note 5

October 1969

SPHBSL

A Subroutine to Generate Spherical Bessel Functions for Complex Arguments and Integer Orders

J. N. Brittingham and P. L. Petronelli

Lawrence Radiation Laboratory

ABSTRACT

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

SPHBSL

A Subroutine to Generate Spherical Bessel Functions for Complex Arguments and Integer Order*

INTRODUCTION:

The present note describes a computer subroutine SPHBSL that generates spherical Bessel functions of the first and second kind for complex arguments and integer order, namely $j_n(Z)$ and $y_n(Z)$. (i.e., $j_n(Z) = (\frac{\pi}{2Z})^{\frac{1}{2}} J_{n+\frac{1}{2}}(Z)$ and $y_n(Z) = (\frac{\pi}{2Z})^{\frac{1}{2}} Y_{n+\frac{1}{2}}(Z)$ when n = 0,1,2,3... .) The subroutine is written in Fortran IV for the CDC 6600 computer. For notational purposes, let X represent the real part of the argument and let Y represent the imaginary part of the argument. The range of arguments for which the subroutine is accurate depends upon the value of n. For n such that $0 \leq n \leq 40$ then $10^{-5} \leq X \leq 10^5$ and $0 \leq Y \leq 8$. For n such that $40 < n \leq 50$ then $10^{-4} \leq X \leq 10^5$ and $0 \leq Y \leq 8$. For n such that $50 < n \leq 60$ then $10^{-3} \leq X \leq 10^5$ and $0 \leq Y \leq 8$. SPHBSL accepts values outside the above range of n and Z, 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 Z and n. Checking the accuracy of the Wronskian is a prime method in evaluating SPHBSL.

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

USE OF SPHBSL:

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

CALL SPHBSL (Z, N, JJ, YY)

Where

Z - a two dimension floating point array, is the complex argument of the spherical Bessel function. The first number in the array is the real part of the argument and the second is the imaginary part of the argument. The numerical value of this variable must be assigned by the main program prior to using the CALL statement.

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

JJ - a two dimension floating point array, is the variable name under which the spherical Bessel function of the first kind returns to the main program. The first number in the array is the real part of the function and the second is the imaginary part of the function.

YY - a two dimension floating point array, is the variable name under which the spherical Bessel function of the second kind returns to the main program. The first number in the array is the real part of the function and the second is the imaginary part of the function.

The names given the parameters of the above CALL statement are used for illustration purposes only. The actual parameters must agree in type and need not be identical in name.

RANGE:

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

for $0 \leq n \leq 40$ then $10^{-5} \leq X \leq 10^5$ and $0 \leq Y \leq 8$;

for $40 < n \leq 50$ then $10^{-4} \leq X \leq 10^5$ and $0 \leq Y \leq 8$;

for $50 < n \leq 60$ then $10^{-3} \leq X \leq 10^5$ and $0 \leq Y \leq 8$.

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

The accuracy and range of SPHBSL 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:

SPHBSL uses two external subroutines SCD and EXPD. SCD is initiated by the Fortran IV CALL statement. EXPD is initiated by the Fortran IV FUNCTION statement. These subroutines are presented in the Appendix.

SCD is a subroutine that generates a double precision sine and cosine function of the given argument.

EXPD is a subroutine that generates a double precision exponential function of the given argument.

The full accuracy of SCD and EXPD is not known, but the accuracy of SPHBSL with these subroutines is improved over the accuracy with the single precision routines of the same functions. If a double precision routine for sine, cosine and exponential functions exist on the system to be used, it may be desirable to replace SCD and EXPD by the improved subroutines. The modification is accomplished by replacing statements 135, 136, 142 and 148 in SPHBSL with the appropriate system calls.

MEMORY SIZE:

SPHBSL - 2502 Octal Words
SCD - 365 Octal Words
EXPD - 70 Octal Words

METHOD:

The spherical Bessel functions are representable as finite series and infinite series.

The finite series are

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

and

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

$n = 0,1,2,3\ldots$.

Where

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

and

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

$n = 0,1,2,3\ldots$.

The $\{\alpha\}$ represents the largest integer contained in α. Γ represents the gamma function.

The infinite series for the spherical Bessel functions are

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

and

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

$$n = 0,1,2,3\ldots\ .$$

The present program uses a mating of the finite and infinite series solution to obtain the range of n and Z. The finite series are used when

$$1 \leq |Z| \qquad \text{for} \qquad n = 0,1$$

and

$$52n \leq 60|Z| \qquad \text{for} \qquad n = 2,3,4,\ldots\ .$$

A one hundred term sum of the infinite series is used when

$$1 > |Z| \qquad \text{for} \qquad n = 0,1$$

and

$$52n > 60|Z| \qquad \text{for} \qquad n = 2,3,4,\ldots\ .$$

USE OF CHECK:

CHECK is a subroutine written to calculate the accuracy of the Wronskian of the spherical Bessel functions for a given complex 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 (Z, N, ERROR)
```

Where

Z - a two dimension floating point array, is the complex argument of the spherical Bessel functions used in the Wronskian check. The first number in the array is the real part of the argument and the second is the imaginary part of the argument. The numerical value of this variable must be assigned by the main program 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 Z^{-2}. Symbolically

$$j_n(Z) \cdot \frac{dy_n(Z)}{dZ} - y_n(Z) \cdot \frac{djn(Z)}{dX} = \frac{1}{Z^2} .$$

The derivative of the spherical Bessel functions is representable by

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

when $T_n(Z)$ is either $j_n(Z)$ or $y_n(Z)$.

A recurrence relation for the spherical Bessel functions is

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

when $T_n(Z)$ is as above.

The recurrence relation and the derivative expression reduce the Wronskian relation to

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

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

$$\varepsilon_n(Z) = \left| \frac{[y_n(Z) \cdot j_{n+1}(Z) - y_{n+1}(Z) \cdot j_n(Z) - \frac{1}{Z^2}]}{\frac{1}{Z^2}} \right| \quad .$$

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

SUMMARY:

The subroutine SPHBSL calculates the spherical Bessel function of the first and second kind when given a complex argument and integer order. The average elapse time for execution in the specified range is 87 milliseconds. A subroutine CHECK is presented as a means to indicate the accuracy of SPHBSL.

REFERENCES:

(1) Methods of Theoretical Physics, Vol. I and II, P. Morse and H. Feshback, McGraw-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 LIST

```
      SUBROUTINE SPHBSL(ZS,N,JJ,YY)
      DOUBLE EXPD,CSIND,CCOSD
      REAL JJD,JJ
      DIMENSION ZS(2),Z(2),Y(2),JJD(2),JJ(2),YY(2),YYD(2)
      DIMENSION ZM(2),ZN(2),ZCON(2),DUL(2),STERM(2),ZT(2)
      DIMENSION ZXI(2),QRRZ(2),SQRZI(2),SLAT(2),SINA(2),COSA(2)
      DIMENSION SINP(2),COSP(2),QS(2),PS(2),SUM(2),SUPU(2)
      DIMENSION PSSI(2),PSCI(2),QSSI(2),QSCI(2)
C     ***** THESE ARE DOUBLE
      DOUBLE SIH,COH,SIN,COS,SINA,COSA,SINP,COSP,SUPU,SUM,SLAT
      DOUBLE PS,QS,AJXI,FABE,FX,ABDF,ABGG,DID,DI,FXF,AIR,BIR
      DOUBLE GATTR,PSSI,PSCI,QSSI,QSCI
      DOUBLE Z,Y,ZM,ZN,TWO,CON,ZCON,T2,DUL,STERM,SIGN,X,XO,X2,ZT
      DOUBLE JJD,YYD,QRRZ,SQRZI,ZXI,F,FN,PIE,PIETO,ZARG,EXP,EXM,FNPIE
      Z(1)=ZS(1)
      Z(2)=ZS(2)
      DATA(TWO=2.0)
      XX=SQRTF(ZS(1)**2+ZS(2)**2)
      XN=N
      IF(60.*XX-52.*XN)99,,
      IF(XX-1.0),160,160
99    CONTINUE
      NL=100
C     COMPUTE POWER
      Y(1)=Z(1)
      Y(2)=Z(2)
      ZM(1)=Y(1)
      ZM(2)=Y(2)
      CON=1.0
      ZCON(1)=1.0
      ZCON(2)=0.0
      DO 1 I=1,N
      CON=CON+TWO
      T2=1.0/CON
      ZN(1)=ZM(1)*T2
      ZN(2)=ZM(2)*T2
      DUL(1)=ZCON(1)
      DUL(2)=ZCON(2)
      ZCON(1)=DUL(1)*ZN(1)-DUL(2)*ZN(2)
 1    ZCON(2)=DUL(2)*ZN(1)+DUL(1)*ZN(2)
      IF(N)6,,6
      ZCON(1)=1.0
      ZCON(2)=0.0
 6    CON=1.0
      T2=0.5
      XOS=2*N+1
      XO=XOS
      STERM(1)=1.0
      STERM(2)=0.0
      ZT(1)=STERM(1)
      ZT(2)=STERM(2)
      ZM(1)=(Y(1)*Y(1)-Y(2)*Y(2))*T2
      ZM(2)=Y(1)*Y(2)
```

```
      SIGN=-1.0
      X2=TWO
      DO 3 I=1,NL-1
      XS=I
      X=XS
      CON=SIGN/(X*(XO+X2))
      ZN(1)=ZM(1)*CON
      ZN(2)=ZM(2)*CON
      DUL(1)=ZT(1)
      DUL(2)=ZT(2)
      ZT(1)=DUL(1)*ZN(1)-DUL(2)*ZN(2)
      ZT(2)=DUL(2)*ZN(1)+DUL(1)*ZN(2)
      STERM(1)=ZT(1)+STERM(1)
      STERM(2)=ZT(2)+STERM(2)
    3 X2=X2+TWO
      JJD(1)=STERM(1)*ZCON(1)-STERM(2)*ZCON(2)
      JJD(2)=STERM(1)*ZCON(2)+STERM(2)*ZCON(1)
      JJ(1)=JJD(1)
      JJ(2)=JJD(2)
C     COMPUTE  POWER
      STERM(1)=1.0
      STERM(2)=0.0
      CON=Y(1)*Y(1)+Y(2)*Y(2)
      ZN(1)=Y(1)/CON
      ZN(2)=-Y(2)/CON
      ZCON(1)=1.0
      ZCON(2)=0.0
      CON=-1.0
      DO 11 I=1,N+1
      ZM(1)=ZN(1)*CON
      ZM(2)=ZN(2)*CON
      DUL(1)=ZCON(1)
      DUL(2)=ZCON(2)
      ZCON(1)=ZM(1)*DUL(1)-ZM(2)*DUL(2)
      ZCON(2)=ZM(1)*DUL(2)+ZM(2)*DUL(1)
   11 CON=CON+TWO
C     COMPUTE SERIES
      XOS=-2*N
      XO=XOS
      X2=0.5
      ZM(1)=(Y(1)*Y(1)-Y(2)*Y(2))*T2
      ZM(2)=Y(1)*Y(2)
      SIGN= -1.0
      ZT(1)=STERM(1)
      ZT(2)=STERM(2)
      X2=1.0
      DO 13 I=1,NL-1
      XS=I
      X=XS
      CON=SIGN/(X*(X2+XO))
      ZN(1)=ZM(1)*CON
      ZN(2)=ZM(2)*CON
      DUL(1)=ZT(1)
      DUL(2)=ZT(2)
```

```
      ZT(1)=DUL(1)*ZN(1)-DUL(2)*ZN(2)
      ZT(2)=DUL(2)*ZN(1)+DUL(1)*ZN(2)
      STERM(1)=ZT(1)+STERM(1)
      STERM(2)=ZT(2)+STERM(2)
   13 X2=X2+TWO
      YYD(1)=ZCON(1)*STERM(1)-ZCON(2)*STERM(2)
      YYD(2)=ZCON(1)*STERM(2)+ZCON(2)*STERM(1)
      YY(1)=YYD(1)
      YY(2)=YYD(2)
      GO TO 2000
  160 CONTINUE
      DATA(PIE=3.14159265358979)
C     INITIALIZE
      SFN=N
      FN=SFN
      F=FN
      NP=N/2
      NQ=(N-1)/2
      XJIX=2.0
      T2=0.5
      XO=(Z(1)*Z(1)+Z(2)*Z(2))
      ZXI(1)=Z(1)/XO
      ZXI(2)=-Z(2)/XO
      QRRZ(1)=ZXI(1)*T2
      QRRZ(2)=ZXI(2)*T2
      SQRZI(1)=QRRZ(1)*QRRZ(1)-QRRZ(2)*QRRZ(2)
      SQRZI(2)=TWO*QRRZ(1)*QRRZ(2)
      EXP=EXPD(Z(2))
      EXM=EXPD(-Z(2))
      SIH=(EXP-EXM)*T2
      COH=(EXP+EXM)*T2
      PIETO=PIE*.5
      FNPIE=FN*PIETO
      ZARG=Z(1)-FNPIE
      CALL SCD(SIN,COS,ZARG)
      SINA(1)=SIN*COH
      SINA(2)=COS*SIH
      COSA(1)=COS*COH
      COSA(2)=-SIN*SIH
      ZARG=Z(1)+FNPIE
      CALL SCD(SIN,COS,ZARG)
      SINP(1)=SIN*COH
      SINP(2)=COS*SIH
      COSP(1)=COS*COH
      COSP(2)=-SIN*SIH
      IF(N-1)   ,162,182
      JJD(1)=ZXI(1)*SINA(1)-ZXI(2)*SINA(2)
      JJD(2)=ZXI(1)*SINA(2)+ZXI(2)*SINA(1)
      YYD(1)=ZXI(1)*COSP(1)-ZXI(2)*COSP(2)
      YYD(2)=ZXI(1)*COSP(2)+ZXI(2)*COSP(1)
      YYD(1)=-YYD(1)
      YYD(2)=-YYD(2)
      GO TO 1010
  162 PS(1)=1.0
```

```
      PS(2)=0.0                                                        SPB00162
      CON=TWO                                                          SPB00163
      QS(1)=CON*QRRZ(1)                                                SPB00164
      QS(2)=CON*QRRZ(2)                                                SPB00165
      SUPU(1)=QS(1)*COSA(1)-QS(2)*COSA(2)                              SPB00166
      SUPU(2)=QS(1)*COSA(2)+QS(2)*COSA(1)                              SPB00167
      SUM(1)=SUPU(1)+SINA(1)                                           SPB00168
      SUM(2)=SUPU(2)+SINA(2)                                           SPB00169
      JJD(1)=ZXI(1)*SUM(1)-ZXI(2)*SUM(2)                               SPB00170
      JJD(2)=ZXI(1)*SUM(2)+ZXI(2)*SUM(1)                               SPB00171
      SUPU(1)=QS(1)*SINP(1)-QS(2)*SINP(2)                              SPB00172
      SUPU(2)=QS(1)*SINP(2)+QS(2)*SINP(1)                              SPB00173
      SUM(1)=COSP(1)-SUPU(1)                                           SPB00174
      SUM(2)=COSP(2)-SUPU(2)                                           SPB00175
      YYD(1)=SUM(1)*ZXI(1)-SUM(2)*ZXI(2)                               SPB00176
      YYD(2)=SUM(1)*ZXI(2)+SUM(2)*ZXI(1)                               SPB00177
      GO TO 1010                                                       SPB00178
  182 PS(1)=1.0                                                        SPB00179
      PS(2)=0.0                                                        SPB00180
      L=N/2                                                            SPB00181
      AJXI=1.0                                                         SPB00182
      IF(N-(2*L))ODD,,ODD                                              SPB00183
      AJXI=-1.0                                                        SPB00184
  ODD CONTINUE                                                         SPB00185
      SUM(1)=1.0                                                       SPB00186
      SUM(2)=0.0                                                       SPB00187
      SUPU(1)=1.0                                                      SPB00188
      SUPU(2)=0.0                                                      SPB00189
      SLAT(1)=1.0                                                      SPB00190
      SLAT(2)=0.0                                                      SPB00191
      FABE=1.0                                                         SPB00192
      DO 700 NSN=1,NP                                                  SPB00193
      SFN=NSN                                                          SPB00194
      FX=SFN                                                           SPB00195
      FXF=2.0*FX                                                       SPB00196
      ABDF=FXF-1.0                                                     SPB00197
      ABGG=FXF-2.0                                                     SPB00198
      FABE=(-1.0)*    (FN+FXF)*(FN+ABDF)*(FN-ABDF)*(FN-ABGG)/(FXF*ABDF)   SPB00199
      DUL(1)=SLAT(1)                                                   SPB00200
      DUL(2)=SLAT(2)                                                   SPB00201
      SLAT(1)=(SQRZI(1)*DUL(1)-SQRZI(2)*DUL(2))*FABE                   SPB00202
      SLAT(2)=(SQRZI(1)*DUL(2)+SQRZI(2)*DUL(1))*FABE                   SPB00203
      SUM(1)=SLAT(1)+SUM(1)                                            SPB00204
  700 SUM(2)=SLAT(2)+SUM(2)                                            SPB00205
      PS(1)=SUM(1)                                                     SPB00206
      PS(2)=SUM(2)                                                     SPB00207
      IF(NQ-1) ,720,720                                                SPB00208
      QS(1)=QRRZ(1)*6.0                                                SPB00209
      QS(2)=QRRZ(2)*6.0                                                SPB00210
      GO TO 800                                                        SPB00211
  720 CON=(FN+1.0)*FN                                                  SPB00212
      SUM(1)=QRRZ(1)*CON                                               SPB00213
      SUM(2)=QRRZ(2)*CON                                               SPB00214
      SLAT(1)=SUM(1)                                                   SPB00215
```

```
      SLAT(2)=SUM(2)
      DO 730 NSN=1,NQ
      SFN=NSN
      FX=SFN
      FXF=2.0*FX
      ABDF=FXF
      ABGG=FXF+1.0
      AIR=FN+ABDF
      BIR=FN+ABGG
      GATTR=ABDF-1.0
      CON=(-1.0)*AIR*BIR*(FN-ABDF)*(FN-GATTR)/(ABDF*ABGG)
      SUPU(1)=SLAT(1)*CON
      SUPU(2)=SLAT(2)*CON
      SLAT(1)=SUPU(1)*SQRZI(1)-SUPU(2)*SQRZI(2)
      SLAT(2)=SUPU(1)*SQRZI(2)+SUPU(2)*SQRZI(1)
      SUM(1)=SLAT(1)+SUM(1)
  730 SUM(2)=SLAT(2)+SUM(2)
      QS(1)=SUM(1)
      QS(2)=SUM(2)
  800 PSSI(1)=PS(1)*SINA(1)-PS(2)*SINA(2)
      PSSI(2)=PS(1)*SINA(2)+PS(2)*SINA(1)
      PSCI(1)=PS(1)*COSP(1)-PS(2)*COSP(2)
      PSCI(2)=PS(1)*COSP(2)+PS(2)*COSP(1)
      QSSI(1)=QS(1)*SINP(1)-QS(2)*SINP(2)
      QSSI(2)=QS(1)*SINP(2)+QS(2)*SINP(1)
      QSCI(1)=QS(1)*COSA(1)-QS(2)*COSA(2)
      QSCI(2)=QS(1)*COSA(2)+QS(2)*COSA(1)
      SLAT(1)=PSSI(1)+QSCI(1)
      SLAT(2)=PSSI(2)+QSCI(2)
      JJD(1)=ZXI(1)*SLAT(1)-ZXI(2)*SLAT(2)
      JJD(2)=ZXI(1)*SLAT(2)+ZXI(2)*SLAT(1)
      SLAT(1)=PSCI(1)-QSSI(1)
      SLAT(2)=PSCI(2)-QSSI(2)
      YYD(1)=(SLAT(1)*ZXI(1)-SLAT(2)*ZXI(2))*AJXI
      YYD(2)=(SLAT(1)*ZXI(2)+SLAT(2)*ZXI(1))*AJXI
 1010 CONTINUE
      JJ(1)=JJD(1)
      JJ(2)=JJD(2)
      YY(1)=YYD(1)
      YY(2)=YYD(2)
  ONE CONTINUE
 2000 CONTINUE
      RETURN
      END
```

```
*       LIST8
*       CARDS COLUMN
*       FORTRAN          SCD
        SUBROUTINE   SCD(CSIND,CCOSD,ARG)
        DOUBLE CSIND,CCOSD,ZARG,ARG
        DOUBLE PIE,PIETO
        DOUBLE SIN,COS,D,FNDD,RAMD,SD,CO,DOX,BOX,SQRAN,TOFN
        ZARG=ARG
        DATA(PIE=3.14159265358979)
        PIETO=PIE*0.5
        PIETS=PIETO
  221   ZARGS=ZARG
        IF(ZARGS)MIT,   ,PUS
        SIN=0.0
        COS=1.0
        GO TO 471
  MIT   D=-1.0
        ZARG=-ZARG
        ZARGS=-ZARGS
        GO TO 237
  PUS   D=1.0
  237   CONTINUE
        AMAD=ZARGS/PIETS
        MAD=AMAD
        NXD=MAD-((MAD/4)*4)
        FMADS=MAD
        FMAD=FMADS
        RAMD=ZARG-(PIETO*FMAD)
        SD=RAMD
        CO=1.0
        DOX=RAMD
        BOX=1.0
        SQRAN=RAMD*RAMD
        DO 247 NDD=1,20
        FNDDS=NDD
        FNDD=FNDDS
        TOFN=2.0*FNDD
        DOX=-DOX*(SQRAN/((TOFN+1.0)*TOFN))
        BOX=-BOX*(SQRAN/(TOFN*(TOFN-1.0)))
        SD=SD+DOX
  247   CO=CO+BOX
        IF(NXD-1)  ,258,261
        SIN=D*SD
        COS=CO
        GO TO 471
  258   SIN=D*CO
        COS=-SD
        GO TO 471
  261   IF(NXD-2)471,   ,273
        SIN=-D*SD
        COS=-CO
        GO TO 471
  273   SIN=-D*CO
```

```
      COS=SD
 471  CONTINUE
      CSIND=SIN
C
      CCOSD=COS
      RETURN
      END
```

```
*       LIST8
*       CARDS COLUMN
*       FORTRAN         EXPD
        FUNCTION EXPD (ZARGI)                                  XPD00001
        DOUBLE EXPD,ZARGI,EXP,EXM,SD,CO,FNDD                   XPD00002
        EXP=1.0                                                XPD00003
        SD=1.0                                                 XPD00004
        DO 219 NDD=1,200                                       XPD00005
        FNDDS=NDD                                              XPD00006
        FNDD=FNDDS                                             XPD00007
        SD=SD*(ZARGI/FNDD)                                     XPD00008
        EXP=EXP+SD                                             XPD00009
  219   CONTINUE                                               XPD00010
        EXPD=EXP                                               XPD00011
        RETURN                                                 XPD00012
        END                                                    XPD00013
```

```
SUBROUTINE CHECK(Z,N,ERR)
COMPLEX ZONE,Z,SSJ,SSY,SJ,SY,ERROR,SW
ZONE=1.0/(Z*Z)
CALL SPHBSL (Z,N+1,SJ,SY)
CALL SPHBSL (Z,N,SSJ,SSY)
SW=SJ*SSY-SY*SSJ
ERROR=(SW-ZONE)/ZONE
ERR=SQRTF(REAL(ERROR)**2+AIMAG(ERROR)**2)
RETURN
END
```