2.5 Bessel Functions of General Orders J_ν and Y_ν

A. Purpose

These subroutines compute a sequence of values $J_\nu(x)$ or $Y_\nu(x)$ for $\nu = \alpha, \alpha+1, ..., \alpha + \text{NUM} - 1$. J_ν and Y_ν are Bessel functions of the first and second kinds, respectively, as described in [1]. J_ν and Y_ν are a pair of linearly independent solutions of the differential equation

$$x^2 \frac{d^2 w}{dx^2} + x \frac{dw}{dx} + (x^2 - \nu^2) w = 0$$

Y_ν is also sometimes called the Neumann function and denoted by N_ν.

B. Usage

B.1 Program Prototype, Single Precision

REAL X, ALPHA, BJ(≥NUM),**BY**(≥NUM)

INTEGER NUM

Assign values to X, ALPHA, and NUM. To evaluate J Bessel functions:

CALL SBESJN (X, ALPHA, NUM, BJ)

To evaluate Y Bessel functions:

CALL SBESYN (X, ALPHA, NUM, BY)

The results are stored in BJ() or BY(), respectively.

B.2 Argument Definitions

X [in] Argument for function evaluation. Require X ≥ 0 for the J function and X > 0 for the Y function. Require X $< (16\rho)^{-1}$ for both functions, where ρ denotes the machine precision.

ALPHA [in] Lowest order, ν, for which $J_\nu(x)$ or $Y_\nu(x)$ is to be computed. Require ALPHA ≥ 0. For sufficiently large ν, depending on x, positive values of $J_\nu(x)$ will be smaller than the computer's underflow limit and the magnitude of $Y_\nu(x)$ will exceed the overflow limit. SBESYN issues an error message before overflow occurs.

NUM [in] Number of values of ν for which $J_\nu(x)$ or $Y_\nu(x)$ is to be computed. Require NUM ≥ 1.

BJ() [out] Array in which SBESJN will store results. BJ(i) $= J_{\alpha+i-1}(x)$ for $i = 1, 2, ..., \text{NUM}$.

BY() [out] Array in which SBESYN will store results. BY(i) $= Y_{\alpha+i-1}(x)$ for $i = 1, 2, ..., \text{NUM}$.

B.3 Modifications for Double Precision

For double precision usage, change the REAL statement to DOUBLE PRECISION and change the subroutine names to DBESJN and DBESYN, respectively.

©1997 Calif. Inst. of Technology, 2015 Math à la Carte, Inc.

C. Examples and Remarks

These Bessel functions satisfy the Wronskian identity ([1], Eq. 9.1.16)

$$z(\nu, x) = \frac{x\pi}{2} \left[J_{\nu+1}(x) Y_\nu(x) - J_\nu(x) Y_{\nu+1}(x) \right] - 1 = 0$$

The program DRSBESJN evaluates this expression for a few values of ν and x. The results are shown in ODSBESJN.

D. Functional Description

D.1 Properties of J and Y

In the region $x \geq \nu$, both J and Y are oscillatory and are bounded in magnitude by one. For fixed $\nu \geq 0$ and increasing x these functions have asymptotic behavior described by ([1], Eqs. 9.2.1 – 9.2.2)

$$J_\nu(x) \sim [2/(\pi x)]^{1/2} \cos(x - (\nu + 0.5)\pi/2) \quad (1)$$

$$Y_\nu(x) \sim [2/(\pi x)]^{1/2} \sin(x - (\nu + 0.5)\pi/2) \quad (2)$$

In the region $\nu \geq x$, $J_\nu(x)$ is positive and bounded and approaches zero as ν increases with fixed $x > 0$, while $Y_\nu(x)$ is negative and unbounded and approaches $-\infty$ as ν increases with fixed $x > 0$. For fixed $x > 0$ and increasing ν, these functions have asymptotic behavior described by ([1], Eqs. 9.3.1 – 9.3.2).

$$J_\nu(x) \sim (2\pi\nu)^{-1/2} (ex/(2\nu))^\nu \quad (3)$$

$$Y_\nu(x) \sim -(2/(\pi\nu))^{\frac{1}{2}} (ex/(2\nu))^{-\nu} \quad (4)$$

where $e = 2.718 \cdots$.

Both J and Y satisfy the recursion ([1], Eq. 9.1.27)

$$f_{\nu+1}(x) - (2\nu/x) f_\nu(x) + f_{\nu-1}(x) = 0 \quad (5)$$

For $\nu > x$ this recursion is stable in the forward direction for Y and in the backward direction for J. For $x > \nu$ the recursion is stable in either direction for both J and Y.

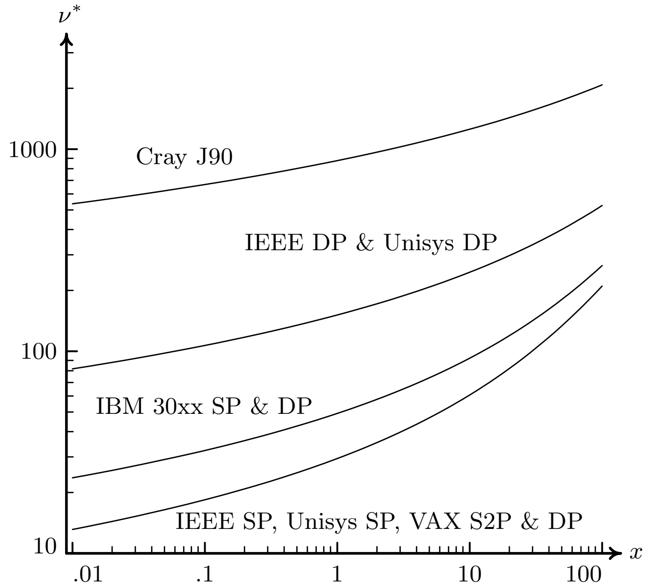
D.2 Machine dependent quantities

Let ρ denote the machine precision, *i.e.*, R1MACH(3) or D1MACH(3) of Chapter 19.1. Let Ω denote the overflow limit, *i.e.*, R1MACH(2) or D1MACH(2). Define

$$XPQ = 1.1293(-\log_{10}(\rho/4)) - 0.59$$

The asymptotic series used in these subroutines is valid for $x \geq XPQ$ and $0 \leq \nu \leq 2$.

Let $\nu^*(x)$ denote the value of ν for which Eq. (4) reaches the overflow limit, Ω, for a given value of x. It happens that $\nu^*(x)$ is very close to the value of ν for which Eq. (3) reaches the underflow limit on the same machine. The figure below shows plots of $\nu^*(x)$ for some computer systems currently in use at JPL.



D.3 Computation of $J_\nu(x)$

Given x, α, and NUM, define $\beta = \alpha + \mathrm{NUM} - 1$. Thus, β is the largest requested order.

For $x = 0$ the result is 1 if $\nu = 0$, and 0 if $\nu > 0$.

For $0 < x \le 0.1$ the Taylor series in x is used ([1],Eq. 9.1.10). For $0.1 < x \le \max(\beta, XPQ)$ forward recursion on ν is used to determine a starting point for backward recursion. The execution time in this region increases linearly with β and can be substantial for large β.

For $\max(\beta, XPQ) < x < (16\rho)^{-1}$ the subroutine evaluates the asymptotic series in x ([1], Eqs. 9.2.5, 9.2.9, and 9.2.10) for two values of ν in the range [0, 2], and then uses forward recursion. The execution time in this region increases linearly with β and decreases with increasing x.

If $x > (16\rho)^{-1}$ an error message is issued because the phase of the sine and cosine functions will not be known with any accuracy.

D.4 Computation of $Y_\nu(x)$

If $x = 0$ an error message is issued since the result would be $-\infty$. The output values are set to $-\Omega/2$.

For $0 < x \le \rho$ and $\nu = 0$, the result is $(2/\pi)(\gamma + \ln(x/2))$ ([1], Eq. 9.1.13), where γ denotes Euler's constant, $0.57721\cdots$. For $0 < x \le \rho$ and $\nu > 0$, the result is $-\pi^{-1}\Gamma(\nu)(x/2)^{-\nu}$ ([1], Eq. 9.1.9).

For $\rho < x < XPQ$ the subroutine first computes values of J. From these values it computes Y for two values of ν in [0, 2], and then uses forward recursion on ν to obtain the requested values.

For $XPQ \le x \le (16\rho)^{-1}$ the subroutine evaluates the asymptotic series in x for two values of ν in [0, 2], and then uses forward recursion.

If $x > (16\rho)^{-1}$ an error message is issued as noted previously for J.

D.5 Accuracy tests

The subroutines SBESJN and SBESYN were tested on an IBM compatible PC using IEEE arithmetic by comparison with the corresponding double precision subroutines. Tables 1 and 2 give a summary of the errors found in these tests. Each number in a rectangular cell is the maximum value of the error observed at 2592 points tested in the indicated range. Each number in a triangular cell is the maximum over 1296 points. The underflow limit for J_ν, and the overflow limit for Y_ν, actually extend down the ν axis (see Figure 3). Where the function underflows or overflows, fewer samples are used.

Table 1. Maximum errors found in indicated regions for SBESJN. Relative error is shown above the diagonal and absolute error below. Error is shown as a multiple of the machine precision, $\approx 1.19 \times 10^{-7}$ for these tests.

ν \ x	0–2	2–5	5–10	10–20	20–50	50–100
50–100	OVERFLOW		18	30	44	39 / 96
20–50	17	17	24	26	21 / 27	16
10–20	27	10	10	9 / 4	4	3
5–10	10	7	5 / 2	2	2	3
2–5	4	3 / 1	1	1	2	3
0–2	2 / 1	1	1	1	2	3

As a test of the double precision subroutines, and an additional test of the single precision subroutines, the expression $z(\nu, x)$ defined in Section C was evaluated at 40 points. Nine values are shown in Table 3 from these tests of SBESJN and SBESYN and in Table 4 from the tests of DBESJN and DBESYN.

These subroutines are designed for use with arithmetic precision to about 10^{-20}. The auxiliary subroutine DBESPQ has no inherent accuracy limitations.

Table 2. Maximum errors found in indicated regions for SBESYN. Relative error is shown above the diagonal and absolute error below. Error is shown as a multiple of the machine precision, $\approx 1.19 \times 10^{-7}$ for these tests.

ν \ x	0–2	2–5	5–10	10–20	20–50	50–100
50–100	OVERFLOW		162	286	585	659 / 68
20–50	44	84	128	143	153 / 19	21
10–20	24	25	27	29 / 4	3	3
5–10	8	9	9 / 9	2	2	3
2–5	9	9 / 14	15	1	2	3
0–2	12 / 28	15	10	1	2	3

Table 3. Single precision Wronskian test. Tabulated value is $z(\nu, x)/\text{R1MACH}(3)$ where $\text{R1MACH}(3) \approx 5.96 \times 10^{-8}$.

ν	$x \Rightarrow$ 5.1	15.3	30.6
30.6	6.2	24.6	3.5
15.3	5.9	5.9	0.2
5.1	1.3	0.7	3.5

Table 4. Double precision Wronskian test. Tabulated value is $z(\nu, x)/\text{D1MACH}(3)$ where $\text{D1MACH}(3) \approx 1.11 \times 10^{-16}$.

ν	$x \Rightarrow$ 5.1	15.3	30.6
30.6	32.2	17.2	16.0
15.3	3.9	5.8	0.4
5.1	1.0	1.8	0.1

References

1. Milton Abramowitz and Irene A. Stegun, **Handbook of Mathematical Functions**, *Applied Mathematics Series 55*, National Bureau of Standards (1966) Chapter 9, 355–433.
2. F. W. J. Olver and D. J. Sookne, **Math. of Comp. 26** (1972) 941–947.
3. D. E. Amos, S. L. Daniel, and M. K. Weston, *CDC 6600 subroutines IBESS and JBESS for Bessel functions $I_\nu(x)$ and $J_\nu(x)$, $x \geq 0, \nu \geq 0$*, **ACM Trans. on Math. Software 3**, 1 (March 1977) 76–92.

E. Error Procedures and Restrictions

These subroutines require $x \geq 0$, ALPHA ≥ 0, and NUM ≥ 1. Violation of any of these conditions causes an error message and an immediate return.

The subroutines attempt to anticipate and avoid overflow conditions. Intermediate overflows are avoided by dynamic rescaling. If a final value of Y would be beyond the overflow limit the value is set to $-\Omega/2$ and an error message is issued. It is assumed that the host system will set underflows to zero. No messages are issued for underflow.

If $x > (16\rho)^{-1}$ an error message is issued since no accuracy can be obtained.

Subroutines SBESYN and DBESYN each contain an internal array AJ() to hold values of $J_\nu(x)$ needed to compute $Y_\nu(x)$. The size requirement of this array varies with the machine precision and is about $3(-\log_{10}\rho)+3$. For example, for precisions of 10^{-10}, 10^{-20}, and 10^{-30} the required size is 33, 63, and 95. The array is nominally dimensioned 95 to handle all anticipated computers. An error message will be issued in the unlikely event that a larger dimension is needed.

Error messages are issued by the error message processor of Chapter 19.2.

The user should be aware that these subroutines require a substantial amount of execution time, generally increasing linearly with the sum, ALPHA+NUM.

F. Supporting Information

The source language for these subroutines is ANSI Fortran 77.

Original subroutines SBJNU, SBYNU, DBJNU, DBYNU, BESJ, and BESY were designed and programmed by W. V. Snyder and E. W. Ng, JPL, 1973, with modifications by S. Singletary in 1974. The present subroutines are modifications of the earlier subroutines to improve portability and accuracy, avoid overflows, and conform to Fortran 77. These subroutines were produced in 1984 by C. L. Lawson and S. Y. Chiu in consultation with Snyder and Ng.

Entry	**Required Files**
DBESJN	AMACH, DBESJN, DBESPQ, DERM1, DERV1, DGAMMA, ERFIN, ERMSG, IERV1
DBESYN	AMACH, DBESPQ, DBESYN, DERM1, DERV1, DGAMMA, DLGAMA, ERFIN, ERMOR, ERMSG, IERV1
SBESJN	AMACH, ERFIN, ERMSG, IERV1, SBESJN, SBESPQ, SERM1, SERV1, SGAMMA
SBESYN	AMACH, ERFIN, ERMOR, ERMSG, IERV1, SBESPQ, SBESYN, SERM1, SERV1, SGAMMA, SLGAMA

DRSBESJN

```
c      DRSBESJN
c>> 1999-01-07 DRSBESJN Krogh Added external statement.
c>> 1996-05-31 DRSBESJN Krogh  Changes to use M77CON
c>> 1994-09-01 DRSBESJN WVS Moved formats to top for C conversion
c>> 1992-04-29 DRSBESJN CAO Replaced '1' in format.
c>> 1987-12-09 DRSBESJN Lawson  Initial Code.
c--S replaces "?": DR?BESJN, ?BESJN, ?BESYN
c      DEMONSTRATION PROGRAM FOR BESSEL function.
c
      real             X(3),ALPHA(3),BJ(2),BY(2),Z,PI2
      external SBESJN, SBESYN
      integer I, N
c
      data X / 0.5E0, 1.5E0, 3.2E0 /
      data ALPHA / 1.5E0, 3.0E0, 7.8E0 /
      data PI2 / 1.57079632679489661923132169E0 /
c
  100 format(' ',4X,A1,9X,A2,11X,A7,11X,A7,12X,A1)
  200 format(' ',26X,A9,9X,A9/' ')
  300 format(' ',F6.2,5X,F6.2,4X,G15.8,5X,G15.8,G13.2)
  400 format(' ',21X,2(G15.8,5X)/' ')
c
      print 100,'X','NU','J(NU,X)','Y(NU,X)','Z'
      print 200,'J(NU+1,X)','Y(NU+1,X)'
c
      do 500 I = 1,3
        N = 2
        call SBESJN(X(I),ALPHA(I),N,BJ)
        call SBESYN(X(I),ALPHA(I),N,BY)
        Z = PI2 * X(I) * (BJ(2)*BY(1) - BJ(1)*BY(2)) - 1.E0
        print 300,X(I),ALPHA(I),BJ(1),BY(1),Z
        print 400,BJ(2),BY(2)
  500 continue
c
      end
```

ODSBESJN

```
    X          NU          J(NU,X)            Y(NU,X)             Z
                           J(NU+1,X)          Y(NU+1,X)

  0.50        1.50     0.91701694E-01     -2.5214655           0.0
                       0.92364084E-02     -14.138548

  1.50        3.00     0.60963951E-01     -2.0735416          0.12E-06
                       0.11768132E-01     -7.3619728

  3.20        7.80     0.11046740E-02     -40.619846         -0.12E-06
                       0.20715481E-03     -187.70990
```