2.4 Bessel Functions J_0, J_1, Y_0 and Y_1

A. Purpose

These subprograms compute values of the cylindrical Bessel functions of the first kind, J_0 and J_1, and of the cylindrical Bessel functions of the second kind, Y_0 and Y_1. These functions are discussed in [1] and [2].

B. Usage

B.1 Program Prototype, Single Precision

REAL X,SBESJ0,SBESJ1,SBESY0,SBESY1,W

Assign a value to X and use one of the following function references.

To compute J_0 :	**W = SBESJ0(X)**
To compute J_1 :	**W = SBESJ1(X)**
To compute Y_0 for $x > 0$:	**W = SBESY0(X)**
To compute Y_1 for $x > 0$:	**W = SBESY1(X)**

B.2 Argument Definitions

X [in] Argument of function. Require X > 0 for the Y functions.

B.3 Modifications for Double Precision

For double precision usage, change the REAL type statement to DOUBLE PRECISION, and change the function names to DBESJ0, DBESJ1, DBESY0, and DBESY1, respectively.

C. Examples and Remarks

The listing of DRSBESJ0 and ODSBESJ0 gives an example of using these subprograms to evaluate the Wronskian identity

$$z(x) = (x\pi/2)[J_1(x)Y_0(x) - J_0(x)Y_1(x)] - 1 = 0$$

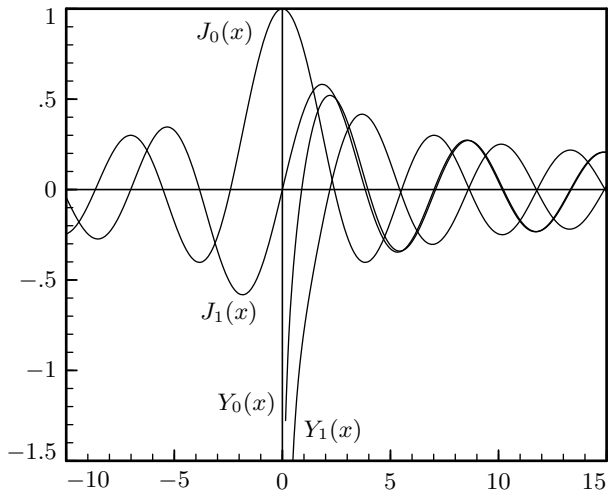
D. Functional Description

The functions J_n and Y_n are a pair of linearly independent solutions for the differential equation

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

The functions J_0 and J_1 are defined for all real x. The function J_0 is even, and J_1 is odd. As $x \to \infty$, $J_0(x)$ and $J_1(x)$ oscillate an infinite number of times about zero with an amplitude that diminishes asymptotically to $[2/(\pi x)]^{\frac{1}{2}}$, *i.e.*, approximately $0.80x^{-\frac{1}{2}}$. The distance between successive zeros approaches π as $|x| \to \infty$.

The functions Y_0 and Y_1 have real values for positive real x, approach $-\infty$ as $x \to 0^+$ and have complex values for negative real x. For large positive x the Y functions have oscillatory behavior similar to the J functions, *i.e.*, with amplitude approaching $0.80x^{-\frac{1}{2}}$ and zero spacing approaching π. As $x \to 0^+$, $Y_0(x) \to (2/\pi)\ln(x)$ and $Y_1(x) \to -(2/\pi x)$.



The Y subprograms treat $x \leq 0$ as an error condition. If the complex values of Y_0 and Y_1 for negative x are desired they may be computed from the formulae

$$\begin{aligned} Y_0(x) &= Y_0(-x) + 2iJ_0(-x) \\ Y_1(x) &= -Y_1(-x) - 2iJ_1(-x), \end{aligned} \qquad x < 0$$

where i denotes the imaginary unit. See Equation 9.1.36 in [1].

The computer approximations for these functions were developed by L. W. Fullerton, [3] and [4], using functional forms involving sine, cosine, square root, logarithm, and Chebyshev polynomial approximations. These subprograms select the polynomial degrees to adapt to machine accuracy of up to 30 decimal places.

The single precision subprograms for $J_n(x)$ and $Y_n(x)$ were tested on a Univac 1100 by comparison with the corresponding double precision subprograms over various argument ranges. The relative precision of Univac single precision arithmetic is $\rho = 2^{-27} \approx 0.745 \times 10^{-8}$.

The results show that the relative error can be very large near the zeros of any of these functions with the exception that relative accuracy can be, and is, maintained for J_1 near its zero at $x = 0$. The absolute error is large for Y_0 and Y_1 near the singularity at $x = 0$.

Test results may be summarized as follows.

Function	Argument Interval	Maximum Error	(Abs. or Rel.)
SBESJ0	[−5.6, 5.6]	6.5ρ	(Abs.)
	[1.0, 1.0E6]	$2.1\rho x^{\frac{1}{2}}$	(Abs.)
SBESJ1	[−7.2, 7.2]	3.8ρ	(Abs.)
	[1.0, 1.0E6]	$1.1\rho x^{\frac{1}{2}}$	(Abs.)
SBESY0	[0.00, 0.32]	7.6ρ	(Rel.)
	[0.32, 1.12]	6.0ρ	(Abs.)
	[1.12, 4.00]	1.9ρ	(Abs.)
	[1.0, 1.0E6]	$2.1\rho x^{\frac{1}{2}}$	(Abs.)
SBESY1	[0.0, 1.1]	6.6ρ	(Rel.)
	[1.1, 5.5]	3.0ρ	(Abs.)
	[1.0, 1.0E6]	$1.3\rho x^{\frac{1}{2}}$	(Abs.)

For the functions $J_n(x)$ and $Y_n(x)$ the absolute error is approximated by $2|x|^{\frac{1}{2}}\rho$ for large $|x|$ while the amplitude of the function values decreases like $0.8|x|^{-\frac{1}{2}}$. Thus no accuracy at all can be expected when $2|x|^{\frac{1}{2}}\rho > 0.8|x|^{-\frac{1}{2}}$, *i.e.*, when $|x| > 0.4\rho^{-1}$. The subprograms assume less than one decimal digit of accuracy could be produced when $x > 0.04\rho^{-1}$, and issue an error message.

As a test of the double precision subprograms and an additional test of the single precision subprograms the test function, $z(x)$, defined above in Section C, was evaluated in double precision and in single precision at selected points ranging from 10^0 to 10^7. The machine arithmetic accuracies were $\rho_1 = 2^{-27} \approx 0.745 \times 10^{-8}$ for single precision and $\rho_2 = 2^{-60} \approx 1.15 \times 10^{-18}$ for double precision. The magnitude of $z(x)$ computed in single precision was bounded by $8\rho_1$ for $10^0 \le x \le 10^4$ and had the value $22\rho_1$ for $x = 10^5$ and $x = 10^6$. The magnitude of $z(x)$ computed in double precision was bounded by $9\rho_2$ for $10^0 \le x \le 10^7$.

This accuracy is much greater than the individual subprograms, SBESJ0, etc., deliver for large arguments. Thus the very accurate values of $z(x)$ must be due to a functional relation existing between the algorithms implemented in the different cylindrical Bessel function subprograms.

References

1. Milton Abramowitz and Irene A. Stegun, **Handbook of Mathematical Functions**, *Applied Mathematics Series 55*, National Bureau of Standards (1966).
2. J. F. Hart et al., **Computer Approximations**, J. Wiley and Sons, New York (1968) Section 6.8.
3. L. Wayne Fullerton, **Program Library Write–up B001**. Technical report, Los Alamos Scientific Laboratory (1973).
4. L. Wayne Fullerton, *Portable special function routines*, in Wayne Cowell, editor, **Portability of Numerical Software**, *Lecture Notes in Computer Science 57*, 452–483, Springer Verlag, Berlin (1977).

E. Error Procedures and Restrictions

These subprograms return a zero result and issue an error message if

(a) $x \le 0$ for Y_0 or Y_1,

or

(b) $|x| > 0.04\rho^{-1}$ for $J_0,\ J_1,\ Y_0$ or Y_1.

The subprograms use R1MACH(4) or D1MACH(4) for ρ. The system-supplied sine and cosine subprograms may also have a cutoff value close to ρ^{-1}. If it is less than $0.04\rho^{-1}$ then there will be values of x that will pass through the tests and trigger an error message from the sine or cosine subprogram.

F. Supporting Information

The source language is ANSI Fortran 77.

Entry	Required Files
DBESJ0	AMACH, DBESJ0, DBMP0, DCSEVL, DERM1, DERV1, DINITS, ERFIN, ERMSG, IERM1, IERV1
DBESJ1	AMACH, DBESJ1, DBMP1, DCSEVL, DERM1, DERV1, DINITS, ERFIN, ERMSG, IERM1, IERV1
DBESY0	AMACH, DBESJ0, DBESY0, DBMP0, DCSEVL, DERM1, DERV1, DINITS, ERFIN, ERMSG, IERM1, IERV1
DBESY1	AMACH, DBESJ1, DBESY1, DBMP1, DCSEVL, DERM1, DERV1, DINITS, ERFIN, ERMSG, IERM1, IERV1
SBESJ0	AMACH, ERFIN, ERMSG, IERM1, IERV1, SBESJ0, SBMP0, SCSEVL, SERM1, SERV1, SINITS
SBESJ1	AMACH, ERFIN, ERMSG, IERM1, IERV1, SBESJ1, SBMP1, SCSEVL, SERM1, SERV1, SINITS
SBESY0	AMACH, ERFIN, ERMSG, IERM1, IERV1, SBESJ0, SBESY0, SBMP0, SCSEVL, SERM1, SERV1, SINITS
SBESY1	AMACH, ERFIN, ERMSG, IERM1, IERV1, SBESJ1, SBESY1, SBMP1, SCSEVL, SERM1, SERV1, SINITS

Subprograms SBESJ0, SBESJ1, SBESY0 and SBESY1 designed and developed by L. W. Fullerton, Los Alamos, 1977. Adapted to Fortran 77 and the MATH77 library by C. L. Lawson and S. Chiu, JPL, 1984.

DRSBESJ0

```
c      DRSBESJ0
c>> 2009-10-26 DRSBESJ0 Krogh Moved function calls from print statement.
c>> 1996-05-30 DRSBESJ0 Krogh Added external statement.
c>> 1994-10-19 DRSBESJ0 Krogh  Changes to use M77CON
c>> 1994-09-13 DRSBESJ0 CLL Typed all variables.
c>> 1994-09-01 DRSBESJ0 WVS Moved formats to top for C conversion
c>> 1994-08-09 DRSBESJ0 WVS set up for CHGTYP
c>> 1992-04-29 DRSBESJ0 CAO Replaced '1' in format.
c>> 1987-12-09 DRSBESJ0 Lawson  Initial Code.
c--S replaces "?": ?BESJ0, ?BESJ1, ?BESY0, ?BESY1, DR?BESJ0
c      Demonstration driver for Bessel functions.
c
c      Z = (PI / 2) * X * ( J1(X)*Y0(X) - J0(X)*Y1(X) ) - 1.0
c
      integer IX
      real              BEJ0, BEJ1, BEY0, BEY1
      external SBESJ0, SBESJ1, SBESY0, SBESY1
      real              SBESJ0, SBESJ1, SBESY0, SBESY1, PI2, X, Z
c
      data PI2 / 1.57079632679489661923132169E0 /
c
  100 format (' ',2X,A1,7X,A5,8X,A5,8X,A5,8X,A5,8X,A1/3X,'-',7X,
     *        5('-'),8X,5('-'),8X,5('-'),8X,5('-'),8X,'-')
  150 format (' ',F4.1,1X,2(F12.7,1X),3X,'-INFINITY',4X,'-INFINITY')
  200 format (' ',F4.1,1X,4(F12.7,1X),1X,G9.2)
c
      print 100, 'X','J0(X)','J1(X)','Y0(X)','Y1(X)','Z'
c
      X = 0.E0
      BEJ0 = SBESJ0(X)
      BEJ1 = SBESJ1(X)
      print 150, X, BEJ0, BEJ1
c
      do 250 IX = 5, 50, 5
        X = IX / 10.E0
        BEJ0 = SBESJ0(X)
        BEJ1 = SBESJ1(X)
        BEY0 = SBESY0(X)
        BEY1 = SBESY1(X)
        Z = PI2 * X * ( BEJ1*BEY0  - BEJ0*BEY1 ) - 1.0e0
        print 200, X,BEJ0,BEJ1,BEY0,BEY1,Z
  250 continue
      stop
c
      end
```

ODSBESJ0

X	J0(X)	J1(X)	Y0(X)	Y1(X)	Z
0.0	1.0000000	0.0000000	-INFINITY	-INFINITY	
0.5	0.9384696	0.2422685	-0.4445186	-1.4714723	-0.18E-06
1.0	0.7651977	0.4400506	0.0882570	-0.7812129	0.12E-06
1.5	0.5118276	0.5579365	0.3824489	-0.4123086	0.0
2.0	0.2238908	0.5767248	0.5103757	-0.1070324	0.12E-06
2.5	-0.0483838	0.4970941	0.4980704	0.1459181	0.0
3.0	-0.2600519	0.3390590	0.3768500	0.3246744	0.0
3.5	-0.3801277	0.1373775	0.1890219	0.4101884	0.0
4.0	-0.3971498	-0.0660434	-0.0169407	0.3979257	-0.12E-06

4.5	−0.3205426	−0.2310604	−0.1947050	0.3009973	0.24E−06
5.0	−0.1775968	−0.3275791	−0.3085176	0.1478631	−0.60E−07