ELSEVIER

Computer Physics Communications 92 (1995) 252-266

Computer Physics Communications

# RFSFNS: A portable package for the numerical determination of the number and the calculation of roots of Bessel functions

M.N. Vrahatis, O. Ragos, T. Skiniotis, F.A. Zafiropoulos, T.N. Grapsa

*Department of Mathematics, University of Patras, GR-261.10 Patras, Greece*

Received 21 March 1995; revised 21 July 1995

**Abstract**

A portable software package, named RFSFNS, is presented for the localization and computation of the simple real zeros of the Bessel functions of first and second kind, $J_\nu(z)$, $Y_\nu(z)$, respectively, and their derivatives, where $\nu \geq 0$ and $z > 0$. This package implements the topological degree theory for the localization portion and a modified bisection method for the computation one. It localizes, isolates and computes with certainty all the desired zeros of the above functions in a predetermined interval within any accuracy (subject to relative machine precision). It has been implemented and tested on different machines utilizing the above Bessel functions of various orders and several intervals of the argument.

*Keywords:* Bessel functions; Simple real zeros; Topological degree; Kronecker-Picard integral; Localization of zeros; Isolation of zeros; Bounds of zeros; Bisection method; Computation of zeros; Steed's method; Barnett's algorithm

## PROGRAM SUMMARY

*Title of program:* RFSFNS

*Catalogue number:* ADCK

*Program obtainable from:* CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

*Licensing provisions:* none

*Computer for which the program is designed and others on which it has been tested:* HP-715 (with a PA-Risc 7100/75 MHz processor) and PC IBM compatible (with an i486/66 MHz processor)

*Operating systems under which the program has been tested:* UNIX, MS-DOS

*Programming language used:* FORTRAN-77

*Memory required to execute with typical data:* Less than 60 Kbytes (using double precision)

*No. of bits in a word:* For Unix 32 bits. For MS-DOS it depends on the particular compiler used.

*No. of processors used:* One

*Has the code been vectorised?:* No

*No. of lines in distributed program, including test data, etc.:* 1671

*CPC Program Library subprograms used:* COULFG (Cat. no. ABNK; A.R. Barnett, Comput. Phys. Commun. 27 (1982) 147-166)

*Keywords:* Bessel functions, simple real zeros, topological degree, Kronecker-Picard integral, localization of zeros, isolation of zeros, bounds of zeros, bisection method, computation of zeros, Steed's method, Barnett's algorithm

*Nature of physical problem*
Bessel functions and their zeros are encountered in many problems of Mathematical Physics, such as the cyclic membrane vibrations, the temperature distribution in a solid cylinder or in a solid sphere, the diffraction of a plane electromagnetic wave by a conducting cylinder, the quantum billiards, etc.

*Method of solution*
The total number of real zeros of a Bessel function is obtained using Picard's extension and the Kronecker integral representation of the topological degree. Subsequently, a modified bisection method is employed for the computation of these zeros.

*Restrictions on the complexity of the problem*
The functions considered here are Bessel functions of first and second kind and their first derivatives. Their order is real and non-negative while their argument has to be positive

*Typical running time*
On an HP-715 computer (with a PA-Risc 7100/75 MHz processor) using the HP FORTRAN/9000 compiler, the elapsed CPU times for the computation of the total number of zeros in the given interval, their isolation and their computation, for the four test runs of Section 4, were as follows: (1) 9.02, 0.64, 29.56 msec, (2) 11.53, 0.75, 39.58 msec, (3) 14.27, 0.89, 46.87 msec and (4) 32.81, 1.91, 94.33 msec. The corresponding times on a PC IBM compatible (with an i486/66 MHz processor) using Microsoft Fortran 5.10 were on average 0.20, 0.05, 0.60 sec.

## LONG WRITE UP

## 1. Introduction

The *Bessel equation*

$$z^2 u''(z) + z\, u'(z) + \left(z^2 - \nu^2\right) u(z) = 0\,, \tag{1}$$

where $\nu$ is its order, is related to the solution of some boundary value problems or eigenvalue problems of partial differential equations by the method of separation of variables, as, for example, the wave equation in cylindrical or spherical coordinates. Its solutions are the Bessel function of order $\nu$ and argument $z$ of the first kind, $J_\nu(z)$, and the Bessel function of order $\nu$ and argument $z$ of the second kind, $Y_\nu(z)$.

The problem of localizing and computing zeros of Bessel functions has drawn a lot of attention since they are very important in many branches of physical sciences and technology. Specifically, they are involved in the problem of cyclic membrane vibrations, the temperature distribution in a solid cylinder or in a solid sphere, the diffraction of a plane electromagnetic wave by a conducting cylinder, the quantum billiards, etc.

In the present paper we develop a portable package which localizes, isolates and computes zeros of $J_\nu(z)$, $Y_\nu(z)$ as well as their first derivatives. A theoretical background, development and further applications of the methods used here can be found in [1-3].

## 2. The methods

For the computation of the total number of zeros of a Bessel function within an interval $(a,b)$ we implement the concept of the topological degree of a continuous mapping following the process briefly described below (for details we refer the interested reader to [2]).

Let $f(z)$ be a real function defined and twice continuously differentiable in a bounded interval $[a,b]$ such that $f(a)\,f(b) \neq 0$. The notion of the topological degree can be used to calculate the total number $\mathcal{N}^r$ of simple solutions of $f(z) = 0$ within $(a,b)$. According to Picard's extension and the Kronecker integral representation of the topological degree, $\mathcal{N}^r$ is given by

$$\mathcal{N}^r = -\frac{\gamma}{\pi}\int_a^b \frac{f(z)f''(z) - f'^2(z)}{f^2(z) + \gamma^2 f'^2(z)}\,dz + \frac{1}{\pi}\arctan\left(\frac{\gamma\big(f(a)f'(b) - f(b)f'(a)\big)}{f(a)f(b) + \gamma^2 f'(a)f'(b)}\right), \tag{2}$$

where $\gamma$ is an arbitrary positive constant (for a detailed derivation of Eq. (2) see for example [2]).

In the case $f(z)$ is a Bessel function, Eq. (2) has to be adapted accordingly. For example, if $f(z) = J_\nu(z)$, then

$$\mathcal{N}^r = -\frac{\gamma}{\pi}\int_a^b \frac{J_\nu(z)J_\nu''(z) - J_\nu'^2(z)}{J_\nu^2(z) + \gamma^2 J_\nu'^2(z)}\,dz + \frac{1}{\pi}\arctan\left(\frac{\gamma\big(J_\nu(a)J_\nu'(b) - J_\nu(b)J_\nu'(a)\big)}{J_\nu(a)J_\nu(b) + \gamma^2 J_\nu'(a)J_\nu'(b)}\right). \tag{3}$$

Using the Bessel equation (1) we replace the second derivative $J_\nu''$ and after some straightforward calculations, Eq. (3) assumes the form

$$\mathcal{N}^r = \frac{\gamma}{\pi}\int_a^b G_1(z)\,dz + \frac{1}{\pi}\arctan\left(\frac{\gamma\big(J_\nu(a)J_\nu'(b) - J_\nu(b)J_\nu'(a)\big)}{J_\nu(a)J_\nu(b) + \gamma^2 J_\nu'(a)J_\nu'(b)}\right), \tag{4}$$

where

$$G_1(z) = \frac{J_\nu'^2(z) + \frac{1}{z}J_\nu'(z)J_\nu(z) + \left(1 - \frac{\nu^2}{z^2}\right)J_\nu^2(z)}{J_\nu^2(z) + \gamma^2 J_\nu'^2(z)}. \tag{5}$$

In the case $f(z) = J_\nu'(z)$, Eq. (2) yields

$$\mathcal{N}^r = \frac{\gamma}{\pi}\int_a^b G_2(z)\,dz + \frac{1}{\pi}\arctan\left(\frac{\gamma\big(J_\nu'(a)J_\nu''(b) - J_\nu'(b)J_\nu''(a)\big)}{J_\nu'(a)J_\nu'(b) + \gamma^2 J_\nu''(a)J_\nu''(b)}\right), \tag{6}$$

where

$$G_2(z) = \frac{\left(1 - \frac{1+\nu^2}{z^2}\right)J_\nu'^2(z) + \left(\frac{1}{z} + \frac{\nu^2}{z^3}\right)J_\nu'(z)\,J_\nu(z) + \left(1 - \frac{\nu^2}{z^2}\right)^2 J_\nu^2(z)}{J_\nu'^2(z) + \gamma^2\left[\frac{1}{z}J_\nu'(z) + \left(1 - \frac{\nu^2}{z^2}\right)J_\nu(z)\right]^2}. \tag{7}$$

Working similarly we extract relevant expressions for higher derivatives of $J_\nu(z)$ as well as for $Y_\nu(z)$ and its derivatives.

Picard has shown that $\mathcal{N}^r$ is independent of the choice of $\gamma$. But we have observed that when $\gamma$ is close to zero, the computation time increases. Our experience, from all the test cases we have run, is that the most appropriate value of $\gamma$ is near to one. Fig. 1 presents examples of the way the computation time varies with $\gamma$. Yet, RFSFNS permits the user to make his own choice according to the application.

Very accurate values for the Bessel functions are obtained utilizing Steed's and Temme's methods [4–6]. Specifically, Steed's method calculates $J_\nu$, $J_\nu'$, $Y_\nu$ and $Y_\nu'$ simultaneously employing the Wronskian relation

$$W(z) \equiv J_\nu(z)Y_\nu'(z) - Y_\nu(z)J_\nu'(z) = \frac{2}{\pi z}, \tag{8}$$

as well as the following two continued fractions the first of which is real, while the second is complex:

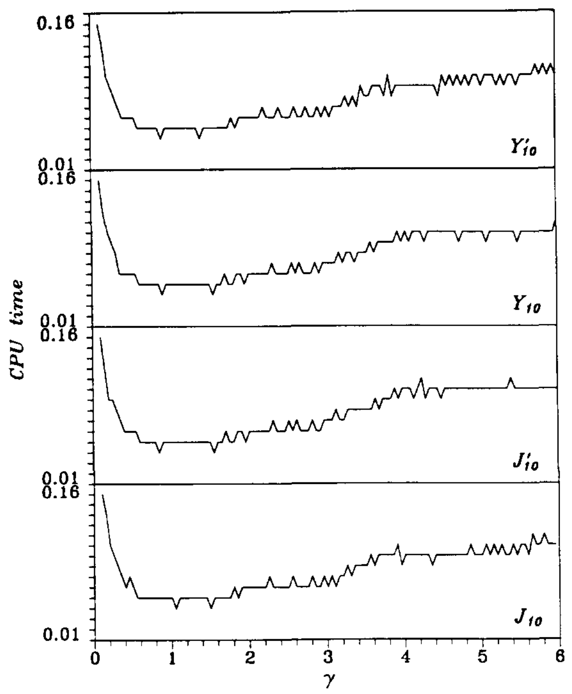

Fig. 1. The variation of the CPU time for the computation of $\mathcal{N}^r$ versus $\gamma$.

$$f_\nu(z) \equiv \frac{J'_\nu(z)}{J_\nu(z)} = \frac{\nu}{z} - \frac{J_{\nu+1}(z)}{J_\nu(z)} = \frac{\nu}{z} - \frac{1}{2(\nu+1)/z-}\,\frac{1}{2(\nu+2)/z-}\cdots, \tag{9}$$

$$p(z) + iq(z) \equiv \frac{J'_\nu(z) + iY'_\nu(z)}{J_\nu(z) + iY_\nu(z)} = -\frac{1}{2z} + i + \frac{i}{z}\,\frac{(1/2)^2 - \nu^2}{2(z+i)+}\,\frac{(3/2)^2 - \nu^2}{2(z+2i)+}\cdots. \tag{10}$$

Efficient algorithms for the evaluation of such continued fractions are given in Barnett [7] and Lentz [8].

Relations (8), (9) and (10) provide four equations to be solved for $J_\nu$, $J'_\nu$, $Y_\nu$ and $Y'_\nu$. So, introducing the auxiliary variable

$$\beta(z) \equiv \frac{p(z) - f_\nu(z)}{q(z)}, \tag{11}$$

we obtain

$$J_\nu(z) = \pm\left(\frac{W(z)}{q(z) + \beta(z)\Big(p(z) - f_\nu(z)\Big)}\right)^{1/2}, \tag{12}$$

$$J'_\nu(z) = f_\nu(z) J_\nu(z), \tag{13}$$

$$Y_\nu(z) = \beta(z) J_\nu(z), \tag{14}$$

$$Y'_\nu(z) = Y_\nu(z)\left(p(z) + \frac{q(z)}{\beta(z)}\right). \tag{15}$$

The number of iterations of the continued fraction (9) is of order $z$ for large $z$. The sign of $J_\nu$ is the same as the sign of the denominator of the continued fraction (9) once it has converged. For the derivation of (9) and (10) as well as for their rate of convergence, see [4,6].

For the calculation of the values of Bessel functions we use here Barnett's algorithm [9,10] which computes Coulomb and Bessel functions and their derivatives for real arguments, utilizing Steed's method. His code COULFG loses accuracy when the argument is less than the turning point $z_\nu$, which in the case of Bessel functions is defined as follows:

$$z_\nu = \sqrt{\nu^2 - 1/4}. \tag{16}$$

Obviously, $z_\nu < \nu$. Since the order $\nu$ is a lower bound for all the positive zeros of the cylindrical Bessel functions [11] the arguments in our case should be bounded below by $\nu$. Therefore, we are able to avoid this inaccuracy problem.

Once the total number $\mathcal{N}^r$ of simple roots has been computed, we isolate one arbitrary zero bisecting $(a,b)$ and using Bolzano's criterion so as to obtain a subinterval $(a_m, b_m)$ which contains exactly one zero of the considered function. Subsequently, we utilize a modified version of the bisection method, described in [12,13], in order to compute this particular root. The iterative formula used is the following:

$$z_{i+1} = z_i + c\ \mathrm{sgn} f(z_i)/2^{i+1}, \quad z_0 = a, \quad i = 0,1,\ldots, \tag{17}$$

where $c = \mathrm{sgn} f(a)(b-a)$ and sgn defines the sign function with values

$$\mathrm{sgn}\,\psi = \begin{cases} -1 & \text{if } \psi < 0, \\ 0 & \text{if } \psi = 0, \\ 1 & \text{if } \psi > 0. \end{cases} \tag{18}$$

The iterations (17) converge to a root $r \in (a,b)$ if, for some $z_i$, $i = 1,2,\ldots$, the following holds:

$$\mathrm{sgn} f(z_0)\,\mathrm{sgn} f(z_i) = -1, \tag{19}$$

which verifies the traditional Bolzano's criterion in the interval $(z_0, z_i)$.

The number of iterations $\xi$, which are required in order to obtain an approximate root $r^*$ such that $|r - r^*| \leq \delta$, for some $\delta \in (0,1)$, is given by

$$\xi = \left\lceil \log_2(h\,\delta^{-1}) \right\rceil, \tag{20}$$

where the notation $\lceil\cdot\rceil$ refers to the smallest integer not less than the real number quoted.

Instead of the iterative formula (17) we can also use the following one:

$$z_{i+1} = z_i - c\ \mathrm{sgn} f(z_i)/2^{i+1}, \quad z_0 = b, \quad i = 0,1,\ldots, \tag{21}$$

with $c = \mathrm{sgn} f(b)(b-a)$.

The bisection method is a global convergence method, it always converges within the given interval and it is optimal [14,15], in the sense that it possesses asymptotically the best rate of convergence. Furthermore, it has a known behavior concerning the number of iterations required when we seek a root with a predetermined accuracy. Last, but not least, it is the only method that can be applied to problems with imprecise function values. It is evident from (17) and (21) that the only computable information required by the bisection method is the algebraic signs of the considered function.

For a generalization of the above process, as well as Bolzano's criterion, see [12,13,16].

Using the above modified bisection method we are also able to produce bounds for any real zero of a Bessel function as close to this zero as the user desires (subject to relative machine precision).

## 3. Program description

The package RFSFNS (Root-Finder of Special FuNctionS) is written for the numerical localization and computation of roots of special functions. It contains about 1,400 lines of code, 50 percent of which are comments. The total storage required for RFSFNS is of the order of 4*MAXRT+2*MAXDIV+100 locations, where MAXRT determines the maximum number of roots requested in the given interval and MAXDIV is the maximum number of subdivisions of the interval for the isolation of these roots. RFSFNS is coded in ANSI standard FORTRAN (1977) [17], and has been tested on an HP-715 system as well as on a PC IBM compatible.

RFSFNS consists of a set of six subprograms, namely the subroutines INTSUB, TOPDEG, ISOLAT and BISECT and the functions FNC and G. The user must call only the subroutine INTSUB; all the other subprograms are evoked by INTSUB. RFSFNS requires also the subroutine COULFG [9] of the CPC Program Library.

INTSUB is an interface subroutine between the main program and the subroutines TOPDEG, ISOLAT and BISECT.

The purpose of TOPDEG (*top*ological *deg*ree) is to compute the total number NR of roots of a Bessel function in a given interval. It implements the topological degree of a continuous mapping and especially the Kronecker–Picard integral. The corresponding integration is performed by Romberg's method [6].

When the total number NR of roots is computed, ISOLAT (*isolat*ion of all the zeros) isolates the roots in NR intervals using Bolzano's criterion, so that each one of these intervals contains exactly one root. The left and right endpoints of these intervals are stored in the two columns of the matrix ROOTIS.

The purpose of BISECT (*bisect*ion) is to compute the roots which have been isolated by the subroutine ISOLAT. It utilizes the modified bisection method described by Scheme (17). The bisection portion of RFSFNS is normally evoked when the total number NR of roots is computed and their isolation is completed; but the user can compute a desired number of roots without involving TOPDEG or ISOLAT (see description of input variable ICON below).

The values of the Bessel functions are calculated by means of the function FNC. The relative definition statement is

```
REAL*8 FUNCTION FNC( ICASE, X )
```

and the function calculates the corresponding Bessel function determined by the input parameter ICASE (see below) at X utilizing the COULFG subroutine.

The function G calculates the values of the corresponding integrand function in Relation (2). Its definition statement is

```
REAL*8 FUNCTION G( ICASE, X )
```

Complete details for each one of the above subprograms are given in the RFSFNS documentation.

RFSFNS must be supplied with the following input parameters:

| | |
|---|---|
| ICASE | an integer variable specifying the desired Bessel function with values<br>(1) for the Bessel function of the first kind, $J$,<br>(2) for the derivative of $J$,<br>(3) for the Bessel function of the second kind, $Y$,<br>(4) for the derivative of $Y$. |
| XNU | the order of the considered Bessel function. |
| A | the left endpoint of the given interval. If A is less than DMAX1(XNU, 0.5D0) then A becomes equal to DMAX1(XNU, 0.5D0). |
| B | the right endpoint of the given interval. B should be greater than A. |
| ICON | a conditional integer variable with values<br>(1) for the calculation of the total number of roots in the given interval, only,<br>(2) for the calculation of the total number of roots in the given interval and for isolating each one of them,<br>(3) for the calculation of the total number of roots in the given interval, isolation and computation of each one of them,<br>(4) for the calculation of NR roots in the given interval.<br>Note that if ICON=4 the user must also supply the desired number of roots NR (see below). |
| EPSILO | the accuracy of the computation of the Kronecker–Picard integral. If EPSILO is less than the machine |

| | |
|---|---|
| | precision EPSMCH, EPSILO becomes equal to 1.D5*DSQRT(EPSMCH). EPSMCH is computed within RFSFNS. |
| DELTA | the accuracy of the computation of the roots of the function. If DELTA is less than the machine precision EPSMCH, DELTA becomes equal to EPSMCH. |

The default value of GAMMA, involved in Relation (2), is chosen to be 1.D0. But it is at the user's disposal to change it in his own main program. In this case, if the given value of GAMMA is less than 0.001D0, RFSFNS sets GAMMA equal to 1.D0.

INTSUB is evoked by the following FORTRAN statement:

```
CALL INTSUB( ICASE, A, B, ICON, EPSILO, DELTA, MAXRT, NR,
+              ROOTIS, ROOTS, VAR )
```

The usage of some of its arguments (ICASE,A,B,ICON,EPSILO,DELTA) has already been described above. The rest of these arguments are the following:

| | |
|---|---|
| MAXRT | is a positive integer input variable which determines the maximum number of roots that may be requested in the given interval. |
| NR | is a positive integer variable which indicates the number of roots found. If TOPDEG is going to be used (ICON=1,2 or 3) then NR outputs the total number of roots within (A,B). In the case that ICON=4 it inputs the number of roots requested by the user. NR must be less than MAXRT. |
| ROOTIS | is an output MAXRT$\times$2 array. Its two columns provide lower and upper root bounds. |
| ROOTS | is an output array of length MAXRT. It contains the final approximate roots. |
| VAR | is an array of length MAXRT which outputs the function values at the final approximate roots. |

The program execution terminates normally after the completion of its task. This type of termination is indicated by the value 1 of the conditional output variable INF. If the value of this parameter is not equal to 1, the termination of the program is abnormal.

The cases of abnormal termination of the program are the following:

| | |
|---|---|
| INF=0 | the order XNU of the considered function is negative, or<br>the input values of ICASE or ICON are out of range ([1,4]), or<br>NR exceeds MAXRT. |
| INF=2 | FNC or G failed. |
| INF=3 | TOPDEG was not successful; the evaluation of the integral in Formula (2) failed. |
| INF=4 | ISOLAT failed; MAXDIV, the maximum number of steps or the maximum dimension of its internal working arrays WA1,WA2 was exceeded. |
| INF=5 | BISECT failed; the number of iterations of Scheme (17) exceeded its maximum, specified by Relation (20). |

## 4. Example of RFSFNS usage

We shall give an example which demonstrates how RFSFNS is used to compute the total number of zeros within a given interval, to isolate these zeros and to compute all of them. Suppose that we wish to calculate the roots of the zero order Bessel function of the first kind, $J_0$, in the interval (A, B) = (0.D0, 30.1D0).

The corresponding input values are ICASE = 1, XNU = 0.D0, A = 0.D0, B = 30.1D0, ICON = 3 and NR = 0. Also, we choose the values EPSILO = 1.D-3 and DELTA = 1.D-15.

The following FORTRAN program can be used to evoke RFSFNS for the above example:

```
*-------------------------------------------------------------
      PROGRAM MAIN
*
*     Example of RFSFNS usage.
```

```
*
      IMPLICIT REAL*8(A-H, O-Z), INTEGER*4(I-N)
      PARAMETER (MAXRT=1000)
      DIMENSION  ROOTIS(MAXRT,2), ROOTS(MAXRT), VAR(MAXRT)
      COMMON / BLK1 / INF, NRF
      COMMON / BLK2 / XNU, GAMMA
*
*     Set the starting values :
*         a) the considered Bessel function specified by ICASE,
*         b) the order of the corresponding Bessel function XNU,
*         c) the endpoints of the given interval [A, B] and
*         d) the conditional variable ICON. If ICON = 4, supply
*            also the desired number of roots NR.
*
      DATA   ICASE, XNU,    A,     B,       ICON, NR
     +     / 1,     0.DO,   0.DO,  30.1DO,  3,    0  /
*
*     Set the values of EPSILO and DELTA.
*
      DATA   EPSILO, DELTA
     +     / 1.D-3,  1.D-15 /
*
*     GAMMA is a parameter involved in the Kronecker-Picard
*           integral. Its default value is taken equal to 1.DO
*           but the user is able to change it in his own main
*           program. If this value is less than the machine
*           precision EPSMCH, INTSUB sets GAMMA equal to 1.DO.
*
      GAMMA=1.DO
*
      PRINT 9999, ICASE, XNU, A, B, ICON, EPSILO, DELTA
*
*     Call the interface subroutine INTSUB.
*
      CALL INTSUB (ICASE, A, B, ICON, EPSILO, DELTA, MAXRT, NR,
     +              ROOTIS, ROOTS, VAR)
*
      IF ( INF .EQ. 0 ) THEN
         PRINT 9998
         GO TO 10
      ENDIF
      IF ( INF .EQ. 2 ) THEN
         PRINT 9997
         GO TO 10
      ENDIF
      IF ( INF .EQ. 3 ) THEN
         PRINT 9996
         PRINT 9995, NR
         GO TO 10
      ENDIF
      PRINT 9995, NR
      IF ( ICON .EQ. 1 ) GO TO 10
      IF ( INF .EQ. 4 ) THEN
         PRINT 9994
         GO TO 10
      ENDIF
      PRINT 9990, NR,NRF
      PRINT 9993, (J, ROOTIS(J,1), ROOTIS(J,2), J = 1, NRF)
      IF ( ICON .EQ. 2 ) GO TO 10
      IF ( INF .EQ. 5 ) THEN
         PRINT 9992
         GO TO 10
      ENDIF
      PRINT 9991, (J, ROOTS(J), VAR(J), J = 1, NRF)
      PRINT 9989, INF
*
   10 STOP
*
 9999 FORMAT (/2X, ' STARTING VALUES :' /3X, 17('-'),
     +        /2X, ' ICASE  :    ', I1,
     +        /2X, ' ORDER  : ', F20.15,
     +        /2X, ' A      : ', F20.15,
     +        /2X, ' B      : ', F20.15,
     +        /2X, ' ICON   :    ', I1,
     +        /2X, ' EPSILO : ', F20.15,
     +        /2X, ' DELTA  : ', F20.15 )
 9998 FORMAT (/2X, '  * * * IMPROPER INPUT PARAMETERS * * *'//)
 9997 FORMAT (/2X, '  * * * THE PROCEDURE FOR THE CALCULATION',
     +             ' OF THE BESSEL FUNCTION FAILED * * *'//)
 9996 FORMAT (/2X, '  * * * THE PROCEDURE FOR THE CALCULATION',
     +             ' OF THE TOPOLOGICAL DEGREE FAILED * * *'//)
 9995 FORMAT (/2X, ' THE COMPUTED TOTAL NUMBER OF ROOTS',/,
     +         2X, ' WITHIN THE INTERVAL (A,B) IS    : ', I5)
 9994 FORMAT (/2X, ' * * * THE PROCEDURE FOR THE ROOTS',
     +             ' ISOLATION FAILED * * *'//)
 9993 FORMAT (/2X, ' INTERVALS OF THE ISOLATED ROOTS :'
     +        /3X, 33('-') /(2X, I4,')',
     +         2X, ' (', F20.15, ',', F20.15,' )'))
 9992 FORMAT (/2X, '* * * THE ROOTFINDING PORTION',
     +             ' FAILED * * *'//)
 9991 FORMAT (/2X, ' FINAL APPROXIMATE ROOTS :',
     +         7X, ' VERIFICATION :' /3X, 25('-'), 8X, 14('-')
     +             /(2X, I4,')', F20.15, 5X, F20.15))
 9990 FORMAT (/2X, ' NUMBER OF ROOTS REQUESTED       : ',I5,/
     +        /2X, ' NUMBER OF ROOTS ISOLATED        : ',I5)
 9989 FORMAT (/2X, ' EXIT PARAMETER :   INF = ',I2)
```

```
*
*     Last statement of the main program.
*
      END
*-----------------------------------------------------------------
```

The output results of the above program for four test cases are exhibited in the test run outputs below.

## 5. Concluding remarks

The program RFSFNS has been applied using Bessel functions of various orders and random intervals. Tests were carried out for orders and interval lengths up to one thousand. It behaves predictably and accurately. It calculates with certainty the total number of roots in a given interval, isolates each one of them and then computes these roots.

Generally, the total number `NR` of the zeros in `(A,B)` is not known beforehand, thus we must assign the value 3 to the input parameter `ICON`. Besides, the knowledge of the exact number is, in many cases, of great importance.

Nevertheless, we may wish to avoid this portion of the program. For example, if the number of zeros is known, say 9, we could assign the value 4 to `ICON` and the value 9 to `NR`. If the value assigned to `NR` is greater than the total number of zeros, say 12, then we shall obtain the 9 existing roots. Also, if less than 9 zeros are required, for example 2 of them, we should assign the corresponding value to `NR`. In this case ISOLAT will subdivide the initial interval `(A,B)` until two of the roots are isolated. Then BISECT will be evoked to compute them.

Our package can be applied to any special function, provided there exists a routine to compute it and its derivative.

## Acknowledgements

The authors wish to express their profound gratitude to Dr. A.R. Barnett for most kindly permitting the usage of his code COULFG as well as for his useful advice. They, also, wish to acknowledge the referee's most constructive suggestions.

## References

[1] M.N. Vrahatis, O. Ragos, F.A. Zafiropoulos and T.N. Grapsa, Locating and Computing Zeros of Airy Functions, Z. Angew. Math. Mech. 75 (1995) 9.

[2] M.N. Vrahatis, T.N. Grapsa, O. Ragos and F.A. Zafiropoulos, On the localization and computation of zeros of Bessel functions, submitted.

[3] M.N. Vrahatis, O. Ragos, T. Skiniotis, F.A. Zafiropoulos and T.N. Grapsa, Locating and computing complex zeros of Bessel functions, submitted.

[4] A.R. Barnett, D.H. Feng, J.W. Steed and L.J.B. Goldfarb, Comput. Phys. Commun. 8 (1974) 377.

[5] N.M. Temme, J. Comput. Phys. 21 (1976) 343.

[6] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, Numerical Recipes in Fortran, The Art of Scientific Computing, 2nd Ed. (Cambridge University Press, Cambridge, 1992).

[7] A.R. Barnett, Comput. Phys. Commun. 21 (1981) 297.

[8] W.J. Lentz, Applied Optics 15 (1976) 668.

[9] A.R. Barnett, Comput. Phys. Commun. 27 (1982) 147.

[10] A.R. Barnett, Comput. Phys. Commun. 24 (1981) 141.

[11] G.N. Watson, A treatise on the theory of Bessel functions (Cambridge University Press, Cambridge, 1966).

[12] M.N. Vrahatis, ACM Trans. Math. Software 14 (1988) 312.
[13] M.N. Vrahatis, ACM Trans. Math. Software 14 (1988) 330.
[14] K. Sikorski, Numer. Math. 40 (1982) 111.
[15] K. Sikorski and G.M. Trojan, Numer. Math. 57 (1990) 421.
[16] M.N. Vrahatis, Proc. Amer. Math. Soc. 107 (1989) 701.
[17] American National Standards Institute, ANSI FORTRAN X3.9-1978 (ANSI, New York, 1978) [Also known as FORTRAN 77].

## TEST RUN OUTPUT 1

```
STARTING VALUES :
-----------------
ICASE  :    1
ORDER  :      .000000000000000
A      :      .000000000000000
B      :    30.100000000000000
ICON   :    3
EPSILO :      .001000000000000
DELTA  :      .000000000000001
THE COMPUTED TOTAL NUMBER OF ROOTS
WITHIN THE INTERVAL (A,B) IS    :     9
NUMBER OF ROOTS REQUESTED       :     9
NUMBER OF ROOTS ISOLATED        :     9
INTERVALS OF THE ISOLATED ROOTS :
---------------------------------
  1)  (    .500000000000000,   3.788888888888890 )
  2)  (   3.788888888888890,   7.077777777777779 )
  3)  (   7.077777777777779,  10.366666666666667 )
  4)  (  10.366666666666667,  13.655555555555558 )
  5)  (  13.655555555555558,  16.944444444444447 )
  6)  (  16.944444444444447,  20.233333333333334 )
  7)  (  20.233333333333334,  23.522222222222226 )
  8)  (  23.522222222222226,  26.811111111111113 )
  9)  (  26.811111111111113,  30.100000000000000 )

FINAL APPROXIMATE ROOTS :         VERIFICATION :
-------------------------         --------------
  1)   2.404825557695772          .000000000000000
  2)   5.520078110286310          .000000000000000
  3)   8.653727912911013          .000000000000000
  4)  11.791534439014282          .000000000000000
  5)  14.930917708487784          .000000000000000
  6)  18.071063967910918         -.000000000000001
  7)  21.211636629879257          .000000000000000
  8)  24.352471530749303          .000000000000000
  9)  27.493479132040262         -.000000000000001

EXIT PARAMETER :    INF =  1
```

## TEST RUN OUTPUT 2

```
STARTING VALUES :
-----------------
ICASE  :    2
ORDER  :    3.140000000000000
A      :   10.500000000000000
B      :   45.200000000000000
ICON   :    3
EPSILO :     .001000000000000
DELTA  :     .000000000000001

THE COMPUTED TOTAL NUMBER OF ROOTS
WITHIN THE INTERVAL (A,B) IS     :    11

NUMBER OF ROOTS REQUESTED        :    11

NUMBER OF ROOTS ISOLATED         :    11

INTERVALS OF THE ISOLATED ROOTS :
---------------------------------
  1)   (  10.500000000000000,  13.654545454545453 )
  2)   (  13.654545454545453,  16.809090909090909 )
  3)   (  16.809090909090909,  19.963636363636365 )
  4)   (  19.963636363636365,  23.118181818181818 )
  5)   (  23.118181818181818,  26.272727272727271 )
  6)   (  26.272727272727271,  29.427272727272731 )
  7)   (  29.427272727272731,  32.581818181818179 )
  8)   (  32.581818181818179,  35.736363636363632 )
  9)   (  35.736363636363632,  38.890909090909093 )
 10)   (  38.890909090909093,  42.045454545454546 )
 11)   (  42.045454545454546,  45.200000000000000 )

FINAL APPROXIMATE ROOTS :        VERIFICATION :
-------------------------        --------------
  1)  11.585290479133392        -.000000000000001
  2)  14.817793490469953        -.000000000000001
  3)  18.017150352437752        -.000000000000003
  4)  21.198859787655240         .000000000000000
  5)  24.369995856264936         .000000000000000
  6)  27.534276360895168         .000000000000001
  7)  30.693850896882283         .000000000000001
  8)  33.850051913205407         .000000000000001
  9)  37.003750546240724         .000000000000000
 10)  40.155540912559330         .000000000000002
 11)  43.305842381541125         .000000000000001

EXIT PARAMETER :   INF =  1
```

## TEST RUN OUTPUT 3

```
STARTING VALUES :
-----------------
ICASE  :    3
ORDER  :   10.000000000000000
A      :   15.300000000000000
B      :   55.700000000000000
ICON   :    3
EPSILO :     .001000000000000
DELTA  :     .000000000000001

THE COMPUTED TOTAL NUMBER OF ROOTS
WITHIN THE INTERVAL (A,B) IS     :     12

NUMBER OF ROOTS REQUESTED        :     12

NUMBER OF ROOTS ISOLATED         :     12

INTERVALS OF THE ISOLATED ROOTS :
---------------------------------
  1)   (  15.300000000000000,  18.666666666666668 )
  2)   (  18.666666666666668,  22.033333333333334 )
  3)   (  22.033333333333334,  25.400000000000002 )
  4)   (  25.400000000000002,  28.766666666666668 )
  5)   (  28.766666666666668,  32.133333333333339 )
  6)   (  32.133333333333339,  35.500000000000000 )
  7)   (  35.500000000000000,  38.866666666666671 )
  8)   (  38.866666666666671,  42.233333333333335 )
  9)   (  42.233333333333335,  45.600000000000007 )
 10)   (  45.600000000000007,  48.966666666666671 )
 11)   (  48.966666666666671,  52.333333333333343 )
 12)   (  52.333333333333343,  55.700000000000000 )

FINAL APPROXIMATE ROOTS :         VERIFICATION :
-------------------------         --------------
  1)  16.447852748486492         -.000000000000001
  2)  20.223031412681701          .000000000000000
  3)  23.760715860327446          .000000000000000
  4)  27.182021527190530          .000000000000000
  5)  30.534504754007071         -.000000000000001
  6)  33.841965775135710          .000000000000001
  7)  37.118000423665612          .000000000000001
  8)  40.371068905333876         -.000000000000001
  9)  43.606764901379510          .000000000000000
 10)  46.828959446564562         -.000000000000001
 11)  50.040428970943443         -.000000000000001
 12)  53.243223214220538          .000000000000001
```

```
EXIT PARAMETER :   INF =  1
```

**TEST RUN OUTPUT 4**

```
STARTING VALUES :
-----------------
ICASE  :    4
ORDER  :   55.500000000000000
A      :  100.100000000000000
B      :  150.200000000000000
ICON   :    3
EPSILO :     .001000000000000
DELTA  :     .000000000000001

THE COMPUTED TOTAL NUMBER OF ROOTS
WITHIN THE INTERVAL (A,B) IS     :    14

NUMBER OF ROOTS REQUESTED        :    14

NUMBER OF ROOTS ISOLATED         :    14

INTERVALS OF THE ISOLATED ROOTS :
---------------------------------
  1)   ( 100.100000000000000, 103.678571428571410 )
  2)   ( 103.678571428571410, 107.257142857142850 )
  3)   ( 107.257142857142850, 110.835714285714260 )
  4)   ( 110.835714285714260, 114.414285714285710 )
  5)   ( 114.414285714285710, 117.992857142857130 )
  6)   ( 117.992857142857130, 121.571428571428550 )
  7)   ( 121.571428571428550, 125.149999999999980 )
  8)   ( 125.149999999999980, 128.728571428571430 )
  9)   ( 128.728571428571430, 132.307142857142850 )
 10)   ( 132.307142857142850, 135.885714285714260 )
 11)   ( 135.885714285714260, 139.464285714285730 )
 12)   ( 139.464285714285730, 143.042857142857130 )
 13)   ( 143.042857142857130, 146.621428571428560 )
 14)   ( 146.621428571428560, 150.200000000000000 )

FINAL APPROXIMATE ROOTS :         VERIFICATION :
-------------------------         --------------
  1) 102.349963347284800           .000000000000000
  2) 106.062557620259270           .000000000000000
  3) 109.726598816333270           .000000000000000
  4) 113.348911193323370          -.000000000000001
  5) 116.934973022143080           .000000000000001
  6) 120.489253965099970          -.000000000000001
  7) 124.015451547702800           .000000000000001
  8) 127.516661274097680          .000000000000000
  9) 130.995501771667760          .000000000000000
 10) 134.454208656554660         -.000000000000001
 11) 137.894706145736210          .000000000000000
 12) 141.318662519059830         -.000000000000002
 13) 144.727533652350420          .000000000000000
 14) 148.122597599802940         -.000000000000002

EXIT PARAMETER :   INF =  1
```

ELSEVIER

Computer Physics Communications 117 (1999) 290

Computer Physics Communications

Erratum

# Erratum to: "RFSFNS: A portable package for the numerical determination of the number and the calculation of roots of Bessel functions" [Comput. Phys. Commun. 92 (1995) 252–266] *

M.N. Vrahatis, O. Ragos, T. Skiniotis, F.A. Zafiropoulos, T.N. Grapsa

*Department of Mathematics, University of Patras, GR-261.10, Patras, Greece* [1]

**Program information**

*Title of program:* RFSFNS

*Catalogue identifier:* ADCK

**Description of the erratum**

In the description of the input parameter ICASE of the package (Section 3, p. 257), whose value determines which of the Bessel functions $J_\nu(z)$, $J'_\nu(z)$, $Y_\nu(z)$ and $Y'_\nu(z)$ will be considered by the program, it is written that

ICASE an integer variable specifying the desired Bessel function with values
(1) for the Bessel function of the first kind, $J$,
(2) for the derivative of $J$,
(3) for the Bessel function of the second kind, $Y$,
(4) for the derivative of $Y$.

The correct correspondence of the values of ICASE to the above functions is the following:

ICASE an integer variable specifying the desired Bessel function with values
(1) for the Bessel function of the first kind, $J$,
(2) for the Bessel function of the second kind, $Y$,
(3) for the derivative of $J$,
(4) for the derivative of $Y$.

Relevant corrections are needed in the introductory comments of the subprograms INTSUB, TOPDEG, ISOLAT, BISECT, FNC and G of RFSFNS.

* SSDI of original article: 0010-4655(95)00115-8.

[1] E-mail: {vrahatis,ragos,stakis,phikapa,grapsa}@math.upatras.gr

*PII* S0010-4655(98)00109-X