ELSEVIER

Computer Physics Communications 113 (1998) 220–238

Computer Physics Communications

ZEBEC: A mathematical software package for computing simple zeros of Bessel functions of real order and complex argument

P. Kravanja [a,1], O. Ragos [b,c,2], M.N. Vrahatis [b,c,3], F.A. Zafiropoulos [b,c,4]

[a] *Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200 A, B-3001 Heverlee, Belgium*

[b] *Department of Mathematics, University of Patras, GR-26110, Patras, Greece*

[c] *University of Patras Artificial Intelligence Research Center (UPAIRC), GR-26110, Patras, Greece*

Received 7 April 1998

Abstract

A reliable and portable software package, called ZEBEC (ZEros of BEssel functions Complex), is presented, which localizes and computes simple zeros of Bessel functions of the first, the second or the third kind, or their derivatives. The Bessel functions are of real order and complex argument. ZEBEC calculates with certainty the total number of zeros within a given box whose edges are parallel to the coordinate axes. Cauchy's Theorem is used for this calculation. Then the program isolates each one of the zeros, utilizing the above-mentioned theorem, and finally computes them to a given desired accuracy using a generalized method of bisection. © 1998 Elsevier Science B.V.

PACS: 02.30.Gp; 02.30.Dk

Keywords: Bessel functions; Simple complex zeros; Cauchy's Theorem; Logarithmic residue integral; Quadrature method; Isolation of zeros; Computation of zeros; Generalized bisection method

PROGRAM SUMMARY

Title of program: ZEBEC

Catalogue identifier: ADIO

Program Summary URL:
http://www.cpc.cs.qub.ac.uk/cpc/summaries/ADIO

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland

Licensing provisions: none

Computer for which the program is designed and others on which it is operable:
DEC DS5000/7012, HP 9000 B160L, IBM RS6000 7012, SUN SPARC Ultra-2 m1170, and PC IBM compatible (with an Intel Pentium/133 MHz processor)

Operating systems under which the program has been tested:
UNIX (ULTRIX V4.4, HPUX 10.20, AIX 3.2.5, SunOS 5.5.1) and MS-DOS

[1] E-mail: Peter.Kravanja@na-net.ornl.gov
[2] E-mail: ragos@math.upatras.gr
[3] E-mail: vrahatis@math.upatras.gr
[4] E-mail: phikapa@math.upatras.gr

PII S0010-4655(98)00064-2

Programming language used: FORTRAN-77

Memory required to execute with typical data: Less than 500 Kbytes (in double precision)

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

No. of bytes in distributed program, including test data, etc.: 267374

Distribution format: ASCII

CPC Program Library subprograms used: BESSCC, I.J. Thompson, A.R. Barnett, Comput. Phys. Commun. 47 (1987) 245–257.

Keywords: Bessel functions, simple complex zeros, Cauchy's Theorem, logarithmic residue integral, quadrature method, isolation of zeros, computation of zeros, generalized bisection method

Nature of physical problem
Bessel functions and their zeros or turning points are encountered in many problems of mathematical physics, e.g. 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, quantum billiards, etc.

Method of solution
The number of zeros inside a given box is calculated by evaluating a logarithmic residue integral along the edges of this box. The longest edges of the box are then cut in half and the box is split into two equal boxes. This process is repeated until a set of boxes is obtained, each of which contains precisely one zero. The zeros are then calculated via a generalized method of bisection.

Restrictions on the complexity of the problem
The functions considered here are Bessel functions of the first, the second or the third kind, or their derivatives. They are of real order and complex argument. The package calculates all the simple zeros that lie inside a given box in the complex plane cut along the nonpositive real axis. The edges of this box are to be parallel to the coordinate axes. In general, we have not found any restrictions in the applicability of the package. The only restriction we have found is that, for real values of the argument greater than 25, some of the Bessel functions are not evaluated correctly through BESSCC. To overcome this problem one should use the specialized package RFSFNS [1], when looking for real zeros greater than 25, while, for computing complex zeros with real part greater than 25, BESSCC performs fine as long as the considered box used by ZEBEC does not cross the real axis.

Typical running time
The running times (in seconds) for the four test runs of Section 5, are given in the following table:

	DEC	HP	IBM	SUN
Test run 1	1.8 (1.3)	0.55 (0.43)	2.0 (1.0)	0.61 (0.53)
Test run 2	93.1 (62.8)	28.0 (22.0)	91.0 (52.0)	28.0 (24.0)
Test run 3	102.2 (67.3)	28.0 (22.0)	98.0 (57.0)	29.0 (24.0)
Test run 4	38.2 (25.6)	11.0 (8.9)	40.0 (23.0)	12.0 (9.8)

The subroutine DTIME was used for timing on the UNIX machines. The parenthesized times have been obtained by optimizing the code using the option +O1 during compilation.

LONG WRITE-UP

1. Introduction

The *Bessel equation*

$$z^2u''(z) + zu'(z) + (z^2 - \nu^2)u(z) = 0, \quad z, \nu \in \mathbb{C}, \tag{1}$$

appears in many problems of mathematical physics. Two linearly independent solutions are given by the *Bessel function of the first kind* of order ν,

$$J_\nu(z) = \sum_{k=0}^{\infty} \frac{(-1)^k (z/2)^{\nu+2k}}{k!\,\Gamma(k+\nu+1)}, \quad |\arg z| < \pi,$$

and the *Bessel function of the second kind* of order ν (also called *Neumann's function*),

$$Y_\nu(z) = \frac{J_\nu(z)\cos(\nu\pi) - J_{-\nu}(z)}{\sin(\nu\pi)}, \quad |\arg z| < \pi, \quad \nu \in \mathbb{C} \setminus \mathbb{Z}.$$

For integral ν, the right-hand side becomes indeterminate, and in this case

$$Y_n(z) = \lim_{\nu \to n} Y_\nu(z), \quad n \in \mathbb{Z}.$$

Also of interest are the *Bessel functions of the third kind*, or *Hankel functions*,

$$H_\nu^{(1)}(z) = J_\nu(z) + iY_\nu(z) \quad \text{and} \quad H_\nu^{(2)}(z) = J_\nu(z) - iY_\nu(z).$$

The above considered functions are analytic with respect to z in the complex plane cut along the nonpositive real axis and entire functions of the order ν for fixed z. They also have several other interesting features [2].

The zeros and turning points of Bessel functions are important in many branches of physical sciences and technology. They appear 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, quantum billiards, etc.

Some theoretical results about these zeros and turning points are available: inequalities, bounds, regions of existence or non-existence, power series expansions, Chebyshev series expansions, and qualitative results concerning their location [3–14,2].

Hurwitz's Theorem, for example, gives information about the zeros of J_ν where ν is real.

Theorem 1. Let ν be an arbitrary real number, and suppose that $|\arg z| < \pi$. Then the function $J_\nu(z)$ has an infinite number of positive real zeros, and a finite number $2N(\nu)$ of conjugate complex zeros, where

(i) $N(\nu) = 0$ if $\nu > -1$ or $\nu = -1, -2, \ldots$;

(ii) $N(\nu) = m$ if $-(m+1) < \nu < -m, m = 1, 2, \ldots$.

In the second case, if $[-\nu]$ is odd, then there is a pair of purely imaginary zeros among the conjugate complex zeros.

It is known that any solution of (1) has only simple zeros, except possibly at $z = 0$ [15, p. 79], [2, p. 479]. In particular, this holds for J_ν, Y_ν, $H_\nu^{(1)}$ and $H_\nu^{(2)}$. Besides, the derivative of any solution of (1) also has only simple zeros, except possibly at $z = 0$ or $z = \pm\nu$ [16–18].

In [19,20] a software package is presented for computing zeros of $J_\nu(x)$, where $x > 0$ and $\nu > -1$, and turning points of $J_\nu(x)$, where $x > 0$ and $\nu > 0$. The package RFSFNS [1] can be used to calculate zeros and turning points of $J_\nu(x)$ and $Y_\nu(x)$, where $x > 0$ and $\nu \geq 0$. In this paper we present a reliable and portable software package, called ZEBEC (ZEros of BEssel functions Complex), for computing simple zeros or turning points of $J_\nu(z)$, $Y_\nu(z)$, $H_\nu^{(1)}(z)$ and $H_\nu^{(2)}(z)$, where the argument z belongs to the complex plane cut along the nonpositive real axis and the order ν is real. ZEBEC is capable of calculating all the zeros or turning points that lie inside a rectangle whose edges are parallel to the coordinate axes.

2. The method

The main phases of our method are the following:

(a) Calculation of the total number of zeros within a predetermined region.

(b) Isolation of all the zeros by consecutive subdivisions of the initial region.

(c) Computation of each zero.

For the first two phases we employ logarithmic residue integrals, while for phase (c) a generalized method of bisection is utilized. All these will be briefly explained in the sequel.

Let W be a simply connected region in $\mathbb{C} \setminus \{z \in \mathbb{C} : \operatorname{Re} z \leq 0; \operatorname{Im} z = 0\}$, f one of the Bessel functions mentioned in the introduction, and γ a positively oriented Jordan curve in W that does not pass through any

zero of f. As f is analytic in W, the total number of zeros of f that lie inside γ is given by the following *logarithmic residue* integral [21]:

$$N = \frac{1}{2\pi i} \int_{\gamma} \frac{f'(z)}{f(z)}\, dz\,. \tag{2}$$

Suppose that a parametric representation of the curve γ is given by $z = \gamma(t)$, $a \le t \le b$, and that the mapping $\gamma : [a,b] \to \mathbb{C}$ is piecewise continuously differentiable in $[a,b]$. Thus, a partition $a = t_0 < t_1 < \cdots < t_{r-1} < t_r = b$ of the interval $[a,b]$ can be found, such that the restriction of γ to each subinterval $[t_{j-1}, t_j]$ is continuously differentiable.

Let $W^* = \{\, (x,y) \in \mathbb{R}^2 \,:\, x + iy \in W \,\}$ and $u, v : W^* \to \mathbb{R}$ with

$$u(x,y) = \operatorname{Re} f(x+iy) \quad \text{and} \quad v(x,y) = \operatorname{Im} f(x+iy)\,.$$

The restriction of u and v on the curve γ is

$$u(t) = u\Big(x(t), y(t)\Big) \quad \text{and} \quad v(t) = v\Big(x(t), y(t)\Big)\,,$$

where $x(t) = \operatorname{Re}\gamma(t)$, $y(t) = \operatorname{Im}\gamma(t)$ and $t \in [a,b]$. Then, by using the Cauchy–Riemann equations for u and v, one can easily verify that

$$N = \frac{1}{2\pi} \sum_{j=1}^{r} \int_{t_{j-1}}^{t_j} \frac{u\,\frac{dv}{dt} - v\,\frac{du}{dt}}{u^2 + v^2}\, dt\,, \tag{3}$$

where we have considered the above partition of $[a,b]$. This formula for N is the Kronecker integral for the topological degree of the real mapping $F = (u,v)$ at the origin relative to the interior of γ. For an introduction to degree theory we refer the interested reader to [22]. Also, for a detailed description of the Kronecker integral, see [23,24].

Let us consider the case that γ is a rectangle whose edges are parallel to the coordinate axes. Suppose that it has a left lower vertex (x_0, y_0) and that its edges have length h_1 and h_2. In other words, suppose that γ is the boundary of $[x_0, x_0 + h_1] \times [y_0, y_0 + h_2]$. Define the functions ψ and φ in W^* as

$$\psi(x,y) = \frac{u(x,y)\, v_x(x,y) - v(x,y)\, u_x(x,y)}{[u(x,y)]^2 + [v(x,y)]^2}$$

and

$$\varphi(x,y) = \frac{u(x,y)\, v_y(x,y) - v(x,y)\, u_y(x,y)}{[u(x,y)]^2 + [v(x,y)]^2}\,.$$

Then N is obtained by the following relation:

$$N = h_1\big(I_1 - I_3\big) + h_2\big(I_2 - I_4\big)\,,$$

with

$$I_1 = \frac{1}{2\pi} \int_0^1 \psi(x_0 + t h_1, y_0)\, dt\,,$$

$$I_2 = \frac{1}{2\pi} \int_0^1 \varphi(x_0 + h_1, y_0 + t h_2)\, dt\,,$$

$$I_3 = \frac{1}{2\pi} \int_0^1 \psi(x_0 + th_1, y_0 + h_2)\, dt\,,$$

$$I_4 = \frac{1}{2\pi} \int_0^1 \varphi(x_0, y_0 + th_2)\, dt\,.$$

These formulae are the basis of the phases (a) and (b) of our method.

As f may have zeros on the boundary of the rectangular box specified by the user, the algorithm starts by perturbing this box. For this purpose a tolerance is introduced that is taken to be proportional to a power of the machine precision, for instance 10 times the square root of the machine precision. The box is then slightly enlarged asymmetrically. The reason for this asymmetric perturbation is to eliminate the possibility of having a zero close to or on any boundary of the consecutive subdivisions. For example, if the starting box is symmetric with respect to the imaginary axis, the inner boundary at the first subdivision will pass through any imaginary zeros of f.

The total number of zeros of f that lie inside the perturbed box is obtained by calculating the integrals $I_1, \ldots, I_4$ via the adaptive integrator DQAG from QUADPACK [25]. A zero near one of the edges of the rectangle causes the integrand of the corresponding integral to have a "peak." The closer the zero lies to the edge, the sharper is this peak. If the zero lies on the edge, then the integral is divergent. DQAG uses adaptive strategies that enable it to cope with such peaks efficiently. However, if a zero lies too close to an edge (the corresponding peak is too sharp), then DQAG warns us that it had problems in calculating the integral. Our algorithm then slightly moves this edge and restarts. By enlarging the user's box, we may of course include additional zeros. We have decided not to discard any of these zeros ourselves. Rather, we provide the user with the box that eventually has been considered, all the zeros that lie inside this box, and leave it to him/her to filter out unwanted zeros.

If the starting box (as perturbed by the method) contains a single zero, then this zero is calculated via the package CHABIS [26,27]. This package implements a generalized method of bisection, called *characteristic bisection*, which will be explained below. Otherwise, the longest edges of the box are halved, and the box is subdivided into two equal boxes. The number of zeros in each of these boxes is calculated via numerical integration. If DQAG detects a zero near the inner edge, then this edge is shifted, a process that results in an asymmetric subdivision of the box. Then the two smaller boxes are examined. A box that does not contain any zero is abandoned. A box that contains precisely one zero is handed to CHABIS. A box that contains more than one zero is subdivided again. This process is repeated until all the zeros have been isolated – a set of boxes has been found, each of which contains precisely one zero – and computed.

The method employed for the computation of the isolated zeros is based on the notion of the topological degree and is used to solve systems of nonlinear algebraic and/or transcendental equations [26,27]. First a so-called *characteristic polyhedron* (CP) is constructed, which, under certain assumptions on its boundary, secures the nonzero value of the topological degree without computing it. This nonzero value implies, by Kronecker's Theorem, the existence of at least one zero within this CP. The CP has the following property: the signs of the functions of the system at its vertices produce all the possible combinations of -1, 1. Based on this property, a smaller CP can be created by replacing a certain vertex of CP with the midpoint of its longest edge. The replaced vertex is the one where the signs of the function components coincide with those at the midpoint of the longest edge. The polyhedron thus obtained contains also the root since it remains characteristic. This process, which is called *characteristic bisection*, is repeated until the diameter of the box is smaller than a predetermined accuracy.

This method has the advantage that it always converges within the given region and it is a global convergence method. Also, the number of iterations required for the attainment of an approximate root to a predetermined accuracy is a priori known. Furthermore, since it depends only on the signs of the functions, it is suitable for

solving problems with imprecise function values or involving infinite series expansions. For example, in the case of Bessel and Airy functions it has been shown [28,29] that the sign stabilizes after summing a relatively small number of terms of the series and the calculations can be accelerated considerably. A theoretical approach to computing complex zeros of Bessel functions utilizing only the signs of the respective series is presented in [30].

As mentioned above, the derivatives of the Bessel functions may have multiple zeros at the points $z = \pm\nu$. Since our algorithm can manage only simple zeros, we have to examine the existence of such multiple zeros within the initial box in case we consider J'_ν, Y'_ν, $H_\nu^{(1)\prime}$ or $H_\nu^{(2)\prime}$. If the box contains $z = \nu$ or $z = -\nu$, the algorithm checks the values of the corresponding function and its derivative. If they are both equal to zero, the program terminates. Otherwise, the execution is continued.

3. A brief description of the packages used

In this section we briefly outline the packages BESSCC, QUADPACK and CHABIS incorporated in our code.

BESSCC: BESSCC [31] calculates the modified Bessel functions $I_\nu(z)$ and $K_\nu(z)$ as well as their first derivative for complex argument z and real order ν. To obtain J_ν from these functions we used the following analytic continuation and reflection formulae [3]: if $\nu \geq 0$, then

$$J_\nu(z) = \begin{cases} e^{i\frac{\pi}{2}\nu} I_\nu(-iz) & \text{if } \operatorname{Im} z \geq 0, \\ e^{-i\frac{\pi}{2}\nu} I_\nu(iz) & \text{if } \operatorname{Im} z < 0, \end{cases}$$

else

$$J_\nu(z) = J_{-\nu}(z)\cos(\pi\nu) + Y_{-\nu}(z)\sin(\pi\nu).$$

For Y_ν the following holds: if $\nu \geq 0$, then

$$Y_\nu(z) = \begin{cases} ie^{i\frac{\pi}{2}\nu} I_\nu(-iz) - \dfrac{2}{\pi} e^{-i\frac{\pi}{2}\nu} K_\nu(-iz) & \text{if } \operatorname{Im} z \geq 0, \\ \overline{Y_\nu(\bar{z})} & \text{if } \operatorname{Im} z < 0, \end{cases}$$

else

$$Y_\nu(z) = Y_{-\nu}(z)\cos(\pi\nu) - J_{-\nu}(z)\sin(\pi\nu).$$

The Bessel functions of the third kind were evaluated by using their definition. To obtain formulae for the derivatives, we differentiated the previous identities.

For compatibility purposes, and with the permission of the authors, we also made some small changes to BESSCC.

QUADPACK: QUADPACK [25] is a widely used package for automatic integration. It consists of 12 quadrature routines. Here the routine DQAG is used, which implements a globally adaptive integrator for calculating definite integrals over finite intervals. DQAG uses double precision arithmetic and is based on Gauss–Kronrod quadrature rules. It is written in ANSI standard FORTRAN-77.

CHABIS: CHABIS [27] is a mathematical software package for the numerical solution of a system of n nonlinear equations in n variables when the only computable information is the algebraic signs of the components of the function. First CHABIS locates at least one solution of the system within an n-dimensional polyhedron. Then, it applies a new generalized method of bisection to this n-polyhedron in order to obtain an approximate solution of the system according to a predetermined accuracy. CHABIS consists of a set of nine subprograms, one of which is called by the user-driver program. The package CHABIS contains about 1 200 lines of code, 50 percent of which are comments. The total storage required for CHABIS is $6n + (6n + 1)2^n$ locations, where n determines the dimension of the problem. CHABIS is coded in ANSI standard FORTRAN-77. More details can be found in [26,27].

4. Program description

The package ZEBEC (ZEros of BEssel functions Complex) contains about 7 000 lines of code including comments. It is written in FORTRAN-77 and has been tested on various UNIX machines as well as on a PC IBM compatible.

ZEBEC consists of seven parts, namely, the main program ZEBEC, the subroutines MANAGE, INBOX, SPLIT, RCOMP, and FDF and the function FNC, and a set of functions related to the integrals $I_1, \ldots, I_4$. ZEBEC also requires the subroutine DQAG from the package QUADPACK [25], the package CHABIS [27] and the subroutine BESSCC [31] of the CPC Program Library.

In the main program ZEBEC the following parameters have to be set:

MAXRT a positive integer that determines the maximum number of zeros that may be requested.

ICASE an integer in $\{1, \ldots, 8\}$ that specifies which function is to be considered:
- 1 J, the Bessel function of the first kind;
- 2 the derivative of J;
- 3 Y, the Bessel function of the second kind;
- 4 the derivative of Y;
- 5 $H^{(1)} = J + iY$, the Bessel function of the third kind;
- 6 the derivative of $H^{(1)}$;
- 7 $H^{(2)} = J - iY$, the Bessel function of the third kind;
- 8 the derivative of $H^{(2)}$.

XNU a real variable that specifies the order of the Bessel function.

XO a real array of length 2 that contains the x- and y-coordinates of the left lower vertex of the rectangle that is to be examined (see also Section 2).

H a real array of length 2 that specifies the size of this rectangle along the x- and y-direction.

ICON an integer in $\{1, \ldots, 4\}$ that specifies which calculations are to be done:
- 1 calculation of the total number of zeros, only;
- 2 calculation of the total number of zeros and isolation of each one of them;
- 3 calculation of the total number of zeros, isolation and computation of each one of them;
- 4 calculation of the total number of zeros; isolation and computation of NR zeros.

Note that if ICON=4 the user must also supply the desired number of zeros NR. In the other cases (ICON=1,2,3) a value of NR may be supplied but it will not be used by the package.

EPSILO a real variable that is used in the stopping criterion for the computation of the zeros. Termination occurs if the algorithm estimates that the infinity norm of the function value at an approximate solution is at most EPSILO or if the size of the box containing a zero is at most 4.D0*EPSILO. If EPSILO is set to a value that is less than the machine precision EPSMCH, then EPSILO is set equal to 5.D0*EPSMCH. EPSMCH is computed within ZEBEC. The value of EPSILO needs to be set only in case ICON=3 or ICON=4.

EPSABS a real variable that determines the absolute accuracy to which the integrals are to be evaluated. If EPSABS=0.D0, then only a relative precision criterion will be used.

EPSREL a real variable that determines the relative accuracy to which the integrals are to be evaluated. If EPSREL=0.D0, then only an absolute precision criterion will be used.

If EPSABS and EPSREL are both too small, then the numerical integration may be time consuming. If they are both too large, then the calculated number of zeros may be wrong. The default values of EPSABS and EPSREL are 0.07D0 and 0.0D0, respectively.

ACC a variable whose value determines a target relative accuracy for the subroutine BESSCC which is used for the calculation of the Bessel functions. If ACC is greater than 0.0001 or less than the machine accuracy, then it is set equal to the default value ACCDEF=10E-6.

Let us briefly describe the various parts of ZEBEC.

The subroutine INBOX calculates the total number of zeros that lie inside the box specified by the user. INBOX slightly perturbs this box and enlarges it, if necessary, namely when zeros lie too close to the boundary and the quadrature routine DQAG fails.

The subroutine SPLIT takes a box and splits it into two boxes. A symmetric splitting, by halving the longest edges, is tried first. If it is necessary for DQAG, the inner edge will be shifted.

The subroutine RCOMP takes a box containing precisely one zero and returns an approximation to this zero. RCOMP calls CHABIS.

The subroutine MANAGE forms the main part of the package. MANAGE starts by calling INBOX. If there are no zeros inside the user's box, then the program stops. If there is precisely one zero inside this box, then RCOMP is called (if ICON=3 or 4). Else, the box is given to SPLIT. The two boxes returned by SPLIT are examined. A box that does not contain any zero is abandoned. A box that contains precisely one zero is given to RCOMP (if ICON=3 or 4). A box that contains more than one zero is put in a list. Then MANAGE takes the next box from this list and evokes SPLIT. This procedure is repeated until all the zeros are isolated and computed, if it is required.

DQAG needs function and derivative values of the Bessel function that is to be considered. These are provided by the subroutine FDF. CHABIS needs only the function values. These are provided by the function FNC. Both FDF and FNC call BESSCC.

The program execution terminates normally after the completion of its task. This type of termination is indicated by the value 1 of the output variable INFO. If the value of this parameter is different from 1, the termination of the program is abnormal. The cases of abnormal termination are the following:

INFO=0 Improper input parameters:
the input values of ICASE or ICON are out of range, or
NR exceeds MAXRT, or
the box specified by XO and H crosses the nonpositive real axis, or
the initial box contains multiple zeros (ICASE=2, 4, 6, or 8), or
H(1) or H(2) is negative, or
EPSABS or EPSREL is negative.

INFO=2 The procedure for the calculation of the total number of zeros failed.

INFO=3 The procedure for the isolation of the zeros failed.

INFO=4 The procedure for the computation of the zeros failed.

Upon normal termination, the main output values of the program are given by the following parameters of MANAGE:

NRPERT an integer that gives the number of zeros existing in the examined box when ICON is not equal to 3.

LFLRT an integer that gives the number of zeros existing in the examined box if ICON is equal to 3.

XOFIN a real array of size 2×MAXRT that contains the x- and y-coordinates of the left lower vertices of the

rectangles that are found to contain a single zero.

HFIN a real array of size 2×MAXRT that specifies the size of these rectangles along the x- and y-direction.

ROOTS a real array of size 2×MAXRT that contains the real and imaginary parts of the zeros found in the examined box.

FROOTS a real array of size 2×MAXRT that contains the values of the Bessel function (real and imaginary parts) at the approximations of the roots found.

5. Example of ZEBEC usage

Let us demonstrate how ZEBEC can be used to calculate the total number of zeros inside a given box, to isolate these zeros, and to compute all of them. Suppose that we want to calculate all the zeros of $J_{-1.4}(z)$ that lie inside the box

$$\{z \in \mathbb{C} : -1 \leq \mathrm{Re}\, z \leq 2,\ 0.5 \leq \mathrm{Im}\, z \leq 4\}.$$

The corresponding input values are ICASE=1, XNU=-1.4D0, ICON=3, NR=0, and XO(1)=-1.D0, XO(2)=0.5D0, H(1)=3.D0, H(2)=3.5D0. We request an accuracy of EPSILO=1.0D-13.

For this example the main program of ZEBEC is the following:

```
*---------------------------------------------------------------------*
      PROGRAM ZEBEC
      IMPLICIT NONE

      INTEGER MAXRT
      PARAMETER (MAXRT = 100)

      INTEGER IFLRT(MAXRT), ICASE, ICON, NR, INFO, NRKEEP, IBESS
      INTEGER J, LFLRT, NRPERT, IMAXRT

      DOUBLE PRECISION XO(2), H(2), XOPERT(2), HPERT(2)
      DOUBLE PRECISION ROOTS(2,MAXRT), FROOTS(2,MAXRT)
      DOUBLE PRECISION POINTS(2,MAXRT), STEPS(2,MAXRT), NRS(MAXRT)
      DOUBLE PRECISION XOFIN(2,MAXRT), HFIN(2,MAXRT)
      DOUBLE PRECISION RINTS(6,MAXRT), ERRS(6,MAXRT)
      DOUBLE PRECISION EPSILO, XNU, EPSABS, EPSREL, ACC

      EXTERNAL MANAGE

      COMMON /BLK1/ ICON, NR
      COMMON /BLK2/ INFO
      COMMON /BLK3/ XNU
      COMMON /BLK4/ ICASE
      COMMON /BLK5/ IBESS
      COMMON /BLK6/ EPSABS, EPSREL
      COMMON /BLK7/ ACC
      COMMON /BLK8/ IMAXRT

      DATA   ICASE,    XNU,      ICON,    NR
     +     / 1,        -1.4D0,   3,       0 /
      DATA   XO(1),    XO(2),    H(1),    H(2)
     +     / -1.D0,    0.5D0,    3.D0,    3.5D0 /

      EPSILO = 1.0D-13
```

```
      EPSABS = 0.07D0
      EPSREL = 0.0D0

      ACC = 1.D-13

      PRINT 9999, ICASE, XNU, XO, H, ICON, EPSILO

      NRKEEP = NR

      IBESS = 0
      IMAXRT = 0
*
*     Call the interface subroutine MANAGE.
*
      CALL MANAGE(MAXRT,XO,H,XOPERT,HPERT,EPSILO,NRPERT,XOFIN,HFIN,
     +            ROOTS,FROOTS,POINTS,STEPS,RINTS,ERRS,NRS,IFLRT,LFLRT)

      IF ( INFO .EQ. 0 ) THEN
        PRINT 9998
        GO TO 10
      END IF

      IF ( IBESS .EQ. 2 ) PRINT 9997

      IF ( INFO .EQ. 2 ) THEN
        PRINT 9996
        GO TO 10
      END IF

      IF ( IMAXRT .EQ. 1 ) THEN
        PRINT 9995, MAXRT
        STOP
      END IF

      PRINT 9994, XOPERT(1), XOPERT(2), HPERT(1), HPERT(2)

      IF ( ICON .NE. 3 ) THEN
        PRINT 9993, NRPERT
      ELSE
        PRINT 9993, LFLRT
      END IF

      IF ( NRPERT .EQ. 0 ) GO TO 10
      IF ( ICON .EQ. 1 ) GO TO 10

      IF ( INFO .EQ. 3 ) THEN
        PRINT 9992
        GO TO 10
      ENDIF

      IF ( ICON .EQ. 4 ) THEN
        PRINT 9991, NRKEEP, NR
      ELSE
        PRINT 9990, LFLRT
      END IF

      PRINT 9989
```

```
      DO J = 1,LFLRT
        PRINT 9988, J, XOFIN(1,J), XOFIN(2,J), HFIN(1,J), HFIN(2,J)
      ENDDO

      IF ( ICON .EQ. 2 ) GO TO 10

      IF ( INFO .EQ. 4 ) THEN
        PRINT 9987
        GO TO 10
      ENDIF

      PRINT 9986
      DO J = 1,LFLRT
        IF ( IFLRT(J) .EQ. 1 ) THEN
          PRINT 9985, J, ROOTS(1,J), ROOTS(2,J),
     +                   FROOTS(1,J), FROOTS(2,J)
        ELSE
          PRINT 9984, J
        END IF
      ENDDO

   10 PRINT 9983, INFO
      STOP

 9999 FORMAT (/2X, ' STARTING VALUES :' /3X, 17('-'),
     +       //2X, ' ICASE  :    ', I1,
     +        /2X, ' ORDER  :   ', F22.15,
     +       //2X, ' XO     :   ', F22.15, F23.15,
     +        /2X, ' H      :   ', F22.15, F23.15,
     +       //2X, ' ICON   :    ', I1,
     +        /2X, ' EPSILO :   ', F22.15,
     +       //3X, 67('-'),
     +       //2X, ' RESULTS :' /3X, 9('-'))

 9998 FORMAT (/2X, '  * * * IMPROPER INPUT PARAMETERS * * *'//)

 9997 FORMAT (/2X, ' * THE PROCEDURE FOR THE CALCULATION OF THE',
     +             ' BESSEL',
     +        /2X, '   FUNCTION FAILED. THE RESULTS MAY BE',
     +             ' INACCURATE *')

 9996 FORMAT (/2X, '  * * * THE PROCEDURE FOR THE CALCULATION OF',
     +        /2X, '        THE TOTAL NUMBER OF ZEROS FAILED * * *'//)

 9995 FORMAT (/2X, ' * * * THE NUMBER OF ZEROS EXCEEDS MAXRT = ',I5,
     +        /2X, '       INCREASE THE VALUE OF MAXRT * * *'//)

 9994 FORMAT (/2X, ' THE FOLLOWING BOX WAS CONSIDERED:',
     +       //5X, 'XO = ', F22.15, F24.15,
     +        /5X, 'H  = ', F22.15, F24.15 )

 9993 FORMAT (/2X, ' THE TOTAL NUMBER OF ZEROS WITHIN',
     +             ' THIS BOX IS : ',I5)

 9992 FORMAT (/2X, ' * * * THE PROCEDURE FOR THE ISOLATION OF',
     +        /2X, '       THE ZEROS FAILED * * *'//)
```

```
 9991 FORMAT (/2X, ' NUMBER OF ZEROS REQUESTED  : ',I5,/
     +        /2X, ' NUMBER OF ZEROS ISOLATED   : ',I5)

 9990 FORMAT (/2X, ' NUMBER OF ZEROS ISOLATED   : ',I5)

 9989 FORMAT (/2X, ' BOXES CONTAINING A SINGLE ZERO :',
     +        /3X, 32('-'))

 9988 FORMAT ( /2X, I4,')  XO = ', F22.15, F24.15,
     +              /9X, 'H  = ', F22.15, F24.15 )

 9987 FORMAT (/2X, ' * * * THE PROCEDURE FOR THE COMPUTATION OF',
     +        /2X, '       THE ZEROS FAILED * * *'//)

 9986 FORMAT (/2X, ' FINAL APPROXIMATE ZEROS AND',
     +             ' VERIFICATION :',
     +        /3X, 42('-') )

 9985 FORMAT ( /2X, I4,')  Z    = (', F22.15, ',', F22.15, ' )',
     +              /9X, 'F(Z) = (', F22.15, ',', F22.15, ' )' )

 9984 FORMAT ( /2X, I4,')'/)

 9983 FORMAT (/2X, ' EXIT PARAMETER :   INFO = ',I2)
*
*     Last statement of the main program.
*
      END
*-----------------------------------------------------------------------*
```

The output results of ZEBEC for four test cases are shown in the test run outputs below.

6. Concluding remarks

The package ZEBEC has been applied to Bessel functions of various orders and random boxes. We have found that it behaves predictably and accurately. It calculates with certainty the total number of zeros that lie inside a given box, isolates each one of them, and then computes all these zeros.

The user will appreciate the flexibility offered by the input parameter ICON. If nothing is known about the zeros that lie inside the given box, one may call ZEBEC with ICON = 1 to obtain the total number of zeros. Then one may proceed with ICON = 3 to isolate and compute all these zeros, or, if less than the total number of zeros are required, with ICON = 4 and NR equal to the requested number of zeros. If only a set of boxes is required, each of which contains precisely one zero, then one may set ICON = 2.

Our package can be applied to any special function that has only simple zeros in the considered box, provided a FORTRAN-77 routine exists to evaluate this function and its first derivative.

Acknowledgements

The authors wish to thank Profs. A.R. Barnett and I.J. Thompson, as well as Prof. R. Piessens for most kindly permitting the usage of the packages BESSCC and QUADPACK, respectively, as well as Prof. R. Cools for useful discussions.

One of the authors (P. K.) would like to thank the Department of Mathematics, University of Patras, for its hospitality, during which time the main part of this work was carried out, and Vlaamse Leergangen Leuven for supporting his visit to Patras. He also wishes to acknowledge the support granted to him by the Flemish Institute for the Promotion of Scientific and Technological Research in the Industry (IWT).

References

[1] M.N. Vrahatis, O. Ragos, T. Skiniotis, F.A. Zafiropoulos, T.N. Grapsa, Comput. Phys. Commun. 92 (1995) 252.
[2] G.N. Watson, A Treatise on the Theory of Bessel Functions, 2nd ed. (Cambridge University Press, Cambridge, 1966).
[3] M. Abramowitz, I.A. Stegun, Handbook of Mathematical Functions: with formulas, graphs, mathematical tables, Dover Books on Intermediate and Advanced Mathematics, 7th ed. (Dover, New York, 1970).
[4] L.G. Chambers, Math. Comput. 38 (1982) 589.
[5] E.K. Ifantis, C.G. Kokologiannaki, Z. Anal. Anwend. 12 (1993) 605.
[6] E.K. Ifantis, P.D. Siafarikas, Appl. Anal. 23 (1986) 85.
[7] E.K. Ifantis, P.D. Siafarikas, J. Comput. Appl. Math. 44 (1992) 115.
[8] C.G. Kokologiannaki, P.D. Siafarikas, Z. Anal. Anwend. 10 (1991) 563.
[9] C.G. Kokologiannaki, P.D. Siafarikas, C.B. Kouris, J. Comput. Appl. Math. 40 (1992) 337.
[10] N.N. Lebedev, Special Functions and their Applications (Dover, New York, 1972).
[11] R. Piessens, J. Comput. Phys. 42 (1981).
[12] R. Piessens, J. Comput. Phys. 53 (1984).
[13] R. Piessens, Math. Comput. 42 (1984) 195.
[14] R. Piessens, J. Comput. Phys. 64 (1986).
[15] A. Gray, G.B. Mathews, A Treatise on Bessel Functions and Their Applications to Physics (Dover, New York, 1966).
[16] M.K. Kerimov, S.L. Skorokhodov, U.S.S.R. Comput. Maths. Math. Phys. 25 (1985) 101.
[17] M.K. Kerimov, S.L. Skorokhodov, Sov. Phys. Dokl. 33 (1988) 196.
[18] N.M. Temme, Special Functions: An Introduction to the Classical Functions of Mathematical Physics (Wiley, New York, 1996).
[19] R. Piessens, Bull. Greek Math. Soc. 31 (1990) 117.
[20] R. Piessens, G. Engelen, The computation of zeros and turning points of the Bessel functions of the first kind, Report TW 72, K.U. Leuven, Dept. Computer Science, Celestijnenlaan 200 A, B-3001 Heverlee, Belgium (1985).
[21] P. Henrici, Applied and Computational Complex Analysis: I. Power Series–Integration–Conformal Mapping–Location of Zeros (Wiley, New York, 1974).
[22] N.G. Lloyd, Degree Theory, Cambridge Tracts in Mathematics, Vol. 73 (Cambridge Univ. Press, Cambridge, 1978).
[23] B.J. Hoenders, C.H. Slump, Computing 30 (1983) 137.
[24] B.J. Hoenders, C.H. Slump, Computing 47 (1992) 323.
[25] R. Piessens, E. de Doncker-Kapenga, C.W. Überhuber, D.K. Kahaner, QUADPACK: A Subroutine Package for Automatic Integration, Springer Series in Computational Mathematics, Vol. 1 (Springer, Berlin, 1983).
[26] M.N. Vrahatis, ACM Trans. Math. Softw. 14 (1988) 312.
[27] M.N. Vrahatis, ACM Trans. Math. Softw. 14 (1988) 330.
[28] M.N. Vrahatis, T.N. Grapsa, O. Ragos, F.A. Zafiropoulos, Z. Angew. Math. Mech. 77 (1997) 467.
[29] M.N. Vrahatis, O. Ragos, F.A. Zafiropoulos, T.N. Grapsa, Z. Angew. Math. Mech. 76 (1996) 419.
[30] M.N. Vrahatis, O. Ragos, T. Skiniotis, F.A. Zafiropoulos, T.N. Grapsa, Num. Func. Anal. & Optimiz. 18 (1997) 227.
[31] I.J. Thompson, A.R. Barnett, Comput. Phys. Commun. 47 (1987) 245.

TEST RUN OUTPUT 1

```
STARTING VALUES :
-----------------

ICASE  :    1
ORDER  :       -1.400000000000000

XO     :       -1.000000000000000        .500000000000000
H      :        3.000000000000000       3.500000000000000

ICON   :    3
EPSILO :          .000000000000100

-------------------------------------------------------------------

RESULTS :
---------

THE FOLLOWING BOX WAS CONSIDERED:

  XO =      -1.000000163912773         .499999806284904
  H  =       3.000000357627869        3.500000417232513

THE TOTAL NUMBER OF ZEROS WITHIN THIS BOX IS :     1

NUMBER OF ZEROS ISOLATED   :     1

BOXES CONTAINING A SINGLE ZERO :
--------------------------------

  1)  XO =      -1.000000163912773         .499999806284904
      H  =       3.000000357627869        3.500000417232513

FINAL APPROXIMATE ZEROS AND VERIFICATION :
------------------------------------------

  1)  Z    = (      -.000000000000055,      1.118783284992136 )
      F(Z) = (       .000000000000080,      -.000000000000015 )

EXIT PARAMETER :   INFO =  1
```

TEST RUN OUTPUT 2

```
STARTING VALUES :
-----------------

ICASE  :    3
ORDER  :      -15.300000000000001

XO     :      -22.000000000000002        .500000000000000
H      :       45.000000000000000     100.000000000000000

ICON   :    4
EPSILO :          .000000000000100

------------------------------------------------------------------

RESULTS :
---------

THE FOLLOWING BOX WAS CONSIDERED:

  XO =     -22.000000163912774         .499999806284904
  H  =      45.000000357627865      100.000000417232520

THE TOTAL NUMBER OF ZEROS WITHIN THIS BOX IS :    16

NUMBER OF ZEROS REQUESTED  :     5

NUMBER OF ZEROS ISOLATED   :     5

BOXES CONTAINING A SINGLE ZERO :
--------------------------------

  1)  XO =      11.750000104308129         .499999806284904
      H  =      11.250000089406966       12.500000052154063

  2)  XO =        .500000014901161         .499999806284904
      H  =      11.250000089406966        6.250000026077032

  3)  XO =       6.125000059604645        6.749999832361937
      H  =       2.812500022351742        1.562500006519258

  4)  XO =       6.125000059604645        8.312499838881195
      H  =       2.812500022351742        1.562500006519258

  5)  XO =        .500000014901161        9.874999845400453
      H  =       5.625000044703484        3.125000013038516

FINAL APPROXIMATE ZEROS AND VERIFICATION :
------------------------------------------

  1)  Z    = (     12.507257919321071,      4.095557539693683 )
      F(Z) = (      -.000000000000003,      -.000000000000039 )

  2)  Z    = (     10.378711252301940,      6.178243183678395 )
      F(Z) = (      -.000000000000009,       .000000000000031 )
```

```
  3)  Z    = (     8.447945724224951,     7.613850557712050 )
      F(Z) = (      .000000000000075,      .000000000000033 )

  4)  Z    = (     6.607246778783210,     8.648294108469159 )
      F(Z) = (      .000000000000034,     -.000000000000027 )

  5)  Z    = (     1.305877373221013,    10.127220235489998 )
      F(Z) = (      .000000000000057,     -.000000000000045 )

EXIT PARAMETER :   INFO =  1
```

TEST RUN OUTPUT 3

```
STARTING VALUES :
-----------------

ICASE  :    4
ORDER  :        -.100000000000000

XO     :      -22.000000000000002        .500000000000000
H      :       45.000000000000000     100.000000000000000

ICON   :    3
EPSILO :         .000000000000100

-------------------------------------------------------------------

RESULTS :
---------

THE FOLLOWING BOX WAS CONSIDERED:

  XO =    -22.000000163912774         .499999806284904
  H  =     45.000000357627865      100.000000417232520

THE TOTAL NUMBER OF ZEROS WITHIN THIS BOX IS :     7

NUMBER OF ZEROS ISOLATED   :     7

BOXES CONTAINING A SINGLE ZERO :
--------------------------------

  1)  XO =     -5.125000029802322         .499999806284904
      H  =      2.812500022351742        3.125000013038516

  2)  XO =     -2.312500007450581         .499999806284904
      H  =      2.812500022351742        3.125000013038516

  3)  XO =    -10.750000074505806         .499999806284904
      H  =      2.812500022351742        3.125000013038516

  4)  XO =     -7.937500052154064         .499999806284904
      H  =      2.812500022351742        3.125000013038516

  5)  XO =    -16.375000119209289         .499999806284904
      H  =      5.625000044703484        6.250000026077032

  6)  XO =    -22.000000163912774         .499999806284904
      H  =      2.812500022351742        3.125000013038516

  7)  XO =    -19.187500141561031         .499999806284904
      H  =      2.812500022351742        3.125000013038516

FINAL APPROXIMATE ZEROS AND VERIFICATION :
------------------------------------------

  1)  Z    = (    -3.887205157313519,       .539553358417456 )
      F(Z) = (      .000000000000067,      -.000000000000002 )
```

```
  2)  Z    = (     -.548875794670892,      .753840972468368 )
      F(Z) = (     -.000000000000034,     -.000000000000012 )

  3)  Z    = (   -10.227206121588486,      .529025446346720 )
      F(Z) = (     -.000000000000043,     -.000000000000012 )

  4)  Z    = (    -7.069495108154051,      .531052479620184 )
      F(Z) = (     -.000000000000077,     -.000000000000036 )

  5)  Z    = (   -13.377415057395032,      .528243018784274 )
      F(Z) = (      .000000000000068,     -.000000000000047 )

  6)  Z    = (   -19.669612573645452,      .527646968933275 )
      F(Z) = (      .000000000000002,     -.000000000000057 )

  7)  Z    = (   -16.524366351420319,      .527861289720225 )
      F(Z) = (      .000000000000047,     -.000000000000020 )

EXIT PARAMETER :   INFO =  1
```

TEST RUN OUTPUT 4

```
STARTING VALUES :
-----------------

ICASE  :    5
ORDER  :         3.000000000000000

XO     :      -10.000000000000000     -10.000000000000000
H      :       20.000000000000000       9.500000000000000

ICON   :    3
EPSILO :          .000000000000100

------------------------------------------------------------------

RESULTS :
---------

THE FOLLOWING BOX WAS CONSIDERED:

  XO =     -10.000000163912772      -10.000000193715095
  H  =      20.000000357627870        9.500000417232513

THE TOTAL NUMBER OF ZEROS WITHIN THIS BOX IS :      3

NUMBER OF ZEROS ISOLATED   :     3

BOXES CONTAINING A SINGLE ZERO :
--------------------------------

  1)  XO =        .000000014901161      -10.000000193715095
      H  =      10.000000178813935        9.500000417232513

  2)  XO =      -2.500000029802322       -2.874999880790710
      H  =       1.250000022351742        2.375000104308128

  3)  XO =      -1.250000007450581       -2.874999880790710
      H  =       1.250000022351742        2.375000104308128

FINAL APPROXIMATE ZEROS AND VERIFICATION :
------------------------------------------

  1)  Z    = (      1.308012032273956,     -1.681788804745902 )
      F(Z) = (      -.000000000000073,      -.000000000000027 )

  2)  Z    = (     -2.242469255140806,     -1.006482383164907 )
      F(Z) = (      -.000000000000067,      -.000000000000098 )

  3)  Z    = (      -.431821001058069,     -1.958584527573394 )
      F(Z) = (      -.000000000000073,       .000000000000017 )
```