Author(s) : I.J. Thompson, A.R. Barnett
Submitter : K.S. Kölbig
Language : Fortran

Library: MATHLIB
Submitted: 15.01.1988
Revised: 15.11.1995

Coulomb Wave, Bessel, and Spherical Bessel Functions for Complex Argument(s) and Order

Subroutine subprograms `CCLBES` and `WCLBES` calculate any one of the following sequences of functions:

1. Regular and irregular Coulomb wave functions $F_{\lambda+n}(\eta,z),\ G_{\lambda+n}(\eta,z)$
   and their first derivatives with respect to z, $F'_{\lambda+n}(\eta,z),\ G'_{\lambda+n}(\eta,z)$,
   or simple combination of these;
2. Spherical Bessel functions $j_{\lambda+n}(z),\ y_{\lambda+n}(z)$
   and their first derivatives with respect to z, $j'_{\lambda+n}(z),\ y'_{\lambda+n}(z)$,
   or simple combination of these (spherical Hankel functions);
3. Bessel functions $J_{\lambda+n}(z),\ Y_{\lambda+n}(z)$
   and their first derivatives with respect to z, $J'_{\lambda+n}(z),\ Y'_{\lambda+n}(z)$,
   or simple combination of these (Hankel functions);
4. Modified Bessel functions $I_{\lambda+n}(z),\ K_{\lambda+n}(z)$
   and their first derivatives with respect to z, $I'_{\lambda+n}(z),\ K'_{\lambda+n}(z)$;

for complex arguments η, z, complex order λ, and $n = 0, 1, \ldots, N$.

The double-precision version `WCLBES` is available only on computers which support a `COMPLEX*16` Fortran data type.

Structure:

`SUBROUTINE` subprograms
User Entry Names: `CCLBES`, `WCLBES`
Internal Entry Names: `C309R1`, `C309R2`, `C309R3`, `C309R4`, `C309R5`, `C309R6`, `C309R7`, `C309R8`
Files Referenced: `Unit 6`
External References: `CLGAMA` (C306), `WLGAMA` (C306), `CPSIPG` (C317), `WPSIPG` (C317)

Usage:

For `t` = `C` (type `COMPLEX`), `t` = `W` (type `COMPLEX*16`),

```
CALL tCLBES(Z,ETA,ZLMIN,NL,F,G,FP,GP,SIG,KFN,MODE,JFAIL,JPR)
```

`Z`	(type according to `t`) Argument $z \neq 0$.
`ETA`	(type according to `t`) Argument η (ignored if `KFN` > 0).
`ZLMIN`	(type according to `t`) Order λ_{min} of the first function in the computed sequence.
`NL`	(`INTEGER`) Specifies the order λ_{min} + `NL` of the last function in the computed sequence. (`NL` ≥ 0).
`F,G,FP,GP`	(type according to `t`) One-dimensional arrays with dimension `(0:d)` where `d` is in each case $\geq$ `NL` + 1. On exit, each of `F(n),G(n),FP(n),GP(n)` may contain the value of a function of order $\lambda_{min} + n$, or its first order derivative, ($n = 0, 1, \ldots,$ `NL`), as specified jointly by `KFN` and `|MODE|`.

SIG (type according to t) One-dimensional array with dimension (0:d), where $d \geq NL + 1$. On exit, provided KFN $= 0$, SIG(n) contains the Coulomb phase shift $\sigma(\eta)$ for $\lambda = \lambda_{min} + n$, $(n = 0, 1, \ldots, NL)$.

KFN (INTEGER) Specifies, in conjunction with the absolute value of MODE, the type of functions which are stored.

MODE (INTEGER) The absolute value of MODE specifies, in conjunction with KFN, the type of function which are stored, and also specifies which of the arrays F,G,FP,GP are in fact set to meaningful values. The sign of MODE specifies whether or not the functions are multiplied by a scaling factor.

JFAIL (INTEGER) On exit, JFAIL is set to zero if no error condition is detected. Otherwise JFAIL is set as described under **Error handling**.

JPR (INTEGER)

$= 0$: Suppress printing of error messages.

$= 1$: Print error messages.

The type of function which is stored in array F depends only on KFN, while the type of function which is stored in array G depends both on KFN and on |MODE|. Arrays FP and GP (if set) contain the first order derivatives with respect to z of the functions in F and G, respectively. Using the abbreviations ($i = \sqrt{-1}$)

$$\begin{array}{lll} F_\lambda \equiv F_\lambda(\eta, z), & G_\lambda \equiv G_\lambda(\eta, z), & H_\lambda^{\pm} \equiv G_\lambda \pm iF_\lambda, \\ j_\lambda \equiv j_\lambda(z), & y_\lambda \equiv y_\lambda(z), & h_\lambda^{(1,2)} \equiv j_\lambda \pm iy_\lambda, \\ J_\lambda \equiv J_\lambda(z), & Y_\lambda \equiv Y_\lambda(z), & H_\lambda^{(1,2)} \equiv J_\lambda \pm iY_\lambda, \\ I_\lambda \equiv I_\lambda(z), & K_\lambda \equiv K_\lambda(z), & \end{array}$$

the choice of function is given by the following table:

Array	|MODE|	KFN			
		-1 or 0	1	2	3
F	all values	F_λ	j_λ	J_λ	I_λ
G	$1, 2, 3, 4$	G_λ	y_λ	Y_λ	K_λ
	$11, 12$	H_λ^+	$h_\lambda^{(1)}$	$H_\lambda^{(1)}$	$-$
	$21, 22$	H_λ^-	$h_\lambda^{(2)}$	$H_\lambda^{(2)}$	$-$

If KFN=0 the phase shifts $\sigma(\eta)$ are stored in array SIG. Otherwise SIG is not set.

Which of arrays F,G,FP,GP are in fact set is determined by |MODE| according to the following table:

|MODE|	F	G	FP	GP
1, 11, 21	set	set	set	set
2, 12, 22	set	set	-	-
3	set	-	set	-
4	set	-	-	-

In both the tables above, a dash indicates that the corresponding array does not contain meaningful values on exit. These arrays are, however, used internally as working space, and must therefore be dimensioned correctly. The sign of MODE specifies whether or not the functions are to be multiplied by a scaling factor, depending only on z, which will bring their values closer to unity when $|z|$ is large, or η is small and $|\lambda| < |z|$. The same scaling factor is applied to the first order derivatives in FP or GP as is applied to the functions in F or G, respectively.

MODE > 0 : No scaling factor.

MODE < 0 : Let $S = \mathrm{Im}(z)$ if KFN < 3, $S = \mathrm{Re}(z)$ if KFN $= 3$; then the scaling factors for F and G are

$$\begin{array}{lll} \mathtt{F}: & \exp(-|S|) & \{F, j, J, I\} \\ \mathtt{G}: & \exp(-|S|) & \{G, y, Y\} \\ & \exp(S) & \{H^+, h^{(1)}, H^{(1)}, K\} \\ & \exp(-S) & \{H^-, h^{(2)}, H^{(2)}\}. \end{array}$$

Method:

The method is described in the References.

Restrictions:

See Ref. 1, in particular Sect. 4.

Accuracy:

The absolute values of the results are usually accurate to within two or three decimal digits of the machine precision. For details of exceptions see Ref. 1, Sect. 4.

Error handling:

If an error condition is detected, JFAIL is set to one of the following values and a message is printed if JPR $= 1$.

- > 0 An arithmetic error occurred during the final recursion. Correct results are available up to and including subscript value NL-JFAIL-1.
- -1 One of the continued fraction calculations failed or there was an arithmetic error before any results could be calculated.
- -2 Argument out of range.
- -3 One or more functions corresponding to λ_{min} could not be calculated. Some values corresponding to $\lambda > \lambda_{min}$ may be correct.
- -4 Excessive internal cancellation probably renders the result meaningless.

Source:

This program package is a modified version of the CPC Program Library package COULCC (see Ref. 1). The changes are formal, not computational.

References:

1. I.J. Thompson and A.R. Barnett, COULCC: A continued-fraction algorithm for Coulomb functions of complex order with complex arguments, Comput. Phys. Comm. **36** (1985) 363–372.
2. I.J. Thompson and A.R. Barnett, Coulomb and Bessel functions of complex arguments and order, J. Comput. Phys. **64** (1986) 490–509.

Long Write-up:

A copy of Ref. 1 is available in the Program Library Office.

•