Computer Physics Communications 27 (1982) 147–166
North-Holland Publishing Company

COULFG: COULOMB AND BESSEL FUNCTIONS AND THEIR DERIVATIVES, FOR REAL ARGUMENTS, BY STEED'S METHOD

A.R. BARNETT

Department of Physics, Schuster Laboratory, The University, Manchester, M13 9PL, UK

Received 15 December 1981

PROGRAM SUMMARY

Title of program: COULFG: Coulomb, Bessel Functions

Catalogue number: ABNK

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

Computer: IBM 370/165 and AS/7000; *Installation*: Daresbury Laboratory, Warrington, Lancs.

Operating system: OS/360 GI compiler and HX compiler (level 2.2.1)

Programming language used: ASA FORTRAN

High speed storage required: 180 Kbytes

No. of bits in a word: 32

Overlay structure: none

Peripherals used: card reader, printer

No. of cards in combined program and test deck: 432

Card punching code: EBCD1C

Keywords: Klein–Gordon, Coulomb for real angular momentum, recurrence relations, Schrödinger, Bessel, spherical Bessel, continued fraction, reactions, scattering, heavy ion, nuclear, molecular, atomic, pionic, kaonic, exotic atoms, scattering states

Nature of physical problem
Coulomb interaction of charged particles in spherical coordinates (Coulomb function) and uncharged particles (spherical Bessel functions), and general problems in cylindrical coordinates yielding Bessel-function solutions can be solved with the program. COULFG computes the Coulomb wavefunctions $F_\lambda(\eta, x)$, $G_\lambda(\eta, x)$, $F'_\lambda(\eta, x)$ and $G'_\lambda(\eta, x)$ for a range of λ-values in integer steps, $l \geqslant \lambda \geqslant m > -1$, for real $x > 0$ and real η, $10^4 \geqslant |\eta|$. Values of the maximum angular momentum, l, of several thousand can be treated. When the functions have oscillating character the absolute accuracies are about 10^{-14} but this is only limited by the word length used; extended precision variables can yield absolute accuracies of 10^{-30}. COULFG will generate spherical Bessel functions and cylindrical Bessel functions for a wide range of integer-spaced real orders.

Method of solution
An enhanced version of Steed's method, used previously for integer λ in subroutine RCWFN [1] is adopted. The more recent subroutine KLEIN [2] for a single λ value also is similar, the additional feature being the stable recurrence relations for a range of λ values.

Restriction on the complexity of the problem
It is well known that the method used loses accuracy as $x < x_m$ (the turning point for the minimum order required) and eventually, when $G_m \gtrsim 10^6$, a JWKB approximate solution is adopted which is accurate to $\lesssim 1\%$ as a rule. Several output variables signal that this has occurred.

Features of the program
A considerably revised version is presented of the CPC program 'RCWFN' (catalogue ABPC) used to compute the Coulomb functions $F_L(\eta, x)$, $G_L(\eta, x)$ and their x-derivatives over a range of integer L-values $\geqslant 0$. The new program, COULFG, calculates $F_\lambda(\eta, x)$, $G_\lambda(\eta, x)$, $F'_\lambda(\eta, x)$ and $G'_\lambda(\eta, x)$ when λ is real ($\lambda > -1$) and for a similar range in the (η, x) plane as before, $10^4 \geqslant x > 0$, $10^4 \geqslant |\eta|$. Integer-spaced λ-values are obtained by a suitable combination of stable recurrence relations. Subroutine COULFG will return, furthermore, for a range of integer-spaced orders;

a) Spherical Bessel functions $j_\lambda(x)$, $y_\lambda(x)$, $j'_\lambda(x)$, $y'_\lambda(x)$, and
b) Cylindrical Bessel functions $J_\mu(x)$, $Y_\mu(x)$, $J'_\mu(x)$, $Y'_\mu(x)$, where both λ, η can be real or integral. Values of order in excess of 1000 have been tested.

0010-4655/82/0000–0000/$02.75

A "mode" option is provided so that, for each case, arrays of the regular and irregular functions and their derivatives, or just the two functions, or the regular function alone ($F_\lambda, j_\lambda, J_\mu$) are calculated and stored. Both core and execution time are saved by this technique.

In the region of x where the functions oscillate, i.e. $x \geqslant x_\lambda$ (the turning point for the λth partial wave) the relative accuracy of the functions for IBM machines (REAL*8) as programmed, is $\approx 10^{-14}$–10^{-16} and for CDC machines (single precision) is $\approx 10^{-12}$–10^{-14}. With no change in the code except for the accuracy parameter the accuracy can be increased to $\approx 10^{-30}$ by using the AUTODOUBLE facility on an extended-precision IBM compiler. The accuracy decreases in a predictable way as x decreases below x_λ; a measure of this decreased accuracy is provided by the program. JWKB approximations are provided when x is sufficiently smaller than x_λ.

Typical running time

The test deck ran in $2\frac{1}{2}$ s on the GI compiler and the HX compiler, one half of which demonstrates error conditions.

References

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

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

LONG WRITE-UP

1. Introduction

This is the third paper in the current series which discusses programs deriving from Steed's method of calculating Coulomb wavefunctions. The previous papers, to be referred to as I, II, dealt with an overview of the algorithms and their family relationships [1], and with KLEIN [2], a one-shot program for a fixed real angular momentum quantum number, λ, real η and real x (i.e. positive energy). KLEIN returns the values of the regular solution, $F_\lambda(\eta, x)$, the irregular solution, $G_\lambda(\eta, x)$ and their x-derivatives to the same accuracy. The present paper describes COULFG, which is designed to extend the methods of KLEIN to a range of integer-spaced λ-values, $m \leqslant \lambda \leqslant l$, where $l - m \equiv 0 \pmod 1$ and $m > -1$, and this is equivalent for integer $\lambda = L$ to the original subroutine RCWFN of Barnett et al. [3]. Hence it provides Coulomb wavefunction solutions to the non-relativistic Schrödinger equation for real $\eta \gtrless 0$. For $m = l$ the solutions to the relativistic Schrödinger equation, i.e. the Klein–Gordon equation, are obtained, by setting (see I, II for notation)

$$\lambda = \left[\left(L + \tfrac{1}{2}\right)^2 - Z^2\alpha^2\right]^{1/2} - \tfrac{1}{2}, \qquad (1)$$

and the results of KLEIN are reproduced. By the use of similar expressions for lambda, given in II, scattering-state solutions with positive energy of both the (relativistic) Dirac equation, and the Klein–Gordon equation in N-dimensional space, can be obtained. In addition, both Bessel functions and spherical Bessel functions (regular and irregular) over a wide range of orders can be obtained readily from COULFG. The accuracies of all functions, when x is in the oscillating region, are about 10^{-14}–10^{-16} in the standard (REAL*8) IBM program using 56-bit words; on a single precision CDC compiler with 48-bit words the accuracy becomes 10^{-12}–10^{-14}. By changing one card (line 107) and using an extended-precision IBM compiler with the AUTODOUBLE facility the accuracy can be squared ($\approx 10^{-30}$). Subsequent papers will deal with programs specifically for Bessel functions, for the real Gamma function and for the Airy functions.

The present program COULFG supercedes RCWFN [3] in its capabilities but is identical in principle and for identical parameters will produce identical results, with the exception that for x-values below the turning point

$$x_\lambda = \eta + \left[\eta^2 + \lambda(\lambda + 1)\right]^{1/2} \qquad (2)$$

for the minimum $\lambda = m$, the earlier program provided an integration method for G and G', which has been removed. If x is sufficiently smaller than x_m, that is for $F \lesssim 10^{-6}$ and $G \gtrsim 10^6$, a JWKB approximation, see II, is employed which in general provides results to better than 1%. The specific methods of the standard programs of Bardin et al. [4] treat the small-x case directly.

The subroutine COULFG includes KLEIN (in

effect) and it includes the improvements made in RCWFF [5]; thus it incorporates a 'mode' parameter such that when MODE = 1 then F_λ, G_λ, F'_λ and G'_λ are used in the calculation and the arrays filled, when MODE = 2 only the arrays of the functions F_λ, G_λ are used and filled, while when MODE = 3 only the array F_λ is used and has results stored in it. This technique economises on core storage and time when some of the quantities are not required. Thus, in part of a typical application [6], values of F_λ from $m = 0$ to $l = 800$ were required but not the irregular solution or the derivatives. Thus the array declaration statement DIMENSION FC(801), GC(1), FCP(1), GCP(1) was adequate in the calling program.

An extensive description of the algorithm, also called COULFG, is given in I and full details of the Coulomb functions for real λ appeared in II and in related papers which examine Steed's method [3] in detail for accuracy [7], and range and limitations [8], and comprehensive references to the extensive literature on Coulomb functions are given there. The most recent summary [9] is that of Kölbig in 1972, while Fullerton [10] has compiled a large bibliography of calculations of special functions which includes the Coulomb functions but not Klein–Gordon ones.

The relativistic Coulomb equation, solved by COULFG for real $x > 0$, real $\lambda > -1$ and real η, including $\eta = 0$, is

$$d^2U/dx^2 + \left[1 - 2\eta/x - \lambda(\lambda+1)/x^2\right] U_\lambda = 0. \quad (3)$$

The next section details the relationship to the Bessel functions and the method of solution, section 3 contains program notes and the last section describes the test output and program listing.

2. Subroutine COULFG

2.1. The Coulomb functions

The programming closely follows the algorithm specified in I (with some errata corrected [11]) and the details in II. Familiarity with these two references will be assumed and the range of λ-values will be taken as $m \leqslant \lambda \leqslant l$, with $l - m$ an integer. First the ratio $f_l = F'_l(\eta, x)/F_l(\eta, x)$ is obtained for the maximum λ-value (XLMAX) required, by means of the continued fraction CF1. The relative accuracy is set by the internal parameter ACCUR, which is 10^{-16} in the listing. Then f_l is set to $s\beta$ and F'_l to $s\beta f_l$, where $\beta = 10^{-30}$ and $s = \text{sign } F_l$. The recurrence relations [1],

$$\begin{aligned} F_{\lambda-1} &= (S_\lambda F_\lambda + F'_\lambda)/R_\lambda, \\ F'_{\lambda-1} &= S_\lambda F_{\lambda-1} - R_\lambda F_\lambda, \end{aligned} \quad (4)$$

are used to find $F_m(\eta, x)$ and $F'_m(\eta, x)$, with F'_λ not being stored if MODE = 2, 3. Their ratio, F'_m/F_m, is f_m and it is combined with $p_m + iq_m$ obtained by means of the continued fraction CF2 (see I, II) to obtain F_m, G_m, F'_m and G'_m at the minimum λ-value required (XLMIN). The regular function is, for example,

$$F_m(\eta, x) = \text{sign}(F_m)\left[(f_m - p_m)^2/q_m + q_m\right]^{-1/2}, \quad (5)$$

(the expression following (25) in I is corrected here [11]), and the others are

$$F'_m = f_m F_m; \quad G_m = \gamma F_m; \quad G'_m = (p_m\gamma - q_m) F_m, \quad (6)$$

where $\gamma = (f_m - p_m)/q_m$.

In order to preserve the accuracy of the functions if it should happen that F_m is close to a zero, i.e. $f_m > \text{ACCUR}^{-1/2}$, then the solutions could be recast to be multiples of F'_m instead of F_m, i.e.

$$\begin{aligned} F &= F' f^{-1}, \quad G = (1 - pf^{-1}) F', \\ G' &= \left[(1 - pf^{-1})p/q - qf^{-1}\right] F', \end{aligned} \quad (7)$$

where F' is determined as

$$F'_m = \text{sign}(F_m)\left[(1 - pf^{-1})^2/q + qf^{-2}\right]^{-1/2}. \quad (8)$$

However, provided F_m is not exactly zero in the computer representation (see line 203 of the listing) the method as programmed loses no accuracy in F'_m, G_m and G'_m. As in KLEIN, ref. II, a quantity PACCQ is made available in a COMMON block which provides a guide to the final relative accuracy of the functions (see next section).

The final stage of the calculation is the upward recurrence of G_λ and G' from $\lambda = m$ to $\lambda = l - 1$

means of

$$G_{\lambda+1} = (S_{\lambda+1}G_\lambda - G'_\lambda)/R_{\lambda+1},$$
$$G'_{\lambda+1} = R_{\lambda+1}G_\lambda - S_{\lambda+1}G_{\lambda+1}, \quad (9)$$

(corrected eqs. (26) from I). This is bypassed entirely if MODE = 3 (F_λ only required) and only G_λ is stored if MODE = 2.

2.2. The subroutine COULFG for Bessel functions

The relationship between the Coulomb functions and the Bessel functions is a very close one, and it is fruitful to think of the Coulomb functions as independent basis functions in that they have a unit Wronskian and remain finite as $x \to \infty$, behaving in the simplest way as circular functions of unit amplitude, $F_\lambda \to \sin\theta_\lambda$ and $G_\lambda \to \cos\theta_\lambda$. The angle θ_λ is the Coulomb phase

$$\theta_\lambda = x - \eta\ln(2x) - \tfrac{1}{2}\pi\lambda + \arg\Gamma(\lambda+1+i\eta).$$

Bessel functions from this view point, I, are merely renormalised Coulomb function for uncharged particles ($\eta = 0$), and the relationships are;

a) *Spherical Bessel functions for real* $\lambda > -1$

$$j_\lambda(x) = F_\lambda(0,x)/x,$$
$$y_\lambda(x) = -G_\lambda(0,x)/x,$$
$$j'_\lambda(x) = \left[F'_\lambda(0,x) - x^{-1}F_\lambda(0,x)\right]/x, \quad (10)$$
$$y'_\lambda(x) = -\left[G'_\lambda(0,x) - x^{-1}G_\lambda(0,x)\right]/x, \text{ and}$$

b) *Cylindrical Bessel functions for real* $\mu > -\frac{1}{2}$

$$J_\mu(x) = (2/\pi x)^{1/2}F_{\mu-1/2}(0,x),$$
$$Y_\mu(x) = -(2/\pi x)^{1/2}G_{\mu-1/2}(0,x),$$
$$J'_\mu(x) = (2/\pi x)^{1/2} \times \left[F'_{\mu-1/2}(0,x) - \tfrac{1}{2}x^{-1}F_{\mu-1/2}(0,x)\right], \quad (11)$$
$$Y'_\mu(x) = -(2/\pi x)^{1/2} \times \left[G'_{\mu-1/2}(0,x) - \tfrac{1}{2}x^{-1}G_{\mu-1/2}(0,x)\right].$$

They are calculated precisely in this manner in COULFG, controlled by the 'function' parameter KFN which determines the function; KFN = 0 (Coulomb), = 1 (spherical Bessel), = 2 (cylindrical Bessel). A related paper in this series will describe the companion subroutines BESSJY and SBESJY which evaluate cylindrical and spherical Bessel functions, respectively, and which are progressively more compact, as was outlined in fig. 1 of I.

3. Notes on subroutine COULFG

The calling sequence is:

CALL COULFG(XX, ETA1, XLMIN, XLMAX, FC, GC, FCP, GCP, MODE 1, KFN, IFAIL)

and the named COMMON block contains COMMON/STEED/PACCQ, NFP, NPQ, IEXP, M1. The variables have the following type and meaning (and 'real' indicates REAL*8 on the IBM version). The basic accuracy is set, within 2–3S, by the parameter ACCUR. An appropriate value for the IMB (REAL*8) version is 10^{-16} (56-bit mantissa), and for the CDC (single precision) version it is 10^{-14} (48-bit mantissa); this is set within the code, and although the value could be transmitted as an argument if desired, for most purposes variable accuracy is not needed. For ultra-precise work (REAL*16 variables on an IBM system) then 10^{-33} is a suitable value (112-bit mantissa) while the use of DOUBLE PRECISION variables on CDC machines can allow ACCUR = 10^{-28} (96-bit mantissa).

XX (real) $x > 0$ The accuracy of the Coulomb functions is reduced if $x < x_m$, where x_m is the turning point for the lowest angular momentum, m (eq. (2)). An estimate of the reduction in accuracy, is to be found in the variable PACCQ in the COMMON block.
COULFG fails at once (IFAIL = −1) if $x <$ ACCUR$^{1/2}$ ($= 10^{-8}$ in the standard version) and no change is made to the output arrays.

ETA1 (real) η Positive, negative or zero. COULFG has been tested for the η-range $10^4 \geqslant \eta \geqslant -10^4$. If Bessel functions are requested (KFN = 1,2), then η is set to zero (line 117)

XLMIN (real) m Minimum value of the order (angular momentum) for F_λ, j_λ or J_μ.

XLMAX (real) l Maximum value of the order.

The subroutine fails if $l < m$. The difference between these two quantities must be an integer $\geqslant 0$. Otherwise the calculation proceeds from $\lambda = m$ to $\lambda = m + \mathrm{INT}|l - m + \mathrm{ACCUR}|$ and an informative message (FORMAT 2040, line 328) is output.

Limitation on m For the function parameter; KFN = 0, 1 then $m > -1$. For cylindrical Bessels, when KFN = 2, then $m > -\frac{1}{2}$.

KFN (integer) Function parameter, i.e.
KFN = 0 Coulomb functions,
KFN = 1 spherical Bessel functions,
KFN = 2 cylindrical Bessel functions.

MODE1 (integer) For each of these choices of KFN there is the option of not calculating or storing the derivatives (MODE1 = 3) or the derivatives and the irregular function MODE1 = 2). Table 1 contains the details.

FC (N1), GC (N2), FCP (N3), GCP (N3) These arrays contain the output functions F_λ, G_λ, F'_λ, G'_λ, arranged by order, and should be dimensioned to at least to the size L1, the integer part of XLMAX + 1 = int(l + 1). When MODE1 = 1, all arrays must be at least this size- N1, N2, N3 $\geqslant$ L1.

N2, N3 $\geqslant$ 1. The unused arrays are not accessed during the program execution.

IFAIL (integer) Monitors the subroutine execution. A non-zero value indicates a program failure and is accompanied by an error message.

IFAIL = 0 Successful calculation, with no errors detected.

= −1 Input x-value XX $\leqslant$ SQRT (ACCUR). The values are printed according to FORMAT 2000, and control returns to the calling program.

= −2 Problem with the order values XLMIN, XLMAX or XLM (minimum λ for the equivalent Coulomb function), where XLM = XLMIN for KFN = 0, 1 and XLM = XLMIN − 0.5DO for KFN = 2.
Either XLMAX < XLMIN or XLM $\leqslant$ −1. The incorrect values are output via FORMAT 2005, and control returns to the calling program.

= 1 The continued fraction CF1 has failed to converge to within a relative accuracy ACCUR, after 20000 iterations (the loop around statement 4). This will occur for $x/(l+1) \gtrsim 20000$. This maximum number can be increased (it is the value of ABORT, line 101) but the subroutine is in an inefficient region [7,8] and asymptotic

Table 1
The choice of the parameters KFN and MODE1 and the functions stored in the arrays FC, GC, FCP, GCP

KFN	MODE1 = 1	MODE1 = 2	MODE1 = 3
0	$F_\lambda(\eta, x)$, $G_\lambda(\eta, x)$ $F'_\lambda(\eta, x)$ $G'_\lambda(\eta, x)$	$F_\lambda(\eta, x)$, $G_\lambda(\eta, x)$	$F_\lambda(\eta, x)$
1	$j_\lambda(x)$, $y_\lambda(x)$, $j'_\lambda(x)$, $y'_\lambda(x)$	$j_\lambda(x)$, $y_\lambda(x)$	$j_\lambda(x)$
2	$J_\mu(x)$, $Y_\mu(x)$, $J'_\mu(x)$, $Y'_\mu(x)$	$J_\mu(x)$, $Y_\mu(x)$	$J_\mu(x)$

Notes

1. The derivatives F', G', j', y', J', Y', are with respect to the dimensionless variable x.
2. The range of λ for KFN = 0, 1, is $-1 < m \leqslant \lambda \leqslant l$ and the arrays are filled from the starting element

   M1 = max(int(m), 0) + 1, i.e. M1 $\geqslant$ 1.

   For KFN = 2 then the range of the order μ is

   $-\frac{1}{2} < m \leqslant \mu \leqslant l$,

   starting from M1 as defined above. The value of M1 is available in the COMMON block.
3. The internal parameter XLM, lines 126–127, is, in each case, the equivalent Coulomb function angular momentum (eqs. (10), (11)).

methods, eg. ref. [4], might well be preferable. Values of the relevant variables are output with FORMAT 2010, e.g. the maximum λ-value of XLM + int($l-1+$ACCUR) and control returns to the calling program with the arrays undefined.

= 2 The continued fraction CF2 has failed to converge after 20000 iterations (the loop around statement 8), according to the criterion (line 242).

$$|dp| + |dq| \leqslant \text{ACCUR} \times (|p| + |q|).$$

The failure may occur for $x \ll x_m$ and values of the relevant variables are output according to FORMAT 2020. Control returns to the calling program with arrays FC and possibly, FCP and GC modified, and GCP not defined.

= 3 Apparent convergence of CF2 but either

a) q is negative, or

b) q is so small ($< 10^4 \times \text{ACCUR} \times |p|$, i.e. $10^{-10} \times |p|$) that its value and the 'convergence' are suspect. Control returns as for IFAIL = 2.

COMMON/STEED/PACCQ, NFP, NPQ, IEXP, M1 contains the following variables:

PACCQ (real) has the value:
$\frac{1}{2}$ACCUR$/q$ if $|p/q| \leqslant 1$,
$|p| \times \frac{1}{2}$ACCUR$/q$ if $|p/q| > 1$,
$\frac{1}{2}$ACCUR if $q > 1$.
It is a rough guide to the final relative accuracy and hence monitors the number of significant digits lost as the magnitude of q falls greatly below that of p [7,8]. If $G_m > (10^4 \text{ ACCUR})^{-1/2}$ then PACCQ = 1.0DO, and JWKB approximations are used.

NFP (integer) The number of iterations required to evaluate CF1.

NPQ (integer) The number of iterations required to evaluate CF2.
These two qualities are a crude measure of the efficiency of the code for the given input parameters [8].

IEXP (integer) If IEXP = 1 then JWKB was not used and $x(2\eta - x) > \max(m^2 + m, 0)$. If IEXP = 0 and $G > 10^6$, or if IEXP > 70, then the JWKB estimates for G_m and F_m are used and should be scaled by $10^{\pm \text{IEXP}}$. The constant, 70, is close to the underflow exponent on the IBM compiler.

M1 (integer) The starting array element for the function arrays. Thus if the Coulomb functions for $m = 350$ to $l = 449$ were required in the course of a larger calculation (see e.g. ref. [6]) then M1 = 351 and L1 = M1 + 99 = 450 (see line 138).

3.1. Informative message

During the calculation of CF1 it is possible for a loss of significant digits to occur at the step (line 168)

D = TK − D*(ONE + EK*EK).

i.e. $D_k^{-1} = T_k - D_{k-1}(1 + \eta^2/k^2)$, and information that this has happened is printed out (FORMAT 1000) together with appropriate variables. It does not follow that the loss will propagate through to the final f-value [7], or to the Coulomb functions. However, after a second such occurrence, the subroutine fails with IFAIL = 1 since for an extremely large value of x a semi-infinite loop can be encountered. Further remarks are given in II, section 4.1. An additional complication with COULFG arises with the recurrence between the maximum and minimum λ-values.

3.2. Range errors during recurrence in order

During the downward recurrence, DO LOOP 6, for F_λ, F'_λ it is possible for overflows to occur if the λ-range is large enough. An offset of 10^{-30} = TM30 is provided so that a range of $\approx 10^{105}$ is available on the IBM compiler, as programmed. However, upon renormalization by W (lines 296, 297) underflows are unavoidable. Similarly during the upward recurrence for G_λ, G'_λ, in DO LOOP 12, overflows are inevitable after the value of G'_λ reaches $\approx 7.2 \times 10^{75}$ (the limit for the IBM compiler). On the CDC machine the limits are raised to $\approx 10^{290}$ although eventually the same errors will occur. The overflow and underflow errors are not trapped in COULFG on the grounds that normal usage will not require such small values of x/x_λ.

The problems of overflows during the recurrence relations can be overcome by the addition of a scaling factor, for example in powers of $10^{\pm 60}$, which could be extracted from the Coulomb functions as soon as G'_λ, exceeded a power of 10^{60} or F_λ, F'_λ, was less than a power of 10^{-60}.

4. Test calculations with subroutine COULFG

4.1. Test output for COULFG

The test sequence for COULFG examines the use of the subroutine for each function, for a wide range of arguments and for most of the error conditions. The tests were run on an IBM 370/165 system at Daresbury Laboratory and on the replacement NAS 7000 system, using both the G and the HX compilers and REAL*8 variables; they were also verified on a CDC 7600 system at the University of Manchester Regional Computing Centre using single-precision variables and on a GEC 4065 machine.

Examples which appear in Bardin et al. [4] are given for a range of λ-values, for both positive and negative values of η; some tests from Kölbig's review [9], a few calculations from earlier papers [12–14], and an example from the heavy-ion DWBA program A-THREE of Auerbach [15] using parameters appropriate to the scattering of ^{18}O on ^{64}Ni at 65 MeV.

For each set (range of λ) the values of NFP and NPQ are given, the first being the number of iterations to evaluate CF1 at $\lambda = l$ (the maximum value) and the second referring to CF2 at $\lambda = m$ (the minimum value). If the value of PACCQ is greater than $\frac{1}{2}$ACCUR ($= 5 \times 10^{-17}$ in this case), then the value of x is less than the turning point for the lowest partial wave, x_m, and accuracy begins to be lost for this reason. The value of PACCQ is very approximately the relative final accuracy of the Coulomb wavefunctions [1,7,8]. An example is for $\eta = 10$, $x = 5$, $\lambda = 0$–20 when PACCQ $= 2 \times 10^{-6}$.

In cases when the x-value is less than x_m, the JWKB approximation will be used if $G_m > (100\,\mathrm{ACCUR}^{1/2})^{-1}$. These cases are identified by PACCQ = 1.0, NPQ = 0 and by a value of the exponent IEXP which is either 0 or >70 (on the IBM system). The user should check the value of IEXP since the arrays F_λ, F'_λ should be multiplied by $10^{-\mathrm{IEXP}}$ and the arrays G_λ, G'_λ by $10^{+\mathrm{IEXP}}$. The rest of the calculation continues in the normal way using, where necessary, the upward recurrence relations (9).

The use of the MODE parameter is not illustrated but it can be readily verified by setting the function arrays to zero before the call to COULFG, and then using, in turn, MODE = 2, 3 (section 3). There is no change in the derivative arrays when MODE = 2, 3 or in the irregular function array G_λ when MODE = 3.

The Bessel-function tests, which follow in the output, are illustrative only as a full discussion is deferred until the next paper of this series, which will describe the subroutine BESSJY and its MODE-3 version BESSJ. Some of the standard tabulations in Abramowitz and Stegun are recalculated for the cylindrical Bessel [16] and the spherical Bessel [17] cases. The well-known difficulties [1] experienced by non-direct methods in obtaining the irregular function $Y_\mu(x)$ or $y_\lambda(x)$ are emphasised by noting that there are errors in the last two or three places on occasion in refs. [16,17]. Bessel functions of order 1/3, $J_{1/3}$, $Y_{1/3}$ are compared with the 7D values of table III in Watson [18] and complete agreement is found. Once again the trends of NFP and NPQ emphasise the general conclusions drawn in refs. [1,8] that the singularity in the irregular function at $x = 0$ forces a rapid increase of NPQ as $x \to 0$ and that NFP increases approximately as x does. The general methods of COULFG are most effective in the "non-asymptotic" region where x is neither too small nor too large. The JWKB approximation for Bessel functions is contained within the Coulomb formula [2,8] (by setting $\eta = 0$) and is invoked in cases where both $G_m > (100\,\mathrm{ACCUR}^{1/2})^{-1}$ and $x^2 < m(m+1)$ for j_m, y_m, and $x^2 < m^2 - \frac{1}{4}$ for J_m, Y_m, but not for J_0 and Y_0.

The use of the recurrence relations for a λ-value where one of the functions has an approximate zero does not increase the errors for any of the functions: a test example of Blanch [19] is a convenient example of the stability of the methods of COULFG.

Table 2
High-precision calculations using COULFG with ACCUR = 10^{-33}

COMPARISON WITH MAKINOUCHI RESULTS FOR Y(1/3,X)

X	J(1/3,X)	Y(1/3, X)	NFP	NPQ	LOG (PACCQ)
0.7	7.18562 68176 92193 34221 18520 24943 71Q-01	-5.80858 02999 57995 71269 67919 26450 02Q-01	12	497	-33
1	7.30876 40216 94480 47749 29356 76242 81Q-01	-2.78801 64127 59975 99215 39242 05162 13Q-01	14	349	-33
2	4.42939 81814 85762 12250 42241 77359 65Q-01	3.43199 96626 03460 34434 22614 99177 31Q-01	18	179	-33
3	-4.49638 20940 23335 83789 81050 98897 99Q-02	4.56893 03457 23072 30632 40357 02546 11Q-01	21	120	-33
4	-3.55427 37345 45759 87000 89048 87435 90Q-01	1.79416 76634 39443 94849 53963 70935 30Q-01	23	91	-33
6	-1.06747 39474 18904 40139 24765 41850 50Q-02	-3.25257 99210 09400 94932 10079 02183 57Q-01	28	62	-33
8	2.59776 16110 83496 56047 55939 65052 51Q-01	1.09587 79463 36033 60625 60512 57142 16Q-01	32	47	-33
10	-1.86145 16704 86957 60465 75335 78669 90Q-01	1.70201 11788 26882 68761 03280 60054 25Q-01	35	38	-33
20	1.76060 58001 29389 97642 20169 56597 43Q-01	-2.87777 07635 71557 15168 95115 78876 74Q-02	52	21	-33
40	6.92029 42818 85805 20803 34879 10406 39Q-02	1.05478 70367 09870 98929 09112 29602 83Q-01	80	13	-33
60	-5.56181 47270 52814 27615 87694 39943 96Q-02	8.66991 73295 71857 18001 55631 86080 33Q-02	105	11	-33
80	-8.81997 84400 03455 60405 99159 86647 60Q-02	-1.33588 49535 98459 84077 64403 85137 86Q-02	130	10	-33
100	-2.12712 44853 70254 01809 16224 76936 56Q-02	-7.69005 04962 13621 36508 25791 12098 72Q-02	154	9	-33

ALDIS CALCULATIONS OF J0 , -(PI/2)Y0 ; J1 , -(PI/2)Y1 ; FOR SOME X VALUES

X	L	JL(X)	(-PI/2)*YL(X)	NFP	NPQ	LOG (PACCQ)
0.5	0	9.38469 80724 08129 04228 40467 35999 46Q-01	6.98248 39378 38541 94778 28382 83242 92Q-01			
0.5	1	2.42268 45767 48738 86383 95457 61415 92Q-01	2.31138 34293 86515 57283 41934 19262 69Q+00	12	700	-33
0.9	0	8.07523 79812 25447 77302 40904 22876 68Q-01	-8.84092 33886 56204 88250 64906 49226 27Q-03			
0.9	1	4.05949 54607 88056 74604 92234 55339 73Q-01	1.37150 40285 49382 72978 21182 11370 19Q+00	14	396	-33
1.5	0	5.11827 67173 59181 28749 05174 42834 33Q-01	-6.00749 36468 81809 15674 12441 24872 09Q-01			
1.5	1	5.57936 50791 00996 41990 12121 31561 15Q-01	6.47652 87675 64670 95194 07340 73970 98Q-01	16	240	-33
1.9	0	2.81818 55937 43854 70713 55163 25576 62Q-01	-7.80402 98597 09707 98204 91749 17735 15Q-01			
1.9	1	5.81157 07271 34340 72685 55469 14323 84Q-01	2.58247 98328 23478 84848 91189 11665 40Q-01	18	191	-33
2.5	0	-4.83837 76468 19799 63272 87778 85119 80Q-02	-7.82367 09136 90194 68034 91449 14136 49Q-01			
2.5	1	4.97094 10246 42740 38010 81627 62643 70Q-01	-2.29207 67513 09780 77462 05820 58630 37Q-01	19	146	-33
2.9	0	-2.24311 54579 19681 14186 85408 78405 43Q-01	-6.40746 30877 27090 85294 38943 89590 65Q-01			
2.9	1	3.75427 48181 30958 96390 65394 62210 99Q-01	-4.64861 55072 92166 27288 21282 12817 73Q-01	21	126	-33
3.5	0	-3.80127 73998 72633 77378 74930 43869 80Q-01	-2.96914 97519 44652 15872 72527 25170 03Q-01			
3.5	1	1.37377 52736 23271 85716 13 99 71839 79Q-01	-6.44322 46011 15137 33523 49534 95723 26Q-01	22	106	-33
3.9	0	-4.01826 01488 76399 05034 70509 00704 84Q-01	-3.67187 90734 04239 97840 36203 62214 33Q-02			
3.9	1	-2.72440 39620 77992 62531 32367 16774 00Q-02	-6.40602 18866 57087 42081 36413 64872 27Q-01	24	95	-33
4.5	0	-3.20542 50898 51214 24355 48895 95316 18Q-01	3.05841 91236 37943 69171 69716 97961 18Q-01			
4.5	1	-2.31060 43192 33706 34008 09652 46452 64Q-01	-4.72805 48945 29102 79055 67556 75182 04Q-01	25	83	-33
4.9	0	-2.09738 32758 53263 14754 91962 57280 88Q-01	4.58758 28386 20714 00447 97279 72241 21Q-01			
4.9	1	-3.14694 67101 51906 03202 84043 15494 30Q-01	-2.84701 63810 80882 75409 24992 49267 57Q-01	26	76	-33
5.5	0	-6.84386 94178 19196 82395 86787 74184 91Q-03	5.33254 86831 70595 86379 58495 84483 78Q-01			
5.5	1	-3.41438 21542 90433 50179 73099 95445 57Q-01	3.73193 54483 81223 05165 34053 48969 82Q-02	27	68	-33
5.9	0	1.22033 35459 28226 73483 50104 91200 37Q-01	4.78096 88484 10384 43599 34993 49142 75Q-01			
5.9	1	-2.95142 44472 90161 23856 51931 79131 37Q-01	2.32599 04727 79093 57025 11251 12607 18Q-01	28	63	-33

STRECOCK AND GREGORY COULOMB WAVEFUNCTION RESULTS FOR G0(ETA,X), G'0(ETA,X)

VALUES OF JL , J'L ; GL , G'L ARE GIVEN FOR L = 0, L = 10

ETA	X	L			NFP	NPQ	LOG (PACCQ)
2	1	0	2.8898146 8533649 6044189 6440711 0971Q-02	6.1308181 0584794 1514451 4407462 7676Q-02			
2	1	0	9.8003357 6844524 6642042 9032229 8051Q+00	-1.3812624 1208365 8895789 5796491 3417Q+01			
2	1	10	3.0512059 9288145 4939052 0913166 0470Q-12	3.3982794 0432182 7404040 4049918 9356Q-11			
2	1	10	1.5398544 2667003 7875341 8375791 7220Q+10	-1.5623803 9231546 1445944 5938099 0091Q+11	11	423	-31
2	2	0	1.4444656 6874273 3245634 6914741 9981Q-01	1.7961512 8186340 2098609 8641152 4542Q-01			
2	2	0	3.5123830 9130932 8645479 4769757 6902Q+00	-2.5554284 1067469 9965596 5562870 2987Q+00			
2	2	10	7.0154338 9752437 6504592 0144763 6632Q-09	3.9244559 7460866 8968596 8517642 6314Q-08			
2	2	10	1.3336785 7542012 7290338 3151880 9134Q+07	-6.7936455 7645786 3844484 4475364 6860Q+07	13	228	-32
2	4	0	7.7519719 1073771 1204381 2672793 6659Q-01	4.0400781 6494934 7393539 3581787 9682Q-01			
2	4	0	1.3974834 2679301 5884269 9836430 6567Q+00	-5.6167098 7933261 9703070 3057889 4988Q-01			
2	4	10	1.5909141 8373286 0808433 7228075 5249Q-05	4.3875144 8662284 1799079 9095228 8592Q-05			
2	4	10	1.1979164 2273294 2025205 2567189 1717Q+04	-2.9820114 6862891 8110911 8995127 9284Q+04	17	120	-33
10	10	0	1.6262711 2501358 7824895 3638972 0996Q-03	1.7060476 3207928 0601360 1374368 4015Q-03			
10	10	0	3.0787321 6610908 3798699 3827904 6932Q+02	-2.9192772 3808268 2287028 7067923 0969Q+02			
10	10	10	8.1831296 0541446 7250800 8488450 2754Q-06	1.4268600 4231612 9578157 8181769 1323Q-05			
10	10	10	4.2163027 5897239 4716851 7106767 9835Q+04	-5.8989492 4249464 0656065 6063651 8568Q+04	25	93	-28
10	12	0	1.0980049 3843500 0679482 4888890 6944Q-02	9.5118486 4742148 8393239 3274792 5915Q-03			
10	12	0	5.6013146 5285647 8179024 5580876 5244Q+01	-4.2550940 4921594 7716271 6229021 2768Q+01			
10	12	10	1.2544328 6944379 5339357 6133470 4942Q-04	1.5582581 7394648 9182018 2035976 5532Q-04			
10	12	10	3.3339953 9934307 4369145 7281601 3745Q+03	-3.8302363 8339775 0317831 7802006 1744Q+03	27	81	-30
10	18	0	5.0960447 1039365 1023051 0818515 4868Q-01	2.1715082 9106903 7372537 2519023 0321Q-01			
10	18	0	2.8897604 7548786 4071576 3792894 6003Q+00	-7.3093180 6912417 6453045 3079472 9000Q-01			
10	18	10	4.0281043 2861964 6123218 1794449 1001Q-02	2.9087271 7784386 3884088 4072240 6706Q-02			
10	18	10	1.8699033 0697648 0686230 8059256 7472Q+01	-1.1322848 3151536 1176717 6725932 5334Q+01	33	59	-33
15	30	0	1.1077060 5607417 5110755 4649877 5250Q+00	2.6764246 3576946 0718671 8673041 1520Q-01			
15	30	0	1.9237558 4878908 0421641 8173163 3938Q+00	-4.3795124 4059142 7860086 0014457 6125Q-01			
15	30	10	3.7365318 8544987 7622688 2784532 8491Q-01	1.5445324 5077982 0776377 6373537 8907Q-01			
15	30	10	3.9509440 4215345 1827023 1635648 0184Q+00	-1.0431166 7481181 4879887 9893521 0602Q+00	45	49	-32

Table 3
Curtis calculations

X	FL(ETA,X)	Z	PL(A,Z)	P'L(A,Z)	QL(A,Z)	Q'L(A,Z)	NFP	NPQ
			A-VALUE = 2.000	ETA =	-0.707107	L = 0		
1.414	0.424398	1.0	0.141533	-0.465728	0.247936	0.308650	8	60
2.828	-0.872102	2.0	-0.290838	-0.152161	0.082660	-0.503983	12	31
4.243	-0.160995	3.0	-0.053690	0.497580	-0.309635	-0.094746	14	22
5.657	0.936694	4.0	0.312378	0.074412	-0.045234	0.498719	17	17
7.071	0.142960	5.0	0.047676	-0.487531	0.318892	0.077302	19	14
8.485	-0.946141	6.0	-0.315529	-0.091492	0.059404	-0.487182	22	12
9.899	-0.226086	7.0	-0.075398	0.473737	-0.317380	-0.116720	24	11
11.314	0.927308	8.0	0.309248	0.145221	-0.097265	0.468976	26	10
12.728	0.358766	9.0	0.119645	-0.449815	0.305600	0.181299	28	9
14.142	-0.874426	10.0	-0.291613	-0.215643	0.146682	-0.437306	30	8
			A-VALUE = 2.000	ETA =	-0.707107	L = 1		
1.414	0.806284	1.0	0.465728	0.424598	-0.308650	0.743809		
2.828	0.515181	2.0	0.297580	-0.723723	0.462653	0.479306		
4.243	-0.799461	3.0	-0.461786	-0.468929	0.301169	-0.728125		
5.657	-0.534425	4.0	-0.308696	0.705613	-0.464793	-0.484297		
7.071	0.777999	5.0	0.449390	0.502540	-0.332416	0.690743		
8.485	0.613607	6.0	0.354433	-0.651225	0.437678	0.542944		
9.899	-0.708266	7.0	-0.409110	-0.576859	0.388760	-0.618917		
11.314	-0.719871	8.0	-0.415813	0.563908	-0.383869	-0.627681		
12.728	0.594617	9.0	0.343464	0.664236	-0.452943	0.514183		
14.142	0.827694	10.0	0.478095	-0.444553	0.305292	0.714808		
			A-VALUE = 2.000	ETA =	-0.707107	L = 2		
1.414	0.316231	1.0	0.608874	1.415327	-2.681744	2.479364		
2.828	1.007017	2.0	1.938918	0.518441	-0.551065	2.588800		
4.243	0.452389	3.0	0.871032	-2.454104	1.729599	1.217576		
5.657	-0.814388	4.0	-1.568029	-1.543479	1.076216	-2.323967		
7.071	-0.631876	5.0	-1.216619	2.125287	-1.461114	-1.808190		
8.485	0.683438	6.0	1.315896	1.991482	-1.368646	1.960285		
9.899	0.766966	7.0	1.476723	-1.729111	1.190247	2.198852		
11.314	-0.530860	8.0	-1.022122	-2.334597	1.608107	-1.517316		
12.728	-0.876747	9.0	-1.688096	1.248405	-0.863035	-2.504448		
14.142	0.347545	10.0	0.669166	2.591224	-1.791991	0.988865		
			A-VALUE = 0.200	ETA =	-2.236068	L = 0		
0.447	0.447884	1.0	0.267189	-0.230547	0.200725	0.422468	6	180
0.894	-0.186654	2.0	-0.111350	-0.406813	0.368443	-0.083227	7	94
1.342	-0.668542	3.0	-0.398824	-0.133387	0.118312	-0.359491	9	65
1.789	-0.613009	4.0	-0.365695	0.184173	-0.241068	-0.313804	10	50
2.236	-0.144028	5.0	-0.085921	0.340922	-0.446516	-0.080629	11	40
2.683	0.414747	6.0	0.247421	0.294976	-0.397197	0.169717	12	35
3.130	0.765813	7.0	0.456851	0.108862	-0.143252	0.314239	14	30
3.578	0.760264	8.0	0.453541	-0.112429	0.179676	0.306376	14	27
4.025	0.423852	9.0	0.252852	-0.272584	0.426167	0.170015	15	24
4.472	-0.087295	10.0	-0.052077	-0.315866	0.499464	-0.026716	16	22
			A-VALUE = 0.200	ETA =	-2.236068	L = 1		
0.447	0.352790	1.0	0.576367	0.801566	-1.056169	0.602174		
0.894	0.707714	2.0	1.156220	0.244060	-0.252486	0.979087		
1.342	0.610975	3.0	0.998174	-0.531021	0.701541	0.822631		
1.789	0.137871	4.0	0.225246	-0.928151	1.236514	0.204181		
2.236	-0.416507	5.0	-0.680464	-0.802134	1.094605	-0.463863		
2.683	-0.766890	6.0	-1.252898	-0.301820	0.403201	-0.855591		
3.130	-0.765802	7.0	-1.251121	0.298165	-0.478628	-0.840008		
3.578	-0.435227	8.0	-0.711049	0.738456	-1.158981	-0.475080		
4.025	0.073185	9.0	0.119565	0.864837	-1.372077	0.058878		
4.472	0.555068	10.0	0.906838	0.659924	-1.057003	0.547088		
			A-VALUE = 0.200	ETA =	-2.236068	L = 2		
0.447	0.057470	1.0	0.699823	1.832099	-24.293641	31.159617		
0.894	0.304806	2.0	3.711672	3.925266	-12.281445	4.878293		
1.342	0.636330	3.0	7.748707	3.699421	-7.841190	4.814571		
1.789	0.846894	4.0	10.312784	1.126228	-2.268677	6.182570		
2.236	0.793999	5.0	9.668670	-2.435452	3.937810	5.866806		
2.683	0.465932	6.0	5.673740	-5.318868	8.759894	3.475985		
3.130	-0.027435	7.0	-0.334079	-6.327195	10.473014	-0.148924		
3.578	-0.511608	8.0	-6.229937	-5.112727	8.498059	-3.670391		
4.025	-0.819428	9.0	-9.978323	-2.173931	3.580603	-5.865773		
4.472	-0.850384	10.0	-10.355285	1.427603	-2.555408	-6.051639		

4.2. High-accuracy test calculations

Test calculations using the high-accuracy mode of COULFG, with ACCUR $= 10^{-33}$, are given in table 2 and these represent the current convenient accuracy limit of the code. Table 2 contains examples of Bessel functions of order $\mu = 1/3$ taken from the exceptional work of Makinouchi, who lists in ref. [20] values of $Y_{1/3}(x)$ between $x = 0.01$ and $x = 100$ to 30 decimals. It should be noted that these were computed from his values of $J_{1/3}(x)$ and are not obtained directly as in the present method. Nevertheless the agreement is virtually complete with the maximum discrepancy being in the last digit. At the same time, of course, COULFG also produces $J_{1/3}(x)$, $J'_{1/3}(x)$ and $Y'_{1/3}(x)$ to the same accuracy. The calculation of Airy functions using COULFG (or KLEIN) and Bessel functions of order 1/3 is discussed in I and also in a subsequent paper of this series.

The equally remarkable calculations of Aldis in 1900 [21], who obtained $J_0(x)$, $J_1(x)$, $-\frac{1}{2}\pi Y_0(x)$, $-\frac{1}{2}\pi Y_1(x)$ to 21D, for $x = 0.1$ (0.1) 6.0 are illustrated in table 2. A few of his x-values are displayed and every digit is correct for his complete table. The irregular function is that defined by Heine (ref. [18] p. 65) and is of no particular current relevance.

Finally, a few of the high-precision results of Strecok and Gregory [21] for $\lambda = 0$ are given; here values for $\lambda = 10$ are also quoted. The full comparison was made in ref. [7].

4.3. The results of Curtis for negative η

Table 3 is a comparison of some of the results of Curtis [23] which are appropriate to electron scattering at positive energies. The relation between the functions of Curtis, $P_L(a,z)$, $Q_L(a,z)$ and the Coulomb functions $F_\lambda(\eta,x)$, $G_\lambda(\eta,x)$ is given by his equations (2.74)–(2.77) and (6.15)–(6.16) namely;

$$\begin{aligned} P_L(a,z) &= K_L(a)\, F_L(\eta,x), \\ Q_L(a,z) &= K_L(a)(1-\mathrm{e}^{2\pi\eta})^{-1} G_L(\eta,x), \end{aligned} \tag{12}$$

where

$$\eta = -a^{-1/2}, \quad x = za^{1/2}, \quad z = -\eta x,$$

$$\begin{aligned} K_0 &= a^{-1/4}(2\pi)^{-1/2}(1-\mathrm{e}^{2\pi\eta})^{1/2}, \\ K_1 &= 3(1+a)^{-1/2} K_0, \\ K_2 &= 10(1+4a)^{-1/2} K_1. \end{aligned} \tag{13}$$

In the table results are given for $a = +2.0$ ($\eta = -0.7071$) and $a = +0.2$ ($\eta = -2.2361$) for a range of z-values and for both forms of the Coulomb functions. The agreement between the calculations is exact.

5. Comments on the program

5.1. Modifications to COULFG

The ultimate accuracy of Steed's method, around which COULFG is based, is not that of table 2 (i.e. $\approx 10^{-33}$) but is limited by the effective word-length of the computer used. In principle, then, arbitrary accuracy can be achieved with a suitable multiple-precision package, such as that of Schonfelder [24] or of Brent [25], and these could be exploited to provide an enhancement of COULFG or of KLEIN for research purposes. Indeed, such efforts are being made by the author for the Γ-function subroutine [1].

In normal use when the value of PACCQ, which approximates the resultant relative accuracy, becomes too large for the application in hand, i.e. when x becomes significantly less than x_m, then one has recourse to various alternatives:

1) the integration which was removed from RCWFN [3], lines 60–67, 107 and 149–182, could be restored;

2) the value of ACCUR could be decreased with the use of extended precision (at the cost of significantly greater computing time);

3) the asymptotic methods exploited by Bardin et al. [4] could be used; or

4) the JWKB estimates employed in KLEIN [2] could be adopted. This last alternative is programmed in COULFG and the resultant accuracies are in general better than 1%.

To retain high accuracy over an extended integration range is not really possible even with the 4th-order Runge–Kutta method [3], and improved techniques such as those of Strecok and Gregory [22] are required.

5.2. Integer values of the order

Since calculations with λ integral are likely to be a frequent requirement it may be convenient to detail here the modifications necessary to convert the subroutine to accept integer values of the Coulomb or Bessel order parameter. They are given in table 4.

5.3. Modifications for complex-variable programming

In operating the code on a machine with adequate single-precision word-length (e.g. a CDC computer with a 48-bit mantissa), the evaluation of CF2 can be replaced by rather more compact and transparent coding as is detailed in table 4. Use is made of a local arithmetic function CXMOD which does not use the square-root function. Certain compilers support NON-ANSI DCMPLX variables, i.e. COMPLEX*16, and indeed the IBM HX compiler allows extended-precision complex functions. Suitable changes for these are also given in the table.

Table 4
Possible modifications to COULFG

```
***                INTEGER VALUES OF THE ORDER                          ***

CHANGE XLMIN,XLMAX  TO  LMIN,LMAX AT LINES 55,126,128,137,309,310
CHANGE FORMAT IN LINE 311 TO  2I10,1PD15.6
DELETE LINES 71 - 77,131,132,328
REPLACE LINE 70 BY  'AND INTEGER LAMBDA .GE. 0'
CHANGE LINE 133 TO  LXTRA = LMAX - LMIN
CHANGE LINE 138 TO  M1    = LMIN + 1
    IN CALLING PROGRAM LINE 30 MAKE THE LAMBDA VALUES INTEGRAL BY,FOR
    EXAMPLE, CHANGING  XM,XL(JMAX)  INTO  IDINT(XM),IDINT(XL(JMAX))


***  COMPLEX-VARIABLE PROGRAMMING WITH SINGLE-PRECISION VARIABLES    ***

TO CONVERT COULFG INTO SINGLE PRECISION, DELETE 95,332, CHANGE 'D' INTO
'E' EXPONENTS IN LINES 101 - 103,107 & FUNCTION NAMES LISTED IN 100,335
DELETE DZERO & 0.0D0 IN 337 AND CHANGE DZERO TO ZERO IN 341
ADD FOLLOWING DECLARATION AFTER LINE 97
      COMPLEX       C1,AA,BB,DD,DL,PQ,TWOI

INSERT NEXT TWO LINES AFTER LINE 106 - DATA & LOCAL ARITHMETIC FUNCTION
      DATA C1,TWOI / (1.0E0,0.0E0) , (0.0E0,2.0E0) /
      CXMOD(PQ) = ABS(REAL(PQ)) + ABS(AIMAG(PQ))
NOTE THAT SINGLE PRECISION CONSTANTS AND FUNCTIONS ARE USED

REPLACE LINES 215 - 242 BY THE NEXT 15 LINES
      PQ = CMPLX(ZERO,ONE - ETA*XI)
      AA = CMPLX(-E2MM1,ETA)
      BB = CMPLX(TWO*(X - ETA),TWO)
      DD = C1/BB
      DL = AA*DD*CMPLX(ZERO,XI)
    8 PQ     = PQ + DL
         PK  = PK + TWO
         AA  = AA + CMPLX(PK,WI)
         BB  = BB + TWOI
         DD  = C1/(AA*DD + BB)
         DL  = DL*(BB*DD - C1)
             IF(PK .GT. TA)                    GO TO 120
             IF(CXMOD(DL) .GE. CXMOD(PQ)*ACC)  GO TO 8
         P   = REAL (PQ + DL)
         Q   = AIMAG(PQ + DL)

REPLACE OUTPUT LINES 319,324 BY NEXT TWO LINES
      WRITE (6,2020) ABORT,PQ,DL,ACC
      WRITE (6,2030) PQ,ACC,LXTRA,M1


***          NON-ANSI DOUBLE-PRECISION COMPLEX VARIABLES              ***

CHANGE FUNCTION TYPES ABOVE    COMPLEX   , CMPLX, ABS, REAL,AIMAG
                      BECOME   COMPLEX*16,DCMPLX,DABS,DREAL,DIMAG


NOTE THAT EXTENDED-PRECISION WITH COMPLEX*32 VARIABLES IS AVAILABLE
ON THE IBM HX COMPILER SIMPLY BY THE USE OF THE AUTODOUBLE FACILITY
A SUITABLE ACCUR IS THEN 10**(-33) FOR THE 112-BIT MANTISSA
MAKE SURE THE EXPONENTS OF LINES 102, 103 READ  'Q0'


*** IF RESULTS OF DOUBLE-PRECISION CALCULATIONS ARE TO BE STORED IN
      SINGLE-PRECISION ARRAYS                                          ***

REPLACE LINE 288 WHICH EXTRACTS RL FROM STORAGE(NOW SINGLE PRECISION) BY
      IF(ETANE0) RL = DSQRT(ONE + EL*EL)
```

5.4. Single-precision arrays

It occasionally happens that restricting the Coulomb function arrays to REAL*4 accuracy is thought to be adequate (though probably not for matching to internal wavefunctions) while their calculation should be carried out in REAL*8 variables. For such a situation the only change that needs to be made to COULFG is to avoid the use of the stored value $RL = (1 + \eta^2/\lambda^2)^{1/2}$, which is in the array GC, since it is there truncated to REAL*4. Instead the value should be recalculated at the point where it is required in the upward recurrence; that is, replace in line 288 RL = GC(L + 1) by RL = DSQRT(ØNE + EL*EL). This modification will preserve REAL*8 accuracy of the irregular functions until the final truncation by storage. In this respect the coding in COULFG is superior to that in RCWFN.

References

[1] A.R. Barnett, Comput. Phys. Commun. 21 (1981) 297.
[2] A.R. Barnett, Comput. Phys. Commun. 24 (1981) 141.
[3] A.R. Barnett, D.H. Feng, J.W. Steed and L.J.B. Goldfarb, Comput. Phys. Commun. 8 (1974) 377.
[4] C. Bardin, Y. Dandeu, L. Gauthier, J. Guillermin, T. Lena, J.-M. Pernet, H.H. Wolter and T. Tamura, Comput. Phys. Commun. 3 (1972) 73.
[5] A.R. Barnett, Comput. Phys. Commun. 11 (1976) 141.
[6] D.H. Feng and A.R. Barnett, Comput. Phys. Commun. 10 (1975) 401.
D.H. Feng, A.R. Barnett and L.J.B. Goldfarb, Phys. Rev. 13C (1976) 1151.
[7] A.R. Barnett, J. Comput. Appl. Math. 8 (1982) 29.
[8] A.R. Barnett, J. Comput. Phys. 46 (2) (1982).
[9] K.S. Kölbig, Comput. Phys. Commun. 4 (1972) 214.
[10] L.W. Fullerton, Bell. Lab. Comp. Sci. Technical Report 86, unpublished (1980).

[11] A number of important errors in ref. [1] should be corrected:
a) the quantity F_{LMAX} after eq. (25), before eq. (34), and in section 5 should be read as sign(F_m);
b) in the upward recurrence relations eq. (26), (36), for the quantity S_λ read $S_{\lambda+1}$;
c) the angular momentum term in square brackets in eq. (46) is indeed negative.
[12] I.A. Stegun and M. Abramowitz, Phys. Rev. 98 (1955) 1851.
[13] M. Abramowitz and P. Rabinowitz, Phys. Rev. 96 (1954) 77.
[14] L.C. Biedenharn, R.L. Gluckstern, M.H. Hull and G. Breit, Phys. Rev. 97 (1955) 542.
[15] E.H. Auerbach, Comput. Phys. Commun. 15 (1978) 165.
[16] F.T.W. Olver, in: Handbook of mathematical functions, eds. M. Abramowitz and I.A. Stegun, N.B.S. Appl. Math. Series 55 (US Govt. Printing Office, Washington, 1965) chap. 9, p. 358.
[17] H.A. Antosiewicz, ibid, chap. 10, p. 435.
[18] G.N. Watson, A treatise on the theory of Bessel functions, 2nd ed. (Cambridge Univ. Press, London, 1966).
[19] G. Blanch, SIAM Rev. 6 (1964) 383.
[20] S. Makinouchi, Techn. Repts. Osaka Univ. 15 (1965) 185.
[21] W.S. Aldis, Proc. Roy. Soc. LXVI (1900) 32.
[22] A.J. Strecok and J.A. Gregory, Math. Comput. 26 (1972) 955.
[23] A.R. Curtis, Royal Society Math. Tables, vol. 11 (Cambridge Univ. Press, London, 1964) p. 92.
[24] J.L. Schonfelder, MLARITHA, University of Birmingham, unpublished.
[25] R.P. Brent, ACM Trans. Math. Software 4 (1978) 57.

TEST RUN OUTPUT

TEST OF THE MANCHESTER COULOMB WAVEFUNCTION PROGRAM - COULFG

ETA	X	XL	F	G	FP	GP	NFP	NPQ	PACCQ
-50.000	5.000	0.0	1.52236975714D-01	4.41680690236D-01	2.03091041166D+00	-6.76485374767D-01			
-50.000	5.000	10.0	-3.68114360218D-01	3.31588324611D-01	1.33846751032D+00	1.51088862814D+00			
-50.000	5.000	50.0	4.20266913301D-23	1.32186136031D+21	3.83028028856D-22	-1.17470596213D+22	7	45	5.D-17
-50.000	50.000	10.0	2.77129451057D-01	-7.10519441453D-01	-1.22068610371D+00	-4.78761065984D-01			
-50.000	50.000	20.0	7.55876294797D-01	-1.51281009656D-01	-2.52360602589D-01	-1.27246037462D+00	91	14	5.D-17
-4.000	5.000	0.0	4.07862723006D-01	6.74327035383D-01	1.09821233636D+00	-6.36110427280D-01			
-4.000	5.000	5.0	2.83657834139D-01	-8.70189296614D-01	-1.04544995379D+00	-3.18202528554D-01			
-4.000	5.000	10.0	7.21985521560D-02	5.28003488054D+00	1.10084353023D-01	-5.79998855560D+00	11	21	5.D-17
-4.000	50.000	50.0	1.59533986051D+00	-5.15310138675D-03	-8.64299712691D-02	-6.26546507323D-01			
-4.000	50.000	100.0	3.14179162347D-19	9.37788817811D+17	5.37489508885D-19	-1.57855264866D+18	13	28	1.D-16
10.000	5.000	0.0	1.72074051358D-06	1.67638043033D+05	3.09759065782D-06	-2.79371561379D+05			
10.000	5.000	5.0	1.31825659927D-07	1.84916325924D+06	2.80452562756D-07	-3.65177329860D+06			
10.000	5.000	20.0	4.04282716774D-18	2.77871505109D+16	1.83778234798D-17	-1.21037242653D+17	8	61	2.D-06
10.000	50.000	30.0	-3.52635436430D-01	-1.39731750174D+00	-6.64436751932D-01	2.02960027068D-01			
10.000	50.000	50.0	5.76729101127D-03	1.34226818062D+02	3.89565495430D-03	-8.27249102369D+01	22	20	1.D-16
100.000	1000.000	0.0	-1.65581311975D-01	-1.04432596294D+00	-9.34063273097D-01	1.48165711330D-01			
100.000	1000.000	50.0	-1.02728883766D+00	2.53963904213D-01	2.26855903270D-01	9.17353264488D-01			
100.000	1000.000	100.0	9.36270607951D-01	-4.98554253164D-01	-4.43161964632D-01	-8.32088192320D-01	868	6	6.D-17
			JWKB EXAMPLES CALCULATIONS OF BARDIN ET AL.						
100.000	1.000	0.0	9.01215659555D-03	3.93122386932D+00	1.29458381023D-01	-5.44897458484D+01			
100.000	1.000	5.0	1.07533483439D-03	3.07107311677D+01	1.65864639122D-02	-4.56246324475D+02			
100.000	1.000	10.0	4.72013782540D-06	6.02336440657D+03	8.46148386750D-05	-1.03881287072D+05			
100.000	1.000	30.0	7.10143042253D-26	2.09518618848D+23	2.41881682995D-24	-6.94526045033D+24	5	0	1.D+00
	**** IEXP = 123 F,FP *10**(-IEXP) G,GP *10**(IEXP)								
100.000	5.000	0.0	1.63540865026D-02	4.89523124695D+00	1.02979707085D-01	-3.03221167381D+01			
100.000	5.000	20.0	4.77275504854D-08	1.40230439512D+06	3.59703975637D-07	-1.03836364737D+07	12	0	1.D+00
	**** IEXP = 108 F,FP *10**(-IEXP) G,GP *10**(IEXP)								
100.000	50.000	0.0	1.70424082262D-54	1.69384477356D+53	2.96321541644D-54	-2.92257584011D+53			
100.000	50.000	20.0	4.53710831483D-56	6.19148506714D+54	8.10665703799D-56	-1.09778631121D+55	31	0	1.D+00
10.000	5.000	30.0	3.20735724701D-28	2.45851379719D+26	2.06510422204D-27	-1.53488133636D+27			
10.000	5.000	50.0	2.15139724500D-52	2.26795313756D+50	2.22594022701D-51	-2.30160742729D+51	6	0	1.D+00
-4.000	5.000	30.0	4.63196983554D-19	1.83494540622D+17	2.77366206877D-18	-1.06012813184D+18			
-4.000	5.000	50.0	4.43423340067D-43	1.13093510555D+41	4.46626563407D-42	-1.11607643453D+42	6	0	1.D+00
500.000	0.001	0.0	6.15007019274D-05	7.51159667969D+00	8.81383260404D-02	-5.49490383314D+03	5	0	1.D+00
	**** IEXP = 679 F,FP *10**(-IEXP) G,GP *10**(IEXP)								
500.000	0.025	0.0	4.87216122025D-04	5.11375617981D+00	1.02722117830D-01	-9.74319431732D+02	10	0	1.D+00
	**** IEXP = 676 F,FP *10**(-IEXP) G,GP *10**(IEXP)								
			OTHER TEST EXAMPLES OF COULOMB FUNCTIONS						
2.000	10.000	0.0	-1.06160862047D+00	-3.99306648097D-01	-2.93530453784D-01	8.31560069659D-01			
2.000	10.000	1.0	-8.65737387536D-01	-7.46893228412D-01	-5.55780527035D-01	6.75598970652D-01	23	15	6.D-17
4.000	10.000	1.0	1.37343785651D+00	5.23328665858D-01	1.37992480244D-01	-6.75519882474D-01			
4.000	10.000	4.0	8.89699380561D-01	1.56161024485D+00	3.43085462667D-01	-5.21787737277D-01	18	21	1.D-16
5.000	10.000	0.0	9.17944918946D-01	1.60852455560D+00	3.31032101931D-01	-5.09318942458D-01			
5.000	10.000	1.0	8.53200693407D-01	1.70871952005D+00	3.29295517733D-01	-5.12571455183D-01			
5.000	10.000	7.0	8.94776433905D-02	7.63318878690D+00	7.46133154652D-02	-4.81083835599D+00	16	26	2.D-16
10.000	10.000	0.0	1.62627112503D-03	3.07873216608D+02	1.70604763209D-03	-2.91927723806D+02			
10.000	10.000	7.0	9.92870627105D-05	4.03294624069D+03	1.28959191579D-04	-4.83360570871D+03			
10.000	10.000	10.0	8.18312960548D-06	4.21630275894D+04	1.22686004233D-05	-5.89094924245D+04	14	40	4.D-12
25.000	10.000	0.0	1.54838713210D-16	1.61423768840D+15	3.14570932195D-16	-3.17884161766D+15			
25.000	10.000	1.0	1.42761067568D-16	1.74645170091D+15	2.90760570837D-16	-3.44772363283D+15	21	0	1.D+00
5.000	5.000	0.0	2.76730116686D-02	1.81934951590D+01	3.03600120489D-02	-1.61762396201D+01			
5.000	5.000	1.0	2.22669570738D-02	2.17262189626D+01	2.53170497141D-02	-2.02073517694D+01			
5.000	5.000	19.0	1.56591550129D-13	7.93097863019D+11	6.46100173074D-13	-3.11370142920D+12			
5.000	5.000	20.0	1.88342640121D-14	6.29076034991D+12	8.13652532644D-14	-2.59182249219D+13	8	45	1.D-14
4.000	6.000	0.0	2.94449215620D-01	3.01378301329D+00	2.09173395936D-01	-1.25521341389D+00	17	35	2.D-16
4.000	7.000	0.0	5.51582922742D-01	2.11640513364D+00	3.01934534735D-01	-6.54407580471D-01	18	31	6.D-17
4.000	8.000	0.0	8.82452654489D-01	1.55260815154D+00	3.46666242351D-01	-5.23272454238D-01	19	27	2.D-16
1.000	0.600	0.0	1.07096894447D-01	2.79244038685D+00	2.51925443598D-01	-2.76864439770D+00	7	166	4.D-16
1.000	3.000	0.0	1.08405268420D+00	6.27039514889D-01	3.01916785821D-01	-7.47829194001D-01	13	38	8.D-17
1.000	6.000	0.0	-1.67183367846D-01	-1.09083851821D+00	-8.92504992651D-01	1.58037110148D-01	19	20	6.D-17
1.995	1.200	0.0	4.34422920587D-02	7.46842788220D+00	8.07579506900D-02	-9.13545418683D+00	9	103	3.D-15
3.981	2.400	0.0	5.62399799825D-03	5.82540355738D+01	9.43642344138D-03	-8.00658629852D+01	11	70	2.D-13
18.564	116.790	0.0	-2.19713365418D-01	-1.07820215706D+00	-8.90272795108D-01	1.02537606990D-01			
18.564	116.790	30.0	1.06071645759D+00	3.88736687223D-01	3.02917093766D-01	-8.31744436652D-01			
18.564	116.790	60.0	-1.23263406427D+00	-1.86707561606D-01	-1.14689834856D-01	7.93898678648D-01			
18.564	116.790	80.0	-8.25210276064D-01	1.22923127192D+00	5.71410186401D-01	3.60641085568D-01			
18.564	116.790	100.0	5.02651913712D-01	4.21053181928D+00	1.44949393217D-01	-7.75260089628D-01			
18.564	116.790	120.0	6.10641818529D-05	1.32512621382D+04	3.85720485185D-05	-8.00586627116D+03			
18.564	116.790	140.0	5.59992983995D-11	1.02087306246D+10	4.94812366777D-11	-8.83688542406D+09			
18.564	116.790	160.0	1.47286871729D-18	3.09072708542D+17	1.62844530136D-18	-3.37227611779D+17			
18.564	116.790	180.0	2.14722540027D-27	1.78257094266D+26	2.81866160496D-27	-2.31719302741D+26			
18.564	116.790	200.0	2.64572896764D-37	1.25568168697D+36	3.99743384864D-37	-1.88246626253D+36	15	8	6.D-17
-4.000	5.000	0.0	4.07862723006D-01	6.74327035383D-01	1.09821233636D+00	-6.36110427280D-01			
-4.000	5.000	1.0	-6.42256329143D-01	-4.67204210149D-01	-7.58912963040D-01	1.00494559453D+00			
-4.000	5.000	2.0	7.98957414374D-01	-1.15121213166D-01	-1.57796947944D-01	-1.22889431433D+00			
-4.000	5.000	3.0	-2.56863093558D-01	7.87989922393D-01	1.14322942201D+00	3.85990587811D-01	17	21	5.D-17
			CYLINDRICAL BESSEL FUNCTIONS *** ABRAMOWITZ & STEGUN PAGES 407 - 408						
0.0	1.000	0.0	7.65197686558D-01	8.82569642157D-02	-4.40050585745D-01	7.81212821300D-01			
0.0	1.000	10.0	2.63061512369D-10	-1.21618014279D+08	2.61863505622D-09	1.20939993785D+09			
0.0	1.000	30.0	3.48286979425D-42	-3.04812878323D+39	1.04429904344D-40	9.13912933615D+40			
0.0	1.000	40.0	1.10791585113D-60	-7.18487479680D+57	4.43031209141D-59	2.87302862548D+59	4	85	5.D-17
0.0	2.000	0.0	2.23890779141D-01	5.10375672650D-01	-5.76724807757D-01	1.07032431541D-01			

```
 0.0      2.000    10.0    2.51538628272D-07 -1.29184542208D+05  1.23465029377D-06  6.31362881664D+05
 0.0      2.000    30.0    3.65025626647D-33 -2.91322384822D+30  5.46359748625D-32  4.35977776972D+31
 0.0      2.000    50.0    3.22409583944D-65 -1.97615057652D+62  8.05391545651D-64  4.93634176475D+63    4   45  5.D-17
 0.0      5.000     0.0   -1.77596771314D-01 -3.08517625249D-01  3.27579137591D-01 -1.47863143391D-01
 0.0      5.000    10.0    1.46780264731D-03 -2.51291100956D+01  2.58467784485D-03  4.24943370028D+01
 0.0      5.000    30.0    2.67117727825D-21 -4.02856841855D+18  1.58102720945D-20  2.38214034474D+19
 0.0      5.000    50.0    2.29424761595D-45 -2.78803701758D+42  2.28297467652D-44  2.77457024037D+43    6   19  5.D-17
 0.0     10.000     0.0   -2.45935764451D-01  5.56711672836D-02 -4.34727461689D-02 -2.49015424207D-01
 0.0     10.000    10.0    2.07486106633D-01 -3.59814152183D-01  8.43695786318D-02  1.60514886378D-01
 0.0     10.000    30.0    1.55109607826D-12 -7.25614231610D+09  4.39647875200D-12  2.04761666074D+10
 0.0     10.000    50.0    1.78451360787D-30 -3.64106650180D+27  8.74593525529D-30  1.78297577623D+28    7   10  5.D-17
 0.0     50.000     0.0    5.58123276693D-02 -9.80649954701D-02  9.75118281252D-02  5.67956685620D-02
 0.0     50.000    10.0   -1.13847849149D-01  5.72389718205D-03 -4.42289121408D-03 -1.11614574783D-01
 0.0     50.000    30.0    4.84342572455D-02 -1.16457234935D-01  9.24533752861D-02  4.05814213506D-02
 0.0     50.000   100.0    1.11592736908D-21 -3.29380018820D+18  1.93650320925D-21  5.69386591665D+18   13   4  5.D-17
 0.0    100.000     0.0    1.99858503042D-02 -7.72443133651D-02  7.71453520141D-02  2.03723120028D-02
 0.0    100.000    10.0   -5.47321769355D-02  5.83315742364D-02 -5.77635437125D-02 -5.47531535058D-02
 0.0    100.000    30.0    8.14601295812D-02  6.13883921201D-03 -6.30376589012D-03  7.76760355155D-02
 0.0    100.000   100.0    9.63666732959D-02 -1.66921411418D-01  1.88772520272D-02  3.33640257742D-02   34   3  5.D-17

                          SPHERICAL BESSEL FUNCTIONS *** ABRAMOWITZ & STEGUN  PAGES 465 - 466

 0.0      1.000     0.0    8.41470984808D-01 -5.40302305868D-01 -3.01168678940D-01  1.38177329068D+00
 0.0      1.000    10.0    7.11655264005D-11 -6.72215008256D+08  7.08555712150D-10  7.35887504239D+09
 0.0      1.000    30.0    5.56683126698D-43 -2.94642854750D+40  1.66916554020D-41  9.12893306467D+41
 0.0      1.000    40.0    1.53821037424D-61 -8.02845085085D+58  6.15098796870D-60  3.29064842213D+60    4   1  5.D-17
 0.0      2.000     0.0    4.54648713413D-01  2.08073418274D-01 -4.35397774980D-01  3.50612004276D-01
 0.0      2.000    10.0    6.82530086497D-08 -3.55414720085D+05  3.35288171637D-07  1.91689203038D+06
 0.0      2.000    30.0    5.83661788752D-34 -1.40739387104D+31  8.73637975213D-33  2.17668398491D+32
 0.0      2.000    50.0    4.01157529034D-66 -1.23502194437D+63  1.00211458762D-64  3.14680992417D+64    4   1  5.D-17
 0.0      5.000     0.0   -1.91784854933D-01 -5.67324370926D-02  9.50894080792D-02 -1.80438367514D-01
 0.0      5.000    10.0    4.07344244249D-04 -2.66561144057D+01  7.21942378124D-04  5.09540067582D+01
 0.0      5.000    30.0    4.28273021730D-22 -7.76071756976D+18  2.53543820515D-21  4.74537949931D+19
 0.0      5.000    50.0    2.85747935044D-46 -6.96410918827D+42  2.84357588031D-45  7.06812704746D+43    6   1  5.D-17
 0.0     10.000     0.0   -5.44021110889D-02  8.39071529076D-02 -7.84669417988D-02 -6.27928263797D-02
 0.0     10.000    10.0    6.46051544926D-02 -1.72453672088D-01  2.90307396067D-02  7.72932499058D-02
 0.0     10.000    30.0    2.51205738500D-13 -6.90831864609D+09  7.12720577762D-13  2.02077355936D+10
 0.0     10.000    50.0    2.23069602322D-31 -4.52822727235D+27  1.09348679947D-30  2.26316952204D+28    7   1  5.D-17
 0.0     50.000     0.0   -5.24749707408D-03 -1.92993205698D-02  1.94042705113D-02 -4.86151066268D-03
 0.0     50.000    10.0   -1.50392214635D-02  1.35246875112D-02 -1.29162985287D-02 -1.49815666420D-02
 0.0     50.000    30.0   -1.49467345361D-03 -2.24122681205D-02  1.78070439540D-02 -6.04651449647D-04
 0.0     50.000   100.0    1.01901226293D-22 -1.12569289133D+18  1.76984656820D-22  1.97023762370D+18   13   1  5.D-17
 0.0    100.000     0.0   -5.06365641110D-03 -8.62318872288D-03  8.67302528699D-03 -4.97742452387D-03
 0.0    100.000    10.0   -1.95657859714D-04  1.00257773736D-02 -9.96852414545D-03 -2.95394082628D-04
 0.0    100.000    30.0    8.70062851445D-03 -5.41292934887D-03  5.06366936035D-03  8.34315766796D-03
 0.0    100.000   100.0    1.08804770114D-02 -2.29838504916D-02  2.28730043501D-03  4.35909461714D-03   33   1  5.D-17

                          BESSELS WITH ORDER 1/3  ***  WATSON TABLE III   PAGES 714 - 729

 0.0      0.240     0.3    5.46408741152D-01 -1.37617973522D+00  7.09495139391D-01  3.06764776175D+00    4  319  5.D-17
 0.0      0.500     0.3    6.72830829498D-01 -8.40627826043D-01  3.19790290150D-01  1.49281941953D+00    6  160  5.D-17
 0.0      1.000     0.3    7.30876402169D-01 -2.78801641276D-01 -5.52851752674D-02  8.92125355306D-01    8   83  5.D-17
 0.0      2.000     0.3    4.42939818149D-01  3.43199966260D-01 -4.56138918066D-01  3.65203168163D-01   10   43  5.D-17
 0.0      4.000     0.3   -3.55427373455D-01  1.79416766344D-01 -1.36039481820D-01 -3.79113116300D-01   15   22  5.D-17
 0.0      5.000     0.3   -3.06420663800D-01 -1.81923211293D-01  2.12890980263D-01 -2.89126067463D-01   16   18  5.D-17
 0.0      8.000     0.3    2.59776161108D-01  1.09587794634D-01 -1.25906384515D-01  2.53216724183D-01   22   12  5.D-17
 0.0     10.000     0.3   -1.86145167049D-01  1.70201117883D-01 -1.61022875224D-01 -1.94771126449D-01   25   10  5.D-17
 0.0     16.000     0.3   -1.04162684107D-01  1.70082756218D-01 -1.66875284009D-01 -1.09503010813D-01   33    7  5.D-17

                          BLANCH TEST OF RECURRENCE THROUGH THE ZERO OF J5 NEAR X = 8.7714838

 0.0      8.771     0.0   -3.16762512697D-02  2.67319051367D-01 -2.65945081557D-01 -4.69170892834D-02
 0.0      8.771     1.0    2.65945081557D-01  4.69170892834D-02 -6.19955322056D-02  2.61970231218D-01
 0.0      8.771     2.0    9.23148131415D-02 -2.56621411070D-01  2.44896233894D-01  1.05429746189D-01
 0.0      8.771     4.0   -2.45434210955D-01  1.44479105276D-01 -1.11923695074D-01 -2.29828225183D-01
 0.0      8.771     5.0    3.91711840794D-09  2.95714047272D-01 -2.45434213188D-01 -2.40864726892D-02
 0.0      8.771     6.0    2.45434215421D-01  1.92652050654D-01 -1.67888535873D-01  1.63933343997D-01
 0.0      8.771    10.0    1.07825846963D-01 -6.03285674072D-01  7.11703664600D-02  2.74908871645D-01
 0.0      8.771    14.0    2.92664873665D-03 -1.00251142551D+01  3.73421850293D-03  1.20077195602D+01
 0.0      8.771    18.0    1.99004280877D-05 -1.01853380518D+03  3.59862493115D-05  1.80524423005D+03
 0.0      8.771    20.0    1.11736283101D-06 -1.58571185507D+04  2.30378879561D-06  3.22606868408D+04   10   11  5.D-17


 0.0      1.000   100.0    8.45870866831D-04 -3.76330645768D+00  8.45828991019D-02  3.76308505766D+02    3   0  1.D+00
          **** IEXP =    185  F,FP *10**(-IEXP)  G,GP *10**(IEXP)

 0.0      2.000   100.0    1.06424628231D-03 -2.99155272347D+00  5.32017760004D-02  1.49546182065D+02    4   0  1.D+00
          **** IEXP =    155  F,FP *10**(-IEXP)  G,GP *10**(IEXP)


                          EXAMPLES OF ERROR CONDITIONS

CF2 HAS FAILED TO CONVERGE AFTER  20000. ITERATIONS
P,Q,DP,DQ,ACCUR =       2.1651719D+02     9.2513855D+02     1.1838276D-08    -2.3027605D-08    1.000D-16


IFAIL =          2 JMAX,KFN =    1    0       0.0010    -500.0000       0.0          0.0

FOR XX =    1.000D-08 TRY SMALL-X  SOLUTIONS OR X NEGATIVE
SQUARE ROOT ACCURACY PARAMETER =      1.000D-08

IFAIL =         -1 JMAX,KFN =    1    0       0.0000    -500.0000       0.0          0.0

FOR XX =   -2.000D+04 TRY SMALL-X  SOLUTIONS OR X NEGATIVE
SQUARE ROOT ACCURACY PARAMETER =      1.000D-08

IFAIL =         -1 JMAX,KFN =    1    0  -20000.0000       1.0000       0.0         -2.2000

FOR XX =   -5.500D+01 TRY SMALL-X  SOLUTIONS OR X NEGATIVE
SQUARE ROOT ACCURACY PARAMETER =      1.000D-08

IFAIL =         -1 JMAX,KFN =    1    0     -55.0000       0.0          0.0          1.0000
```

PROGRAM LISTING

```
//ARCOUCPC   JOB (30210,AR,'R=MANC'),BARNETT,MSGLEVEL=(1,1),NOTIFY=AR   ABNK0001
// EXEC FHXCLG,                                                         ABNK0002
// PARM.C='SOURCE,XREF,MAP',REGION.G=180K,TIME.G=(0,59)                 ABNK0003
//C.SYSIN  DD  *                                                       ABNK0004
      IMPLICIT REAL*8(A-H,O-Z)                                        ABNK0005
C *** MAIN PROGRAM TO TEST COULOMB WAVEFUNCTIONS  MANCHESTER  JUNE 1981 ABNK0006
      DIMENSION FC(1200),GC(1200),FCP(1200),GCP(1200),XX(21),XL(10)   ABNK0007
      DIMENSION NJ(21) ,TEXT(9)                                       ABNK0008
         COMMON /STEED/ PACCQ,NFP,NPQ,IEXP,M1                         ABNK0009
      DATA NJ / 40,3*50,2*100,15*0/ , ONE /1.0D0/                     ABNK0010
      WRITE(6,1)                                                      ABNK0011
      MODE = 1                                                        ABNK0012
   10 READ(5,2) ETA,N,(XL(I),I=1,10),XM,KFN                           ABNK0013
      IF(N .EQ. 0) STOP                                               ABNK0014
      READ(5,3) (XX(I),I=1,N)                                         ABNK0015
                IF(KFN .LT. 0) READ (5,4) (TEXT(I),I=1,9)             ABNK0016
                IF(KFN .LT. 0) WRITE(6,6) (TEXT(I),I=1,9)             ABNK0017
      DO 60 I = 1,N                                                   ABNK0018
         JMAX = 1                                                     ABNK0019
      DO 20 J = 2,10                                                  ABNK0020
   20 IF(XL(J)  .GT.  XM )     JMAX = J                               ABNK0021
      IF(XL(10) .LE. -ONE) XL(JMAX) = NJ(I)                           ABNK0022
         KFN  = IABS(KFN)                                             ABNK0023
      IF(KFN .EQ. 4) KFN   = 0                                        ABNK0024
      IF(KFN .NE. 3)                          GO TO 30                ABNK0025
                     JMAX  = 1                                        ABNK0026
                     XL(1) = ONE/XM                                   ABNK0027
                     XM    = XL(1)                                    ABNK0028
                     KFN   = 2                                        ABNK0029
   30 CALL COULFG(XX(I),ETA,XM,XL(JMAX),FC,GC,FCP,GCP,MODE,KFN,IFAIL) ABNK0030
      IF(IFAIL .NE. 0) WRITE(6,7) IFAIL,JMAX,KFN,XX(I),ETA,XM,XL(JMAX) ABNK0031
      IF(IFAIL .NE. 0)                        GO TO 10                ABNK0032
      IF(JMAX  .EQ. 1)                        GO TO 50                ABNK0033
      DO 40 KK = 2,JMAX                                               ABNK0034
            K = M1 + IDINT(XL(KK-1) - XM)                             ABNK0035
   40 WRITE(6,8) ETA,XX(I),XL(KK-1),FC(K),GC(K),FCP(K),GCP(K)         ABNK0036
   50       K = M1 + IDINT(XL(JMAX) - XM)                             ABNK0037
      WRITE(6,8) ETA,XX(I),XL(JMAX),FC(K),GC(K),FCP(K),GCP(K),        ABNK0038
     *           NFP,NPQ,PACCQ                                        ABNK0039
   60 IF(IEXP .GT. 1) WRITE(6,9) IEXP                                 ABNK0040
                                          GO TO 10                    ABNK0041
    1 FORMAT(1H1,35X,61H TEST OF THE MANCHESTER COULOMB WAVEFUNCTION PROABNK0042
     *GRAM - COULFG//,5X,4H ETA,6X,4H X  ,6X,2HXL,10X,2H F,18X,2H G,   ABNK0043
     *18X,2HFP,19X,2HGP,11X,16HNFP   NPQ  PACCQ/)                      ABNK0044
    2 FORMAT(F10.3,I5,11F5.0,I2)                                      ABNK0045
    3 FORMAT(7F10.3)                                                  ABNK0046
    4 FORMAT(9A8)                                                     ABNK0047
    5 FORMAT(/)                                                       ABNK0048
    6 FORMAT(/28X,9A8/)                                               ABNK0049
    7 FORMAT(1X,'IFAIL =',I10,' JMAX,KFN = ',2I4,4F12.4)              ABNK0050
    8 FORMAT(1X,F9.3,F10.3,F8.1,1P4D20.11,2I6,D9.0)                   ABNK0051
    9 FORMAT(12X,13H **** IEXP = ,I6,31H  F,FP *10**(-IEXP)  G,GP *10** ABNK0052
     *,6H(IEXP)/)                                                     ABNK0053
      END                                                             ABNK0054
      SUBROUTINE COULFG(XX,ETA1,XLMIN,XLMAX, FC,GC,FCP,GCP,           ABNK0055
     *                  MODE1,KFN,IFAIL)                              ABNK0056
C                                                                     ABNK0057
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCABNK0058
C                                                                   CABNK0059
C  REVISED COULOMB WAVEFUNCTION PROGRAM USING STEED'S METHOD        CABNK0060
C                                                                   CABNK0061
C  A. R. BARNETT           MANCHESTER  MARCH   1981                 CABNK0062
C                                                                   CABNK0063
C  ORIGINAL PROGRAM 'RCWFN'       IN    CPC  8 (1974) 377-395       CABNK0064
C                 + 'RCWFF'       IN    CPC 11 (1976) 141-142       CABNK0065
C  FULL DESCRIPTION OF ALGORITHM IN    CPC 21 (1981) 297-314       CABNK0066
C  THIS VERSION WRITTEN UP       IN    CPC XX (1982) YYY-ZZZ       CABNK0067
C                                                                   CABNK0068
```

```
C  COULFG RETURNS F,G,F',G', FOR REAL XX.GT.0,REAL ETA1 (INCLUDING 0),  CABNK0069
C   AND REAL LAMBDA(XLMIN) .GT. -1 FOR INTEGER-SPACED LAMBDA VALUES     CABNK0070
C   THUS GIVING POSITIVE-ENERGY SOLUTIONS TO THE COULOMB SCHRODINGER    CABNK0071
C   EQUATION,TO THE KLEIN-GORDON EQUATION AND TO SUITABLE FORMS OF      CABNK0072
C   THE DIRAC EQUATION ,ALSO SPHERICAL & CYLINDRICAL BESSEL EQUATIONS   CABNK0073
C                                                                       CABNK0074
C  FOR A RANGE OF LAMBDA VALUES (XLMAX - XLMIN) MUST BE AN INTEGER,     CABNK0075
C  STARTING ARRAY ELEMENT IS M1 = MAX0(IDINT(XLMIN+ACCUR),0) + 1        CABNK0076
C      SEE TEXT FOR MODIFICATIONS FOR INTEGER L-VALUES                  CABNK0077
C                                                                       CABNK0078
C  IF 'MODE' = 1  GET F,G,F',G'   FOR INTEGER-SPACED LAMBDA VALUES      CABNK0079
C            = 2       F,G       UNUSED ARRAYS MUST BE DIMENSIONED IN   CABNK0080
C            = 3       F               CALL TO AT LEAST LENGTH (1)      CABNK0081
C  IF 'KFN'  = 0 REAL         COULOMB FUNCTIONS ARE RETURNED            CABNK0082
C            = 1 SPHERICAL    BESSEL       "       "      "             CABNK0083
C            = 2 CYLINDRICAL BESSEL       "       "      "             CABNK0084
C  THE USE OF 'MODE' AND 'KFN' IS INDEPENDENT                           CABNK0085
C                                                                       CABNK0086
C  PRECISION:  RESULTS TO WITHIN 2-3 DECIMALS OF 'MACHINE ACCURACY'     CABNK0087
C   IN OSCILLATING REGION X .GE. ETA1 + SQRT(ETA1**2 + XLM(XLM+1))      CABNK0088
C   COULFG IS CODED FOR REAL*8 ON IBM OR EQUIVALENT  ACCUR = 10**-16    CABNK0089
C   USE AUTODBL + EXTENDED PRECISION ON HX COMPILER  ACCUR = 10**-33    CABNK0090
C   FOR MANTISSAS OF 56 & 112 BITS. FOR SINGLE PRECISION CDC (48 BITS)  CABNK0091
C   REASSIGN DSQRT=SQRT ETC.  SEE TEXT FOR COMPLEX ARITHMETIC VERSION   CABNK0092
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCABNK0093
C                                                                       ABNK0094
      IMPLICIT REAL*8 (A-H,O-Z)                                         ABNK0095
      DIMENSION     FC(1),GC(1),FCP(1),GCP(1)                           ABNK0096
      LOGICAL       ETANE0,XLTURN                                       ABNK0097
      COMMON        /STEED/ PACCQ,NFP,NPQ,IEXP,M1                       ABNK0098
C***  COMMON BLOCK IS FOR INFORMATION ONLY.  NOT REQUIRED IN CODE       ABNK0099
C***  COULFG HAS CALLS TO: DSQRT,DABS,DMOD,IDINT,DSIGN,DFLOAT,DMIN1     ABNK0100
      DATA ZERO,ONE,TWO,TEN2,ABORT /0.0D0, 1.0D0, 2.0D0, 1.0D2, 2.0D4/  ABNK0101
      DATA HALF,TM30 / 0.5D0, 1.0D-30 /                                 ABNK0102
      DATA RT2DPI /0.79788 45608 02865 35587 98921 19868 76373 D0/      ABNK0103
C *** THIS CONSTANT IS  DSQRT(TWO/PI):  USE Q0 FOR IBM REAL*16: D0 FOR  ABNK0104
C ***  REAL*8 & CDC DOUBLE P:  E0 FOR CDC SINGLE P; AND TRUNCATE VALUE.  ABNK0105
C                                                                       ABNK0106
                        ACCUR = 1.0D-16                                 ABNK0107
C ***             CHANGE ACCUR TO SUIT MACHINE AND PRECISION REQUIRED    ABNK0108
      MODE  = 1                                                         ABNK0109
      IF(MODE1 .EQ. 2 .OR. MODE1 .EQ. 3 ) MODE = MODE1                  ABNK0110
      IFAIL = 0                                                         ABNK0111
      IEXP  = 1                                                         ABNK0112
      NPQ   = 0                                                         ABNK0113
      ETA   = ETA1                                                      ABNK0114
      GJWKB = ZERO                                                      ABNK0115
      PACCQ = ONE                                                       ABNK0116
      IF(KFN .NE. 0) ETA = ZERO                                         ABNK0117
                ETANE0  = ETA .NE. ZERO                                 ABNK0118
      ACC   = ACCUR                                                     ABNK0119
      ACC4  = ACC*TEN2*TEN2                                             ABNK0120
      ACCH  = DSQRT(ACC)                                                ABNK0121
C ***    TEST RANGE OF XX, EXIT IF.LE.DSQRT(ACCUR) OR IF NEGATIVE       ABNK0122
C                                                                       ABNK0123
      IF(XX .LE. ACCH)                              GO TO 100           ABNK0124
      X     = XX                                                        ABNK0125
      XLM   = XLMIN                                                     ABNK0126
      IF(KFN .EQ. 2)  XLM = XLM - HALF                                  ABNK0127
      IF(XLM .LE. -ONE .OR. XLMAX .LT. XLMIN)   GO TO 105               ABNK0128
      E2MM1 = ETA*ETA + XLM*XLM + XLM                                   ABNK0129
      XLTURN= X*(X - TWO*ETA) .LT. XLM*XLM + XLM                        ABNK0130
      DELL  = XLMAX - XLMIN + ACC                                       ABNK0131
      IF(DABS(DMOD(DELL,ONE)) .GT. ACC) WRITE(6,2040)XLMAX,XLMIN,DELL   ABNK0132
      LXTRA = IDINT(DELL)                                               ABNK0133
      XLL   = XLM + DFLOAT(LXTRA)                                       ABNK0134
C ***       LXTRA IS NUMBER OF ADDITIONAL LAMBDA VALUES TO BE COMPUTED   ABNK0135
C ***       XLL  IS MAX LAMBDA VALUE, OR 0.5 SMALLER FOR J,Y BESSELS     ABNK0136
C ***         DETERMINE STARTING ARRAY ELEMENT (M1) FROM XLMIN           ABNK0137
      M1   = MAX0(IDINT(XLMIN + ACC),0) + 1                             ABNK0138
      L1   = M1 + LXTRA                                                 ABNK0139
C                                                                       ABNK0140
```

```
C ***     EVALUATE CF1  =  F   =  FPRIME(XL,ETA,X)/F(XL,ETA,X)          ABNK0141
C                                                                       ABNK0142
      XI  = ONE/X                                                       ABNK0143
      FCL = ONE                                                         ABNK0144
      PK  = XLL + ONE                                                   ABNK0145
      PX  = PK  + ABORT                                                 ABNK0146
    2 EK  = ETA / PK                                                    ABNK0147
      F   = (EK + PK*XI)*FCL + (FCL - ONE)*XI                           ABNK0148
      PK1 =  PK + ONE                                                   ABNK0149
C ***   TEST ENSURES B1.NE. ZERO FOR NEGATIVE ETA; FIXUP IS EXACT.      ABNK0150
            IF(DABS(ETA*X + PK*PK1) .GT. ACC)  GO TO 3                  ABNK0151
            FCL  = (ONE + EK*EK)/(ONE + (ETA/PK1)**2)                   ABNK0152
            PK   =  TWO + PK                                            ABNK0153
      GO TO 2                                                           ABNK0154
    3 D   =  ONE/((PK + PK1)*(XI + EK/PK1))                             ABNK0155
      DF  = -FCL*(ONE + EK*EK)*D                                        ABNK0156
          IF(FCL .NE. ONE )  FCL = -ONE                                 ABNK0157
          IF(D   .LT. ZERO)  FCL = -FCL                                 ABNK0158
      F   =  F  + DF                                                    ABNK0159
C                                                                       ABNK0160
C ***   BEGIN CF1 LOOP ON PK = K = LAMBDA + 1                           ABNK0161
C                                                                       ABNK0162
      P     = ONE                                                       ABNK0163
    4 PK    = PK1                                                       ABNK0164
        PK1 = PK1 + ONE                                                 ABNK0165
        EK  = ETA / PK                                                  ABNK0166
        TK  = (PK + PK1)*(XI + EK/PK1)                                  ABNK0167
        D   =  TK - D*(ONE + EK*EK)                                     ABNK0168
          IF(DABS(D) .GT. ACCH)              GO TO 5                    ABNK0169
          WRITE (6,1000) D,DF,ACCH,PK,EK,ETA,X                          ABNK0170
          P = P  +   ONE                                                ABNK0171
          IF( P .GT. TWO )                   GO TO 110                  ABNK0172
    5 D     = ONE/D                                                     ABNK0173
          IF (D .LT. ZERO) FCL = -FCL                                   ABNK0174
        DF  = DF*(D*TK - ONE)                                           ABNK0175
        F   = F  + DF                                                   ABNK0176
          IF(PK .GT. PX)                     GO TO 110                  ABNK0177
      IF(DABS(DF) .GE. DABS(F)*ACC)          GO TO 4                    ABNK0178
                NFP = PK - XLL - 1                                      ABNK0179
      IF(LXTRA .EQ. 0)                       GO TO 7                    ABNK0180
C                                                                       ABNK0181
C *** DOWNWARD RECURRENCE TO LAMBDA = XLM. ARRAY GC,IF PRESENT,STORES RLABNK0182
C                                                                       ABNK0183
      FCL = FCL*TM30                                                    ABNK0184
      FPL = FCL*F                                                       ABNK0185
      IF(MODE .EQ. 1) FCP(L1) = FPL                                     ABNK0186
                      FC (L1) = FCL                                     ABNK0187
      XL  = XLL                                                         ABNK0188
      RL  = ONE                                                         ABNK0189
      EL  = ZERO                                                        ABNK0190
      DO 6  LP = 1,LXTRA                                                ABNK0191
         IF(ETANE0) EL = ETA/XL                                         ABNK0192
         IF(ETANE0) RL = DSQRT(ONE + EL*EL)                             ABNK0193
         SL    =  EL  + XL*XI                                           ABNK0194
         L     =  L1  - LP                                              ABNK0195
         FCL1  = (FCL *SL + FPL)/RL                                     ABNK0196
         FPL   =  FCL1*SL - FCL *RL                                     ABNK0197
         FCL   =  FCL1                                                  ABNK0198
         FC(L) =  FCL                                                   ABNK0199
         IF(MODE .EQ. 1) FCP(L)  = FPL                                  ABNK0200
         IF(MODE .NE. 3 .AND. ETANE0) GC(L+1) = RL                      ABNK0201
    6 XL = XL - ONE                                                     ABNK0202
      IF(FCL .EQ. ZERO) FCL = ACC                                       ABNK0203
      F  = FPL/FCL                                                      ABNK0204
C ***   NOW WE HAVE REACHED LAMBDA = XLMIN = XLM                        ABNK0205
C ***   EVALUATE CF2 = P + I.Q  AGAIN USING STEED'S ALGORITHM           ABNK0206
C ***   SEE TEXT FOR COMPACT COMPLEX CODE FOR SP CDC OR NON-ANSI IBM    ABNK0207
C                                                                       ABNK0208
    7 IF( XLTURN ) CALL JWKB(X,ETA,DMAX1(XLM,ZERO),FJWKB,GJWKB,IEXP)    ABNK0209
      IF( IEXP .GT. 1 .OR. GJWKB .GT. ONE/(ACCH*TEN2))  GO TO 9         ABNK0210
          XLTURN = .FALSE.                                              ABNK0211
      TA =  TWO*ABORT                                                   ABNK0212
      PK =  ZERO                                                        ABNK0213
      WI =  ETA + ETA                                                   ABNK0214
```

```
      P  =  ZERO                                                          ABNK0215
      Q  =  ONE - ETA*XI                                                  ABNK0216
      AR = -E2MM1                                                         ABNK0217
      AI =  ETA                                                           ABNK0218
      BR =  TWO*(X - ETA)                                                 ABNK0219
      BI =  TWO                                                           ABNK0220
      DR =  BR/(BR*BR + BI*BI)                                            ABNK0221
      DI = -BI/(BR*BR + BI*BI)                                            ABNK0222
      DP = -XI*(AR*DI + AI*DR)                                            ABNK0223
      DQ =  XI*(AR*DR - AI*DI)                                            ABNK0224
    8 P  = P  + DP                                                        ABNK0225
      Q  = Q  + DQ                                                        ABNK0226
      PK = PK + TWO                                                       ABNK0227
      AR = AR + PK                                                        ABNK0228
      AI = AI + WI                                                        ABNK0229
      BI = BI + TWO                                                       ABNK0230
      D  = AR*DR - AI*DI + BR                                             ABNK0231
      DI = AI*DR + AR*DI + BI                                             ABNK0232
      C  = ONE/(D*D + DI*DI)                                              ABNK0233
      DR =  C*D                                                           ABNK0234
      DI = -C*DI                                                          ABNK0235
      A  = BR*DR - BI*DI - ONE                                            ABNK0236
      B  = BI*DR + BR*DI                                                  ABNK0237
      C  = DP*A  - DQ*B                                                   ABNK0238
      DQ = DP*B  + DQ*A                                                   ABNK0239
      DP = C                                                              ABNK0240
      IF(PK .GT. TA)                          GO TO 120                   ABNK0241
    IF(DABS(DP)+DABS(DQ).GE.(DABS(P)+DABS(Q))*ACC)   GO TO 8              ABNK0242
                  NPQ  = PK/TWO                                           ABNK0243
                  PACCQ = HALF*ACC/DMIN1(DABS(Q),ONE)                     ABNK0244
                  IF(DABS(P) .GT. DABS(Q)) PACCQ = PACCQ*DABS(P)          ABNK0245
C                                                                         ABNK0246
C *** SOLVE FOR FCM = F AT LAMBDA = XLM,THEN FIND NORM FACTOR W=W/FCM      ABNK0247
C                                                                         ABNK0248
      GAM = (F - P)/Q                                                     ABNK0249
            IF(Q .LE. ACC4*DABS(P))           GO TO 130                   ABNK0250
      W   = ONE/DSQRT((F - P)*GAM + Q)                                    ABNK0251
            GO TO 10                                                      ABNK0252
C *** ARRIVE HERE IF G(XLM) .GT. 10**6 OR IEXP .GT. 70 & XLTURN = .TRUE.  ABNK0253
    9 W   = FJWKB                                                         ABNK0254
      GAM = GJWKB*W                                                       ABNK0255
      P   = F                                                             ABNK0256
      Q   = ONE                                                           ABNK0257
C                                                                         ABNK0258
C *** NORMALISE FOR SPHERICAL OR CYLINDRICAL BESSEL FUNCTIONS             ABNK0259
C                                                                         ABNK0260
   10                     ALPHA = ZERO                                    ABNK0261
         IF(KFN  .EQ. 1) ALPHA = XI                                       ABNK0262
         IF(KFN  .EQ. 2) ALPHA = XI*HALF                                  ABNK0263
                         BETA  = ONE                                      ABNK0264
         IF(KFN  .EQ. 1) BETA  = XI                                       ABNK0265
         IF(KFN  .EQ. 2) BETA  = DSQRT(XI)*RT2DPI                         ABNK0266
      FCM  = DSIGN(W,FCL)*BETA                                            ABNK0267
          FC(M1)  = FCM                                                   ABNK0268
                    IF(MODE .EQ. 3)           GO TO 11                    ABNK0269
          IF(.NOT. XLTURN)   GCL =  FCM*GAM                               ABNK0270
          IF(      XLTURN)   GCL =  GJWKB*BETA                            ABNK0271
          IF( KFN .NE. 0 )   GCL = -GCL                                   ABNK0272
          GC(M1)  = GCL                                                   ABNK0273
          GPL =  GCL*(P - Q/GAM) - ALPHA*GCL                              ABNK0274
                    IF(MODE .EQ. 2)           GO TO 11                    ABNK0275
          GCP(M1) = GPL                                                   ABNK0276
          FCP(M1) = FCM*(F - ALPHA)                                       ABNK0277
   11 IF(LXTRA .EQ. 0 ) RETURN                                            ABNK0278
C *** UPWARD RECURRENCE FROM GC(M1),GCP(M1)  STORED VALUE IS RL           ABNK0279
C *** RENORMALISE FC,FCP AT EACH LAMBDA AND CORRECT REGULAR DERIVATIVE    ABNK0280
C ***    XL   = XLM HERE  AND RL = ONE , EL = ZERO FOR BESSELS            ABNK0281
      W    = BETA*W/DABS(FCL)                                             ABNK0282
      MAXL = L1 - 1                                                       ABNK0283
      DO 12 L = M1,MAXL                                                   ABNK0284
                    IF(MODE .EQ. 3)           GO TO 12                    ABNK0285
```

```
                    XL = XL + ONE                                          ABNK0286
          IF(ETANE0)   EL = ETA/XL                                         ABNK0287
          IF(ETANE0)   RL = GC(L+1)                                        ABNK0288
                       SL = EL + XL*XI                                     ABNK0289
          GCL1      = ((SL - ALPHA)*GCL - GPL)/RL                          ABNK0290
          GPL       =   RL*GCL -  (SL + ALPHA)*GCL1                        ABNK0291
          GCL       = GCL1                                                 ABNK0292
          GC(L+1)   = GCL1                                                 ABNK0293
                       IF(MODE .EQ. 2)          GO TO 12                   ABNK0294
          GCP(L+1) = GPL                                                   ABNK0295
          FCP(L+1) = W*(FCP(L+1) - ALPHA*FC(L+1))                          ABNK0296
   12 FC(L+1)      = W* FC(L+1)                                            ABNK0297
      RETURN                                                               ABNK0298
 1000 FORMAT(/' CF1 ACCURACY LOSS: D,DF,ACCH,K,ETA/K,ETA,X = ',1P7D9.2/)ABNK0299
C                                                                          ABNK0300
C ***    ERROR MESSAGES                                                    ABNK0301
C                                                                          ABNK0302
  100 IFAIL = -1                                                           ABNK0303
      WRITE(6,2000) XX,ACCH                                                ABNK0304
 2000 FORMAT(' FOR XX = ',1PD12.3,' TRY SMALL-X  SOLUTIONS',               ABNK0305
     *' OR X NEGATIVE'/ ,' SQUARE ROOT ACCURACY PARAMETER =  ',D12.3/)  ABNK0306
      RETURN                                                               ABNK0307
  105 IFAIL = -2                                                           ABNK0308
      WRITE (6,2005) XLMAX,XLMIN,XLM                                       ABNK0309
 2005 FORMAT(/' PROBLEM WITH INPUT ORDER VALUES:XLMAX,XLMIN,XLM = ',      ABNK0310
     *1P3D15.6/)                                                          ABNK0311
      RETURN                                                               ABNK0312
  110 IFAIL =  1                                                           ABNK0313
      WRITE (6,2010) ABORT,F ,DF,PK,PX,ACC                                 ABNK0314
 2010 FORMAT(' CF1 HAS FAILED TO CONVERGE AFTER ',F10.0,' ITERATIONS',/ ABNK0315
     *' F,DF,PK,PX,ACCUR =  ',1P5D12.3//)                                 ABNK0316
      RETURN                                                               ABNK0317
  120 IFAIL =  2                                                           ABNK0318
      WRITE (6,2020) ABORT,P,Q,DP,DQ,ACC                                   ABNK0319
 2020 FORMAT(' CF2 HAS FAILED TO CONVERGE AFTER ',F7.0,' ITERATIONS',/  ABNK0320
     *' P,Q,DP,DQ,ACCUR =  ',1P4D17.7,D12.3//)                            ABNK0321
      RETURN                                                               ABNK0322
  130 IFAIL =  3                                                           ABNK0323
      WRITE (6,2030) P,Q,ACC,DELL,LXTRA,M1                                 ABNK0324
 2030 FORMAT(' FINAL Q.LE.DABS(P)*ACC*10**4 , P,Q,ACC = ',1P3D12.3,4X,  ABNK0325
     *' DELL,LXTRA,M1 = ',D12.3,2I5 /)                                    ABNK0326
      RETURN                                                               ABNK0327
 2040 FORMAT(' XLMAX - XLMIN = DELL NOT AN INTEGER ',1P3D20.10/)           ABNK0328
      END                                                                  ABNK0329
C                                                                          ABNK0330
      SUBROUTINE JWKB(XX,ETA1,XL,FJWKB,GJWKB,IEXP)                         ABNK0331
      REAL*8          XX,ETA1,XL,FJWKB,GJWKB,DZERO                         ABNK0332
C *** COMPUTES JWKB APPROXIMATIONS TO COULOMB FUNCTIONS    FOR XL.GE. 0 ABNK0333
C *** AS MODIFIED BY BIEDENHARN ET AL. PHYS REV 97 (1955) 542-554         ABNK0334
C *** CALLS DMAX1,SQRT,ALOG,EXP,ATAN2,FLOAT,INT        BARNETT FEB 1981 ABNK0335
      DATA   ZERO,HALF,ONE,SIX,TEN/ 0.0E0, 0.5E0, 1.0E0, 6.0E0, 10.0E0 /ABNK0336
      DATA  DZERO, RL35, ALOGE  /0.0D0, 35.0E0, 0.43429 45 E0 /            ABNK0337
      X     = XX                                                           ABNK0338
      ETA   = ETA1                                                         ABNK0339
      GH2   = X*(ETA + ETA - X)                                            ABNK0340
      XLL1  = DMAX1(XL*XL + XL,DZERO)                                      ABNK0341
      IF(GH2 + XLL1 .LE. ZERO) RETURN                                      ABNK0342
       HLL  = XLL1 + SIX/RL35                                              ABNK0343
       HL   = SQRT(HLL)                                                    ABNK0344
       SL   = ETA/HL + HL/X                                                ABNK0345
       RL2  = ONE + ETA*ETA/HLL                                            ABNK0346
       GH   = SQRT(GH2 + HLL)/X                                            ABNK0347
       PHI  = X*GH - HALF*( HL*ALOG((GH + SL)**2/RL2) - ALOG(GH) )         ABNK0348
          IF(ETA .NE. ZERO) PHI = PHI - ETA*ATAN2(X*GH,X - ETA)            ABNK0349
      PHI10 = -PHI*ALOGE                                                   ABNK0350
      IEXP  =  INT(PHI10)                                                  ABNK0351
      IF(IEXP .GT. 70) GJWKB = TEN**(PHI10 - FLOAT(IEXP))                  ABNK0352
      IF(IEXP .LE. 70) GJWKB = EXP(-PHI)                                   ABNK0353
      IF(IEXP .LE. 70) IEXP  = 0                                           ABNK0354
      FJWKB = HALF/(GH*GJWKB)                                              ABNK0355
      RETURN                                                               ABNK0356
      END                                                                  ABNK0357
//G.SYSIN  DD  *                                                           ABNK0358
```

```
-50.0          1    0   10   50                                                ABNK0359
  5.0                                                                          ABNK0360
-50.0          1   10   20                                                10  ABNK0361
 50.0                                                                          ABNK0362
-4.0           1    0    5   10                                               ABNK0363
 5.0                                                                          ABNK0364
-4.0           1   50  100                                                50  ABNK0365
50.0                                                                          ABNK0366
10.0           1    0    5   20                                               ABNK0367
 5.0                                                                          ABNK0368
10.0           1   30   50                                                30  ABNK0369
50.0                                                                          ABNK0370
 100.0         1    0   50  100                                              ABNK0371
1000.0                                                                        ABNK0372
100.0          1    0    5   10   30                                         0-4ABNK0373
  1.0                                                                         ABNK0374
   JWKB EXAMPLES   CALCULATIONS OF BARDIN ET AL.                             ABNK0375
100.0          2    0   20                                                   ABNK0376
  5.0     50.0                                                                ABNK0377
10.0           1   30   50                                                30  ABNK0378
 5.0                                                                          ABNK0379
-4.0           1   30   50                                                30  ABNK0380
 5.0                                                                          ABNK0381
500.0          2    0                                                       ABNK0382
0.001    0.025                                                               ABNK0383
 2.0           1    0    1                                                  0-4ABNK0384
10.0                                                                          ABNK0385
   OTHER TEST EXAMPLES OF COULOMB FUNCTIONS                                   ABNK0386
 4.0           1    1    4                                                  ABNK0387
10.0                                                                          ABNK0388
 5.0           1    0    1    7                                             ABNK0389
10.0                                                                          ABNK0390
10.0           1    0    7   10                                             ABNK0391
10.0                                                                          ABNK0392
25.0           1    0    1                                                  ABNK0393
10.0                                                                          ABNK0394
5.0            1    0    1   19   20                                        ABNK0395
5.0                                                                           ABNK0396
4.0            3    0                                                       ABNK0397
6.0       7.0       8.0                                                     ABNK0398
1.0            3    0                                                       ABNK0399
0.6       3.0       6.0                                                     ABNK0400
1.995          1    0                                                       ABNK0401
1.2                                                                           ABNK0402
3.981          1    0                                                       ABNK0403
2.4                                                                           ABNK0404
 18.564        1    0   30   60   80  100  120  140  160  180  200          ABNK0405
116.790                                                                       ABNK0406
-4.0           1    0    1    2    3                                         ABNK0407
 5.0                                                                          ABNK0408
0.0            6    0   10   30   40                             -1         0-2ABNK0409
1.0       2.0       5.0      10.0      50.0      100.0                       ABNK0410
   CYLINDRICAL BESSEL FUNCTIONS *** ABRAMOWITZ & STEGUN  PAGES 407 - 408ABNK0411
0.0            6    0   10   30   40                             -1         0-1ABNK0412
1.0       2.0       5.0      10.0      50.0      100.0                       ABNK0413
     SPHERICAL BESSEL FUNCTIONS *** ABRAMOWITZ & STEGUN  PAGES 465 - 466ABNK0414
0.0            9                                                         3.0-3ABNK0415
0.24      0.5       1.0       2.0       4.0       5.0       8.0           ABNK0416
10.0      16.0                                                              ABNK0417
   BESSELS WITH ORDER 1/3  ***  WATSON TABLE III   PAGES 714 - 729          ABNK0418
0.0            1    0    1    2    4    5    6   10   14   18   20    0-2ABNK0419
8.7714838                                                                     ABNK0420
   BLANCH TEST OF RECURRENCE THROUGH THE ZERO OF J5 NEAR X = 8.7714838  ABNK0421
0.0            2  100                                                   100 2ABNK0422
1.0       2.0                                                                 ABNK0423
-500.0         2    0                                                      -4ABNK0424
0.001     .000000001                                                          ABNK0425
    EXAMPLES OF ERROR CONDITIONS                                              ABNK0426
1.0            1 -2.2                                                       ABNK0427
-20000.0                                                                      ABNK0428
0.0            1    1                                                       ABNK0429
-55.0                                                                         ABNK0430
0.0       00000                                                               ABNK0431
//                                                                            ABNK0432
```