Computer Physics Communications 36 (1985) 363–372
North-Holland, Amsterdam

COULCC: A CONTINUED-FRACTION ALGORITHM FOR COULOMB FUNCTIONS OF COMPLEX ORDER WITH COMPLEX ARGUMENTS

I.J. THOMPSON

Daresbury Laboratory, Warrington WA4 4AD, UK

and

A.R. BARNETT

Physics Department, The University, Manchester M13 9PL, UK

Received 14 June 1984; in final form 5 December 1984

PROGRAM SUMMARY

Title of program: COULCC: Complex Coulomb & Bessel Functions

Catalogue number: ACDP

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

Computer:	*Installation*:
AS/7000	Daresbury SERC
CRAY-1	ULCC
CDC 7600	UMRCC

Operating system: OS/MVT with VS FORTRAN

Programming language used: Fortran 77

High-speed storage required: 822 Kb for compilation; 202 Kb for execution

No. of bits in a word: 32 (4 bytes of 8 bits)

Peripherals used: card reader or VDU, printer

No. of lines in combined program and test deck: 1658

Keywords: Coulomb, Bessel, Whittaker, hypergeometric, continued fraction, scattering, closed channels, off-shell, resonances, reactions, Regge poles

Nature of the physical problem
The routine COULCC calculates both the oscillating and the exponentially varying Coulomb wave functions, and their radial derivatives, for complex η(Sommerfeld parameter), complex energies and complex angular momenta. The functions for uncharged scattering (spherical Bessels) and cylindrical Bessel functions are special cases which are more easily solved. Two linearly independent solutions are found, in general, to the differential equation $f''(x)+g(x)f(x)=0$, where $g(x)$ has x^0, x^{-1} and x^{-2} terms, with coefficients 1, -2η and $-\lambda(\lambda+1)$, respectively.

Method of solution
The continued fractions of Steed [1,2] are supplemented, if necessary, by a ${}_1F_1$ series expansion or by Padé accelerations of a ${}_2F_0$ asymptotic expansion. Recurrence relations are used for integer steps in λ in a stable manner. It should be noted that the routine will solve for a single arbitrary λ without recurrence, if required. For small x a previous restriction on accuracy has been removed by adding a subroutine to evaluate the irregular solution (singular at $x=0$) by a suitable combination of series.

Restrictions of the complexity of the problem
On the Coulomb bound-state poles the functions are singular (from their boundary conditions). The program returns the residue polynomial but only one solution exists, and it is found.

Typical running time
A direct comparison of COULCC and its predecessor for real arguments shows an increase by a factor of 2 for the new code. The test deck (comprising 36 test values and excluding the error condition) took 1.14 s for execution on the NAS 7000 and 2.2 s on the CDC 7600.

References

[1] A.R. Barnett, D.H. Feng, J.W. Steed and L.B.J. Goldfarb, Comput. Phys. Commun. 8 (1974) 377.
[2] A.R. Barnett, Comput. Phys. Commun. 27 (1982) 147.

0010-4655/85/$03.30 © Elsevier Science Publishers B.V.
(North-Holland Physics Publishing Division)

LONG WRITE-UP

1. Introduction

This program, COULCC, is the complex generalisation of a continued-fraction algorithm for calculating real Coulomb wave functions which has been described and evaluated in a series of papers [1–7]. It is designed to find linearly independent solutions F, G, H^+ and H^- of the equation

$$\frac{d^2 f}{dx^2}+\left(1-\frac{2\eta}{x}-\frac{\lambda(\lambda+1)}{x^2}\right)f(x)=0$$

subject to the boundary conditions

$$F_\lambda(\eta,x)=0,$$

$$G_\lambda(\eta,0)=H_\lambda^+(\eta,0)=H_\lambda^-(\eta,0)=+\infty$$

and

$$F_\lambda(\eta,x)\underset{x\to\infty}{\to}\sin\theta_\lambda,\quad G_\lambda(\eta,x)\underset{x\to\infty}{\to}\cos\theta_\lambda,$$

$$H_\lambda^\pm(\eta,x)\underset{x\to\infty}{\to}e^{\pm i\theta_\lambda},$$

where $\theta_\lambda=x-\eta\ln(2x)-\frac{1}{2}\lambda\pi+\sigma_\lambda$ and σ_λ is the Coulomb phase shift

$$e^{i\sigma_\lambda}=\left[\frac{\Gamma(1+\lambda+i\eta)}{\Gamma(1+\lambda-i\eta)}\right]^{1/2}.$$

For positive-energy scattering problems, x is real and positive, η and λ are real, and the real F and G functions resulting can already be calculated to adequate precision, typically 10^{-12} [1–3]. When looking for example for off-shell structures such as resonance poles, x and η have small imaginary parts, so F and G are now complex in general. Tamura and Rybicki [9] have considered this case, and it is straightforward to extend COULFG [7] by complex continuation to treat off-shell regions near the physical axis. Another common analytic continuation is to complex angular momentum λ, to trace Regge poles, etc., as have been calculated by Takemasa et al. [10].

Negative energy solutions (i.e. x and η imaginary) are also required when seeking bound states, or when closed channels arise in multichannel scattering problems. Traditionally just the decaying Whittaker function is found, for $\mathrm{Im}(\eta)>0$ by Bell and Scott [11] and for $\mathrm{Im}(\eta)<0$ by Hebbard and Robson [12].

Solutions near zero energy are usually found, on the other hand, by 'Z-scaled coordinates' $r=-\eta x$ and $c=i/\eta$ so that they can be calculated, e.g. by Seaton [8] or Curtis [13], continuously across the energy threshold. The resulting wave functions are not dimensionless as are F, G, etc., and Humblet [14] has proposed similar wave functions for general scattering problems. We have chosen instead to continue calculating the standard Coulomb wave functions, but supplementing Steed's method [1] by a series expansion for the regular function near threshold. We also supplement the program by a Padé-accelerated asymptotic expansion for the Whittaker function as in ref. [15] and find that we are now able to include all of the above physical cases with x, η and λ complex in general.

2. Method of calculation

The general formulae for defining the Coulomb functions $F_\lambda(\eta,x)$ and $G_\lambda(\eta,x)$ for complex λ, η and x are described in a related paper [16] by the authors, along with a description of the different combinations of continued fractions used in different (x,η,λ) regions.

The power of the method derives principally from the evaluation by Steed's method of following continued fractions

$$\text{CF1: } f=\frac{F_\lambda'}{F_\lambda}=S_{\lambda+1}-\frac{R_{\lambda+1}^2}{T_{\lambda+1}-}\,\frac{R_{\lambda+2}^2}{T_{\lambda+2}-\dots},$$

where

$$S_\lambda=\frac{\lambda}{x}+\frac{\eta}{\lambda},\quad T_\lambda=S_\lambda+S_{\lambda+1}$$

and

$$R_\lambda^2=1+\eta^2/\lambda^2$$

and

$$\text{CF2}^\omega\text{: } p+i\omega q=H_\lambda^{\omega\prime}/H_\lambda^\omega$$

$$=i\omega\left(1-\frac{\eta}{x}\right)+\frac{i\omega}{x}\,\frac{ab}{2(x-\eta+i\omega)+}$$

$$\frac{(a+1)(b+1)}{2(x-\eta+2\mathrm{i}\omega)+\dots},$$

where $a=\mathrm{i}\omega\eta-\lambda$ and $b=\mathrm{i}\omega\eta+\lambda+1$, with the coefficients given analytically as indicated and $\omega=+1$ or $\omega=-1$.

Steed's algorithm, derived in ref. [1], is a forward recurrence method for evaluating a continued fraction of the form

$$h=a_0+\frac{a_1}{b_1+}\,\frac{a_2}{b_2+}\cdots\frac{a_n}{b_n+\dots}$$

by carrying forward just D_n, the ratio between successive denominators B_n in the equivalent polynomial ratio $h=A_n/B_n$, and thus avoiding possible overflows in evaluating A_n and B_n separately. The method is

```
$h_0 = a_0$,
$D_1 = 1/b_1$,
$\Delta h_1 = a_1 D_1$,
$h_1 = h_0 + \Delta h_1$,
for $n = 2$ upto $n_{\max}$ do begin
    $D_n = 1/(D_{n-1} a_n + b_n)$,
    $\Delta h_n = (b_n D_n - 1)\Delta h_{n-1}$,
    $h_n = h_{n-1} + \Delta h_n$,
    if $|\Delta h_n| < \epsilon |h_n|$ then exit
end
```

If $b_1=0$, then the continued fraction is reformulated as

$$h=a_0+\frac{a_1}{a_2}\left[b_2+\frac{a_3}{b_3+}\,\frac{a_4}{b_4+\dots}\right].$$

The continued fractions CF1 and CF2 form the heart of the method and above their sole use is labelled as case 2 and case 3 within the program. Case 3 is the real-parameter case and it hence corresponds to the parent program COUFLG [7]. Case 2 is for complex parameters but for the region where the resulting wavefunctions still retain oscillatory character. The absolute normalisations are determined from the Wronskian equation $F_\lambda' G_\lambda - G_\lambda' F_\lambda = 1$. For this to hold all parameters need to be close to their real axes. Other regions of the parameters relate to the 4 other cases described below.

These analytic continued fractions are supplemented in 'case 1' for small x by a direct evaluation of a ${}_1F_1$ series to give

$$F_\lambda(\eta,x)=x^{\lambda+1}\,\mathrm{e}^{\pm\mathrm{i}x}C_\lambda(\eta)$$
$$\times{}_1F_1(1+\lambda\pm\mathrm{i}\eta;\ 2\lambda+2;\ \mp2\mathrm{i}x).$$

This value of F_λ together with f from CF1 gives F_λ'. The upper or lower signs throughout are chosen to given $1+\lambda\pm\mathrm{i}\eta$ the *most negative real part possible*, to enable quicker and more accurate convergence. The ${}_1F_1$ series is absolutely convergent, but to extend the maximum values of η and x which are possible before large partial sums cause large fractional errors in the result, the evaluation is performed in extended precision. In COULCC, REAL*16 variables are used, writing out the real and imaginary arithmetic separately for use on machines such as the CRAY-1 which has no COMPLEX*32-type variables.

The corresponding evaluation of G_λ in case 1 is still achieved with CF2; whereas for yet smaller x the computation of CF2 becomes too inefficient and cases 5 and 6 are used instead. In these cases we re-expand the irregular functions in terms of the regular solution. If 2λ is non-integral (case 5), G_λ is found by means of the ${}_1F_1$ series for $F_{-\lambda-1}$, whereas if 2λ is an integer (case 6), G_λ is given by the logarithmic expansion (eq. 13.1.9 of ref. [17]) analogous to the Neumann series for Bessel functions.

For large-x values (case 4) asymptotic expansions based on $H_\lambda^\omega(\eta,x)=\mathrm{e}^{\mathrm{i}\omega\theta_\lambda}\,{}_2F_0(\mathrm{i}\omega\eta-\lambda,\mathrm{i}\omega\eta+\lambda+1;;\omega/(2\mathrm{i}x))$ are used, either directly for $H_\lambda(\omega)$, or in combination to give CF1A $=F'/F=(H^{+\prime}-H^{-\prime})/(H^+-H^-)$.

For intermediate-x values, the above asymptotic expansions are accelerated by Padé techniques, a process which has been found [15] to extend inward the region of convergence to limits set only by machine precision and exponent range. We use the P-algorithm of ref. [18] to calculate term-by-term the coefficients of the continued fraction 'corresponding' to the original series, in parallel with a term-by-term evaluation of this continued fraction by Steed's method. The Padé acceleration is only invoked if the asymptotic expansions are not sufficiently accurate on their own. Since we have an alternative method for smaller x, namely the ${}_1F_1$ series, we may set the maximum length JMAX of the continued fraction at only 50. We see from fig. 1 of ref. [16] that this includes the region where there is a relatively reliable correlation between the numbers of iterations for CF2$^{(\omega)}$ and for $H^{(\omega)}$ itself.

The code is set up to calculate the regular solution F_λ and one of the irregular solutions G_λ, $H_\lambda^+ = G_\lambda + \mathrm{i}F_\lambda$, or $H_\lambda^- = G_\lambda - \mathrm{i}F_\lambda$ according to a MODE variable. In general the user should ask for the linear combination with the smallest absolute magnitude, as it is impossible to obtain this solution accurately from the others due to the excessive cancellation. For large λ, G and $H^\pm$ are all large, whereas F_λ is small so is always returned. For small λ and x complex H^+ will usually be small for $\mathrm{Im}(x) > 0$, and H^- for $\mathrm{Im}(x) < 0$, so the user should choose $H^\pm$ according to the corresponding half-plane containing x. For small λ, and x near the axis, G and H^+ are all oscillating at about the same magnitude, so the choice of MODE is not critical here.

The code also calculates the Coulomb phase shifts $\sigma_\lambda(\eta)$, and these can be returned if required. This is strongly advised if either of $1+\lambda+\mathrm{i}\eta$ or $1+\lambda-\mathrm{i}\eta$ is likely to have a negative real part, as then the cut in $\sigma_\lambda(\eta)$ is approached. On the cut all the Coulomb functions change sign together, and there will be a simultaneous change of π in $\sigma_\lambda(\eta)$. If the Whittaker functions are to be calculated for example by

$$W_{-\mathrm{i}\omega\eta,\lambda+1/2}(-2\mathrm{i}\omega x) = \exp \mathrm{i}\omega\left[\tfrac{1}{2}\pi(\lambda+\mathrm{i}\omega\eta) - \sigma_\lambda(\eta)\right] H_\lambda^\omega(\eta, x)$$

with $\omega \arg x > -\pi/2$,

then the sign of W is only determined if the $\sigma_\lambda(\eta)$ used in this formula is that returned by the same code which calculated the H_λ. A consistent treatment of the cut is then maintained.

If one of $1+\lambda\pm\mathrm{i}\eta$ is zero or a negative integer, then a gamma function in the expression for $\sigma_\lambda(\eta)$ will be evaluated on its pole. These are just the hydrogenic Coulomb bound-state poles, and with these states $\mathrm{e}^{\mathrm{i}\sigma_\lambda}$, F_λ and $H_\lambda^\pm$ are all, strictly speaking, infinite. The residues, however, are finite polynomials, and the code uses the zero returned by the log-gamma routine to calculate these polynomials scaled to finite values. A corresponding "σ_λ" is returned such that the Whittaker function calculated by the formula above is still correct, even though the gamma routine printed an error message.

The above two paragraphs imply that if there is any choice of $1+\lambda\pm\mathrm{i}\eta$ approaching the negative real axis (e.g. for Coulomb eigenstates), the user should calculate the Whittaker function from H^+ by means of the phase shifts returned at the same time, and use the Whittaker function in subsequent calculations.

3. Code description

The calling sequence for the program is (see fig. 1)

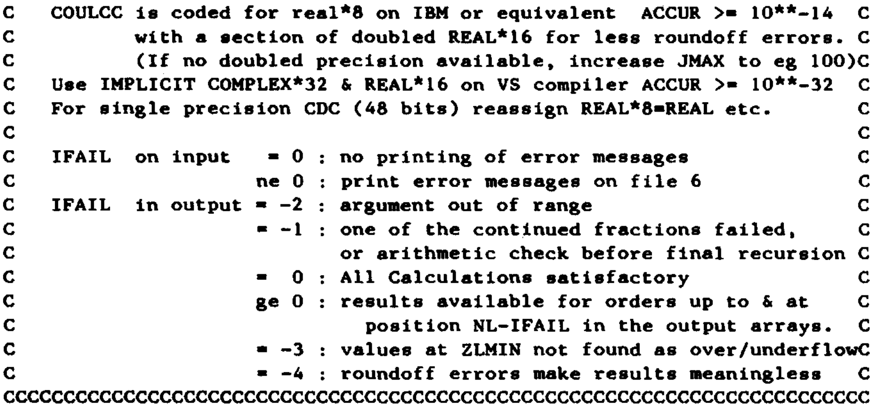
CALL COULCC(X,ETA,ZLMIN,NL,FC,
GC,FCP,GCP,SIG,MODE1,KFN,IFAIL)

where the arguments have the following type and meaning.

X	complex	$x \neq 0$,
ETA	complex	η,
ZLMIN	complex	$\lambda_{\min}$,
NL	integer	number of λ values $\lambda_{\min}$, $\lambda_{\min}+1, \ldots,$ $\lambda_{\min}+\mathrm{NL}-1$ calculated;
FC	complex array	dimension NL, regular solution, F_λ,
GC	complex array	dimension NL, irregular solution G_λ or H_λ,
FCP	complex array	dimension NL, regular derivative $F'_\lambda = \mathrm{d}F_\lambda/\mathrm{d}x$,
GCP	complex array	dimension NL, irregular derivative G'_λ or H'_λ,
SIG	complex array	dimension NL, Coulomb phase shifts if KFN = 0;

MODE1 integer: |MODE1| determines selection of F, G, $H^\pm$, and MODE1 < 0 indicates that exponential scaling is to be used.

```
      SUBROUTINE COULCC(XX,ETA1,ZLMIN,NL, FC,GC,FCP,GCP, SIG,
     X                  MODE1,KFN,IFAIL)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C  COMPLEX COULOMB WAVEFUNCTION PROGRAM USING STEED'S METHOD           C
C                                                                      C
C  A. R. Barnett           Manchester  March     1981                  C
C  modified I.J. Thompson  Daresbury, Sept. 1983 for Complex Functions C
C                                                                      C
C  original program  RCWFN       in    CPC  8 (1974) 377-395           C
C                 +  RCWFF       in    CPC 11 (1976) 141-142           C
C                 +  COULFG      in    CPC 27 (1982) 147-166           C
C  description of real algorithm in    CPC 21 (1981) 297-314           C
C  description of complex algorithm    JCP XX (1984) YYY-ZZZ           C
C  this version written up       in    CPC XX (1984) YYY-ZZZ           C
C                                                                      C
C  COULCC returns F,G,G',G',SIG for complex XX, ETA1, and ZLMIN,       C
C   for NL integer-spaced lambda values ZLMIN to ZLMIN+NL-1 inclusive, C
C   thus giving  complex-energy solutions to the Coulomb Schrodinger   C
C   equation,to the Klein-Gordon equation and to suitable forms of     C
C   the Dirac equation ,also spherical & cylindrical Bessel equations  C
C                                                                      C
C  if /MODE1/= 1  get F,G,F',G'   for integer-spaced lambda values     C
C            = 2      F,G      unused arrays must be dimensioned in    C
C            = 3      F,  F'          call to at least length (1)      C
C            = 4      F                                                C
C            = 11 get F,H+,F',H+' ) if KFN=0, H+ = G + i.F        )    C
C            = 12     F,H+        )       >0, H+ = J + i.Y = H(1) ) in C
C            = 21 get F,H-,F',H-' ) if KFN=0, H- = G - i.F        ) GC C
C            = 22     F,H-        )       >0, H- = J - i.Y = H(2) )    C
C                                                                      C
C     if MODE1<0 then the values returned are scaled by an exponential C
C                factor (dependent only on XX) to bring nearer unity   C
C                the functions for large /XX/, small ETA & /ZL/ < /XX/ C
C        Define SCALE = (  0        if MODE1 > 0                       C
C                       (  IMAG(XX) if MODE1 < 0  &  KFN < 3           C
C                       (  REAL(XX) if MODE1 < 0  &  KFN = 3           C
C        then FC = EXP(-ABS(SCALE)) * ( F, j, J, or I)                 C
C         and GC = EXP(-ABS(SCALE)) * ( G, y, or Y )                   C
C               or EXP(SCALE)       * ( H+, H(1), or K)                C
C               or EXP(-SCALE)      * ( H- or H(2) )                   C
C                                                                      C
C  if  KFN  =  0,-1  complex Coulomb functions are returned   F & G    C
C           =  1   spherical Bessel      "      "     "       j & y    C
C           =  2 cylindrical Bessel      "      "     "       J & Y    C
C           =  3 modified cyl. Bessel    "      "     "       I & K    C
C                                                                      C
C          and where Coulomb phase shifts put in SIG if KFN=0 (not -1) C
C                                                                      C
C  The use of MODE and KFN is independent                              C
C    (except that for KFN=3,  H(1) & H(2) are not given)               C
C                                                                      C
C  With negative orders lambda, COULCC can still be used but with      C
C    reduced accuracy as CF1 becomes unstable. The user is thus        C
C    strongly advised to use reflection formulae based on              C
C    H+-(ZL,,) = H+-(-ZL-1,,) * exp +-i(sig(ZL)-sig(-ZL-1)-(ZL+1/2)pi) C
C                                                                      C
C  Precision:  results to within 2-3 decimals of 'machine accuracy',   C
C               but if CF1A fails because X too small or ETA too large C
C               the F solution  is less accurate if it decreases with  C
C               decreasing lambda (e.g. for lambda.LE.-1 & ETA.NE.0)   C
C              RERR in COMMON/STEED/ traces the main roundoff errors.  C
C                                                                      C
```



```
C   COULCC is coded for real*8 on IBM or equivalent  ACCUR >= 10**-14  C
C          with a section of doubled REAL*16 for less roundoff errors. C
C          (If no doubled precision available, increase JMAX to eg 100)C
C   Use IMPLICIT COMPLEX*32 & REAL*16 on VS compiler ACCUR >= 10**-32  C
C   For single precision CDC (48 bits) reassign REAL*8=REAL etc.       C
C                                                                      C
C   IFAIL  on input   = 0 : no printing of error messages              C
C                    ne 0 : print error messages on file 6             C
C   IFAIL  in output = -2 : argument out of range                      C
C                    = -1 : one of the continued fractions failed,     C
C                           or arithmetic check before final recursion C
C                    =  0 : All Calculations satisfactory              C
C                    ge 0 : results available for orders up to & at    C
C                             position NL-IFAIL in the output arrays.  C
C                    = -3 : values at ZLMIN not found as over/underflowC
C                    = -4 : roundoff errors make results meaningless   C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

Fig. 1. Subroutine COULCC, introductory comment lines.

If |MODE1|

= 1	F, G, F', G'	
= 2	F, G	
= 3	F, F'	
= 4	F	
= 11	$F, H^+, F', H^{+\prime}$	if KFN $\leqslant 0$,
= 12	F, H^+	$H^{\pm} = G \pm \mathrm{i}F$
= 21	$F, H^-, F', H^{-\prime}$	if KFN > 0,
= 22	F, H^-	$H^{\pm} = J \pm \mathrm{i}Y$

where F is in the FC array, G, H^+ or H^- is in the GC array and where the definition of $H^{\pm}$ is fixed by KFN (see below).

If MODE1 < 0, then the values retained are scaled by an exponential factor (dependent only on x), designed to bring nearer unity the functions for large $|x|$ and small λ, η:

Define $s = \mathrm{imag}(x)$ for KFN $\leqslant 2$ and $s = \mathrm{real}(x)$ for KFN = 3,

$$\text{then FC} = \mathrm{e}^{-|s|}\{F,\ j,\ J \text{ or } I\}$$

$$\text{and GC} = \begin{cases} \mathrm{e}^{-|s|}\{G,\ y \text{ or } Y\} \text{ or} \\ \mathrm{e}^{s}\{H^+ \text{ or } K\} \text{ or} \\ \mathrm{e}^{-s}\{H^-\}. \end{cases}$$

KFN integer, determines kind of Coulomb or Bessel functions:

KFN		FC:	GC:
0	complex Coulomb functions	$F_\lambda(\eta, x)$	$G_\lambda(\eta, x)$ or $H^{\pm} = G \pm \mathrm{i}F$
1	complex spherical Bessels	$j_\lambda(x)$	$y_\lambda(x)$ or $h^{\pm} = j \pm \mathrm{i}y$
2	complex cylindrical Bessels	$J_\lambda(x)$	$Y_\lambda(x)$ or $H^{\pm} = J \pm \mathrm{i}Y$
3	modified cylindrical Bessels	$I_\lambda(x)$	$K_\lambda(x)$
−1	Coulomb functions but without using the SIG array.		

The FCP and GCP arrays contain the derivatives of the appropriate Coulomb or Bessel function with respect to x.

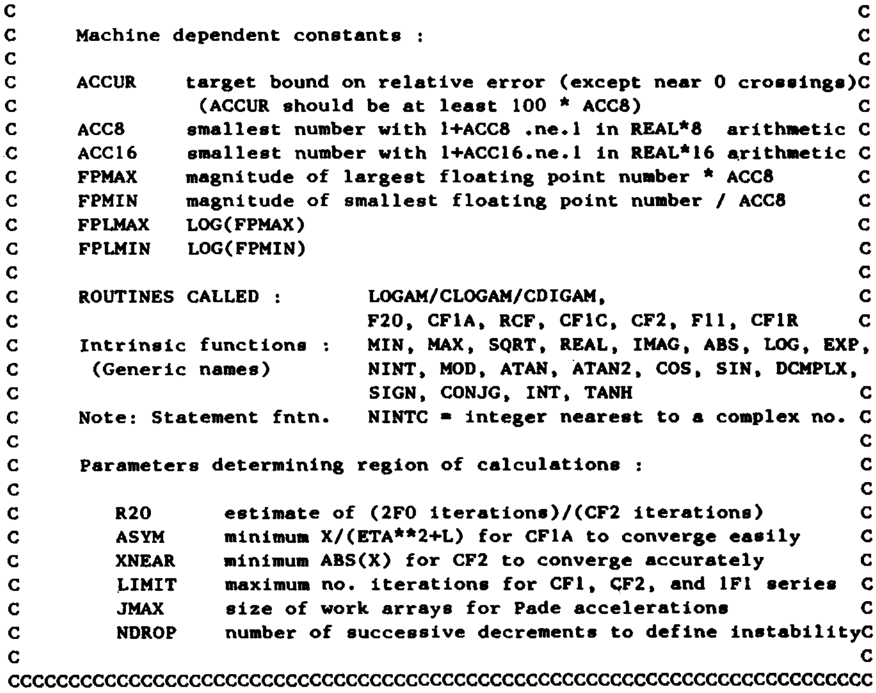
If KFN = 3, then $H^{\pm}$ are not defined, and mod(MODE1, 10) is used in place of MODE1.

If KFN $\neq 3$, the use of KFN and MODE1 is independent.

IFAIL: integer on input
= 0 no printing of error messages,
$\neq 0$ print error messages on file 6;
on output
= 0 no errors occurred,
$\neq 0$ error exit.

The error exits are further classified:

IFAIL
= −2: argument out of range;
= −1: one of the continued fractions failed or arithmetic check, before any results calculated;
= −3: the functions of $\lambda_{\min}$ could not be calculated (but perhaps some at $\lambda > \lambda_{\min}$);



```
C                                                                    C
C     Machine dependent constants :                                  C
C                                                                    C
C     ACCUR    target bound on relative error (except near 0 crossings)C
C               (ACCUR should be at least 100 * ACC8)                C
C     ACC8     smallest number with 1+ACC8 .ne.1 in REAL*8  arithmetic C
C     ACC16    smallest number with 1+ACC16.ne.1 in REAL*16 arithmetic C
C     FPMAX    magnitude of largest floating point number * ACC8     C
C     FPMIN    magnitude of smallest floating point number / ACC8    C
C     FPLMAX   LOG(FPMAX)                                            C
C     FPLMIN   LOG(FPMIN)                                            C
C                                                                    C
C     ROUTINES CALLED :       LOGAM/CLOGAM/CDIGAM,                   C
C                             F20, CF1A, RCF, CF1C, CF2, F11, CF1R   C
C     Intrinsic functions :   MIN, MAX, SQRT, REAL, IMAG, ABS, LOG, EXP,
C      (Generic names)        NINT, MOD, ATAN, ATAN2, COS, SIN, DCMPLX,
C                             SIGN, CONJG, INT, TANH                 C
C     Note: Statement fntn.   NINTC = integer nearest to a complex no. C
C                                                                    C
C     Parameters determining region of calculations :                C
C                                                                    C
C        R20      estimate of (2F0 iterations)/(CF2 iterations)      C
C        ASYM     minimum X/(ETA**2+L) for CF1A to converge easily   C
C        XNEAR    minimum ABS(X) for CF2 to converge accurately      C
C        LIMIT    maximum no. iterations for CF1, CF2, and 1F1 series C
C        JMAX     size of work arrays for Pade accelerations         C
C        NDROP    number of successive decrements to define instabilityC
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

Fig. 2. Machine dependent constants and subroutines used.

> 0: arithmetic error during final recursion, so that results are available in the FC, etc., arrays up to and including position NL − IFAIL;

$= -4$: roundoff errors $> \sqrt[4]{\text{ACCUR}}$.

The criterion for convergence of the internal expansions is consistency to within one half of the value of 'ACCUR' (see fig. 2). In the supplied code, ACCUR is set at 10^{-14}, appropriate for REAL∗8 arithmetic on IBM machines. For CDC 6600/7600 systems set ACCUR = 10^{-12}, and set it also to 10^{-12} on CRAY machines. The ACCUR value may be increased if these full accuracies are not required, and if faster calculations would be useful. If ACCUR is 10^{-8}, for example, then (apart from the limiting cases described in section 4), the relative errors will be less than 40% of this figure except near zero-crossings of the functions. ACCUR is the target upper bound on the relative errors in the results and is deliberately set at about 100 times the unit roundoff errors.

Further information about the performance of COULCC is provided by a named common block /STEED/ containing in order RERR(REAL∗8) and the integers NFP, N11, NPQ(2), N20, KAS(2).

RERR is the estimated maximum relative error in the calculated Coulomb functions FC, GC. Then

NFP = number of iterations for CF1, or
- (number of terms for CF1A);

N11 = number of terms required for ${}_1F_1$ series;

NPQ(IH) = number of terms required for CF2$^+$ or CF2$^-$ for
IH = 1 and 2, respectively;

N20 = number of terms required for ${}_2F_0$ expansion;

KAS(1) = case number, 1, 2, 3, 4, 5 or 6 for upper range of λ values;

KAS(2) = case number, 1, 2, 3, 4, 5 or 6 for lower range of λ values.

The meanings of these case numbers are defined in the accompanying paper [16].

4. Accuracy and range of arguments

As a rule, the results are obtained within 2 or 3 decimals of machine precision, except for the following range limitations:

(a) negative λ:
The code can still be used, but with reduced accuracy as CF1 becomes unstable. The user is thus strongly advised to use λ-reflection formulae based on

$$H_\lambda^\pm = H_{-\lambda-1}^\pm \exp \pm i(\sigma_\lambda - \sigma_{-\lambda-1} - (\lambda + 1/2)\pi).$$

(b) $|\lambda_{\min}|$, $|\mathrm{Im}(\eta)| > 50$:
The convergence of the ${}_1F_1$ series and of the Padé-accelerated ${}_2F_0$ expansions is now significantly reduced and may become inaccurate.

(c) F-unstable cases:
In these cases CF1 cannot be used as the F solution decreases (rather than increases or oscillates) for a range of decreasing λ, for example for $x = 100i$ and $\eta = 50i$. These, however, can only be detected if the number of λ-values required, NL > 6, and in severely unstable cases only if NL is large enough so that the instability is nearly ended (in the above example F is *de*creasing below $\lambda \approx \sqrt{-nx}$). Once the, instability is found, CF1A is used in preference, but if this occurs for η too large CF1A may not converge; this is the limitation of the program.

The code only allows for one reversal of the direction of recurrences because of instabilities, so if negative λ are also required the λ-reflection formulae will have to be used before and after calling COULCC. The presence of poles (see(e)) also leads to reversals, so the program cannot handle both poles and instabilities, but should be called once for $\lambda = \lambda_{\min}$ to $\mathrm{Im}(\eta - 1)$ and then for $\lambda = \mathrm{Im}\ \eta$ up to $\lambda_{\max}$.

(d) Small or medium x, if no REAL*16 arithmetic is available for the ${}_1F_1$ series calculation:
The partial sums in this series may be much larger than the final result if η is large, or if x is just below where asymptotic expansions may be used. For complex Bessels ($\eta = 0$) the error would degrade up to 10^5 for x between 5 and 15.

(e) Near poles and zeros:
The relative accuracy is determined principally by the accuracy with which the distance to the pole or zero can be calculated in the floating-point arithmetic. Poles and zeros are approached whenever $1 + \lambda - i\eta$ or $1 + \lambda + i\eta$ is near a nonpositive integer, and there are the usual zeros when real-valued functions change sign, etc.

(f) Large x:
The oscillating phase is as $\sin(x)$, and the accuracy in the results will decrease just according to the accuracy in $\mathrm{mod}(x, 2\pi)$.

5. Subroutines used

COULCC calls the two routines CF1A and F20 as complex functions to use asymptotic expansions to calculate the regular logarithmic derivative $f = F'/F$ and ${}_2F_0$ values, respectively. If the expansions start to diverge, the two routines call RCF to calculate the corresponding continued fraction coefficients. The RCF code is copied from table 9 of ref. [18], but with a modified restart procedure and a change to complex variables.

The routines CF1R and CF1C calculate the continued fraction CF1 for real and complex arguments, respectively, while the function CF2 evaluates the complex continued fraction CF2. The complex function F11 evaluates the ${}_1F_1$ series for cases 1, 5 and 6, either in normal precision (with COMPLEX*16 variables) or in doubled precision (with REAL*16 variables). It can also calculate the ${}_1F_1$-like series with the digamma factors that appears in the logarithmic solution for the irregular function.

The code uses a combined CLOGAM and CDIGAM function to calculate the natural logarithm and the logarithmic derivative of the gamma function with complex argument, and with the cut along the negative real axis. Initially we used Kölbig's program [19], but subsequently we generalised his methods and his code for arbitrary precision, subject only to machine accuracy and not subject to prestored floating point coefficients. In the test deck this precision is pre-set by a call from COULCC to the ENTRY LOGAM with one

REAL*8 argument ACCUR. The gamma function results decrease worsen by one or two decimals from machine accuracy in general, because of the number of recurrence steps required for very high accuracy calculations and the fixed collection of Bernoulli numbers.

6. Test deck

The test deck contains a main program to call COULCC for a succession of x, η and λ combinations determined by the data read in. The first 36 data cards are designed to reproduce published results of existing Coulomb, Bessel and Whittaker programs. The final set of 6 cards shows typical error conditions that may result if the input ranges are exceeded.

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.

[3] A.R. Barnett, J. Comput. Phys. 46 (1982) 171.

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

[5] A.R. Barnett, J. Comput. Appl. Math. 8 (1982) 29.

[6] D.H. Feng, A.R. Barnett and L.J.B. Goldfarb, Phys. Rev. 13C (1976) 1151.

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

[8] M.J. Seaton, Comput. Phys. Commun. 25 (1982) 87.

[9] T. Tamura and F. Rybicki, Comput. Phys. Commun. 1 (1969) 25.

[10] T. Takemasa, T. Tamura and H.H. Wolter, Comput. Phys. Commun. 17 (1979) 351.

[11] K.L. Bell and N.S. Scott, Comput. Phys. Commun. 20 (1980) 447.

[12] D.F. Hebbard and B.A. Robson, Nucl. Phys. 42 (1963) 563.

[13] A.R. Curtis, Coulomb Wave Functions, vol. 11, Royal Soc. Math. Tables (Cambridge Univ. Press, London/New York, 1964) p. 9.

[14] J. Humblet, Nucl. Phys. 50 (1964) 1; Ann. Phys. 155 (1984) 461.

[15] C.J. Noble and I.J. Thompson, Comput. Phys. Commun. 33 (1984) 413.

[16] I.J. Thompson and A.R. Barnett, J. Comput. Phys. (submitted).

[17] M. Abramowitz, Handbook of Mathematical Functions. eds. M. Abramowitz and I.A. Stegun (Nat. Bur. Std., New York, 1964) p. 537; Tables of Coulomb Wavefunctions, NBS Applied Math. Ser. 17 (1952)

[18] J. Patry and S. Gupta, EIR-Bericht Nr. 247, Eidg. Institut für Reaktorforschung, Würenlingen, Switzerland (1973).

[19] K.S. Kölbig, Comput. Phys. Commun. 4 (1972) 221.

TEST RUN OUTPUT

```
TEST OF THE CONTINUED-FRACTION COULOMB & BESSEL ROUTINES

X =      0.0000  10.0000, ETA =   0.000    0.000, ZLMIN =    0.000     0.000  NL =   2  MODE =  12  KFN = 2          ARDILL: J,H(1)
COULCC  :: IFAIL =   0  RERR =   5.0D-15   ITS =    20    0      8      0  16 4 0

ZL =   0.000    0.000 :: FC = J    =  2.815716628466D+03 -1.136868377216D-13,  GC = H(1) = -1.694065894509D-21 -1.131913922440D-05
ZL =   1.000    0.000 :: FC = J    = -5.684341886081D-14  2.670988303701D+03,  GC = H(1) = -1.187217791111D-05  1.694065894509D-21

X =      0.0000  15.0000, ETA =   0.000    0.000, ZLMIN =    0.000     0.000  NL =   2  MODE =  12  KFN = 2          ARDILL: J,H(1),?
COULCC  :: IFAIL =   0  REPP =   5.0D-15   ITS =   -15    0      6      0  14 4 0

ZL =   0.000    0.000 :: FC = J    =  3.396493732979D+05 -3.125751391053D-08,  GC = H(1) = -6.617444900424D-24 -6.251311080178D-08
ZL =   1.000    0.000 :: FC = J    = -3.223249223083D-08  3.281249219703D+05,  GC = H(1) = -6.456425442792D-08  6.617444900424D-24

X =      0.0100   0.0000, ETA =   0.000    0.000, ZLMIN =    0.200     0.000  NL =  11  MODE =  -1  KFN = 3          CAMPBELL: I,K
COULCC  :: IFAIL =   0  RERR =   5.9D-15   ITS =     2    7      8      0   0 5 0

ZL =   0.200    0.000 :: FC = I    =  3.737124789554D-01 -1.429412144205D-15,  GC = K      =  5.671099356385D+00  3.108624468950D-15
                         FC'=         7.475806699691D+00 -3.086420008458D-14,  GC' =        -1.541397750429D+02 -1.004529792681D-12
ZL =   1.200    0.000 :: FC = I    =  1.557120583573D-03 -6.071532165919D-18,  GC = K      =  2.675617621706D+02  1.094235813071D-12
                         FC'=         1.868580089266D-01 -7.216449660064D-16,  GC' =        -3.211308255983D+04 -1.311093456025D-10
ZL =   3.200    0.000 :: FC = I    =  5.529517605273D-09 -2.176518986780D-23,  GC = K      =  2.825728493070D+07  1.187436282635D-07
                         FC'=         1.769452216439D-06 -7.080666043454D-21,  GC' =        -9.042395398318D+09 -3.770645707846D-05
ZL =  10.200    0.000 :: FC = I    =  5.758335813199D-31 -2.309952937285D-45,  GC = K      =  8.512803984809D+28  3.562417673994D+14
                         FC'=         5.873505100148D-28 -2.286919093778D-42,  GC' =        -8.683064691027D+31 -3.645892596775D+17

X =     12.2000  13.3000, ETA =   0.000    0.000, ZLMIN =    0.100     0.000  NL =  31  MODE =  -2  KFN = 3          CAMPBELL: I,K
COULCC  :: IFAIL =   0  RERR =   6.6D-15   ITS =    13    0      0      6  21 4 0

ZL =   0.100    0.000 :: FC = I    =  8.970674903859D-02  2.916001420868D-02,  GC = K      =  1.217994261495D-01 -2.672433914395D-01
ZL =   1.100    0.000 :: FC = I    =  8.696982963893D-02  3.073786978475D-02,  GC = K      =  1.180862385335D-01 -2.761944082622D-01
ZL =   3.100    0.000 :: FC = I    =  6.851377438327D-02  3.900172579084D-02,  GC = K      =  8.332207749491D-02 -3.417842041719D-01
ZL =  10.100    0.000 :: FC = I    = -9.977468635814D-03  8.634263894440D-03,  GC = K      = -2.076663354998D+00 -8.325220423850D-02
ZL =  30.100    0.000 :: FC = I    = -5.143145898984D-10  4.514431674944D-10,  GC = K      = -2.043957294439D+07 -1.234324404641D+07

X =      0.0000  19.2000, ETA =   0.000    0.000, ZLMIN =    0.728     0.000  NL =  11  MODE =  -2  KFN = 3          CAMPBELL: I
COULCC  :: IFAIL =   0  RERR =   5.0D-15   ITS =    27 122      6      0   0 1 0

ZL =   0.728    0.000 :: FC = I    = -3.136419753777D-05 -6.888616210552D-05,  GC = K      =  1.186538508864D-01 -2.603162181702D-01
ZL =   1.728    0.000 :: FC = I    =  1.656955187336D-01 -7.544195846947D-02,  GC = K      =  1.019544615621D-01 -2.678089736316D-01
ZL =   3.728    0.000 :: FC = I    =  1.572397681015D-01 -7.159201495323D-02,  GC = K      =  2.276629885649D-02 -2.878280139239D-01
ZL =  10.728    0.000 :: FC = I    =  6.550517211471D-03  1.438710459469D-02,  GC = K      = -1.521941617418D-01  2.743438906896D-01

X =      0.0100   0.0000, ETA =   0.000    0.000, ZLMIN =    0.100     0.000  NL =  11  MODE =   2  KFN = 2          CAMPBELL: J,Y
COULCC  :: IFAIL =   0  RERR =   6.7D-15   ITS =     2    7      8      0   0 5 0

ZL =   0.100    0.000 :: FC = J    =  6.187945159047D-01  2.162670435538D-18,  GC = Y      = -3.239318624880D+00  0.000000000000D+00
ZL =   1.100    0.000 :: FC = J    =  2.812732785969D-03  9.830426551842D-21,  GC = Y      = -1.028953666168D+02 -4.543770585065D-15
ZL =   3.100    0.000 :: FC = J    =  1.080164792694D-08  3.775147326981D-26,  GC = Y      = -9.506068466200D+06 -4.198764396184D-10
ZL =  10.100    0.000 :: FC = J    =  1.251675398831D-30  4.374572350539D-48,  GC = Y      = -2.517892893399D+28 -1.112135797443D+12

X =     20.0000  -0.0010, ETA =   0.000    0.000, ZLMIN =    0.000     0.000  NL =  11  MODE =   2  KFN = 1          LENTZ: J,Y SPH.BESSE
COULCC  :: IFAIL =   0  RERR =   5.0D-15   ITS =    29    0      1      1   0 2 0

ZL =   0.000    0.000 :: FC = j    =  4.564728626610D-02 -1.812174217805D-05,  GC = y      = -2.040411095935D-02 -4.666747569223D-05
ZL =   1.000    0.000 :: FC = j    = -1.812174660644D-02 -4.745944458348D-05,  GC = y      = -4.666749169485D-02  1.573735812200D-05
ZL =   3.000    0.000 :: FC = j    =  6.030359527255D-03  4.957160333180D-05,  GC = y      =  5.001848785414D-02 -3.400290009586D-06
ZL =  10.000    0.000 :: FC = j    =  3.968671448388D-02 -2.905654781438D-05,  GC = y      = -3.684342205110D-02 -3.601856606678D-05

X =     20.0000-100.0000, ETA =   0.000    0.000, ZLMIN =    0.000     0.000  NL =  11  MODE =   2  KFN = 1          LENTZ: J,Y SPH.BESSE
COULCC  :: IFAIL =   0  RERR =   5.0D-15   ITS =   -12    0      0      1   2 4 0

ZL =   0.000    0.000 :: FC = j    =  7.633620871309D+40  1.074379471584D+41,  GC = y      =  1.074379471584D+41 -7.633620871309D+40
ZL =   1.000    0.000 :: FC = j    =  1.065516903756D+41 -7.539559526939D+40,  GC = y      = -7.539559526939D+40 -1.065516903756D+41
ZL =   3.000    0.000 :: FC = j    = -1.022204407411D+41  7.085201461840D+40,  GC = y      =  7.085201461840D+40  1.022204407411D+41
ZL =  10.000    0.000 :: FC = j    = -3.791897380371D+40 -6.759622074188D+40,  GC = y      = -6.759622074188D+40  3.791897380371D+40

X =   1000.0000 -10.0000, ETA =   0.000    0.000, ZLMIN =    0.000     0.000  NL =  11  MODE =   2  KFN = 1          LENTZ: J,Y SPH.BESSE
COULCC  :: IFAIL =   0  RERR =   2.0D-13   ITS =    -8    0      0      1   2 4 0
```