MATHEMATICS OF COMPUTATION
VOLUME 51, NUMBER 183
JULY 1988, PAGES S1-S5

Supplement to
A Table of Elliptic Integrals of the Third Kind

By B. C. Carlson

This supplement contains Fortran codes for the functions $R_C(x,y)$ and $R_J(x,y,z,p)$. If y or p is negative, the Cauchy principal value is computed. The codes are followed by some numerical values that were used to check the formulas in Sections 2 and 3.

```
C***********************************************************************
C
      DOUBLE PRECISION FUNCTION RC(X,Y,ERRTOL,IERR)
C
C          THIS FUNCTION SUBROUTINE COMPUTES THE ELEMENTARY INTEGRAL
C          RC(X,Y) = INTEGRAL FROM ZERO TO INFINITY OF
C
C                              -1/2     -1
C                   (1/2)(T+X)     (T+Y)  DT,
C
C          WHERE X IS NONNEGATIVE AND Y IS NONZERO.  IF Y IS NEGATIVE,
C          THE CAUCHY PRINCIPAL VALUE IS COMPUTED BY USING A PRELIMI-
C          NARY TRANSFORMATION TO MAKE Y POSITIVE; SEE EQUATION (2.12)
C          OF THE SECOND REFERENCE BELOW.  WHEN Y IS POSITIVE, THE
C          DUPLICATION THEOREM IS ITERATED UNTIL THE VARIABLES ARE
C          NEARLY EQUAL, AND THE FUNCTION IS THEN EXPANDED IN TAYLOR
C          SERIES TO FIFTH ORDER.  LOGARITHMIC, INVERSE CIRCULAR, AND
C          INVERSE HYPERBOLIC FUNCTIONS ARE EXPRESSED IN TERMS OF RC
C          BY EQUATIONS (4.9)-(4.13) OF THE SECOND REFERENCE BELOW.
C          REFERENCES: B. C. CARLSON AND E. M. NOTIS, ALGORITHMS FOR
C          INCOMPLETE ELLIPTIC INTEGRALS, ACM TRANSACTIONS ON MATHEMA-
C          TICAL SOFTWARE, 7 (1981), 398-403; B. C. CARLSON, COMPUTING
C          ELLIPTIC INTEGRALS BY DUPLICATION, NUMER. MATH. 33 (1979),
C          1-16.
C          AUTHORS: B. C. CARLSON AND ELAINE M. NOTIS, AMES LABORATORY-
C          DOE, IOWA STATE UNIVERSITY, AMES, IA 50011, AND R. L. PEXTON,
C          LAWRENCE LIVERMORE NATIONAL LABORATORY, LIVERMORE, CA 94550.
C          AUG. 1, 1979, REVISED SEPT. 1, 1987.
C
C          CHECK VALUES: RC(0,1/4) = RC(1/16,1/8) = PI,
C                        RC(9/4,2) = LN(2),
C                        RC(1/4,-2) = LN(2)/3.
C          CHECK BY ADDITION THEOREM: RC(X,X+Z) + RC(Y,Y+Z) = RC(0,Z),
C          WHERE X, Y, AND Z ARE POSITIVE AND  X * Y = Z * Z.
C
```

```
      INTEGER IERR,PRINTR
      DOUBLE PRECISION C1,C2,ERRTOL,LAMDA,LOLIM
      DOUBLE PRECISION MU,S,SN,UPLIM,X,XN,Y,YN,W
      LOGICAL FLAG
C          INTRINSIC FUNCTIONS USED: DABS,DMAX1,DSQRT
C
C          PRINTR IS THE UNIT NUMBER OF THE PRINTER.
      DATA PRINTR/6/
C
C          LOLIM DETERMINES THE LOWER LIMIT AND UPLIM THE UPPER LIMIT
C          OF THE RANGE OF ADMISSIBLE VALUES OF X AND Y FOR WHICH THE
C          COMPUTATION WILL PROCEED WITHOUT UNDERFLOW OR OVERFLOW.
C          LOLIM IS NOT LESS THAN THE MACHINE MINIMUM MULTIPLIED BY 5.
C          UPLIM IS NOT GREATER THAN THE MACHINE MAXIMUM DIVIDED BY 5.
C
C          ACCEPTABLE VALUES FOR:   LOLIM          UPLIM
C          IBM 360/370 SERIES    :  3.0D-78        1.0D+75
C          CDC 6000/7000 SERIES :   1.0D-292       1.0D+321
C          UNIVAC 1100 SERIES    :  1.0D-307       1.0D+307
C          CRAY                  :  2.30D-2438     5.40D+2464
C          VAX 11 SERIES         :  1.50D-38       3.0D+37
C          IBM PC                :  1.50D-38       3.0D+37
C
C          WARNING: IF THIS PROGRAM IS CONVERTED TO SINGLE PRECISION,
C          THE VALUES FOR THE UNIVAC 1100 SERIES SHOULD BE CHANGED TO
C          LOLIM = 1.0E-37 AND UPLIM = 1.0E+37 BECAUSE THE MACHINE
C          EXTREMA CHANGE WITH THE PRECISION.
C
      DATA LOLIM/1.50D-38/, UPLIM/3.0D+37/
C
C          ON INPUT:
C
C          X AND Y ARE THE VARIABLES IN THE INTEGRAL RC(X,Y).
C
C          ERRTOL IS CHOSEN TO DETERMINE THE ACCURACY OF THE COMPUTED
C          APPROXIMATION TO THE INTEGRAL.  TRUNCATION OF A TAYLOR SERIES
C          AFTER TERMS OF FIFTH ORDER INTRODUCES A RELATIVE ERROR LESS
C          THAN THE AMOUNT SHOWN IN THE SECOND COLUMN OF THE FOLLOWING
C          TABLE FOR EACH VALUE OF ERRTOL IN THE FIRST COLUMN.  IN ADDI-
C          TION TO THE TRUNCATION ERROR THERE WILL BE ROUNDOFF ERROR,
C          BUT IN PRACTICE THE TOTAL ERROR FROM BOTH SOURCES IS USUALLY
C          LESS THAN THE AMOUNT GIVEN IN THE TABLE.  SINCE THE TRUNCA-
C          TION ERROR IS LESS THAN 16 * ERRTOL ** 6 / (1 - 2 * ERRTOL),
C          DECREASING ERRTOL BY A FACTOR OF 10 YIELDS SIX MORE DECIMAL
C          DIGITS OF ACCURACY AT THE EXPENSE OF ONE OR TWO MORE ITERA-
C          TIONS OF THE DUPLICATION THEOREM.
C
C          SAMPLE CHOICES:  ERRTOL     RELATIVE TRUNCATION
C                                      ERROR LESS THAN
C                           1.0D-3     2.0D-17
C                           3.0D-3     2.0D-14
C                           1.0D-2     2.0D-11
C                           3.0D-2     2.0D-8
C                           1.0D-1     2.0D-5
C
C          ON OUTPUT:
C
C          X, Y, AND ERRTOL ARE UNALTERED.
C
C          IERR IS THE RETURN ERROR CODE:
C               IERR = 0 FOR NORMAL COMPLETION OF THE SUBROUTINE,
C               IERR = 1 FOR ABNORMAL TERMINATION.
C
C          ********************************************************
C          WARNING: CHANGES IN THE PROGRAM MAY IMPROVE SPEED AT THE
C          EXPENSE OF ROBUSTNESS.
C
      FLAG = Y.LT.-2.2360D0/DSQRT(LOLIM) .AND. X.GT.0.0D0 .AND.
     &     X.LT.((LOLIM*UPLIM)**2)/25.0D0
      IF (X.LT.0.0D0 .OR. Y.EQ.0.0D0 .OR. (X+DABS(Y)).LT.LOLIM .OR.
     &     (X+DABS(Y)).GT.UPLIM .OR. FLAG) THEN
         WRITE(PRINTR,104)
  104    FORMAT(1H0,42H*** ERROR - INVALID ARGUMENTS PASSED TO RC)
         WRITE(PRINTR,108) X,Y
  108    FORMAT(1H ,4HX = ,D23.16,4X,4HY = ,D23.16)
         IERR = 1
         RETURN
      END IF
C
      IERR = 0
      IF (Y.GT.0.0D0) THEN
           XN = X
           YN = Y
           W = 1.0D0
         ELSE
C          TRANSFORM TO POSITIVE Y
           XN = X - Y
           YN = - Y
           W = DSQRT(X) / DSQRT(XN)
      END IF
C
  116 MU = (XN + YN + YN) / 3.0D0
      SN = (YN + MU) / MU - 2.0D0
      IF (DABS(SN) .LT. ERRTOL) GO TO 120
      LAMDA = 2.0D0 * DSQRT(XN) * DSQRT(YN) + YN
      XN = (XN + LAMDA) * 0.250D0
      YN = (YN + LAMDA) * 0.250D0
      GO TO 116
C
```

```
  120 C1 = 1.0D0 / 7.0D0
      C2 = 9.0D0 / 22.0D0
      S = SN * SN * (0.30D0 + SN * (C1 + SN * (0.3750D0 + SN * C2)))
      RC = W * (1.0D0 + S) / DSQRT(MU)
C
      RETURN
      END

C*********************************************************************
C
      DOUBLE PRECISION FUNCTION RJ(X,Y,Z,P,ERRTOL,IERR)
C
C          THIS FUNCTION SUBROUTINE COMPUTES AN INCOMPLETE ELLIPTIC
C          INTEGRAL OF THE THIRD KIND,
C          RJ(X,Y,Z,P) = INTEGRAL FROM ZERO TO INFINITY OF
C
C                                                -1/2     -1
C                (3/2)[(T+X)(T+Y)(T+Z)]     (T+P)   DT,
C
C          WHERE X, Y, AND Z ARE NONNEGATIVE, AT MOST ONE OF THEM IS
C          ZERO, AND P IS NONZERO.  IF X OR Y OR Z IS ZERO, THE INTE-
C          GRAL IS COMPLETE.  IF P IS NEGATIVE, THE CAUCHY PRINCIPAL
C          VALUE IS COMPUTED BY USING A PRELIMINARY TRANSFORMATION
C          TO MAKE P POSITIVE; SEE EQUATION (2.22) OF THE SECOND REF-
C          ERENCE BELOW.  WHEN P IS POSITIVE, THE DUPLICATION THEOREM
C          IS ITERATED UNTIL THE VARIABLES ARE NEARLY EQUAL, AND THE
C          FUNCTION IS THEN EXPANDED IN TAYLOR SERIES TO FIFTH ORDER.
C          REFERENCES: B. C. CARLSON AND E. M. NOTIS, ALGORITHMS FOR
C          INCOMPLETE ELLIPTIC INTEGRALS, ACM TRANSACTIONS ON MATHEMA-
C          TICAL SOFTWARE, 7 (1981), 398-403; B. C. CARLSON, COMPUTING
C          ELLIPTIC INTEGRALS BY DUPLICATION, NUMER. MATH. 33 (1979),
C          1-16.
C          AUTHORS: B. C. CARLSON AND ELAINE M. NOTIS, AMES LABORATORY-
C          DOE, IOWA STATE UNIVERSITY, AMES, IA 50011, AND R. L. PEXTON,
C          LAWRENCE LIVERMORE NATIONAL LABORATORY, LIVERMORE, CA 94550.
C          AUG. 1, 1979, REVISED SEPT. 1, 1987.
C
C          CHECK VALUES: RJ(2,3,4,5) = 0.14297 57966 71567 53833 23308
C                        RJ(2,3,4,-5) = - 0.12711 23004 29638 83590 80083
C          CHECK BY ADDITION THEOREM: RJ(X,X+Z,X+W,X+P)
C          + RJ(Y,Y+Z,Y+W,Y+P) + (A-B) * RJ(A,B,B,A) + 3 / DSQRT(A)
C          = RJ(0,Z,W,P), WHERE X,Y,Z,W,P ARE POSITIVE AND X * Y
C          = Z * W,  A = P * P * (X+Y+Z+W),  B = P * (P+X) * (P+Y),
C          AND B - A = P * (P-Z) * (P-W).  THE SUM OF THE THIRD AND
C          FOURTH TERMS ON THE LEFT SIDE IS 3 * RC(A,B).
C
      INTEGER IERR,PRINTR
      DOUBLE PRECISION A,B,ALFA,BETA,C1,C2,C3,C4,EA,EB,EC,E2,E3,EPSLON
      DOUBLE PRECISION ERRTOL,ETOLRC,LAMDA,LOLIM,MU,P,PN,PNDEV,POWER4
      DOUBLE PRECISION RC,RCX,RF,RHO,SIGMA,S1,S2,S3,TAU,UPLIM,X,XN,XNDEV
      DOUBLE PRECISION XNROOT,Y,YY,YN,YNDEV,YNROOT,Z,ZN,ZNDEV,ZNROOT
C          INTRINSIC FUNCTIONS USED: DABS,DMAX1,DMIN1,DSQRT
C
C          PRINTR IS THE UNIT NUMBER OF THE PRINTER.
      DATA PRINTR/6/
C
C          RC AND RF ARE FUNCTIONS COMPUTED BY EXTERNAL SUBROUTINES.
C
C          LOLIM DETERMINES THE LOWER LIMIT AND UPLIM THE UPPER LIMIT
C          OF THE RANGE OF ADMISSIBLE VALUES OF X,Y,Z,P FOR WHICH
C          THE COMPUTATION WILL PROCEED WITHOUT UNDERFLOW OR OVERFLOW.
C          LOLIM IS NOT LESS THAN THE CUBE ROOT OF THE VALUE
C          OF LOLIM USED IN THE SUBROUTINE FOR RC.
C          UPLIM IS NOT GREATER THAN 0.3 TIMES THE CUBE ROOT OF
C          THE VALUE OF UPLIM USED IN THE SUBROUTINE FOR RC.
C
C          ACCEPTABLE VALUES FOR:   LOLIM        UPLIM
C          IBM 360/370 SERIES   :   2.0D-26      3.0D+24
C          CDC 6000/7000 SERIES :   5.0D-98      3.0D+106
C          UNIVAC 1100 SERIES   :   5.0D-103     6.0D+101
C          CRAY                 :   2.90D-813    1.10D+821
C          VAX 11 SERIES        :   2.50D-13     9.0D+11
C          IBM PC               :   2.50D-13     9.0D+11
C
C          WARNING: IF THIS PROGRAM IS CONVERTED TO SINGLE PRECISION,
C          THE VALUES FOR THE UNIVAC 1100 SERIES SHOULD BE CHANGED TO
C          LOLIM = 5.0E-13 AND UPLIM = 6.0E+11 BECAUSE THE MACHINE
C          EXTREMA CHANGE WITH THE PRECISION.
C
      DATA LOLIM/2.50D-13/, UPLIM/9.0D+11/
C
C          ON INPUT:
C
C          X, Y, Z, AND P ARE THE VARIABLES IN THE INTEGRAL RJ(X,Y,Z,P).
C
C          ERRTOL IS CHOSEN TO DETERMINE THE ACCURACY OF THE COMPUTED
C          APPROXIMATION TO THE INTEGRAL.  TRUNCATION OF A TAYLOR SERIES
C          AFTER TERMS OF FIFTH ORDER INTRODUCES A RELATIVE ERROR LESS
C          THAN THE AMOUNT SHOWN IN THE SECOND COLUMN OF THE FOLLOWING
C          TABLE FOR EACH VALUE OF ERRTOL IN THE FIRST COLUMN.  IN ADDI-
C          TION TO THE TRUNCATION ERROR THERE WILL BE ROUNDOFF ERROR,
C          BUT IN PRACTICE THE TOTAL ERROR FROM BOTH SOURCES IS USUALLY
C          LESS THAN THE AMOUNT GIVEN IN THE TABLE.  SINCE THE TRUNCA-
C          TION ERROR IS LESS THAN 3 * ERRTOL ** 6 / (1-ERRTOL) ** 3/2,
C          DECREASING ERRTOL BY A FACTOR OF 10 YIELDS SIX MORE DECIMAL
C          DIGITS OF ACCURACY AT THE EXPENSE OF ONE OR TWO MORE ITERA-
C          TIONS OF THE DUPLICATION THEOREM.
C          ERROR TOLERANCES (ETOLRC AND ERRTOL) WILL BE PASSED TO THE
C          SUBROUTINES FOR RC AND RF TO MAKE THE TRUNCATION ERROR FOR
C          RC AND RF LESS THAN FOR RJ.
C
```

```
C         SAMPLE CHOICES:  ERRTOL      RELATIVE TRUNCATION
C                                      ERROR LESS THAN
C                          1.0D-3      4.0D-18
C                          3.0D-3      3.0D-15
C                          1.0D-2      4.0D-12
C                          3.0D-2      3.0D-9
C                          1.0D-1      4.0D-6
C
C         ON OUTPUT:
C
C         X, Y, Z, P, AND ERRTOL ARE UNALTERED.
C
C         IERR IS THE RETURN ERROR CODE:
C              IERR = 0 FOR NORMAL COMPLETION OF THE SUBROUTINE,
C              IERR = 1 FOR ABNORMAL TERMINATION.
C
C         ********************************************************
C         WARNING: CHANGES IN THE PROGRAM MAY IMPROVE SPEED AT THE
C         EXPENSE OF ROBUSTNESS.
C
      IF (DMIN1(X,Y,Z).LT.0.0D0 .OR. DMIN1(X+Y,X+Z,Y+Z,DABS(P))
     &      .LT.LOLIM .OR. DMAX1(X,Y,Z,DABS(P)).GT.UPLIM) THEN
         WRITE(PRINTR,102)
  102    FORMAT(1H0,42H*** ERROR - INVALID ARGUMENTS PASSED TO RJ)
         WRITE(PRINTR,104) X,Y,Z,P
  104    FORMAT(1H ,4HX = ,D23.16,4X,4HY = ,D23.16,4X,4HZ = ,D23.16,
     &      4X,4HP = ,D23.16)
         IERR = 1
         RETURN
      END IF
C
      IERR = 0
      ETOLRC = 0.50D0 * ERRTOL

      IF (P.GT.0.0D0) THEN
          XN = X
          YN = Y
          ZN = Z
          PN = P
         ELSE
C         ORDER X,Y,Z AND TRANSFORM TO POSITIVE P
          XN = DMIN1(X,Y)
          YY = DMAX1(X,Y)
          ZN = DMAX1(YY,Z)
          YY = DMIN1(YY,Z)
          YN = DMAX1(XN,YY)
          XN = DMIN1(XN,YY)
          A = 1.0D0 / (YN - P)
          B = (ZN - YN) * A * (YN - XN)
          PN = YN + B
          RHO = XN * ZN / YN
          TAU = P * PN / YN
          RCX = RC(RHO,TAU,ETOLRC,IERR)
          IF (IERR .NE. 0) RETURN
         END IF
C
      SIGMA = 0.0D0
      POWER4 = 1.0D0
C
  116 MU = (XN + YN + ZN + PN + PN) * 0.20D0
      XNDEV = (MU - XN) / MU
      YNDEV = (MU - YN) / MU
      ZNDEV = (MU - ZN) / MU
      PNDEV = (MU - PN) / MU
      EPSLON = DMAX1(DABS(XNDEV),DABS(YNDEV),DABS(ZNDEV),DABS(PNDEV))
      IF (EPSLON .LT. ERRTOL) GO TO 120
      XNROOT = DSQRT(XN)
      YNROOT = DSQRT(YN)
      ZNROOT = DSQRT(ZN)
      LAMDA = XNROOT * (YNROOT + ZNROOT) + YNROOT * ZNROOT
      ALFA = PN * (XNROOT + YNROOT + ZNROOT) + XNROOT * YNROOT * ZNROOT
      ALFA = ALFA * ALFA
      BETA = PN * (PN + LAMDA) * (PN + LAMDA)
      SIGMA = SIGMA + POWER4 * RC(ALFA,BETA,ETOLRC,IERR)
      IF (IERR .NE. 0) RETURN
      POWER4 = POWER4 * 0.250D0
      XN = (XN + LAMDA) * 0.250D0
      YN = (YN + LAMDA) * 0.250D0
      ZN = (ZN + LAMDA) * 0.250D0
      PN = (PN + LAMDA) * 0.250D0
      GO TO 116
C
  120 C1 = 3.0D0 / 14.0D0
      C2 = 1.0D0 / 3.0D0
      C3 = 3.0D0 / 22.0D0
      C4 = 3.0D0 / 26.0D0
      EA = XNDEV * (YNDEV + ZNDEV) + YNDEV * ZNDEV
      EB = XNDEV * YNDEV * ZNDEV
      EC = PNDEV * PNDEV
      E2 = EA - 3.0D0 * EC
      E3 = EB + 2.0D0 * PNDEV * (EA - EC)
      S1 = 1.0D0 + E2 * (- C1 + 0.750D0 * C3 * E2 - 1.50D0 * C4 * E3)
      S2 = EB * (0.50D0 * C2 + PNDEV * (- C3 - C3 + PNDEV * C4))
      S3 = PNDEV * EA * (C2 - PNDEV * C3) - C2 * PNDEV * EC
      RJ = 3.0D0 * SIGMA + POWER4 * (S1 + S2 + S3) / (MU * DSQRT(MU))
C
      IF (P .GT. 0.0D0) RETURN
      RJ = A * (B * RJ + 3.0D0 * (RCX - RF(XN,YN,ZN,ERRTOL,IERR)))
C
      RETURN
      END
```

Numerical Checks. The 31 formulas in Section 2 and the 10 in Section 3 were checked numerically when $x = 2.0$, $y = 0.5$, $a_i = 0.1+0.2i$, $b_i = 0.5-0.2i$, $1 \le i \le 4$, $a_5 = 0.8$, $b_5 = 1$. The functions R_J and R_C in (2.15) take their Cauchy principal values, while those in (2.16) have a positive last argument. In each of the 41 formulas the integral on the left side, defined by (2.18), was integrated numerically by the SLATEC code QNG. On the right-hand side, I_1, I_2, I_3, and I_3' were calculated from (2.13) to (2.16) by using the codes for R-functions in the Supplements to this paper and [4]. The remaining calculations were done with a hand calculator. For each of the 41 formulas the values obtained for the two sides agreed to better than one part in a million.

Some of the intermediate values in these calculations are listed here:

$U_{12}^2 = 0.24791575$, $W^2 = -0.35688425$, $W_1^2 = 0.21911575$,
$U_{13}^2 = 0.20471575$, $P^2 = 3.0168477$, $P_1^2 = 0.55102655$,
$U_{14}^2 = 0.19031575$, $Q^2 = -3.2075523$, $Q_1^2 = 0.54102655$,

$R_C(P^2,Q^2) = 0.34465425$,
$R_C(P_1^2,Q_1^2) = 1.3553823$,
$R_F(U_{12}^2,U_{13}^2,U_{14}^2) = 2.1642326$, $I_1 = 4.3284652$,
$R_D(U_{12}^2,U_{13}^2,U_{14}^2) = 10.860876$, $I_2 = 6.3592902$,
$R_J(U_{12}^2,U_{13}^2,U_{14}^2,W^2) = -6.4342997$, $I_3 = 1.4305398$,
$R_J(U_{12}^2,U_{13}^2,U_{14}^2,W_1^2) = 9.9863171$, $I_3' = 2.9408494$,

$A(1,1,1,-1) = 0.56155363$, $A(1,1,1,-1,-2) = 0.0096998758$,
$A(1,1,1,1) = -0.039947562$, $A(1,1,1,1,-2) = -0.15740823$,
$A(1,1,1,-3) = 2.7981283$, $A(3,1,1,1) = 0.12035743$,
$A(1,1,-1,-1) = 1.3368648$

As a check on Section 6 the Cauchy principal value of I_3 was computed with the same values of $x,y,a_1,\ldots,a_4,b_1,\ldots,b_4$ used previously but with $a_5+b_5t = -1+t$ so that the integrand has a simple pole on the interval of integration. The result obtained from (2.15), in which I_3 and R_C take their principal values, is $I_3 = 2.55226304$. To check this, the integral over the interval $(0.5,2)$ was approximated by the sum of the integrals over $(0.5,1-10^{-9})$ and $(1+10^{-9},2)$, and the first of these two was replaced by the negative of the integral in which $a_5+b_5t = 1-t > 0$. No principal values are involved in computing the two integrals by (2.15), and the result is $-32.991526227 + 35.543789264 = 2.55226304$, in agreement with the first computation.