NPS-53-89-014

NAVAL POSTGRADUATE SCHOOL

Monterey, California

FORTRAN SUBROUTINES FOR THE EVALUATION OF THE CONFLUENT HYPERGEOMETRIC FUNCTIONS

WILLIAM GRAGG

BENY NETA

August 1989

Prepared for:
Naval Postgraduate School
Monterey, CA 93943

NAVAL POSTGRADUATE SCHOOL
Department of Mathematics

Rear Admiral R. W. West, JR.
Superintendent

Harrison Shull
Provost

This report was prepared in conjunction with research funded by the Naval Postgraduate School Research Council.

Prepared by:

Prepared by:

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NPS-53-89=014	5 MONITORING ORGANIZATION REPORT NUMBER(S) NPS-53-89-014
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School — 6b OFFICE SYMBOL (If applicable) 53	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943	7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943
8a NAME OF FUNDING/SPONSORING ORGANIZATION Naval Postgraduate School — 8b OFFICE SYMBOL (If applicable) 53	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER O&MN, Direct Funding
8c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

FORTRAN SUBROUTINES FOR THE EVALUATION OF THE CONFLUENT HYPERGEOMETRIC FUNCTIONS

12 PERSONAL AUTHOR(S)
William Gragg and Beny Neta

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Technical Report	FROM 2/89 TO 8/89	89 August 14	12

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	confluent hypergeometric functions, stable algorithm Fortran subroutine, recurrence relation

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

In this report we list the Fortran subroutines for evaluating the confluent hypergeometric functions M(a,b;x) and U(a,b;x). These subroutines use the stable recurrence relations given e.g. in Wimp.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL William Gragg and Beny Neta	22b TELEPHONE (Include Area Code) (408) 646-2194,2235 — 22c OFFICE SYMBOL 53Gr and 53Nd

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Fortran Subroutines for the Evaluation of the Confluent Hypergeometric Functions

W. Gragg

and

B. Neta

Naval Postgraduate School
Department of Mathematics
Monterey, CA 93943

Abstract

In this report we list the Fortran subroutines for evaluating the confluent hypergeometric functions M(a,b;x) and U(a,b;x). These subroutines use the stable recurrence relations given e.g. in Wimp.

Key words:
confluent hypergeometric functions
stable algorithm
Fortran subroutine
recurrence relation

Introduction

It is well known that the ordinary differential equation

$$x \frac{d^2y}{dx^2} + (1-x) \frac{dy}{dx} - ay = 0$$

has a solution

$$y(x) = AM(a,1;x) + BU(a,1;x)$$

if a is not a negative integer.

This problem arises e.g. when solving the linearized shallow water equations with the full linear variation in depth included (see Williams, Staniforth and Neta, [1]).

The computation of the confluent hypergeometric functions is based on the Miller algorithm (see e.g. Wimp, [2]). In general, one has a second order difference equation

$$z(n) + a(n)z(n+1) + b(n)z(n+2) = 0, \quad n \geq 0, \quad b(n) \neq 0 .$$

If $b(n) = 0$ for some n, in some cases one can make a change of variable $Y(n) = \lambda(n)z(n)$ which will produce an equation of the desired type. Let $w(n)$ be a nontrivial solution and the sum of the normalizing series

$$S = \sum_{k=0}^{\infty} c(k)w(k) \neq 0$$

is known. Let N be a large integer and define $z_N(n)$, $0 \le n \le N+1$, by

$$z_N(n) = \begin{cases} 0 & n = N+1 \\ 1 & n = N \end{cases}$$

$$z_N(n) + a(n)z_N(n+1) + b(n)z_N(n+2) = 0, \quad n = N-1, \ldots, 1, 0 .$$

One can approximate $w(n)$ by $w_N(n)$

$$w_N(n) = Sz_N(n)/S_N$$

where

$$S_N = \sum_{k=0}^{N} c(k)z_N(k).$$

The algorithm is said to converge if

$$w_N(n) \to w(n) \quad \text{as} \quad N \to \infty .$$

The function $M(a,b;x)$ satisfies the recurrence relation

$$(2n+b+2)(n+a)z(n) - (2n+b+1)\left\{(2a-b) + \frac{(2n+b)(2n+b+2)}{x}\right\}z(n+1)$$
$$- (2n+b)(n+b+1-a)z(n+2) = 0 .$$

The minimal solution is

$$w(n) = \frac{x^n (a)_n}{(b)_{2n}} M(a+n, 2n+b; x)$$

where

$$(c)_n = \frac{\Gamma(n+c)}{\Gamma(c)} .$$

The normalization relationship used in our subroutine is

$$S = b-1 = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!}(b-1)_k(b+2k-1)w(k) .$$

An obvious modification must be made if b = 1. The algorithm is not defined if b, b+1-a, a are negative integers or zero.

The function U(a,b;x) satisfies the relationship

$$(n+a)(n+a+1-b)z(n) - (n+1)[2(n+a+1)+x-b]z(n+1)$$
$$+ (n+1)(n+2)z(n+2) = 0 .$$

The minimal solution is

$$w(n) = \frac{x^n(a)_n(a+1-b)_n}{n} U(a+n,b;x)$$

for $|\arg x| < \pi$. A normalization relation is

$$1 = \sum_{k=0}^{\infty} w(k) .$$

In the next section we give a listing of the Fortran subroutines.

Subroutine Miller

```
      SUBROUTINE MILLER(N,ALPHA,BETA,X,S,SS,COEFF)
      INTEGER N
      REAL*8  ALPHA,BETA,X,SS
      REAL*8  S(0:1000)
      EXTERNAL COEFF
C     USES THE J.C.P. MILLER ALGORITHM TO COMPUTE
C     S(0:N).
C     BEGIN
         INTEGER NN,K
         REAL*8 T,D,EPS,A,B,C
         REAL*8 OLDS(0:1000)
         EPS = 0.000000001
C        INITIALIZE OLDS.
         DO 1 K = 0, 1000
            OLDS(K) = 0
   1     CONTINUE
C        CHOOSE INITIAL NN.
         NN = N + 2
C        INITIALIZE K, S AND T.
   2     K = NN
         S(K+1) = 0
         S(K)   =1
         CALL COEFF(K,ALPHA,BETA,X,A,B,C)
         T      = 2*C*S(K)
C        TAKE A BACKWARD RECURRENCE STEP AND UPDATE IT.
   3     K = K - 1
         CALL COEFF(K,ALPHA,BETA,X,A,B,C)
         S(K) = A*S(K+1) + B*S(K+2)
C        CHECK FOR OVERFLOW AND RESCALE IF NECESSARY.
         D= DABS(S(K))
         IF (D .GT. 1.D30) THEN
C        BEGIN
            CALL SCALE(K,NN,S,T,D)
         END IF
         IF (K .GT. 0) THEN
C        BEGIN
            T = T + 2*C*S(K)
            GO TO 3
         END IF
         T = T + C*S(0)
         DO 4 K = 0, N
            S(K) = S(K)/T
   4     CONTINUE
C        TEMPORARY PRINT STATEMENT.
C        PRINT*, S(0)
C        TEST FOR CONVERGENCE.
         D = 0
         DO 5 K = 0, N
   5        D = D + S(K)**2
         CONTINUE
         D = DSQRT(D)
         T = 0
```

```
         DO 6 K = 0, N
            T = T + (S(K) - OLDS(K))**2
    6    CONTINUE
         T = DSQRT(T)
C        TAKE ANOTHER STEP IF NO CONVERGENCE.
         IF (T .GT. EPS*D) THEN
C        BEGIN
            NN = 2*NN
            DO 7 K = 0, N
               OLDS(K) = S(K)
    7       CONTINUE
            IF(NN .LE. 1000) GO TO 2
         PRINT 999,NN,ALPHA,BETA,X,T
999      FORMAT(' ** NO CONVERGENCE ** NN AP CP X T ',I5,4E14.7)
        END IF
        SS=S(0)
        RETURN
      END
```

```
      SUBROUTINE COEFF(N,ALPHA,BETA,X,A,B,C)
      INTEGER N
      REAL*8 ALPHA,BETA,X,A,B,C
C     COMPUTES COEFFICIENTS USED BY J.C.P. MILLER ALGORITHM FOR
C     A CONFLUENT HYPERGEOMETRIC FUNCTION  M(a,b:x)
C     SEE JET WIMP. COMPUTATION WITH RECURRENCE RELATIONS.
C     PITMAN 1984 PP. 61-62
C     BEGIN
         INTEGER M,K
         REAL*8,T,U,V,W
         S = 2*ALPHA - BETA
         T = N + ALPHA
         M = 2*N
         U = M + BETA
         V = U + 1
         W = V + 1
         A = (S/W + U/X)*V/T
         B = (N + BETA - ALPHA + 1)*U/T/W
         T = 1
         IF (N .GT. 0) THEN
C        BEGIN
            S = BETA - 1
            DO 1 K = 1, N-1
               T = -T*(1+S/K)
   1        CONTINUE
            T = -T*(1+S/M)
         END IF
         C = T
         RETURN
      END




      SUBROUTINE SCALE(K,N,S,T,D)
      INTEGER N,K
      REAL*8 T,D
      REAL*8 S(0:1000)
C     BEGIN
         INTEGER J
         T = T/D
         DO 1 J = K, N
            S(J) = S(J)/D
   1     CONTINUE
         RETURN
      END
```

```
      SUBROUTINE COEFU(N,ALPHA,BETA,X,A,B,C)
      INTEGER N
      REAL*8 ALPHA, BETA,X,A,B,C
C     COMPUTES COEFFICIENTS USED BY J.C.P. MILLER ALGORITHM FOR
C     A CONFLUENT HYPERGEOMETRIC FUNCTION  U(a,b;x)
C     SEE JET WIMP, COMPUTATION WITH RECURRENCE RELATIONS,
C     PITMAN 1984 PP. 63-64
C     BEGIN
         INTEGER M,K
         REAL*8  S,T,U,V,W
         S =   ALPHA + QFLOAT(N)
         T = S + 1.DO
         U = S*(T - BETA)
         V = QFLOAT(N + 1)
         W = V + 1.DO
         A = (2*T + X - BETA)*V/U
         B = - V*W/U
         C = 1
         RETURN
      END
```

Remark: The program that calls Miller must supply as a last parameter either COEFF (for M) or COEFU (for U).

The subroutines are available on a diskette from either author upon request. These subroutines were tested extensively for various values of a, b and x.

Remark: If the parameter is a negative integer, the solution of the differential equation is

$$y = AL_n(x) + B\{\ln|x|L_n(x) + \sum_{m=0}^{\infty} \beta_m x^m\}$$

where n = -a.

$L_n(x)$ are Laguerre polynomials whose coefficients α_i satisfy

$$\alpha_i = \frac{i-n-1}{i^2}\alpha_{i-1}, \qquad i = 2, \ldots, n.$$

$$\alpha_1 = -n.$$

The coefficients β_m satisfy

$$\beta_{m+1} = \frac{(m-n)\beta_m + \left(1 - \frac{2(m-n)}{m+1}\alpha_m\right)}{(m+1)^2} \qquad m = 1, \ldots, n-1$$

$$\beta_m = \frac{1}{(n+1)^2}\alpha_n \qquad m = n$$

$$\beta_m = \frac{m-n-1}{m^2}\beta_{m-1} \qquad m = n+1, n+2, \ldots$$

Acknowledgement:

This research was conducted for the Office of Naval Research and was funded by the Naval Postgraduate School.

References

1. R.T. Williams, A.N. Staniforth and B. Neta, Solution of a generalized Sturm-Liouville Problem, IMA Conference on Computational Ordinary Differential Equations, Imperial College, London, July 3-7, 1989.

2. J. Wimp. Computation with Recurrence Relations, Pitman Advanced Pub. Program, Boston, 1984.

DISTRIBUTION LIST

	NO. OF COPIES
Director Defense Tech. Information Center Cameron Station Alexandria, VA 22314	2
Director of Research Administration Code 012 Naval Postgraduate School Monterey, CA 93943	1
Library Code 0142 Naval Postgraduate School Monterey, CA 93943	2
Department of Mathematics Code 53 Naval Postgraduate School Monterey, CA 93943	1
Center for Naval Analyses 4401 Ford Avenue Alexandria, VA 22302-0268	1
Professor Beny Neta Code 53Nd Department of Mathematics Naval Postgraduate School Monterey, CA 93943	6
Dr. C.P. Katti J. Nehru University School of Computer & Systems Sciences New Delhi 110067 India	5
Professor Paul Nelson Texas A&M University Department of Nuclear Engineering and Mathematics College Station, TX 7 843-3133	1
Professor H.B. Keller Department of Applied Mathematics California Institute of Technology Pasadena, CA 91125	1

Professor W. Gragg 6
Code 53Gr
Department of Mathematics
Naval Postgraduate School
Monterey, CA 93943

Profesor H. Dean Victory, Jr. 1
Texas Tech University
Department of Mathematics
Lubbock, TX 79409

Professor Gordon Latta 1
Code 53Lz
Department of Mathematics
Naval Postgraduate School
Monterey, CA 93943

Professor Arthur Schoenstadt 1
Code 53Zh
Department of Mathematics
Naval Postgraduate School
Monterey, CA 93943

Professor M.M. Chawla, Head 1
Department of Mathematics
III/III/B-1, IIT Campus
Hauz Khas, New Delhi 110016
India

Professor R.T. Williams 1
Code 63Wu
Department of Meteorology
Naval Postgraduate School
Monterey, CA 93943