JOURNAL OF COMPUTATIONAL PHYSICS 51, 502–518 (1983)

Associated Legendre Functions on the Cut

F. W. J. OLVER

Institute for Physical Science and Technology, University of Maryland, College Park, Maryland 20742 and National Bureau of Standards, Washington, D. C. 20234

AND

J. M. SMITH

George Mason University, Fairfax, Virginia 22030 and National Bureau of Standards, Washington D. C. 20234

Received September 21, 1982; revised January 13, 1983

Algorithms and a code are described for the computation of the associated Legendre functions $P_\nu^\mu(\cos\theta)$, $P_\nu^{-\mu}(\cos\theta)$, $Q_\nu^\mu(\cos\theta)$ and the normalized Legendre polynomial $\bar{\mathcal{P}}_\nu^\mu(\cos\theta)$ in the ranges $0 < \theta \leqslant \frac{1}{2}\pi$, $\mu = 0, 1, 2, \ldots, -\frac{1}{2} \leqslant \nu < \infty$. The algorithms are based on power-series expansions and recurrence relations. They are executed in extended-range arithmetic, thereby admitting very extensive ranges of μ and ν without causing overflow or underflow.

1. INTRODUCTION AND SUMMARY

The associated Legendre equation is given by

$$(1-x^2)\frac{d^2w}{dx^2} - 2x\frac{dw}{dx} + \left\{\nu(\nu+1) - \frac{\mu^2}{1-x^2}\right\} w = 0. \tag{1.1}$$

Here μ and ν are parameters, usually known as the *order* and *degree*, respectively. In [10] Gautschi furnished a package for generating solutions of (1.1) in the interval $(1, \infty)$ for extensive real ranges of μ and ν, and also purely imaginary values of $\nu + \frac{1}{2}$. Other algorithms, covering other ranges of the variables, will be found in [6, 7, 9, 11, 12, 22]. The present object is to provide a comprehensive package for generating solutions in the interval $(-1, 1)$, again for real values of μ and ν. This interval joins the singularities of (1.1) at $x = \pm 1$ and is often called the *cut* even though standard solutions have long been defined that are real- and single-valued there.

In physical applications, for example [8, 15, 18], the commonest real values of the parameters are those for which μ and 2ν are integers. In the case of the degree no

0021-9991/83 $3.00

significant complications arise in our package by permitting ν to assume any real value such that $\nu \geqslant -\frac{1}{2}$, and we allow for this possibility. In the case of the order, however, we restrict μ to nonnegative integer values.

In constructing the package we have sought robust algorithms that will cover extensive ranges of values of μ and ν, and also admit values of x very close to the singularities at ± 1. This has been achieved by the use of fairly simple mathematical methods, namely, power-series expansions and recurrence relations. Direct implementation of these methods in ordinary floating-point arithmetic would encounter frequent failures owing to overflow or underflow. To overcome these difficulties without the use of awkward mathematical devices such as the computation of ratios or logarithms of function values, we employ the extended-range arithmetic subroutines that were introduced in [13, 20]. In this arithmetic a separate storage location is allocated to the exponent of each floating-point number. Subsequently, computed function values are converted back to ordinary floating-point form whenever this is possible. The main price that was paid by adopting this approach in [13] and [20] to compute normalized Legendre polynomials was an increase in execution time by a factor of about two (on the UNIVAC 1108) when compared with ordinary floating-point arithmetic. A similar price is paid in the present package. The gain, however, is an enormous increase in the range of values of x, μ, and ν that can be accommodated.

The paper is arranged as follows. In Section 2 we discuss the choice of standard solutions of (1.1). In Section 3 we describe recursion methods for generating these solutions and discuss their stability. In Section 4 we give power series for computing the solutions for small values of ν, together with estimates of the errors incurred when these series are truncated. In Section 5 we give explicit formulas for cross products of solutions that can be used as checks. The complete package is described in Section 6, and the concluding section, Section 7, provides an account of tests that have been carried out.

To obtain a complete code for the package, including a test program, users should write to the Computer Products Office, National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia 22161, and quote Accession No. PB 82–250853.

2. Choice of Solutions

Definitions and properties of solutions of (1.1) are supplied in [2, Chap. 3; 16, Chap. 8; 17, Chap. 5], and we employ the notation adopted in these references. Two solutions are the *Ferrers functions* $P_\nu^\mu(x)$, $Q_\nu^\mu(x)$, defined by

$$P_\nu^\mu(x) = \frac{1}{\Gamma(1-\mu)} \left(\frac{1+x}{1-x}\right)^{\mu/2} F\left(-\nu, \nu+1; 1-\mu; \frac{1}{2} - \frac{1}{2}x\right), \tag{2.1}$$

$$Q_\nu^\mu(x) = \frac{1}{2}\Gamma(\mu)\cos(\mu\pi)\left(\frac{1+x}{1-x}\right)^{\mu/2} F\left(-\nu, \nu+1; 1-\mu; \frac{1}{2}-\frac{1}{2}x\right)$$
$$+\frac{1}{2}\Gamma(-\mu)\frac{\Gamma(\nu+\mu+1)}{\Gamma(\nu-\mu+1)}\left(\frac{1-x}{1+x}\right)^{\mu/2} F\left(-\nu, \nu+1; 1+\mu; \frac{1}{2}-\frac{1}{2}x\right), \quad (2.2)$$

where F denotes the hypergeometric function in the usual notation. When μ is zero or an integer the right-hand side of (2.2) is to be replaced by its limiting value.

Other solutions of (1.1) include $P_\nu^{\pm\mu}(\pm x)$, $Q_\nu^{\pm\mu}(\pm x)$, $P_{-\nu-1}^{\pm\mu}(\pm x)$, and $Q_{-\nu-1}^{\pm\mu}(\pm x)$, and connection formulas are available to express any of these solutions in terms of any linearly independent pair of the others. Accordingly, we may restrict the variable and parameters to the ranges

$$0 \leqslant x < 1, \ \mu \geqslant 0, \ \nu \geqslant -\tfrac{1}{2}. \quad (2.3)$$

However, the chosen standard solutions must comprise a numerically satisfactory pair; this means that excessive cancellation will not occur in computing any other solution from this pair, except of course in the immediate neighborhoods of zeros of the required solution. This problem was investigated in [17], and from the discussion on p. 186 of [17] it follows that if x, μ, and ν are restricted by (2.3) and μ is an integer, then $P_\nu^{-\mu}(x)$ and $Q_\nu^\mu(x)$ comprise a numerically satisfactory pair.[1] We therefore adopt these solutions as standard.

We also provide for the calculation of $P_\nu^\mu(x)$, and (in the case of integer ν only) the so-called normalized Legendre polynomials $\bar{\mathcal{P}}_\nu^\mu(x)$.[2] When μ is an integer $P_\nu^\mu(x)$ is related to $P_\nu^{-\mu}(x)$ by

$$P_\nu^\mu(x) = (-)^\mu\,(\nu-\mu+1)(\nu-\mu+2)\cdots(\nu+\mu)\,P_\nu^{-\mu}(x). \quad (2.4)$$

And when μ and ν are nonnegative integers $\bar{\mathcal{P}}_\nu^\mu(x)$ is defined by

$$\bar{\mathcal{P}}_\nu^\mu(x) = (-)^\mu\left\{\left(\nu+\frac{1}{2}\right)\frac{(\nu-\mu)!}{(\nu+\mu)!}\right\}^{1/2} P_\nu^\mu(x), \quad (2.5)$$

or, equivalently,

$$\bar{\mathcal{P}}_\nu^\mu(x) = \left\{\left(\nu+\frac{1}{2}\right)\frac{(\nu+\mu)!}{(\nu-\mu)!}\right\}^{1/2} P_\nu^{-\mu}(x). \quad (2.6)$$

[1] The reason that $P_\nu^\mu(x)$ and $Q_\nu^\mu(x)$ are not adopted as the fundamental pair is that $P_\nu^\mu(x)$ vanishes identically when $\mu - \nu$ is positive integer; compare (2.4), below.

[2] If only the $\bar{\mathcal{P}}_\nu^\mu(x)$ are required, then the user may prefer the algorithm described in [13, 20].

3. Recursion Methods

Recurrence formulas used to compute $P_\nu^{-\mu}(x)$ and $Q_\nu^\mu(x)$ are the ν-wise relations

$$(\nu+\mu+1)\,P_{\nu+1}^{-\mu}(x)-(2\nu+1)\,xP_\nu^{-\mu}(x)+(\nu-\mu)\,P_{\nu-1}^{-\mu}(x)=0, \tag{3.1}$$

$$(\nu-\mu+1)\,Q_{\nu+1}^{\mu}(x)-(2\nu+1)\,xQ_\nu^{\mu}(x)+(\nu+\mu)\,Q_{\nu-1}^{\mu}(x)=0, \tag{3.2}$$

and the μ-wise relations

$$(\nu-\mu)(\nu+\mu+1)\,P_\nu^{-(\mu+1)}(x)-2\mu x(1-x^2)^{-1/2}\,P_\nu^{-\mu}(x)+P_\nu^{-(\mu-1)}(x)=0, \tag{3.3}$$

$$Q_\nu^{\mu+1}(x)+2\mu x(1-x^2)^{-1/2}\,Q_\nu^{\mu}(x)+(\nu+\mu)(\nu-\mu+1)\,Q_\nu^{\mu-1}(x)=0. \tag{3.4}$$

In each of these equations we may replace μ by $-\mu$ throughout; thus $Q_\nu^\mu(x)$ satisfies the same equations as $P_\nu^\mu(x)$, and $Q_\nu^{-\mu}(x)$ satisfies the same equations as $P_\nu^{-\mu}(x)$.

The stability of these recurrence processes may be discussed by analogy with the general homogeneous linear difference equation of the second order with real and constant coefficients, given by

$$aw_{j+1}+2bw_j+cw_{j-1}=0, \qquad ac\neq 0. \tag{3.5}$$

If α_1 and α_2 denote the (real or complex) roots of the corresponding characteristic equation

$$a\alpha^2+2b\alpha+c=0, \tag{3.6}$$

then linearly independent solutions of (3.5) are α_1^j and α_2^j if $\alpha_1\neq\alpha_2$, or α_1^j and $j\alpha_1^j$ if $\alpha_1=\alpha_2$. There are three distinct cases:

(i) If $b^2-ac>0$ and $b\neq 0$, then α_1 and α_2 are real and $|\alpha_1|\neq|\alpha_2|$. As j increases only the most rapidly growing solution, corresponding to the greater of $|\alpha_1|$ and $|\alpha_2|$, can be computed in a stable manner. Similarly as j decreases only the solution corresponding to the lesser of $|\alpha_1|$ and $|\alpha_2|$ can be computed in a stable manner.

(ii) If $b^2-ac=0$, then α_1 and α_2 are real and equal. Again one solution grows more rapidly than the other, but the relative rate of growth is algebraic rather than geometric, consequently instabilities caused by a "wrong" choice of recurrence direction are relatively mild.

(iii) If $b^2-ac<0$ or if $b=0$, then $|\alpha_1|=|\alpha_2|$ but $\alpha_1\neq\alpha_2$. Either solution may be recurred in either direction in a stable manner.

The stability of the recurrence relations (3.1)–(3.4) may be inferred by regarding these equations as having relatively slowly changing coefficients, an assumption that

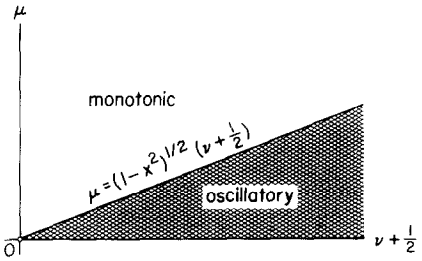

FIG. 1. Stable and unstable recurrence regions.

is justified when μ or ν is large. In each case the "local" discriminant is positive, zero, or negative approximately when

$$\mu \gtrless (1-x^2)^{1/2}(\nu+\tfrac{1}{2}), \tag{3.7}$$

it being assumed again that μ and $\nu+\frac{1}{2}$ are nonnegative.

Figure 1 depicts the situation schematically in the plane of $\nu+\frac{1}{2}$ and μ. In the shaded region below the line

$$\mu = (1-x^2)^{1/2}(\nu+\tfrac{1}{2}), \tag{3.8}$$

the discriminant is negative, consequently the solutions of (3.1)–(3.4) are oscillatory in nature and may be generated in a stable manner in any direction. On the other hand in the region above the line (3.8) the solutions are monotonic and the recurrence direction is crucial in order to maintain stability. By considering the asymptotic forms of the Ferrers functions for large μ and large ν [2, Chap. 3; 17, Chaps. 5 and 12; 21], it may be verified that with conditions (2.3):

(a) *The stable directions for recurring* $P_\nu^{-\mu}(x)$ *in the monotonic region are increasing* ν *and decreasing* μ.

(b) *The stable directions for recurring* $Q_\nu^{\mu}(x)$ *in the monotonic region are decreasing* ν *and increasing* μ.

These results are illustrated diagrammatically in Figs. 2–5 below.

4. SERIES EXPANSIONS

From (2.1) we have the expansion

$$P_\nu^{-\mu}(x) = \frac{1}{\mu!}\left(\frac{1-x}{1+x}\right)^{\mu/2} \sum_{j=0}^{\infty} A_j(\mu,\nu)\left(\frac{1}{2}-\frac{1}{2}x\right)^j, \tag{4.1}$$

in which the coefficients $A_j(\mu, \nu)$ are given by

$$A_0(\mu, \nu) = 1; \quad A_{j+1}(\mu, \nu) = \frac{(j-\nu)(j+\nu+1)}{(j+\mu+1)(j+1)} A_j(\mu, \nu), \quad j \geqslant 0. \tag{4.2}$$

The region of validity of this result includes $x \in (-1, 1)$, $\mu \in [0, \infty)$, $\nu \in (-\infty, \infty)$. We use it to compute $P_\nu^{-\mu}(x)$ only when $x \in [0, 1)$, $\mu \in [0, \infty)$, $\nu \in (-\frac{3}{2}, \frac{1}{2}]$, however. In these circumstances it can be shown that if the series is truncated at the term for which $j = J - 1$, with J arbitrary, then an approximate bound for the *relative* error incurred by the truncation is

$$2^{3-J}/(\pi J). \tag{4.3}$$

The corresponding expansion for $Q_\nu^\mu(x)$ involves natural logarithms (ln) and the logarithmic derivative (ψ) of the Gamma function. Since we shall use this expansion only when $\mu = 0$ and 1, we state it explicitly in these cases:

$$Q_\nu^0(x) = \sum_{j=0}^{\infty} \left\{ \frac{1}{2} \ln\left(\frac{1+x}{1-x}\right) + \psi(j+1) - \psi(\nu+1) \right\} A_j(0, \nu) \left(\frac{1}{2} - \frac{1}{2}x\right)^j, \tag{4.4}$$

$$\begin{aligned} Q_\nu^1(x) = &-(1-x^2)^{-1/2} + \left(\frac{1-x}{1+x}\right)^{1/2} \\ &\times \sum_{j=0}^{\infty} \left[-\frac{\nu(\nu+1)}{2} \ln\left(\frac{1+x}{1-x}\right) + \frac{(j-\nu)(j+\nu+1)}{2(j+1)} \right. \\ &\left. + \nu(\nu+1)\{\psi(\nu+1) - \psi(j+1)\} \right] A_j(1, \nu)(\tfrac{1}{2} - \tfrac{1}{2}x)^j. \end{aligned} \tag{4.5}$$

Here $A_j(\mu, \nu)$ is again defined by (4.2).

We use (4.4) and (4.5) when $x \in [0, 1)$ and $\nu \in [-\frac{1}{2}, \frac{3}{2})$. In these circumstances it can be shown that if (4.4) is truncated at the term for which $j = J - 1$, with $J \geqslant 2$, then an approximate bound for the *absolute* error is

$$\{2 \ln(J+1) + 4.50\}/(\pi J 2^J). \tag{4.6}$$

Similarly if (4.5) is truncated when $j = J - 1$, with $J \geqslant 2$, then an approximate bound for the *absolute* error is

$$\{J + 7.50 \ln(J+1) + 19.56\}/(\pi J^2 2^J). \tag{4.7}$$

Values of all ψ-functions needed in (4.4) and (4.5) are calculated from the asymptotic expansion

$$\psi(t) \sim \ln t - \frac{1}{2t} - \sum_{j=1}^{\infty} \frac{B_{2j}}{2jt^{2j}}, \qquad t \to +\infty, \tag{4.8}$$

in which B_{2j} denotes the $(2j)$th Bernoulli number, and the recurrence relation

$$\psi(t) = \psi(t+1) - (1/t). \tag{4.9}$$

For all positive values of t the absolute error incurred by truncating (4.8) is bounded by the absolute value of the first neglected term.[3]

5. Casoratians

In the theory of linear difference equations given, for example, in [14, Chap. XII], the Casorati determinants, or "Casoratians," play a role analogous to that of the Wronski determinants in the theory of linear differential equations.

Casoratians associated with Eqs. (3.1)–(3.4) are given by

$$P^{\mu}_{\nu+1}(x)\, Q^{\mu}_{\nu}(x) - P^{\mu}_{\nu}(x)\, Q^{\mu}_{\nu+1}(x) = \frac{\Gamma(\nu+\mu+1)}{\Gamma(\nu-\mu+2)}, \tag{5.1}$$

$$(\mu+\nu+1)\, P^{-\mu}_{\nu+1}(x)\, Q^{\mu}_{\nu}(x) + (\mu-\nu-1)\, P^{-\mu}_{\nu}(x)\, Q^{\mu}_{\nu+1}(x) = \cos(\mu\pi), \tag{5.2}$$

$$P^{\mu+1}_{\nu}(x)\, Q^{\mu}_{\nu}(x) - P^{\mu}_{\nu}(x)\, Q^{\mu+1}_{\nu}(x) = \frac{\Gamma(\nu+\mu+1)}{\Gamma(\nu-\mu+1)} \frac{1}{(1-x^2)^{1/2}}, \tag{5.3}$$

$$(\mu+\nu+1)(\mu-\nu)\, P^{-(\mu+1)}_{\nu}(x)\, Q^{\mu}_{\nu}(x) - P^{-\mu}_{\nu}(x)\, Q^{\mu+1}_{\nu}(x) = \frac{\cos(\mu\pi)}{(1-x^2)^{1/2}}. \tag{5.4}$$

These identities are valuable as checks. They may be derived by combination of the Wronskians for $P^{\mu}_{\nu}(x)$ and $Q^{\mu}_{\nu}(x)$ with the recurrence relations for these functions.

6. Description of the Package

Both single- and double-precision versions of the code are available. The following description pertains to the double-precision version. The single-precision version is very similar.

Although a total of fifteen subroutines are employed, for most purposes the user need be concerned with two, XDSET and XDLEGF, and perhaps a third, XDCSRT.[4] We begin this section with descriptions of these three subroutines. This is followed by brief descriptions of the other subroutines and an outline of the construction of XDLEGF.

[3] This method for computing the ψ-functions was preferred to methods based on approximations of the type described in [5], because it is of comparable speed in the present context, and more portable.

[4] Names of all subroutines in the double-precision version begin with the letters XD. Names of the corresponding subroutines in the single-precision version begin instead with XS.

(i) **XDSET.** As explained in the introductory section, to avoid frequent failure arising from overflow or underflow all computations are carried out in extended-range arithmetic. At the conclusion, however, the results are converted to ordinary double-precision floating-point form whenever this is possible.

Extended-range arithmetic represents each nonzero real number ξ as an ordered pair (ϖ, k) in which ϖ is an ordinary double-precision floating-point number and k is a signed integer such that

$$\xi = \varpi \times r^k, \tag{6.1}$$

r being the internal arithmetic base. The numbers ϖ and k are called, respectively, the *principal part* and *auxiliary index* of ξ. Full details are supplied in [13, 21]. For present purposes the user need only set the parameters to initialize XDSET and to call XDSET prior to calling XDLEGF or XDCSRT:

CALL XDSET(IRAD, NRADPL, DZERO, NBITS),

where

IRAD	is the radix r, that is, the double-precision arithmetic base of the computer;
NRADPL	is the number of radix places that are carried in double precision;
DZERO	is the largest representable double-precision number, expressed in base 10;
NBITS	is the number of bits, exclusive of the sign bit, in an integer word.

On the UNIVAC 1100 series computers, for example, IRAD = 2, NRADPL = 60, DZERO = 8.9×10^{307}, NBITS = 35.

(ii) **XDLEGF.** This is the driver program for the main part of the package. It has six input scalar parameters $v_1, \Delta v, \mu_1, \mu_2, \theta$ and ID, and two vectors of computed results PQA and IPQA. The call for XDLEGF is

CALL XDLEGF (DNU1, NUDIFF, MU1, MU2, THETA, ID, PQA, IPQA),

where DNU1 = v_1,NUDIFF = Δv,MU1 = μ_1,MU2 = μ_2,THETA = θ. The values of v_1, θ and PQA must be typed double precision, and those of Δv, μ_1, μ_2,ID and IPQA must be typed integer.

Input parameters. (a) v_1 and $v_2 \equiv v_1 + \Delta v$ are the extremes of the chosen values of the degree v. Neither v_1 nor v_2 need be an integer but Δv must be a nonnegative integer; furthermore $v_1 \geqslant -\frac{1}{2}$.

(b) μ_1 and μ_2 are the extremes of the chosen values of the order μ; each of μ_1, μ_2, and $\mu_2 - \mu_1$ must be a nonnegative integer.

(c) At least one of Δv and $\mu_2 - \mu_1$ is zero.

(d) θ is the value, in radians, of arc cos x; θ must satisfy $0 < \theta \leqslant \frac{1}{2}\pi$.

(e) ID determines which solution of the associated Legendre equation (1.1) is to be computed. The possible values of ID are 1, 2, 3, and 4: they correspond to $P_\nu^{-\mu}(x)$, $Q_\nu^\mu(x)$, $P_\nu^\mu(x)$, and $\mathscr{P}_\nu^\mu(x)$, respectively.

(f) All of the conditions included in (a)–(e) are checked before computation commences. An error message is returned in the event of failure.

If $\mu_2 - \mu_1 = 0$, then each of PQA and IPQA has $\Delta\nu + 1$ elements, the jth pair corresponding to $\nu = \nu_1 + j - 1$. Similarly if $\Delta\nu = 0$, then each of PQA and IPQA has $\mu_2 - \mu_1 + 1$ elements, the jth pair corresponding to $\mu = \mu_1 + j - 1$. The arrays PQA and IPQA must be singly dimensioned in the user's program, and of size at least $\mu_2 - \mu_1 + \Delta\nu + 1$. Failure to do so will result in errors that cannot be detected by XDLEGF.

Computed results. Corresponding elements of the vectors PQA and IPQA represent the principal part ϖ and auxiliary index k, respectively, of the extended-range form (6.1) of the wanted solution.

(a) *If an element of* IPQA *is zero* (and this is arranged to be so whenever possible), *then the corresponding element of* PQA *can be used as an ordinary double-precision floating-point representation of the wanted solution.*

(b) If an element of IPQA is nonzero, then the corresponding number cannot be represented in ordinary double-precision form because its exponent would be too large in absolute value. If desired, the user can convert these results to base 10 numbers by calling the subroutine XDCON mentioned in (iv) below. The line

CALL XDCON (PQA(I), IPQA(I))

of the code replaces each principal part ϖ in PQA and the corresponding index k in IPQA by a decimal floating-point number ϖ_1 in PQA and an integer k_1 in IPQA such that

$$\xi \equiv \varpi \times r^k = \varpi_1 \times 10^{k_1}.$$

The results may then be printed as decimal numbers; compare Table V below.

(iii) **XDCSRT.** This subroutine is for checking purposes. After sequences of values of the wanted solutions of (1.1) have been computed by XDLEGF, XDCSRT may be called as an optional feature. If $\mu_2 - \mu_1 = 0$, then XDCSRT computes the ratio of the two sides of Eq. (5.1) and the ratio of the two sides of Eq. (5.2) for $\nu = \nu_1, \nu_1 + 1, \ldots, \nu_2 - 1$. Alternatively, if $\Delta\nu = 0$, then XDCSRT computes the ratio of the two sides of Eq. (5.3) and the ratio of the two sides of Eq. (5.4) for $\mu = \mu_1, \mu_1 + 1, \ldots, \mu_2 - 1$. Since each of these ratios should be exactly 1, the computed ratios indicate the number of accurate significant figures in the calculated solutions.

(iv) *Other subroutines.* Of the remaining twelve subroutines in the package, five are for implementing extended-range arithmetic. They are called **XDADJ, XDADD, XDRED, XDC210,** and **XDCON** and are the same as the subroutines

ADJUST, ADD, REDUCE, CNV210, and CONVRT, respectively, that were described in [20, Sect. 3]. The others are as follows:

XDPQNU. This serves three purposes:

(a) Initializing three constants: the number of terms (called JO) to be used in the series expansions (4.1), (4.4), and (4.5); the smallest value of t (called IPSIX) to be used in summing the asymptotic expansion (4.8); the number of terms (called IPSIK) to be taken in the sum in (4.8).

Typical values are JO = 56, for (4.1), (4.4), and (4.5); IPSIX = 13; IPSIK = 7. These ensure full accuracy when 60 bits are carried in the mantissae of double-precision floating-point numbers.

(b) For arbitrary μ and arbitrary $v_0 \in (-\frac{3}{2}, -\frac{1}{2}]$, XDPQNU uses (4.1) to compute $P_{v_0}^{-\mu}(x)$ and $P_{v_0+1}^{-\mu}(x)$, and then recurs forwards in v by means of (3.1).

(c) For $\mu = 0, 1$ and arbitrary $v_0 \in [-\frac{1}{2}, \frac{1}{2})$ XDPQNU uses (4.4) and (4.5) to compute $Q_{v_0}^{\mu}(x)$ and $Q_{v_0+1}^{\mu}(x)$, and then recurs forwards in v by means of (3.2). The values of $\psi(j+1)$ and $\psi(v+1)$ needed in (4.4) and (4.5) are generated from (4.8) and (4.9) by means of a subroutine called **XDPSI**.

XDPMU. For arbitrary μ and v, XDPMU calls XDPQNU (a) and (b) to compute $P_v^{-\mu}(x)$ and $P_v^{-(\mu-1)}(x)$, and then recurs backwards in μ by means of (3.3).

XDQMU. For arbitrary v, XDQMU calls XDPQNU (a) and (c) to compute $Q_v^0(x)$ and $Q_v^1(x)$, and then recurs forwards in μ by means of (3.4).

XDQNU. For arbitrary v, XDQNU calls XDPQNU (a) and (c) to compute $Q_{v-1}^0(x)$, $Q_v^0(x)$, $Q_{v-1}^1(x)$, and $Q_v^1(x)$, and then recurs forwards in μ by means of (3.4) to give $Q_{v-1}^{\mu}(x)$ and $Q_v^{\mu}(x)$ for an arbitrary value of μ. Lastly, for this value of μ XDQNU recurs backwards in v by means of (3.2).

XDPMUP. For arbitrary μ and v, XDPMUP computes $P_v^{\mu}(x)$ from $P_v^{-\mu}(x)$ by use of (2.4).

XDPNRM. For arbitrary μ and v, v now being a nonnegative integer, XDPNRM computes $\bar{\mathscr{P}}_v^{\mu}(x)$ from $P_v^{-\mu}(x)$ by means of (2.6).

(v) *Construction of* XDLEGF.

(a) ID = 1, $\mu_1 = \mu_2$. XDLEGF calls XDPQNU (a) and (b) to compute $P_v^{-\mu_1}(\cos\theta)$ for $v = v_0, v_0+1, \ldots, v_2$ $(\equiv v_1 + \Delta v)$. Here $v_0 \in (-\frac{3}{2}, -\frac{1}{2}]$ and $v_1 - v_0$ is a positive integer. The process is depicted schematically in Fig. 2. In this diagram, and the others in this section, crosses ($\times$) indicate points at which the series expansions of Section 4 are used, and arrows ($\rightarrow$) indicate directions in which the recursions of Section 3 are applied.

(b) ID = 1, $v_1 = v_2$, $\mu_1 \neq \mu_2$. XDLEGF calls XDPMU to compute $P_{v_1}^{-\mu}(\cos\theta)$ for $\mu = \mu_2, \mu_2 - 1, \ldots, \mu_1$; see Fig. 3.

(c) ID = 2, $\mu_1 = \mu_2$. XDLEGF calls XDQNU to compute $Q_v^{\mu_1}(\cos\theta)$ for $v = v_2, v_2 - 1, \ldots, v_1$; see Fig. 4.

(d) ID = 2, $v_1 = v_2$, $\mu_1 \neq \mu_2$. XDLEGF calls XDQMU to compute $Q_{v_1}^{\mu}(\cos\theta)$ for $\mu = 0, 1, \ldots, \mu_2$; see Fig. 5.

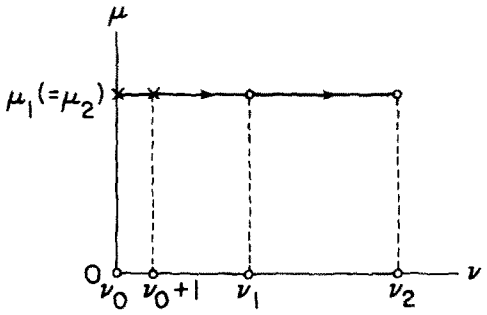

FIG. 2. $P_\nu^{-\mu}(\cos\theta)$; ID = 1, $\mu_1 = \mu_2$.

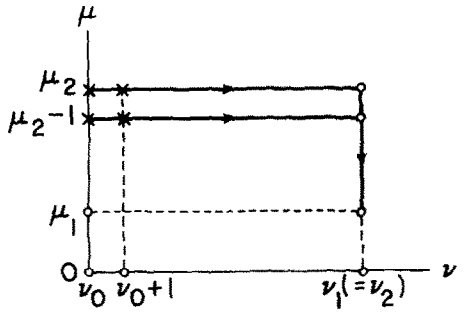

FIG. 3. $P_\nu^{-\mu}(\cos\theta)$; ID = 1, $\nu_1 = \nu_2$.

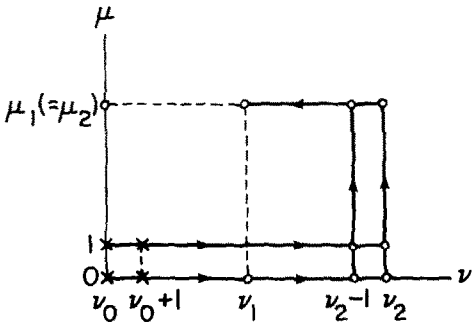

FIG. 4. $Q_\nu^\mu(\cos\theta)$; ID = 2, $\mu_1 = \mu_2$.

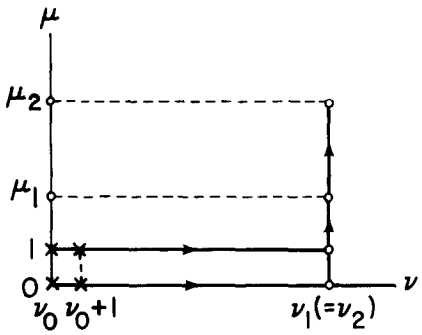

FIG. 5. $Q_\nu^\mu(\cos\theta)$; ID = 2, $\nu_1 = \nu_2$.

(e) ID = 3. XDLEGF computes $P_\nu^{-\mu}(\cos\theta)$ as in (a) or (b), and then calls XDPMUP to compute the corresponding values of $P_\nu^\mu(\cos\theta)$.

(f) ID = 4. XDLEGF computes $P_\nu^{-\mu}(\cos\theta)$ as in (a) or (b), and then calls XDPNRM to compute the corresponding values of $\bar{\mathcal{P}}_\nu^\mu(\cos\theta)$.

7. TESTS

The package has been used to evaluate $P_\nu^{\pm\mu}(\cos\theta)$, $Q_\nu^\mu(\cos\theta)$, and $\bar{\mathcal{P}}_\nu^\mu(\cos\theta)$ for a variety of values of the parameters in the following ranges:

$$0°.1 \leqslant \theta \leqslant 90°, \qquad \mu = 0(1)100{,}000, \qquad -0.5 \leqslant \nu \leqslant 100{,}000. \tag{7.1}$$

Parts of the tables of the Centre National d'Études des Télécommunications [4] and of Belousov [3] were recomputed. The C.N.E.T. tables are of $P_\nu^\mu(\cos\theta)$ and cover the ranges

$$\theta = 0(1°)180°, \qquad \mu = 0(1)5, \qquad \nu = -0.5(0.1)10.0,$$

to a variable number of decimal places. Belousov's tables are of $\bar{\mathcal{P}}_\nu^\mu(\cos\theta)$ and cover the ranges

$$\theta = 0(2°.5)90°, \qquad \mu = 0(1)36, \qquad \nu = \mu(1)56,$$

with a precision of six decimal places. In all cases agreement was satisfactory.

Few tables of $Q_\nu^\mu(\cos\theta)$ exist. Tables 8.3 and 8.4 of [16] give $Q_\nu^0(x)$ and its derivative for

$$\nu = 0, 1, 2, 3, 9, 10, \qquad x = 0.00(0.01)0.99,$$

to eight decimal places for $Q_\nu^0(x)$ and 8, 7, or 6 decimal places for $dQ_\nu^0(x)/dx$. These tables were recomputed. Agreement was satisfactory, except for discrepancies of a

few units in the last decimal place of some of the entries for $Q_\nu^0(x)$. These discrepancies are reported elsewhere [19].

The checking by means of tables covers only a small part of the parameter space (7.1); moreover, the precision of the entries in the tables is much less than is available from the package. More comprehensive tests were as follows.

First, from Figs. 2–5 it is evident that most values of $P_\nu^{-\mu}(\cos\theta)$ and $Q_\nu^\mu(\cos\theta)$ can be generated by XDLEGF by two largely independent recurrence schemes. Independently computed values were compared systematically.

Second, values of $\bar{\mathcal{P}}_\nu^\mu(\cos\theta)$ computed by XDLEGF were compared with corresponding values computed by the package NORMP described in [13, 20]. NORMP uses a different algorithm.

Third, Casoratian checks were applied systematically by calling subroutine XDCSRT.

As expected, the tests indicate that for large values of μ or ν, or both, there is an accumulation of rounding errors during the recursions. The growth is in direct proportion to the total number of recurrence steps, and the loss of decimal significant figures from full double-precision accuracy is approximately equal to the logarithm, to base 10, of the number of steps. This agrees with theoretical estimates of error propagation based on asymptotic representations of the associated Legendre functions given in [17, Chaps. 5 and 12].

Sample running times, in seconds, for double-precision computations on a UNIVAC 1100/82 are given in Tables I–IV. The times are virtually independent of θ: the actual entries in the tables are averages for $\theta = 0°.1$, $1°.0$, and $89°.9$. The times to compute $P_\nu^\mu(\cos\theta)$ and $\bar{\mathcal{P}}_\nu^\mu(\cos\theta)$, corresponding to ID = 3 and 4, respectively, are the same as, or slightly in excess of those, for $P_\nu^{-\mu}(\cos\theta)$. In no case was the increase more than 55%.

A sample output is supplied in Table V. The first row in the block headed $P(-\text{MU}, \text{NU})$, for example, means

$$P_{10000}^{-9990}(\cos 89°.9) = -0.370832029707166839 \times 10^{-38645}.$$

TABLE I

Running Times (sec): $P_\nu^{-\mu}(\cos\theta)$; ID = 1, fixed μ.

ν \ μ	10	100	1000	10000	100000
0(1)10	0.02	0.02	0.07		
0(1)110	0.03	0.04	0.07		
0(1)1010			0.17		
0(1)10010				1.53	
0(1)100000					15.05

TABLE II

Running Times (sec): $P_\nu^{-\mu}(\cos\theta)$; ID = 1, fixed ν.

ν \ μ	0(1)10	90(1)100	990(1)1000	9990(1)10000
0	0.03	0.04	0.12	0.89
10	0.04	0.04	0.12	
100	0.05	0.06	0.14	
1000			0.33	
10000				3.04

TABLE III

Running Times (sec): $Q_\nu^\mu(\cos\theta)$; ID = 2, fixed μ.

ν \ μ	10	100	1000	10000	100000
0(1)10	0.07	0.09	0.27		
100(1)110	0.09	0.11	0.29		
1000(1)1010			0.47		
10000(1)10010				4.19	
99990(1)100000					41.55

TABLE IV

Running Times (sec): $Q_\nu^\mu(\cos\theta)$; ID = 2, fixed ν.

ν \ μ	0(1)10	0(1)100	0(1)1000	0(1)10000
0	0.06	0.07	0.17	1.11
10	0.06	0.07	0.17	
100	0.08	0.09	0.19	
1000			0.37	
10000				3.13

TABLE V

Sample Output

THETA	NU	MU	P(-MU,NU)		P(MU,NU)		Q(MU,NU)	
89.90	10000.0	9990	-.37083202970716839+000	-38645	-.18195899195585144 5+000	38643	-.29422062102588816 1+000	38643
89.90	10000.0	9991	.84159097477171292 5+000	-38648	-.82552812257905780 9+000	38645	.13617958973793972 1+000	38646
89.90	10000.0	9992	.22241268011594147 3+000	-38650	.39254468293530779 2+000	38648	.54068353304997666 6+000	38648
89.90	10000.0	9993	-.47767739584852384 8+000	-38653	.13484417769587382 2+000	38651	-.26388354836577092 7+000	38651
89.90	10000.0	9994	-.17081918276483313 6+000	-38655	-.67488826434506909 0+000	38653	-.77274252033441628 7+000	38653
89.90	10000.0	9995	.34849209663217920 9+000	-38658	-.16518128054325086 4+000	38656	.39628378221232238 4+000	38656
89.90	10000.0	9996	.18301438465982963 9+000	-38660	.86729382384464317 8+000	38658	.78879909036736338 6+000	38658
89.90	10000.0	9997	-.35584526879398290 1+000	-38663	.13488605755381603 9+000	38661	-.42372782221828487 4+000	38661
89.90	10000.0	9998	-.32575259859625108 0+000	-38665	-.74080102279832895 0+000	38663	-.48307992635197379 4+000	38663
89.90	10000.0	9999	.60542723968851590 9+000	-38668	-.55069853044047322 4+000	38665	.27107058477660618 5+000	38666
89.90	10000.0	10000	.17344195207035480 9+000	-38669	.31552669539947412 6+000	38668	.98610190829167117 9+000	38667

THETA	NU	MU	NORM P		CASORATI 4		CASORATI 3	
89.90	10000.0	9990	-.25976835088511671 1+001	0	.99999999999971956+000	0	.99999999999946968+000	0
89.90	10000.0	9991	.26358902769911190 7+001	0	.99999999999971956+000	0	.99999999999947057+000	0
89.90	10000.0	9992	.29546482252503771 7+001	0	.99999999999971956+000	0	.99999999999947104+000	0
89.90	10000.0	9993	-.25380156364740382 3+001	0	.99999999999971955+000	0	.99999999999947104+000	0
89.90	10000.0	9994	-.33954326081677513 7+001	0	.99999999999971955+000	0	.99999999999946996+000	0
89.90	10000.0	9995	.23993175905074875 6+001	0	.99999999999971955+000	0	.99999999999947071+000	0
89.90	10000.0	9996	.39841584036338552 2+001	0	.99999999999971955+000	0	.99999999999947008+000	0
89.90	10000.0	9997	-.21909122613800481 2+001	0	.99999999999971955+000	0	.99999999999947008+000	0
89.90	10000.0	9998	-.49125342147635394 2+001	0	.99999999999971955+000	0	.99999999999946964+000	0
89.90	10000.0	9999	.18259916301005774 2+001	0	.99999999999971955+000	0	.99999999999946964+000	0
89.90	10000.0	10000	.73978579506970372 9+001	0				

Similarly the first row in the block headed NORM P means that

$$\bar{\mathcal{P}}^{9990}_{10000}(\cos 89°.9) = -0.259768350885116711 \times 10^{1}.$$

Since all the auxiliary indices vanish in the case of NORM P, the other entries in this block may be used as ordinary double-precision floating-point numbers. The blocks of columns headed CASORATI 3 and CASORATI 4 give the ratios of the left- and right-hand sides of Eqs. (5.3) and (5.4), respectively, Since these ratios should be exactly unity, the entries indicate that approximately five decimal figures have been lost during the recursions. Thus the computed values of $P_\nu^{-\mu}(\cos\theta)$, $P_\nu^{\mu}(\cos\theta)$, $Q_\nu^{\mu}(\cos\theta)$, and $\bar{\mathcal{P}}_\nu^{\mu}(\cos\theta)$ are correct to about 13 decimal figures. This build-up of error agrees with that described earlier in this section.

Finally, all programs described in this paper are believed to be portable because they have been written in the American National Standard Programming Language FORTRAN (FORTRAN 77) as specified in [1].

ACKNOWLEDGMENTS

The authors are pleased to acknowledge several helpful suggestions by D. W. Lozier and the referees. The research of F. W. J. Olver was supported in part by the U. S. Army Research Office, Durham, under Contract DAAG 29-80-C-0032 and by the U.S. National Science Foundation under Grant MCS 81-11725.

REFERENCES

1. AMERICAN NATIONAL STANDARDS INSTITUTE, "American National Standard Programming Language FORTRAN," Report No. X3.9-1978, New York, 1978.
2. BATEMAN MANUSCRIPT PROJECT, "Higher Transcendental Functions" (A. Erdélyi, Ed.), Vol. I, McGraw-Hill, New York, 1953.
3. S. L. BELOUSOV, "Tables of Normalized Associated Legendre Polynomials," (translated by D. E. Brown), *in* Mathematical Tables Series, Vol. 18, Pergamon, New York, 1962.
4. CENTRE NATIONAL D'ÉTUDES DES TÉLÉCOMMUNICATIONS, "Tables des fonctions de Legendre associées. Fonction associée de première espèce $P_n^m(\cos\theta)$," premier, deuxième, et troisième fascicules, Éditions de la Revue d'Optique, Paris, 1952; 1959; 1966.
5. W. J. CODY, A. J. STRECOK, AND H. C. THACHER, JR., Chebyshev approximations for the Psi function, *Math. Comput.* **27** (1973), 123.
6. J. N. L. CONNOR AND D. C. MACKAY, Calculation of angular distributions in complex angular momentum theories of elastic scattering, *Mol. Phys.* **37** (1979), 1703.
7. J. V. DAVE AND B. H. ARMSTRONG, Computations of high-order associated Legendre polynomials, *J. Quant. Spectrosc. Radiat. Transfer* **10** (1970), 557.
8. A. R. EDMONDS, "Angular Momentum in Quantum Mechanics," 2nd ed., Princeton Univ. Press, Princeton, N. J., 1960.
9. H. E. FETTIS, A new method for computing toroidal harmonics, *Math. Comput.* **24** (1970), 667.
10. W. GAUTSCHI, Algorithm 259. Legendre functions for arguments larger than one, *Commun. ACM* **8** (1965), 488.

11. J. R. HERNDON, Algorithm 47. Associated Legendre functions of the first kind for real or imaginary arguments, *Commun ACM* **4** (1961), 178; Certification of Algorithm 47, *ibid.* **6** (1963), 446.
12. J. R. HERNDON, Algorithm 62. A set of associate Legendre polynomials of the second kind. *Commun ACM* **4** (1961), 320; Remark on Algorithm 62, *ibid.* **4** (1961), 544.
13. D. W. LOZIER AND J. M. SMITH, Algorithm 567. Extended-range arithmetic and normalized Legendre polynomials, *ACM Trans. Math. Software* **7** (1981), 141.
14. L. M. MILNE–THOMSON, "The Calculus of Finite Differences," 2nd ed., Chelsea, New York, 1980.
15. P. M. MORSE AND H. FESHBACH, "Methods of Theoretical Physics," Parts I and II, McGraw-Hill, New York, 1953.
16. NATIONAL BUREAU OF STANDARDS, "Handbook of Mathematical Functions," Appl. Math. Ser. No. 55 (M. Abramowitz and I. A. Stegun, Eds.), U.S. Govt. Printing Office, Washington, D.C., 1964.
17. F. W. J. OLVER, "Asymptotics and Special Functions," Academic Press, New York, 1974.
18. M. E. ROSE, "Elementary Theory of Angular Momentum," Wiley, New York, 1957.
19. J. M. SMITH, "Table errata, 592 - National Bureau of Standards, Handbook of Mathematical Functions," *Math. Comput.* **40** (1983), 723.
20. J. M. SMITH, F. W. J. OLVER, AND D. W. LOZIER, Extended-range arithmetic and normalized Legendre polynomials, *ACM Trans. Math. Software* 7 (1981), 93.
21. R. C. THORNE, The asymptotic expansion of Legendre functions of large degree and order, *Philos. Trans. Roy. Soc. London Ser. A* **249** (1957), 597.
22. R. A. WIGGINS AND M. SAITO, Evaluation of computational algorithms for the associated Legendre polynomials by interval analysis, *Bull. Seismol. Soc. Amer.* **61** (1971), 375.
23. F. W. J. OLVER, Legendre functions with both parameters large, *Philos. Trans. Roy. Soc. London Ser. A* **278** (1975), 175.