FR-7161

A Fortran Computer Program for Calculating the Prolate and Oblate Angle Functions of the First Kind and Their First and Second Derivatives

B.J. King
L. Van Buren

November 30, 1970

Naval Research Laboratory

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Research Laboratory Washington, D. C. 20390	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

A FORTRAN COMPUTER PROGRAM FOR CALCULATING THE PROLATE AND OBLATE ANGLE FUNCTIONS OF THE FIRST KIND AND THEIR FIRST AND SECOND DERIVATIVES

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

An interim report on a continuing NRL problem

5. AUTHOR(S) *(First name, middle initial, last name)*

B. J. King and A. L. Van Buren

6. REPORT DATE November 30, 1970	7a. TOTAL NO. OF PAGES 24	7b. NO. OF REFS 11
8a. CONTRACT OR GRANT NO. NRL Problem S01-28 b. PROJECT NO. RR 102-08-41-5225 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) NRL Report 7161 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*	

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Department of the Navy (Office of Naval Research), Arlington, Va. 22217

13. ABSTRACT

The Helmholtz or scalar wave equation $(\nabla^2 + k^2)\psi = 0$ is separable in both prolate and oblate spheriodal coordinates η, ξ, ϕ with solutions $\psi = S(h, \eta)\, R(h, \xi)\, \Phi(\phi)$ and $\psi = S(ih, \eta)\, R(ih, -i\xi)\, \Phi(\phi)$, respectively. Here $h = kd/2$, where d is the interfocal distance. A Fortran computer program called ANGLFN has been devised to evaluate numerically the angle solutions $S(h, \eta)$ and $S(ih, \eta)$. The printed output from ANFLFN consists of angle functions of the first kind, $S^{(1)}(h, \eta)$ and $S^{(1)}(ih, \eta)$, and their first and second derivatives with respect to η. This report first describes the input data cards and the output format. The theory of the spheroidal wave functions is then discussed. A description of the principal internal features of ANGLFN is then given. Finally, a computer listing of ANGLFN is attached as an appendix.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

Security Classification

Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Fortran computer program						
Prolate angle functions						
Oblate angle functions						
Helmholtz wave equation						
Wave equations						
Underwater acoustics						
Acoustic fields						
Sonar						

DD FORM 1 NOV 65 1473 (BACK)
(PAGE 2)

Security Classification

CONTENTS

Abstract	ii
Problem Status	ii
Authorization	ii
INTRODUCTION	1
INPUT	1
OUTPUT	2
PARAMETER RANGES AND ACCURACY	2
COMPUTATION TIME	2
SOLUTION OF THE HELMHOLTZ EQUATION IN SPHEROIDAL COORDINATES	3
ANGLE FUNCTIONS OF THE FIRST KIND	6
DESCRIPTION OF THE COMPUTER PROGRAM ANGLFN	9
Special Functions	9
Eigenvalues, Expansion Constants, and Conversion Factors	10
Angle Functions	11
REFERENCES	11
Appendix A - Sample Output from ANGLFN	12
Appendix B - Listing in Fortran IV of ANGLFN, a Program to Calculate Prolate and Oblate Spheroidal Angle Wave Functions	14

ABSTRACT

The Helmholtz or scalar wave equation $(\nabla^2 + k^2)\psi = 0$ is separable in both prolate and oblate spheroidal coordinates η, ξ, ϕ with solutions $\psi = S(h, \eta)R(h, \xi)\Phi(\phi)$ and $\psi = S(ih, \eta)R(ih, -i\xi)\Phi(\phi)$, respectively. Here $h = kd/2$, where d is the interfocal distance. A Fortran computer program called ANGLFN has been devised to evaluate numerically the angle solutions $S(h, \eta)$ and $S(ih, \eta)$. The printed output from ANGLFN consists of angle functions of the first kind, $S^{(1)}(h, \eta)$ and $S^{(1)}(ih, \eta)$, and their first and second derivatives with respect to η. This report first describes the input data cards and the output format. The theory of the spheroidal wave functions is then discussed. A description of the principal internal features of ANGLFN is then given. Finally, a computer listing of ANGLFN is attached as an appendix.

PROGRAM STATUS

This is an interim report on a continuing NRL problem.

AUTHORIZATION

NRL Problem S01-28
Project RR 102-08-41-5225

Manuscript submitted July 15, 1970.

A FORTRAN COMPUTER PROGRAM FOR CALCULATING THE PROLATE AND OBLATE ANGLE FUNCTIONS OF THE FIRST KIND AND THEIR FIRST AND SECOND DERIVATIVES

INTRODUCTION

The Helmholtz or scalar wave equation for steady waves $(\nabla^2 + k^2)\psi = 0$, where $k = 2\pi/\lambda$ and λ = wavelength, is separable in both prolate and oblate spheroidal coordinates (ξ, η, ϕ). The factored solution is written as $\psi = R(h, \xi)S(h, \eta)\,\Phi(\phi)$ in prolate coordinates and as $\psi = R(ih, -i\zeta)\,S(ih, \eta)\,\Phi(\phi)$ in oblate coordinates. Here $h = kd/2$, where d is the interfocal distance of the elliptic cross section of the spheroid.

The angle function of the first kind, $S^{(1)}(h, \eta)$ or $S^{(1)}(ih, \eta)$, is one of the two independent solutions to the ordinary differential equation in the angle coordinate η, where $-1 \leq \eta \leq 1$, arising from the separation of variables. These solutions are characterized by the four entities m, h, ℓ, and η (represented by the Fortran variable names M, H, L, and ARG, respectively). M is the integer separation constant relating to the solution for the rotational angle ϕ. For each choice of ARG, M, and H a set of solutions exists to both the prolate and oblate angle equations, each solution characterized by a separation constant or eigenvalue A. These eigenvalues are ordered in an ascending sequence and labeled with integers L, beginning with L = M for the smallest eigenvalue and continuing with L = M + 1, L = M + 2, etc.

The computer program ANGLFN (ANGLe FuNction) calculates numerical values for the angle functions of the first kind $S^{(1)}_{m\ell}(h, \eta)$ and $S^{(1)}_{m\ell}(ih, \eta)$ and their first and second derivatives with respect to η using expansions in associated Legendre functions.

The user of ANGLFN inputs desired values for the parameters M, H, L, and ARG. Operationally, the program is divided into several parts. In the first part, M and ARG are set, and the associated Legendre functions are calculated. In the second part, H is chosen, and the eigenvalues and the expansion constants are obtained for each choice of L. Finally, the expansion constants and the associated Legendre functions are combined to give the angle functions and their first and second derivatives. Two different normalizations of the angle functions are available. This procedure is followed for both prolate and oblate spheroidal geometry.

INPUT

The input consists of five data cards:

Data Card 1: Format 2I1 - This card contains the integer variables IØ1P and NRM, where IØ1P = 0 for oblate, 1 for prolate, and NRM = 2 for Flammer normalization, 3 for Morse and Feshbach normalization (see section on Angle Functions of the First Kind for an explanation of these normalizations).

Data Card 2: Format 3I5 - This card contains the integer variables M1, IDM, and NM, where M1 = the initial value of M desired, IDM = the increment in M used to generate other values of M, and NM = the number of values of M that are desired.

Data Card 3: Format 2D20.10, I5 - This card contains the variables H1, DH, and NH, where H1 is the initial double precision value of H, DH is the double precision increment in H, and NH is the integer number of values of H that are desired.

Data Card 4: Format 3I5 - This card contains the integer variables L1, IDL, and NL, where L1 = the initial value of L desired, IDL = the increment in L, and NL = the number of values of L that are desired.

Data Card 5: Format 2D20.10, 2I5 - This card contains the variables ARG1, DARG, NARG, and KTP, where ARG1 is the double precision initial value of ARG, DARG is the double precision increment in ARG, NARG is the integer number of values of ARG that are desired (NARG $\leq$ 20), and KTP is an integer equal to 0 if the input for ARG is in terms of η ($-1 \leq$ ARG ≤ 1) and equal to 1 if the input for ARG is in terms of $\cos^{-1} \eta$ in degrees ($0° \leq$ ARG $\leq 180°$).

OUTPUT

The output from ANGLFN consists of numerical tables, as shown in Appendix A. Numerical values for the angle function of the first kind, S, and its first and second derivatives with respect to η, SD and SDD, respectively, are provided for particular choices of M, H, L, and ARG. Only 11 significant figures are printed in the tables, although 26 figures are used in the computation.

PARAMETER RANGES AND ACCURACY

Within the following ranges, the angle functions and their derivatives are believed to be accurate to at least 8 significant figures, assuming calculations are performed by a computer that has a word length of 26 digits, e.g., the Control Data Corporation 3800 computer at NRL using double precision arithmetic:

$$0 \leq M \leq 6$$

$$M \leq L \leq M + 49$$

PROLATE

$$\begin{cases} 0.1 \leq H \leq 20.0 \\ -1.0 \leq ARG \leq 1.0 \end{cases} \qquad \begin{cases} 20.0 < H \leq 35.0 \\ 0 \leq |ARG| \leq 0.99,\ |ARG| = 1.0 \end{cases} \qquad \begin{cases} 35.0 < H \leq 40.0 \\ 0 \leq |ARG| \leq 0.9 \end{cases}$$

OBLATE

$$\begin{cases} 0.1 \leq H \leq 30.0 \\ -1.0 \leq ARG \leq 1.0 \end{cases} \qquad \begin{cases} 30.0 < H \leq 40.0 \\ 0.2 \leq |ARG| \leq 1.0 \end{cases}$$

COMPUTATION TIME

The following example is representative of the computation time for ANGLFN on the CDC 3800 computer at NRL:

PROLATE

$$\left.\begin{array}{l} M = 0(1)3 \\ H = .5(.5)5.0 \\ L = M(1)M + 3 \\ ARG = 0°(10°)180° \end{array}\right\} \text{Time} \approx 4.5 \text{ minutes}$$

The average computation time for a line of output decreases as more parameter combinations are requested. This is to be expected, since the Legendre functions and the eigenvalues are not functions of all four parameters and need not be calculated for every combination of M, H, L, and ARG. Neither the choice of prolate or oblate, nor the choice of input for ARG (degrees or pure numbers), nor the choice of the normalization (Flammer or Morse and Feshbach) significantly affect the computation time.

SOLUTION OF THE HELMHOLTZ EQUATION IN SPHEROIDAL COORDINATES

The prolate and oblate spheroidal coordinate systems can be formed by rotating the two-dimensional elliptic coordinate system, consisting of confocal ellipses and hyperbolas, about the major and minor axes of the ellipses, respectively. It is customary to make the z axis the axis of revolution in each case.

The prolate spheroidal coordinates ξ, η, and ϕ where $-1 \le \eta \le -1$, $1 \le \xi < \infty, 0 \le \phi < 2\pi$ are related to the rectangular coordinates x, y, z by the transformation

$$\begin{aligned} x &= \frac{d}{2}\sqrt{(1-\eta^2)(\xi^2-1)}\cos\phi, \\ y &= \frac{d}{2}\sqrt{(1-\eta^2)(\xi^2-1)}\sin\phi, \\ z &= \frac{d}{2}\eta\xi, \end{aligned} \tag{1}$$

where d denotes the interfocal distance of the generating ellipse. The surfaces of constant ξ, η, and ϕ are illustrated in Fig. 1.

The oblate spheroidal coordinates ξ, η, and ϕ where $-1 \le \eta \le 1$, $0 \le \xi < \infty$, $0 \le \phi < 2\pi$ are related to the rectangular coordinates x, y, z by the transformation

$$\begin{aligned} x &= \frac{d}{2}\sqrt{(1-\eta^2)(\xi^2+1)}\cos\phi, \\ y &= \frac{d}{2}\sqrt{(1-\eta^2)(\xi^2+1)}\sin\phi, \\ z &= \frac{d}{2}\eta\xi, \end{aligned} \tag{2}$$

The surfaces of constant ξ, η, and ϕ are illustrated in Fig. 2.

The prolate and oblate spheroidal coordinate systems are two of the 11 orthogonal systems of second degree in which the Helmholtz scalar wave equation,

$$(\nabla^2 + k^2)\psi = 0, \tag{3}$$

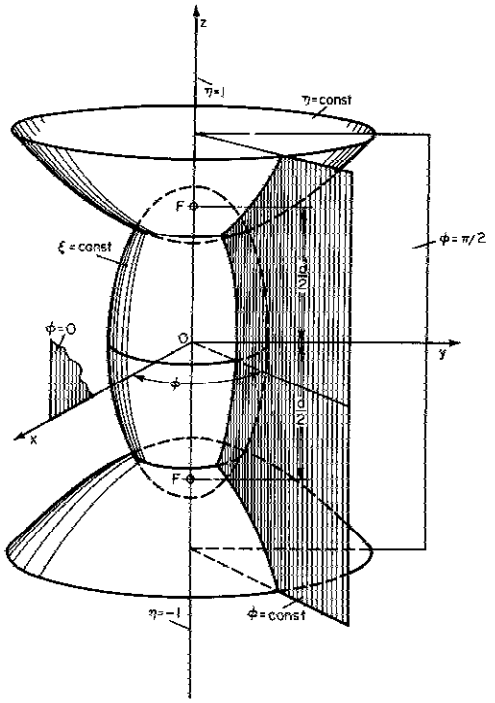

Fig. 1 - The three orthogonal surfaces which determine a point in prolate spheroidal geometry

is separable. The Helmholtz equation, when expressed in prolate spheroidal coordinates is

$$\left[\frac{\partial}{\partial\eta}(1-\eta^2)\frac{\partial}{\partial\eta}+\frac{\partial}{\partial\xi}(\xi^2-1)\frac{\partial}{\partial\xi}+\frac{\xi^2-\eta^2}{(\xi^2-1)(1-\eta^2)}\frac{\partial^2}{\partial\phi^2}+h^2(\xi^2-\eta^2)\right]\psi=0, \tag{4}$$

and when expressed in oblate spheroidal coordinates is

$$\left[\frac{\partial}{\partial\eta}(1-\eta^2)\frac{\partial}{\partial\eta}+\frac{\partial}{\partial\xi}(\xi^2+1)\frac{\partial}{\partial\xi}+\frac{\xi^2+\eta^2}{(\xi^2+1)(1-\eta^2)}\frac{\partial^2}{\partial\phi^2}+h^2(\xi^2+\eta^2)\right]\psi=0, \tag{5}$$

where $h = kd/2$. Note that Eq. (5) can be obtained from Eq. (4) by the transformation $\xi \to -i\xi$, $h \to ih$.

Separation of variables in Eq. (4) and Eq. (5) yields

$$\psi_{m\ell}(h|\xi,\eta,\phi) = R_{m\ell}(h,\xi)\,S_{m\ell}(h,\eta)\,\Phi_m(\phi) \tag{6}$$

and

$$\psi_{m\ell}(ih|-i\xi,\eta,\phi) = R_{m\ell}(ih,-i\xi)\,S_{m\ell}(ih,\eta)\,\Phi_m(\phi), \tag{7}$$

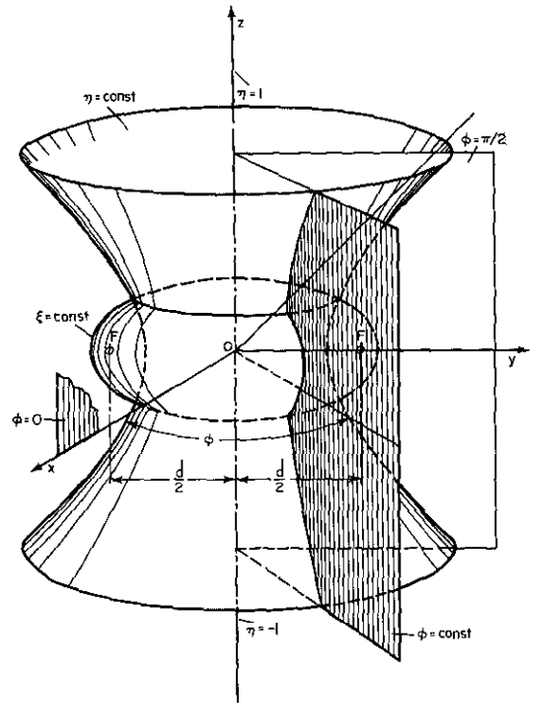

Fig. 2 - The three orthogonal surfaces which determine a point in oblate spheroidal geometry

respectively, where the dependence of the oblate solution on h and ξ is represented as ih and $-i\xi$ in order to show the correspondence between oblate and prolate spheroidal geometry.

The five functions $S_{m\ell}(h, \eta)$, $R_{m\ell}(h, \xi)$, $S_{m\ell}(ih, \eta)$, $R_{m\ell}(ih, -i\xi)$, and $\Phi_m(\phi)$ satisfy the following ordinary differential equations:

$$\frac{d}{d\eta}\left[(1-\eta^2)\frac{d}{d\eta}S_{m\ell}(h, \eta)\right] + \left[A_{m\ell}(h) - h^2\eta^2 - \frac{m^2}{1-\eta^2}\right]S_{m\ell}(h, \eta) = 0, \tag{8}$$

$$\frac{d}{d\xi}\left[(\xi^2-1)\frac{d}{d\xi}R_{m\ell}(h, \xi)\right] - \left[A_{m\ell}(h) - h^2\xi^2 + \frac{m^2}{\xi^2-1}\right]R_{m\ell}(h, \xi) = 0, \tag{9}$$

$$\frac{d}{d\eta}\left[(1-\eta^2)\frac{d}{d\eta}S_{m\ell}(ih, \eta)\right] + \left[A_{m\ell}(ih) + h^2\eta^2 - \frac{m^2}{1-\eta^2}\right]S_{m\ell}(ih, \eta) = 0, \tag{10}$$

$$\frac{d}{d\xi}\left[(\xi^2+1)\frac{d}{d\xi}R_{m\ell}(ih, -i\xi)\right] - \left[A_{m\ell}(ih) - h^2\xi^2 - \frac{m^2}{\xi^2+1}\right]R_{m\ell}(ih, -i\xi) = 0, \tag{11}$$

and

$$\frac{d^2}{d\phi^2}\Phi_m(\phi) + m^2\Phi_m(\phi) = 0, \tag{12}$$

in which m and $A_{m\ell}(h)$ or $A_{m\ell}(ih)$ are the two separation constants or eigenvalues occuring in the separation of variables. In physical problems where it is necessary that the field be periodic and unique over the entire range of the azimuthal coordinate ϕ, it is

required that m be an integer. When $h \to 0$, Eq. (8) and Eq. (10) reduce to the standard differential equation for associated Legendre functions with separation constants given by $A_{m\ell} = \ell(\ell + 1)$, where $\ell = m, m + 1, \ldots$. When $h \neq 0$, the values of the separation constant $A_{m\ell}$ for which nontrivial convergent solutions exists to Eqs. (8) through (11) will be a function of m and h. It is convenient to order these values in an ascending sequence and label them with integers ℓ beginning with $\ell = m$ for the smallest eigenvalue, $\ell = m + 1$ for the next one, etc. The eigenvalues obtained for Eq. (8) and Eq. (9) will not be equal to the eigenvalues for Eq. (10) and Eq. (11).

The two independent solutions of Eq. (8), Eq. (9), Eq. (10), and Eq. (11) define the following eight functions,

$S^{(1)}_{m\ell}(h, \eta)$, prolate angle function of the first kind,

$S^{(2)}_{m\ell}(h, \eta)$, prolate angle function of the second kind,

$R^{(1)}_{m\ell}(h, \xi)$, prolate radial function of the first kind,

$R^{(2)}_{m\ell}(h, \xi)$, prolate radial function of the second kind,

$S^{(1)}_{m\ell}(ih, \eta)$, oblate angle function of the first kind,

$S^{(2)}_{m\ell}(ih, \eta)$, oblate angle function of the second kind,

$R^{(1)}_{m\ell}(ih, -i\xi)$, oblate radial function of the first kind,

$R^{(2)}_{m\ell}(ih, -i\xi)$, oblate radial function of the second kind.

This report concerns the calculation of $S^{(1)}_{m\ell}(h, \eta)$ and $S^{(1)}_{m\ell}(ih, \eta)$. The calculation of prolate and oblate radial functions of both the first and second kinds is discussed in previous reports (1-4).

ANGLE FUNCTIONS OF THE FIRST KIND

The behavior of Eq. (8) and Eq. (10) as $h \to 0$ suggests that the angle functions of the first kind can be expanded in a series of associated Legendre functions,

$$S^{(1)}_{m\ell}(h, \eta) = \sum_{n=0,1}^{\infty}{}' d_n(h|m\ell)\, P^m_{m+n}(\eta), \tag{13}$$

and

$$S^{(1)}_{m\ell}(ih, \eta) = \sum_{n=0,1}^{\infty}{}' d_n(ih|m\ell)\, P^m_{m+n}(\eta), \tag{14}$$

where the d_n's are expansion constants, and the $P^m_{m+n}(\eta)$'s are associated Legendre functions of the first kind. The prime sign on the sum indicates that $n = 0, 2, 4, \ldots$ if $\ell - m$ is even, and $n = 1, 3, 5 \ldots$ if $\ell - m$ is odd.

The derivatives of the angle functions are then given by

$$\frac{d}{d\eta} S^{(1)}_{m\ell}(h, \eta) = \sum_{n=0,1}^{\infty}{}' d_n(h|m\ell) \frac{d}{d\eta} P^m_{m+n}(\eta), \tag{15}$$

$$\frac{d}{d\eta} S^{(1)}_{m\ell}(ih, \eta) = \sum_{n=0,1}^{\infty}{}' d_n(ih|m\ell) \frac{d}{d\eta} P^m_{m+n}(\eta), \tag{16}$$

$$\frac{d^2}{d\eta^2} S^{(1)}_{m\ell}(h, \eta) = \sum_{n=0,1}^{\infty}{}' d_n(h|m\ell) \frac{d^2}{d\eta^2} P^m_{m+n}(\eta), \tag{17}$$

and

$$\frac{d^2}{d\eta^2} S^{(1)}_{m\ell}(ih, \eta) = \sum_{n=0,1}^{\infty}{}' d_n(ih|m\ell) \frac{d^2}{d\eta^2} P^m_{m+n}(\eta). \tag{18}$$

The following discussion will be restricted to the prolate expansion constants $d_n(h|m\ell)$, but it is recognized that the resulting formulas apply equally well to the oblate expansion constants $d_n(ih|m\ell)$ when the transformation $h \to ih$ is made.

Substitution of Eq. (13) into Eq. (8) and use of standard recursion formulas for the associated Legendre Functions lead to the following three-term recursion formula in the unknown expansion constants d_n:

$$\begin{aligned} &\frac{(2m+n+2)(2m+n+1)}{(2m+2n+3)(2m+2n+5)} h^2 d_{n+2} \\ &+ \left[(m+n)(m+n+1) - A_{m\ell}(h) + \frac{2(m+n)(m+n+1) - 2m^2 - 1}{(2m+2n+3)(2m+2n-1)} h^2\right] d_n \\ &+ \frac{n(n-1)}{(2m+2n-3)(2m+2n-1)} h^2 d_{n-2} = 0. \end{aligned} \tag{19}$$

If the choice of $A_{m\ell}$ is arbitrary, solutions given by Eq. (13) with constants d_n determined by Eq. (19) may be divergent either at $\eta = +1$ or $\eta = -1$. However, in physical applications it is necessary that $S^{(1)}_{m\ell}(h, \eta)$ be finite at both of these points. This requires that the separation constants $A_{m\ell}$ be chosen so that $d_{n+2}/d_n \to 0$ as $n \to \infty$. These $A_{m\ell}$ values are the characteristic values (eigenvalues) of Eq. (8). Use of these values leads to successful expansions for $S^{(1)}_{m\ell}(h, \eta)$ of the form given in Eq. (13).

The d_n constants obtained using Eq. (19) will be unnormalized since this equation is homogeneous. The normalization is determined by requiring that the resulting angle functions $S_{m\ell}(h, \eta)$ have some special form at $\eta = 1$ or at $\eta = 0$. The normalization used by Morse and Feshbach (5) requires that

$$\left[(1-\eta^2)^{-m/2} S^{(1)}_{m\ell}(h, \eta)\right]_{\eta=1} = \left[(1-\eta^2)^{-m/2} P^m_\ell(\eta)\right]_{\eta=1}. \tag{20}$$

This requirement leads to the normalization for the d_n constants,

$$\sum_{n=0,1}^{\infty} \frac{(n+2m)!}{n!} \; d_n(h|m\ell) = \frac{(\ell+m)!}{(\ell-m)!} \; . \tag{21}$$

The normalization used by Flammer (6) requires that

$$S_{m\ell}^{(1)}(h, 0) = P_\ell^m(0), \; \ell - m \text{ even},$$

$$\left[\frac{d}{d\eta} S_{m\ell}^{(1)}(h, \eta)\right]_{\eta=0} = \left[\frac{d}{d\eta} P_\ell^m(\eta)\right]_{\eta=0}, \; \ell - m \text{ odd}. \tag{22}$$

The Flammer d_n constants (denoted $d_{n,F}$) are related to the Morse and Feshbach d_n constants (denoted $d_{n,MF}$) by the formulas

$$d_{n,F} = d_{n,MF}\left[\frac{(-1)^{(\ell-m)/2}(\ell+m)!}{2^{\ell-m}\left(\frac{\ell-m}{2}\right)!\left(\frac{\ell+m}{2}\right)!} \Bigg/ \sum_{r=0}^{\infty}{}' \frac{(-1)^{r/2}(r+2m)!}{2^r\left(\frac{r}{2}\right)!\left(\frac{r+2m}{2}\right)!} \; d_{r,MF}\right], \; \ell - m \text{ even}, \tag{23}$$

and

$$d_{n,F} = d_{n,MF}\left[\frac{(-1)^{(\ell-m-1)/2}(\ell+m+1)!}{2^{\ell-m}\left(\frac{\ell-m-1}{2}\right)!\left(\frac{\ell+m+1}{2}\right)!} \Bigg/ \sum_{r=1}^{\infty}{}' \frac{(-1)^{(r-1)/2}(r+2m+1)!}{2^r\left(\frac{r-1}{2}\right)!\left(\frac{r+2m+1}{2}\right)!} \; d_{r,MF}\right], \; \ell - m \text{ odd}. \tag{24}$$

Before obtaining the d_n constants, it is necessary to obtain the characteristic values $A_{m\ell}$. An accurate first approximation for $A_{m\ell}$ is obtained by a matrix method. Here an equation similar to Eq. (19) is derived by using normalized associated Legendre functions. Written in matrix form the equation reduces to

$$[B]\{v\} = A\{v\}, \tag{25}$$

where $\{v\}$ is one of the characteristic vectors, A is one of the $A_{m\ell}$ characteristic numbers, and $[B]$ is an $n \times n$ square symmetric matrix with elements constructed from modified forms of the coefficients in Eq. (19). The problem of determining $A_{m\ell}$ is reduced to the problem of diagonalizing the matrix $[B]$. The eigenvalues then appear as the resulting diagonal elements when ordered numerically from lowest to highest. Although the exact determination of the eigenvalues would require a matrix of order infinity, good first approximations are obtained using matrices of modest proportions. Details of this matrix method are given in Ref. 7.

The eigenvalues obtained above are now used as starting values in a variational procedure devised by Bouwkamp (8) and Blanch (9). The three-term recursion formula Eq. (19) is rewritten in two forms:

$$N_n^m = \frac{-\beta_n^m}{\gamma_n^m - A_{m\ell}(h) + N_{n+2}^m}, \; n \geq 2, \tag{26}$$

and

$$N_{n+2}^m = -\gamma_n^m + A_{m\ell}(h) - \frac{\beta_n^m}{N_n^m}, \; n \geq 2, \tag{27}$$

where

$$\gamma_n^m = (m+n)(m+n+1) + \frac{1}{2}h^2 \left[1 - \frac{4m^2-1}{(2m+2n-1)(2m+2n+3)}\right], \quad n \geq 0, \tag{28}$$

$$\beta_n^m = \frac{n(n-1)(2m+n)(2m+n-1)h^4}{(2m+2n-1)^2(2m+2n-3)(2m+2n+1)}, \quad n \geq 2, \tag{29}$$

$$N_n^m = \frac{(2m+n)(2m+n-1)}{(2m+2n-1)(2m+2n+1)} h^2 \frac{d_n}{d_{n-2}}, \quad n \geq 2. \tag{30}$$

Requiring that $d_n/d_{n-2} \to 0$ as $n \to \infty$ is equivalent to requiring that $N_n^m \to 0$ as $n \to \infty$. If the sequence N_n^m is terminated at a reasonably large value for n, say n_1, Eq. (26) can be written as a continued fraction where any N_n^m, $n < n_1$, is obtained in terms of γ_n^m, γ_{n+2}^m, γ_{n+4}^m, ..., β_n^m, β_{n+2}^m, β_{n+4}^m, ..., and $A_{m\ell}(h)$. Similarly, Eq. (27) can be written in a continued fraction expressing N_n^m in terms of γ_{n-2}, γ_{n-4}, γ_{n-6}, ..., β_{n-2}, β_{n-4}, β_{n-6}, ..., and $A_{m\ell}(h)$, using the fact that $N_2^m = -\gamma_0^m + A_{m\ell}$ and $N_3^m = -\gamma_1^m + A_{m\ell}$ to terminate the sequence.

The starting value for the eigenvalue $A_{m\ell}$ is inserted into the two continued fractions, one with diminishing subscripts and one with increasing subscripts. $N_{\ell-m+2}^m$ is calculated using both expressions. A correction to the eigenvalue is now obtained in terms of the difference between the two values for $N_{\ell-m+2}^m$. The process is repeated until $A_{m\ell}$ is obtained to a desired accuracy.

When the values of $A_{m\ell}$ have been accurately determined, the numbers N_n^m are available to determine the ratios d_n/d_{n-2}. The individual d_n constants are then obtained using the normalization condition given in Eq. (21). The prolate angle functions calculated using these d_n constants in Eq. (13) will satisfy Eq. (20). Angle functions normalized according to Eq. (22) can be obtained by multiplication by the conversion factor given in Eq. (23) or Eq. (24). Similarly, oblate angle functions can be calculated using Eq. (14) and oblate expansion constants $d_n(ih|m\ell)$.

DESCRIPTION OF THE COMPUTER PROGRAM ANGLFN

The Fortran IV computer program ANGLFN used to calculate the prolate and oblate spheroidal angle functions of the first kind and their first and second derivatives is listed in Appendix B. ANGLFN was written in double precision arithmetic for the Control Data Corporation 3800 computer at NRL. Here double precision numbers have 26 decimal digits and an exponent range of -307 to +307. A description of the input and output for ANGLFN has already been given. Some details of the program are given below.

Special Functions

Two special functions are required: the factorial function and the associated Legendre functions of the first kind.

1. The first 170 integer factorials are calculated in lines 90 through 110 and stored in the array FACT(N), where FACT (N + 1) = N!. The first 275 integer factorials, scaled by 10^{-250}, are calculated in lines 120 through 140 and stored in the array FBCT(N), where FBCT (N + 1) = 10^{-250} N! . These scaled factorials are necessary to prevent exponent overflow during the calculation of the associated Legendre functions.

2. The associated Legendre functions of the first kind $P_N^M(\cos\theta)$ and their first and second derivatives with respect to $\cos\theta$ are calculated in subroutine POLY. Subroutine POLY is called in line 580 for fixed M and $\cos\theta$, $0 \le \cos\theta \le 1$, and returns values for $N = 0, 1, 2, \ldots, 120$. $P_N^M(\cos\theta)$ is calculated using

$$P_N^M(\cos\theta) = \frac{\sin^M\theta}{2^{N-M-1}(2M-1)!!}\left\{\sum_{p=0}^{[(N-M-1)/2]} \frac{(2N-2M-1)!!\,(2M+2P-1)!!}{P!\,(N-M-P)!}\cos[(N-M-2P)\theta] + \begin{cases} 0, & N-M \text{ odd} \\ \dfrac{(N+M+1)!!}{2[(N-M)/2]!^2}, & N-M \text{ even}, \end{cases}\right\} \tag{31}$$

where $[(N-M-1)/2]$ means the greatest integer less than or equal to $(N-M-1)/2$, and where $(2N-1)!! = 1, 3, 5, \ldots, (2N-1)$ with $(-1)!!$ being defined equal to unity. Special formulas are used for $\cos\theta = 0, 1$ (Ref. 10). Values for negative argument, $-1 \le \cos\theta < 0$, are obtained using the relation $P_N^M(-\cos\theta) = (-1)^{N+M} P_N^M(\cos\theta)$.

An equal number of first and second derivatives of $P_N^M(\cos\theta)$ are calculated using the recursion formulas,

$$\frac{d}{d\cos\theta} P_N^M(\cos\theta) = \frac{1}{1-\cos^2\theta}\left[(M+N)P_{N-1}^M(\cos\theta) - N\cos\theta\, P_N^M(\cos\theta)\right], \tag{32}$$

$$\frac{d^2}{d\cos\theta^2} P_N^M(\cos\theta) = \frac{1}{1-\cos^2\theta}\left\{2\cos\theta\,\frac{dP_N^M(\cos\theta)}{d\cos\theta} + \left[\frac{M^2}{1-\cos^2\theta} - N(N+1)\right]P_N^M(\cos\theta)\right\}. \tag{33}$$

When $1 \ge \cos\theta > (1-10^{-10})$, $\cos\theta$ is approximated by unity. Limiting forms for the associated Legendre functions and their first and second derivatives are obtained using the definition

$$P_N^M(\cos\theta) = \sin^M\theta\,\frac{d^M}{d\cos\theta^M} P_N(\cos\theta) \tag{34}$$

and the special formula

$$\left[\frac{d^M P_N(\cos\theta)}{d\cos\theta^M}\right]_{\cos\theta=1} = \frac{(N+M)!}{2^M M!\,(N-M)!}. \tag{35}$$

See Ref. 11 for a derivation of Eq. (35).

Eigenvalues, Expansion Constants, and Conversion Factors

Starting values or numbers agreeing to at least two places with the correct values are obtained for the eigenvalues by the diagonalization of a matrix [B], as explained above. The size of the matrix is determined in lines 630 through 640. The matrix elements are calculated in lines 710 through 830. Subroutine EIGEN, which diagonalizes the matrix, is called in line 840. The resulting diagonal elements are used as starting values for the variational procedure, which is programmed in lines 860 through 1400. Convergence is assumed when the relative contribution of the correction is less than 10^{-24}. Using the corrected eigenvalues the d_n constants are evaluated in lines 1410 through 1650 and stored in the array DLIST. These d_n constants are normalized to satisfy

the Morse and Feshbach normalization. If the normalization of Flammer is desired, a Flammer-Morse and Feshbach conversion factor, as programmed in lines 1680 through 1840, is used.

Angle Functions

The angle function of the first kind $S^{(1)}$ and its first and second derivatives are obtained by combining the d_n constants and the associated Legendre functions. Here Eqs. (13), (15), and (17) for the prolate case or Eqs. (14), (16), and (18) for the oblate case, as programmed in lines 1860 through 2470, are used. The series are terminated when the relative contribution of the last term is less than 10^{-14}.

REFERENCES

1. King, B. J., Baier, R. V., and Hanish S., "A Fortran Computer Program for Calculating the Prolate Spheroidal Radial Functions of the First and Second Kind and Their First Derivatives," NRL Report 7012, Mar. 30, 1970

2. Van Buren, A. L., Baier, R. V., and Hanish S., "A Fortran Computer Program for Calculating the Oblate Spheroidal Radial Functions of the First and Second Kind and Their First Derivatives," NRL Report 6959, Jan. 20, 1970

3. Hanish, S., Baier, R. V., Van Buren, A. L., and King, B. J., "Tables of Radial Spheroidal Wave Functions, Vols. 1-3, Prolate, M=0, 1, 2," NRL Reports 7088-7090, June 30, 1970

4. Hanish, S., Baier, R. V., Van Buren, A. L., and King, B. J., "Tables of Radial Spheroidal Wave Functions, Vols. 4-6, Oblate, M=0, 1, 2," NRL Reports 7091-7093, June 30, 1970

5. Morse, P. M., and Feshbach, H., "Methods of Theoretical Physics," Part I, New York: McGraw-Hill, 1953

6. Flammer, C., "Spheroidal Wave Functions," Stanford: Stanford Univ. Press, 1957

7. Hanish, S., and King, B., "Tables of Eigenvalues of the Wave Equation in Prolate Spheroidal Coordinates," NRL Report 6856, Apr. 4, 1969

8. Bouwkamp, C. J., "Theoretische en Numerieke Behandeling van der Buiging door een Ronde Opening," Diss. Groningen, Groningen-Batavia, 1941

9. Blanch, G., "On the Computation of Mathieu Functions," J. Math. Phys. 25:1-20 (1946)

10. Berkowitz, R. L., and Van Buren, A. L., "A Double Precision Fortran Subroutine to Evaluate the Associated Legendre Functions of the First Kind for Real and Imaginary Arguments," NRL Memorandum Report 2086, Aug. 1969

11. Robin, L., "Fonctions Sphériques de Legendre et Fonctions Sphéroïdales, Vol. I," Gauthier-Villars, Paris, p. 77, 1957

Appendix A

SAMPLE OUTPUT FROM ANGLFN

PROLATE ANGLE FUNCTIONS

FLAMMER NORMALIZATION

M	H	L	ARG	S	SD	SDD
0	1,000	0	30,000	8,8469867618-001	-2,5646771481-001	-2,5164012760-001
0	1,000	0	60,000	9,6061108379-001	-1,5562274048-001	-2,9587327765-001
0	1,000	0	90,000	1,0000000000+000	0,0000000000+000	-3,1900005515-001
0	1,000	0	120,000	9,6061108379-001	1,5562274048-001	-2,9587327765-001
0	1,000	0	150,000	8,8469867618-001	2,5646771481-001	-2,5164012760-001
0	1,000	1	30,000	8,0350733589-001	7,8726925148-001	-4,6936655154-001
0	1,000	1	60,000	4,8775317760-001	9,2695348094-001	-2,8785795777-001
0	1,000	1	90,000	0,0000000000+000	1,0000000000+000	0,0000000000+000
0	1,000	1	120,000	-4,8775317760-001	9,2695348094-001	2,8785795777-001
0	1,000	1	150,000	-8,0350733589-001	7,8726925148-001	4,6936655154-001
0	10,000	0	30,000	1,1234514974-002	-1,6700874183-001	1,7985818963+000
0	10,000	0	60,000	2,9233710736-001	-1,5403838192+000	4,0936907744+000
0	10,000	0	90,000	1,0000000000+000	0,0000000000+000	-9,2283042972+000
0	10,000	0	120,000	2,9233710736-001	1,5403838192+000	4,0936907744+000
0	10,000	0	150,000	1,1234514974-002	1,6700874183-001	1,7985818963+000
0	10,000	1	30,000	1,3142041990-002	-1,6406576280-001	1,3270070023+000
0	10,000	1	60,000	1,3705641631-001	-4,6300033148-001	-1,2735078880+000
0	10,000	1	90,000	0,0000000000+000	1,0000000000+000	0,0000000000+000
0	10,000	1	120,000	-1,3705641631-001	-4,6300033148-001	1,2735078880+000
0	10,000	1	150,000	-1,3142041990-002	-1,6406576280-001	-1,3270070023+000
1	1,000	1	30,000	4,6429122705-001	-1,6886391629+000	-6,9551973491+000
1	1,000	1	60,000	8,4504245526-001	-6,4655371249-001	-1,5518685305+000
1	1,000	1	90,000	1,0000000000+000	0,0000000000+000	-1,1955483554+000
1	1,000	1	120,000	8,4504245526-001	6,4655371249-001	-1,5518685305+000
1	1,000	1	150,000	4,6429122705-001	1,6886391629+000	-6,9551973491+000
1	1,000	2	30,000	1,2314879303+000	-2,9967750720+000	-2,9011753864+001
1	1,000	2	60,000	1,2762087065+000	1,6109240931+000	-6,0902254738+000
1	1,000	2	90,000	0,0000000000+000	3,0000000000+000	0,0000000000+000
1	1,000	2	120,000	-1,2762087065+000	1,6109240931+000	6,0902254738+000
1	1,000	2	150,000	-1,2314879303+000	-2,9967750720+000	2,9011753864+001
1	10,000	1	30,000	1,0535522536-002	-1,6172704416-001	1,7751992128+000
1	10,000	1	60,000	2,8963210120-001	-1,5396717101+000	4,1435184936+000
1	10,000	1	90,000	1,0000000000+000	0,0000000000+000	-9,2877687674+000
1	10,000	1	120,000	2,8963210120-001	1,5396717101+000	4,1435184936+000
1	10,000	1	150,000	1,0535522536-002	1,6172704416-001	1,7751992128+000
1	10,000	2	30,000	3,6571752335-002	-4,7764733913-001	3,9555333224+000
1	10,000	2	60,000	4,6614233632-001	-1,3993777549+000	-3,7338773771+000
1	10,000	2	90,000	0,0000000000+000	3,0000000000+000	0,0000000000+000
1	10,000	2	120,000	-4,6614233632-001	-1,3993777549+000	3,7338773771+000
1	10,000	2	150,000	-3,6571752335-002	-4,7764733913-001	-3,9555333224+000

Appendix B

LISTING IN FORTRAN IV, A PROGRAM TO CALCULATE PROLATE AND OBLATE SPHEROIDAL ANGLE WAVE FUNCTIONS

```
      PROGRAM ANGLFN
      TYPE DOUBLE ARG,ARG1,ARR,BLIST,CL,CORA,CORB,DARG,DE,DH,DL,DLIST,
     1DN,DND0,EA,EM,ENR,ENRC,EYE,FACT,GLIST,H,H1,HSQ,P,PD,PDD,S,SD,SDD,
     2TERM,TNS,W,W26,W28,FBCT,C101,DNUM,AAA
      DIMENSION BLIST(150),DLIST(150),DN(150),DND0(150),ENR(150),
     1GLIST(150),P(20,125),PD(20,125),PDD(20,125),A(80,80),EIG(80)
      COMMON /BLK1/FACT(171),FBCT(300)
      TNS = 1.D-14
      FACT(1) = 1.D
      DO 77 J = 1,170
   77 FACT(J+1) = J*FACT(J)
      FBCT(1) = 1.D-250
      DO 2949 J = 1,275
 2949 FBCT(J+1) = J*FBCT(J)
      READ 5, IOOIP,NRM
    5 FORMAT (2I1)
C
C        IOOIP IS 0 FOR OBLATE AND 1 FOR PROLATE
C        NRM IS 2 FOR FLAMMER NORMALIZATION AND 3 FOR MORSE AND FESHBACH
C        NORMALIZATION
C
      IF (IOOIP.EQ.0) GO TO 55
      PRINT 56
   56 FORMAT (56X23HPROLATE ANGLE FUNCTIONS//)
      GO TO 58
   55 PRINT 57
   57 FORMAT (56X22HOBLATE ANGLE FUNCTIONS//)
   58 IF (NRM.EQ.2) GO TO 61
      PRINT 62
   62 FORMAT (51X*MORSE AND FESHBACH NORMALIZATION*//)
      GO TO 64
   61 PRINT 63
   63 FORMAT (57X*FLAMMER NORMALIZATION*//)
   64 READ 90, M1,IDM,NM
   90 FORMAT (3I5)
      READ 91,H1,DH,NH
   91 FORMAT (2D20.10,I5)
      READ 92,L1,IDL,NL
   92 FORMAT (3I5)
      READ 93,ARG1,DARG,NARG,KTP
   93 FORMAT (2D20.10,2I5)
C
C        KTP IS 0 FOR INPUT OF ARG IN TERMS OF ETA AND 1 FOR INPUT IN
C        TERMS OF DEGREES
C
      PRINT 100
  100 FORMAT (10X1HM5X1HH5X1HL5X3HARG19X1HS32X2HSD30X3HSDD//)
      DO 2 IM = 1,NM
      M = M1 + (IM-1)*IDM
      IF (L1.LT.M) L1 = M
      NBL = L1 + (NL-1)*IDL+1-M
      EM = M
      DO 1 INA = 1,NARG
```

```
      ARG = ARG1 + (INA-1)*DARG
      IF (ARG.LT.0.D) ARG = -ARG
      IF (ARG.GT.90.D) ARG = 180.D0-ARG
      IF (KTP.EQ.1) ARG = DCOS(.01745329251994329576923690D*ARG)
    1 CALL POLY(INA,M,120,ARG,P,PD,PDD)
      DO 2 INH = 1,NH
      H = H1 + (INH-1)*DH
      LCK = 0
      IH = H
      N = MAXOF(50,50+(IH-5)/2)
      IF (N.GT.80) N = 80
      HSQ = H*H
      IF (IOOIP.EQ.0) HSQ = -HSQ
      DO 2 IL = 1,NL
      L = L1 + (IL-1)*DL
      IF (LCK.EQ.1) GO TO 4
      LCK = 1
      DO 31 J = 1,N
      DO 31 I = 1,N
   31 A(I,J) = 0.
      DO 32 I = 1,N
      W = I-1.+EM
   32 A(I,I) = W*(W+1.)+HSQ*(2.*W*(W+1.)-2*EM*EM-1.)/((2.*W-1.)*(2.*W+3.
     1))
      NM2 = N-2
      DO 33 I = 1,NM2
      W = I-1.+EM
      A(I,I+2) = (HSQ/(2.*W+3.))*DSQRT(((W+2.+EM)*(W+1.+EM)*(W+2.-EM)*(W
     1+1.-EM))/((2.*W+5.)*(2.*W+1.)))
   33 A(I+2,I) = A(I,I+2)
      CALL EIGEN (A,EIG,N,NBL)
    4 CL = EIG(L+1-M)
      IW6 = (L-M)/2
      IX = L-M - 2*IW6
      ISC = 2 + IX
      LIM1 = 278
      J=1
      DO 13 I=ISC,LIM1,2
      EYE =I
      EA = 2.D*EM+2.D*EYE
      BLIST(J) = (EYE*(EYE-1.D)*(EA-EYE)*(EA-EYE-1.D)*HSQ*HSQ)/((EA-1.D)
     1*(EA-1.D)*(EA-3.D)*(EA+1.D))
   13 J=J+1
      J=1
      ID21 = ISC-1
      LIM11=LIM1+1
      DO 14 I=ID21,LIM11,2
      EYE=I-1.D
      EA = EM + EYE
      GLIST(J) = EA*(EA+1.D)+0.5D*HSQ*((1.D)-(4.D*EM*EM-1.D)/((2.D*EA-1.
     1D)*(2.D*EA+3.D)))
   14 J=J+1
      IFC=0
      IBLIM = 139 - IX
      IGLIM=IBLIM+1
      IRIO=IW6+1
      IW1 = IW6+2
   17 ENR(1)=CL-GLIST(1)
      DO 18 I=1,IW6
```

```
 18 ENR(I+1)=-BLIST(I)/ENR(I)-GLIST(I+1)+CL
    ENR(IBLIM)=-BLIST(IBLIM)/(GLIST(IGLIM)-CL)
    IW15=IBLIM-1
    IP=IW1  +IW15
    DO 19 I=IW1  ,IW15
    IPI=IP-I
 19 ENR(IPI)=-BLIST(IPI)/(GLIST(IPI+1)-CL+ENR(IPI+1))
    ENRC=-BLIST(IRIO)/(GLIST(IRIO+1)-CL+ENR(IRIO+1))
    DE = ENRC*ENRC/BLIST(IRIO)
    CORB = DE
    DO 20 I = IW1,IBLIM
    DE = ENR(I)*ENR(I)/BLIST(I)*DE
    CORB = CORB + DE
    IF (DABS(DE/CORB).LT.1.D-27) GO TO 23
 20 CONTINUE
 23 CORA = 1.D
    DE = 1.D
    DO 26   I = 1,IW6
    DE = BLIST(IRIO-I)/(ENR(IRIO-I)*ENR(IRIO-I))*DE
    CORA = CORA + DE
    IF (DABS(DE/CORA).LT.1.D-27) GO TO 27
 26 CONTINUE
 27 DL = (ENRC-ENR(IRIO))/(CORA+CORB)
    CL=DL+CL
    IF (DABS(DL/CL).LT.1.0D-24) GO TO 22
    IFC=IFC+1
    IF (IFC.LT.20) GO TO 17
 22 EIG(L+1-M) = CL
    DO 118 I = 1,IBLIM
    ARR = IX + 2.D*I
    EA = 2.D*EM + 2.D*ARR
118 DND0(I) = (EA-1.D)*(EA+1.D)*ENR(I)/((EA-ARR)*(EA-ARR-1.D)*HSQ)
    DN(1) = DND0(1)
    DO 119 J = 2,IBLIM
119 DN(J) = DND0(J)*DN(J-1)
    IW21 = 2*IBLIM
    W28 = 0.D
    IWW = 0
    IF (M.EQ.0) GO TO 165
    M2 = 2*M
    DO 120 J = ISC,IW21,2
    W26 = 1.D
    DO 160 K = 1,M2
160 W26 = W26*(J+K)
    IWW = IWW + 1
120 W28 = W28 + W26*DN(IWW)
    GO TO 167
165 DO 166 J = ISC,IW21,2
    IWW = IWW + 1
166 W28 = W28 + DN(IWW)
167 DLIST(1) = FACT(L+M+1)/FACT(L-M+1)/(W28+FACT(M+M+IX+1))
    DO 121 J = 1, IBLIM
121 DLIST(J+1) = DN(J)*DLIST(1)
    C101 = 1.D
    IF (NRM.EQ.3) GO TO 6257
    IC30 = L-M
    IF (2*(IC30/2).EQ.IC30) 70,71
 70 IC1 = IC30/2
    DNUM = FACT(M+M+1)/FACT(M+1)*DLIST(1)
```

```
      DO 717 J = 1,48
  717 DNUM = FACT(J+J+M+M+1)/(4.D**J*FACT(J+1)*FACT(J+M+1))*DLIST(J+1)
     1-DNUM
      C101 = FACT(L+M+1)/(2.D**IC30*FACT(IC1+1)*FACT((L+M)/2+1)*DNUM)
      IF (2*(IC1/2).NE.IC1) C101 = -C101
      GO TO 6257
   71 IC1 = (IC30-1)/2
      DNUM = FACT(M+M+2)/FACT(M+1)*DLIST(1)
      DO 718 J = 1,48
  718 DNUM = FACT(J+J+M+M+3)/(2.D**(J+J+1)*FACT(J+1)*FACT(J+M+2))*DLIST(
     1J+1) - DNUM
      C101 = FACT(L+M+2)/(2.D**IC30*FACT(IC1+1)*FACT((L+M+1)/2+1)*DNUM)
      IF (2*(IC1/2).NE.IC1) C101 = -C101
 6257 CONTINUE
      DO 2 INA = 1,NARG
      ARG = ARG1 + (INA-1)*DARG
      AAA = ARG
      INX = 1
      IF (ARG.GE.0.D) GO TO 40
      INX = -1
      ARG = -ARG
      GO TO 41
   40 IF (ARG.LE.90.D) GO TO 41
      INX = -1
      ARG = 180.D0-ARG
   41 IF (KTP.EQ.1) ARG = DCOS(.0174532925199432957692369D*ARG)
      IF (DABS(1.D0-ARG).LT.1.D-10) ARG = 1.D
      MIN = L-M+10
      J = 2 + IX + M
      K = 1
      S = 0.D
      SD = 0.D
      SDD = 0.D
      DO 80 I = J,MIN,2
      II = I + 2
      S = S + DLIST(K)*P(INA,I)
   80 K = K + 1
   84 TERM = DLIST(K)*P(INA,II)
      S = S + TERM
      IF (S.EQ.0.D) GO TO 81
      IF (DABS(TERM/S).LT.TNS) 81,82
   82 K = K + 1
      II = II + 2
      GO TO 84
   81 IF (INX.EQ.-1.AND.2*((M+L)/2).NE.M+L) S = -S
      S = S*C101
      IF (M.EQ.1.AND.ARG.EQ.1.D) GO TO 50
      K = 1
      DO 85 I = J,MIN,2
      II = I + 2
      SD = SD + DLIST(K)*PD(INA,I)
   85 K = K + 1
   88 TERM = DLIST(K)*PD(INA,II)
      SD = SD + TERM
      IF (SD.EQ.0.D) GO TO 86
      IF (DABS(TERM/SD).LT.TNS) 86,87
   87 K = K + 1
      II = II + 2
      GO TO 88
```

```
 86 IF (INX.EQ.-1.AND.2*((M+L)/2).EQ.M+L) SD = -SD
    SD = SD*C(0)
    IF (M.EQ.3.AND.ARG.EQ.1.D) GO TO 51
    K = 1
    DO 89 I = J.MIN.2
    II = I + 2
    SDD = SDD + DLIST(K)*PDD(INA.I)
 89 K = K + 1
 97 TERM = DLIST(K)*PDD(INA.II)
    SDD = SDD + TERM
    IF (SDD.EQ.0.D) GO TO 95
    IF (DABS(TERM/SDD).LT.TNS) 95.96
 96 K = K + 1
    II = II + 2
    GO TO 97
 95 IF (INX.EQ.-1.AND.2*((M+L)/2).NE.M+L) SDD = -SDD
    SDD = SDD*C(0)
    PRINT 101. M.H.L.AAA.S.SD.SDD
101 FORMAT ( 9X I2.F8.3.I4.F9.3.10XD17.10.2(16XD17.10))
    GO TO 2
 50 PRINT 102.M.H.L.AAA.S
102 FORMAT ( 9X I2.F8.3.I4.F9.3.10XD17.10.20X*INFINITY*25X*INFINITY*)
    GO TO 2
 51 PRINT 103.M.H.L.AAA.S.SD
103 FORMAT ( 9X I2.F8.3.I4.F9.3.10XD17.10.16XD17.10.20X*INFINITY*)
  2 CONTINUE
    STOP
    END


    SUBROUTINE POLY(INA.M.LN.ARG.P1.P2.P3)
    TYPE DOUBLE FACT.P.ANGLE.TERM.SUM.DM.DN.R.PP.SN.P1.P2.P3.ZP.ARG.S.
   *FBCT.Y
    DIMENSION P(125).P1(20.125).P2(20.125).P3(20.125).Y(125)
    COMMON /BLK1/FACT(171).FBCT(300)
    II = 0
    IF (DABS(1.D0-ARG).LT.1.D-10) ARG = 1.D
    ANGLE = DATAN2(DSQRT(1.D0 - ARG*ARG).ARG)
    LNN=LN+1
    LDN = LN + 2
    DO 2 N = 1.LNN
    P(N) = 0.D
  2 Y(N) = DCOS((N-1)*ANGLE)
    MM=M+1
    M2 = 2 + M
    SN = DSIN(ANGLE)
    IF ((ANGLE.NE.0.).AND.((ANGLE.LT.3.1415926).OR.(ANGLE.GT.3.1415926
   19))) GO TO 6
    ARG = 1.
    IF (M.NE.0) GO TO 8
    DO 5 N=1.LNN
  5 P(N)= 1.D
    IF( ANGLE .EQ.0) GO TO 14
    DO 4 N=2.LNN.2
  4 P(N)=-P(N)
    GO TO 14
  6 IF ((ANGLE.LT.1.570796320).OR.(ANGLE.GT.1.570796329))GO TO 10
    ARG = 0.
```

```
      IF (M.EQ.0) 3,9
    3 P(1) = 1.D
      MA = 2
      GO TO 1
    9 MA = M
    1 DO 7 N = MA,LN
      P(N+1)= FACT(N+M)/(FACT((N+M)/2)*FACT((N-M)/2+1)*2.D**(N-1))
      IF(2*((M+N)/2).NE.(M+N))  P(N+1)=0.D
    7 IF (2*((N-M)/4).NE.(N-M)/2)  P(N+1)=-P(N+1)
      GO TO 14
   10 P(MM) = SN**M*FACT(2*M+1)/(FACT(MM)*2.D**M)
   11 DO 13 N=MM,LN
      S = 0.D
      IN = (N-M-1)/2
      DO 12 I = II,IN
   12 S = S+FACT(2*M+2*I+1)*FBCT(2*N-2*I+1)*Y(N-M-2*I+1)/(2.D**
     1(M+N)*FBCT(N-I+1)*FACT(MM+I)*FACT(I+1)*FACT(N-M-I+1))
      P(N+1)= SN**M*FACT(MM)/(FACT(2*M+1)*2.D**(N-2*M-1))*S
   13 IF (2*((M+N)/2).EQ.M+N) P(N+1)=P(N+1)+SN**M*FACT(MM)*(FACT(N+MM)/
     1(FACT((N+M)/2+1)*FACT((N-M)/2+1)))**2/(FACT(2*M+1)*2.D**(2*N-M))
      GO TO 14
    8 IF (ANGLE.NE.0.D) ARG = -1.D
   14 DO 15 I = II,LN
   15 P1(INA,LDN-I) = P(LNN-I)
      P1(INA,1) = 0.D
      IF (ARG.EQ.1.D) 40,41
   40 IF (M-1) 20,21,22
   20 P2(INA,2) = 0.D
      P2(INA,3) = 1.D
      P3(INA,2) = P3(INA,3) = 0.D
      DO 30 I = M2,LN
      P2(INA,I+2) = (I+1.D)*I/2.D
   30 P3(INA,I+2) = FACT(I+3)/(8.D*FACT(I-1))
   21 RETURN
   22 IF (M.EQ.2) 23,24
   23 DO 31 I = M,LN
      P2(INA,I+2) = -FACT(I+3)/(4.D0*FACT(I-1))
      P3(INA,I+2) = P2(INA,I+2)
   31 IF(I.GT.2) P3(INA,I+2) = P2(INA,I+2) - FACT(I+4)/(12.D*FACT(I-2))
      RETURN
   24 DO 32 I = M,LN
   32 P2(INA,I+2) = 0.D
      IF (M.EQ.3) RETURN
      IF (M.EQ.4) 25,26
   25 DO 33 I = M,LN
   33 P3(INA,I+2) = FACT(I+5)/(48.D*FACT(I-3))
      RETURN
   26 DO 34 I = M,LN
   34 P3(INA,I+2) = 0.D
      RETURN
   41 ZP =  1.D/(1.D0-ARG*ARG)
      DO 16 JX = M2,LDN
      P2(INA,JX) = ZP*((M+JX-2)*P1(INA,JX-1)-(JX-2)*ARG*P1(INA,JX))
   16 P3(INA,JX) = ZP*(2.D*ARG*P2(INA,JX)+(ZP*M*M-(JX-2)*(JX-1))
     1*P1(INA,JX))
      RETURN
      END
```

```
      SUBROUTINE EIGEN (A,VALU,N,NBL)
      DIMENSION A(80,N),VALU(80),DIAG(80),Q(80),VALL(80)
      ANORM = A(1,1)*A(1,1) + A(2,2)*A(2,2)
      DO 6 I = 3,N
    6 ANORM = ANORM + A(I,I)*A(I,I)+ 2*A(I-2,I)*A(I-2,I)
      ANORM = SQRTF(ANORM)
      NN=N-2
      DO 160 I=1,NN
      II=I+2
      DO 160 J=II,N
      T1=A(I,I+1)
      T2=A(I,J)
      IF (T2.EQ.0.) GO TO 160
      T=1./SQRT(T1*T1+T2*T2)
      SIN=T2*T
      COS=T1*T
      DO 105 K=I,N
      T2=COS*A(K,I+1)+SIN*A(K,J)
      A(K,J)=COS*A(K,J)-SIN*A(K,I+1)
  105 A(K,I+1)=T2
      DO 125 K=I,N
      T2=COS*A(I+1,K)+SIN*A(J,K)
      A(J,K)=COS*A(J,K)-SIN*A(I+1,K)
  125 A(I+1,K)=T2
  160 CONTINUE
      DO15I=1,N
      DIAG(I)=A(I,I)
      Q(I)=A(I,I-1)*A(I,I-1)
      VALL(I) = -ANORM
   15 VALU(I) = ANORM
      I=1
   18 TAU=(VALL(I)+VALU(I))/2.
      MATCH=0
      T0=0.
      T1 = 1.
      DO20J=1,N
      T2=(DIAG(J)-TAU)*T1-Q(J)*T0
      IF (ABS(T2).LT.1.E130) GO TO 26
      T2 = T2*1.E-200
      T1 = T1*1.E-200
   26 IF((T1.NE.0.).AND.((T2*T1).LE.0.))  MATCH=MATCH+1
      T0=T1
   20 T1=T2
      DO 25 K = 1, MATCH
   25 VALU(K) = TAU
      NATCH = MATCH + 1
      DO 30 K = NATCH, NBL
   30 IF (VALL(K).LT.TAU) VALL(K) = TAU
   40 IF (VALU(I).EQ.0.) VALU(I) = 1.E-20
      IF (ABS((VALU(I)-VALL(I))/VALU(I)).GT.1.E-7) GO TO 18
      I=I+1
      IF(I.LE.NBL)  GO TO 40
      END
```