UNCLASSIFIED

NRL Report 6109

A Neliac Program for Calculating Prolate Radial Spheroidal Wave Functions

S. HANISH

Transducer Branch
Sound Division

PLEASE RETURN THIS COPY TO:

NAVAL RESEARCH LABORATORY
WASHINGTON, D.C. 20375
ATTN: CODE 2628

Because of our limited supply you are requested to return this copy as soon as it has served your purposes so that it may be made available to others for reference use. Your cooperation will be appreciated.

NDW-NRL-5070/2616 (1-84)

July 27, 1964

U.S. NAVAL RESEARCH LABORATORY
Washington, D.C.

CONTENTS

Abstract 1
Problem Status 1
Authorization 1

INTRODUCTION 1

THEORETIC CONSIDERATIONS 1

PROCEDURE FOR USING NELIAC VERSION 1

THE NELIAC PROGRAM 2

REFERENCES 2

APPENDIX A—Equations Referred to in Appendix B 3

APPENDIX B—The Neliac Program 7

UNCLASSIFIED

A Neliac Program For Calculating Prolate Radial Spheroidal Wave Functions

S. HANISH

Transducer Branch
Sound Division

This report contains a complete transcription of a digital computer program for the computation of prolate radial spheroidal wave functions. The program is written in the Neliac dialect of Algol 60. A brief explanation of the sequence of commands is made by listing all the formulas used by the author in its construction and by indicating in the body of the program the sets of commands corresponding to each formula in the list. The program has been compiled by the Neliac-N compiler of the NAREC and has worked successfully for the following ranges of input parameters: $0 \leq m \leq 10$, $0 \leq \ell \leq 20$, $0 \leq h \leq 20$, $\xi \geq 1.000005$, with a precision of at least four significant digits.

INTRODUCTION

In NRL Report 5947 (1), the author published a digital computer program for calculating prolate radial spheroidal wave functions. The program was written in NIP (NAREC Interpretive Program) language. After the appearance of that report a new programming language became available at the NAREC computing center for the rapid compilation of computer programs. This was the Neliac dialect of Algol, implemented on the NAREC by the Neliac-N compiler. When Neliac appeared it was thought advisable to rewrite the program of Report 5947 in this new language. The following report contains a complete transcription of this Neliac version, together with explanatory notes. Examination of this version will show that no portion of it (excluding the input procedure) is machine dependent. It is therefore valuable to users in other institutions.

THEORETIC CONSIDERATIONS

NRL Report 5947 contains a summary of all the pertinent formulas and references needed for the computation of the prolate radial spheroidal wave functions. The reader is referred to Flammer (2) for additional details. For convenience we list in Appendix A all of the equations referred to in the explanatory notes in the main body of the program itself (Appendix B). The source of each formula is noted in the heading above it.

This program was compiled on the NAREC by the Neliac-N compiler and tested for various combinations of the six input parameters m, ℓ, h, zero, R, ξ. In all cases the results have shown greater precision by one or two significant digits than that achieved by the outputs of the earlier NIP program of Report 5947.

PROCEDURE FOR USING NELIAC VERSION

The tape containing the Neliac version of our program is designated by the identifying number F5061. When the Neliac-N compiler is in the NAREC memory, this tape may be read in directly by the automatic reader and the program compiled without additional commands. Six input parameters must be in the memory of the computer before computation can begin. In addition, 51 spheroidal wave function expansion constants must be found on address DD, DD + 1, etc., of the program, and the value of the prolate spheroidal eigenvalue must be found in address 0923 of the NAREC memory, or cell 2 of the program in Appendix B.

NRL Problem S02-07; Project RF 001-03-44-4052. This is an interim report on one phase of the problem; work is continuing. Manuscript submitted April 8, 1964.

The meaning of the input parameters may be understood by referring to Eq. (14) of Appendix A. Here m and ℓ are modal (integral) numbers, h is $kd/2$, where $k = 2\pi/\lambda$, λ is the wavelength, and d is the interfocal distance of the prolate spheroid. The symbol R represents an address which contains a zero for the prolate case. The symbol ζ is the radial spheroidal coordinate.

For any computer on which this program is to be executed we must make input assignments in conformity with the following input table (Table 1). For convenience of NAREC users the input assignments are shown in their correct NAREC locations.

TABLE 1
Input Assignments

NAREC Address	Input Requirement
0800	m, in Neliac floating point
0801	ℓ, in Neliac floating point
0802	h, in Neliac floating point
0803	Zero*
0804	R, either zero (prolate case) or hexidecimal 1 (oblate case)
0805	ξ, in Neliac floating point
1A01, 1A02, ., ., 1A33	51 prolate spheroidal wave function expansion constants, $d_r(h/m\ell)$ in Neliac floating point.
0923	Prolate spheroidal eigenvalue, in Neliac floating point.

*This address is included to permit use of Tape F5188 with the above noted input data.

When the input is in the memory, the compiled program may be executed by starting at the command "IO Statement" which appears on page 46 of the program (last page of Appendix B). Previous compilation on NAREC has shown this command to be located at address 3876. Hence the NAREC starting order is LO 3876.

The program automatically prints out the value of $R^{(2)}_{m\ell}$ and the value of $dR^{(2)}_{m\ell}/d\,\xi$.

THE NELIAC PROGRAM

Appendix B contains a complete reprint of the entire contents of Tape F5061. On most pages of this reprint we have appended on the right a series of brief explanatory remarks and references to formulas. By consulting Appendix A the reader may find the formula whose compilation is being effected by the series of command following the corresponding arrow in the explanatory remarks.

REFERENCES

1. Hanish, S., "A Digital Computer Program in NAREC Code for Calculating Prolate Radial Spheroidal Wave Functions," NRL Report 5947, July 1963
2. Flammer, C., "Spheroidal Wave Functions," Stanford Univ. Press, Stanford, Calif., 1957

3. Meixner, J., "Asymptotische Entwicklung der Eigenwerte und Eigenfunktionen der Differentialgleichungen der Sphäroid-Funktionen und der Mathieuschen Funktionen," Zeitshrift angew. *Math. Mech.* **28**:304-310 (1948)
4. Stratton, J.A., Morse, P.M., *et al.*, "Spheroidal Wave Functions," New York:Wiley, 1956
5. Sternberg, D., "A Computation Procedure for the Scattering of Sound By a Prolate Spheroid," Columbia Univ., Hudson Lab., Tech. Report 61, 1958

Appendix A

Equations Referred to in Appendix B

Equation (1)—Ref. 3, p. 11

$$U_n = \frac{(n+2m)(n+2m-1)}{(2n+2m+1)(2n+2m-1)}$$

Equation (2)—Ref. 3, p. 11

$$V_n = \frac{2(n+m)(n+m+1)-2m^2-1}{(2n+2m+3)(2n+2m-1)} + \frac{(n+m)(n+m+1)-A_{m\ell}(h)}{h^2}$$

Equation (3)—Ref. 3, p. 11

$$W_n = \frac{(n+2)(n+1)}{(2n+2m+3)(2n+2m+1)}$$

Equation (4a)—Ref. 3, p. 18

$$\frac{d_{-2m}}{d_{-2m+2}} = -\frac{U_{-2m+2}}{V_{-2m}}$$

Equation (4b)—Ref. 3, p. 18

$$\frac{d_{-2m+1}}{d_{-2m+3}} = -\frac{U_{-2m+3}}{V_{-2m+1}}$$

Equation (5a)—Ref. 3, p. 18

$$\frac{d_n}{d_{n+2}} = -\frac{U_{n+2}}{V_n + W_{n-2}\left(\frac{d_{n-2}}{d_n}\right)}$$

Equation (5b)—Ref. 3, p. 19

$$\frac{d_{-2m-2|\rho}}{d_{-2m}} = \frac{\frac{1}{(2m+1)(2m-1)}}{V_{-2m-2} + W_{-2m-4}\left(\frac{d_{-2m-4}}{d_{-2m-2}}\right)}$$

Equation (5c)—Ref. 3, p. 19

$$\frac{d_{-2m-1|\rho}}{d_{-2m+1}} = -\frac{\frac{1}{(2m-1)(2m-3)}}{V_{-2m-1} + W_{-2m-3}\left(\frac{d_{-2m-3}}{d_{-2m-1}}\right)}$$

Equation (6a)—Ref. 2, p. 86

$$\text{Deesum even} = \sum_{r=0}^{\infty}{}' \frac{(-1)^{r/2}\,(r+2m)!}{2^r\left(\frac{r}{2}\right)!\,\left(\frac{r+2m}{2}\right)!}\, d_r(h \mid m\ell)$$

Equation (6b)—Ref. 2, p. 86

$$\text{Deesum odd} = \sum_{r=1}^{\infty}{}' \frac{(-1)^{(r-1)/2}\,(r+2m+1)!}{2^r\left(\frac{r-1}{2}\right)!\,\left(\frac{r+2m+1}{2}\right)!}\, d_r(h \mid m\ell)$$

Equation (7a)—Ref. 2, p. 86

$$\text{Flammstrattn} = \frac{\dfrac{(-1)^{(\ell-m)/2}\,(\ell+m)!}{2^{\ell-m}\left(\frac{\ell-m}{2}\right)!\,\left(\frac{\ell+m}{2}\right)!}}{\text{Eq. (6a)}},\ (\ell-m)\ \text{even}$$

Equation (7b)—Ref. 2, p. 86

$$\text{Flammstrattn} = \frac{\dfrac{(-1)^{(\ell-m-1)/2}\,(\ell+m+1)!}{2^{\ell-m}\left(\frac{\ell-m-1}{2}\right)!\,\left(\frac{\ell+m+1}{2}\right)!}}{\text{Eq. (6b)}}$$

Equation (8a)—Ref. 2, p. 33

$$k^{(2)}_{m\ell}(h) = \text{Joinfac} = \frac{2^{\ell-m}(2m)!\,\left(\frac{\ell-m}{2}\right)!\,\left(\frac{\ell+m}{2}\right)!\ d_r(h \mid m\ell)}{(2m-1)\ m!\ (\ell+m)!\ k^{m-1}}$$

$$\times\left[\frac{(\ell+m)!}{(\ell-m)!}\times \text{Flammstrattn}\right],\ (\ell-m)\ \text{even}$$

Equation (8b)—Ref. 2, p. 33

$$K^{(2)}_{m\ell}(h) = \text{Joinfac} = \frac{2^{\ell-m}(2m)!\,\left(\frac{\ell-m-1}{2}\right)!\,\left(\frac{\ell+m+1}{2}\right)!\ d_r(h \mid ml)}{(2m-3)(2m-1)m!\ (\ell+m+1)!\ C^{m-2}}$$

$$\times\left[\frac{(\ell+m)!}{(\ell-m)!}\times \text{Flammstrattn}\right],\ (\ell-m)\ \text{odd}$$

Equation (9)—Ref. 5, p. 31

$$Q^{m}_{-m} = -\frac{1}{\sqrt{\xi^2-1}}\left\{2(m-1)\ Q^{m-1}_{-m+1} + \frac{[2(m-1)]!}{2^{m-1}(m-1)!}\ \xi(\xi^2-1)^{(m-1)/2}\right\}$$

UNCLASSIFIED

Equation (10)—Ref. 5, p. 31

$$Q^{m}_{-m+1} = -\frac{1}{\sqrt{\xi^2 - 1}}\left[(2m-3)\, Q^{m-1}_{-m+2} + \xi\, Q^{m-1}_{-m+1}\right]$$

Equation (11)—Ref. 5, p. 31

$$Q^{1}_{-1} = -\frac{\xi}{\sqrt{\xi^2 - 1}};\; Q^{1}_{0} = -\frac{1}{\sqrt{\xi^2 - 1}}$$

Equation (12)—Ref. 5, p. 32

$$Q^{m}_{m+n} = \frac{1}{n}\left[(2m + 2n - 1)\,\xi\, Q^{m}_{m+n-1} - (2m + n - 1)\, Q^{m}_{m+n-2}\right]$$

Equation (13)—Ref. 5, p. 32

$$Q^{m'}_{m+n} = \frac{1}{\xi^2 - 1}\left(m\,\xi\, Q^{m}_{m+n} + \sqrt{\xi^2 - 1}\, Q^{m+1}_{m+n}\right)$$

Equation (14)—Ref. 2, p. 33

$$R^{(2)}_{m\ell}(h, \xi) = \frac{1}{K^{(2)}_{m\ell}(h)}\left[\sum_{r=-2m,-2m+1}^{\infty} d_r\,(h \mid m\ell)\, Q^{m}_{m+1}(\xi) + \sum_{r=2m+2,2m+1}^{\infty}{}' d_{r|\rho}(h \mid m\ell)\, P^{m}_{r-m-1}(\xi)\right]$$

UNCLASSIFIED

APPENDIX B
THE NELIAC PROGRAM

F5061 - Compute Radial Functions of Second Type

```
5
(:PAGE F5061-1)
COMPUTE RADIAL FUNCTIONS OF SECOND TYPE, , ..

5
(COMMENTS: THIS FLOWCHART CALCULATES THE EXPANSION CONSTANTS OF NEGATIVE
   SUBSCRIPT AND THE SPECIAL DEERHO CONSTANTS.)

   ELL, EMM, TESTDIGIT, TESTNUMBER, TESTNUMBER TWO, TESTNUMBER THREE, ENN,
   RATIO (50). TEST DIGIT TWO, TRACK, T, YEWENN. VEEENN. DOUBLE YEW ENN.
   RATIO DENOM. R, DEE ZERO. DEE NEGFREE. DEE NEG (20). DEERHO FREE.
   Q, H, NUMBER TWO. DEERHORATIO (50). NUMBER, FLOATNUM. RUSH. A,
   AA, DEENEGFLOAT. DEERHO (21). NUMBERFOUR. NUMBERFIVE, HURRY. NUMBERSIX.
   DEERATIO (50). DD = {#1A00}. EXPANSION (51). AF, DEEONE.
   EIGENVALUE.;

CALCNEGSUBSCRDEE:

   I= 0(1)50 {DD [I+1] → EXPANSION[I]},
```

← There must exist in the memory at location DD+1, DD+2, etc., 51 spheroidal wave expansion constants with positive subscripts. These are obtained from Tape F5188.

```
(:PAGE F5061-2)
ODD EVEN TEST: ELL - EMM → TEST DIGIT,

   TEST DIGIT = 0: EVENSIDE. TEST DIGIT/2 → TEST NUMBER, TEST NUMBER *2
      → TEST NUMBER TWO, TEST NUMBER TWO - TEST DIGIT → TEST NUMBER THREE;

   TEST NUMBER THREE = 0: EVENSIDE. ODDSIDE.
   EVENSIDE: EMM=0: EXPANSION [0] → DEENEG [0], CALCDEERHOEVEN.;
      EXPANSION [0] → DEEZERO,                                   ← Start compute Eq. (4a)
      -2 * EMM +2 → ENN, YEWSUBENN (ENN, EMM; YEWENN),
      -2* EMM → ENN, VEESUBENN (ENN, EMM; VEE ENN),
      -YEWENN/VEE ENN → RATIO [0], -1 → T,

      I = 2(2)40 { -2 * EMM + I → TEST DIGIT TWO → TRACK,
   TRACK ≥ 0: CALCNEGTWO.; TEST DIGIT TWO +2 → ENN,
      YEWSUBENN (ENN, EMM; YEWENN), TEST DIGIT TWO → ENN,
      VEESUBENN (ENN, EMM; VEE ENN),
      TEST DIGIT TWO -2 → ENN,

(:PAGE F5061-3)
   DOUBLEYEWSUBENN (ENN, EMM; DOUBLEYEWENN), (I-2)/2→ T,
   DOUBLEYEWENN * RATIO [T] + VEE ENN → RATIO DENOM, - YEWENN/RATIO DENOM
   → RATIO [T+1]},                                               ← Complete compute Eq. (4a)
```


UNCLASSIFIED

```
CALCNEGTWO: T+1 → R, DEEZERO → DEENEGFREE, J = R(-1) 0 {RATIO [J] *      ← Computes all d_-n (n even)
      DEENEGFREE → DEENEG [J] → DEENEGFREE},

      CALCDEERHOEVEN.

ODDSIDE: EMM = 0: EXPANSION [0] → DEENEG [0], CALCDEERHOODD.;
         EXPANSION [0] → DEEONE,                                         ← Start compute Eq. (4b)
      -2 * EMM +3 → ENN, YEWSUB ENN (ENN, EMM; YEWENN),
      -2 * EMM +1 → ENN, VEESUBENN (ENN, EMM; VEEENN),
      -YEWENN/VEE ENN → RATIO [0], -1 → T,

      I = 2(2)40 {-2*EMM +1 + I → TESTDIGIT TWO → TRACK,

TRACK ≥ 0: CALC NEGONE.; TEST DIGIT TWO +2 → ENN,
      YEWSUBENN (ENN, EMM; YEW ENN),

(:PAGE F5061-4)
      TEST DIGIT TWO → ENN, VEESUBENN (ENN, EMM; VEE ENN),
      TEST DIGIT TWO -2 → ENN, DOUBLEYEWSUBENN (ENN, EMM; DOUBLEYEWENN),
      (I-2)/2 → T, DOUBLE YEWENN * RATIO [T] + VEEENN →
      RATIO DENOM, - YEWENN/RATIODENOM → RATIO [T+1]},                   ← Finish compute Eq. (4b)

CALCNEGONE: T+1 → R, DEEONE → DEENEGFREE, J = R (-1) 0 {                 ← Compute all d_-n (n odd)
```

```
      RATIO [J] * DEENEGFREE → DEENEG [J] → DEENEGFREE},
      CALCDEERHOODD.

CALCDEERHOEVEN: -2 * EMM - 4 → Q, |Q| → Q, 0 → DEERHOFREE,        ← Start compute Eq. (5a)

      K = 40 (-2) Q {- K +2 → ENN, - ENN/2 → H,

      YEWSUBENN (ENN, EMM; YEWENN), - K → ENN,
      VEESUBENN (ENN, EMM; VEEENN), - K -2 → ENN,
      DOUBLEYEWSUBENN (ENN, EMM; DOUBLEYEWENN),

(:PAGE F5061-5)
DOUBLEYEWENN * DEERHOFREE + VEEENN → NUMBERTWO, - YEWENN/
         NUMBERTWO → DEERHORATIO [H] → DEERHO FREE},

         -2 * EMM -2 → ENN, VEESUBENN (ENN, EMM; VEEENN),          ← Compute Eq. (5b)
         (2*EMM +1) * (2*EMM-1) → NUMBER, FX TO FL (NUMBER; FLOATNUM),
         -2 * EMM -4 → ENN,
            DOUBLEYEWSUBENN (ENN, EMM; DOUBLEYEWENN),
            DOUBLEYEWENN * DEERHORATIO [H] + VEEENN
            → RUSH, FLOATNUM * RUSH → RUSH,
               1.0 / RUSH → DEERHORATIO [H-1],                     ← Finish compute Eq. (5a) and Eq. (5b)

         DEENEG [0] → DEENEG FLOAT, H -1 → A, 0 → AA,
```


UNCLASSIFIED

```
L = A (1) 19 {AA +1 → AA, DEERHORATIO [A] * DEENEGFLOAT → DEERHO [AA] →
        DEENEGFLOAT, A+1 → A}, END 4.
CALCDEERHOODD: -2 * EMM -3 → Q, |Q| → Q, 0 → DEERHOFREE,
      K = 41 (-2) Q {-K +2 → ENN, - (ENN +1)/2→ H,

(:PAGE F5061-6)
        YEWSUBENN (ENN, EMM; YEWENN), -K → ENN,

        VEESUBENN (ENN, EMM; VEEENN), -K -2 → ENN,

        DOUBLE YEWSUBENN (ENN, EMM; DOUBLEYEWENN),

DOUBLE YEW ENN * DEERHOFREE + VEEENN → NUMBERFOUR, - YEWENN /
      NUMBERFOUR → DEERHORATIO [H] → DEERHOFREE},

      -2 * EMM -1 → ENN, VEESUBENN (ENN, EMM; VEEENN),
      (2*EMM -1) * (2* EMM-3) → NUMBER FIVE, FX TO FL
      (NUMBER FIVE; HURRY), - 2 * EMM - 3 → ENN,
      DOUBLEYEWSUBENN (ENN, EMM; DOUBLEYEWENN),
      DOUBLEYEWENN * DEERHORATIO [H] + VEEENN
      →RUSH, HURRY * RUSH → RUSH,
      -1.0 / RUSH → DEERHORATIO [H-1],
            DEENEG [0] → DEE NEGFLOAT, H -1 → A, 0 → AA,
```

Compute all $d_{-n|\rho}$ (n even)

Start compute Eq. (5a)

Compute Eq. (5b)

Finish compute Eq. (5a) and (5b)

```
L = A (1) 19 {AA +1 → AA, DEERHORATIO [A] * DEENEGFLOAT → DEERHO [AA]
            → DEENEGFLOAT, A+1 → A}, END 4.
```

Compute all $d_{-n|\rho}$ (n odd)

```
END 4: CONNECTION  3. ..

5
(:PAGE F5061-7)
(COMMENTS: THIS FLOWCHART CONTAINS THE SUBROUTINE FOR CONVERTING EXPANSION
CONSTANTS DEFINED BY FLAMMER INTO EXPANSION CONSTANTS DEFINED BY STRATTON,MORSE
ET AL.  THE RESULT IS FOUND IN LOCATION CHU MORSE.)

EBB, ECC, CELLONE, CELL2. CELL3, CELL 4. CELL 5, CELL 6.  CELL 7,
CELL 7A. CELL9A. CELL14A.  CELL 8, CELL 9, CELL 10, CELL 10A. CELL 11A. CELL 12.
DEESUMNUMBER. CELL 80, CELL 81, CELL 82, CELL 83,

CHU MORSE. CELL 22. CELL 13, CELL 14, CELL 15, CELL 15A. CELL 16. CELL 17,
CELL 17A. CELL 18, CELL 19, CELL 19A. CELL 20. CELL 20A. CELL 21.;

FLAMMSTRATTN (EBB, ECC, CHU MORSE.): {
      EBB - ECC → CELL 80,

CELL 80 = 0: EVEN FLAMMSTRATTN. CELL 80/2 → CELL 81, CELL 81 * 2 → CELL 82,
      CELL 80 - CELL 82 → CELL 83;
CELL 83 = 0: EVENFLAMMSTRATTN. ODD FLAMMSTRATTN.
```

Subroutine for computing Eq. (7a) or Eq. (7b)

UNCLASSIFIED

```
(:PAGE F5061-8)

EVENFLAMMSTRATTN: (EBB-ECC)/2 → CELLONE,                          ← Start Eq. (7a)
   FACTORIAL (CELL ONE; CELL 2), EBB + ECC → CELL 3, FACTORIAL (CELL 3; CELL 4),
   (EBB + ECC)/2 → CELL 5, FACTORIAL (CELL 5; CELL 6), 1 → CELL 7,

CELL ONE = 0:
          ;
   I = 1(1) CELL ONE {-1 * CELL 7 → CELL 7};
   EBB - ECC → CELL 8, 1 → CELL 9, M = 1 (1) CELL 8{2 * CELL 9 → CELL 9},
   FXTOFL (CELL7; CELL7A),
FXTOFL(CELL9; CELL9A),  CELL 7A * CELL 4 →
   CELL 10A,  CELL 9 A* CELL 2 * CELL 6 → CELL 11A,
 CELL 10A/CELL 11A → CELL 12, DEESUMEVEN
   (ECC; DEESUM NUMBER), CELL 12/ DEESUMNUMBER → CHU MORSE, END 3.  ← Finish Eq. (7a)

(:PAGE F5061-9)
ODDFLAMMSTRATTN: (EBB - ECC -1)/2 → CELLONE,                      ← Start Eq. (7b)
   FACTORIAL (CELLONE; CELL 2), EBB + EMM +1 → CELL 3,FACTORIAL (CELL 3;CELL 4),
   (EBB + ECC +1)/2 → CELL 5, FACTORIAL (CELL 5; CELL 6), 1 → CELL 7,

CELL ONE = 0:
   ;  I = 1(1) CELL ONE {-1 * CELL 7 → CELL 7}; EBB - ECC → CELL 8, 1 → CELL9,
   M = 1(1) CELL8{2* CELL9 → CELL9},
```

```
    FXTOFL(CELL7; CELL7A), FXTOFL(CELL9; CELL9A), CELL 7A * CELL 4 → CELL 10A,
CELL 9A * CELL 2 * CELL 6 → CELL 11A,
    CELL 10A/CELL 11A → CELL 12, DEESUMODD (ECC; DEESUMNUMBER),
  CELL 12/DEESUMNUMBER → CHU MORSE, END 3.                          ← Finish Eq. (7b)
(:PAGE F5061-10)
END 3:}
,DEESUMEVEN (ECC, DEESUMNUMBER.): {0 → CELL 22, J = 0 (2)96{J/2 →   ← Compute Eq. (6a)

  CELL 13, 1 → CELL 14,
CELL 13 = 0:
          ;
  K = 1(1) CELL 13 {-1 * CELL 14 → CELL 14};
  J +2 * ECC → CELL 15, FACTORIAL (CELL 15; CELL 15A), FACTORIAL (CELL 13;
  CELL 16), CELL 15/2 → CELL 17, FACTORIAL (CELL 17; CELL 17A),
  1.0 → CELL 20A, M = 1 (1) J {2 .0 * CELL 20A → CELL20A},
   FXTOFL (CELL14; CELL14A),
  CELL 14A * CELL 15A → CELL 19A, CELL 20A * CELL16
  * CELL 17A → CELL 20,  CELL 19A/CELL 20 → CELL 21,
  CELL 21 * EXPANSION [CELL 13] + CELL 22 → CELL 22, CELL 22 → DEESUMNUMBER}},

(:PAGE F5061-11)
DEESUMODD (ECC, DEESUMNUMBER.): {0 → CELL 22, J = 1 (2) 97 {(J-1)/2   ← Compute Eq. (6b)
  → CELL 13, 1 → CELL 14,
```


UNCLASSIFIED

```
CELL 13 = 0: ; K = 1(1) CELL 13 {-1 * CELL 14 → CELL 14};
   J +2 * ECC +1 → CELL 15, FACTORIAL (CELL 15; CELL 15A),
   FACTORIAL (CELL 13; CELL 16), CELL 15/2 → CELL 17,
   FACTORIAL (CELL17; CELL 17A),
   1 .0 → CELL 20A,  M = 1 (1) J {2.0 * CELL 20A → CELL 20A},
FXTOFL(CELL14; CELL14A),
   CELL 14A * CELL 15A → CELL 19A, CELL  20A * CELL  16
   *CELL 17 A → CELL 20,  CELL 19A/CELL 20
   → CELL 21, CELL 21 * EXPANSION [CELL 13] + CELL 22 → CELL 22,
   CELL 22 → DEESUMNUMBER}},..
```

5

(:PAGE F5061-12)

(COMMENTS: THIS FLOWCHART IS A SUBROUTINE FOR CALCULATING THE JOINING FACTORS WHICH CONNECT THE RADIAL FUNCTIONS OF THE SECOND TYPE WITH THE ANGLE FUNCTIONS OF THE SECOND TYPE.)

```
   EDD, EFF, CELL 60, CELL 61, CELL 62,
      CELL 30, CELL 31, CELL 32, CELL 33. CELL 34, CELL 35. CELL 36,
      CELL 37. CELL31A.
   CELL 38. CELL 39. CELL 40, CELL 41. CELL 42, CELL 42A.
   CELL 43. CELL 44, CELL 45. CELL 46, CELL 47. CELL 48. AITCHH.
   CELL 47A. CELL 49. CELL 50, CELL 47B. CELL 51. CELL 52. CELL 53. JOINFAC.
   CELL 55, CELL 56, CELL 57. CELL 47C. CELL 58. CELL 59.
```

```
CELL 102. CELL 101. CELL 41A.
CELL 44A. CELL 55A.
:
CONNECTION 3: CALCJOININGFACTO (EDD, EFF, AITCHH; JOINFAC), CONNECTION 4.

(:PAGE F5061-13)
CALCJOINING FACTO (EDD, EFF, AITCHH. JOINFAC.): {EDD - EFF → CELL 30,
   CELL 30 = 0: CALCEVENJOINING. CELL 30/2 → CELL 60, CELL 60*2 → CELL 61,
      CELL 61 - CELL 30 → CELL 62,

   CELL 62 = 0: CALCEVENJOINING. CALCODDJOINING.
CALCEVENJOINING: 1 → CELL 31, M = 1 (1) CELL 30 {2 * CELL 31 → CELL 31},
   EXTO FL (CELL 31; CELL 31A),
   2 * EFF → CELL 32, FACTORIAL (CELL 32; CELL 33), FACTORIAL
   (CELL 60; CELL 35), (EDD + EFF)/2 → CELL 36, FACTORIAL (CELL 36; CELL 37),
   CELL 31A * CELL 33 * CELL 35 * CELL 37 → CELL 39,
   EDD + EFF → CELL 40, FACTORIAL (CELL 40; CELL 41A),
FACTORIAL (CELL 30; CELL 42A),
   FLAMMSTRATTN (EDD, EFF; CELL 101),
   CELL 39 * CELL 101 * DEENEG [0] * CELL 41A/CELL 42A
  → CELL 43, 2 * EFF -1 → CELL 44, FACTORIAL (EFF; CELL 45),
   EFF -1 → CELL 46, 1.0 → CELL 47A,
EXTOFL (CELL 44; CELL44A),
```

← Subroutine for calculating Eq. (8a) or Eq. (8b)

← Start compute Eq. (8a)

UNCLASSIFIED

```
(:PAGE F5061-14)
CELL 46 = 0: NEXT STEP A.;

CELL 46 < 0:|CELL 46| → CELL 48,K = 1(1) CELL 48{AITCHH * CELL 47A → CELL 47A},

  1.0/CELL 47A → CELL 47A, NEXT STEP A. |CELL 46| → CELL 50,
  K = 1(1) CELL 50 {AITCHH * CELL 47A → CELL 47A};

NEXT STEP A: CELL 44A * CELL 45 * CELL 41A → CELL 51,
  CELL 51 * CELL 47A → CELL 53, CELL 43/CELL 53 → JOINFAC, END 2.
```

← Finish compute Eq. (8a)

```
CALCODDJOINING: 1→ CELL 31, M = 1 (1) CELL 30 {CELL 31 * 2 → CELL 31},
FXTOFL (CELL31; CELL31A),   2 *EFF → CELL 32, FACTORIAL (CELL 32; CELL 33),
  (EDD - EFF -1)/2
  → CELL 34, FACTORIAL (CELL 34; CELL 35), (EDD +EFF+1)/2
  → CELL 36, FACTORIAL (CELL 36; CELL 37), CELL 31A * CELL 33 * CELL 35
  * CELL 37 → CELL 38, (EDD + EFF)/2
  → CELL 40, FACTORIAL (CELL 40; CELL 41),
  FACTORIAL (CELL 30; CELL 42A), FX TO FL (CELL 42; CELL 42A),
```

← Start compute Eq. (8b)

```
(:PAGE F5061-15)
  FLAMMSTRATTN (EDD, EFF; CELL 102),
  CELL 39 * CELL 102 * DEENEG [0] * CELL 41A
  /CELL 42A → CELL 43, 2 * EFF -3 → CELL 44, 2 * EFF -1 → CELL 55,
```

```
   FACTORIAL (EFF; CELL 45), EDD + EFF +1 → CELL 56, FACTORIAL (CELL 56;
   CELL 57), EFF -2 → CELL 46, 1.0 → CELL 47,
FXTOFL (CELL44; CELL44A), FXTOFL (CELL 55; CELL 55A),
CELL 46 = 0: NEXT STEP B.;

CELL 46 < 0:| CELL 46 | → CELL 48,K = 1(1) CELL 48 {AITCHH * CELL 47 → CELL 47},
   1.0/CELL 47 → CELL 49, NEXT STEP B. | CELL 46 | → CELL 50,
   K = 1(1) CELL 50 {AITCHH * CELL 47 → CELL 47};

NEXT STEP B: CELL 44A * CELL 55A * CELL 45 * CELL 57 → CELL 58,
 CELL 58 * CELL 47 → CELL 53, CELL 43/CELL 53
→JOINFAC, END 2.                                          ← Finish compute Eq. (8b)

END 2:} ..

5
(:PAGE F5061-16)

(COMMENTS: THIS FLOWCHART CONSISTS OF FOUR SUBROUTINES, UN, VN, WN, AND
THE FACTORIAL.)

ESS, ETT, UN, MILONONE, MILON2, MILON 3, MILON 4, MILON 5, MILON 6,
MILON 5A, MILON 6A, MILON 7, MILON 8, MILON 9, MILON 9A,
MILON 10, MILON 11, AITCH, MILON 12, MILON 7A, MILON 8A,
```


UNCLASSIFIED

```
VN, MILON 13, MILON 14, MILON 14A, WN, MILON 15, MILON 16, MILON 17, MILON 15AA,
MILON 13A.

;

(:PAGE F5061-17)

YEWSUBENN (ESS, ETT, UN.): {ESS +2 * ETT → MILON ONE,
   MILONONE -1 → MILON 2, 2 * ESS +2 * ETT +1 → MILON 3,
   MILON 3 -2 → MILON 4, MILONONE * MILON 2 → MILON 5, MILON 3
   * MILON 4 → MILON 6, FX TO FL (MILON 5; MILON 5A), FX TO FL
   (MILON 6; MILON 6A), MILON 5A/MILON 6A → UN},

YEESUBENN (ESS, ETT, VN.) : {2* (ESS + ETT) *(ESS + ETT
   +1) -2 * ETT * ETT -1 → MILON 7, (2 * ESS +2 * ETT
   +3) * (2 * ESS +2 * ETT -1) → MILON 8, (ESS + ETT)
   * (ESS + ETT +1) → MILON 9, FX TO FL (MILON 9; MILON 9A),

   MILON 9A - EIGENVALUE → MILON 10, AITCH * AITCH → MILON 11,
   MILON 10/MILON 11 → MILON 12, FX TO FL (MILON 7; MILON 7A),
   FX TO FL (MILON 8; MILON 8A), MILON 7A/MILON 8A + MILON 12
   → VN},
```

← Subroutine for computing Eq. (1)

← Subroutine for computing Eq. (2)

```
(:PAGE F5061-18)

DOUBLEYEWSUBENN (ESS, ETT, WN.) :{

   (ESS +2 ) * (ESS +1) → MILON 13, (2 * ESS +2 * ETT +3)
   *(2 *ESS +2 * ETT +1) → MILON 14, FX TO FL (MILON 13;
   MILON 13A), FX TO FL (MILON 14; MILON 14A), MILON 13A/
   MILON 14A → WN},

(:PAGE F5061-19)

FACTORIAL (MILON 15AA, MILON 16.):{

FXTOFL (MILON 15AA; MILON 15),
MILON 15 = 0: 1.0 → MILON 16, END 8.;
MILON 15 = 1.0 : 1.0 → MILON 16, END 8.
                 1 .0 → MILON 17, PROCESS.
PROCESS: MILON 15 * MILON 17 → MILON 17, MILON 15 - 1.0 → MILON 15.

MILON 15 ≤ 1.0: MILON 17 → MILON 16, END 8. PROCESS.

END 8:} ..
```

← Subroutine for computing Eq. (3)

UNCLASSIFIED

Subroutine for calculating Assoc. Legendre Functions of Second Type

```
5
(:PAGE F5061-20)
   (COMMENTS: THIS PROGRAM IS ALGORITHIM 62 APPEARING IN THE ACM OF JULY 1961.
   FOR VALUES OF V=0 TO V=VMAX AND REAL X [IF RI=0] OR IMAGINARY X [RI≠0]
   IT PLACES A SET OF Q [UV] [X] IN THE ARRAY Q.)

U|, VMAX, X|. RI, V|, P|. B|. Q|1. Q|20. Q|6. Q|2. T|. TOT|. C0|, C|1, C|2, D|0.
D|1. D|2. H|0. H|1. S|. Q|4. Q|5. R|(50). Q|(50). C|3, C|4, C|5, D|3. D|4. D|5.
D|40. D|41. D|42. BOAT|. C|6. SAIL|. C|7. C|77.C|8. C|15.C|10. C|11.C|12. C|13.
 C|9. C|14. C|16,C|17,C|18,D|16. D|17.D|18. D|OT.D|ASH.DA|RT. Q3=1.000000*-35,

   OUTPUT (50).;

(:PAGE F5061-21)
QLEG (U, VMAX, X. RI,): {40→V, FX TO FL (V; P), FX TO FL (U; B),

VMAX>33: VMAX+7→V, SEQUENCE 1. SEQUENCE 1.

SEQUENCE 1:
   RI=0:
      BEGINREAL.
         BEGINIMAG.
BEGINREAL:
U=0:
```

```
      (X+1.0)/(X-1.0)→Q1, LN (Q1; Q20), 0.5*Q20→Q[0], BEGIN 3,
         BEGIN 2.

BEGIN 2: X*X-1.0→Q6, SQRT (Q6; Q2), -1.0/Q2→T, 0.0→TOT,
            T→Q[0],
            I=2(1) U {I-1→C0, 3*I→C1, I*I→C2, FX TO FL (C0; D0),

(:PAGE F5061-22)
FX TO FL (C1; D1), FX TO FL (C2; D2), (X+X) * D0 * T * Q [0]→H0,
   (D1-D2-2.0) * TOT→H1, H0+H1→S, Q[0]→TOT, S→Q[0]}
BEGIN 3:
X=1.0:
      Q3→Q[0], JOB.
      JOB.
JOB: X*X-1.0→Q4, SQRT (Q4; Q5), X-Q5→R[V+1],

      J=V(-1) 1 {J+U→C3, J+J+1→C4, U-J-1→C5, FX TO FL(C3; D3),
      FX TO FL (C4; D4),  FX TO FL (C5; D5), D4*X→D40,
      D5*R [J+1]→D42, D40+D42→D41, D3/D41→R[J]},

         THE.
BEGINIMAG:
U=0:
      STEP 1.
```


UNCLASSIFIED

```
(:PAGE F5061-23)

  STEP 2.

STEP 1:
X<0.5:
      ARCTAN (X; BOAT), BOAT-1.50707963→Q[0], STEP 3.
      1/X→C6, ARCTAN (C6; SAIL), -SAIL→C7, STEP 3.

STEP 2: X*X+1.0→C77, SQRT (C77; C8), 1.0/C8→T, 0.0→TOT, T→Q[0],

        K=2(1)u{X+X→C15, FX TO FL (K;C9), C9-1.0→C10, 3*C9→C11,

                C9*C9→C12, C11+C12-2.0→C13, C13*TOT→C14,
                C15*C10*T*Q[0]+C14→S, Q[0]→TOT, S→Q[0]},
                STEP 3.

STEP 3: X-C8→R[V+1],

        L=V(-1) 1  {L+U→C16, L-U+1→C17, L+L+1→C18,

(:PAGE F5061-24)

FX TO FL (C16; D16), FX TO FL (C17; D17), FX TO FL (C18; D18),
```

```
D17*R[L+1]→DOT, D18*X→DASH, DOT-DASH→DART, D16/DART→
R[L]},

   M=1(2) VMAX {-R[M]→R[M]}, THE.

THE: Q[0] → OUTPUT [0], N=1 (1) VMAX {Q[N-1] * R [N]→Q[N]→OUTPUT[N]}}, END.

E|ND:..

5
(:PAGE F5061-25)
(COMMENTS:  THIS FWCHART CALCULATES THE DERIVATIVES OF THE ASSOCIATED LEGENDRE
FUNCTIONS OF THE SECONKIND WITH NEGATIVE SUBSCRIPTS, ALL THE QMN WITH
POSITIVE SUBSCRIPTS, ALL THE DERIVATIVES OF QMN WITH POSITIVE SUBSCRIPTS,AND ALL
THE PMN PLUS THEIR DERIVATIVES.)

EVV, SAILONE, EXXI, SHIPONE (50).  SAIL 2.  SAIL 3.  SAIL 4.  SHIPTWO (50).
SAIL 5.  SHIPTHREE (50).  SHIPFOUR (50).  EVV 2,
SAIL 6.  SAIL 7.  SHIPFIVE (50).  SHIPSEVEN(50).  SAIL 8, SHIPEIGHT (50).

SAIL 9, SAIL 10, SAIL 11. SAIL 12. SAIL 13. SHIPNINE (50).  SHIPTEN (50).
SHIPELEVEN (50).  SAIL 14.  SAIL 15.  SHIPTWELVE (50).
;
```


UNCLASSIFIED

```
(:PAGE F5061-26)
   (COMMENTS:  IN THIS SECTION WE CALCULATE THE DERIVATIVES OF THE NEGATIVE Q,
SUPERSCRIPT M, SUBSCRIPT - M, - M + 1, ETC. BY USE OF EQ. 3.43)

CALCDERIVNEGQUE:
        K = 1 (1) EVV {FX TO FL (EVV; SAILONE), SAILONE * EXXI
        * SHIPONE [K] → SAIL 2, (EXXI - 1.0) * (EXXI + 1.0) → SAIL 3,
        SQRT (SAIL 3; SAIL 4), SAIL 4 * SHIPTWO [K+1] + SAIL 2 →
        SAIL 5, SAIL 5 / SAIL 3 → SHIP THREE [K]},

(COMMENTS:  IN THIS SECTION WE CALCULATE ALL Q SUPERSCRIPT M, SUBSCRIPT 0,
   1, 2, ETC AND ALL Q SUPERSCRIPT M + 1, SUBSCRIPT 0, 1, 2, ETC.)

CALCQUE: , QLEG (EVV, 40, EXXI, RI;), M = 0 (1) 50 {OUTPUT [M] → SHIPFOUR [M]},
          EVV + 1 → EVV 2
         , QLEG (EVV2, 40, EXXI, RI;), M = 0 (1) 50 {OUTPUT [M] → SHIPFIVE [M]},

(:PAGE F5061-27)
(COMMENTS:  IN THIS SECTION WE CALCULATE DERIVATIVE OF QUE SUPERSCRIPT 0,
   SUBSCRIPT 0, 1, 2, ETC. AND DERIVATIVE OF QUE SUPERSCRIPT M,
   SUBSCRIPT 0, 1, 2, ETC.)

FORM DERIVQUE: (EXXI + 1.0) * (EXXI -1.0) → SAIL 6, SQRT (SAIL 6; SAIL 7),
```

← Start compute Eq. (13) to form $Q^{m'}_{-m}$, $Q^{m'}_{-m+1}$, etc

← Use subroutine "Qleg" to form Q^m_0, Q^m_1, Q^m_2, $\cdots$, Q^m_{40}

← Use subroutine "Qleg" to form Q^{m+1}_0, Q^{m+1}_1, Q^{m+1}_2, Q^{m+1}_3, $\cdots$, Q^{m+1}_{40}

```
EVV = 0;
      PROCEDURE ONE.
      PROCEDURE 2.

PROCEDURE ONE:
              K = 0 (1) 40 {SHIPFIVE [K] / SAIL 7 → SHIPSEVEN [K]}.
              FORMASSOCONE.

PROCEDURE 2:  EVV - 1 → SAIL 8
                      , QLEG (SAIL 8, 40, EXXI, RI;),
                        M = 0 (1) 50 {OUTPUT [M] → SHIPEIGHT [M]},

              I = 0 (1) 40 { -EVV + I → L, 2 * EVV + L → SAIL 9,

(:PAGE F5061-28)
        L + 1 → SAIL 10, FX TO FL ( SAIL 9; SAIL 11), FX TO FL (SAIL 10;
        SAIL 12), SHIPEIGHT [I] * SAIL 11 * SAIL 12 / SAIL 7 → SAIL 13,

        -SAILONE * EXXI * SHIPFOUR [I] / SAIL 6 + SAIL 13 → SHIPNINE [I]},
        FORMASSOCONE.

(COMMENTS:  IN THIS SECTION WE FORM LEGENDRE SUPERSCRIPT M, SUBSCRIPT
0, 1, 2, ETC. AND LEGENDRE SUPERSCRIPT M PLUS 1, SUBSCRIPT 0, 1, 2, ETC. AND
DERIVATIVE OF LEGENDRE SUPERSCRIPT M, SUBSCRIPT 0, 1, 2, ETC.)
```

← Use Eq. (13) to form $Q_0^{0'}$, $Q_1^{0'}$, $Q_2^{0'}$, etc.

← Use "Qleg" to form Q_0^{m-1}, Q_1^{m-1}, ..., Q_{40}^{m-1}

← Start use Eq. (13) to form $Q_0^{m'}$, $Q_1^{m'}$, $Q_2^{m'}$, Q_3^{m}, etc.

← Finish use Eq. (13)

UNCLASSIFIED

```
FORMASSOCONE: N = 0 (1) 40 {, LEGENDREA (EVV, N, EXXI, RI; PUV),
                                PUV → SHIPTEN [N]},
              N = 0 (1) 40 {, LEGENDREA (EVV2, N, EXXI, RI; PUV),
                                PUV → SHIPELEVEN [N]},
              N = 0 (1) 40 {SAILONE * EXXI * SHIPTEN [N] → SAIL 14,

(:PAGE F 5061-29)

              SAIL 7 * SHIPELEVEN [N] + SAIL 14 → SAIL 15, SAIL 15/SAIL 6
              → SHIPTWELVE [N]},

END 7: REALIGNONE. ..

5
(:PAGE F5061-30)

(COMMENTS: THIS PROGRAM IS A NELIAC VERSION OF ALGORITHM 47 PUBLISHED IN THE ACM
APRIL 1961.  IT COMPUTES P[MN] [X] IF R = 0, OR P[MN] [IX] IF R ≠ 0.)

V|, U|, X|. RRR|, PUV. STORY|, P|. Z|. W|. GAMMA(84). TROUBLE|.
BB|, AA|, Y|. CC|, DD|, DENOM 1|. EE|, DENOM 2|. NUM 1|. PART 1|. FF,
JOEL|. FRANK|. FOOT|, HAND|, FINGER|, PALM|, NAIL|, Z1|.

;
```

← Use "Legendrea" to form P_0^m, P_1^m, P_2^m, ..., P_{40}^m

← Use "Legendrea" for form P_0^{m+1}, P_1^{m+1}

← Use "Legendrea" to form $P_0^{m'}$, $P_1^{m'}$, $P_2^{m'}$, ..., $P_{40}^{m'}$

```
(:PAGE F5061-31)

LEGENDREA (U, V, X. RRR, PUV.):{

V < U:
   0.0 → P, GATE. ;

V = 0:
   1.0 → P, GATE. ;

      1.0 → Z, Z → W,
V = U:
   MAIN.
      V - U → AA, X = 0: 0.0 → Z, MAIN. ;
         I = 1(1) AA {Z*X → Z};

MAIN: 1.0 → GAMMA [1], V + V + 1 → BB,

(:PAGE F5061-32)
J = 2(1) BB {GAMMA [J-1] * W → GAMMA [J], W+1.0 → W}, 1.0 → W,
X = 0: ; X*X→TROUBLE, W/TROUBLE→Y;

RRR = 0:
   -Y → Y, - W → W;;
```


UNCLASSIFIED

```
X = 0:
   CAR.
      BRAKE.

C|AR: (V-U)/2 → CC, V - U → DD,
CC + CC ≠ DD:
      0.0 → P, GATE. U+CC→FOOT, U+V→HAND,
         GAMMA [CC+1] * GAMMA [FOOT+1] → DENOM1, GAMMA [HAND+1]/DENOM 1
         → P, (V-U)/ 4 → STORY,
4 * STORY = DD: LAST. -P → P, LAST. LAST.
BRAKE: 3 → EE, 0.0 → P,

(:PAGE F5061-33)
K = 1(1) 12 {

(V-U+2)/2 < K:
      LAST.
         ;
         V-K→FINGER, V-K-K-U+EE→PALM, V+V-K-K→NAIL,


         GAMMA [K] * GAMMA [FINGER+2] * GAMMA [PALM] → DENOM 2,
         GAMMA [NAIL+3] * Z → NUM 1, NUM 1/DENOM 2 → PART 1,
         P + PART 1 → P, Z * Y → Z},
```

```
LAST: 1.0 → Z, L = 1(1) V {Z+Z→Z}, P/Z→P,

R≠0: V-V/4→FF,

FF>1:
    -P→P, FORTUNE. FORTUNE. FORTUNE.

F|ORTUNE:
U = 0:

(:PAGE F5061-34)
      GATE.
          U/2→EE, X*X+W→JOEL, JOEL*JOEL → FRANK, SQRT (FRANK;Z),
      Z→Z1;

EE+EE≠U:
      SQRT (Z1;Z), U→EE; ;
          M = 1(1) EE {Z*P→P},

GATE: P → PUV}
..
```


UNCLASSIFIED

```
5
(:PAGE F5051-35)
(COMMENTS: THIS FLOWCHART IS A SUBROUTINE FOR CALCULATING THE
   ASSOCIATED LEGENDRE FUNCTION OF THE SECOND KIND WITH NEGATIVE SUBSCRIPTS.
   THE ANSWER IS IN QUENEG.)

CLOVA ONE (50), CLOVA 2, CLOVA 3, CLOVA 4,

     CLOVA 5, CLOVA 6, CLOVA 7, CLOVA 8, CLOVA 5A, CLOVA 7A, CLOVA 8A,

     XI, CLOVA 10, CLOVA 11, CLOVA 12, CLOVA 13, CLOVA 14, CLOVA 15,

     CLOVA 11A, PORT ONE (50), PORT 2, PORT 4, PORT 5, PORT 6, PORT 7,
     TESTENN, FIRST CELL, LAST CELL, HH, PORT 8, PORT 9, PORT 10,

     PORT 11, PORT 12, PORT 13, PORT 14, PORT 15,QUENEG (50),
     EYY, EYYONE, EGG, EGGONE,
        ;
        CONNECTION 4: CALCNEGQUE (EYY, EXXI;),
           EGG +1 → EGGONE,

     M = 0 (1) 20 {QUENEG [M] → SHIPONE [M]},
     EYY +1 → EYYONE,
     CALCNEGQUE (EYYONE, EXXI;),
```

← Start compute Eqs. (9), (10), (11), and (12) to form Q^{m}_{-m}, Q^{m}_{-m+1}, Q^{m}_{-m+2}, etc., and Q^{m+1}_{-m-1}, Q^{m+1}_{-m}, Q^{m+1}_{-m+1}, etc.

```
     M = 0 (1) 20 {QUENEG [M] → SHIPTWO [M]},
CALCDERIVNEGQUE.

(:PAGE F5061-36)
CALCNEGQUE (EYY, XI.): {0 → CLOVAONE [0], XI * XI -1.0 →
          CLOVA 10, SQRT (CLOVA 10; CLOVA 15),
          J = 1(1) EYY {2 * (J-1) → CLOVA 2, FX TO FL (CLOVA 2; CLOVA 3),
          CLOVA 3 * CLOVA ONE [J-1] → CLOVA 4, FACTORIAL (CLOVA 2; CLOVA 5A),
          J-1 → CLOVA 6, FACTORIAL (CLOVA 6; CLOVA 7A), 1 → CLOVA 8, M = 1(1)
          CLOVA 6 {2 * CLOVA 8 → CLOVA 8},
               FX TO FL (CLOVA 8; CLOVA 8A),
          1.0 → CLOVA 11.

CLOVA 6 = 0:
     1.0 → CLOVA 11, NEXT MOVE.
     ;              K = 1(1) CLOVA 6 {CLOVA 10 * CLOVA 11 → CLOVA 11},
NEXT MOVE: SQRT (CLOVA 11; CLOVA 11A), CLOVA 5A * XI * CLOVA 11A
          → CLOVA 12, CLOVA 8A * CLOVA 7A → CLOVA 13, CLOVA 12
          / CLOVA 13 + CLOVA 4 → CLOVA 14,
          - CLOVA 14 / CLOVA 15 → CLOVA ONE [J]},
(:PAGE F5061-37)
          -1.0 / CLOVA 15 → PORTONE [1] → PORT 2,
        M = 1(1) EYY {2* (M+1) -3 → PORT 4, FX TO FL (PORT 4; PORT 5),
          PORT 5 * PORT 2 → PORT 6, XI * CLOVAONE [M] + PORT 6 →
```


UNCLASSIFIED

```
        PORT 7, PORT 7 * PORTONE [1] → PORTONE [M+1] → PORT 2},

            (1-EYY) * 2 → TESTENN, |TESTENN| → PORT 15, CLOVAONE [EYY]
        → LAST CELL → QUENEG [1], PORTONE [EYY] → QUENEG [2]
        → FIRST CELL, 2 → HH,

      L = EYY (-1)0 |

PORT 15 ⩽ EYY:
    TABULATE.
      ;
        2 * EYY + TESTENN -1 → PORT 8, FX TO FL (PORT 8; PORT 9),
        LAST CELL * PORT 9 → PORT 10, 2 * EYY + 2 * TESTENN -1
        → PORT 11, FX TO FL (PORT 11; PORT 12), PORT 12 * XI * FIRST CELL
        -PORT 10 → PORT 13,

(:PAGE F5061-38)

      FX TO FL (TEST ENN; PORT 14), HH + 1 → HH,
       PORT 13 / PORT 14 → QUENEG [HH], FIRST CELL → LAST CELL,
       QUENEG [HH] → FIRST CELL, TESTENN + 1 → TESTENN, |TESTENN| → PORT 15},

TABULATE:}, ..
```

```
5
(:PAGE F5061-40)
        EJJ, EJJONE, EJJJ, FORTYEJJ, W,
           RADSEC = 0 * 0,
        RADSECDERIV = 0 * 0,
        SAIL 16, SAIL 18. SAIL 17 (50). SAIL 19. SAIL 20, SAIL 21 (50).
        SAIL 22.

        ISLAND (50).
        BAY (50).

        SHORE (50).
        B,

        ;

(:PAGE F5061-41)
        (COMMENTS: THIS SECTION ALIGNS Qs, Ds, EXPANSION
        CONSTANTS, AND CALCULATES THE SUM OF THE INFINITE SERIES OF
        PRODUCTS OF ASSOCIATED LEGENDRE FUNCTIONS AND THE Ds.)

        REALIGNONE: K = 1(1) EJJ {SHIPONE [K] → ISLAND [K] },
           EJJ + 1 → EJJONE,
           K = 0(1)40 { K + EJJONE → B, SHIPFOUR [K] → ISLAND [B]},
```

This section computes Eq. (14)

UNCLASSIFIED

```
CALCFUNCTION: EJJ -1 → EJJJ,
        L = 0(1) EJJJ {DEENEG [L] → BAY [L+1]},
      L = 0(1)40 {EJJONE + L → W, EXPANSION [L] → BAY [W]},

TESTNUMBERTHREE = 0: - 1 → SAIL16; 0 → SAIL16;

FORMRADSEC: 0→ SAIL18, 20 + EJJ → FORTYEJJ,
      I = 1(1) FORTYEJJ {SAIL16 + 2 → SAIL16, BAY [I] * ISLAND [SAIL16]
        → SAIL17 [I], SAIL17 [I] + SAIL18 → SAIL18},

        0 → SAIL19 → SAIL20, EJJ + 1 → EJJONE,
        20 + FORTY EJJ → FORTY EJJ,

(:PAGE F5061-42)

      J = EJJONE (2) FORTYEJJ {SAIL20 + 1 → SAIL20, DEERHO [SAIL20]
          * SHIPTEN [J] → SAIL21 [SAIL20],
        SAIL21 [SAIL20] + SAIL19 → SAIL19},

(:PAGE F5061-43)
      SAIL18 + SAIL19 → SAIL22, SAIL22 / JOINFAC → RADSEC,

      RADSECDERIVSETUP:
          K = 1(1) EJJ {SHIPTHREE [K] → SHORE [K]},
```

Sail 18 = $\sum_{r=-2m,\,-2m+1}^{\infty 1} d_r(h|m\ell)\, Q^m_{m+r}(\xi)$

Sail 19 = $\sum_{r=2m+2,\,2m+1} d_{r|\rho}(h|m\ell)\, P^n_{r-m-1}(\xi)$

Rad sec = $R^{(2)}_{m\ell}$

Start compute $dR^{(2)}_{m\ell}/d\xi$

```
      EJJ + 1 → EJJONE,
EJJ = 0:   K = 0 (1)40 {K + EJJONE → B, SHIPSEVEN [K] →
         SHORE [B]};
      K = 0(1)40 {K + EJJONE → L, SHIPNINE [K] → SHORE [L]};

TESTNUMBERTHREE = 0: -1 → SAIL 16; 0 → SAIL 16;
   0 → SAIL 18, 20 + EJJ → FORTYEJJ,

   I = 1 (1) FORTYEJJ {SAIL16 + 2 → SAIL16,

      BAY [I] * SHORE [SAIL16] → SAIL17 [I],
      SAIL17 [I] + SAIL18 → SAIL18},
      0 → SAIL19 → SAIL20, EJJ + 1 → EJJONE,
      20 + FORTY EJJ → FORTY EJJ,

(:PAGE F5061-44)
            J = EJJONE (2) FORTYEJJ {SAIL20 + 1 → SAIL20,
                  DEERHO [SAIL20] * SHIPTWELVE [J] → SAIL21 [SAIL20],
                  SAIL21 [SAIL20] + SAIL19 → SAIL19},

                  (SAIL18 + SAIL19) / (JOINFAC) → RADSECDERIV,

                  FINISH:  SIGNADJUSTMENT. ..
```

← Finish compute $dR_m^{(2)}/d\xi$

UNCLASSIFIED

```
5
(:PAGE F5061-45)
;
SIGNADJUSTMENT:

RI = 0:  DELIVER ANSWERS.;

TEST DIGIT = 0:  SIDEEVEN.;
TEST NUMBER THREE = 0:  SIDEEVEN. SIDEODD.
SIDEEVEN:  EMM/2 → CELL 3, CELL3 * 2 → CELL 7,
   CELL 7 = CELL 3:  EMMEVEN. EMMODD.

EMMEVEN:  EMM/2 → CELL 3, POWERHEMM (CELL3; CELL22), ATTACH PROP SIGN.
EMMODD:  (EMM -1) / 2 → CELL 3, POWERHEMM ( CELL3; CELL22), ATTACH PROP SIGN.
SIDE ODD:  EMM/ 2 → CELL3, CELL3 * 2 → CELL 7,
CELL 7 = CELL 3:  EVENHEMM. EMMODD.
EVENHEMM:  EMM/ 2 → CELL 3, POWERHEMM (CELL3; CELL22), -1.0 * CELL 22 → CELL 22,
ATTACH PROP SIGN:  CELL 22 * RADSEC → RADSEC,
            CELL 22 * RADSECDERIV → RADSECDERIV,
            DELIVER ANSWERS.

  POWERHEMM (ENTRYONE, ENTRYTWO.): {1.0 → CELL 6, -1.0 → CELL 4,
     ENTRYONE = 0:  1.0 → ENTRYTWO, DEPART.;
              J = 1 (1) ENTRYONE {CELL 6 * CELL 4 → CELL 6},
```

This procedure makes adjustments for oblate case when m=0

```
          CELL 6 → ENTRYTWO,
DEPART:}..

5
(:PAGE F 5061-46)
;
   IO STATEMENT:  FL TO FX ([#0800]; Q),
                  FL TO FX ([#0801]; H),
                  [#0923] → CELL 2, [#0804] → R,
                  [#0805] → RUSH, [#0802] → HURRY,
                  CALL RMN (Q, H, CELL 2, RUSH, HURRY, R),

                  CALCNEGSUBSCRDEE.
   DELIVER ANSWERS: ,{<< THE | VALUE | OF | R2MN | IS | > | RADSEC>,,},

                  {<< THE | VALUE | OF | DERIV | R2MN | IS | > | RADSECDERIV>},
                  FINALLY.
   CALL RMN (MODALEMM, MODALENN, EIGENNUMBER. RADIALEXXI. ECCENTRICH.
          PROLAOBLA,): {MODALEMM → EMM → ECC → EFF → EVV
          → EYY → EJJ, MODALENN → EBB → ELL → EDD,
          EIGENNUMBER → EIGENVALUE, RADIALEXXI → EXX
          ECCENTRICH → AITCHH → AITCH, PROLAOBLA → RI}
FINALLY:..
5..
```

← Program begins here by operating on input data. There must exist in the memory the spheroidal wave function eigenvalue at location 0923, obtainable from Tape F5188. Other input requirements are

location	input
0800	m
0801	ℓ
0802	h
0805	ξ
0804	R

(See text for details on input requirements)

UNCLASSIFIED