NRL Report 6110

A Neliac Program for Calculating Spheroidal Wave Function Eigenvalues and Expansion Constants

S. HANISH

Transducer Branch
Sound Division

July 27, 1964

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

CONTENTS

A Neliac Program for Calculating Spheroidal Wave Function Eigenvalues and Expansion Constants

S. Hanish

Transducer Branch
Sound Division

A digital computer program, written in the Neliac dialect of Algol, for the computation of spheroidal wave function eigenvalues and expansion constants is presented in this report. A brief explanation of the program is made by listing all of the formulas used 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 shown has been compiled by the Neliac-N compiler on the NAREC and has worked successfully for the following ranges of input parameters: $0 \leq m \leq 10$, $0 \leq \ell \leq 20$, $0 \leq h \leq 20$, with a precision of at least six significant digits.

INTRODUCTION

In NRL Report 5837 (1), the author and Baier published a digital computer program for calculating spheroidal wave function eigenvalues and expansion constants. The program was written in NIP (NAREC Interpretive Program) language. After the appearance of that report a new language became available at the NAREC computing center for the rapid composition of computer programs. This was the Neliac dialect of Algol, implemented on the NAREC by the Neliac-N compiler. It was then thought advisable to rewrite the program of Report 5837 in this new language. The following report contains a complete transcription of this Neliac version together with explanatory notes. The advantage of having the new version is apparent when it is realized that no portion of this Neliac program (excluding the input procedure) is machine dependent.

THEORETIC CONSIDERATIONS

NRL Report 5837 contains a brief but complete summary of all the pertinent formulas and references needed for the computation of the spheroidal wave function eigenvalues and expansion constants. The reader is referred to the work of Flammer (2) for additional details.

In order to make the following program understandable 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 origin 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 five input parameters m, ℓ, h, Tally, and R. In all cases the results have agreed with the earlier outputs of the NIP program of Report 5837 with a precision of (at least) six significant digits.

PROCEDURE FOR USING NELIAC VERSION

The tape containing the Neliac version of our program is designated by the identifying number F5188. When the Neliac-N compiler is in the NAREC memory, this tape is simply read in by the automatic reader and the program is compiled. Five input parameters must be in the memory before computation can begin. These parameters are derived from the

*NRL Problem S02-07 and Project RF 001-03-44-4052. This is an interim report on one phase of the problem; work is continuing. Manuscript submitted April 9, 1964.

following general expansion of the angle spheroidal wave functions $S_{m\ell}$ in series of associated Legendre functions:

$$S_{m\ell}(h, \cos\vartheta) = \sum_{r=0,1}^{\infty}{}' d_r(h|m\ell)P_{m+r}^{m}(\cos\vartheta) .$$

Here, m and ℓ are integral modal numbers. The number h is $kd/2$, where $k = 2\pi/\lambda$, λ is the wavelength, and d is the interfocal distance of the spheroid. Since only a finite number of terms is required by the sum we designate the maximum desired number r by Tally. If $\ell - m$ is even, Tally is an even number. Otherwise it is odd. The last input parameter appears as the letter R, which is either a zero (prolate case) or a hexidecimal 1 (oblate case).

For any particular computer we must reserve five consecutive locations in the memory to contain the five input parameters. In the NAREC we have selected locations 800, 801, 802, 803, and 804. These assignments lead to the following input requirement table (Table 1).

TABLE 1
Input Locations and Requirements

NAREC Address	Input Requirement
0800	m, in Neliac floating point
0801	ℓ, in Neliac floating point
0802	h, in Neliac floating point
0803	Tally,* in NAREC hexidecimal
0804	R, either zero (prolate case) or hexidecimal 1 (oblate case)

*The maximum value of Tally is decimal 250 (even) or decimal 249 (odd).

When the input is in the memory the compiled program may be executed by starting at the command (appearing on page 2 of the flow chart) entitled "Calc Eigen Value." Previous compilation has shown this command to be in location 0948. Hence the NAREC starting order is LO 0948.

The program automatically prints out the computed eigenvalue and a list of all of the expansion constants called for by the number Tally. If Tally is an even number (≤250) the constants are labeled d_0, d_2, d_4, etc. If Tally is an odd number (≤249) the constants are d_1, d_3, d_5, etc. Previous compilation on the NAREC has shown that the expansion constants are located at ADDR. 1A00, No. 1A01, etc., and the eigenvalue at ADDR. 0923.

Appendix B is a complete reprint of the entire contents of Tape F5188. On most pages of this reprint we have appended at the right a series of brief explanatory remarks. By consulting Appendix A the reader may find the formula whose compilation is being effected by the series of commands indicated by the explanatory remarks.

REFERENCES

1. Hanish, S., and Baier, R.V., "A Digital Computer Program For Calculating Spheroidal Wave Function Eigenvalues and Expansion Constants," NRL Report 5837, Jan. 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.

Appendix A
EQUATIONS REFERRED TO IN APPENDIX B

Equation (1) – Ref. 3, p. 306

$$A_{m\ell(h)} = hq + m^2 - \frac{1}{8}(q^2+5) - \frac{q}{64h}(q^2+11-32m^2)$$

$$-\frac{1}{1024h^2}[5(q^4+26q^2+21) - 384m^2(q^2+1)]$$

$$-\frac{1}{h^3}\left[\frac{1}{128\cdot128}(33q^5+1594q^3+5621q) - \frac{m^2}{128}(37q^3+167q) + \frac{m^4}{8}q\right]$$

$$-\frac{1}{h^4}\left[\frac{1}{256\cdot256}(63q^6+4940q^4+43327q^2+22470)\right.$$

$$\left.-\frac{m^2}{512}(115q^4+1310q^2+735) + \frac{3m^4}{8}(q^2+1)\right]$$

$$-\frac{1}{h^5}\left[\frac{1}{1024\cdot1024}(527q^7+61529q^5+1043961q^3+2241599q)\right.$$

$$-\frac{m^2}{32\cdot1024}(5739q^5+127550q^3+298951q)$$

$$\left.+\frac{m^4}{512}(355q^3+1505q) - \frac{m^2}{16}q\right] + O\left(\frac{1}{h^6}\right) \cdot$$

Equation (2) – Ref. 3, p. 307

$$A_{m\ell(-ih)} = -h^2 + 2hp - \frac{1}{2}(p^2-m^2+1) - \frac{p}{8h}(p^2-m^2+1)$$

$$-\frac{1}{64h^2}[5p^4+10p^2+1-2m^2(3p^2+1)+m^4]$$

$$-\frac{p}{512h^3}[33p^4+114p^2+37-2m^2(23p^2+25)+13m^4]$$

$$-\frac{1}{1024h^4}[63p^6+340p^4+239p^2+14-10m^2(10p^4+23p^2+3)$$

$$+3m^4(13p^2+6)-2m^6]$$

$$-\frac{p}{8192h^5}[527p^6+4139p^4+5221p^2+1009-m^2(939p^4+3750p^2+1591)$$

$$+m^4(465p^2+635)-53m^6] + O\left(\frac{1}{h^6}\right) \cdot$$

Equation (3) — Ref. 4, p. 57

$$A_{m\ell}(h) = \ell(\ell+1) + h^2 \left[\frac{2\ell(\ell+1) - 2n^2 - 1}{(2\ell-1)(2\ell+3)} + t_{m\ell}(h)\right]$$

$t_{m\ell}(h) \rightarrow 0$, to start computation.

Equation (4) — Ref. 2, p. 17

$$\beta_r^m = \frac{r(r-1)(2n+r)(2n+r-1)\ h^4}{(2n+2r-1)^2(2n+2r-3)(2n+2r+1)},\ r \geqslant 2\,.$$

Equation (5) — Ref. 2, p. 17

$$\gamma_r^n = (n+r)(m+r+1) + \frac{1}{2} h^2 \left[1 - \frac{4n^2-1}{(2n+2r-1)(2n+2r+3)}\right],\ r \geqslant 0\,.$$

Equation (6a) — Ref. 2, p. 17

$$N_2^m = -\gamma_0^n + A_{m\ell}\,.$$

Equation (6b) — Ref. 2, p. 17

$$N_3^m = -\gamma_1^n + A_{m\ell}\,.$$

Equation (7) — Ref. 2, p. 17

$$N_{r+2}^m = \frac{-\beta_r^n}{N_r^m} - \gamma_r^n + A_{m\ell}\,.$$

Equation (8) — Ref. 2, p. 17

$$N_r^n = \frac{-\beta_r^n}{-A_{m\ell} + \gamma_r^n + N_{r+2}^n}\,.$$

Equation (9) — Ref. 2, p. 21

$$\text{Correction B} = \frac{(N_{\ell-m+2}^m)^2}{\beta_{\ell-m+2}^m} + \frac{(N_{\ell-m+2}^m)^2(N_{\ell-m+4}^m)^2}{\beta_{\ell-m+2}^m\,\beta_{\ell-m+4}^m} + \cdots\,.$$

Equation (10) — Ref. 2, p.21

$$\text{Correction A} = 1 + \frac{\beta_{\ell-m}^m}{(N_{\ell-m}^m)^2} + \frac{\beta_{\ell-m}^m\,\beta_{\ell-m-2}^m}{(N_{\ell-m}^m)^2(N_{\ell-m-2}^m)^2} + \cdots\,.$$

Equation (11) – Ref. 2, p. 21

$$\delta A_{m\ell} = \frac{\text{Eq. (8)} - \text{Eq. (7)}}{\text{Eq. (9)} + \text{Eq. (10)}}.$$

Equation (12) – Ref. 2, p. 17

$$\frac{d_r}{d_{r-2}} = \frac{(2m+2r-1)(2m+2r+1)}{(2m+r)(2m+r-1)h^2} N_r^m, \; r \geqslant 2.$$

Equation (13) – Ref. 1, p. 4

$$\frac{d_n}{d_x} = \left(\frac{d_n}{d_{n-2}} \frac{d_{n-2}}{d_{n-4}} \cdots \frac{d_{x+2}}{d_x}\right).$$

Equation (14a) – Ref. 1, p. 4

$$d_0 = \frac{(\ell+m)!}{(\ell-m)!\left[\sum_{n=2}^{\infty}{}' \frac{(n+2m)!}{n!}\left(\frac{d_n}{d_{n-2}} \frac{d_{n-2}}{d_{n-4}} \cdots \frac{d_2}{d_0}\right) + (2m)!\right]}.$$

Equation (14b) – Ref. 1, p. 4

$$d_1 = \frac{(\ell+m)!}{(\ell-m)!\left[\sum_{n=3}^{\infty}{}' \frac{(n+2m)!}{n!}\left(\frac{d_n}{d_{n-2}} \frac{d_{n-2}}{d_{n-4}} \cdots \frac{d_3}{d_1}\right) + (1+2m)!\right]}.$$

Appendix B
THE NELIAC PROGRAM

```
5
(:PAGE F5188-1)

COMPUTE SPHEROIDAL EIGENVALUE AND CONSTANTS. #0900. ..

5
  (COMMENTS: THIS FLOWCHART CALCULATES THE STARTING VALUE OF THE EIGENNUMBER
  USING MEIXNERS FORMULAS.)

     EMM = {#0800}. ELL = {#0801}. H = {#0802}. TALLY = {#0803}.
     R = {#0804}.
     Q. WORD1. WORD2. WORD3. WORD4. AA. P. START LAMBDA = 0*0.
                    CURRENT LAMBDA = 0*0.
       WORD102. WORD103. WORD104. WORD105. WORD106.
       WORD107. WORD122. WORD123. WORD124. WORD125.
       EMMSQUARE. EMMFOURTH. EMMSIXTH.
       WORD130. WORD131. WORD132. WORD133. WORD134.
       WORD135. WORD136. WORD137A. WORD137B.
       WORD138. WORD139A. WORD139B. WORD139C.
       WORD139. WORD140A. WORD140B. WORD140C.
       WORD140. WORD141A. WORD141B. WORD141C.
       WORD141D. WORD141.
```

This program computes the eigenvalue $A_{m\ell}$ and the constants $d_r(h|m\ell)$, for prolate or oblate case. There are five input parameters, m, ℓ, h, Tally, and R. Set R = 0 for prolate, R = 1 for oblate.

Five consecutive addresses in memory are assigned to m, ℓ, h, r_{max}, and R. Tally = r_{max}.

Program	Comment
(:PAGE F5188-2)	
CALC EIGEN VALUE:	
R = 0: (ELL-EMM) * 2.0 + 1.0 → Q, |H*Q| → WORD1, CALCSTARTERS.	← WORD 1 contains first term of Eq. (1) (prolate) or Eq. (2) (oblate) of Appendix A
(ELL-EMM) * 2.0 + EMM + 1.0 → Q, |H * H| → WORD1, CALCSTARTERS.	
CALCSTARTERS:	
Q * Q → WORD102, WORD102 * Q → WORD103, WORD103 * Q → WORD104,	
WORD104 * Q → WORD105, WORD105 * Q → WORD106, WORD106 * Q → WORD107,	
1.0/H/H → WORD122, WORD122/H → WORD123, WORD123/H → WORD124,	
WORD124/H → WORD125, EMM * EMM → EMMSQUARE, EMMSQUARE * EMMSQUARE → EMMFOURTH,	
EMMFOURTH * EMMSQUARE → EMMSIXTH, R = 0: CALCPROLATE CASE. CALCOBLATE CASE.	
CALCPROLATE CASE:	
(527.0 * WORD107 + 61529.0 *WORD 105 + 1043961.0*WORD103+ 2241599.0*Q)	← Compute last term of Eq. (1) or Eq. (2) and set in WORD 3
/1024.0/ 1024.0 → WORD130,	
(5739.0 * WORD105 + 127550.0 * WORD103 + 298951.0 * Q) * EMMSQUARE	
/32.0 / 1024.0 → WORD131,	
(355.0 * WORD103 + 1505.0 * Q) * EMMFOURTH/ 512.0 → WORD132,	
EMMSQUARE * Q/ 16.0 → WORD133,	
(WORD130 - WORD131 + WORD132 - WORD133) * WORD125 → WORD2,	
|WORD2| → WORD3, WORD1 / 200.0 → WORD4,	
WORD4 > WORD3: H*H → AA, PROLASYMP LAMBDA. H * H → AA,	← If 1/200 of first term is greater than last term of Eq. (1) or Eq. (2), then use Eq. (1) or Eq. (2) for computing starting value of $A_{m\ell}$. Otherwise use Eq. (3).
SIMPLE START LAMB.	

```
(:PAGE F5188-3)
CALCOBLATE CASE:  527.0 * WORD106 + 4139.0 * WORD104 + 5221.0
   * WORD102 + 1009.0 → WORD130, EMMSQUARE * (939.0 * WORD104
+ 3750.0 * WORD102 + 1591.0 ) → WORD131, EMMFOURTH * (465.0 * WORD102
   + 635.0) → WORD132, 53.0 * EMMSIXTH → WORD133,

   (WORD130 - WORD131 + WORD132 - WORD133) * Q * WORD125 / 8192.0
   → WORD2,
       |WORD2| → WORD3, WORD1 / 200.0 → WORD4,
   WORD4 > WORD3: -H*H → AA,  OBLASYMP LAMBDA. - H * H → AA,
                SIMPLE START LAMB.
..
5
(:PAGE F5188-4)

(COMMENTS: THIS SECTION CALCULATES THE ASYMPTOTIC EIGENVALUE
   FOR THE PROLATE CASE USING MEIXNERS FORMULA.);
PROLASYMP LAMBDA:
H * Q → WORD134, (WORD102 + 5.0) / 8.0 → WORD 135, (WORD102 + 11.0
- 32.0 * EMMSQUARE) *Q / 64.0 / H → WORD136, (WORD 104 + 26.0
* WORD 102 + 21.0) * 5.0 → WORD137A, (WORD102 + 1.0) * EMMSQUARE
* 384.0 → WORD137B, (WORD137A - WORD137B) / 1024.0 * WORD122
→ WORD138, (33.0 * WORD105 + 1594.0 * WORD103 + 5621.0 * Q)
/ 128.0 / 128.0 → WORD139A, EMMSQUARE * (WORD103 * 37.0 +
```

← Computation of Eq. (1)

```
167.0 * Q) / 128.0 → WORD139B, EMMFOURTH * Q/ 8.0 → WORD139C,
(WORD139A - WORD139B + WORD139C) * WORD123 → WORD139.

(63.0 * WORD106 + 4940.0 * WORD104 + 43327.0 * WORD102
+ 22470.0 ) / 256.0 / 256.0 → WORD140A,
(115.0 * WORD104 + 1310.0 * WORD102 + 735.0) * EMMSQUARE / 512.0
→ WORD140B, 3.0 * EMMFOURTH * (WORD102 + 1.0) / 8.0 → WORD140C,
(WORD140A - WORD 140B + WORD 140C) * WORD 124 → WORD140,

(:PAGE F5188-5)
WORD 134 + EMMSQUARE - WORD 135 - WORD 136 - WORD 138
- WORD 139 - WORD 140 - WORD 2 → START LAMBDA → CURRENT
LAMBDA,

          {<| START LAMBDA>,}, {<>,,,},

    ODD EVEN TEST TWO.

..
5
(:PAGE F5188-6)
(COMMENTS:  THIS SECTION CALCULATES THE ASYMPTOTIC EIGENVALUE
 FOR THE OBLATE CASE USING MEIXNERS FORMULAS.);
OBLASYMPLAMBDA: - H * H → WORD 134, 2.0 * H * Q → WORD 135,    ← Computation of Eq. (2)
   (WORD 102 - EMMSQUARE + 1.0) / 2.0 → WORD 136, (WORD 102 - EMMSQUARE
```

```
    + 1.0) * Q / 8.0 / H → WORD 138, 5.0 * WORD 104 + 10.0 * WORD 102
    + 1.0 → WORD 139A, 2.0 * EMMSQUARE * (3.0 * WORD 102 + 1.0) →
    WORD 139B, (WORD 139A - WORD 139B + EMMFOURTH) / 64.0
    * WORD 122 → WORD 139,

    33.0 * WORD 104 + 114.0 * WORD 102 + 37.0 → WORD 140A,
    2.0 * EMMSQUARE * (23.0 * WORD 102 + 25.0) → WORD 140B,
    13.0 * EMMFOURTH → WORD 140C,

    (WORD 140A - WORD 140B + WORD 140C) * Q/512.0 *WORD 123
    → WORD 140,

    63.0 * WORD 106 + 340.0 * WORD 104 + 239.0 * WORD 102 + 14.0
    → WORD 141A,

(:PAGE F 5188-7)
10.0 * EMMSQUARE * (10.0 * WORD 104 + 23.0 * WORD 102 + 3.0) → WORD 141B,
3.0 * EMMFOURTH * (13.0 * WORD 102 + 6.0) → WORD 141C,
2.0 * EMMSIXTH → WORD 141D,

(WORD 141A - WORD 141B + WORD 141C - WORD 141D) / 1024.0 * WORD 124
→ WORD 141,
```

```
WORD 134 + WORD 135 - WORD 136 - WORD 138 - WORD 139
- WORD 140 - WORD 141 - WORD 2 → START LAMBDA → CURRENT
LAMBDA,

          {<|START LAMBDA>,}, {<>,,,},

    ODD EVEN TEST TWO.

SIMPLE START LAMB:  ELL * (ELL+1.0) + AA * (2.0 * ELL * (ELL + 1.0)
       -2.0 * EMM * EMM - 1.0) / (2.0 *ELL - 1.0) / (2.0 * ELL + 3.0)

       + 1.000005 → START LAMBDA → CURRENT LAMBDA,

          {<|START LAMBDA>,}, {<>,,,},

       ODD EVEN TEST TWO.
..
5
(:PAGE F5188-8)
(COMMENTS: THIS FLOWCHART CALCULATES ALL BETAS AND GAMMAS NEEDED
FOR THE PROBLEM.)

    ELLIO, EMMIO, WORD5, CRITICLIMIT, WORD6, WORD7, STARTARR.
    D1, STARTCOUNT, COL, LIMITONE, D2, D3, WW, EYE.
```

← Computation of Eq. (3) (beside the SIMPLE START LAMB line)

← This flow chart computes Eq. (4) and Eq. (5) for all required values of r. (beside the COMMENTS line)

```
    BETA ANS. JAY. GIMMELANS. FIVECOUNT,
    BETA LIST (300) = 0*0, GIMMELLIST (300) = 0*0,
          :
 (:PAGE F5188-9)
 ODD EVEN TEST TWO: FL TO FX (ELL; ELLIO), FL TO FX (EMM; EMMIO),      ←If ℓ-m is even (odd) computation is even (odd).
            ELLIO - EMMIO → WORD5, WORD5 + 2 → CRITICLIMIT,
            WORD5 / 2 → WORD6, 2 * WORD6 → WORD7,
 WORD5 = WORD7: 0 → STARTARR → D1, 2 → STARTCOUNT, 2 * EMMIO → COL,
    CALCBETHS. 1.0 → STARTARR → D1,3 → STARTCOUNT, 2 * EMMIO + 1 → COL, CALCBETHS.

 CALCBETHS: TALLY + 50 → LIMITONE, FL TO FX (D1; D2),                 ←Compute Eq. (4) for all values of r and set answer in table "Betalist."
               D2 + 2 → D3, 0 → WW,

    I = D3 (2) LIMITONE {FX TO FL (I;EYE),

         BETH (EMM, EYE, AA; BETAANS), 1 + WW → WW,

       BETAANS → BETALIST [WW]},

         0 → WW,

 (:PAGE F5188-10)
    J = D2 (2) LIMITONE {FX TO FL (J; JAY),                            ←Compute Eq. (5) and store in table "Gimmellist."
```

```
        GIMMEL (EMM, JAY, AA; GIMMELANS), 1 + WW → WW,

        GIMMELANS → GIMMELLIST [WW]},

        0 → FIVECOUNT, BEGINTASK.

(COMMENTS: THIS FLOWCHART CONTAINS THE SUBROUTINES FOR CALCULATING GIMMEL, BETH.)

GIMMEL (MEM, RESH, CEESQUARE, GEE.): { (MEM + RESH) * (MEM + RESH + 1.0)

      + (1.0 - (4.0 * MEM * MEM - 1.0) / (2.0 * MEM + 2.0 * RESH - 1.0)
        / (2.0 * MEM + 2.0 * RESH + 3.0)) * CEESQUARE / 2.0 → GEE},


BETH (MEM, RESH, CEESQUARE, BEE.): {RESH * (RESH-1.0) * (2.0 * MEM + RESH)
          * (2.0*MEM+RESH-1.0) * CEESQUARE * CEESQUARE

          / (2.0 * MEM + 2.0 * RESH - 1.0) / (2.0 * MEM + 2.0 * RESH -1.0)

          / (2.0 * MEM + 2.0 * RESH - 3.0) / (2.0 * MEM + 2.0 * RESH + 1.0)

          → BEE}, ..
```

```
5
(:PAGE F5188-11)
    (COMMENTS: THIS FLOWCHART CALCULATES NMR AND BOTH EIGENVALUE CORRECTIONS.)

    BETALIMIT, GAMMALIMIT, UPPER COUNT, ELLEMMFOUR, RIO,

    ENNSUBARR (300). WORD15, WORD16, WORD18. CORRECTION B.
    DESCEND CORRECTS (300). DC. CORRECTSRIGHT (300). DE. CORRECTION A.
    WORD17. RISING CORRECTS (300). RC. CORRECTSLEFT (300).
    RE. TYPEWORD = 0*0, WORD 16H, ;

BEGINTASK:

(:PAGE F5188-12)
WORD5 = WORD7: GO EVEN. GO ODD.

GO EVEN: (TALLY+50) / 2 → BETALIMIT, BETALIMIT + 1 → GAMMALIMIT,
            CRITICLIMIT / 2 - 1 → UPPERCOUNT, CRITICLIMIT + 2 →
            ELLEMMFOUR, WORD6 + 1 → RIO, ELLEMMFOUR  / 2 → WORD 16H,
            CALCENNS.

GO ODD: (TALLY + 49) / 2 → BETALIMIT, BETALIMIT + 1 → GAMMALIMIT,
            (CRITICLIMIT - 1) / 2 - 1 → UPPERCOUNT, CRITICLIMIT + 2 → ELLEMMFOUR,
            (WORD5 + 1 ) / 2 → RIO, (ELLEMMFOUR -1) / 2 → WORD 16H,
```

```
(:PAGE F5188-13)

CALCENNS:
        - GIMMELLIST [1] + CURRENT LAMBDA → ENNSUBARR [1],          ← Compute Eq. (6a) or (6b)

        I = 1(1) UPPERCOUNT {- BETALIST[I] / ENNSUBARR[I]           ← Compute Eq. (7)

        - GIMMELLIST [I+1] + CURRENT LAMBDA → ENNSUBARR [I+1]},

        - BETALIST [BETALIMIT] / (GIMMELLIST [GAMMALIMIT] - CURRENT LAMBDA)

        → ENNSUBARR [BETALIMIT], BETALIMIT - 1 → WORD15,
        GAMMALIMIT - 1 → WORD16,

    I = WORD15 (-1) WORD 16H  { - BETALIST [I] / (GIMMELLIST [I+1]   ← Compute Eq. (8)

        - CURRENT LAMBDA + ENNSUBARR [I+1] ) → ENNSUBARR [I]},

        1.0 → WORD18, 0 → CORRECTION B,

(:PAGE F5188-14)
I = RIO (1) BETALIMIT {ENNSUBARR [I] * ENNSUBARR [I] / BETALIST [I]   ← Compute Eq. (9)

      → DESCEND CORRECTS [I] → DC, DC * WORD 18 → WORD 18 →
```

```
        CORRECTSRIGHT [I] → DE, DE + CORRECTION B → CORRECTION B }

        1.0 → CORRECTION A, 1.0 → WORD 17,

I = 1(1) UPPERCOUNT {BETALIST [I] / ENNSUBARR [I] / ENNSUBARR [I]          ← Compute Eq. (10)

      → RISING CORRECTS [I] → RC, RC * WORD17 → WORD17 → CORRECTSLEFT [I]

        → RE, RE + CORRECTION A → CORRECTION A },

COMPFALLRISEENNS. ..

5
(:PAGE F5188-15)
(COMMENTS: THIS FLOWCHART CALCULATES CURRENT LAMBDA AND FINAL LAMBDA.)

    ENNSUBARRCALC. ENNDIFFERENCE. DELTALAMBDA.
                  :

COMPFALLRISEENNS: - BETALIST [RIO] / (GIMMELLIST [RIO+1] - CURRENT LAMBDA  ← Compute Eq. (8) for r = ℓ - m + 2 and
                                                                             store in "Ennsubarrcalc"
      + ENNSUBARR [RIO+1] ) → ENNSUBARRCALC,
                                                                             Subtract Eq. (8) for r = ℓ - m + 2 from
        ENNSUBARRCALC - ENNSUBARR [RIO] → ENNDIFFERENCE,                   ← Eq. (7) of text for r = ℓ - m to form
                                                                             "Enndifference"
```

```
            ENN DIFFERENCE / (CORRECTION A + CORRECTION B) → DELTA LAMBDA,     ← Compute Eq. (11) of text
            DELTA LAMBDA + CURRENT LAMBDA → CURRENT LAMBDA,
            FIVE COUNT + 1 → FIVE COUNT,                                       ← Make 5 successive corrections to "Current Lambda"

FIVE COUNT > 5 : TYPEEIGEN. CALCENNS.

TYPEEIGEN: , {<<|EIGENVALUE|IS|>|CURRENT LAMBDA>,},                            ← The address of eigenvalue is "Current Lambda"

            {<>,,,}, MAKEDEEZERORATI.
    ..
5
(;PAGE F5188-16)
  (COMMENTS: THIS FLOWCHART CALCULATES THE RATIO OF DEE ENN TO DEE ZERO, THE
NORMALIZATION SUM AND ALL THE DEES.)
        ARR. DEE ENN DEE NAUGHT (300) = 0*0, WORD20.

        DEE ENNS (300) = 0*0, WORD21, WORD28. WORD25, WORD26.

        WORD27. NORMALLIZT (300) = 0*0, RC3. WORD29, WORD30.

     WORD31. WORD32, WORD33. WORD34, WORD35. DEESTARTER = 0*0,

     DEEZLIST (300) = 0*0, WORD 221. WORD 24.
      :
```

```
(:PAGE F5188-17)

MAKEDEEZERORATI: WORD 5 = WORD 7 : 2.0 → P; 3.0 → P;

        I = 1(1) BETALIMIT {FX TO FL (I: EYE), P + 2.0 *(EYE-1.0) → ARR,      ← Compute Eq. (12)

            DEEENNDEEZERORATI (ENNSUBARR [I], EMM, ARR, AA;

            DEEENNDEENAUGHT [I])},

MAKEDEEENNS: 1.0 → WORD20,

        I = 1 (1) BETALIMIT {DEEENNDEENAUGHT [I] * WORD20 → DEEENNS [I]       ← Compute Eq. (13)

                → WORD20},

MAKESTARTDEE: 2 * BETALIMIT → WORD21, 0 → WW → WORD28,

(:PAGE F5188 -18)

      J = STARTCOUNT (2) WORD21 {2 * EMMIO + J → WORD25,                      ← Compute Eq. (14a) or (14b)

            FACTORIAL (WORD25; WORD26), FACTORIAL (J; WORD27),
```

```
        1 + WW → WW,

        WORD26 * DEEENNS [WW] / WORD27 → NORMALLIZT [WW] → RC3,

        RC3 + WORD28 → WORD28},

            COL → WORD29, FACTORIAL (WORD29: WORD30),

        WORD28 + WORD30 → WORD31,

        ELLIO + EMMIO → WORD32, FACTORIAL (WORD32; WORD33),
        ELLIO - EMMIO → WORD34, FACTORIAL (WORD34;WORD35),

        WORD33 / WORD35 / WORD31 → DEESTARTER,

(:PAGE F5188-19)
MAKEALLDEEZ: I = 1(1) BETALIMIT {DEEENNS [I] * DEESTARTER → DEEZLIST [I+1]},
        DEESTARTER → DEEZLIST [1],

TABULATE RESULTS:

       J = 1(1) BETALIMIT {,{<|DEEZLIST [J]>,}},

       TERMINATE.
```

← Compute all d_r by multiplying Eq. (13) by Eq. (14a) or (14b). Set answers in "Deezlist."

```
(:PAGE F5188-20)

DEEENNDEEZERORATI (ENNLIST. MEM. RESH. CEESQUARE. DEEFRACTION) : {

        ENNLIST * (2.0 * MEM + 2.0 * RESH -1.0) * (2.0 * MEM + 2.0 * RESH
        + 1.0) / (2.0 * MEM + RESH) / (2.0 * MEM + RESH - 1.0) / CEESQUARE

        → DEEFRACTION},

FACTORIAL (WORD22, WORD23.): {FX TO FL (WORD22; WORD221),

WORD221 = 0 ∪ WORD221 = 1.0 : 1.0 → WORD23, END80, 1.0 → WORD24,
               PROCESSEIGEN,
PROCESSEIGEN: WORD221  * WORD 24 → WORD24, WORD221 - 1.0 → WORD221,
WORD 221 ≤ 1.0 : WORD 24 → WORD23, END80, PROCESSEIGEN.

END 80: }, TERMINATE: ..
5..
```