Computer Physics Communications 30 (1983) 177–185
North-Holland Publishing Company

PROLATE RADIAL SPHEROIDAL WAVE FUNCTIONS

T.A. BEU * and R.I. CÂMPEANU

University Babeş-Bolyai, Department of Physics, 3400 Cluj-Napoca, Romania

Received 24 November 1982; in revised form 14 April 1983

PROGRAM SUMMARY

Title of program: PRSWFN

Catalog number: ACEY

Program available from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Computer: FELIX C-256; *Installation*: University Computer Centre, Cluj-Napoca, Romania

Operating system: SIRIS 2/3

Programming language used: FORTRAN IV

High speed storage required: 10 Kwords

Number of bits in a word: 32

Peripherals used: card reader, line printer

Number of cards in combined program and test deck: 570

Card punching code: EBCDIC

Keywords: general purpose, two-centre frame, radial spheroidal wave functions, expansion, Bessel functions

Nature of physical problem
The prolate radial spheroidal wave functions appear in a wide range of physical applications, and in particular in two-centre systems. The package PRSWFN contains six subprograms which compute these functions of both the first and second kind for any argument and accuracy.

Method of solution
The prolate radial spheroidal wave functions $R_{ml}^{(1,2)}(c, \xi)$ are calculated by summing spherical Bessel series, the coefficients being calculated separately, once for all the values of ξ; first the ratios of two consecutive coefficients are calculated recursively and then the final values of the coefficients are obtained through a normalization procedure [1].

Restrictions on the complexity of the problem
The package can be in principle applied to calculate any $R_{ml}^{(1,2)}(c, \xi)$. However, for high accuracy calculations of $R_{ml}^{(2)}(c, \xi)$ with ξ close to unity and large m, the number of terms to be retained in the spherical Bessel series becomes very large, imposing large storage space and considerable computing time.

Typical running time
For $\xi = 1.077$, $c = 0.1$, ERRB $= 10^{-4}$ the number of terms in the $R_{ml}^{(2)}(c, \xi)$ series is 52. The run of this case took 7.7 s, from which about 90% of the time was dedicated to the calculation of the coefficients. The test run took about 9 min.

Reference
[1] C. Flammer, Spheroidal wave functions (Stanford Univ. Press, Stanford, 1957) (Russian translation, 1962).

* Now at University of Bucharest, Department of Physics, Bucharest-Măgurele, Romania.

LONG WRITE-UP

1. Introduction

The prolate radial spheroidal functions are basic to the solution of the free wave scattering in a two-centre frame of reference. The natural coordinates in this symmetry are $\xi = (r_1 + r_2)/d$, $\eta = (r_1 - r_2)/d$ and φ, where r_1 and r_2 are the distances from the current point to the centres separated by d and φ is the rotation angle about the axis of centres. If one assumes ξ, η and the wave number k real and writes the solution to the wave equation $(\nabla^2 + k^2)\psi = 0$ as $\psi_{ml} = S_{ml}(c, \eta) R_{ml}(c, \xi) \, {}^{\cos}_{\sin}(m\varphi)$ one obtains the radial equation:

$$\frac{\mathrm{d}}{\mathrm{d}\xi}\left[(\xi^2 - 1)\frac{\mathrm{d}}{\mathrm{d}\xi} R_{ml}(c, \xi)\right] - \left[\lambda_{ml} - c^2\xi^2 + \frac{m^2}{\xi^2 - 1}\right] R_{ml}(c, \xi) = 0, \tag{1}$$

where $c = kd/2$ and λ_{ml} are the separation constants (or eigenvalues) determined so that $R_{ml}(c, \xi)$ are finite at $\xi = \pm 1$. The solution of (1) is the prolate radial spheroidal function, which is often encountered in scattering problems describing the long-range part of the system wave function. An example taken from atomic physics is the scattering of electrons and positrons by diatomic molecules.

2. The calculation of $R_{ml}^{(1,2)}(c, \xi)$

The solution of eq. (1) can be written [1] as a spherical Bessel series:

$$R_{ml}^{(p)}(c, \xi) = \left[\left(\frac{\xi^2 - 1}{\xi^2}\right)^{m/2} \Big/ \sum_{r=0,1}^{\infty}{}' \frac{(2m+r)!}{r!} d_r^{ml}(c)\right] \sum_{r=0,1}^{\infty}{}' i^{r+m-l} \frac{(2m+r)!}{r!} d_r^{ml}(c) Z_{m+r}^{(p)}(c, \xi), \tag{2}$$

where

$$Z_{m+r}^{(p)} = \begin{cases} j_{m+r}(c, \xi) & \text{for } p = 1, \\ y_{m+r}(c, \xi) & \text{for } p = 2, \end{cases}$$

with j_n and y_n the Bessel spherical functions of the first and second kind. The summations in (2) imply only terms with even r when $l - m$ is even and only terms with odd r for old $l - m$.

The coefficients $d_r^{ml}(c)$ of the Bessel series (2) can be calculated from the recurrence relation (3.1.4) in ref. [1] or as ratios of the form:

$$\frac{d_r^{ml}}{d_{r-2}^{ml}} = \frac{(2m + 2r - 1)(2m + 2r + 1)}{(2m + r)(2m + r - 1)c^2} N_r^m \quad (r \geq 2), \tag{3}$$

where the factor N_r^m can be calculated from:

$$\left.\begin{aligned} N_{r+2}^m &= \gamma_r^m - \lambda_{ml} - \beta_r^m / N_r^m \\ N_2^m &= \gamma_0^m - \lambda_{ml}, \quad N_3^m = \gamma_1^m - \lambda_{ml} \end{aligned}\right\} \quad \text{for } r \leq l - m, \tag{4, 5}$$

$$N_r^m = \beta_r^m (\gamma_r^m - \lambda_{ml} - N_{r+2}^m)^{-1} \qquad \text{for } r > l - m, \tag{6}$$

where γ_r^m and β_r^m are expressed as:

$$\gamma_r^m = (m+r)(m+r+1) + \tfrac{1}{2}c^2\left[1 - (4m^2-1)/(2m+2r-1)(2m+2r+3)\right] \quad (r \geqslant 0), \tag{7}$$

$$\beta_r^m = r(r-1)(2m+r)(2m+r-1)c^4/(2m+2r-1)^2(2m+2r-3)(2m+2r+1) \quad (r \geqslant 2). \tag{8}$$

The eqs. (4) and (6) can also be expressed as continued fractions:

$$N_{r+2}^m = \gamma_r^m - \lambda_{ml} - \frac{\beta_r^m}{\gamma_{r-2}^m - \lambda_{ml} -}\,\frac{\beta_{r-2}^m}{\gamma_{r-4}^m - \lambda_{ml} -}\cdots \quad (r \leqslant l-m), \tag{9}$$

$$N_{r+2}^m = \frac{\beta_{r+2}^m}{\gamma_{r+2}^m - \lambda_{ml} -}\,\frac{\beta_{r+4}^m}{\gamma_{r+4}^m - \lambda_{ml} -}\,\frac{\beta_{r+6}^m}{\gamma_{r+6}^m - \lambda_{ml} -}\cdots \quad (r > l-m). \tag{10}$$

The eigenvalue λ_{ml} is the solution of the transcendental equation

$$U(\lambda_{ml}) = U_1(\lambda_{ml}) + U_2(\lambda_{ml}) = 0, \tag{11}$$

where

$$U_1(\lambda_{ml}) = \lambda_{l-m}^m - \lambda_{ml} - \frac{\beta_{l-m}^m}{\gamma_{l-m-2}^m - \lambda_{ml} -}\,\frac{\beta_{l-m-2}^m}{\gamma_{l-m-4}^m - \lambda_{ml} -}\cdots \tag{12}$$

and

$$U_2(\lambda_{ml}) = -\frac{\beta_{l-m+2}^m}{\gamma_{l-m+2}^m - \lambda_{ml} -}\,\frac{\beta_{l-m+4}^m}{\gamma_{l-m+4}^m - \lambda_{ml} -}\cdots \tag{13}$$

It can be calculated as a c^2 powers series

$$\lambda_{ml}(c) = \sum_k L_{2k}^{ml} c^{2k}, \tag{14}$$

where the coefficients have a rather complicated form (eqs. (3.1.18)–(3.1.23) in ref. [1]). The value of λ_{ml} obtained from (11) or from (14) can be further refined by taking into account the correction

$$\delta\lambda_{ml} = \left[U_1(\lambda_{ml}^{(1)}) + U_2(\lambda_{ml}^{(2)})\right] \Bigg/ \left[1 + \frac{\beta_{l-m}^m}{(N_{l-m}^m)^2} + \frac{\beta_{l-m}^m \beta_{l-m-2}^m}{(N_{l-m}^m)^2 (N_{l-m-2}^m)^2} + \ldots\right.$$
$$\left. + \left[\frac{(N_{l-m+2}^m)^2}{\beta_{l-m+2}^m} + \frac{(N_{l-m+2}^m)^2 (N_{l-m+4})^2}{\beta_{l-m+2}^m \beta_{l-m+4}^m} + \ldots\right]\right]. \tag{15}$$

The coefficients $d_r^{ml}(c)$ of the Bessel series (2), given by the above procedure, are correct to a multiplicative constant, which can be determined by imposing a normalization condition. Flammer [1] proposed the following normalization scheme:

$$\sum_{r=0}^{\infty}{}' \frac{(-1)^{r/2}(r+2m)!}{2^r\left(\frac{r}{2}\right)!\left(\frac{r+2m}{2}\right)!} d_r^{ml} = \frac{(-1)^{(l-m)/2}(l+m)!}{2^{l-m}\left(\frac{l-m}{2}\right)!\left(\frac{l+m}{2}\right)!} \quad \text{for even } l-m, \tag{16}$$

$$\sum_{r=1}^{\infty}{}' \frac{(-1)^{(r-1)/2}(r+2m+1)!}{2^r\left(\frac{r-1}{2}\right)!\left(\frac{r+2m+1}{2}\right)!} d_r^{ml} = \frac{(-1)^{(l-m-1)/2}(l+m+1)!}{2^{l-m}\left(\frac{l-m-1}{2}\right)!\left(\frac{l+m+1}{2}\right)!} \quad \text{for odd } l-m. \tag{17}$$

3. Code description

The package consists of 6 subprograms which must be called from the main program in the following sequence: VALPROP, COEF, NORM, FFACT, R1 and R2. The first 3 subroutines calculate the eigenvalue $\lambda_{ml}(c)$ and then the coefficients $d_r^{ml}(c)$. The last 3 subprograms are of FUNCTION type and calculate factorials, and prolate radial spheroidal functions of first and second kind, respectively, employing the coefficients stored in vector form. Although this procedure can require considerable storage space (as the number of terms in (2) can be large) it has the advantage of saving computing time when one has to calculate $R_{ml}^{(p)}(c, \xi)$ for many values of ξ; this is particularly the case when one has to employ the functions $R_{ml}^{(p)}(c, \xi)$ in numerical integrations on the ξ coordinate.

We shall now give a few details on the way the subprograms were written.

3.1. VALPROP

As we already mentioned, the eigenvalue $\lambda_{ml}(c)$ can be obtained by solving eq. (11) or by using the expansion (14), followed in either way by the refinement (15). We found, however, that it is mostly convenient to use as a starting value in the refinement procedure only the first term of (14), that is $\lambda_{ml} = l(l+1)$, followed by a procedure more rapidly convergent than the iterative use of (15): we took at the beginning only a small number of terms in the expansion (15) and we used it to refine λ_{ml} iteratively to the desired accuracy ERRV; then we repeated the refinement with an increased number of terms in (15), employing as starting value the last value of λ_{ml}. The calculation ends when for a particular number of terms in (15) one arrives at the accuracy ERRV after a single iteration. Our method is shown graphically in fig. 1. The power series expansion (14) is, however, not appropriate for $c > 4$. For these values of c we employed the asymptotic expansion (21.7.6) of ref. [2].

3.2. COEF and NORM

The calculation of the coefficients $d_r^{ml}(c)$ by the recurrence relation (3.1.4) of ref. [1] is not convenient since it becomes unstable as r increases. We have therefore used the eqs. (3)–(10) to obtain coefficients to an accuracy ERRD. The normalization scheme (16) and (17) was then applied in subroutine NORM to give the final values of the coefficients. Because the coefficients are extremely small for large values of r, we employed a vector D to store the mantissae of the coefficients, while a second vector ORD was used to store the ratios of the orders of magnitude of each two consecutive coefficients.

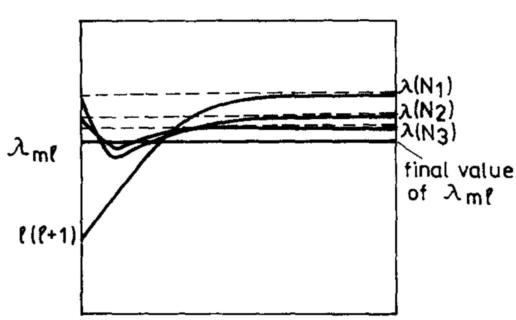

Fig. 1. The method used in the subroutine VALPROP to determine $\lambda_{ml}(c)$; $\lambda(N_i)$ is the eigenvalue obtained iteratively with N_i terms in the expansion (15) and $N_{i+1} = N_{i+1}$ $(i = 1, 2, \ldots)$.

While the functions $R_{ml}^{(1)}(c, \xi)$ given by (2) converge to any accuracy after summing a small number of terms (for ERRB = 10^{-6} the number of terms is typically between 3 and 8), the functions $R_{ml}^{(2)}(c, \xi)$ are slowly convergent, implying a fairly large number of terms and hence coefficients. COEF and NORM

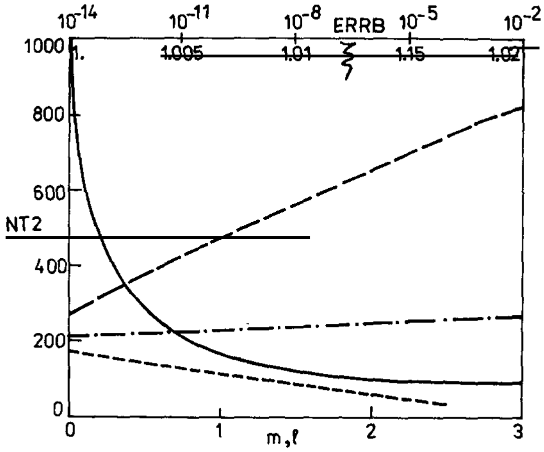

Fig. 2. The variation of NT2 with different parameters: ——— with ξ ($m = l = 0$, $c = 0.1$, ERRB $= 10^{-4}$); — — — with m ($l = 3$, $\xi = 1.005$, $c = 1.$, ERRB $= 10^{-4}$); ·-·-· with l ($m = 0$, $\xi = 1.005$, $c = 1.$, ERRB $= 10^{-4}$); ------ with ERRB ($l = m = 0$, $c = 1.$, $\xi = 1.077$).

require NIT, an estimate of this number, which depends on the values of the relative accuracy ERRB and of the arguments l, m, c, ξ (see fig. 2). The subroutines are called from the main program with the instructions:

$$\text{CALL} \begin{matrix}\text{COEF}\\ \text{NORM}\end{matrix} (\text{NIT, D, ORD}).$$

3.3. *R1 and R2*

These functions calculate the series (2) giving $R_{ml}^{(1)}(c, \xi)$ and $R_{ml}^{(2)}(c, \xi)$, respectively. They employ the coefficients provided by NORM. The spherical Bessel functions of the first kind j_n were calculated with the series expansion [2]:

$$j_n(z) = \frac{z^n}{1 \times 3 \times 5 \ldots (2n+1)} \left\{ 1 - \frac{z^2/2}{1!(2n+3)} + \frac{(z^2/2)^2}{2!(2n+3)(2n+5)} - \ldots \right\}, \tag{18}$$

to an accuracy ERRB, while the spherical Bessel functions of the second kind were obtained by employing the recurrence relation:

$$y_{n-1}(z) + y_{n+1}(z) = (2n+1) z^{-1} y_n(z), \tag{19}$$

with the initial functions:

$$y_0(z) = -\cos(z)/z \quad \text{and} \quad y_1(z) = -\cos(z)/z^2 - \sin(z)/z.$$

The functions R1 and R2 end when the relative accuracy ERRB is achieved; the number of terms

corresponding to this situation is stored in NT1 and NT2, respectively, and can be printed in the main program. When the value of NIT for some particular set of parameters is smaller than the number of terms required by ERRB a message is printed indicating that the corresponding radial spheroidal function was calculated with an accuracy poorer than ERRB. The functions appear in the main program as:
R1
R2 (XI, NIT, D, ORD) where XI corresponds to ξ.

3.4. The COMMON block SF

The COMMON block contains the following variables:

L, M, C the parameters in $R_{ml}(c, \xi)$,
LM = L − M,
RI an integer variable containing 0 for even LM and 1 for odd LM,
VAL real variable corresponding to $\lambda_{ml}(c)$,
ERRB, ERRV, ERRD, real variables corresponding to the accuracies mentioned already,
NT1, NT2 integer variables containing the number of terms in the series (2) of $R^{(1)}_{ml}(c, \xi)$ and $R^{(2)}_{ml}(c, \xi)$, respectively, for which the accuracy ERRB can be achieved.

The coefficient vectors D and ORD were not included in the COMMON block because of the particular features of our system, but this procedure might be required when the package is used on different computers.

3.5. Input and output in the main program

The main program has to attribute values to the following variables: ERRB, ERRV, ERRD, NIT, L, M, C, XI. Except for XI, which contains the value of ξ, all the other variables have to be defined before calling the sequence VALPROP, COEF, NORM. Then the functions R1 and R2 will provide the radial spheroidal functions for different values of XI.

In addition to the values of $R^{(1,2)}_{ml}(c, \xi)$, in the main program can also be printed the values of the eigenvalues stored in VAL, of the coefficients from the vectors D and ORD of dimension NIT and the values stored in NT1 and NT2.

4. Test deck and accuracy checks

The test deck contains a main program which calls the subprograms for $l = 0(1)3$, $m = 0(1)l$ and $c = 1.(1.)5.$, $\xi = 1.005$, 1.02, 1.044, 1.077 and lists the results in a tabular form, that is easy to compare with the tables already existing in the literature [1,2]. The program is given in single precision and all the error variables have the value 10^{-6}. This precision is sufficient in some applications as it gives about 4 accurate figures. Corresponding to ERRB = 10^{-6} the maximum number of terms for the parameters used in the test run is NT2MX = 1379; this number corresponds to $\xi = 1.005$ and M = L = 3. In order to show the message which indicates too small a value of NIT, we employed in the test run NIT = 1370 and the output will show the message for M = L = 3 and each value of c.

The test program calculates NIT = 1370 coefficients for all the combinations m, l, c, although NT2MX, given for each combination in the last column of the output, is, except for M = L = 3, smaller than NIT. The difference in the execution time due to the calculation of the unnecessary coefficients is, however, very small.

Fig. 2 helps the user to guess the choice of NIT by providing some information on how the value of NT2 varies with different parameters. It can be seen from fig. 2 that NT2 increases rapidly as $\xi \to 1$, the variation with m is almost linear and the variation with l is relatively small. On the other hand the variation of NT2 with c is very small (see the test run output for the variation of NT2MX with c).

The accuracy implied by ERRB $= 10^{-6}$ was found satisfactory by the present authors. However, if one desires a better accuracy one has to employ lower values of ERRB, ERRV, ERRD, and one has implicitly to run the routines in double precision. To do this one has to include at the beginning of the routines the following statements:

ABS(X) = DABS(X) in VALPROP, COEF, R1 and R2,
INT(X) = IDINT(X) in VALPROP, NORM, R1, R2,
FLOAT(I) = DFLOAT(I) in FFACT,
SIN(X) = DSIN(X) and COS(X) = DCOS(X) in R2.

The decrease in ERRB, ERRV and ERRD does not influence the results significantly. The variation of NT2 with ERRB depends on the values of the other parameters; for instance in the case of $\xi = 1.077$, $l = m = 0$, the variation of NT2 is about linear with the order of ERRB, (fig. 2), while as $\xi \to 1$ the variation becomes much faster. For $\xi \to 1$, very small values of ERRB and large m, NT2 can become very large, implying large storage space for the vectors D and ORD and a considerable increase in the computing time.

References

[1] C. Flammer, Spheroidal Wave Functions (Stanford Univ. Press, Stanford, 1957) (Russian translation, 1962).

[2] Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables, eds. M. Abramowitz and I.A. Stegun (Dover, New York, 1970).

TEST RUN OUTPUT

NIT= 1370
ERRB= 1.0E-06

M	L	C	R1(1.005)	R1(1.02)	R1(1.04)	R1(1.077)	R2(1.005)	R2(1.02)	R2(1.044)	R2(1.077)	NT2MX
0	0	1.	9.4675E-01	9.4186E-01	9.3392E-01	9.2276E-01	-2.8376E+00	-2.0958E+00	-1.6663E+00	-1.3557E+00	580
0	0	2.	8.2565E-01	8.0769E-01	7.7890E-01	7.3922E-01	-1.2445E+00	-8.0205E-01	-5.3424E-01	-3.3342E-01	602
0	0	3.	7.0257E-01	6.6620E-01	6.0913E-01	5.3307E-01	-7.1089E-01	-3.5088E-01	-1.2875E-01	3.4306E-02	629
0	0	4.	6.0543E-01	5.4713E-01	4.5854E-01	3.4627E-01	-4.5179E-01	-1.2986E-01	6.4845E-02	1.9392E-01	655
0	0	5.	5.3149E-01	4.4897E-01	3.2846E-01	1.8700E-01	-2.9783E-01	-1.3343E-03	1.6612E-01	2.5396E-01	682
0	1	1.	3.1531E-01	3.1902E-01	3.2489E-01	3.3277E-01	-6.9113E+00	-4.8010E+00	-3.6692E+00	-2.9198E+00	597
0	1	2.	5.2887E-01	5.2981E-01	5.3079E-01	5.3113E-01	-2.1883E+00	-1.5403E+00	-1.1767E+00	-9.2153E-01	592
0	1	3.	6.0635E-01	5.9601E-01	5.7856E-01	5.5286E-01	-1.1333E+00	-7.3659E-01	-4.9878E-01	-3.2080E-01	602
0	1	4.	5.8921E-01	5.6124E-01	5.1623E-01	4.5423E-01	-6.7401E-01	-3.5273E-01	-1.5339E-01	-4.9280E-03	623
0	1	5.	5.3812E-01	4.8878E-01	4.1249E-01	3.1364E-01	-4.3047E-01	-1.4018E-01	3.7271E-02	1.5718E-01	650
0	2	1.	4.4700E-02	4.6546E-02	4.9538E-02	5.3728E-02	-3.5923E+01	-2.1843E+01	-1.4835E+01	-1.0558E+01	625
0	2	2.	1.6961E-01	1.7490E-01	1.8329E-01	1.9465E-01	-5.2369E+00	-3.3542E+00	-2.3983E+00	-1.8013E+00	616
0	2	3.	3.2951E-01	3.3463E-01	3.4211E-01	3.5093E-01	-2.0245E+00	-1.3563E+00	-9.9887E-01	-7.5939E-01	605
0	2	4.	4.5065E-01	4.4765E-01	4.4124E-01	4.2932E-01	-1.1123E+00	-7.2321E-01	-4.9658E-01	-3.2978E-01	605
0	2	5.	4.9521E-01	4.7631E-01	4.4438E-01	3.9761E-01	-6.8149E-01	-3.7960E-01	-1.9271E-01	-5.1527E-02	619
0	3	1.	3.9117E-03	4.2495E-03	4.8137E-03	5.6379E-03	-3.2873E+02	-1.7749E+02	-1.0807E+02	-6.9145E+01	645
0	3	2.	3.0945E-02	3.3170E-02	3.7002E-02	4.2491E-02	-2.1948E+01	-1.2241E+01	-7.7128E+00	-5.1316E+00	640
0	3	3.	9.9557E-02	1.0535E-01	1.1466E-01	1.2746E-01	-5.0167E+00	-2.9711E+00	-1.9912E+00	-1.4154E+00	631
0	3	4.	2.1065E-01	2.1826E-01	2.2983E-01	2.4426E-01	-2.0426E+00	-1.2921E+00	-9.1301E-01	-6.7330E-01	618
0	3	5.	3.2979E-01	3.3291E-01	3.3596E-01	3.3616E-01	-1.1359E+00	-7.2794E-01	-5.0132E-01	-3.4018E-01	610
1	1	1.	3.2699E-02	6.5443E-02	9.7155E-02	1.2864E-01	-1.5054E+01	-7.2938E+00	-4.7341E+00	-3.4319E+00	925
1	1	2.	6.1371E-02	1.2267E-01	1.7933E-01	2.3230E-01	-4.0786E+00	-2.0770E+00	-1.4174E+00	-1.0708E+00	922
1	1	3.	8.5960E-02	1.6769E-01	2.3864E-01	2.9730E-01	-2.0187E+00	-1.0747E+00	-7.4528E-01	-5.4789E-01	919
1	1	4.	1.0535E-01	2.0070E-01	2.7444E-01	3.2214E-01	-1.2732E+00	-6.9122E-01	-4.5871E-01	-2.9277E-01	915
1	1	5.	1.2109E-01	2.2350E-01	2.8029E-01	3.1169E-01	-9.0904E-01	-4.8688E-01	-2.8534E-01	-1.2353E-01	912
1	2	1.	6.5033E-03	1.3221E-02	2.0118E-02	2.7537E-02	-7.2945E+01	-3.2693E+01	-1.9389E+01	-1.2753E+01	929
1	2	2.	2.3779E-02	4.8016E-02	7.2266E-02	9.7376E-02	-1.0141E+01	-4.7171E+00	-2.9321E+00	-2.0382E+00	927
1	2	3.	4.6581E-02	9.2957E-02	1.3720E-01	1.7975E-01	-3.5511E+00	-1.7507E+00	-1.1563E+00	-8.4720E-01	924
1	2	4.	6.9748E-02	1.3674E-01	1.9596E-01	2.4598E-01	-1.8423E+00	-9.5975E-01	-6.5312E-01	-4.7190E-01	920
1	2	5.	9.0348E-02	1.7285E-01	2.3764E-01	2.8030E-01	-1.1775E+00	-6.3196E-01	-4.1651E-01	-2.6427E-01	916

```
 1  3   1.      7.5857E-04  1.5769E-03  2.4832E-03  3.5559E-03      -6.0131E+02 -2.4911E+02 -1.3542E+02 -8.1241E+01   934
 1  3   2.      5.7249E-03  1.1929E-02  1.6450E-02  2.6066E-02      -4.0269E+01 -1.7076E+01 -9.5575E+00 -5.9419E+00   932
 1  3   3.      1.7370E-02  3.5533E-02  5.4527E-02  7.5289E-02      -9.0273E+00 -4.0015E+00 -2.3677E+00 -1.5719E+00   930
 1  3   4.      3.5156E-02  7.0499E-02  1.0625E-01  1.4184E-01      -3.4501E+00 -1.6308E+00 -1.0349E+00 -7.3269E-01   927
 1  3   5.      5.6043E-02  1.1083E-01  1.6082E-01  2.0481E-01      -1.7996E+00 -9.0871E-01 -6.0363E-01 -4.2889E-01   922


 2  2   1.      6.6119E-04  2.6588E-03  5.8978E-03  1.0435E-02      -3.7492E+02 -9.1118E+01 -3.9736E+01 -2.1563E+01  1170
 2  2   2.      2.5659E-03  1.0250E-02  2.2494E-02  3.9194E-02      -4.8516E+01 -1.2027E+01 -5.4176E+00 -3.0774E+00  1169
 2  2   3.      5.5203E-03  2.1967E-02  4.6982E-02  7.9738E-02      -1.5148E+01 -3.8902E+00 -1.8545E+00 -1.1313E+00  1168
 2  2   4.      9.3014E-03  3.6162E-02  7.5869E-02  1.2390E-01      -6.8188E+00 -1.8426E+00 -9.4291E-01 -6.1319E-01  1167
 2  2   5.      1.3719E-02  5.2225E-02  1.0577E-01  1.6392E-01      -3.7546E+00 -1.0806E+00 -5.9040E-01 -3.9044E-01  1164


 2  3   1.      9.4154E-05  3.8445E-04  8.7354E-04  1.5962E-03      -2.6080E+03 -6.0955E+02 -2.5170E+02 -1.2786E+02  1171
 2  3   2.      7.1284E-04  2.8957E-03  6.5244E-03  1.1781E-02      -1.7279E+02 -4.0946E+01 -1.7270E+01 -9.0309E+00  1171
 2  3   3.      2.2065E-03  8.8948E-03  1.9731E-02  3.4901E-02      -3.7424E+01 -9.0936E+00 -3.9925E+00 -2.2074E+00  1170
 2  3   4.      4.6828E-03  1.8616E-02  4.0479E-02  6.9460E-02      -1.3338E+01 -3.3692E+00 -1.5730E+00 -9.3961E-01  1168
 2  3   5.      8.0600E-03  3.1498E-02  6.6567E-02  1.0958E-01      -6.2728E+00 -1.6713E+00 -8.4092E-01 -5.3796E-01  1167


FOR  L= 3  M= 3  C=1.00000   XI=1.00500   ** ERROR GREATER THEN ERRR, INCREASE NIT **
 3  3   1.      9.4962E-06  7.6695E-05  2.5403E-04  6.0004E-04      -1.7427E+04 -2.1153E+03 -6.1960E+02 -2.5230E+02  1371
FOR  L= 3  M= 3  C=2.00000   XI=1.00500   ** ERROR GREATER THEN ERRR, INCREASE NIT **
 3  3   2.      7.4529E-05  5.9887E-04  1.9671E-03  4.5920E-03      -1.1121E+03 -1.3637E+02 -4.0615E+01 -1.6937E+01  1371
FOR  L= 3  M= 3  C=3.00000   XI=1.00500   ** ERROR GREATER THEN ERRR, INCREASE NIT **
 3  3   3.      2.4424E-04  1.9459E-03  6.3022E-03  1.4420E-02      -2.2689E+02 -2.8311E+01 -8.6855E+00 -3.7861E+00  1371
FOR  L= 3  M= 3  C=4.00000   XI=1.00500   ** ERROR GREATER THEN ERRR, INCREASE NIT **
 3  3   4.      5.5796E-04  4.3914E-03  1.3941E-02  3.0994E-02      -7.4792E+01 -9.5746E+00 -3.0735E+00 -1.4346E+00  1371
FOR  L= 3  M= 3  C=5.00000   XI=1.00500   ** ERROR GREATER THEN ERRR, INCREASE NIT **
 3  3   5.      1.0453E-03  8.0973E-03  2.5038E-02  5.3577E-02      -3.2111E+01 -4.2563E+00 -1.4549E+00 -7.4047E-01  1371
```