UNCLASSIFIED

NRL Report 5947

A DIGITAL COMPUTER PROGRAM IN NAREC CODE FOR CALCULATING PROLATE RADIAL SPHEROIDAL WAVE FUNCTIONS

S. Hanish

Transducer Branch
Sound Division

July 10, 1963

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

UNCLASSIFIED

CONTENTS

Abstract	ii
Problem Status	ii
Authorization	ii
INTRODUCTION	1
SPHEROIDAL WAVE FUNCTION EXPANSION CONSTANTS	2
RADIAL FUNCTIONS OF THE FIRST TYPE	2
RADIAL FUNCTIONS OF THE SECOND TYPE	3
THE DIGITAL COMPUTER PROGRAM FOR THE RADIAL FUNCTIONS	3
PREPARATION OF INPUT DATA TAPE	4
LIST AND TRANSCRIPTIONS OF THE TAPES	6
ACKNOWLEDGMENT	6
REFERENCES	7
APPENDIX A – List of NAREC-Coded Programs	8
APPENDIX B – Asymptotic Formulas for Calculating the Complex Prolate Radial Spheroidal Wave Function $R^{(3)}_{m\ell}$ of the Third Kind	12
APPENDIX C – NAREC Coding for Tape No. 4712A	13
APPENDIX D – NAREC Coding for Tape No. 4712D	57

ABSTRACT

The numerical computation of prolate radial spheroidal wave functions of the first and second type and their derivatives as defined by Morse and Feshbach ("Methods of Theoretical Physics," Vol. II, page 1505, McGraw-Hill, 1953) in series of spherical Bessel and spherical Neumann functions has been coded by the author for the NAREC (the high-speed digital computer at the Naval Research Laboratory). Following a brief review of the pertinent theory the author presents the complete digital computer program and explains the meaning of individual commands and blocks of commands in detail. The coded program in action accepts a read-in of the following parameters: the prolate spheroidal wave function expansion constants (calculated by a special coded program found in Hanish and Baier, "A Digital Computer Program for Calculating Spheroidal Wave Function Eigenvalues and Expansion Constants," NRL Report 5837, Jan. 31, 1963), the modal numbers m and ℓ, the radial coordinate ξ, the eccentricity parameter h, and the total number of terms to be summed in the several series. The program delivers the following numerical results: $je_{m\ell}(h, \xi)$, $ne_{m\ell}(h, \xi)$, $dje_{m\ell}/d\xi$, $dne_{m\ell}/d\xi$, the calculated Wronskian, and the theoretical Wronskian. A comparison of the two final quantities reveals the precision of the calculation.

PROBLEM STATUS

This is an interim report on one phase of the problem; work is continuing.

AUTHORIZATION

NRL Problem S02-07
Project RF 001-03-44-4052

Manuscript submitted March 14, 1963.

UNCLASSIFIED

A DIGITAL COMPUTER PROGRAM IN NAREC CODE FOR PROLATE RADIAL SPHEROIDAL WAVE FUNCTIONS

INTRODUCTION

When the scalar Helmholtz equation

$$\nabla^2\psi + k^2\psi = 0, \qquad k = \omega/c \tag{1}$$

is separated in prolate spheroidal coordinates (ξ, η, φ) (see Ref. 1), the ordinary differential equation for the radial coordinate ξ becomes

$$\frac{d}{d\xi}\left[(\xi^2 - 1)\frac{dR}{d\xi}\right] - \left[A - h^2\xi + \frac{m^2}{\xi^2 - 1}\right]R = 0 \tag{2}$$

where $\xi \geq 1$, m and A are separation constants, and h is 1/2 the product of k and the interfocal distance d. In physical problems m is usually an integer.

Equation (2) is generally invalid for arbitrary A, so that a search must be made to find particular values of A for given h that make the equation valid. These special values of A are the eigenvalues of the differential equation. Noting that Eq. (2) for $h = 0$ is identical with Legendre's equation for associated functions, whose eigenvalues are $\ell(\ell + 1)$, where ℓ is an integer, it is convenient to distinguish the eigenvalues A among themselves by assigning the subscripts $m\ell$. For each $A_{m\ell}$ there is a particular solution $R_{m\ell}$ of Eq. (2), designated as the radial wave function in spheroidal coordinates. Since equations of the second degree have two independent solutions, it is appropriate to add the superscripts 1 or 2 to indicate which of these solutions are meant. Thus we have the notations $R_{m\ell}^{(1)}$ and $R_{m\ell}^{(2)}$ denoting radial functions of the first or second type corresponding to the eigenvalues $A_{m\ell}$. These notations are not standard. They differ for example from Morse and Feshbach (2) who use $je_{m\ell}$ and $ne_{m\ell}$ to designate these radial functions respectively.

The radial functions satisfying Eq. (2) are functions of the radial coordinate ξ and the parameter h. On a given prolate spheroidal envelope for which $\xi = \xi_o$, the ratio of semiminor axis a to semimajor axis b is

$$\frac{a}{b} = \frac{\sqrt{\xi_o^2 - 1}}{\xi_o}. \tag{3a}$$

If one examines the equations of transformation from Cartesian coordinates (x, y, z) to spheroidal coordinates (ξ, η, φ), one sees that

$$ka = h\sqrt{\xi_o^2 - 1} \tag{3b}$$

$$kb = h\xi_o. \tag{3c}$$

Since the parameters $h\xi$, $h\sqrt{\xi^2 - 1}$, and h may be varied independently to take on values from zero to infinity, the radial functions are observed to span great ranges of argument, and are predicted to have several asymptotic formulas over these ranges. Diversity in asymptotic formulas is a notable feature of spheroidal functions in general (3). The computer programs described below have provisions for application of a few of these special equations.

SPHEROIDAL WAVE FUNCTION EXPANSION CONSTANTS

The separation of the scalar Helmholtz equation in angle coordinate η leads to the ordinary differential equation

$$\frac{d}{d\eta}\left[(1 - \eta^2)\frac{dS}{d\eta}\right] + \left(A - h^2\eta^2 - \frac{m^2}{1 - \eta^2}\right)S = 0. \tag{4}$$

Since the eigenvalues A of this equation are identical with the eigenvalues of Eq. (2), one may classify the solutions of Eq. (4) by the subscript $m\ell$, as before. When $h \to 0$ the solution $S_{m\ell}(h, \eta)$ approaches the associated Legendre function P_ℓ^m and $A_{m\ell}$ approaches $\ell(\ell + 1)$. It is therefore natural to expand $S_{m\ell}(h, \eta)$ in a sum of products of associated Legendre functions $P_{n+m}^m(\eta)$ and expansion constants $d_n(h|m\ell)$:

$$S_{m\ell}(h, \eta) = \sum_{n=0,1}{}' \, d_n(h|m\ell)\, P_{n+m}^m(\eta) \tag{5}$$

in which the prime indicates a sum to infinity on even integers or odd integers exclusively.

A full discussion on the theory leading to the method of calculating these expansion constants together with tabulations are found in Refs. 4 and 5. For those desiring to compute these constants on an available large computing machine for values other than tabulated in the literature a digital computer program is presented with complete detail in Ref. 6.

RADIAL FUNCTIONS OF THE FIRST TYPE

When a selected prolate spheroid differs but little from a true sphere, the scalar radiation field generated by it ought to approximate very closely the radiation field of the sphere. In particular the radial functions of the spheroid should reduce to the spherical Bessel functions of the sphere. It is therefore convenient to expand the radial functions in series of products of spherical Bessel functions and expansion constants. The similarity of Eqs. (2) and (4) lead to the fortunate result that the expansion constants for the radial functions can be made identical with the expansion constants for the angle functions (see Eq. (5)). If we adopt the normalization procedure of Morse and Feshbach (2), we can write $R_{m\ell}^{(1)}$ to be identical with Eq. (11.3.91) of their book. This formula has the form

$$R_{m\ell}^{(1)}(h, \xi) = je_{m\ell}(h, \xi) = \frac{(\ell - m)!}{(\ell + m)!}\left(\frac{\xi^2 - 1}{\xi^2}\right)^{m/2} \sum_{n}^{p}{}' \, i^{n+m-\ell} d_n(h|m\ell)\, \frac{(n + 2m)!}{n!}\, j_{n+m}(h\xi) \tag{6}$$

where $|\xi| > 1$ and $j_{n+m}(h\xi)$ are spherical Bessel functions of the first type.

UNCLASSIFIED

Clearly the utility of this formula is restricted to the case of spheroids of moderate eccentricity (i.e., $a/b < 3$). For highly eccentric prolate spheroids Eq. (6) must be extended or replaced by asymptotic forms. A useful asymptotic formula found in Ref. 3 has been programmed by the author and is presented in Appendix B of this report. Experience with the numerical computation of Eq. (6) shows that it converges rapidly in all cases where the magnitude of h is not too large (say $h < 8$). The derivative of $R_{m\ell}^{(1)}$ $(= R_{m\ell}^{(1)\prime})$ with respect to ξ is easily programmed because of the simplicity of the well-known recursion formulas for the spherical Bessel functions. The rate of convergence of the series for $R_{m\ell}^{(1)\prime}$ is slower than for $R_{m\ell}^{(1)}$, but in the author's experience this rate has always proved satisfactory in problems of physics.

RADIAL FUNCTIONS OF THE SECOND TYPE

The radial functions of the second type may similarly be expanded in series of products of spherical Neumann functions $(= n_n)$. We adopt the normalization procedure of Ref. 2 again and copy below Eq. (11.3.93) of this reference:

$$R_{m\ell}^{(2)} = ne_{m\ell}(h, \xi)$$

$$= \frac{(\ell - m)!}{(\ell + m)!} \left(\frac{\xi^2 - 1}{\xi^2}\right)^{m/2} \sum_{n=0,1}^{p}{}' \; i^{n+m-\ell} \, d_n(h \mid m\ell) \, \frac{(n + 2m)!}{n!} \, n_{n+m}(h\xi) \tag{7}$$

where $|\xi| > 1$.

This is an asymptotic series, by which is meant that for n finite the series will not converge absolutely unless h becomes indefinitely large. Many authors have found this equation to converge so slowly for fixed (i.e., finite) h that they disparage its employment. The present author has, however, retained Eq. (7) for all his calculations of the radial function of the second type. He has been helped in this choice by the availability of a high speed digital computer which overcomes the difficulties of slow convergence by calculating hundreds of Bessel functions and expansion constants in a relatively short time. In those cases where ξ is not too close to unity (i.e., $\xi > 1.05$) and h is not too close to zero (i.e., $h > 0.5$) or where h is not too large (i.e., $h < 10$) the series has proved adequate. For the special cases outside the useful range noted above the author has prepared subsidiary programs of the asymptotic type which deliver useful calculations, although with some decrease in precision of the numerical results.

Equation (7) may occasionally be displaced in favor of one-term or two-term asymptotic forms for values of h and ξ within the limits of these short formulas. However, finding the correct simple formula is mostly a matter of chance. For guidance the reader may search through Ref. 3, which contains a good number of these equations of restricted range. For those workers whose knowledge of radial functions is small, it is decidedly preferable to obtain values needed for calculation by computing them on a digital computer machine from some formulas of general enough range, such as Eqs. (6) and (7).

THE DIGITAL COMPUTER PROGRAM FOR THE RADIAL FUNCTIONS

The computation of the radial prolate spheroidal wave functions of the first and second types, their first derivatives with respect to ξ, and the value of the Wronskian is carried

out in a single step. The basic coding of this program is done in NAREC fixed point (7). There is as well a large floating point section written in the interpretive NIP I language (8). The main radial function coding is punched on the basic tape No. 4712A. Three other tapes (No. 4712B, No. 4712C, No. 4712D) incorporating additional coding for improving the precision of calculation at special ranges of the arguments are available. When any of these tapes are used, it is necessary to read into the memory NIP I tape No. 2700 to interpret all floating point commands of the coding into fixed point language. In addition, before starting, the memory of the computer must contain all of the wave function expansion constants (if even-numbered, d_0, d_2, d_4, ..., d_{xxx}; if odd-numbered, d_1, d_3, d_5, ..., d_{xxx1}) as floating point numbers in consecutive locations beginning with address 16AO. These constants are supplied by tape No. 4593 (7). After the input data is read in (the method of preparing input data will be presented below) the starting order is given and the computation proceeds automatically until the required radial functions are delivered on the output sheet.

The actual computation is based upon Eqs. (6) and (7) and is quite straightforward. The data input tape contains specific values of the parameters m, ℓ, h, and ξ. It also contains the number of terms $p/2$ to be summed in the indicated series. At the onset of calculation the difference $\ell - m$ is formed, and the computer determines whether the difference is an even or odd integer. If even, the symbol n takes on all even values from 0 to p. If odd, n takes on all odd values from 1 to p. The program proceeds to calculate all spherical Bessel and spherical Neumann functions from order m to order $p + m$ with argument $h\xi$, together with their first derivatives. Although the magnitude of p may be selected at will, tape No. 4712A permits a maximum p of 400. Since the expansion constants $d_n(h|m\ell)$ are in consecutive addresses of the memory beginning at address 16AO (having been previously put there by tape No. 4593), the products of expansion constants and Bessel functions required by Eqs. (6) and (7) are easily formed. Final summations yield all the radial functions and their derivatives. A check on the precision of computation is made by comparing the calculated Wronskian with the theoretical Wronskian ($= 1/h(\xi^2 - 1)$). If the precision of the answers is insufficient, the entire calculation may be repeated with p set at a higher (integer) value.

When h is small (i.e., $h < 0.5$), convergence of the series in Eq. (6) or (7) is very poor. Since the consecutive terms in these series eventually diminish at the rate of $1/\xi^2$, it is possible to sum these series from $n = 0$ or $n = 1$ to p and then to estimate the remainder (from p to ∞) by assuming a geometric series with ratio $1/\xi^2$. This procedure is followed by the coding punched on tape No. 4712D.

When the magnitude of p is required to exceed 400 it is necessary to use tape No. 4712C (or 4712D).

PREPARATION OF INPUT DATA TAPE

The input data tape is the same for all programs. It contains two separate read-in sections. The first section is made of numerical values of ℓ, m, $h\xi$, and p in sequence written as floating point numbers. Since each such number occupies two NAREC words, the section will contain eight entries. The second section is made up of the numerical values of ξ and h in sequence, written as floating point numbers. This section therefore contains four entries. As a typical example let $\ell = 5$, $m = 1$, $h\xi = 7.26$, $p = 200$, $\xi = 1.1$, and $h = 6.6$. The input data tape will read as follows.

UNCLASSIFIED

Section I

nnnnn

$$\left.\begin{array}{l}0A00\ 00\ 000000\\ 0000\ 00\ 000000\end{array}\right\}\ \ell = 5$$

$$\left.\begin{array}{l}0800\ 00\ 000000\\ 0001\ 00\ 000000\end{array}\right\}\ m = 1$$

$$\left.\begin{array}{l}0E85\ 1E\ B851EB\\ 0003\ 00\ 000000\end{array}\right\}\ h\xi = 7.26$$

$$\left.\begin{array}{l}0CA0\ 00\ 000000\\ 0007\ 00\ 000000\end{array}\right\}\ p = 200$$

mmmm

Section II

nnnn

$$\left.\begin{array}{l}08CC\ CC\ CCCCCC\\ 0001\ 00\ 000000\end{array}\right\}\ \xi = 1.1$$

$$\left.\begin{array}{l}0D33\ 33\ 333333\\ 0003\ 00\ 000000\end{array}\right\}\ h = 6.6$$

mmmm

All input tapes must be sent to the NRL Research Computation Center, Code 4550, for further processing.

The output has various forms and depends upon the tape number. Specific outputs for specific tapes are given in the Appendixes. However, all tapes will yield the following output (for the above example) though not necessarily in the order shown below:

$$R_{15}^{(1)}(6.6, 1.1)$$

$$\frac{dR_{15}^{(1)}(6.6,\ 1.1)}{d\xi}$$

$$R_{15}^{(2)}(6.6,\ 1.1)$$

$$\frac{dR_{15}^{(2)}(6.6,\ 1.1)}{d\xi}$$

Calculated Wronskian

Theoretical Wronskian.

LIST AND TRANSCRIPTIONS OF THE TAPE

Appendix A lists all tapes, together with the following pertinent information:

First Word of Coding

Last Word of Coding

Standard Check Sum

Manual Changes

Starting Order

Output Format

Auxiliary Remarks.

Appendix C contains a complete transcription of the coding for tape No. 4712A, together with a detailed explanation of the meaning of the various commands and various blocks of commands. Appendix D contains a complete transcription of the coding for tape No. 4712D. Except for the small section located in addresses 2C4a to 2C9B which is explained in detail, no further comments on this program are offered.

The digital computer programs described in the Appendixes of this report are complete in every respect and have been thoroughly checked out. They may be transcribed into the language of other computer systems with assurance.

ACKNOWLEDGMENT

The author expresses his gratitude to R. V. Baier of the Transducer Branch, Sound Division, for diligent assistance in the preparation of tape No. 4712A.

REFERENCES

1. Meixner, J., and Schafke, F.W., "Mathieusche Funktionen und Sphäroidfunktionen," Berlin:Springer-Verlag, 1954

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

3. Silbiger, A., "Asymptotic Formulas and Computational Methods for Spheroidal Wave Functions," Cambridge Acoustical Associates, Inc., Report U-123-48, ONR Contract Nonr-2739(00)

4. Flammer, C., "Spheroidal Wave Function," Stanford Research Institute Monograph, Stanford, Calif., 1957

5. Stratton, J.A., Morse, P.M., et al., "Spheroidal Wave Functions," New York:Wiley, 1956

6. Hanish, S., and Baier, R.V., "A Digital Computer Program for Calculating Spheroidal Wave Function Eigenvalues and Expansion Constants," NRL Report 5837, Jan. 31, 1963

7. Steward, J.S., "Programming Manual for the NAREC," NRL Report 4652, Nov. 17, 1955

8. Bearce, L.S., Stennett, E., and Mason, R.P., "NIP I," NAREC Bulletin 27, U.S. Naval Research Laboratory, July 27, 1961

APPENDIX A
LIST OF NAREC-CODED PROGRAMS

This Appendix contains a list of programs written in NAREC code by the author for the automatic computation of radial prolate spheroidal wave functions. When the information supplied below is forwarded with an appropriate data tape to the NRL Research Computation Center the input to the NAREC will be complete and the computer will deliver the output called for by the specific tape.

Tape No. 4712A

First Word: 2700 - 2700 50 0200 11

Last Word: 2C49 - 2700 10 0000 00

Standard Check Sum: B324 9E DB 90 DE

Manual Changes: For Prolate Case, none
For Oblate Case, two changes shown below:

address: 275E - 2943 E8 2907 E9

2A24 - 2907 E9 2A2C EB

Input Data Tape: Insert on Reader

Starting Order: LO 2C48

Output: line 1 - $R_{m\ell}^{(1)}$

line 2 - $dR_{m\ell}^{(1)}/d\xi$

line 3 - $R_{m\ell}^{(2)}$

line 4 - $dR_{m\ell}^{(2)}/d\xi$

line 5 - Calculated Wronskian

line 6 - Theoretical Wronskian.

Tape No. 4712B

First Word: 2700 - 2700 50 0200 11

Last Word: 2C47 - 0000 00 0000 00

Standard Check Sum:

UNCLASSIFIED

Manual Changes: For Prolate Case, three changes shown below.

address: 2650 - 28FD 32 2A2A 32

2651 - 2700 10 0000 00

28F2 - 0000 E0 0008 F0

For Oblate Case, five changes shown below:

address: 2650 - 28FD 32 2A2A 32

2651 - 2700 10 0000 00

275E - 2943 E8 2907 E9

2A24 - 2907 E9 2A2C EB

28F2 - 0000 E0 0008 F0

Input Data Tape: Insert on Reader

Starting Order: LO 2650

Output: Many Blocks, as follows:

Block 1 $i^{n+m-\ell}$ for all n

Block 2 $(-1)^{n+m-\ell}$ for all n

Block 3 j_{n+m} for all n

Block 4 $dj_{n+m}/dh\xi$ for all n

Block 5 n_{n+m} for all n

Block 6 $dn_{n+m}/dh\xi$ for all n

Block 7 j_{n+m+1} for all n

Block 8 n_{n+m+1} for all n

Line $R^{(1)}_{m\ell}$

Line $dR^{(1)}_{m\ell}/dh\xi$

Line $R^{(2)}_{m\ell}$

Line Last summand in series for $R^{(2)}_{m\ell}$

Block 9 List of all partial summands in the series for $R^{(2)}_{m\ell}$

Block 9 $dR^{(2)}_{m\ell}/dh\xi$

Block 9 Last summand in series for $dR^{(2)}_{m\ell}/dh\xi$

Block 10 List of all partial summands in the series for $dR^{(2)}_{m\ell}/dh\xi$

Line Calculated Wronskian

Line Theoretical Wronskian.

Tape No. 4712C

First Word: 2700 - 2700 50 0200 11

Last Word: 2C5F - 2C53 E2 0000 E0

Standard Check Sum:

Manual Changes: For Prolate Case, none

For Oblate Case, two changes shown below:

address: 275E - 2943 E8 2907 E9

2A24 - 2907 E9 2A2C EB

Input Data Tape: Insert on Reader

Starting Order: LO 2C48

Output: Same as Tape No. 4712A.

Remarks: This tape sums only 1/2 of the terms in the series for $R^{(1)}_{m\ell}$ and $dR^{(1)}_{m\ell}/d\xi$. It also stores certain spherical Bessel functions in address 0400, etc., to avoid previously noted read-over difficulties when the number of terms in the series (i.e., p/2) exceeded 200.

Tape No. 4712D

First Word: 2700 - 2700 50 0200 11

Last Word: 2CA7 - 2C7D E2 0000 E0

Standard Check Sum: A173 EA F701 99

Manual Changes: Same as for Tape No. 4712C

Input Data Tape: Insert on Reader

Starting Order: LO 2C48

Output: Line $R^{(1)}_{m\ell}$

Line $dR^{(1)}_{m\ell}/d\xi$

UNCLASSIFIED

Line $R^{(2)}_{m\ell}$

Line $i^{p+m-\ell-2} \dfrac{(p+2m-2)!}{(p-2)!} d_{p-2} n_{p+m-2}$

Line $i^{p+m-\ell} \dfrac{(p+2m)!}{p!} d_p n_{p+m}$

Line $i^{p+m-\ell-2} \dfrac{(p+2m-2)!}{(p-2)!} d_{p-2} \left[\dfrac{dn_{p+m-2}}{dh\xi}\right]$

Line $i^{p+m-\ell} \dfrac{(p+2m)!}{p!} d_p \left[\dfrac{dn_{p+m}}{dh\xi}\right]$

Line $\sum_{t=p}^{\infty} i^{t+m-\ell} \dfrac{(t+2m)!}{t!} d_t n_{t+m}$

Line $\sum_{t=p}^{\infty} i^{t+m-\ell} \dfrac{(t+2m)!}{t!} d_t \left[\dfrac{dn_{t+m}}{dh\xi}\right]$

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

Calculate Wronskian

Theoretical Wronskian.

Remarks: This tape is the same as tape No. 4712C except that the series for $R^{(2)}_{m\ell}$ and $dR^{(2)}_{m\ell}/d\xi$ have been summed to infinity by assuming that they have become geometric series with the ratio $1/\xi^2$.

APPENDIX B
ASYMPTOTIC FORMULAS FOR CALCULATING THE COMPLEX PROLATE RADIAL SPHEROIDAL WAVE FUNCTION $R_{m\ell}^{(3)}$ OF THE THIRD KIND

The formulas are

$$R_{m\ell}^{(3)}(h, \xi) \sim \left(\frac{\pi}{2h}\right)^{1/2} H_m^{(1)}(x)$$

$$\frac{dR_{m\ell}^{(3)}(h, \xi)}{d\xi} \sim \frac{x}{\xi^2 - 1} \left(\frac{\pi}{2h}\right)^{1/2} \frac{\partial}{\partial x} H_m^{(1)}(x)$$

$$x = h\sqrt{\xi^2 - 1}\left[1 - \frac{2(n - m) + 1}{2h} - \frac{(n - m)^2 + (n - m) + 1}{4h^2} + \cdots\right].$$

These formulas are valid for $kb \to \infty$, ka arbitrary, $a/b \to 0$, and $\sqrt{\xi^2 - 1} \ll 1$. Here a is the semiminor axis and b is the semimajor axis of the prolate spheroid $\xi = \xi_o$.

UNCLASSIFIED

APPENDIX C
NAREC CODING FOR TAPE NO. 4712A

2700	270050020011	Enter floating point
2701	28fde628ffea	Bring up "ℓ" || Subtract "m"
2702	0000f9290fe8	Form abs. value || Store in 290F
2703	2a50df290f00	Enter odd-even subroutine using (ℓ - m) as parameter
2704	291fe82907dc	If even (odd) subr. stores zero (one) in 291F || Compare with one
2705	2707e22714e2	Even: go to 2707 || Odd: Go to 2714
2706	2706e5000082	Number in 291F > 1. Error: stop
2707	290fe62905dc	Bring up $\|\ell - m\|$ || Compare with zero
2708	2709e5270ae2	If $\|\ell - m\| < 0$: error. Go to right 2709 || If $\|\ell - m\| = 0$ go to 270A
2709	270ce2000082	If $\|\ell - m\| > 0$ go to 270C || Error stop
270a	2907e62913e8	Bring up "1" || Store in 2913
270b	271ee20000e0	Go to 271E ||
270c	290fe6290bea	Bring up $\|\ell - m\|$ || Subtract "4"
270d	2915e80000e0	Store in 2915 || *
270e	2915e62905dc	Bring up 2915 || Compare with zero
270f	2712e2270ae3	If < 0 go to 2712 || If = 0 go to 270A
2710	2915e6290bea	If > 0 bring up 2915 || Subtract "4"
2711	2915e8270ee2	Store in 2915 || Go to 270E
2712	2907e72913e8	Bring up "1" || Store in 2913
2713	271ee20000e0	Go to 271E || *
2714	2907e628fdea	Compute $i^{n+m-\ell}$ Bring up "1" || Subtract ℓ
2715	28ffe90000f9	Add "m" || Form abs. value
2716	2911e82905dc	Store in 2911 || Compare with zero
2717	2718e5270ae2	If < 0 error: go to right of 2718 || If = 0 go to 270A
2718	2719e2000082	If > 0 go to 2719 || Error stop
2719	2911e62917e8	Bring up $1 - \ell + m$ || Store in 2917

271a	2917e6290bea	Bring up $1 - \ell + m$ || Subtract "4"
271b	2917e82905dc	Store in 2917 || Compare with zero
271c	2712e2270ae2	If < 0 go to 2712 || If = 0 go to 270A
271d	271ae20000e0	If > 0 go to 271A || *
271e	2913e6291be8	Bring up 2913 || Store in 291B
271f	2905e6291de8	Bring up zero || Store in 291D
2720	291de62907e9	Bring up 291D || Add "1"
2721	291de82903dc	Store in 291D || Compare 2903
2722	2724e22724e2	Not finished: Go to 2724
2723	2729e20000e0	Finished: Go to 2729 || *
2724	291be60000e0	Bring up 291B || *
2725	3b00e80000f8	Store in 3B00 || Form neg. of this number
2726	291be80000e0	Store in 291B || *
2727	0002d0272500	Increment address 3B00 by 2 ||
2728	2720e20000e0	Go to 2720 ||
2729	291fe62907dc	Bring up "0" (even) or "1" (odd) || Compare with +1
272a	272ce22737e2	If < 0, even: Go to 272C || If = 0, odd: Go to 2737
272b	272be5000082	If > 0, error: Go to stop
272c	2905e62921e8	
272d	2921e628ffe9	
272e	2907e92925e8	
272f	2a50df292500	Compute $(-1)^{n+m+1}$
2730	2907dc2733e2	
2731	2735e20000e0	
2732	2732e5000082	
2733	2907e6292be8	
2734	2739e20000e0	
2735	2907e7292be8	
2736	2739e20000e0	
2737	2907e62921e8	
2738	272de20000e0	

UNCLASSIFIED

2739	291fe62907dc
273a	273ee2273be2
273b	2753d8274900
273c	2750e20000e0
273d	273de5000082
273e	2905e6292de8
273f	292fe828ffe6
2740	2909eb2931e8
2741	2753d8274900
2742	292de62931e9
2743	2933e80000e0
2744	2a34df293300
2745	2935e80000e0
2746	2a34df292d00
2747	2937e82935e6
2748	2937ec0000e0
2749	0000e8292de6
274a	2909e9292de8
274b	0002d0274900
274c	292fe62907e9
274d	292fe82903dc
274e	2742e22742e2
274f	2755e20000e0
2750	2907e6292de8
2751	2905e6292fe8
2752	273fe30000e0
2653	2d0000000000
2754	000000275510

Compute $\frac{(n + 2m)!}{n!}$

2755	28fde628ffea	Compute $\frac{(\ell - m)!}{(\ell + m)!}$
2756	2939e828fde6	
2757	28ffe9293be8	
2758	2a34df293900	
2759	293de80000e0	
275a	2a34df293b00	
275b	293fe8293de6	
275c	293fec2941e8	|| Store $(\ell - m)!/(\ell + m)!$ in 2941
275d	2a2ae62a2aeb	Compute $\left(\frac{\xi^2 - 1}{\xi^2}\right)^{m/2}$
275e	2943e82907ea	
275f	2945e82945e6	
2760	2943ec2947e8	
2761	0000f90000fa	
2762	2949e82763e4	
2763	28ff50000041	
2764	2764502a6310	
2765	276820276610	
2766	276650020011	
2767	2949e60000e0	
2768	0000ef0000e0	
2769	294be82941eb	Store $(z^2 - 1/z^2)^{m/2}$ in 294B ||
276a	294de80000e0	Store $(\ell - m)!/(\ell + m)! \times [(z^2 - 1)/z^2]^{m/2}$ in 294D
276b	291fe62907dc	Bring up zero (even) or "1" (odd) || Compare with "1"
276c	276ee227a0e2	If < 0 even: Go to 276E || If = 0 odd go to 27A0
276d	276de5000082	If > 0 error stop

UNCLASSIFIED

276e	28ffe6294fe8
276f	2903e62909eb
2770	294fe929a9e8
2771	2f00e63300e6
2772	3700e63900e6
2773	2901e62b86e6
2774	2b87e627b0e6
2775	28bce62798e6
2776	27a8e6277ae6
2777	2774e62775e6
2778	2776e62772e6
2779	286fe627a8e6
277a	2889e627b0e6
277b	289de60000e0
277c	28b1e62779e6
277d	277be62771e6
277e	277ce6277ae6
277f	2a50df294f00
2780	0000e02907dc
2781	2784e22786e2
2782	2783e40000e0
2783	000082278410
2784	2b86d9279400
2785	2788e20000e0
2786	2b87d9279400
2787	2788e20000e0
2788	2775d7279700
2789	278ae40000e0
278a	294f50000041
278b	278b502a6310
278c	279520278d10

(ℓ - m even)

Compute j_{n+m} and store in 2F00, 2F02, etc.

278d	29a950000041	
278e	278e502a6310	
278f	279521279010	
2790	279050020011	
2791	2773d8279600	
2792	2771d9279600	
2793	2794e40000e0	
2794	279450000010	
2795	000000000000	
2796	000000000000	
2797	000010000000	Program fixes exit to 2798
2798	279850020011	
2799	28ffe62907e9	(ℓ - m even)
279a	294fe82903e6	Compute j_{n+m+1} and store in 3700, 3702
279b	2909eb294fe9	
279c	29a9e80000e0	
279d	277ed7278800	Program fixes exit to 27B0
279e	2778d7279200	
279f	277fe20000e0	
27a0	2907e628ffe9	
27a1	294fe82903e6	
27a2	2909eb294fe9	(ℓ - m odd)
27a3	29a9e80000e0	Compute j_{n+m+1} and store in 2F00, 2F02, etc.
27a4	277cd7278800	
27a5	277dd7279200	
27a6	277fe20000e0	
27a7	0000e00000e0	

UNCLASSIFIED

27a8	27a850020011	
27a9	28ffe62909e9	(ℓ - m odd)
27aa	294fe82903e6	Compute j_{n+m+2} and store in 3700, 3702, etc,
27ab	2909eb294fe9	
27ac	29a9e80000e0	
27ad	277ed7278800	
27ae	2778d7279200	
27af	277fe20000e0	
27b0	27b050020011	Enter floating point
27b1	291fe62907dc	
27b2	27b4e227cae4	
27b3	27b3e5000082	
27b4	28ffe62907ea	
27b5	2955e82905dc	
27b6	27b8e227bae4	
27b7	27bae40000e0	
27b8	2905e62953e8	
27b9	27c1e20000e0	
27ba	295550000041	
27bb	27bb502a6310	
27bc	27bd2127bd10	
27bd	290150000041	
27be	27be502b8810	
27bf	27bf50020011	
27c0	2953e80000e0	
27c1	28ffe62907e9	
27c2	295be827c3e4	
27c3	295b50000041	
27c4	27c4502a6310	
27c5	27c62127c610	
27c6	290150000041	
27c7	27c7502b8810	
27c8	27c850020011	

27c9	2965e827dce2	
27ca	28ff50000041	
27cb	27cb502a6310	
27cc	27cd2127cd10	
27cd	290150000041	
27ce	27ce502b8810	
27cf	27cf50020011	
27d0	2953e80000e0	
27d1	28ffe62907e9	
27d2	295be8295be6	
27d3	2907e92967e8	
27d4	27d5e40000e0	
27d5	296750000041	
27d6	27d6502a6310	
27d7	27d82127d810	
27d8	290150000041	
27d9	27d9502b8810	
27da	27da50020011	
27db	2965e80000e0	
27dc	28ffe62907e9	
27dd	295be82907e9	
27de	2967e82907e9	
27df	2969e828ffe7	
27e0	296be82907ea	
27e1	2961e82907ea	
27e2	296de82907ea	Compute starting values of j_x, n_x to assist calculation of $\frac{dj_x}{dz}$, $\frac{dn_x}{dz}$
27e3	296fe80000e0	
27e4	291fe62907dc	
27e5	27e7e42805e4	
27e6	27e6e5000082	
27e7	28ff50000041	
27e8	27e8502a6310	
27e9	27ec2027ea10	

UNCLASSIFIED

27ea	27ea50020011
27eb	2907e70000e0
27ec	0000ef2957e8
27ed	27eee40000e0
27ee	296750000041
27ef	27ef502a6310
27f0	27f32027f110
27f1	27f150020011
27f2	2907e70000e0
27f3	0000ef2971e8
27f4	27f5e40000e0
27f5	296b50000041
27f6	27f6502a6310
27f7	27f82127f810
27f8	290150000041
27f9	27f9502b8810
27fa	27fa50020011
27fb	2957eb2973e8
27fc	27fde40000e0
27fd	296d50000041
27fe	27fe502a6310
27ff	280021280010
2800	290150000041
2801	2801502b8810
2802	280250020011
2803	2971eb2975e8
2804	2820e20000e0
2805	295b50000041
2806	2806502a6310
2807	280d20280810
2808	296950000041
2809	2809502a6310
280a	280f20280b10

280b	280b50020011
280c	2907e70000e0
280d	0000ef295fe8
280e	2907e70000e0
280f	0000ef2977e8
2810	2811e40000e0
2811	296150000041
2812	2812502a6310
2813	281421281410
2814	290150000041
2815	2815502b8810
2816	281650020011
2817	295feb2973e8
2818	2819e40000e0
2819	296f50000041
281a	281a502a6310
281b	281c21281c10
281c	290150000041
281d	281d502b8810
281e	281e50020011
281f	2977eb2975e8
2820	2953e6297de8
2821	2965e62981e8
2822	28ffe62983e8
2823	2907e92987e8
2824	2909ea2989e8
2825	28ffe62909eb
2826	2907e92985e8
2827	2905e62979e8
2828	291fe62907dc
2829	282ae22842e2
282a	2905e6298be8
282b	0002d0283f00

Compute $\frac{d}{dz}(j_x)$

UNCLASSIFIED

282c	2979e62903dc	
282d	282fe2282fe2	
282e	2845e20000e0	
282f	2983e6298be9	
2830	2953e82987e6	
2831	298be92965e8	
2832	298be62909eb	
2833	2985e92989e8	
2834	2953e62989ec	
2835	297deb298de8	
2836	2965e62989ec	
2837	2981eb298fe8	
2838	298de6298fea	
2839	3100e80000e0	Store j_x in 3100, 3102, etc.
283a	0002d0283900	
283b	2979e62907e9	
283c	2979e8298be6	
283d	2909e9298be8	
283e	2981e6297de8	
283f	3700e62981e8	
2840	0002d0283f00	
2841	282ce20000e0	
2842	0002d0283f00	
2843	2907e6298be8	
2844	282ce20000e0 ←	
2845	291fe62907dc	Bring up zero (even) or one (odd) || Compare with one
2846	2848e22894e2	If < 0 even: Go to 2848 || If = 0, odd: Go to 2894
2847	2847e5000082	If > 0 error stop

2848	28ffe72907ea
2849	2991e80000f9
284a	2993e82903e7
284b	2951e82909eb
284c	2991e929a9e8
284d	28ffe62907e9
284e	295be8284fe4
284f	295b50000041
2850	2850502a6310
2851	285420285210
2852	285250020011
2853	2907e70000e0
2854	0000ef29ade8
2855	2a50df299300
2856	0000e02907dc
2857	2859e2285be2
2858	2783e40000e0
2859	2b86d9286b00
285a	285de20000e0
285b	2b87d9286b00
285c	285de20000e0
285d	2779d8286e00
285e	285fe40000e0
285f	299150000041
2860	2860502a6310
2861	286c20286210
2862	29a950000041
2863	2863502a6310
2864	286c21286510
2865	286550020011
2866	2773d8286d00
2867	2771d7286a00
2868	286ad9286d00

(ℓ - m even)
Compute n_{n+m} ($= j_{-m-1}$) and store in 3300, 3302, etc.

UNCLASSIFIED

2869	286be40000e0
286a	000000286b10
286b	286b50000010
286c	000000000000
286d	000000000000
286e	000010000000
286f	286f50020011
2870	2905e629a3e8
2871	29a3e62907ea
2872	29a3e82951dc
2873	287ae22875e2
2874	2875e20000e0
2875	3300e629adeb
2876	3300e80000e0
2877	0002d0287500
2878	0002d0287600
2879	2871e20000e0
287a	28ffe72909ea
287b	2991e80000f9
287c	2993e82903e7
287d	2951e82909eb
287e	2991e929a9e8
287f	2776d7285d00
2880	2778d7286700
2881	2882e40000e0
2882	28ff50000041
2883	2883502a6310
2884	288720288510
2885	288550020011
2886	2907e70000e0
2887	0000ef29ade8
2888	2855e20000e0
2889	288950020011

(ℓ - m even)
Compute n_{n+m+1} (= j_{-n-m-2}) and store in 3900, 3902, etc.

288a	2905e629a3e8
288b	29a3e62907ea
288c	29a3e82951dc
288d	28bce2288fe2
288e	288fe20000e0
288f	3900e629adeb
2890	3900e80000e0
2891	0002d0288f00
2892	0002d0289000
2893	288be20000e0
2894	28ffe72909ea
2895	2907ea2991e8
2896	0000f92993e8
2897	2903e72951e8
2898	2909eb2991e9
2899	29a9e80000e0
289a	277dd8285d00
289b	2778d7286700
289c	284de20000e0

(ℓ - m odd)
Compute n_{n+m+2} ($= j_{-m-3-n}$) and store in 3900, 3902, etc.

UNCLASSIFIED

289d	289d50020011
289e	2905e629a3e8
289f	29a3e62907ea
28a0	29a3e82951dc
28a1	28a8e228a3e2
28a2	28a3e20000e0
28a3	3900e629adeb
28a4	3900e80000e0
28a5	0002d028a300
28a6	0002d028a400
28a7	289fe20000e0
28a8	28ffe72909ea
28a9	2991e80000f9
28aa	2993e82903e7
28ab	2951e82909eb
28ac	2991e929a9e8
28ad	277ed8285d00
28ae	277dd7286700
28af	2882e40000e0
28b0	0000e00000e0
28b1	28b150020011
28b2	2905e629a3e8
28b3	29a3e62907ea
28b4	29a3e82951dc
28b5	28bce228b7e2
28b6	28b7e20000e0
28b7	3300e629adeb
28b8	3300e80000e0
28b9	0002d028b700
28ba	0002d028b800
28bb	28b3e20000e0

(ℓ - m odd)
Compute n_{n+m+1} ($= j_{-m-2-n}$) and store in 3300, 3302, etc.

28bc	2973e62995e8	
28bd	2975e62999e8	
28be	28ffe6299be8	
28bf	2907e9299de8	
28c0	2909ea299fe8	
28c1	28ffe62909eb	
28c2	2907e929ale8	
28c3	2905e629a3e8	
28c4	291fe62907dc	
28c5	28c7e228dfe2	
28c6	28c6e5000082	
28c7	2905e6298be8	
28c8	0002d028dc00	
28c9	29a3e62903dc	Compute $\frac{d}{dz}(n_x)$
28ca	28cce228cce2	
28cb	28e2e20000e0	
28cc	299be6298be9	
28cd	2913e8299de6	
28ce	298be92921e8	
28cf	298be62909eb	
28d0	29ale92929e8	
28d1	2913e62929ec	
28d2	2995eb29a5e8	
28d3	2921e62929ec	
28d4	2999eb29a7e8	
28d5	29a5e629a7ea	
28d6	3500e80000e0	Store n_x' in 3500, 3502, etc.
28d7	0002d028d600	
28d8	29a3e62907e9	
28d9	29a3e8298be6	
28da	2909e9298be8	
28db	2999e62995e8	
28dc	3900e62999e8	

UNCLASSIFIED

28dd	0002d028dc00		
28de	28c9e20000e0		
28df	0002d028dc00		
28e0	2907e6298be8		
28e1	28c9e20000e0		
28e2	2905e629a9e8		
28e3	29abe80000e0		
28e4	3b00e60000e0		
28e5	2d00e60000e0		
28e6	2f00e60000e0		
28e7	3100e60000e0		
28e8	3300e60000e0		
28e9	3500e60000e0		
28ea	3700e60000e0		
28eb	3900e60000e0		
28ec	28e4d828f100		
28ed	29a9e62907e9	Print-out of intermediate results	
28ee	29a9e82903dc		
28ef	28f1e228f1e2		
28f0	28f5e20000e0		
28f1	0000e60000e0		
28f2	0000e00000e0		
28f3	0002d028f100		
28f4	28ede20000e0		
28f5	2905e629a9e8		
28f6	0001d028ec00		
28f7	29abe62907e9		
28f8	29abe8290de6		
28f9	29abea28fbf4		|| Finished? No: Go to 28FB
28fa	29b0e20000e0	Yes: Go to 29B0	|| *
28fb	0025f30029f3		
28fc	28ece20000e0		

28fd	080000000000	ℓ — Read-in (data of problem)
28fe	000100000000	
28ff	000000000000	m — Read-in (data of problem)
2900	000000000000	
2901	080000000000	$h\xi$ — Read-in (data of problem)
2902	000200000000	
2903	0a0000000000	Number of terms in series to be computed
2904	000400000000	
2905	000000000000	
2906	000000000000	
2907	080000000000	"1"
2908	000100000000	
2909	080000000000	
290a	000200000000	
290b	080000000000	"4"
290c	000300000000	
290d	070000000000	
290e	000400000000	
290f	000000000000	
2910	000000000000	
2911	000000000000	
2912	000000000000	
2913	000000000000	
2914	000000000000	
2915	000000000000	
2916	000000000000	
2917	000000000000	
2918	000000000000	
2919	000000000000	
291a	000000000000	
291b	000000000000	
291c	000000000000	
291d	000000000000	

UNCLASSIFIED

291e	000000000000
291f	000000000000
2920	000000000000
2921	000000000000
2922	000000000000
2923	000000000000
2924	000000000000
2925	000000000000
2926	000000000000
2927	000000000000
2928	000000000000
2929	000000000000
292a	000000000000
292b	000000000000
292c	000000000000
292d	000000000000
292e	000000000000
292f	000000000000
2930	000000000000
2931	000000000000
2932	000000000000
2933	000000000000
2934	000000000000
2935	000000000000
2936	000000000000
2937	000000000000
2938	000000000000
2939	000000000000
293a	000000000000
293b	000000000000
293c	000000000000
293d	000000000000
293e	000000000000

293f	000000000000
2940	000000000000
2941	000000000000
2942	000000000000
2943	000000000000
2944	000000000000
2945	000000000000
2946	000000000000
2947	000000000000
2948	000000000000
2949	000000000000
294a	000000000000
294b	000000000000
294c	000000000000
294d	000000000000
294e	000000000000
294f	000000000000
2950	000000000000
2951	000000000000
2952	000000000000
2953	000000000000
2954	000000000000
2955	000000000000
2956	000000000000
2957	000000000000
2958	000000000000
2959	000000000000
295a	000000000000
295b	000000000000
295c	000000000000
295d	000000000000
295e	000000000000
295f	000000000000

UNCLASSIFIED

2960	000000000000
2961	000000000000
2962	000000000000
2963	000000000000
2964	000000000000
2965	000000000000
2966	000000000000
2967	000000000000
2968	000000000000
2969	000000000000
296a	000000000000
296b	000000000000
296c	000000000000
296d	000000000000
296e	000000000000
296f	000000000000
2970	000000000000
2971	000000000000
2972	000000000000
2973	000000000000
2974	000000000000
2975	000000000000
2976	000000000000
2977	000000000000
2978	000000000000
2979	000000000000
297a	000000000000
297b	000000000000
297c	000000000000
297d	000000000000
297e	000000000000
297f	000000000000
2980	000000000000

2981	000000000000
2982	000000000000
2983	000000000000
2984	000000000000
2985	000000000000
2986	000000000000
2987	000000000000
2988	000000000000
2989	000000000000
298a	000000000000
298b	000000000000
298c	000000000000
298d	000000000000
298e	000000000000
298f	000000000000
2990	000000000000
2991	000000000000
2992	000000000000
2993	000000000000
2994	000000000000
2995	000000000000
2996	000000000000
2997	000000000000
2998	000000000000
2999	000000000000
299a	000000000000
299b	000000000000
299c	000000000000
299d	000000000000
299e	000000000000
299f	000000000000
29a0	000000000000
29a1	000000000000

UNCLASSIFIED

29a2	000000000000	
29a3	000000000000	
29a4	000000000000	
29a5	000000000000	
29a6	000000000000	
29a7	000000000000	
29a8	000000000000	
29a9	000000000000	
29aa	000000000000	
29ab	000000000000	
29ac	000000000000	
29ad	000000000000	
29ae	000000000000	
29af	00000029b010	
29b0	2905e6291be8	Bring up zero ‖ Store zero in 291B
29b1	292de8292fe8	~ in 292D ‖ ~ in 292F
29b2	2933e82929e8	~ in 2933 ‖ ~ in 2929
29b3	2929e62903dc	Bring up 2929 (counter) ‖ Compare with no. of terms in series
29b4	29b6e229d5e2	Finished? No: Go to 29B6 ‖ No: Go to 29B6 (left)
29b5	29d5e20000e0	Yes: Go to 29D5
29b6	3b00e62d00eb	Bring up $i^{n+m-\ell}$ ‖ Multiply by $(n + 2m)!/n!$
29b7	16a0eb2a32e8	Multiply by d_n ‖ Store $(i^{n+m-\ell})[(n + 2m)!/n!][d_n]$ in 2A32
29b8	2f00eb0400e8	Multiply by j_{n+m} ‖ Store in 0400
29b9	0002d129b800	Change address 0400 (in 2698) to 0400 + 2x
29ba	29b8d729bc00	Place address (0400 + 2x) in left of 28BC
29bb	2907e60000e0	Bring up "one" ‖ *
29bc	0000e82a32e6	Store "1" in 0400 + 2x ‖ Bring up 2A32
29bd	3100eb0600e8	Compute all terms in series ‖ Multiply by $[dj_x/dz]$ ‖ Store in 0600
29be	0002d129bd00	$\sum_n' i^{n+m-\ell} \frac{(n + 2m)!}{n!} d_n f_n(hz)$ ‖ Change address 0600 (in 29BD) to 0600 + 2x
29bf	29bdd729c100	$f_n(hz) = j_n$ (in 0400) or $= j_n'$ (in 0600) ‖ Place address (0600 + 2x) in left of 29C1
29c0	2907e60000e0	or $= n_n$ (in 0800) or $= n_n'$ (in 1000) ‖ Bring up "1" ‖ *
29c1	0000e82a32e6	Store "1" in 0600 + 2x ‖ Store in 2A32
29c2	3300eb0800e8	Multiply by n_{n+m} ‖ Store in 0800

29c3	2c60e20000e0	Change address 0800 (in 29C2) to 0800 + 2x
29c4	29c2d729c600	Place address (0800 + 2x) in left of 29C6
29c5	2907e60000e0	Bring up "1" || *
29c6	0000e82a32e6	Store "1" in (0800 + 2x) || Bring up 2A32
29c7	3500eb1000e8	Multiply by $[dn_{n+m}/dz]$ || Store in 1000
29c8	2c65e20000e0	Change address 1000 (in 29C7) to (1000 + 2x)
29c9	29c7d729cb00	Place address (1000 + 2x) in left of 29CB
29ca	2907e60000e0	Bring up "1" || *
29cb	0000e80000e0	Store "1" in (1000 + 2x) || *
29cc	0002d029b600	Change 3B00 (in 29B6) to (3B00 + 2x)
29cd	0002d129b600	Change 2D00 (in 29B6) to (2D00 + 2x)
29ce	0002d029b700	Change 16A0 (in 29B7) to (16A0 + 2x)
29cf	0002d029b800	Change 2F00 (in 29B8) to (2F00 + 2x)
29d0	0002d029bd00	Change 3100 (in 29BD) to (3100 + 2x)
29d1	0002d029c200	Change 3300 (in 29C2) to (3300 + 2x)
29d2	0002d029c700	Change 3500 (in 29C7) to (3500 + 2x)
29d3	2929e62907e9	Bring up 2929 (counter) || Add "1"
29d4	2929e829b3e2	Store in 2929 || Go to 29B3

29d5	2c53e20000e0	
29d6	29d8e229dde2	
29d7	29d8e20000e0	
29d8	291be60400e9	
29d9	291be80000e0	Compute $R^{(1)}_{m\ell}$
29da	0000e00000e0	
29db	0002d129d800	
29dc	2c58e20000e0	
29dd	291be6294deb	
29de	291be80025f3	Store $R^{(1)}_{m\ell}$ in 291B || Carriage return (C.R.) on typing
29df	0000f12c5ce2	Print $R^{(1)}_{m\ell}$

UNCLASSIFIED

29e0	0600e62907dc	
29e1	29e3e229e8e2	
29e2	29e3e20000e0	
29e3	292de60600e9	
29e4	292de80000e0	Compute $\frac{dR_{m\ell}^{(1)}(h, z)}{dz}$
29e5	0000e00000e0	
29e6	0002d129e300	
29e7	2c58e20000e0	
29e8	292de6294deb	
29e9	2a2ceb292de8	|| Store $R_{m\ell}^{(1)}{}'$ in 2920
29ea	2c4ae20000e0	Go to 2C4A || *
29eb	0800e62907dc	
29ec	29eee229f7e2	
29ed	29eee20000e0	
29ee	2951e8292fe9	
29ef	292fe80000e0	Store $R_{m\ell}^{(2)}(h, z)$ in 292F || *
29f0	1200e80000e0	
29f1	0002d029f000	Compute $R_{m\ell}^{(2)}(h, z)$ (save last term of series in 2951)
29f2	29f0d829f400	
29f3	2907e60000e0	
29f4	0000e80000e0	
29f5	0002d029eb00	
29f6	29ebe20000e0	
29f7	2c6ae20000e0	
29f8	292fe80025f3	|| C.R.
29f9	0000f12951e6	Print $R_{m\ell}^{(2)}$ ||
29fa	294deb0000e0	Provision for typing out 2951
29fb	0000e00000e0	

29fc	1200e62907dc	
29fd	29ffe22a04e2	
29fe	29ffe20000e0	Provision for typing out all terms in series $R^{(2)}_{m\ell}$
29ff	1200e6294deb	
2a00	0000e00000e0	C.R. || Print out all terms in $R^{(2)}_{m\ell}$
2a01	0002d029ff00	
2a02	0002d029fc00	
2a03	29fce20000e0	
2a04	1000e62907dc	
2a05	2a07e22a10e2	
2a06	2a07e20000e0	
2a07	2951e82933e9	
2a08	2933e82c52e2	Store $[dR^{(2)}_{m\ell}/dz]$ in 2933 || Go to 2C52
2a09	1200e80000e0	
2a0a	0002d02a0900	
2a0b	2a09d82a0d00	
2a0c	2907e60000e0	Compute $\left[d\,\frac{R^{(2)}_{m\ell}}{dz}\right]$
2a0d	0000e80000e0	(Same last term of series in 2951)
2a0e	0002d02a0400	
2a0f	2a04e20000e0	
2a10	2c73e20000e0	
2a11	2a2ceb2933e8	
2a12	0025f30000f1	C.R. || Print out $[dR^{(2)}_{m\ell}/dz]$
2a13	2951e6294deb	
2a14	0000e00000e0	Provision for print-out last term in series of $[dR^{(2)}_{m\ell}dz]$
2a15	1200e62907dc	
2a16	2a18e22a1ee2	
2a17	2a18e20000e0	
2a18	1200e6294deb	Provision for print-out of all terms in series $[dR^{(2)}_{m\ell}/dz]$
2a19	2a2ceb0000e0	
2a1a	0000e00000e0	
2a1b	0002d02a1500	
2a1c	0002d02a1800	
2a1d	2a15e20000e0	

2a1e	292fe6292deb	
2a1f	2a2ee8291be6	
2a20	2933eb2a2eea	
2a21	0000e02a2ee8	Compute Wronskian and theoretical Wronskian
2a22	0025f30000f1	C.R. || Print out computed Wronskian
2a23	2a2ae62a2aeb	
2a24	2907ea2a2ceb	
2a25	2a30e82907e6	
2a26	2a30ec2a30e8	
2a27	0025f30000f1	C.R. || Print out theoretical Wronskian
2a28	0025f32a29e4	
2a29	0000822a2a10	Program finished: Stop
2a2a	080000000000	ξ
2a2b	000200000000	Read-in (data of problem)
2a2c	080000000000	h
2a2d	000100000000	
2a2e	000000000000	
2a2f	000000000000	
2a30	000000000000	
2a31	000000000000	
2a32	000000000000	
2a33	000000000000	

2a34	2a4d432a4320
2a35	2a43222a4d50
2a36	2a39202a3710
2a37	0202812a4e02
2a38	2a3850020011
2a39	0000e62a45dc
2a3a	2a43e52a44e2
2a3b	2a49e82a47ea
2a3c	2a4be82a45dc
2a3d	2a40e32a40e3
2a3e	2a49e62a4beb
2a3f	2a49e82a4be6
2a40	2a3be32a49e6
2a41	2a42e40000e0
2a42	2a4e81020202
2a43	000010fbad82
2a44	2a47e62a42e4
2a45	000000000000
2a46	000000000000
2a47	080000000000
2a48	000100000000
2a49	000000000000
2a4a	000000000000
2a4b	000000000000
2a4c	000000000000
2a4d	000000000000
2a4e	000000000000
2a4f	000000000000

Subroutine for determining factorial of an integer [= (x!)]

UNCLASSIFIED

2a50	2a4d432a5c20
2a51	2a5c222a4d50
2a52	2a57202a5521
2a53	2a55232a5410
2a54	0202812a4e02
2a55	2a5d50000055
2a56	2a57212a5710
2a57	000050000031
2a58	2a5e26000080
2a59	2a61142a5a10
2a5a	2a5a50020011
2a5b	2a45e62a5fe4
2a5c	000010000000
2a5d	000c00000000
2a5e	000100000000
2a5f	2a4e81020202
2a60	2a5c102a6110
2a61	2a6150020011
2a62	2a47e62a5fe4
2a63	2a6b202a5e55
2a64	2a65202a6510
2a65	0000502a6820
2a66	2a69212a6822
2a67	2a6b222a6810
2a68	0000512a5d54
2a69	2a6a20000050
2a6a	0000312a6b10
2a6b	000010000000

Subroutine for determining whether an integer is odd or even.

Subroutine for converting floating point numbers to decimal.

2a6c	2b33202b3322
2a6d	2a7c202b3450
2a6e	2ac7202a7510
2a6f	2b33202b3322
2a70	2a7c202b3550
2a71	2ac7202a7510
2a72	2b33202b3322
2a73	2a7c202b3650
2a74	2ac7202a7510
2a75	2a7550020010
2a76	2b3ce62b62e8
2a77	2b3ee62b64e8
2a78	2b40e62b66e8
2a79	2a7ae40000e0
2a7a	2b37502b5442
2a7b	2b38502b5542
2a7c	0000502b5642
2a7d	2b33222a7e20
2a7e	0000502b5742
2a7f	2a80202b3322
2a80	0000812b6802
2a81	2c4a10002031
2a82	2b5b422b5b51
2a83	2b5a422b5951
2a84	2b59422aca10
2a85	0020312b5a42
2a86	2b5955002038
2a87	2b5c422b5c54
2a88	2b58542b7e42
2a89	2b4f502b2b42
2a8a	2b50502b1442
2a8b	2a8b50020010
2a8c	2b68e60000f9

UNCLASSIFIED

2a8d	2b66ea2a8ff4
2a8e	2a93e50000e0
2a8f	2b66e62b40e9
2a90	2b66e82a91e4
2a91	2b54502b3754
2a92	2b54422a9310
2a93	2a8b102b5450
2a94	2b5a552a9712
2a95	2b54502b3754
2a96	2b54422a9410
2a97	2b3a502b5454
2a98	2b54422b5d42
2a99	2b54512b5f42
2a9a	2b5a512b6042
2a9b	2b7e502ab720
2a9c	2abc202a9d10
2a9d	2a9d50020010
2a9e	2b5de60000e1
2a9f	2b6ae82b42e6
2aa0	2b6ce82b3ce6
2aa1	2b6ee82b6ae6
2aa2	2b44eb2b46e9
2aa3	2b68ec2b6ceb
2aa4	2b6eea2b7ae8
2aa5	2b6ce62b6ee8
2aa6	2b7ae62b6ce8
2aa7	2aa8e40000e0
2aa8	2b3b502b5f54
2aa9	2b5f422b6055
2aaa	2ab0102aab10
2aab	2aab50020010
2aac	2b6ae62b46ea
2aad	2b6ae82b44eb

Subroutine for computing all odd-subscript, or all even-subscript, or all subscript spherical Bessel functions of positive or negative orders lying between specified upper and lower limits.

2aae	2b46e92b68ec
2aaf	2b6ceb2aa4e2
2ab0	2b61422ab212
2ab1	2aab102ab210
2ab2	2b5f512b5955
2ab3	2ab5122b5f50
2ab4	2abb122aab10
2ab5	2ab550020010
2ab6	2b7ae60000e0
2ab7	0000e82ab8e4
2ab8	2ab7502b3955
2ab9	2ab7202b5f50
2aba	2abb122aab10
2abb	2abb50020010
2abc	0000e62b7ce8
2abd	2b48e62b68ea
2abe	0000fc2b68ec
2abf	2b7aec2b70e8
2ac0	2b7ceb2b72e8
2ac1	2b62ea2b72ec
2ac2	0000f92b4aea
2ac3	2ac8f42ac4e4
2ac4	2b7e502b3954
2ac5	2ac6212ac610
2ac6	2b5181000003
2ac7	0000102ac810
2ac8	2b72e62b62e8
2ac9	2a95e40000e0
2aca	2aca50020010
2acb	2b68e60000fc
2acc	2b68ec2b76e8
2acd	2acee40000e0
2ace	2b58502ad620

UNCLASSIFIED

2acf	2aeb202b0a20
2ad0	2b4c502b1442
2ad1	2b4e502b2b42
2ad2	2b55502b5955
2ad3	2ad4122adf10
2ad4	2ad450020010
2ad5	2b76e60000e0
2ad6	0000e82ad7e4
2ad7	2ad6222ad622
2ad8	2aeb222aeb22
2ad9	2b0a222b0a22
2ada	2b55502b5a55
2adb	2adc122adf10
2adc	2ad6502add21
2add	2b5181000003
2ade	2ac7102adf10
2adf	2adf50020010
2ae0	2b76e62b68ec
2ae1	2b74e82b48e6
2ae2	2b68ea0000fc
2ae3	2b68ec2b74e9
2ae4	0000f82b74e8
2ae5	2ae6e40000e0
2ae6	2b55502b3b54
2ae7	2b55422b5955
2ae8	2ae9122af310
2ae9	2ae950020010
2aea	2b74e60000e0
2aeb	0000e82aece4
2aec	2aeb222aeb22
2aed	2b0a222b0a22
2aee	2b55502b5a55
2aef	2af0122af310

2af0	2aeb502af121
2af1	2b5181000003
2af2	2ac7102af310
2af3	2af350020010
2af4	2b44e62b64eb
2af5	2b3ee92b68ec
2af6	2b74eb2b76e9
2af7	2b78e82af8e4
2af8	2b55502b3b54
2af9	2b55422b5955
2afa	2b00122afb10
2afb	2afb50020010
2afc	2b74e62b76e8
2afd	2b78e72b74e8
2afe	2b64e62b3ee9
2aff	2b64e82af4e2
2b00	2b0050020010
2b01	2b78e62b7ae8
2b02	2b03e40000e0
2b03	2b5550000041
2b04	000131000130
2b05	2b08142b0610
2b06	2b0650020010
2b07	2b7ae62b0ae2
2b08	2b0850020010
2b09	2b7ae72b7ae8
2b0a	0000e82b0be4
2b0b	2b0a222b0a22
2b0c	2b55502b5a55
2b0d	2b0e122afb10
2b0e	2b0a502b0f21
2b0f	2b5181000003
2b10	2ac7102b1110

UNCLASSIFIED

2b11	2b58502b1420
2b12	2b83502b1520
2b13	2b1350020010
2b14	0000e62b70eb
2b15	0000e82b16e4
2b16	2b14502b3954
2b17	2b14202b1a20
2b18	2b15502b3954
2b19	2b15202b1a10
2b1a	0000502b5124
2b1b	2b13182b3310
2b1c	2b83502b2c20
2b1d	2b59502b3826
2b1e	0000802b2114
2b1f	2b58502b3954
2b20	2b2b202b2a10
2b21	2b58502b2b20
2b22	2b2a102b2310
2b23	2b83502b2c20
2b24	2b59502b3826
2b25	0000802b2814
2b26	2b58502b2b20
2b27	2b2a102b2810
2b28	2b58502b3954
2b29	2b2b202b2a10
2b2a	2b2a50020010
2b2b	0000e62b70eb
2b2c	0000e82b2de4
2b2d	2b2b502b3954
2b2e	2b39542b2b20
2b2f	2b31202b2c50
2b30	2b39542b2c20
2b31	0000502b5124

2b32	2b2a182b3310
2b33	0000102b3410
2b34	2b1100000000
2b35	2b1c00000000
2b36	2b2300000000
2b37	000000000008
2b38	000000000001
2b39	000200000000
2b3a	000000000005
2b3b	000000000001
2b3c	000000000000
2b3d	000000000000
2b3e	080000000000
2b3f	000100000000
2b40	080000000000
2b41	000400000000
2b42	080000000000
2b43	1fc000000000
2b44	080000000000
2b45	000200000000
2b46	080000000000
2b47	000100000000
2b48	0c90fdaa2217
2b49	000100000000
2b4a	080000000000
2b4b	1ff300000000
2b4c	0000e62b15e2
2b4d	000000000000
2b4e	0000e62b2ce2
2b4f	0000e62b70eb
2b50	0000e62b70eb
2b51	100000000000
2b52	100000000000

UNCLASSIFIED

2b53	100000000000
2b54	000000000008
2b55	000000000001
2b56	000000000000
2b57	000000000000
2b58	000000000000
2b59	000000000000
2b5a	000000000000
2b5b	000000000000
2b5c	000000000000
2b5d	000000000000
2b5e	002c00000000
2b5f	000000000000
2b60	000000000000
2b61	000000000000
2b62	000000000000
2b63	000000000000
2b64	080000000000
2b65	000100000000
2b66	080000000000
2b67	000400000000
2b68	000000000000
2b69	000000000000
2b6a	000000000000
2b6b	000000000000
2b6c	000000000000
2b6d	000000000000
2b6e	000000000000
2b6f	000000000000
2b70	000000000000
2b71	000000000000
2b72	000000000000
2b73	000000000000

2b74	000000000000
2b75	000000000000
2b76	000000000000
2b77	000000000000
2b78	000000000000
2b79	000000000000
2b7a	000000000000
2b7b	000000000000
2b7c	000000000000
2b7d	000000000000
2b7e	000000000000
2b7f	2a6c00000000
2b80	2a7200000000
2b81	2a6f00000000
2b82	040000000000
2b83	000000000000
2b84	000000000000
2b85	2a6c00000000
2b86	2a7200000000
2b87	2a6f00000000
2b88	2bf9202bf922
2b89	2c08552b8a20
2b8a	0000502c0a42
2b8b	2b8c202b8c10
2b8c	0000812c3c02
2b8d	2c0a50001730
2b8e	2c09422c0954
2b8f	2b9212002031
2b90	2c09422c0951
2b91	2c36422bc010
2b92	0020312c3642
2b93	2b9350020010
2b94	2c3ce60000f9

2b95	2c10e82c0eea
2b96	2b97f42b9ae5
2b97	2c0ee62c26e9
2b98	2c0ee82b99e4
2b99	2c0d502c2554
2b9a	2bc3112c0d50
2b9b	2c36552b9e12
2b9c	2c0d502c2554
2b9d	2c0d422b9b10
2b9e	2c12502c0d54
2b9f	2c0d422c0d51
2ba0	2c13422c3651
2ba1	2c14422c0d50
2ba2	2c15422ba310
2ba3	2ba350020010
2ba4	2c15e60000e1
2ba5	2c2ae82c1be6
2ba6	2c3ee82c43e6
2ba7	2c40e82c2ae6
2ba8	0000e00000e0
2ba9	2c00e22c40ea
2baa	2c23e82c3ee6
2bab	2c40e82c23e6
2bac	2c3ee82bade4
2bad	2c42502c1354
2bae	2c13422c1455
2baf	2bc1132bb211
2bb0	2bb050020010
2bb1	2c23e62c28e8
2bb2	2bb2e52c1350
2bb3	2bb7122bb410
2bb4	2bb450020010
2bb5	2c02e30000e0

2bb6	0000e00000e0
2bb7	2bb750020010
2bb8	2c2ee62c3cea
2bb9	0000fc2c3cec
2bba	2c23ec2c28eb
2bbb	2c30e82c32ea
2bbc	2c30ec0000f9
2bbd	2c34ea2bbef5
2bbe	2bc4e32c30e6
2bbf	2c32e82b9ce4
2bc0	2bc050020010
2bc1	2bc8e22c4542
2bc2	2c45512bb012
2bc3	2bb2112c0d42
2bc4	2b93102c30e6
2bc5	2c23e82c43e6
2bc6	2c32e82bf1e2
2bc7	000000000000
2bc8	2c3ce60000fc
2bc9	2c3cec2c1de8
2bca	2bcae52c3750
2bcb	2c36552bcd12
2bcc	2bd010000000
2bcd	2bcd50020010
2bce	2c1de72c23e8
2bcf	2be6e5000000
2bd0	2bd050020010
2bd1	2c3ce60000fc
2bd2	2c3cec2c3cec
2bd3	2c1fe82c2ee6
2bd4	2c3cea0000fc
2bd5	2c3cec2c1fe9
2bd6	0000f82c1fe8

UNCLASSIFIED

2bd7	2bd7e52c3750
2bd8	2c42542c3742
2bd9	2c36552bf911
2bda	2bda50020010
2bdb	2c1fe72c23e8
2bdc	2be6e5000000
2bdd	2bdd50020010
2bde	2c19e62c38eb
2bdf	2c3ae92c3cec
2be0	2c1feb2c1de9
2be1	2c21e82be2e4
2be2	2c37502c4254
2be3	2c37422bfa11
2be4	2be450020010
2be5	2c21e62c23e8
2be6	2be6e52c3750
2be7	000041000131
2be8	0001302c0514
2be9	2be950020010
2bea	2c23e62c23e8
2beb	2bf1e2000000
2bec	2bec50020010
2bed	2c1fe62c1de8
2bee	2c21e72c1fe8
2bef	2c38e62c3ae9
2bf0	2c38e82bdee2
2bf1	2c23e62bf2e4
2bf2	2c0b502c3742
2bf3	2bf610000010
2bf4	2bf450020010
2bf5	2bf1e2000000
2bf6	2bf650020010
2bf7	2c3ae62c38e8

Subroutine for determining spherical Bessel functions of specified order

2bf8	2c23e62bfce2
2bf9	0000102bda12
2bfa	2bdd102c3655
2bfb	2be4122bec10
2bfc	2c26e62c0ee8
2bfd	2c23e62bfee2
2bfe	2bfee52c2550
2bff	2c0d422bf910
2c00	2c19eb2c17e9
2c01	2c3cec2c3eeb
2c02	2ba9e32c2ae6
2c03	2c17ea2c2ae8
2c04	2ba7e3000000
2c05	2c0550020010
2c06	2c23e72c23e8
2c07	2bf1e2000000
2c08	000200000000
2c09	000000000000
2c0a	000000000000
2c0b	000000000001
2c0c	100000000000
2c0d	000000000008
2c0e	080000000000
2c0f	000400000000
2c10	000000000000
2c11	000000000000
2c12	000000000005
2c13	000000000000
2c14	000000000000
2c15	000000000000
2c16	002c00000000
2c17	080000000000
2c18	000100000000

UNCLASSIFIED

2c19	080000000000
2c1a	000200000000
2c1b	080000000000
2c1c	1fc000000000
2c1d	000000000000
2c1e	000000000000
2c1f	000000000000
2c20	000000000000
2c21	000000000000
2c22	000000000000
2c23	000000000000
2c24	000000000000
2c25	000000000008
2c26	080000000000
2c27	000400000000
2c28	000000000000
2c29	000000000000
2c2a	000000000000
2c2b	000000000000
2c2c	000000000000
2c2d	000000000000
2c2e	0c90fdaa2217
2c2f	000100000000
2c30	000000000000
2c31	000000000000
2c32	000000000000
2c33	000000000000
2c34	080000000000
2c35	1ff300000000
2c36	000000000000
2c37	000000000001
2c38	080000000000
2c39	000100000000

2c3a	080000000000	
2c3b	000100000000	
2c3c	000000000000	
2c3d	000000000000	
2c3e	000000000000	
2c3f	000000000000	
2c40	000000000000	
2c41	000000000000	
2c42	000000000001	
2c43	000000000000	
2c44	000000000000	
2c45	000000000000	
2c46	000000000000	
2c47	000000000000	
2c48	28fd322a2a32	Read-in data || Read-in data
2c49	270010000000	Go to 2700 || *
2c4a	2a2ae62a2aeb	Bring up z || Multiply by z
2c4b	2907ea2945e8	Subtract 1 || Store z^2-1 in 2945
2c4c	28ffe62945ec	Bring up m || Divide by z^2-1
2c4d	2a2aec2947e8	Divide by z || Store $m/z(z^2-1)$ in 2947
2c4e	291be62947eb	Bring up $R^{(1)}_{m\ell}$ || Multiply by $m/z(z^2-1)$
2c4f	292de9292de8	Add $R^{(1)}_{m\ell}$,* || Store $R^{(1)}_{m\ell}$, in 292D
2c50	0025f30000f1	C.R. || Print out $R^{(1)}_{m\ell}$,
2c51	29ebe20000e0	Go to 29EB || *
2c52	292fe62947eb	Bring up $R^{(2)}_{m\ell}$ || Multiply by $m/z(z^2-1)$
2c53	2933e92933e8	Add $R^{(2)}_{m\ell}$,* || Store $R^{(2)}_{m\ell}$, in 2933
2c54	0025f30025f3	C.R. || C.R.
2c55	2a09e20000e0	Go to 2A09 || *

UNCLASSIFIED

APPENDIX D
NAREC CODING FOR TAPE NO. 4712D

(Tape No. 4712D is made from No. 4712A by inserting the following changes.)

29EA	2C9C E2 0000 EO	Go to 2C9C || *
2C7C	2CA4 E2 0000 EO	Go to 2CA4 || *
2A1E	2929 E6 292D EB	

2c4a	2b5750001838	Reassign initial storage of output of spherical Bessel functions from subroutine beginning at 2A6C to allow unbounded storage
2c4b	2b83422b8250	
2c4c	2b58422b5650	
2c4d	0001392b5942	
2c4e	2b5954002031	
2c4f	2b59422b5650	
2c50	0017302b5b42	
2c51	2b5b542a8512	
2c52	2a8111000000	
2c53	2905e62929e8	Sum only 1/2 number of terms in the series for $R^{(1)}_{m\ell}$ and $R^{(1)\prime}_{m\ell}$.
2c54	2903e62909ec	
2c55	2c56e82c58e2	
2c56	000000000000	
2c57	000000000000	
2c58	2929e62909e9	
2c59	2929e82c56dc	
2c5a	29d8e229dde2	
2c5b	29dde20000e0	
2c5c	29e1d82c5a00	
2c5d	29e1d72c5b00	
2c5e	2c5bd92c5a00	
2c5f	2c53e20000e0	

2c60	2c92e62c90e8	
2c61	29c2d72c6200	
2c62	0000e62c92e8	
2c63	0002d129c200	
2c64	29c4e20000e0	
2c65	2c96e62c94e8	
2c66	29c7d72c6700	
2c67	0000e62c96e8	
2c68	0002d129c700	
2c69	29c9e20000e0	
2c6a	292fe62c92ea	Bring up $i^{n+m-\ell}\frac{(n+2m)!}{n!}d_n n_n$ || Subtract last term in this finite series
2c6b	2c90ea294deb	Subtract term before last in this finite series || Multiply by $(\ell - m/\ell + m)\ \frac{z^2-1}{z^2}^{m/2}$
2c6c	292fe82c92e6	Store finite sum on $R^{(2)}_{m\ell}$ in 292F || Bring up last term in finite series
2c6d	2c90ec2c98e8	Divide by term before last || Store ratio in 2C98
2c6e	2907e62c98ea	Bring up 1 || Subtract ratio
2c6f	2c98e82c90e6	Store (1 - ratio) in 2C98 || Bring up term before last
2c70	2c98ec294deb	Divide by (1 - ratio) || Multiply by $[(\ell - m)/\ell + m]\,((z^2-1)/z^2)^{m/2}$
2c71	292fe92929e8	Add finite sum of $R^{(2)}_{m\ell}$ || Store infinite sum of $R^{(2)}_{m\ell}$ in 2929
2c72	0025f329f9e2	Bring up $i^{n+m-\ell}\frac{(n+2m)!}{n!}d_n n_n$ || Subtract last term of finite series
2c73	2933e62c96ea	
2c74	2c94ea294deb	Subtract term before last in finite series || Multiply by $[\ell - m/\ell + m]\,(z^2 - 1/z^2)^{m/2}$
2c75	2a2ceb2933e8	Multiply by h || Store finite sum on $R^{(2)\prime}_{m\ell}$ in 2933
2c76	2c96e62c94ec	Bring up last term || Divide by term before last
2c77	2c9ae82907e6	Store ratio in 2C9A || Bring up 1
2c78	2c9aea2c9ae8	Subtract ratio || Store in 2C9A
2c79	2c94e62c9aec	Bring up term before last || Divide by (1 - ratio)
2c7a	294deb2a2ceb	Multiply by $[\ell - m/\ell + m]\,(z^2 - 1/z^2)^{m/2}$ || Multiply by h
2c7b	2933e92933e8	Add finite sum of $R^{(2)}_{m}$, || Store infinite sum of $R^{(2)}_{m\ell}$ in 2933

UNCLASSIFIED

2c7c	0025f30025f3
2c7d	2c90e60025f3
2c7e	0000f12c92e6
2c7f	0025f30000f1
2c80	2c94e60025f3
2c81	0000f12c96e6
2c82	0025f30000f1
2c83	2c98e60025f3
2c84	0000f12c9ae6
2c85	0025f30000f1
2c86	2933e62a12e2
2c87	000000000000
2c88	000000000000
2c89	000000000000
2c8a	000000000000
2c8b	000000000000
2c8c	000000000000
2c8d	000000000000
2c8e	000000000000
2c8f	000000000000
2c90	000000000000
2c91	000000000000
2c92	000000000000
2c93	000000000000
2c94	000000000000
2c95	000000000000
2c96	000000000000
2c97	000000000000
2c98	000000000000
2c99	000000000000
2c9a	000000000000
2c9b	000000000000

Output instructions for punching

(1) Term before last of series for $R^{(2)}_{m\ell}$

(2) Last term of series for $R^{(2)}_{m\ell}$

(3) Term before last of series for $R^{(2)\prime}_{m\ell}$

(4) Last term of series for $R^{(2)\prime}_{m\ell}$

(5) (1 - ratio) for $R^{(2)\prime}_{m\ell}$

(6) (1 - ratio) for $R^{(2)}_{m\ell}$

(7) Infinite sum for $R^{(2)\prime}_{m\ell}$.

2c9c	2a2ae62a2aeb	Bring up z ‖ Multiply by z
2c9d	2907ea2945e8	Subtract 1 ‖ Store in 2945
2c9e	28ffe62945ec	Bring up m ‖ Divide by z^2-1
2c9f	2a2aec2947e8	Divide by z ‖ Store $m/z(z^2-1)$ in 2947
2ca0	291be62947eb	Bring up $R^{(1)}_{m\ell}$ ‖ Multiply by 2947
2ca1	292de9292de8	Add $R^{(1)}_{m\ell}$,* ‖ Store $R^{(1)}_{m\ell}$ in 292D
2ca2	0025f30000f1	C.R. ‖ Print out $R^{(1)}_{m\ell}$,
2ca3	29ebe20000e0	Go to 29EB ‖ *
2ca4	2929e62947eb	Bring up $R^{(2)}_{m\ell}$ ‖ Multiply by $m/z(z^2-1)$
2ca5	2933e92933e8	Add $R^{(2)}_{m\ell}$,* ‖ Store $R^{(2)}_{m\ell}$ in 2933
2ca6	0025f30025f3	C.R. ‖ C.R.
2ca7	2c7de20000e0	Go to 2C7D ‖ *