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

A. L. Van Buren, R. V. Baier, and S. Hanish

Transducer Branch
Acoustics Division

January 20, 1970

NAVAL RESEARCH LABORATORY
Washington, D.C.

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Problem Statement</td>
<td>ii</td>
</tr>
<tr>
<td>Authorization</td>
<td>ii</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>INPUT</td>
<td>2</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>2</td>
</tr>
<tr>
<td>PARAMETER RANGES</td>
<td>3</td>
</tr>
<tr>
<td>ACCURACY CURVES</td>
<td>3</td>
</tr>
<tr>
<td>COMPUTATION TIME</td>
<td>5</td>
</tr>
<tr>
<td>SOLUTION OF THE HELMHOLTZ EQUATION IN OBLATE SPHEROIDAL COORDINATES</td>
<td>5</td>
</tr>
<tr>
<td>DESCRIPTION OF THE COMPUTER PROGRAM OBRAD</td>
<td>10</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>13</td>
</tr>
<tr>
<td>APPENDIX A – Major Calculation Blocks of OBRAD</td>
<td>14</td>
</tr>
<tr>
<td>listed according to statement numbers</td>
<td></td>
</tr>
<tr>
<td>APPENDIX B – Listing in Fortran IV of OBRAD, a Program to Calculate Ob</td>
<td>17</td>
</tr>
<tr>
<td>lace Spheroidal Radial Wave Functions, as compiled for CDC 3800</td>
<td></td>
</tr>
<tr>
<td>APPENDIX C – Sample output from OBRAD</td>
<td>29</td>
</tr>
</tbody>
</table>
ABSTRACT

The Helmholtz scalar wave equation $(\nabla^2 + k^2) \psi = 0$ is separable in oblate spheroidal coordinates $\xi, \phi, \eta$ with solutions $\psi = S(ih, \eta) R(ih, -i\xi) \Phi(\phi)$. The subject of this report is a Fortran computer program called OBRA D which numerically evaluates the radial solutions $R(ih, -i\xi)$. The printed output from OBRA D consists of radial functions of the first and second kind, $R_{m}^{(1),(2)}(ih, -i\xi)$, their first derivatives $\partial R_{m}^{(1),(2)}(ih, -i\xi)/\partial \xi$, the separation constants or eigenvalues $A_{m}^{(1)}(ih)$, and an accuracy check. This report first describes the input data cards and the output format. The theory of the oblate spheroidal wave function is then discussed. A description of the principal internal features of OBRA D is then given. Finally a computer listing of OBRA D is attached as an appendix.

PROBLEM STATUS

This is an interim report on a continuing NRL Problem.

AUTHORIZATION

NRL Problem 501-28
Project RR 102-08-41-5225

Manuscript submitted July 9, 1969.
A FORTRAN COMPUTER PROGRAM FOR CALCULATING THE OBLATE SPHEROIDAL
RADIAL FUNCTIONS OF THE FIRST AND SECOND KIND
AND THEIR FIRST DERIVATIVES

INTRODUCTION

The Helmholtz or scalar wave equation \((\nabla^2 + k^2) \Psi = 0\) is separable in oblate
spheroidal coordinates, with solution \(\Psi = S(\eta) R(\xi) \Phi(\phi)\). The subject of this report is
a Fortran computer program called OBRAD (OBlate RADial) which evaluates the
solutions \(R(\xi)\) in the radial spheroidal coordinate \(\xi\). Although other methods of computing
the radial functions \(R_{ml}(\xi)\) of the first and second kind and their first derivatives are
available, it will be the procedure of this report to obtain them from expansions in terms
of Bessel and Legendre functions.

Oblate spheroidal wave functions of the radial type constitute an essential element
in numerical calculations involving diffraction, radiation, and scattering of acoustic
waves, electromagnetic fields of circular disks and apertures, energy levels of certain
nuclear models, and the resonant behavior of certain spheroidal cavities. An extensive
list of references on calculations and applications of spheroidal wave functions is given
in Ref. 1.

The two independent solutions \(R_{ml}(\xi)\) of the radial equation are characterized
by four parameters: \(\xi\) (called \(\chi\)), \(M\), \(H\), and \(L\). \(M\) is the integer separation constant re-
lating to the solution for the rotational angle \(\phi\). \(H\) is equal to \(kd/2\), where \(d\) is the
interfocal distance, and \(k\) is the propagation constant or wave vector magnitude \(2\pi/k\).
For each choice of \(M\), \(H\), and \(X\) there will be a set of solutions to the radial equation,
each solution characterized by a separation constant or eigenvalue \(\lambda\). It is convenient
to order these eigenvalues in an ascending sequence and label them with integers \(L\), be-
inning with \(L = M\) for the smallest eigenvalue, \(L = M + 1\) for the next one, etc. This
choice is made so that the solutions reduce to that for a spherical coordinate system as
\(H\) approaches zero. In the spherical case the eigenvalues are simply given by \(L (L + 1)\).
For each choice of \(M\), \(H\), \(X\), and \(L\) there will then be two independent radial functions.

Operationally the program OBRAD is divided into several parts. In the first part \(M\)
and \(H\) are set and the eigenvalues are calculated for the desired range of \(L\). In the
second part, \(X\) is chosen, and the expansion functions, Bessel or Legendre, are obtained.
Finally for each choice of \(L\) the expansion constants are obtained and combined with the
expansion functions to give the radial functions and their first derivatives.
INPUT

The input consists of five data cards:

Data Card 1: Format 814. This card contains the integer value $M_1$ of the first $M$ desired; $IDM$, the increment in $M$ used to generate higher values of $M$; $NM$, the number of values of $M$ that are desired; $LI$, the initial integer value of $L$; $IDL$, the increment in $L$; $NL$, the number of values of $L$ that are desired; $NH$, the number of values of $H$ that are desired; and $NX$, the number of values of $X$ that are desired.

Data Card No. 2: Format D32.25. This card contains $AH$, which is the initial decimal value of $H$.

Data Card No. 3: Format D32.25. This card contains $DH$, which is the increment in $H$.

Data Card No. 4: Format D32.25. This card contains $XI$, which is the initial decimal value of $X$.

Data Card No. 5: Format D32.25. This card contains $DX$ which is the increment in $X$.

OUTPUT

The output consists of numerical tables, one page for each set of selected values $H$, $M$, and $X$. Each table gives the radial functions of the first and second kind, $R_1$ and $R_2$, their first derivatives, $R_1D$ and $R_2D$, and the eigenvalue for all choices of $L$ that were requested. Only 18 significant figures are printed in the table, although 26 significant figures are calculated and more than 18 of these may be accurate.

An accuracy check is included for the radial functions and their first derivatives. This is obtained by comparing the theoretical value of the Wronskian $W[R_1, R_2]$ of the radial solutions to the value actually calculated from the radial functions and their first derivatives. It gives either the number of digits that agree in the theoretical and calculated Wronskians (or one less, because of truncation error). When $X = 0$, either $R_1$ or its first derivative is equal to zero. The Wronskian is then insensitive to inaccuracies in either $R_2$ or its first derivative. In this case the accuracy is determined by subtracting from 25 the number of significant figures that are inaccurate due to subtraction errors (Ref. 11).

Experience has demonstrated that this program will deliver correct results if the eigenvalues $A_{m1}$ are correct and the Wronskians check as noted above. Examination of the eigenvalues for continuity is a helpful check on their correctness.

A sample page of the output from OBRAD is presented in Appendix C.
PARAMETER RANGES

To use this program effectively, it is necessary to understand the limitations on the
four parameters $M$, $H$, $L$, and $X$. The ranges that the program has been tested for are as
follows:

$M = 0$ through $10$
$H = 0.01$ through $75$
$L = M$ through $M + 49$
$X = 0; 0.02$ through $100$

Limitations on parameters are as follows.

$M$: In general, accuracy is not much affected with increasing $M$. For this reason one
could reasonably expect good results for values of $M$ greater than $10$.

$H$: The range on $H$ may easily be extended in both directions. Good results should
be obtained for values of $H$ as small as $0.001$. However, the accuracy may fall off for
values of $H$ greater than $75$, especially when $X$ is small and $L$ is large. Since the matrix
determination of starting values for the eigenvalues as programmed in OBRAD is in-
adequate for values of $H$ greater than $75$, a formula given by Meixner (2) is used in this
case.

$L$: The upper limit on $L$ may be extended beyond $M + 49$. For $H < 20$, $L$ can probably
be extended to $M + 79$. As $H$ is increased from 20 to 75, the upper limit on $L$ must be re-
duced from $M + 79$ to the present limit of $M + 49$. As was mentioned, a larger matrix for
computing eigenvalues would be required to extend the upper limit on $L$ beyond this.
When the range is extended, the eigenvalues should be examined carefully for continuity.
Since the difference between successive eigenvalues becomes nearly constant for large
$L$, this is the best check on their validity.

$X$: The range for $X$ was determined by the physical problem. $X = 0$ represents the
surface of a disk and is useful for this reason. The flattest near-disk that one might
consider would probably correspond to a value of $X$ no less than 0.02. The upper limit
on $X$ was chosen arbitrarily and could probably be extended with little difficulty to well
over 100.

ACCURACY CURVES

Several graphs of the calculated accuracy as a function of $H$ and for a fixed value of
$X$ are given below in Figs. 1 through 5. The arrow indicates the range of accuracy for
$L = M$ to $L = M + 49$ and for $M = 0$ to $M = 10$. The lower accuracy usually corresponds to
higher $L$. For the parameter ranges listed above OBRAD will produce values for $R^{(1)}$
and its first derivative that are accurate to at least 20 significant figures. When the
Wronskian check is less than 20 significant figures, it indicates lower accuracy only in
$R^{(2)}$ and its first derivative.
Fig. 1 Calculated accuracy as a function of $H$ for $X = 0.0$

Fig. 2 Calculated accuracy as a function of $H$ for $X = 0.02$

Fig. 3 Calculated accuracy as a function of $H$ for $X = 0.10$
COMPUTATION TIME

Using the CDC 3800 the compilation time of OBRAD is about 48 seconds. The execution time varies, but if all 50 values of L are requested, the average time will be about 0.4 second for each set of R1, R1D, R2, R2D, and eigenvalue.

SOLUTION OF THE HELMHOLTZ EQUATION IN OBLATE SPHEROIDAL COORDINATES

Details of the oblate spheroidal coordinate system, which is an orthogonal coordinate system, are given in Ref. 1. Briefly, the three oblate spheroidal coordinates are $\xi$, $\eta$, and $\varphi$, where $0 \leq \xi \leq \infty$, $-1 \leq \eta \leq 1$, and $0 \leq \varphi \leq 2\pi$. The surfaces of constant $\xi$, the radial coordinate, are represented in the $xyz$ Cartesian system by the locus

$$\frac{x^2+y^2}{\left(\frac{d}{2}\right)^2 \left(\xi^2+1\right)} = 1 - \frac{z^2}{\left(\frac{d}{2}\right)^2 \xi^2}. \quad (1)$$
This describes an oblate spheroid whose interfocal distance is \( d \). The surfaces of constant \( \eta \), the angle coordinate, are represented by the locus

\[
\frac{x^2 + y^2}{(d/2)^2 (1-\eta^2)} = 1 + \frac{z^2}{(d/2)^2 \eta^2}.
\]

(2)

This defines a hyperboloid of one sheet whose focus is located a distance \( d/2 \) from the origin of the Cartesian system.

The surfaces of constant \( \varphi \), the rotational coordinate, are half planes whose edge is the \( a \) axis.

Any point in three-space can then be represented by the triad \((\xi, \eta, \varphi)\).

The scalar wave equation \((\nabla^2 + \kappa^2) \Phi = 0\) is separable in oblate spheroidal coordinates. Adopting with slight modifications the notation of Morse and Feshbach (3), we write

\[
\Phi = S(i\hbar, \eta) R(i\hbar, -i\xi) \cos m \varphi \sin m \varphi,
\]

(3)

where

\[
\hbar = \frac{k d}{2}.
\]

(4)

The angle function \( S \) and the radial function \( R \) satisfy the ordinary differential equations

\[
\frac{d}{d\eta} \left[ (1 - \eta^2) \frac{dS}{d\eta} \right] + \left( A + \hbar^2 \eta^2 - \frac{m^2}{1 - \eta^2} \right) S = 0,
\]

(5)

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

(6)

Here \( A \) represents a separation constant dependent on \( m, \ell, \) and \( \hbar \). There are two solutions to both Eqs. (5) and (6). Consider only the first solution \( S^{(1)} \) to Eq. (5). When \( \hbar = 0 \), Eq. (5) reduces to the standard equation for the associated Legendre function \( P_{\ell}^m(\eta) \) of the first kind, where the separation constants are \( A = \ell (\ell + 1); \ell = m, m + 1, \ldots \). Thus, for each pair of integers \( m \) and \( \ell \), both Eq. (5) and Eq. (6) have a solution only for special values of \( A = A_{m\ell}(i\hbar) \). For \( \hbar \neq 0 \) we can write

\[
S_{m\ell}^{(1)}(i\hbar, \eta) = \sum_n d_n(i\hbar|m\ell) P_{m+n}^\ell(\eta).
\]

(7)

The prime sign means that \( n = 0, 2, 4, \ldots \) if \( \ell - m \) is even and \( n = 1, 3, 5, \ldots \) if \( \ell - m \) is odd. Substituting Eq. (7) into Eq. (5) and using known recursion formulas for \( P_{\ell}^m(\eta) \), one obtains the following three-term formula for the expansion coefficients:
\[- \frac{(2m+n+2)(2m+n+1)\lambda^2}{(2m+2n+3)(2m+2n+5)} d_n^2 + \left[ (m+n)(m+n+1) - A_{m\ell} - \frac{2(m+n)(m+n+1) - 2m^2 - 1}{(2m+2n-1)(2m+2n+3)} \right] d_n \]

\[- \frac{n(n-1)\lambda^2}{(2m+2n-3)(2m+2n+1)} d_{n-2} = 0, \]

with the asymptotic relation
\[ \frac{d_n^2}{d_n} \to 0 \quad n \to \infty \]

and the normalization
\[ \sum_n' \frac{(n+2m)!}{n!} d_n(i\lambda|m\ell) = \frac{(\ell+m)!}{(\ell-m)!}. \]

A knowledge of \(A_{m\ell}\) would then allow \(d_n\) to be calculated by an iterative process.

This program, however, is concerned with only the radial functions \(R_1^{(1)}(ih, -i\xi)\) and \(R_1^{(2)}(ih, -i\xi)\). Using the general principle that any solution of the scalar wave equation (say the angle function \(S\)) is a suitable kernel for the integral representation of a second solution (say the radial function \(R_1^{(1)}\)) the following expression for the radial function of the first type \(R_1^{(1)}\) is obtained by integration over \(S\):
\[ R_1^{(1)}(ih, -i\xi) = \frac{\xi \xi + 1}{\xi^2} \sum_n' \frac{(n+2m)!}{n!} d_n(i\lambda|m\ell) j_n + m(\lambda \xi), \]

where \(j_n\) is the spherical Bessel function of the first type.

Using a known recursion formula for the spherical Bessel functions, the derivative of \(R_1^{(1)}\) is obtained:
\[ \frac{dR_1^{(1)}}{d\xi} (ih, -i\xi) = \frac{\xi \xi + 1}{\xi^2} \sum_n' \frac{(n+2m)!}{n!} d_n(i\lambda|m\ell) \]

\[ \times \left[ \frac{h(n+m)}{(2n+2m+1)} j_n + m - 1(\lambda \xi) - \frac{h(n+m+1)}{(2n+2m+1)} j_n + m + 1(\lambda \xi) - \frac{m}{(\xi^2 + \xi)} i_n + m(\lambda \xi) \right]. \]

Using an asymptotic form for the spherical Bessel function, we can find asymptotic forms for the radial function of the first kind:
\[ R_{m}^{(1)} (i\hbar, -i\ell) = \frac{(\ell - m)! \, i^m - \ell (2m)! \, d_0 (i\hbar|m\ell) \, h^m}{(\ell + m)! \, (2m - 1)!!}, \quad \ell - m = \text{even}, \]
\[ = 0, \quad \ell - m = \text{odd}, \]  
\[ \frac{dR_{m}^{(1)}}{d\zeta} (i\hbar, -i\ell) = \frac{(\ell - m)! \, i^m - \ell + 1 (2m + 1)! \, d_1 (i\hbar|m\ell) \, h^{m+1}}{(\ell + m)! \, (2m + 3)!!}, \quad \ell - m = \text{even}, \]
\[ = 0, \quad \ell - m = \text{odd}. \]  

Here \((2m + 1)!! = (2m + 1)(2m - 1)\ldots(3)(1)\). Similarly the radial function of the second kind \(R_{m}^{(2)} (i\hbar, -i\ell)\) can be expanded in terms of \(y_n (\ell \xi)\), the Neumann function or spherical Bessel function of the second kind:

\[ R_{m}^{(2)} (i\hbar, -i\ell) = \frac{(\ell - m)! \, i^m - \ell + 1 \xi^{m+1}}{(\ell + m)! \, \xi^2} \sum_n i^n + m - \ell \frac{(n + 2m)!}{n!} d_n (i\hbar|m\ell) \, y_{n + m} (\ell \xi), \]  
\[ = 0, \quad \ell - m = \text{odd}. \]  

Since this expansion contains the Neumann function, its usefulness is limited to large values of \(\ell \xi\). A second method is necessary to obtain the radial function of the second kind for small values of \(\ell \xi\).

Consider the special expansion of the oblate angle function first discovered by Baber and Hasse (4):

\[ S_{n\ell} (i\hbar, \hbar) = e^{h\eta} \sum_{n=0}^{\infty} \hat{\Omega}_{n\ell} \, P_{n + m} (\eta). \]  

When this is substituted into Eq. (5), and the recursion formulas of \(P_{n\ell} (\eta)\) used, it is found that the expansion coefficients \(\hat{\Omega}_{n\ell}\) satisfy the three-term recursion formula

\[ \frac{2h (n + m + 1) (n + 2m + 1)}{(2n + 2m + 3)} \hat{\Omega}_{n\ell} - \left[ (n + m)(n + m + 1) - \hbar^2 \right] \hat{\Omega}_{n\ell} \]
\[ - \frac{2h \eta (n + m + 1) \ell}{2n + 2m - 1} \hat{\Omega}_{n\ell - 1} = 0, \]

with the asymptotic condition...
By substituting $i \xi$ for $\eta$ and $Q_{m+n}^m$ for $P_{m+n}^m$, one can obtain a series expansion of an oblate wave function in the radial coordinate $\xi$. Noting that the asymptotic value of $Q_m(\xi)$, the associated Legendre function of the second kind, is

$$Q_m^m(\xi) \approx \frac{1}{\xi^{d+1}},$$

we can use Eq. (17) in its modified form to obtain an expansion of the radial function of the third kind ($= R^{(3)}$) that will have a radial dependence $e^{i\hbar \xi}/\sqrt{\xi}$ as $\xi \to \infty$. Allowing for appropriate constants, this procedure leads to the formula

$$R^{(3)}(i\hbar, -i \xi) = e^{i \left[ \hbar \xi - (\ell + 1)(n/2) \right]} \sum_{n=-m}^m \frac{i^{2m+1}}{m!} \frac{\partial_n^m}{\partial_n^m Q_{m+n}^m} (i \xi).$$

Now the radial function of the third kind is related to the radial functions of the first and second kind by the formula

$$R^{(3)} = R^{(1)} + i R^{(2)}.$$

Equation (21) can be separated into real and imaginary parts:

$$R^{(3)} = (a + i \beta) (y + i \delta),$$

where

$$a + i \beta = e^{i \left[ \hbar \xi - (\ell + 1)(n/2) \right]},$$

and

$$y + i \delta = \sum_{n=-m}^m \frac{i^{2m+1}}{m!} \frac{\partial_n^m}{\partial_n^m Q_{m+n}^m} (i \xi).$$

We can now identify

$$R^{(1)} = ay - \beta \delta$$

and

$$R^{(2)} = a \delta + \beta y$$

or

$$R^{(2)} = a \delta + \beta \left[ \frac{R^{(1)}}{a} + \delta \right].$$

The form given by Eq. (28) is used for $R^{(2)}$ to avoid computational difficulties associated with $y$. $R^{(2)}$ can now be calculated using Eqs. (24) and (25) and the previously calculated value for $R^{(1)}$.

Similarly the first derivative of the second radial function $R^{(2)}$ is obtained:

$$\frac{dR^{(2)}}{d\xi} = a \nu + \beta \left[ \frac{dR^{(1)}}{d\xi} + \nu \beta \right],$$

(29)
where
\[ \mu + i \nu = \sum_{n=-m}^{\infty} \frac{Q_n^m}{\xi^{m-n}} \left\{ \frac{\xi^{m+n}}{\xi^{2}+1} + i n \right\} + i \left( \frac{2m+n}{\xi^{2}+1} Q_n^{m+n-1} (i \xi) \right) \].

When \( X = 0 \), asymptotic forms can be used for \( R^{(2)} \) and \( dR^{(2)}/d\xi \) which take advantage of the loss of independence of \( R^{(1)} \) and \( R^{(2)} \). These special formulas are obtained by rewriting Eqs. (4.6.15) and (4.6.16) in Flammer (10) to include \( d amounts \) satisfying Eq. (10).

DESCRIPTION OF THE COMPUTER PROGRAM OBRAD

The Fortran IV computer program used to calculate the oblate spheroidal radial functions of the first and second kinds, their first derivatives, and the eigenvalues is listed in Appendix B. Some details of this program are given below.

Expansion Functions

Several special functions are required: the factorial functions, the associated Legendre functions of the second kind, and the spherical Bessel functions of the first and second kinds.

1. The factorials are calculated in the main program in statement 96 + 2 lines through statement 53. It was necessary to scale FACT \((N+1) = N! \) for \( N = 170 \) to 296 to prevent overflow, since the maximum exponent available on the CDC 3800 at NRL is 307.

2. The associated Legendre functions of the second kind, \( Q_n^m (iX) \), where \( X \) is the radial coordinate, is calculated in the subroutine QLEG. QLEG is called after \( M \) and \( X \) have been set and when \( H = AH \), the first choice for \( H \), since \( Q_n^m (iX) \) is independent of \( H \). It returns values of \( Q_n^m (iX) \) for \( N = 0 \) to \( N = 126 + 2M \). \( Q_n^m (iX) \) is either purely real or purely imaginary depending on whether \( N \) is even or odd respectively. Therefore when \( N \) is odd, the real answer returned by QLEG must be multiplied by \( i \) to obtain \( Q_n^m (iX) \). For fixed \( M \) these values are stored for all choices of \( X \) in the matrix OUTPUT \((N + 2, IX)\). Here \( IX \) indicated the specific \( X \), and \( N + 2 \) is chosen so that the first element stored in OUTPUT is \( Q_n^m (iX) \). When \( M \) changes, QLEG is again called for each \( X \) when \( H = AH \).

\( Q_n^m (iX) \) is calculated in the main program in statements 4 through 5 using \( Q_n^m (iX) \) and \( Q_n^m (iX) \) in a backward recursion formula. QLEG uses limiting forms for \( Q_n^m (iX) \) when \( X = 0 \). When \( X > 0 \), \( Q_n^m (iX) \) is calculated from a hypergeometric series, and \( Q_n^m (iX) \) is then obtained by a forward recursion formula. These expressions are given as Eqs. 30 through 32 in Ref. 1 and Eq. 9 on page XVI in Ref. 5.

The output of QLEG was carefully checked for the entire range of \( M \) and \( X \) necessary for OBRAD and found to have an accuracy of at least 20 significant figures.
3. The spherical Bessel function of the first kind is calculated in the subroutine SBESF. SBESF is called after $H$ and $X$ have been set and when $L$ is equal to $L_1$, the first choice for $L$. It returns values for $j_n(HX)$ for $N = 0$ to $N = 145$, unless $HX$ is greater than or equal to 100 when it returns values for $N = 0$ to $N = 145 + M$. SBESF calculates $j_n(HX)$ by a series expansion when $HX < 0.4$, by a backward recursion relation when $0.4 \leq HX < 100.0$, and by a forward recursion relation when $HX \geq 100.0$. These expressions are given in Ref. 6 as 10.1.2 and 10.1.19. The accuracy in $j_n(HX)$ is greater than 20 significant figures for the entire range of $HX$ necessary for OBRAD.

4. The spherical Bessel function of the second kind is calculated in the subroutine SPHYN by a forward recursion relation given as Eq. 10.1.19 in Ref. 6. If $X > 1.0$ and $HX > 10.0$, SPHYN is called after $H$ and $X$ have been set and when $L$ is equal to $L_1$. It returns values of $y_n(HX)$ for $N = 0$ to $N = 143 + M$. The accuracy in $y_n(HX)$ when $HX > 1.0$ is greater than 20 significant figures.

Eigenvalues

Before the expansion constants can be evaluated, it is necessary to know the eigenvalues or separation constants for which solutions to Eqs. (5) and (6) exist.

Starting values or numbers agreeing to at least two places with the correct values are obtained for the eigenvalues. These starting values are solutions to an eigenvalue equation which when expressed in matrix form reduces to the problem of diagonalization of the matrix. The eigenvalues then appear as the resulting diagonal elements when ordered numerically from lowest to highest. Although the exact determination of the eigenvalues would require a matrix of order infinity, good starting values are obtained using matrices of modest proportions. The minimum size matrix used in OBRAD is of order 50, giving 50 possible starting values. When $H \leq 20$, all 50 values are adequate as starting values, with the lowest eigenvalue corresponding to $L = M$. However, as $H$ increases, the order $N$ of the matrix must also be increased to maintain good starting values for the 50 lowest eigenvalues. The order $N$ as determined in statement 3 + 14 lines through statement 3 + 18 lines is adequate to give good starting values for the 50 lowest eigenvalues when $H$ is less than or equal to 75.

The matrix elements $A$ are obtained in statement 3 + 19 lines through statement 43. Subroutine EIGEN, which diagonalizes the matrix $A$, is then called. Details of the matrix and its diagonalization are given in Ref. 7. EIGEN returns the $N$ diagonal elements in ascending numerical order. $LF - 1 + M$ of these, where $LF$ is the highest $L$ desired, are now used as starting values in a variational procedure devised by Bouwkamp (8) and Blanch (9). (When $H$ is greater than 75, good starting values are obtained instead from a formula given by Meixner (2). This formula is programmed in statements 35 through 37.) This variational method adds corrections to the starting values, the corrections becoming successively smaller as the correct eigenvalue is approached. Good starting values are necessary to assure the convergence to correct eigenvalues. Convergence is assumed when the relative contribution of the correction is less than $10^{-24}$. Because of the
limited word length in the CDC 3800 at NRL a more stringent test does not give more accurate eigenvalues, but the corrections oscillate around $10^{-25}$. This variational method is programmed in statement 6 + 3 lines through statement 22 + 1 line.

Expansion Constants

1. The first expansion constants that are used $d_n (i\hbar \mid m\ell)$ are calculated in statements 31 through $32 + 1$ line. These calculations make use of the single subscripted variable ENR that has been obtained above in the eigenvalue correction. Although the method is disguised by the intermediate variable ENR, basically the expansion constants $d_n (i\hbar \mid m\ell)$ (called DLIST(J)) are calculated using Eqs. (8) through (10). Here the index $J$ runs consecutively from 1 to 72. For example, DLIST (3) represents $d_3 (i\hbar \mid m\ell) \text{ when } L - M$ is odd but $d_3 (i\hbar \mid m\ell) \text{ when } L - M$ is even.

2. The expansion constants $\Omega_m^{\ell} / \Omega_m^{\ell^*}$, used in the calculation of the radial function of the second kind, are obtained in statements 232 through 291.

First uncorrected values RATIO $(J)$ are successively calculated by use of the reverse recursion form of Eq. (18) until they begin to decrease $(J = IND + 1)$. Here RATIO $(123 + 2M)$ is chosen equal to 0, and RATIO $(122 + 2M)$ is chosen equal to 1.

Next using the fact that $\Omega_{-m - 1}^{\ell} / \Omega_{-m}^{\ell} = \text{ARATIO} (1) = 0$ and $\Omega_{-m}^{\ell} / \Omega_{-m}^{\ell} = \text{ARATIO} (2) = 1$, true values ARATIO $(J)$ are obtained by use of the forward recursion form of Eq. (18) until $J = IND + 1$.

Finally RATIO $(J)$ is corrected by matching to ARATIO $(J)$ at $J = INL' + 1$.

Evaluation of the Radial Functions

The expansion constants and functions are now combined to give the radial functions. The radial functions of both kinds and their first derivatives are calculated for $X = 0$ in statement 237 through statement 246 + 3 lines. For $X \neq 0$ the radial function of the first kind and its first derivative are calculated in statements 211 through 236 using the expansions given in Eqs. (11) through (14) and the radial function of the second kind and its first derivative are calculated in statement 291 + 1 line through statement 311 + 1 line using Eqs. (28) and (29).

A Wronskian check is made on the two radial functions and their first derivatives in statements 311 + 2 lines through 311 + 4 lines. Here the calculated Wronskian $\text{WWRON}$ is compared to the theoretical Wronskian $\text{TWRON}$ to give the number of significant figures that agree NIAC. When $X = 0$ the accuracy is determined instead by subtracting from 25 the number of accurate figures that are lost during subtraction of nearly equal numbers.

When $X > 1.0$ and $XH > 10.0$, Eqs. (15) and (16) are also used to calculate the radial function of the second kind and its first derivative. This is done in statement 311 + 7
through statement 324. A Wronskian check is made using these values, yielding the integer IAC.

IAC is now compared with NIAC in order to choose between the two sets of values for the radial function of the second kind and its first derivative. If IAC > NIAC, the results obtained using Eqs. (15) and (16) are printed. If NIAC ≥ IAC, the results of Eqs. (28) and (29) are printed instead.

REFERENCES


Appendix A

Major Calculation Blocks of OBRAD Listed According to Statement Numbers

<table>
<thead>
<tr>
<th>Calculation Block</th>
<th>Statement Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate factorials</td>
<td>*96 + 2 to 53</td>
</tr>
<tr>
<td>Read data</td>
<td>1 to 3 + 3</td>
</tr>
<tr>
<td>Do loops</td>
<td></td>
</tr>
<tr>
<td>Set $M$</td>
<td>3 + 4</td>
</tr>
<tr>
<td>Set $H$</td>
<td>3 + 10</td>
</tr>
<tr>
<td>Calculate the starting eigenvalues</td>
<td>3 + 14 to 7</td>
</tr>
<tr>
<td>by Subroutine EIGEN</td>
<td>35 to 37</td>
</tr>
<tr>
<td>by Meixner's formula</td>
<td></td>
</tr>
<tr>
<td>Set $X$</td>
<td>38</td>
</tr>
<tr>
<td>Generate Q's Using</td>
<td>38 + 5 to 5</td>
</tr>
<tr>
<td>Subroutine QLEG</td>
<td></td>
</tr>
<tr>
<td>Set $L$</td>
<td>6 + 1</td>
</tr>
<tr>
<td>Correct the eigenvalues</td>
<td>6 + 3 to 22 + 1</td>
</tr>
<tr>
<td>Calculate constants $d$</td>
<td>31 to 32 + 1</td>
</tr>
<tr>
<td>Calculate radial functions $R1$, $R1D$, $R2$, $R2D$</td>
<td>237 to 246 + 3</td>
</tr>
<tr>
<td>for $X = 0$</td>
<td>211 to 236</td>
</tr>
<tr>
<td>Calculate $R1$, $R1D$ using SBESF</td>
<td></td>
</tr>
<tr>
<td>Calculate $d$ ratios</td>
<td>252 to 291</td>
</tr>
<tr>
<td>Calculate $R2$, $R2D$</td>
<td>291 + 1 to 311 + 1</td>
</tr>
<tr>
<td>by use of $Q$ functions</td>
<td></td>
</tr>
<tr>
<td>Decide whether to use</td>
<td></td>
</tr>
<tr>
<td>Neumann functions to calculate $R2$, $R2D$</td>
<td>311 + 5</td>
</tr>
</tbody>
</table>

*Note: The symbol 96 + 1 signifies statement number 96 plus one line.
<table>
<thead>
<tr>
<th>Calculation Block</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate $R_2$, $R_{2D}$ by means of spherical Neumann functions, subroutine SPHYN</td>
<td>311 + 7</td>
<td>324</td>
</tr>
<tr>
<td>Calculate Wronskian (plus accuracy check)</td>
<td>311 + 2</td>
<td>311 + 5</td>
</tr>
<tr>
<td>and 324 + 1</td>
<td>324 + 2</td>
<td></td>
</tr>
<tr>
<td>Decide to print out final result based on Legendre function approach</td>
<td>324 + 3</td>
<td></td>
</tr>
<tr>
<td>or on Bessel function approach</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final printout</td>
<td>326 + 1</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B

LISTING IN FORTRAN IV OF OBRAD,
A PROGRAM TO CALCULATE OBLATE
SPHEROIDAL RADIAL WAVE FUNCTIONS,
AS COMPILED FOR CDC 3800
PROGRAM OBRAD
TYPE DOUBLE AA,AAA,AR,ARRATIO,ARG,ARRAY,ATERM,BL1ST,BOOK,CL,
COEFF1,COEFF2,COEFF3,COER,CORB,CRON,DE,DL,DLIST,DM,DX,E1,E2,
E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,EYE,FACT,FN1,FN2,FNM,GM,LIST,MPI,
OUTPUT,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,RAD2,
4RAD2,RAD20,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ENTRY CRON,DE,DL,DLIST,DM,DX,E1,E2,E3,EJ,EM,ENR,ENRC,ESTORE,E Sum,
EYE,FACT,FN1,FN2,PAPER,PCL,PEN,P8,P9,PLA,PLB,PLC,PLD,R1,R2,R3,RAD1,
RADIUS,RATIO,RESTORE,RSUM,S,SUM1,SUM2,TEMP,TERM,TERM1,TERM2,
STERM,TWtRON, XX,XX,X,X,XX,EIG0,GRAD2,GRAD20,DW,DSUM,DSUM1,DSUM2,DSUM3,
DTERM1,TERM,TERM1,TERM2,
ANORM1 = AA
ANORM2 = LF*(LF+1*)

CALL EIGEN(A+AIG*N+NI+ANORM1+ANORM2)

DO 7 I = 1, NI
7 EIG(I) = AIG(I)
GO TO 38

35 DO 37 L = 1, LF+2
   P = 2*(L-M)/2+M+1
   EIG(L-M+1) = H*M+2.*H*P-5.*(P*P-M*M+1)-P*(P*P-M*M+1)/(8.*H)
   1-(5.*P**4+10.*P**4+1=2,M*M*(3.*P**4+1)+M**4)/(64.*H**4)
   2-P*(33.*P**4+114.*P**4+237-2.,*M*M*(23.*P**4+35.*M**4)+13.*M**4)/(152.*H**4)
   3 -(63.*P**4+340.*P**4+2390.+P**4-16.*M*M*(10.*P**4+23.*P**4+3*)
   4+3.*H*M**4/(13.*P**4+6.-2.*M*M)/(1024.*H**4)
   5-P*(527.*P**6+4139.*P**4+5221.,*P**4+1009.-M*M*(939.*P**4+3750.*P**P
   6+1591.)*H*M**4/(465.*P**4+635.)*P**4+53.*M**6)/(8192.*H**5)
   IF(2*(L-M)/2+NE*L-M) EIG(L-M) = EIG(L-M+1)
37 EIG(L-M+2) = EIG(L-M+1)
38 DO 221 = 1, NX
   X*I = (I-1)*DX
   XX = X*X + 1*O
   ARG = H*XX
   IF(H*NE+AH*OR+X*EQ+O+D) GO TO 6
   LG = 125.+2*M
   CALL OLEG(M,LG,XX,0)
   DO 8 L = 1, LG
   8 OUTPUT(I+1, XX) = Q(I)
   IF(M*EQ+0) = 5
   4 OUTPUT(I+1, XX) = 0
   GO TO 6
5 OUTPUT(I+1, XX) = OUTPUT(2, I+1, XX)/EM+(EM+1, D)*OUTPUT(3, I+1, XX)/EM
6 PRINT 87, H*X+M

D0 221 = 1, NL
   L = L+1*(L-1)+DL
   PLB = 2-D*EM+1*D
   IF(C = 0)
   LF*+1, L = L/:2
   IR0 = IR0+1
   IR = IR0+1
   CL = EIG(L-M+1)
   IF(2*I+1, UCE+L = L-M) GO TO 10
10 I = 2
   I = 75
   I = 2*M
   GLISTU(1, EM*(EM+1, D)+AA*(PLB-2*D)/((PLB-2*D)*PLB+2*D))
   GO TO 12
10 I = 3
   I = 74
   I = 2*M+1
   GLISTU(1, EM*(EM+1, D)+AA*(PLB+2*D)/((PLB+2*D)*PLB+4*D))
12 LIM = 150
   L = L-1
   DO 13 I = 1, LIM+2
   EYE = 1
   BLISTU(1, L*EYE*(EYE-1, D)*(PLB+2*D)/((PLB+2*D)*EYE-2*D)*AA/
   D)
13 GLISTU(1, ID+1, D)*(EM+1, D)*(EM+1, D)+(1, D)*AA*(1, D)+(PLB+2*D)
   1*PLB)/(PLB+2*D)*EYE-2*D)(PLB+2*D)*EYE-2*D))
17 EM(R) = CL-GLISTU(1)
DO 18 I=1,1UCT
18 ENR(I+1)=ENR(I)-GLIST(I+1)+CL
ENR(I)=ENR(I)+GLIST(I+1)-CL
IP=IP+1
DO 19 I=IP+1
19 ENR(IP)=ENR(IP)-GLIST(IP+1)-CL+ENR(IP+1)
DE=DE-ENR/GLIST(IP)
CORB+DE
DO 20 I=IR+1
DE=ENR/BLIST(I)
DO 21 I=IR+1
IF(DABS(DE/CORB)*LT.1.D-27) GO TO 23
23 CORA=
DO 30 J=1,1UCT
DE=LIST(IR+1)/DE
DE=LIST(IR+1)/DE
DO 31 J=1,1UCT
DN(J)=((2*EM+2*D*AR-1*D)+(2*EM+2*D*AR+1*D)*ENR(1))
1/(2*EM+AR)*(2*EM+AR-1*D)*AA
W=DN(J)*FACT(2*M+J)/FACT(J+1)
DO30J=1,1UCT
AR=AR*X*(J-1)
DN(J)=DN(J-1)*((2*EM+2*D*AR-1*D)+(2*EM+2*D*AR+1*D)
*ENR(J))/((2*EM+AR)*(2*EM+AR-1*D)*AA)
W=DN(J)*(FACT(2*M+J)*FACT(J+1))
IF((2*M+J)+ID-1) GT.170) W=W*1.0+300
30 W=W+DN
DO31J=1,1UCT
DN(J)=DN(J)/FACT(L+M+1)*(W+FAC(T(I+1))
DO32J=1,70
32 DLST(J+1)=DN(J)*DLST(1)
DLST(72)=O*D
IF(X*N*E*O*D) GO TO 200
DRATIO(1)=O*D
DNEG=DLST(1)
DO 33 I=1,M
DRATIO(I+1)=((I+1+ID-2*D)+((I+1+ID-3*D)*AA)/(4*ID+ID-M-M-15*D)
*FACTOR(M+M+1+ID-3)*FACT(M+M+1+ID-2)*FACT(M+M+1+ID-1))
FSTRAT=TERM*DLST(1)
TERM2=DABS*FSTRAT
DO 34 I=1,7
TERM=TERM*(M+M+1+ID-6)*(M+M+1+ID-7)/(4*D*(I-1)*(M+M+1+ID-1))
TERM=TERM*DLST(1)
TERM2=DMAX1(TERM2,DABS(TERM1))
34 FSTRAT=FSTRAT+TERM1
 IAC=26=DLOG10(TERM2/DABS(FSTRAT))
 IF(IAC>GT+25) IAC=25
 GO TO 237

200 E1=DSIN(ARG)/(H*FACT(M+1))
 R1=DLOG(SIN(ARG)/(H*FACT(M+1))
 GO TO (201+203+205+206)+L-(L/4)*4+1

201 E2 = - R1
 R2 = E1
 GO TO 207

203 R2 = - R1
 E2 = - E1
 GO TO 207

205 E2 = R1
 R2 = - E1
 GO TO 207

206 R2 = R1
 E2 = E1
 GO TO (208+209+208+209)+M-(M/4)*4+1

208 E3 = R2
 R3 = - E2
 GO TO 210

209 E3 = - R2
 R3 = E2

210 IF(LNE=L1) GO TO 211
 LNE=145
 IF(X+GE+100.0) LNE=LNE+M
 CALL S85F(ARG,LNE+ARRAY)

211 PLA=DSQRT((XX/(XX*X))*M*FACT(L-M+1)/FACT(L-M+1))
 IA=1
 IF(2*IACT+NE-(L-M)) IA=IA+1
 IC=IA+12-M
 IF(X+GE+100.0) IC=IC+M-4

214 SUBSUM = 0+D

215 DO 217 K = IA+1, IC+2
 BOOK(K)=IBOX3+ABS((K+M-L-1)/2)
 BOOK(K)=BOOK(K)*FACT(K+2*M)/FACT(K)
 IF(K+2*M)GT+170) BOOK(K)=BOOK(K)+D+300

216 SUBSUM = BOOK (K) + SUBSUM
 IF(DABS(BOOK(K)/SUBSUM)+L+1,D-27) GO TO 219

217 CONTINUE

219 RAD1=PLA*SUBSUM
 SSUM1=0+D

220 DO 234 K=IA+1,C+2
 BOOK3+IAS5(K+M-L-1)/2)
 IDS = (K + 11/2
 PCL(K)=DLIST(IIDS)*(FACT(K+2*M)/FACT(K))
 IF(K+2*M)GT+170) PCL(K)=PCL(K)*FAK+D+300

230 PEN (K) = - ARRAY (12)
 GO TO 232

231 PEN(K) = PEN(K) + ARRAY(K+M-1)+PCL(K)*D+1)ARRAY(K+M+1)/(2+D)*PLD+1+D)

232 PAPER (K) = PCL (K) * PEN (K)

233 SSUM1 = PAPER (K) + SSUM1
 IF(DABS(PAPER(K)/SSUM1)+L+1,D-27) GO TO 236
CONTINUE

RAD1=(H+SSUM1-EM*SUBSUM/(X*XX1))*PLA
GO TO 252

DO 239 J=2,M

PLC=FACT(L-M+1)*FACT(2*M+1)*DLIST(1)*H*M/(FACT(L+M+1)*PLB)
GO TO (240,242,244,246,248,250,252-J)*(L-M)-(L/M)/4)+1

RAD1=PLC
RAD1=0.D
RAD2=(M+M-1)*PLC/(FACT(M+1)*FACT(L-M+1)*(H/2.0)**M**FSTRAT**FSTRAT
IF(BI**FACT(M)*FACT(L+M+1)**2.0**H**H**0.0)<1.GO TO 326

RAD1=PLC
RAD1=0.D
RAD2=(M+M-1)*PLC/(FACT(M+1)*FACT(L-M+1)*(H/2.0)**M**FSTRAT**FSTRAT
IF(BI**FACT(M)*FACT(L+M+1)**2.0**H**H**0.0)<1.GO TO 326

RAO2=0.D
RAD1=D/H**RAD1)
GO TO 326

RAO2=0.D
RAD1=-PLC
RAD1=-D/H**RAD1)
GO TO 326

MA=2*M+123
GO TO 256

MA=MA-20
J5=MA=M-2
S=JS
RATIO(MA)=0.D
IMA = MA-1
RATIO (IMA) = 1.0
DO 253 J = 1,IMA
I=IMA-J+1
COEFF (I)=2.0*H*S*(S+EM+1.0)/(S+2.0*H**S+2.0*EM+1.0)-CL-H-H
COEFF (I)=2.0*H*S*(S+EM+1.0)/(S+2.0*H**S+2.0*EM+1.0)

S = S + 1.0
ARATIO (2) = 1.0
ARATIO (1) = 0.0

DO 269 J=250,1+JS
K=JS-J+1+M
FN1 = COEFF (K+1) * RATIO (K+2)
FN2 = COEFF (K+1) * RATIO (K+1)

IF(DABS(RATIO(K+1))>GT.1.0+300) GO TO 254
IF(DABS(RATIO(K+1))>GT.DABS(RATIO(K))) GO TO 272

CONTINUE
IND=K
GO TO 274

IND=K+1

IF (IND=L+2) IND=2
281 DO 289 J=1,K
   1J = J + 1
   FN1 = COEFF2(J) * ARATIO(1J)
   FN2 = COEFF3(J) * ARATIO(1J - 1)
   ARATIO(1J + 1) = (FN1 + FN2) / COEFF1(J)
289 CONTINUE
   RATIO(1) = RATIO(IND) / ARATIO(IND+1)
   DO 291 I=IND,MA
291 ARATIO(I+1) = RATIO(I+1) / ARATIO(I)
   RSUM = OUTPUT(3+IX) * ARATIO(3)
   ESUM = OUTPUT(2+IX)
   RSTORE = OUTPUT(2+IX) * [(EM+1.0) * ARATIO(3)+H]
   ESTORE = EM * OUTPUT(1+IX) / XX + OUTPUT(3+IX) * ARATIO(3)*H
   IF(N.EQ.0) ESTORE = ESTORE + 1.0/XX
   DO 306 K=4,MA
      AR = K
      TERM1 = OUTPUT(K+IX) * ARATIO(K)
      TERM3 = ARATIO(K) * (EM+AR-2.0)*OUTPUT(K-1+IX)/XX
      IF(2*(K/2)+EQ.K) GO TO 304
      RSUM = RSUM + TERM1
      ESTORE = ESTORE + TERM1 + H
      RSTORE = RSTORE + TERM3
      IF(ABS(TERM1)/ESTORE .LE. 1.0D-27 .AND. ABS(TERM3/RSTORE).LE.1.0D-27)
         GO TO 309
      GO TO 306
   304 TERM2 = TERM1 * X * (AR-2.0) / XX
   ESUM = ESUM + TERM1
   RSTORE = RSTORE - TERM1 / H
   ESTORE = ESTORE + TERM2 - TERM3
   306 CONTINUE
   309 RSTORE = (RAD1 + E3 + ESUM)/R3
            RSTORE = (RAD1D + ES + ESTORE)/R3
   311 CRAD2 = CRAD2 + ESUM + E3 / RSUM
            CRAD2D = CRAD2D / RAD1
   312 TWON = 1.0/(1.0+XX)
            CWRON = RAD1 * CRAD2D - CRAD2 * RAD1D
            SAT = DLOG10(DABS(TWON-CWRON)/TWON)+1.0D-26
            IF(X .LE. 1.0D-10 .OR. (X*M) .LE. 1.0D-30) GO TO 325
            IF(L .NE. L1) GO TO 313
   313 JN = 1
            IF(2*(L-M)/2 .EQ. (L-M)) JN=0
            RAD2 = 0.0
   315 K=1+JN/2
            TERM = FACT(JN+2*M+1)/FACT(JN+1) * DLIST(K) * FNM(JN+M+1)
            IF(JN+2*M+1 .GT. 170) TERM = TERM * 1.0D300
            IF((JN+M-L)/4 .NE. (JN+M-L)) TERM = TERM
            RAD2 = RAD2 + TERM
            IF(K .LE. 5) GO TO 316
            IF(DABS(TERM / RAD2) .LT. (1.0D-27)) GO TO 316
   316 JN = JN+2
            GO TO 315
   318 RAD2 = RAD2 * PLA
             PL9 = EM/(X**3+X)
             RAD2D = 0.0
            JN = JN+2*(JN/2)
   320 K = 1+JN/2
            EJN = JN
SUBROUTINE SBESF(XH,LJ,RAY)
DIMENSION RAY(250)
TYPE DOUBLE CP,FACT,CRAY,SUM,TERM,TM,XH,XI,Z2H
COMMON FACT(300)
L = 0
1 IF (XH.GE.4D) GO TO 4
   Z2H=XH#XH/2#D
   DO 3 N=L+LJ
   TM=FACT(N+1)#(XH+XH)#N/FACT(N+N+2)
   IF (N.GT.64) TM=TM+l.D-300
   SUM=1.D
   TERM = 1.D
   DO 2 I=1,150
   XI=1*(N+N+I+1)
   TERM=TERM#Z2H/XI
   SUM = SUM + TERM
2 IF(DABS(TERM/SUM) .LE. 1.D-26) GO TO 3
3 RAY(N+1) = TM*SUM
   RETURN
4 N=170
   IF (XH.LT.100.D) GO TO 20
   RAY(1)=DSIN(XH)/XH
   RAY(2)=(-RAY(1)-DCOS(XH))/XH
   DO 11 K=1,LJ
   RAY(K+2)=XH*RAY(K+1)/XH-RAY(K)
11 RETURN
20 IF (XH.GT.10.D) N=210
   RAY(N+1)=1.D-250
   RAY(N+2)=0.D
   I = -N
   M = -1
SUBROUTINE SPHYN(X,N,ARR)

DIMENSION ARR(250)

TYPE DOUBLE X,RAY,TKP1

ARR(1)=-DCOS(X)/X

ARR(2)=ARR(1)/X-DSIN(X)/X

DO 2 K=1,N

TKP1=K*K+1

ARR(K+2)=TKP1*ARR(K+1)/X-ARR(K)

END

SUBROUTINE EIGEN(A,VALU,N,N1,ANORM1,ANORM2)

DIMENSION A(BO:80),VALU(BO),DIAG(80),Q(BO),VALL(BO)

NN=N-2

DO 160 I=1,NN

II=I+2

DO 160 J=II,N

T1=A(I+I)

T2=A(I+J)

IF(T2.EQ.0.0)GO TO 160

T=T1/SQRT(T1*T1+T2*T2)

SIN=T2*T

COS=T1*T

DO 105 K=I,N

T2=COS*A(K+I+1)+SIN*A(K+J)

A(K+J)=COS*A(K+J)-SIN*A(K+I+1)

105 A(K+I+1)=T2

DO 125 K=I,N

T2=COS*A(I+I+K)+SIN*A(J+K)

A(J+K)=COS*A(J+K)-SIN*A(I+I+K)

125 A(I+I+K)=T2

CONTINUE

DIAG(I)=A(I+I)

Q(I)=A(I+1-I)*A(I+1-I)

VALL(I)=ANORM1

15 VALU(I)=ANORM2

I=1

MATCH = N

18 TAU=(VALL(I)+VALU(I))/2+...
IF (MATCH .NE. I-1) LATCH = MATCH
MATCH = 0
TO=0
T1=1.E-100
DO20 J=1,N
T2=(DIAG(J)-TAU)*T1=O(J)*TO
IF((T1.LT.0.) .AND. ((T2*T1).LE.0.)) MATCH=MATCH+1
TO=T1
20 TI=T2
DO25 K=1,MATCH
25 VALU(K)=TAU
MATCH=MATCH+1
DO 20 K=MATCH,LATCH
30 IF(TAU.GT.VALL(K)) VALL(K)=TAU
40 IF((VALU(1)-VALL(1)).GT.(1.E-4)) GO TO 18
18 I=I+1
MATCH = N
IF(I.LE.NL) GO TO 40
END

SUBROUTINE QLEG(M, LN, X, Q)
DIMENSION Q(20)
COMMON FACT(300)
TYPE DOUBLE BE, COEF, DI, DK, DM, DN, FACT, GA, Q, SUM, TERM, XA, YA, Z, ZA
DN=M
NN=0
YA=DSORT(X*X+1.D)
ZA=(YA+X)*(YA+X)
XA=0.25D*(YA+X)
Z=2.E/(YA+X)
LN=LN+1
DO 135 N=NN,LNN
DN=N
BE=DN+1.D
GA=DN+1.5D
IF(N.GT.84) GO TO 500
COEF=Z*X**(N)*/FACT(N+1) /FACT(N+2)*FACT(N+1))
GO TO 510
500 COEF=Z*X**(-N)*/FACT(N+1) *(1.D-300)*FACT(N+1)/FACT(N+2)
510 SUM=TERM=1.D
DK=1.D
130 DK=DK+1.D
IF(DK.GT.5000.) GO TO 135
131 TERM=TERM*(DK+5D)*(BE+DK)/(DK+1.D)*ZA*(GA+DK)
SUM=SUM+TERM
IF(ABS(TERM/SUM).GT.500.E-27) GO TO 130
135 Q(N+1)=-COEF*SUM
DO 30 I=1,M
DI=1
DO 30 N=NN,LNN
DN=N
30 Q(N+1)=-(DI+DN)*X*Q(N+1)+(DN-DI+2.D)*Q(N+2)/YA
DO 40 N=NN,LNN
IF((Z*(N+1)/4)**NEE((N+1)/2)) Q(N+1)=-Q(N+1)
40 IF((Z*(N/2)+NEE) Q(N+1)=-Q(N+1)
END
Appendix C

SAMPLE OUTPUT FROM OBRAD
<table>
<thead>
<tr>
<th>L</th>
<th>R1</th>
<th>R10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.447769847824575 x 10^0</td>
<td>1.447769847824575 x 10^0</td>
</tr>
<tr>
<td>1</td>
<td>2.441330364282346 x 10^1</td>
<td>2.441330364282346 x 10^1</td>
</tr>
<tr>
<td>2</td>
<td>3.462406018926953 x 10^2</td>
<td>3.462406018926953 x 10^2</td>
</tr>
<tr>
<td>3</td>
<td>4.440002993207397 x 10^3</td>
<td>4.440002993207397 x 10^3</td>
</tr>
<tr>
<td>4</td>
<td>5.372941078389476 x 10^4</td>
<td>5.372941078389476 x 10^4</td>
</tr>
<tr>
<td>5</td>
<td>6.239612731777925 x 10^5</td>
<td>6.239612731777925 x 10^5</td>
</tr>
<tr>
<td>6</td>
<td>6.935011393693577 x 10^6</td>
<td>6.935011393693577 x 10^6</td>
</tr>
<tr>
<td>7</td>
<td>7.443944478932523 x 10^7</td>
<td>7.443944478932523 x 10^7</td>
</tr>
<tr>
<td>8</td>
<td>7.727099390771668 x 10^8</td>
<td>7.727099390771668 x 10^8</td>
</tr>
<tr>
<td>9</td>
<td>7.815751625563799 x 10^9</td>
<td>7.815751625563799 x 10^9</td>
</tr>
<tr>
<td>10</td>
<td>7.711806720650255 x 10^10</td>
<td>7.711806720650255 x 10^10</td>
</tr>
<tr>
<td>11</td>
<td>7.418279845941025 x 10^11</td>
<td>7.418279845941025 x 10^11</td>
</tr>
<tr>
<td>12</td>
<td>7.013925042959767 x 10^12</td>
<td>7.013925042959767 x 10^12</td>
</tr>
</tbody>
</table>

**EIGENVALUE**

<table>
<thead>
<tr>
<th>ACC</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
</tr>
</tbody>
</table>

**VAN BUKEN, BAIER, AND HANISH**
A FORTRAN COMPUTER PROGRAM FOR CALCULATING THE OBLATE SPHEROIDAL RADIAL FUNCTIONS OF THE FIRST AND SECOND KIND AND THEIR FIRST DERIVATIVES

An interim report on a continuing NRL Problem

A. L. Van Buren, R. V. Baier, and S. Hanish

January 20, 1970

NRL Problem 501-28
RR 102-08-41-5225

NRL Report 6959

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

The Helmholtz or scalar wave equation 

\( (\nabla^2 + k^2) \psi = 0 \)

is separable in oblate spheroidal coordinates \( \eta \), \( \xi \), with solutions \( \psi = S(\eta, \xi) \cdot R(\eta, -i\xi) \cdot \Phi(\xi) \). The subject of this report is a Fortran computer program called ORRAD which numerically evaluates the radial solutions \( R(\eta, -i\xi) \). The printed output from ORRAD consists of radial functions of the first and second kind, \( R^{(1,2)}(\eta, -i\xi) \), their first derivatives \( \partial R^{(1,2)}(\eta, -i\xi) / \partial \xi \), the separation constants or eigenvalues \( A_{\eta, \xi}(\eta) \), and an accuracy check. This report first describes the input data cards and the output format. The theory of the oblate spheroidal wave function is then discussed. A description of the principal internal features of ORRAD is then given. Finally a computer listing of ORRAD is attached as an appendix.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th></th>
<th></th>
<th>LINK B</th>
<th></th>
<th></th>
<th>LINK C</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Acoustic radiation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer programming</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mathematical models</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis (mathematics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wave equations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spheroidal wave functions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acoustic fields</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sound transducers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sonar transducers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sonar arrays</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Submarines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SUPPLEMENTARY REMARKS CONCERNING

NRL REPORTS 6959(1), 7012(2), and 7161(3)

B. J. KING
A. L. VAN BUREN

CODE 8150
NAVAL RESEARCH LABORATORY
WASHINGTON, D.C. 20390
I. DO LOOP PROPERTIES OF 3800 FORTRAN AND FORTRAN IV

The programs OBRAD 96959), PRAD (7012), and ANGLFN (7161) were written in 3400/3600/3800 FORTRAN, not FORTRAN IV. 3400/3600/3800 FORTRAN has the following property concerning DO loops:

\[ n \leq m_1, m_2, m_3 \]

If \( m_1 \) exceeds \( m_2 \) on the initial entry to the loop, the loop is not executed, and control passes to the next statement after \( n \).

This property is used in all three programs. It is specifically required in the following locations.

### OBRAD (6959)

<table>
<thead>
<tr>
<th>Line</th>
<th>Pages/Columns</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>17 lines from top</td>
<td>DO 26 I = I, IUCT</td>
</tr>
<tr>
<td>20</td>
<td>14 lines from bottom</td>
<td>DO 33 I = I, M</td>
</tr>
<tr>
<td>22</td>
<td>4 lines from top</td>
<td>237 DO 239 I = 2, M</td>
</tr>
<tr>
<td>26</td>
<td>10 lines from top</td>
<td>DO 25 K = I, MATCH</td>
</tr>
<tr>
<td>26</td>
<td>13 lines from top</td>
<td>DO 30 K = NATCH, LATCH</td>
</tr>
<tr>
<td>26</td>
<td>9 lines from bottom</td>
<td>DO 30 I = I, M</td>
</tr>
</tbody>
</table>

### PRAD (7012)

<table>
<thead>
<tr>
<th>Line</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1260</td>
<td>DO 26 I = I, IW6</td>
</tr>
<tr>
<td>3890</td>
<td>DO 700 J = K, KPK</td>
</tr>
<tr>
<td>4160</td>
<td>DO 701 J = K, KPK</td>
</tr>
<tr>
<td>4520</td>
<td>DO 702 J = K, KPK</td>
</tr>
<tr>
<td>4790</td>
<td>DO 703 J = K, KPK</td>
</tr>
<tr>
<td>5390</td>
<td>DO 40 J = I, NLESSM</td>
</tr>
<tr>
<td>5610</td>
<td>70 DO 40 J = 2, MONE</td>
</tr>
<tr>
<td>7320</td>
<td>DO 25 K = I, MATCH</td>
</tr>
<tr>
<td>7350</td>
<td>DO 30 K = NATCH, NBL</td>
</tr>
</tbody>
</table>

### ANGLFN (7161)

<table>
<thead>
<tr>
<th>Line</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1300</td>
<td>DO 26 I = 1, IW6</td>
</tr>
<tr>
<td>3870</td>
<td>DO 25 K = I, MATCH</td>
</tr>
<tr>
<td>3900</td>
<td>DO 30 K = NATCH, NBL</td>
</tr>
</tbody>
</table>

Several versions of FORTRAN have the property that a DO loop is executed at least once regardless of the values of the parameters. An IF statement comparing the parameters \( m_1 \) and \( m_2 \) just before the DO statement will
ensure correct execution when using FORTRAN versions having this property, e.g.:

\[
\begin{align*}
\text{IF (} m_1 \text{ .GT. } m_2 \text{) GO TO } n_1 \\
\text{DO } n = m_1, m_2, m_3 \\
\ldots \\
\ldots \\
n \ldots \ldots \\
n_1 \ldots \ldots \\
\end{align*}
\]

II. WORD SIZE OF THE CDC 3800 AT N.R.L.

Double precision numbers have 26 decimal digits and an exponent range of -307 to +307. For computers with word length less than 26 decimal digits statements such as lines 1260 and 1370 of ANGLFN may not be optimum. In addition, the integer accuracy check which is calculated in lines 4980 and 5020 of PRAD and statement 311 + 4 lines and statement 324 + 2 lines in OBRAD will be 26 whenever the theoretical and calculated Wronskians are identical, regardless of the number of available decimal digits. Because of the extensive use of the exponent range of ±307 in scaling and limits on recursion calculations, it is recommended that the user choose a computer with an exponent range at least this large.

REFERENCES


2