NRL Report 7694

UNCLASSIFIED

Fortran Computer Programs for Calculating the Axisymmetric Acoustic Radiation from Two Coaxial Spheroids

B. J. KING AND A. L. VAN BUREN

Transducer Branch
Acoustics Division

July 16, 1974

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
NRL Report 7694

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
FORTRAN COMPUTER PROGRAMS FOR CALCULATING THE AXISYMMETRIC ACOUSTIC RADIATION FROM TWO COAXIAL SPHEROIDS

5. TYPE OF REPORT & PERIOD COVERED
Interim report on a continuing NRL Problem.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
B.J. King
A.L. Van Buren

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Research Laboratory
Washington, D.C. 20375

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
NRL Problem S01-29
RR 131-03-41-5226

11. CONTROLLING OFFICE NAME AND ADDRESS
Department of the Navy
Office of Naval Research
Arlington, Virginia 22217

12. REPORT DATE
July 16, 1974

13. NUMBER OF PAGES
88

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Acoustic radiation
Fortran computer programs
Two spheroids
Prolate spheroids
Oblate spheroids

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The computer programs PSPRDS and OSPRDS calculate acoustic radiation from two prolate and two oblate spheroids, respectively. The programs use an eigenfunction expansion of the appropriate Green's function in spheroidal wave functions. An addition theorem which expresses spheroidal wave functions in one coordinate system in terms of spheroidal wave functions in another coordinate system is used to facilitate application of homogeneous Neumann boundary conditions on the two spheroids in order to determine the unknown expansion coefficients. Only axially symmetrical problems are handled by the programs.

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Abstract (Continued)

The basic inputs to the programs are the geometry of the spheroidal configuration and the normal velocity distribution on the surface of each spheroid. The output consists of radiation impedances, the azimuthal far-field pressure distribution, and near-field pressures at desired points.

UNCLASSIFIED

CONTENTS

INTRODUCTION 1

ANALYSIS 1

THE COMPUTER PROGRAMS PSPRDS AND OSPRDS 13

BASIC STRUCTURE OF THE COMPUTER PROGRAMS 13

Parameter Input 13
Spheroidal Functions Calculation 18
Matrix Calculation 19
Radiation Calculations and Output 19

ACCURACY OF THE RESULTS 21

REFERENCES 22

APPENDIX A—Listing in Fortran of PSPRDS, a Computer Program to Calculate the Axisymmetric Acoustic Radiation from Two Coaxial Prolate Spheroids 23

APPENDIX B—Listing in Fortran of OSPRDS, a Computer Program to Calculate the Axisymmetric Acoustic Radiation from Two Coaxial Oblate Spheroids 51

APPENDIX C—Sample Input and Output from OSPRDS 78

UNCLASSIFIED

FORTRAN COMPUTER PROGRAMS FOR CALCULATING THE AXISYMMETRIC ACOUSTIC RADIATION FROM TWO COAXIAL SPHEROIDS

INTRODUCTION

The Helmholtz scalar wave equation is separable in both prolate and oblate spheroidal coordinates. Thus the acoustic radiation from spheroidal surfaces vibrating with a known normal velocity distribution can be calculated by use of eigenfunction expansions in terms of spheroidal wave functions.

This report describes the documentation of two computer programs, PSPRDS (Prolate SPheRoiDS) and OSPRDS (Oblate SPheRoiDS), which have been developed to calculate acoustic radiation from two prolate or two oblate spheroids, respectively. The programs handle only axially symmetrical problems. A Green's function approach is used with the necessary Green's function being expanded in spheroidal wave functions about both spheroids. An addition theorem which expresses spheroidal wave functions in one coordinate system in terms of spheroidal wave functions in another coordinate system was developed [1] to facilitate application of homogeneous Neumann boundary conditions on the two spheroids in order to determine the unknown expansion coefficients.

The basic inputs to the programs are the geometry of the spheroidal configuration which includes the shape and the acoustic size of each spheroid, the separation between the centers of the two spheroids, and the normal velocity distribution on the surface of each spheroid. The output consists of self- and mutual-radiation impedances, the azimuthal far-field pressure distribution, and values of the acoustic pressure at desired near-field points.

The analysis is presented in the first section. Although the development parallels that given in a previous report describing some numerical results obtained using PSPRDS and OSPRDS [2], some of the equations have been rearranged so that they closely match the corresponding Fortran expressions. The remainder of the report describes the computer programs. Included are discussions of the programs' main structure, subroutines, input, output, accuracy, and speed of computation. The reader is referred to Ref. 2 for some examples of problems that have been solved using PSPRDS and OSPRDS.

ANALYSIS

The prolate spheroidal coordinates (ξ, η, ϕ) are related to Cartesian coordinates by the transformation

Note: Manuscript submitted November 9, 1973.

$$x = \frac{d}{2}(1 - \eta^2)^{1/2}(\xi^2 - 1)^{1/2} \cos\phi,$$

$$y = \frac{d}{2}(1 - \eta^2)^{1/2}(\xi^2 - 1)^{1/2} \sin\phi,$$

$$z = \frac{d}{2}\xi\eta, \tag{1}$$

where d is the interfocal distance, and where $1 \leqslant \xi < \infty$, $-1 \leqslant \eta \leqslant 1$, and $0 \leqslant \phi < 2\pi$. This geometry is shown in Fig. 1. In the Cartesian coordinate system the surface of constant ξ is a prolate spheroid having a major axis of length ξd and a minor axis of length $(\xi^2 - 1)^{1/2}d$. The surface of constant η is one sheet of a hyperboloid of two sheets. The surface of constant ϕ is a half-plane, as in the spherical coordinate system.

The corresponding transformation for the oblate spheroidal coordinates is

$$x = \frac{d}{2}(1 - \eta^2)^{1/2}(\xi^2 + 1)^{1/2} \cos\phi,$$

$$y = \frac{d}{2}(1 - \eta^2)^{1/2}(\xi^2 + 1)^{1/2} \sin\phi,$$

$$z = \frac{d}{2}\xi\eta, \tag{2}$$

where now $0 \leqslant \xi < \infty$, $-1 \leqslant \eta \leqslant 1$ and $0 \leqslant \phi < 2\pi$.

The oblate geometry is shown in Fig. 2. Here the surface of constant ξ is an oblate spheroid having a major axis of length $(\xi^2 + 1)^{1/2}d$ and a minor axis of length ξd. The surface of constant η is a hyperboloid of one sheet. The oblate spheroidal coordinate system can be obtained from the prolate spheroidal coordinate system by use of the interchange $\xi \to i\xi$ and $d \to -id$. Expressions developed for prolate spheroidal geometry can be converted into analogous expressions for oblate spheroidal geometry by use of the same interchange. Consequently, although only the prolate expressions are given explicitly in the following discussion, the corresponding oblate expressions are also valid.

Consider two spheroidal surfaces S_1 and S_2, which are parallel and share a common z axis, as shown in Fig. 3. Spheroidal coordinate systems $C_1(\xi_1, \eta_1, \phi_1)$ and $C_2(\xi_2, \eta_2, \phi_2)$ are established which contain $S_1(\xi_1 = \xi_{10})$ and $S_2(\xi_2 = \xi_{20})$, respectively, as natural coordinate surfaces. The interfocal lengths of the two systems are d_1 and d_2, and the separation of their origins is r_{12}. The region outside both S_1 and S_2 is assumed to be filled with an infinite homogeneous fluid of mass density ρ and sound speed c. The surfaces S_1 and S_2 are assumed to be vibrating with rotationally symmetrical normal velocity distributions whose spatial dependences are given by $v_1(\eta_1)$ and $v_2(\eta_2)$. The time dependence is taken to be harmonic, with the factor $e^{i\omega t}$ being suppressed.

UNCLASSIFIED

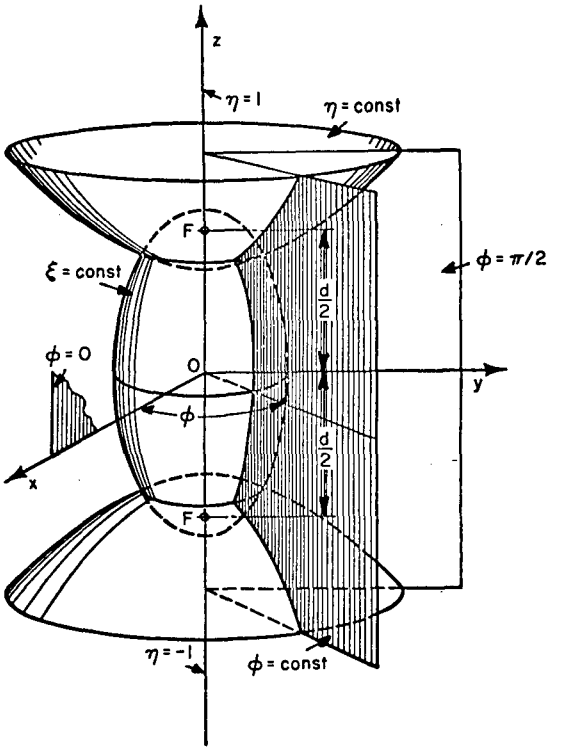

Fig. 1—The prolate spheroidal coordinate system

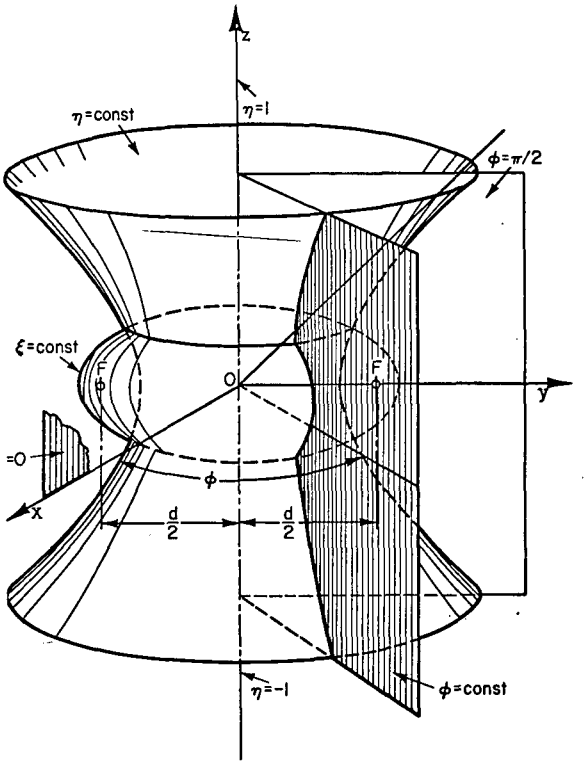

Fig. 2—The oblate spheroidal coordinate system

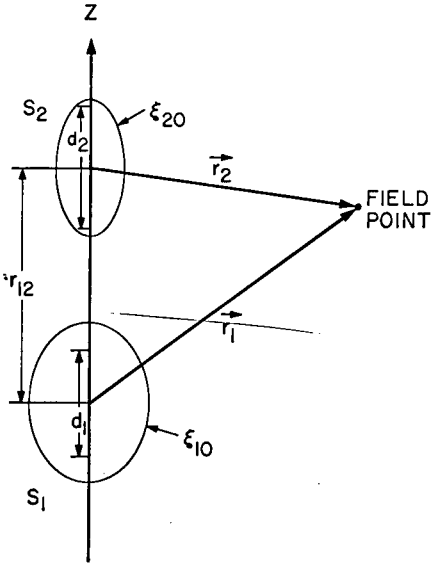

Fig. 3—The two-spheroid geometry

The acoustic pressure produced at a point **r** exterior to both S_1 and S_2 can be obtained by use of the integral formulation [3]

$$p(\mathbf{r}) = \frac{i\omega\rho}{4\pi}\left[\int_{S_1} v_1(\eta_1')G(\mathbf{r},\mathbf{r}')dS(\mathbf{r}') + \int_{S_2} v_2(\eta_2')G(\mathbf{r},\mathbf{r}')dS(\mathbf{r}')\right], \tag{3}$$

where the Green's function $G(\mathbf{r}, \mathbf{r}')$ is the solution to the equation

$$[\nabla^2(\mathbf{r}') + k^2]G(\mathbf{r},\mathbf{r}') = -4\pi\delta(\mathbf{r}' - \mathbf{r}), \tag{4}$$

which satisfies the radiation condition at infinity and whose outward normal derivative $\partial G(\mathbf{r}, \mathbf{r}')/\partial n$ vanishes over both S_1 and S_2. Here $k = 2\pi/\lambda$, where λ is the wavelength. The Green's function is obtained by a procedure analogous to that used by New and Eisler (4) for the two-sphere problem.

First the Green's function is written as a sum of the free-space Green's function, which is the particular solution to Eq. (4), and scattering contributions from both S_1 and S_2, which form the complementary solution and are expressed as eigenfunctions of the homogeneous equation $(\nabla^2 + k^2)\Psi = 0$. Since the normal velocity distribution is chosen to be rotationally symmetrical, only the $m = 0$ terms in the Green's function contribute to the resulting rotationally symmetrical acoustic pressure. Thus, the $m \neq 0$ terms will

UNCLASSIFIED

be dropped, and the effective Green's function is given by

$$G_0(\mathbf{r}, \mathbf{r}') = \left[\frac{e^{-ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r} - \mathbf{r}'|}\right]_{m=0} + \sum_{\ell=0}^{\infty} A_\ell \Psi_{0\ell}^{(4)}(h_1; \xi_1', \eta_1')$$

$$+ \sum_{\ell=0}^{\infty} B_\ell \Psi_{0\ell}^{(4)}(h_2; \xi_2', \eta_2'), \tag{5}$$

where the spheroidal wave functions for $m = 0$ are given by

$$\begin{aligned}\Psi_{0\ell}^{(4)}(h; \xi, \eta) &= R_{0\ell}^{(1)}(h, \xi) S_{0\ell}^{(1)}(h, \eta) \\ &\quad - iR_{0\ell}^{(2)}(h, \xi) S_{0\ell}^{(1)}(h, \eta) \\ &= \Psi_{0\ell}^{(1)}(h; \xi, \eta) - i\Psi_{0\ell}^{(2)}(h; \xi, \eta) .\end{aligned} \tag{6}$$

Here $(\xi_1', \eta_1', \phi_1')$ and $(\xi_2', \eta_2', \phi_2')$ are the spheroidal coordinates of r' in C_1 and C_2, and the acoustic size parameters $h_1 = kd_1/2$ and $h_2 = kd_2/2$. The spheroidal angle wave function of the first kind $S_{0\ell}^{(1)}(h, \eta)$ and the spheroidal radial wave functions of the first and second kind $R_{0\ell}^{(1)}(h, \xi)$ and $R_{0\ell}^{(2)}(h, \xi)$ used in this report are those defined by Flammer [5]. The angle function can be expanded in a series of the corresponding Legendre functions

$$S_{0\ell}^{(1)}(h, \eta) = \sum_{n=0,1}^{\infty}{}' \, d_n(h|0\ell) P_n(\eta), \tag{7}$$

where the prime indicates that the summation is over even or odd values of n according to whether ℓ is even or odd. The radial functions $R_{0\ell}^{(1)}(h, \xi)$ and $R_{0\ell}^{(2)}(h, \xi)$ correspond to spherical Bessel functions of the first and second kinds and are normalized so that $\Psi_{0\ell}^{(4)}(h; \xi, \eta)$ represents outgoing waves for $e^{i\omega t}$ time dependence.

The free-space Green's function can be expanded in terms of spheroidal wave functions in either C_1 or C_2. The $m = 0$ term is given by

$$\left[\frac{e^{-ik|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r} - \mathbf{r}'|}\right]_{m=0} = \sum_{l=0}^{\infty} \left(\frac{-2ik}{N_{0l}}\right) \begin{cases} \Psi_{0l}^{(1)}(h; \xi, \eta)\Psi_{0l}^{(4)}(h; \xi', \eta'), & \xi < \xi' \\ \Psi_{0l}^{(4)}(h; \xi, \eta)\Psi_{0l}^{(1)}(h; \xi', \eta'), & \xi > \xi' \end{cases} \tag{8}$$

where

$$N_{0l} = \int_{-1}^{1} [S_{0l}^{(1)}(h, \eta)]^2 d\eta = \sum_{n=0,1}^{\infty}{}' \frac{2d_n^2(h|0l)}{(2n+1)} . \tag{9}$$

The effective Green's function which results from combining Eqs. (5) and (8) contains spheroidal wave functions in both coordinate systems C_1 and C_2. Application of the boundary conditions on S_1 or S_2 to determine A_ℓ and B_ℓ is difficult if $G_0(\mathbf{r}, \mathbf{r}')$ is not expressed entirely in C_1 or C_2, respectively. Fortunately, this can be accomplished by the use of a spheroidal addition theorem which expresses a spheroidal wave function with reference to one coordinate frame in terms of spheroidal wave functions with reference to a second coordinate frame. The general addition theorem for spheroidal wave functions which is derived in Ref. 1 allows for arbitrary relative position and orientation of the two coordinate frames and is applicable whether the two spheroidal coordinate frames are both prolate, both oblate, or one prolate and one oblate. For the present application, the addition theorem reduces to the following form:

$$\Psi_{0l}^{(4)}(h_i; \xi_i', \eta_i') = \sum_{n=0}^{\infty} C_{ln}^{ij} \Psi_{0n}^{(1)}(h_j; \xi_j', \eta_j'), \tag{10}$$

where

$$C_{\ell n}^{ij} = \sum_{s=0,1}^{\infty}{}' (-1)^{(s-n)/2} \frac{d_s(h_i|0n)}{N_{0n}} \sum_{r=0,1}^{\infty}{}' i^{r-\ell} d_r(h_i|0\ell)$$

$$\times \sum_{t=|r-s|}^{r+s}{}' \frac{2}{2s+1} a(s, t, r, 0, 0) h_t^{(2)}(kr_{12}) P_t(\cos\theta_{ji}) , \tag{11}$$

with the prime sign indicating that the summation is steps of two, and that s or r is taken to be even or odd according to whether n or ℓ is even or odd. The function $h_t^{(2)}(kr_{12})$ is a spherical Hankel function of the second kind. The angles θ_{21} and θ_{12} are equal to 0 and π, respectively, so that $P_t(\cos\theta_{21}) = 1$ and $P_t(\cos\theta_{12}) = (-1)^t$. The coefficient $a(s, t, r, 0, 0)$ is given by

$$a(s, t, r, 0, 0) = \frac{(-1)^\lambda (2s+1)(2t+1)(2\lambda)!(2\lambda_1)!(2\lambda_2)!(\Lambda!)^2}{(2\Lambda+1)!(\lambda!\lambda_1!\lambda_2!)^2} , \tag{12}$$

where

$$\Lambda = \frac{(r+s+t)}{2} ,$$
$$\lambda = \Lambda - r,$$
$$\lambda_1 = \Lambda - s,$$
$$\lambda_2 = \Lambda - t.$$

Use of the addition theorem for $i = 2$ and $j = 1$ leads to the following form of the effective Green's function expressed entirely in C_1:

UNCLASSIFIED

$$G_0(\mathbf{r}, \mathbf{r}') = \sum_{\ell=0}^{\infty} \left(\frac{-2ik}{N_{0\ell}}\right) \Psi_{0\ell}^{(4)}(h_1; \xi_1, \eta_1)\Psi_{0\ell}^{(1)}(h_1; \xi_1', \eta_1')$$

$$+ \sum_{\ell=0}^{\infty} \left[A_\ell \Psi_{0\ell}^{(4)}(h_1; \xi_1', \eta_1')\right.$$

$$\left. + B_\ell \sum_{n=0}^{\infty} C_{\ell n}^{21} \Psi_{0n}^{(1)}(h_1; \xi_1', \eta_1')\right] , \tag{13}$$

where $\xi_1 > \xi_1'$ as required for the present discussion.

Differentiating $G_0(\mathbf{r}, \mathbf{r}')$ with respect to ξ_1', setting this equal to zero at $\xi_1' = \xi_{10}$, multiplying by $S_{0q}^{(1)}(h_1, \eta_1')d\eta_1'$, and integrating over η_1' from -1 to 1, the range of orthogonality for the spheroidal angle wave functions, one obtains the following set of equations:

$$\dot{R}_{0q}^{(4)}(h_1, \xi_{10})A_q + \dot{R}_{0q}^{(1)}(h_1, \xi_{10}) \sum_{\ell=0}^{\infty} C_{lq}^{21} B_l$$

$$= \left(\frac{2ik}{N_{0q}}\right) \dot{R}_{0q}^{(1)}(h_1, \xi_{10})\Psi_{0q}^{(4)}(h_1; \xi_1, \eta_1), \qquad q = 0 \text{ to } \infty, \tag{14a}$$

where the dot indicates the derivative with respect to ξ. The effective Green's function can also be expressed entirely in C_2 in order to apply the boundary condition on S_2. Use of the procedure indicated above leads to the additional set of equations:

$$\dot{R}_{0q}^{(4)}(h_2, \xi_{20})B_q + \dot{R}_{0q}^{(1)}(h_2, \xi_{20}) \sum_{\ell=0}^{\infty} C_{\ell q}^{12} A_\ell$$

$$= \left(\frac{2ik}{N_{0q}}\right) \dot{R}_{0q}^{(1)}(h_2, \xi_{20})\Psi_{0q}^{(4)}(h_2; \xi_2, \eta_2), \qquad q = 0 \text{ to } \infty . \tag{14b}$$

In theory, the doubly infinite set of simultaneous equations obtained by combining Eqs. (14a) and (14b) can be solved for the unknowns A_ℓ and B_ℓ, $\ell = 0$ to ∞. In practice, the summations over ℓ in Eq. (13) are truncated at $\ell = L$ and the resulting $2L + 2$ equations are solved for A_ℓ and B_ℓ, $\ell = 0$ to L. The choice for L depends on the normal velocity distributions $v_1(\eta_1')$ and $v_2(\eta_2')$ and the required accuracy of the results. An estimate for L is given by the highest order that must be retained in the expansion of $v_1(\eta_1')$ and $v_2(\eta_2')$ in terms of the appropriate spheroidal angle wave functions in order to represent them to within the required accuracy. Equations (14a) and (14b) can be written in the truncated form

$$\sum_{\ell=0}^{2L+1} M_{q\ell} b_\ell \doteq g_q, \quad q = 0 \text{ to } 2L + 1, \tag{15}$$

where the coefficients b_ℓ are given by

$$b_\ell = \frac{A_\ell}{2ik} \equiv \hat{A}_\ell \quad \text{for } \ell = 0 \text{ to } L$$

and

$$b_\ell = \frac{B_{\ell-L-1}}{2ik} \equiv \hat{B}_{\ell-L-1} \quad \text{for } \ell = L + 1 \text{ to } 2L + 1 .$$

The matrix elements $M_{q,\ell}$ are given by

$$M_{q\ell} = \begin{cases} \delta_{q\ell}, & \text{for } q,\ell = 0 \text{ to } L \text{ and } q,\ell = L + 1 \text{ to } 2L + 1 \\ \dfrac{\dot{R}^{(1)}_{0q}(h_1, \xi_{10})}{\dot{R}^{(4)}_{0q}(h_1, \xi_{10})} C^{12}_{\ell-L-1,q} & \text{for } q = 0 \text{ to } L; \ell = L + 1 \text{ to } 2L + 1 \\ \dfrac{\dot{R}^{(1)}_{0,q-L-1}(h_2, \xi_{20})}{\dot{R}^{(4)}_{0,q-L-1}(h_2, \xi_{20})} C^{21}_{\ell,q-L-1} & \text{for } q = L + 1 \text{ to } 2L + 1; \ell = 0 \text{ to } L . \end{cases} \tag{16}$$

The right-hand side elements g_q are the expressions

$$\begin{aligned} &\frac{\dot{R}^{(1)}_{0q}(h_1, \xi_{10})}{\dot{R}^{(4)}_{0q}(h_1, \xi_{10}) N_{0q}} \Psi^{(4)}_{0q}(h_1; \xi_1, \eta_1) && \text{for } q = 0 \text{ to } L \\ &\frac{\dot{R}^{(1)}_{0,q-L-1}(h_2, \xi_{20})}{\dot{R}^{(4)}_{0,q-L-1}(h_2, \xi_{20}) N_{0,q-L-1}} \Psi^{(4)}_{0,q-L-1}(h_2; \xi_2, \eta_2) && \text{for } q = L + 1 \text{ to } 2L + 1 . \end{aligned} \tag{17}$$

The symbol $\doteq$ indicates that L is chosen large enough to ensure the desired accuracy in the final results. Note that $\hat{A}_\ell$ and $\hat{B}_\ell$ depend on the field point only through the right-hand side. Equation (15) can be inverted to give

$$b_\ell \doteq \sum_{q=0}^{2L+1} M^{-1}_{\ell q} g_q , \tag{18}$$

UNCLASSIFIED

so that once the inverse elements have been obtained for a given frequency and geometry, $\hat{A}_\ell$ and $\hat{B}_\ell$ and, consequently, the effective Green's function, can be calculated for any field point by simple matrix multiplication.

The acoustic pressure is now evaluated from Eq. (3). Consider the first term, where the area element is given by

$$dS = \left(\frac{d_1{}^2}{4}\right) (\xi_{10}{}^2 - 1)^{1/2}(\xi_{10}{}^2 - \eta_1'{}^2)^{1/2} d\eta_1' d\phi_1' . \tag{19}$$

Expressing the $\mathbf{r}'$ dependence of the effective Green's function entirely in C_1 and integrating, one obtains

$$p_1(\mathbf{r}) \doteq h_1{}^2 \rho c(\xi_{10}^2 - 1)^{1/2} v_{1,\max} \left[\sum_{\ell=0}^{L} R_{0\ell}^{(1)}(h_1, \xi_{10}) I_{0\ell}^{(1)} \Psi_{0\ell}^{(4)}(h_1; \xi_1, \eta_1)/N_{0\ell}(h_1) \right.$$

$$\left. - \sum_{\ell=0}^{L} \hat{A}_\ell R_{0\ell}^{(4)}(h_1, \xi_{10}) I_{0\ell}^{(1)} - \sum_{\ell=0}^{L} \hat{B}_\ell \sum_{n=0}^{L} C_{\ell n}^{21} R_{0n}^{(1)}(h_1, \xi_{10}) I_{0n}^{(1)} \right] , \tag{20}$$

with

$$I_{0\ell}^{(1)} = \int_{-1}^{1} (\xi_{10}^2 - \eta_1'{}^2)^{1/2} \beta_1(\eta_1') S_{0\ell}^{(1)}(h_1, \eta_1') d\eta_1' , \tag{21}$$

where the velocity distribution $v_1(\eta_1')$ has been factored into the product of a normalization $v_{1,\max}$ which is equal to the maximum value on S_1 of the amplitude of $v_1(\eta_1')$ and a dimensionless distribution function $\beta_1(\eta_1')$. Note that the term $N_{0\ell}(h_1)$ depends on the acoustic size parameter h_1.

These expressions give the acoustic pressure when S_1 is vibrating and $v_2(\eta_2')$ is identically zero, so that S_2 acts only as a rigid scatterer. When S_2 is vibrating also, the second term in Eq. (3) must be included. Expressing the $\mathbf{r}'$ dependence of the effective Green's function entirely in C_2 and integrating, one obtains

$$p_2(\mathbf{r}) \doteq h_2^2 \rho c(\xi_{20}^2 - 1)^{1/2} v_{2,\max} \left[\sum_{\ell=0}^{L} R_{0\ell}^{(1)}(h_2, \xi_{20}) I_{0\ell}^{(2)} \Psi_{0\ell}^{(4)}(h_2; \xi_{20}, \eta_2)/N_{0\ell}(h_2) \right.$$

$$\left. - \sum_{\ell=0}^{L} \hat{B}_\ell R_{0\ell}^{(4)}(h_2, \xi_{20}) I_{0\ell}^{(2)} - \sum_{\ell=0}^{L} \hat{A}_\ell \sum_{n=0}^{L} C_{\ell n}^{12} R_{0n}^{(1)}(h_2, \xi_{20}) I_{0n}^{(2)} \right] , \tag{22}$$

where the integrals $I_{0\ell}^{(2)}$ are given by Eq. (21) with the superscript 1 replaced by the subscript 2. Thus the total acoustic pressure $p(\mathbf{r})$, when S_1 is vibrating with the velocity distribution $v_1(\eta_1')$ and S_2 is vibrating with the velocity distribution $v_2(\eta_2')$, is given by

$$p(\mathbf{r}) = p_1(\mathbf{r}) + p_2(\mathbf{r}) . \tag{23}$$

The acoustic pressure in the far field can be obtained from Eqs. (20) and (22) with the use of the limiting form for the spheroidal wave functions,

$$R_{0\ell}^{(4)}(h, \xi) \xrightarrow[\xi \to \infty]{} \frac{e^{-ih\xi}}{h\xi} e^{i(\pi/2)(\ell+1)} . \tag{24}$$

In the following discussion the band-shaped region on the spheroidal surface S_1 bounded by the curves $\eta = \eta_{1L}$ and $\eta = \eta_{1U}$ (see Fig. 4) is assumed to vibrate with uniform normal velocity V_1, while the remainder of S_1 and all of S_2 are assumed to be rigid, i.e., $v_{1,\max} = V_1$ and $\beta_1(\eta_1)$ is equal to unity for $\eta_{1L} \leqslant \eta_1 \leqslant \eta_{1U}$ and equal to zero otherwise. For the case of uniform vibration, the normalized acoustic radiation self-impedance z_{11} is defined by

$$z_{11} = r_{11} + ix_{11} = \left(\frac{1}{\rho c V_1 \mathcal{A}_1}\right) \int_{\eta_{1L}}^{\eta_{1U}} p_1(\mathbf{r}) dS(\mathbf{r}) , \tag{25}$$

where the real quantities r_{11} and x_{11} are the normalized acoustic radiation self-resistance and the normalized acoustic radiation self-reactance, respectively, and where $\mathcal{A}_1$ is the area of the vibrating region. The less descriptive names *radiation impedance*, *radiation resistance*, and *radiation reactance* will be used in the remainder of the report. Note that the use of Eq. (25) requires a knowledge of the acoustic pressure over the surface S_1. Although the free-space Green's function and, consequently, the effective Green's function is singular at the point $\mathbf{r} = \mathbf{r}'$ so that the expansions given in Eq. (8) do not converge at that point, the singularity is integrable, and the resulting expression for the acoustic pressure given in Eq. (20) converges to the correct value for field points on S_1. Expressing $p_1(\mathbf{r})$ entirely in C_1 by use of the addition theorem in Eq. (10) and integrating, one obtains

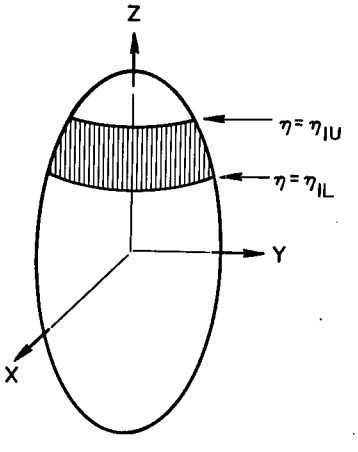

Fig. 4—Velocity band on spheroid 1

UNCLASSIFIED

$$z_{11} \doteq \frac{h_1^2(\xi_{10}^2 - 1)^{1/2}}{W_1} \left[\sum_{\ell=0}^{L} R_{0\ell}^{(1)}(h_1, \xi_{10}) R_{0\ell}^{(4)}(h_1, \xi_{10}) I_{0\ell}^{(1)} I_{0\ell}^{(1)} / N_{0\ell}(h_1) \right.$$

$$\left. - \sum_{\ell=0}^{L} R_{0\ell}^{(4)}(h_1, \xi_{10}) I_{0\ell}^{(1)} T_{\ell}^{(1)} - \sum_{\ell=0}^{L} T_{\ell+L+1}^{(1)} \sum_{n=0}^{L} C_{\ell n}^{21} R_{0n}^{(1)}(h_1, \xi_{10}) I_{0n}^{(1)} \right], \quad (26)$$

where

$$T_{\ell}^{(1)} = \sum_{j=0}^{L} \left[M_{\ell j}^{-1} \frac{\dot{R}_{0j}^{(1)}(h_1, \xi_{10}) R_{0j}^{(4)}(h_1, \xi_{10})}{\dot{R}_{0j}^{(4)}(h_1, \xi_{10}) N_{0j}(h_1)} I_{0j}^{(1)} \right.$$

$$\left. + M_{\ell, j+L+1}^{-1} \frac{\dot{R}_{0j}^{(1)}(h_2, \xi_{20})}{\dot{R}_{0j}^{(4)}(h_2, \xi_{20}) N_{0j}(h_2)} \sum_{r=0}^{L} C_{jr}^{21} R_{0r}^{(1)}(h_1, \xi_{10}) I_{0r}^{(1)} \right], \quad (27)$$

and

$$W_1 = \int_{\eta_{1L}}^{\eta_{1U}} (\xi_{10}^2 - \eta_1^2)^{1/2} d\eta . \quad (28)$$

Note that z_{11} does not depend on the magnitude V_1 of the normal velocity but does depend on the velocity distribution β_1.

Similarly, the force produced by $p_1(\mathbf{r})$ on that part of the spheroid S_2 bounded by the curves $\eta = \eta_{2L}$ and $\eta = \eta_{2U}$ can be calculated and used to obtain the normalized acoustic radiation mutual impedance z_{21}. This gives

$$z_{21} = r_{21} + ix_{21} = \left(\frac{1}{\rho c V_1 \mathcal{A}_2} \right) \int_{\eta_{2L}}^{\eta_{2U}} p_1(\mathbf{r}) dS(\mathbf{r}) , \quad (29)$$

where the real quantities r_{21} and x_{21} are called the normalized acoustic radiation mutual resistance and the normalized acoustic radiation mutual reactance, respectively, and where $\mathcal{A}_2$ is the area of the integration on S_2. Using the addition theorem to express $p_1(\mathbf{r})$ entirely in C_2 and integrating, one obtains

$$z_{21} \doteq \frac{h_1^2(\xi_{10}^2 - 1)^{1/2}}{W_2} \left[\sum_{\ell=0}^{L} \left\{ R_{0\ell}^{(1)}(h_1, \xi_{10}) I_{0\ell}^{(1)} / N_{0\ell}(h_1) \sum_{n=0}^{L} C_{\ell n}^{12} \right. \right.$$

$$\left. \times R_{0n}^{(1)}(h_2, \xi_{20}) I_{0n}^{(2)} \right\} - \sum_{\ell=0}^{L} T_{\ell}^{(2)} R_{0\ell}^{(4)}(h_1, \xi_{10}) I_{0\ell}^{(1)} \quad \text{(30) Continued}$$

(30) Continued

$$- \sum_{\ell=0}^{L} T_{\ell+L+1}^{(2)} \sum_{n=0}^{L} C_{\ell n}^{21} R_{0n}^{(1)}(h_1, \xi_{10}) I_{0n}^{(1)} \Bigg] ,$$

where the expression for W_2 is obtained from Eq. (28) by the interchange of the index 2 for the index 1, and where

$$T_{\ell}^{(2)} = \sum_{j=0}^{L} \Bigg[M_{\ell j}^{-1} \frac{\dot{R}_{0j}^{(1)}(h_1, \xi_{10})}{\dot{R}_{0j}^{(4)}(h_1, \xi_{10}) N_{0j}(h_1)} \sum_{r=0}^{L} C_{jr}^{12} R_{0r}^{(1)}(h_2, \xi_{20}) I_{0r}^{(2)}$$

$$+ M_{\ell, j+L+1}^{-1} \frac{\dot{R}_{0j}^{(1)}(h_2, \xi_{20}) R_{0j}^{(4)}(h_2, \xi_{20})}{\dot{R}_{0j}^{(4)}(h_2, \xi_{20}) N_{0j}(h_2)} I_{0j}^{(2)} \Bigg] . \tag{31}$$

It is now assumed that the region of the spheroidal surface S_2 bounded by the curves $\eta = \eta_{2L}$ and $\eta = \eta_{2U}$ vibrates with uniform normal velocity V_2, while the remainder of S_2 and all of S_1 are assumed to be rigid, i.e., $v_{2,\max} = V_2$ and $\beta_2(\eta_2)$ is equal to unity for $\eta_{2L} \leqslant \eta_2 \leqslant \eta_{2U}$ and equal to zero otherwise. In this case the normalized acoustic radiation self-impedance z_{22} is defined by Eq. (25) with the index 1 replaced everywhere by the index 2. Expressing $p_2(\mathbf{r})$ entirely in C_2 and integrating, one obtains

$$z_{22} \doteq \frac{h_2^2(\xi_{20}^2 - 1)^{1/2}}{W_2} \Bigg[\sum_{\ell=0}^{L} R_{0\ell}^{(1)}(h_2, \xi_{20}) R_{0\ell}^{(4)}(h_2, \xi_{20}) I_{0\ell}^{(2)} I_{0\ell}^{(2)} / N_{0\ell}(h_2)$$

$$- \sum_{\ell=0}^{L} T_{\ell}^{(2)} \sum_{n=0}^{L} C_{\ell n}^{12} R_{0n}^{(1)}(h_2, \xi_{20}) I_{0r}^{(2)} - \sum_{\ell=0}^{L} R_{0\ell}^{(4)}(h_2, \xi_{20}) I_{0\ell}^{(2)} T_{\ell+L+1}^{(2)} \Bigg] . \tag{32}$$

Similarly, the normalized acoustic radiation mutual impedance z_{12} is obtained by expressing $p_2(\mathbf{r})$ entirely in C_1 and integrating over the region $\eta_{1L} \leqslant \eta_1 \leqslant \eta_{1U}$ on S_1. The result is

$$z_{12} \doteq \frac{h_2^2(\xi_{20}^2 - 1)^{1/2}}{W_1} \Bigg[\sum_{\ell=0}^{L} \Bigg\{ R_{0\ell}^{(1)}(h_2, \xi_{20}) I_{0\ell}^{(2)} / N_{0\ell}(h_2) \sum_{n=0}^{L} C_{\ell n}^{21} R_{0n}^{(1)}(h_1, \xi_{10}) I_{0n}^{(1)} \Bigg\}$$

$$- \sum_{\ell=0}^{L} T_{\ell}^{(1)} \sum_{n=0}^{L} C_{\ell n}^{12} R_{0n}^{(1)}(h_2, \xi_{20}) I_{0n}^{(2)} - \sum_{\ell=0}^{L} T_{\ell+L+1}^{(1)} R_{0\ell}^{(4)}(h_2, \xi_{20}) I_{0\ell}^{(2)} \Bigg] . \tag{33}$$

Using the reciprocity of the Green's function (i.e., it is symmetrical to an interchange of the source and the field points), one can show that the normalized acoustic radiation mutual impedance z_{12} is related to the mutual impedance z_{21} by

UNCLASSIFIED

$$z_{12}\mathcal{A}_1 = z_{21}\mathcal{A}_2 . \tag{34}$$

Consequently, in PSPRDS and OSPRDS the impedance z_{21} is obtained by multiplying by $\mathcal{A}_1/\mathcal{A}_2$ the value for z_{12} calculated from Eq. (33).

The four impedance terms z_{11}, z_{21}, z_{12}, and z_{22} are defined to be independent of the normal velocity magnitudes but dependent on the velocity distributions. Therefore, they characterize the two spheroid systems when the region $\eta_{1L} \leqslant \eta_1 \leqslant \eta_{1U}$ on S_1 or the region $\eta_{2L} \leqslant \eta_2 \leqslant \eta_{2U}$ on S_2 is vibrating with a unit normal velocity. Because of linear superposition, the total acoustic force F_1 acting on the vibrating region of S_1 when both S_1 and S_2 are vibrating with normal velocities V_1 and V_2, respectively, can be obtained from z_{11} and z_{12} by use of

$$F_1 = \rho c\mathcal{A}_1(z_{11}V_1 + z_{12}V_2) . \tag{35}$$

Similarly, the total acoustic force acting on the vibrating part of S_2 is given by

$$F_2 = \rho c\mathcal{A}_2(z_{22}V_2 + z_{21}V_1) . \tag{36}$$

THE COMPUTER PROGRAMS PSPRDS AND OSPRDS

The computer programs PSPRDS and OSPRDS are both written in CDC 3600 Fortran for the CDC 3800 computer at the Naval Research Laboratory. Program PSPRDS calculates acoustic radiation from two prolate spheroids, while OSPRDS calculates acoustic radiation from two oblate spheroids. Appendixes A and B contain listings of these programs.

Single-precision numbers in the CDC 3800 have 11 decimal digits, while double-precision numbers have 26 decimal digits. The magnitude of numbers ranges from 10^{-307} to 10^{+307}. These properties are used in PSPRDS and OSPRDS for scaling certain functions and tests on convergence of various summations.

BASIC STRUCTURE OF THE COMPUTER PROGRAMS

The computer programs PSPRDS and OSPRDS are logically similar. Each contains a large main program (over 1100 lines) and several subroutines. The main program consists of four distinct parts: (a) parameter input, (b) spheroidal wave functions calculation, (c) matrix calculation, and (d) radiation calculations and output.

Parameter Input

The main function of this part is to read in all data cards necessary to execute the program. The data cards provide all necessary information with the exception of three parameters. These three parameters are given the following values in the main program:

Parameter	*Value*	*Description*
RHO	1000	Density of the fluid medium in kg/m^3

CEE	1500	Speed of sound in the medium in m/s
ICXI	26	Sum parameter (see Matrix Calculation)

These values of RHO and CEE correspond to water and may be changed by the user if other fluids are desired. The value of ICXI should not be changed.

This part also contains calculations of the limiting form for the spheroidal wave functions (see Eq. (24)) to be used for far-field calculations.

A diagram of the arrangement of data cards for the computer programs PSPRDS and OSPRDS is shown in Fig. 5. This figure is followed by a detailed description of each type of data card.

Card 1. Summation Control Card

Variable	*Format*	*Column*	*Description*
IDEM	I5	1 to 5	Number of terms summed in the impedance and pressure equations, i.e., Eqs. (20), (22), (26), (32), (33). It is equivalent to $L + 1$. Its maximum value is 30.
IDEMPI	I5	6 to 10	Maximum number of terms summed in the inner sum over r of Eq. (11) for the imaginary part of the matrix M. It has a maximum value of 100.
IDEMPR	I5	11 to 15	Maximum number of terms summed in the inner sum over r of Eq. (11) for the real part of the matrix M. It has a maximum value of 40.

Card 2. Geometry for Spheroid 1

Variable	*Format*	*Column*	*Description*
X for oblate Z for prolate	D10.5	1 to 10	Value of the shape parameter ξ_{10} where $0 \leqslant \xi_{10} < \infty$, when spheroid 1 is oblate. Value of $\xi_{10} - 1$ where $1 \leqslant \xi_{10} < \infty$, when spheroid 1 is prolate.
H	D10.5	11 to 20	Acoustic size parameter h_1 of spheroid 1.

UNCLASSIFIED

The expression $\sqrt{(\xi_{10} + 1)(\xi_{10} - 1)}$ is required in PSPRDS. Since values of ξ_{10} for prolate spheroids can be very close to 1, say 1.0000001, $\xi_{10} - 1$ is the input. This avoids a subtraction error and subsequent loss of significant figures which would occur if $\xi_{10} - 1$ were calculated from ξ_{10} in the program.

Card 3. Geometry for Spheroid 2

The input format for the geometry of spheroid 2 is identical to that of the geometry for spheroid 1. Note that if the two spheroids have different values for the acoustic size parameter H (h in the theory), the larger value should be associated with spheroid 1 since more terms can be used in the inner summation (up to 100 terms) for matrix M than in the outer summation (up to 26 terms).

Card 4. Spheroid Separation

Variable	*Format*	*Column*	*Description*
ALPHA	D10.5	1 to 10	Parameter α gives the distance r_{12} between the centers of the two spheroids in units of $d_1/2$, i.e., $r_{12} = (d_1/2)\alpha$. The corresponding acoustic distance $kr_{12} = h_1\alpha$ where h_1 is the value of H input on Card 2.

Note that the two spheroids must not overlap. This requires that $\alpha > \xi_{10} + \xi_{20}h_2/h_1$.

Card 5. Velocity Value and Location on Spheroid 1

Variables	*Format*	*Column*	*Description*
IVEL1	I1	1	Integer parameter set equal to 0 if the normal velocity is equal to 0, set unequal to 0 if the normal velocity is not equal to 0.
VEL1	C(F9.3,F10.3)	2 to 20	Complex value of the normal velocity on the vibrating band on spheroid 1.
THETAL1	D10.5	21 to 30	Lower boundary of the vibrating band in degrees, i.e., THETAL1 = $\cos^{-1}\eta_{1L}$.
THETAU1	D10.5	31 to 40	Upper boundary of the vibrating band in degrees, i.e., THETAU1 = $\cos^{-1}\eta_{1U}$.

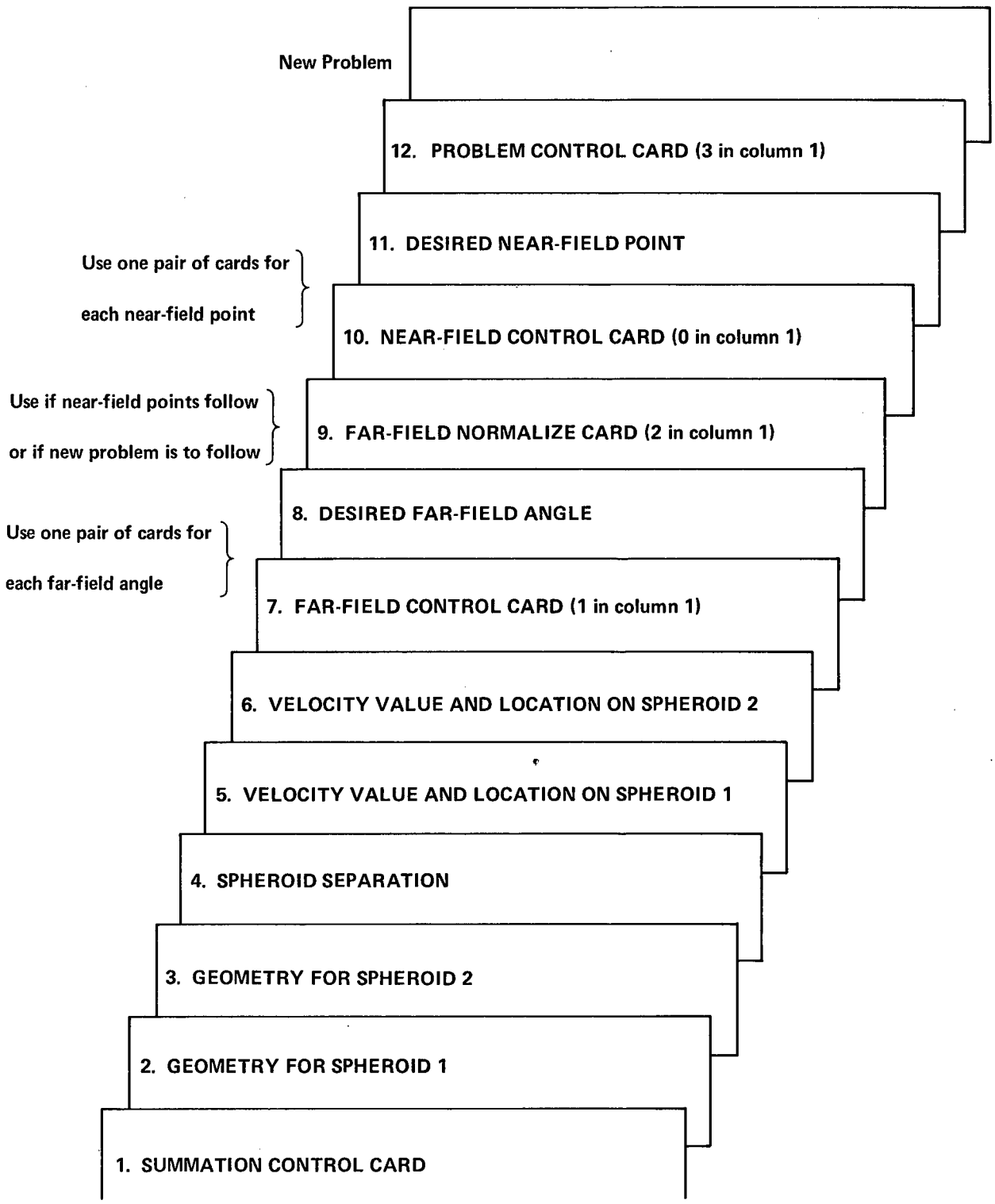

Fig. 5—Input data cards

UNCLASSIFIED

Card 6. Velocity Value and Location on Spheroid 2

The input here is similar to that of Card 5. The corresponding variable names are IVEL2, VEL2, THETAL2, and THETAU2.

Card 7. Far-field Control Card

Variable	*Format*	*Column*	*Description*
NFF	I1	1	Equal to 1

This card announces a far-field calculation and allows the program to use the far-field asymptotic forms for the spheroidal radial wave functions. It must precede Card 8.

Card 8. Desired Far-field Angle

Variable	*Format*	*Column*	*Description*
E	D10.5	1 to 10	Far-field point in degrees; $0° \leqslant E \leqslant 180°$. E corresponds to the polar angle, i.e., the angle that the far-field direction makes with the positive z axis (taken to be directed from spheroid 1 to spheroid 2).

Program dimensioning allows for a maximum of 40 far-field points. This can be changed by adjusting the dimensions of the variables ANGLE, FF, FFA, and FFM.

Card 9. Far-field Normalize Card

Variable	*Format*	*Column*	*Description*
NFF	I1	1	Equal to 2

This card is used if both far-field and near-field points are input. It causes the far-field values to be normalized by the value in the direction of the first input far-field angle and printed out. The card is not necessary if only far-field points are to be calculated and only one problem is being run.

Card 10. Near-field Control Card

Variable	*Format*	*Column*	*Description*
NFF	I1	1	Equal to 0 (or blank)

This card announces a near-field calculation and must precede Card 11.

Card 11. Desired Near-field Point

Variable	*Format*	*Column*	*Description*
XCT	D10.5	1 to 10	x coordinate of point in units of $d_1/2$
YCT	D10.5	11 to 20	y coordinate of point in units of $d_1/2$
ZCT	D10.5	21 to 30	z coordinate of point in units of $d_1/2$

This card describes the near-field point having Cartesian coordinates x, y, z with respect to the Cartesian coordinate system centered at the origin of spheroidal system C_1 (ξ_1, η_1, ϕ_1) and having the z axis as shown in Fig. 3. The coordinates are in units of $d_1/2$, e.g., the point $(0, 0, \alpha/2)$ represents a point midway between the centers of the two spheroids on the line connecting them. Care must be taken to insure that the near-field point is exterior to both spheroidal surfaces.

Note that due to symmetry about the z axis, the acoustic pressures at points (a, b, c) and (b, a, c) are equal.

Card 12. Problem Control Card

A problem control card having the number 3 punched in column 1 is used to separate different problems. This enables any number of completely independent problems to be solved successively. Each separate problem must be described by a full complement of data cards.

Spheroidal Functions Calculation

Accurate values for the spherodial wave functions are obtained using PRAD (6), OBRAD (7), and ANGLEN (8), Fortran computer programs written in double-precision arithmetic for the CDC 3800 computer. The second part of PSPRDS contains the computational sections of PRAD and ANGLFN and requires the subroutines CALCPQ, POLY, SBESF, and EIGEN. Similarly the second part of OSPRDS contains the computational sections of OBRAD and ANGLFN and requires the subroutines POLY, SBESF, SPHYN, QLEG, and EIGEN. Double precision is maintained for these calculations although the resulting wave function values are truncated to single precision for later calculations. A printout of the spheroidal radial wave functions will occur if the Wronskian accuracy check indicates that the accuracy of at least one of the four functions $R_{0\ell}^{(1)}(h, \xi)$, $\dot{R}_{0\ell}^{(1)}(h, \xi)$, $R_{0\ell}^{(2)}(h, \xi)$, or $\dot{R}_{0\ell}^{(2)}(h, \xi)$ is less than five significant figures. (See the discussion of the Wronskian accuracy check in the introduction in Refs. 9 or 10.) The statements controlling this printout are

```
PRINT 933, L, R1, R1D, R2, R2D, CL, IAC

933 FORMAT (1X*RADIAL FUNCTION ACCURACY WARNING*5XI3,

4D15.5, D20.10, I4)
```

The Fortran variable names R1, R1D, R2, and R2D correspond to $R_{0\ell}^{(1)}(h, \xi)$, $\dot{R}_{0\ell}^{(1)}(h, \xi)$, $R_{0\ell}^{(2)}(h, \xi)$, and $\dot{R}_{0\ell}^{(2)}(h, \xi)$, respectively. The variable IAC indicates the minimum number of accurate digits in the radial functions. CL is the associated eigenvalue.

UNCLASSIFIED

The calculation of the spheroidal wave functions makes occasional usage of the following special *DO* loop property of 3600 Fortran: for the loop *DO n i* = 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*.

Matrix Calculation

This part of the main program contains the calculation of the matrix M, whose complex elements $M_{q\ell}$ are defined in Eq. (16). It can be shown that the addition theorem coefficients for the transformation from the coordinate system S_2 to the coordinate system S_1 are equal to the transposed coefficients for the inverse transformation from S_1 to S_2, i.e., $C^{12}_{mn} = C^{21}_{nm}$. Therefore, it is necessary to calculate only the elements C^{12}_{mn} of the transformation matrix C^{12}, as defined in Eq. (11), in order to generate the matrix M.

The real and imaginary parts of C^{12} are calculated separately using real arithmetic. This is done because the convergence of the imaginary parts (due to the presence of spherical Neumann functions) will be much slower than the convergence of the real parts (which contain spherical Bessel functions) whenever the argument kr_{12} is small. The user can choose, within limits, the maximum number of terms used in the summation over r in Eq. (11) for both the imaginary and the real part summations by means of the input parameters IDEMPI and IDEMPR. The outer or s summation limit is fixed at 26 by the parameter ICXI. The imaginary part of C^{12} is calculated before the real part.

The program attempts to obtain elements of C^{12} and, consequently, M accurate to more than four significant digits. However, elements whose accuracy is such that the absolute value of the ratio of the last contribution to the existing summation is less than 10^{-2} are also accepted. Elements are calculated consecutively by rows, i.e., C^{12}_{00}, C^{12}_{01}, . . . , C^{12}_{0L}, C^{12}_{10}, etc. Any element which fails to satisfy the minimum accuracy requirement is given the value zero. If, for imaginary part calculations, such an element is in the first row, say C^{12}_{0N}, the maximum index of C^{12} is reduced to $N - 1$ so as not to include that element or any other element C^{12}_{mn}, $m > N - 1$ or $n > N - 1$. Printout statements in the program inform the user whenever elements of C^{12} have been given the value zero. Finally, the matrix elements $M_{q\ell}$ are formed using the elements of C^{12} and values for the necessary spheroidal radial wave functions.

Radiation Calculations and Output

The integrals $I_{0\ell}$, with $v(\eta) = V$ for $\eta_L < \eta < \eta_U$ and $v(\eta) = 0$ for η elsewhere, are calculated in the subroutine INTEGER. The evaluation is by expansion of the spheroidal angle wave functions in a series of Legendre polynomials as shown in Eq. (7), subsequent expansion of the Legendre polynomials in a series in terms of cos $(r\theta)$ where $\theta = \cos^{-1}\eta$, and computation of the resulting integrals

$$J_r = \int_{\eta_L}^{\eta_U} (\xi_0^2 - \eta^2)^{1/2} \cos(r\theta) d\eta , \quad r = 0, 1, \ldots \tag{37}$$

by use of Gaussian quadrature.

The variable SACK refers to the innermost summation required for $C_{\ell n}^{12}$ as given in Eq. (11). The time in seconds to calculate this summation and store all the terms as a function of the outer indices s and r is printed in the output.

The times in seconds to calculate both the imaginary and the real parts of all of the matrix elements $C_{\ell n}^{12}$, ℓ, $n = 0$ to L (in the program $L + 1$ is called IDEM) are printed in the output under the names CXLIN and CXLRN, respectively.

The matrix M is now inverted in the subroutines CINV and MINC. The inverse matrix M^{-1} of Eq. (18) need be calculated only once for each geometrical configuration. The total time to obtain this matrix may vary from only a few seconds to up to 4 min depending upon IDEM, IDEMPI, and IDEMPR, with IDEM being the most significant parameter. The radiation impedances z_{11}, z_{12}, and z_{22} are then calculated from Eqs. (26), (33), and (32), respectively, and printed. Each of the three running summations in Eqs. (26), (33), and (32) are also printed so that the user can judge their convergence. Written in terms of these summations, Eqs. (26), (33), and (32) are given by

$$z_{11} = \frac{h_1^2(\xi_{10}^2 - 1)^{1/2}}{W_1} (\text{SIMP1} - \text{SIMP2} - \text{SIMP3}) , \tag{38}$$

$$z_{12} = \frac{h_2^2(\xi_{20}^2 - 1)^{1/2}}{W_2} (\text{SIMP4} - \text{SIMP5} - \text{SIMP6}) , \tag{39}$$

$$z_{22} = \frac{h_2^2(\xi_{20}^2 - 1)^{1/2}}{W_2} (\text{SIMP7} - \text{SIMP8} - \text{SIMP9}) . \tag{40}$$

The mutual radiation impedance z_{21} is then obtained from z_{12}.

Similarly, running summations of Eqs. (20) and (22) are provided for both far-field and near-field pressure calculations. The summations are defined as follows:

$$p_i(\mathrm{r}) = h_i^2 \rho c(\xi_{i0}^2 - 1)^{1/2} \left[\text{SM1} - \text{SM2} - \text{SM3} \right] , \quad i = 1, 2. \tag{41}$$

The time required to calculate the pressure at one point depends upon the value of IDEM. The time is of the order of a few seconds. Note that the asymptotic form for $R_{0\ell}^{(4)}(h_1, \xi_1)$ given by Eq. (24) with the factor $e^{-ih_1\xi_1}/(h_1\xi_1)$ suppressed is used in Eqs. (20) and (22) for far-field calculations. The corresponding asymptotic form for $R_{0\ell}^{(4)}(h_2, \xi_2)$ about the origin of S_1 is given by

$$R_{0\ell}^{(4)}(h_2, \xi_2) \underset{\xi\to\infty}{\longrightarrow} \frac{e^{-ih_1\xi_1}}{h_1\xi_1} e^{ih_1\alpha\cos\theta} e^{i(\pi/2)(\ell+1)} , \tag{42}$$

where the additional factor represents the effect in the farfield of the separation between the two spheroids. Again, the factor $\exp(-ih_1\xi_1)/(h_1\xi_1)$ is suppressed.

The resulting complex far-field pressure is then multiplied by $\exp[-i(h_1/2)\alpha\cos\theta]$ to adjust it to an origin midway between the centers of the two spheroids. Finally, all

UNCLASSIFIED

far-field pressures are divided by the pressure at the first angle read in, i.e., they are normalized so that the pressure at the first angle is unity.

The input and output for a sample run are given in Appendix C.

ACCURACY OF THE RESULTS

The acoustic impedance and pressure values should be accurate to at least two or three digits provided that the outer summations, i.e., the summations over ℓ in Eqs. (20), (22), (26), (32), and (33), have converged. These summations are displayed as part of the output (see Appendix C) and can be examined for convergence.

Parameter limitations due to inaccurate spheroidal functions are based on computer programs PRAD, OBRAD, and ANGLFN. Programs PSPRDS and OSPRDS will display values for all spheroidal radial functions whose accuracy is less than five decimal digits. Although accuracy of more than four or five digits in the radial functions should be sufficient, accuracy of less than four digits makes final impedance and pressure results suspect. There is no analogous printout for the spheroidal angle functions. The following statement taken from NRL Report 7161 (Ref. 8) indicates the accuracy to be expected in the corresponding angle function.

"Within the following ranges, the angle functions and their derivatives are believed to be accurate to at least eight significant figures, assuming calculations are performed by a computer that has a word length of 26 digits, e.g., the Control Data Corporation 3800 computer at NRL using double-precision arithmetic:

$$0 \leqslant M \leqslant 6$$

$$M \leqslant L \leqslant M + 49$$

PROLATE

$0.1 \leqslant H \leqslant 20.0$	$20.0 < H \leqslant 35.0$	$35.0 < H \leqslant 40.0$
$-1.0 \leqslant \eta \leqslant 1.0$	$0 \leqslant \lvert\eta\rvert \leqslant 0.99,\ \lvert\eta\rvert = 1.0$	$0 \leqslant \lvert\eta\rvert \leqslant 0.9$

OBLATE

$0.1 \leqslant H \leqslant 30.0$	$30.0 < H \leqslant 40.0$
$-1.0 \leqslant \eta \leqslant 1.0$	$0.2 \leqslant \lvert\eta\rvert \leqslant 1.0$ "

REFERENCES

1. B.J. King and A.L. Van Buren, "A General Addition Theorem for Spheroidal Wave Functions," *SIAM J. Math. Anal.* 4, 149-160 (1973).

2. A.L. Van Buren and B.J. King, "Acoustic Radiation from Two Spheroids," *J. Acoust. Soc. Amer.* 52, *364*-372 (1972).

3. P.M. Morse and K.U. Ingard, *Theoretical Acoustics*, McGraw-Hill, New York, 1968.

4. R. New, and T.J., Eisler, "Acoustic Radiation from Multiple Spheres," *J. Sound and Vibration* 22, 1-17 (1972).

5. C. Flammer, *Spheroidal Wave Functions*, Stanford University Press, Stanford, Calif., 1957.

6. B.J. King, R.V. Baier, and S. Hanish, "A Fortran Computer Program for Calculating the Prolate Spheroidal Radial Functions of the First and Second Kind and Their First Derivatives," NRL Report 7012, Mar. 1970.

7. A.L. Van Buren, R.V. Baier, and S. Hanish, "A Fortran Computer Program for Calculating the Oblate Spheroidal Radial Functions of the First and Second Kind and Their First Derivatives," NRL Report 6959, Jan. 1970.

8. B.J. King and A.L. Van Buren, "A Fortran Computer Program for Calculating the Prolate and Oblate Angle Functions of the First Kind and Their First and Second Derivatives," NRL Report 7161, Nov. 1970.

9. S. Hanish, R.V. Baier, A.L. Van Buren, and B.J. King, "Tables of Radial Spheroidal Wave Functions, Vols. 1-3, Prolate, $m = 0, 1, 2$," NRL Reports 7088-7090, June 1970.

10. S. Hanish, R.V. Baier, A.L. Van Buren, and B.J. King, "Tables of Radial Spheroidal Wave Functions, Vols. 4-6, Oblate, $m = 0, 1, 2$," NRL Reports 7091-7093, June 1970.

UNCLASSIFIED

Appendix A

FORTRAN LISTING OF PSPRDS, A COMPUTER PROGRAM TO CALCULATE THE AXISYMMETRIC RADIATION FROM TWO COAXIAL PROLATE SPHEROIDS

```
      PROGRAM PSPRDS
      TYPE COMPLEX CTS,CX,DACS,DCS,FF,R41,R42,R1D4,R2D4,R1F4,R2F4,
     $SIMP1,SIMP2,SIMP3,SIMP4,SIMP5,SIMP6,SIMP7,SIMP8,SIMP9,SM1,SM2,SM3,
     $STOR1,STOR2,SUM,SUMM,SUM1,SUM2,SVM,TRS1,TRS2,VEL1,VEL2
      TYPE DOUBLE AAA,ALPHA,BSF,CRT,CRZ,CRZZ,DTK,E,EIGL1,EIGL2,ETA1,
     $ETA2,ETAL1,ETAL2,ETAU1,ETAU2,FBCT,HS1,HS2,P,PI,S,TERM,XCT,XIL,
     $XS1,XS2,YCT,YMF,ZCT,THETAL1,THETAU1,THETAL2,THETAU2
      TYPE DOUBLE   ARR,ARRAY,BAY,BLIST,BOOK,BRRAY,C101,CFAC,
     1CL,CORA,CORB,CWRON,DE,DH,DL,DLAST,DLIST,DM,DN,DND0,DNEG,DNEGF,
     2DNFREE,DNUM,DR,DRFREE,DRRATIO,DX,EA,EM,ENR,ENRC,EYE,FACT,
     3GLIST,H,H1,HSQ,PAPER,PL1,PL3,PL4,PL8,PL12,PL13,PL17,PL18,Q,R1,
     4R1D,R2,R2D,RR2,RR2D,SAIL3,SAIL4,SAIL18,
     5SAIL19,SAIL22,SAIL23,SAIL181,SAIL191,SHIP4,
     6SHIP5,SHIP7,SHIP10,SHIP11,SHIP12,SHORE,SILAND,
     7SSUM1,SUBSUM,TWRON,UN,VN,W,W26,W28,WN,X,X1SQ,XTH,Z,Z1
      DIMENSION ANGLE(40),BACT(560),BSF(250),CTS(60),CX(60,60),DACS(60),
     $DCS(60),DDL1(30,110),DDU2(30, 30),EIGL1(30),EIGL2(30),FBCT(300),
     $FF(40),FFA(40),FFM(40),P(125),R11(30),R41(30),R12(30),
     $R42(30),R1D1(30),R1D4(30),R2D1(30),R2D4(30),R1F1(30),R1F4(30),
     $R2F1(30),R2F4(30),S1(30),S2(30),SACKI(8850),
     $SACKR(2780),SIRL1(30),SIRL2(30),SPDINT(2),STORI(30),STORR(30),
     $TRS1(30),TRS2(30),XIL(60),XNML1(30),XNML2(30),YMF(300)
      DIMENSION A(70,70),ARRAY(300),BAY(100),BLIST(150),BOOK(300),
     $BRRAY(250),DLAST(150),DLIST(150),DM(150),DN(150),DNEG(1),
     $DND0(150),DR(100),DRRATIO(100),EIG(70),ENR(150),GLIST(150),
     $PAPER(300),SHIP4(150),SHIP5(150),SHIP7(150),SHIP10(150),
     $SHIP11(150),SHIP12(150),SHORE(175),SILAND(175)
      EQUIVALENCE (SACKI(1),CX(1)),(SACKI(8776),SACKR(1)),
     $(SACKI(1),BLIST(1)),(SACKI(501),GLIST(1)),(SACKI(1001),DN(1)),
     $(SACKI(1501),DM(1)),(SACKI(2001),ENR(1)),(SACKI(2501),BOOK(1)),
     $(SACKI(3101),PAPER(1)),(CX(1),A(1))
      COMMON/5/SACKI,CX,A,BLIST,GLIST,DN,DM,ENR,BOOK,PAPER,SACKR
      COMMON/BLK1/FACT(300)
7777  CONTINUE
      REWIND 1 $ REWIND 2 $ REWIND 3 $ REWIND 4 $ REWIND 7 $ REWIND 8
      REWIND 10
      RHO = 1000.
      CEE = 1500.
      FACT(1)=1.D
      DO 2945 J=2,171
2945  FACT(J)=(J-1)*FACT(J-1)
      FACT(171)=FACT(171)*1.D-300
      DO 2950 J=172,296
2950  FACT(J)=(J-1)* FACT(J-1)
      FBCT(1) = 1.D-250
      DO 2949 J = 1,275
2949  FBCT(J+1) = J*FBCT(J)
      BACT(1) = 1.
      DO 2939 I = 2,558,2
2939  BACT(I+1) = BACT(I-1)*(I-1.)*4./I
      L1 = 0
      IDL = 1
      READ 90,IDEM,IDEMPI,IDEMPR
   90 FORMAT (3I5)
      READ 91,ZS1,HS1
   91 FORMAT (2D10.5)
```

```
      XS1 = ZS1 + 1.D
      READ 91,ZS2,HS2
      XS2 = ZS2 + 1.D
      READ 96,ALPHA
   96 FORMAT (D10.5)
      READ 93,IVEL1,VEL1,THETAL1,THETAU1
   93 FORMAT (I1,C(F9.3,F10.3),2D10.5)
      READ 93,IVEL2,VEL2,THETAL2,THETAU2
      IDEMPI = IDEMPI/2*2
      IDEMPR = IDEMPR/2*2
      ICXI = 25
      ICXII = ICXI+ICXI-1
      IF (IDEMPI.LT.26) IDEMPI = 26
      NDR = IDEMPR + IDEMPR
      NDI = IDEMPI + IDEMPI
      NFFF = 0
      IDEMPII = NDI - 1
      IDEMPRR = NDR - 1
      IMP = MAX0(IDEMPII,IDEMPRR)
      IMPR = (IMP+1)/2
      ICM = MAX0(ICXI,IDEM)
      NL = IDEM
      INDEX = -2
      IDEM2 = 2*IDEM
      ISBT = 0
      PI = 3.14159265358979323846D
      PI2 = 1.5707963268
      IHDF = 0
      ISM = 0
      R41(1) = CMPLX(0.,1.)
      DO 727 I = 2,IDEM
  727 R41(I) = (0.,1.)*R41(I-1)
      PRINT 5555
 5555 FORMAT (1H145X*ACOUSTIC RADIATION FROM TWO PROLATE SPHEROIDS*)
      PRINT 5556,XS1,HS1,XS2,HS2
 5556 FORMAT (//16X*SHAPE PARAMETER XT   ACOUSTIC SIZE PARAMETER H*//5X
     $10HSPHEROID 1 5XF10.6,14XF10.6/ 5X10HSPHEROID 2 5XF10.6,14XF10.6)
      PRINT 5557,ALPHA
 5557 FORMAT (//5X*SEPARATION PARAMETER*//6X7HALPHA =F11.6)
      PRINT 5558,THETAL1,THETAU1,THETAL2,THETAU2
 5558 FORMAT (//16X*LOWER AND UPPER LIMITS (IN DEGREES) OF VIBRATING BAN
     $DS*//30X5HLOWER12X5HUPPER//5X10HSPHEROID 1 13XF8.3,9XF8.3/5X
     $10HSPHEROID 2 13XF8.3,9XF8.3)
      V1M = CABS(VEL1)
      V2M = CABS(VEL2)
      V1A = CANG(VEL1)*180./PI
      V2A = CANG(VEL2)*180./PI
      PRINT 5559,VEL1,V1M,V1A,VEL2,V2M,V2A
 5559 FORMAT (//16X*NORMAL VELOCITY (IN M/SEC) OF VIBRATING BANDS*
     $//23X4HREAL6X9HIMAGINARY5X9HMAGNITUDE2X*PHASE ANGLE (DEGREES)*
     $//5X10HSPHEROID 1 5XC(F8.3,F12.3),2F14.3
     $/ 5X10HSPHEROID 2 5XC(F8.3,F12.3),2F14.3)
      PRINT 5560,IDEM,IDEMPI,IDEMPR
 5560 FORMAT (//16X*SUMMATION PARAMETERS*//7X6HIDEM =I3,3X8HIDEMPI =I4,
     $3X8HIDEMPR =I3)
      ETAL1 = DCOS(PI*THETAL1/180.D)
```

UNCLASSIFIED

```
      ETAU1 = DCOS(PI*THETAU1/180.D)
      ETAL2 = DCOS(PI*THETAL2/180.D)
      ETAU2 = DCOS(PI*THETAU2/180.D)
      DTK = HS1*ALPHA
      CALL SBFSF(DTK,BSF)
      YMF(1) = -DCOS(DTK)/DTK
      YMF(2) = YMF(1)/DTK - DSIN(DTK)/DTK
      KSR1 = ICXII + IDEMPII - 1
      DO 899 K = 1,KSR1
      KK = K
      YMF(K+2) = (K+K+1.D)*YMF(K+1)/DTK-YMF(K)
      IF (DABS(YMF(K+2)).GT.1.D300)GO TO 4501
  899 CONTINUE
      GO TO 4502
 4501 IDEMPI = (1+KK-ICXII)/4*2
      IF (IDEMPI.GE.ICXI) GO TO 4504
      IDEMPI = (IDEMPI + ICXI)/4*2
      ICXI = IDEMPI
 4504 NDI = IDEMPI + IDEMPI
      IDEMPII = NDI - 1
      IMP = MAX0(IDEMPII,IDEMPRR)
      IMPR = (IMP+1)/2
      PRINT 4503, IDEMPI
 4503 FORMAT (/10X8HIDEMPI =I4,1X24HDUE TO LARGE NEUMANN FNS)
      IF (ICXI.LT.25) PRINT 4505,ICXI
 4505 FORMAT (10X6HICXI = I4/)
 4502 M = 0
      NBL = NL
      EM = M
      LCK = 0
      JDNX = -1
  810 INDEX = INDEX + 1
      IF (INDEX) 814,815,816
  816 READ 719, NFF
  719 FORMAT (I1)
      IF (EOF.60) 723,724
  724 NFFF = NFF
      GO TO (729,720,723,7777) NFF + 1
  729 JDNX = -1*JDNX
      IF (NFF+JDNX.EQ.0) GO TO 1001
      IF (JDNX.GT.0) 825,826
  825 READ 817,XCT,YCT,ZCT
  817 FORMAT (3D10.5)
      CRT = XCT*XCT + YCT*YCT + ZCT*ZCT + 1.D
      X = DSQRT(.5D*(CRT+DSQRT(CRT*CRT-4.D*ZCT*ZCT)))
      E = ZCT/X
      Z = X - 1.D
      IF (XCT.EQ.0..AND.YCT.EQ.0.) GO TO 812
      PHI = ATAN2(YCT,XCT)*180./PI
      GO TO 813
  812 PHI = 0.
  813 PRINT 910,XCT,X,YCT,E,ZCT,PHI
  910 FORMAT (/////57X*INPUT NEARFIELD POINT*//37X*CARTESIAN COORDINATES
     $*19X*SPHEROIDAL COORDINATES*//36X10HX/(D1/2) =E13.5,18X4HXI =E13.5
     $/36X10HY/(D1/2) =E13.5,17X5HETA =E13.5/36X10HZ/(D1/2) =E13.5,17X
     $5HPHI =F13.3,8H DEGREES)
```

```
      AAA = E
      IF (E.LT.0.D) E = -E
      CALL POLY(90,E,P,FBCT)
      H = HS1
      HSQ = H*H
      GO TO 827
826   CRZ = X*AAA - ALPHA
      CRZZ = CRZ*CRZ
      CRT = (CRZZ + Z*(Z+2.D)*(1.D0-E*E))*(HS1/HS2)**2 + 1.D
      X = DSQRT(.5D*(CRT+DSQRT(CRT*CRT-4.D*CRZZ*(HS1/HS2)**2)))
      E = CRZ/X*HS1/HS2
      Z = X-1.D
      AAA = E
      IF (E.LT.0.D) E = -E
      CALL POLY(90,E,P,FBCT)
      H = HS2
      HSQ = H*H
827   LRK = 0
      GO TO 1000
720   READ 721, E
721   FORMAT (D20.10)
      IF (THDF.EQ.1) JDNX = -1*JDNX
      AAB = E
      E = DCOS(PI*E/180.D)
      PRINT 932,AAB
932   FORMAT (/////10X17HFAR FIELD ANGLE =F7.2,1X7HDEGREES)
      AAA = E
      ART = AAA*DTK
      DO 730 I = 1,IDEM
730   R42(I) = R41(I)*CMPLX(COSF(ART),SINF(ART))
      IF (E.LT.0.D) E = -E
      CALL POLY(90,E,P,FBCT)
      GO TO 1001
814   Z = ZS1
      H = HS1
      X = XS1
      LRK = 0
      HSQ = H*H
      IH = H
      N = XMINOF(50+IH/2,70)
      EXEX1 = Z*(Z+2.)
      SPDINT(1) = .5*(ETAU1*DSQRT(X*X-ETAU1*ETAU1)+
     1X*X*DATAN(ETAU1/DSQRT(X*X-ETAU1*ETAU1))-ETAL1*DSQRT(X*X-ETAL1
     2*ETAL1)-X*X*DATAN(ETAL1/DSQRT(X*X-ETAL1*ETAL1)))
      GO TO 1000
815   Z = ZS2
      H = HS2
      X = XS2
      LRK = 0
      HSQ = H*H
      IH = H
      N = XMINOF(50+IH/2,70)
      EXEX2 = Z*(Z+2.)
      SPDINT(2) = .5*(ETAU2*DSQRT(X*X-ETAU2*ETAU2)+
     1X*X*DATAN(ETAU2/DSQRT(X*X-ETAU2*ETAU2))-ETAL2*DSQRT(X*X-ETAL2
     2*ETAL2)-X*X*DATAN(ETAL2/DSQRT(X*X-ETAL2*ETAL2)))
```

UNCLASSIFIED

```
      IF (DABS(HS1-HS2).GT.1.D-7) IHDF = 1
      IF (DABS(XS2-XS1).GT.1.D-7) ISM = 1
      IF (IHDF.EQ.1) LCK = 0
      IF (ISM.EQ.INDEX.AND.IHDF.EQ.0) GO TO 851
1000  IF ((X.GT.1.3D.AND.H .GE.10.D).OR.(X.GT.2.D.AND.H .GE.1.D).OR.(X.G
     1T.5.D.AND.H .GE..1D)) GO TO 1001
      CALL CALCPQ(0,140,Z,SHIP10,SHIP4)
      CALL CALCPQ(1,140,Z,SHIP11,SHIP5)
1001  DO 1 IL = 1,NL
      L = IL - 1
      IF (LCK.EQ.1) GO TO 4
      LCK = 1
      DO 31 J = 1,N
      DO 31 I = 1,N
   31 A(I,J) = 0.
      DO 32 I = 1,N
      W = I-1.
   32 A(I,I) = (I-1.)*I+HSQ*(2.*(I-1.)*I-1.)/((2.*I-3.)*(2.*I+1.))
      NM2 = N-2
      DO 33 I = 1,NM2
      A(I,I+2) = (HSQ/(2.*I+1.))*SQRTF((I*I*(I+1.)*(I+1.))/((2.*I+3.)*
     $(2.*I-1.)))
   33 A(I+2,I) = A(I,I+2)
      CALL EIGEN (A,EIG,N,NBL)
      IF (INDEX.EQ.0) GO TO 36
      DO 34 I = 1,NBL
   34 EIGL1(I) = EIG(I)
      GO TO 4
   36 DO 37 I = 1,NBL
   37 EIGL2(I) = EIG(I)
    4 CL = EIGL1(L+1)
      IW6 = L/2
      IX = L - 2*IW6
      ISC = 2 + IX
      LIM1 = 278
      IF (INDEX)774,775,780
  775 IF (IHDF.EQ.1)57,780
   57 CL = EIGL2(L+1)
  774 J=1
      DO 13 I=ISC,LIM1,2
      EYE =I
      EA = I + I
      BLIST(J) = (EYE*(EYE-1.D)*EYE*(EYE-1.D)*HSQ*HSQ)/((EA-1.D)
     1*(EA-1.D)*(EA-3.D)*(EA+1.D))
   13 J=J+1
      J=1
      ID21 = ISC-1
      LIM11=LIM1+1
      DO 14 I=ID21,LIM11,2
      EA = I - 1.D
      GLIST(J) = EA*(EA+1.D)+0.5D*HSQ*((1.D)+1.D/((EA+EA-1.D)*(EA+EA+3.D
     $)))
   14 J=J+1
      IFC=0
      IBLIM = 139 - IX
      IGLIM=IBLIM+1
```

```
      IRIO=IW6+1
      IW1 = IW6+2
   17 ENR(1)=CL-GLIST(1)
      DO 18 I=1,IW6
   18 ENR(I+1)=-BLIST(I)/ENR(I)-GLIST(I+1)+CL
      ENR(IBLIM)=-BLIST(IBLIM)/(GLIST(IGLIM)-CL)
      IW15=IBLIM-1
      IP=IW1  +IW15
      DO 19 I=IW1  ,IW15
      IPI=IP-I
   19 ENR(IPI)=-BLIST(IPI)/(GLIST(IPI+1)-CL+ENR(IPI+1))
      ENRC=-BLIST(IRIO)/(GLIST(IRIO+1)-CL+ENR(IRIO+1))
      DE = ENRC*ENRC/BLIST(IRIO)
      CORB = DE
      DO 20 I = IW1,IBLIM
      DE = ENR(I)*ENR(I)/BLIST(I)*DE
      CORB = CORB + DE
      IF (DABS(DE/CORB).LT.1.D-27) GO TO 23
   20 CONTINUE
   23 CORA = 1.D
      DE = 1.D
      DO 26   I = 1,IW6
      DE = BLIST(IRIO-I)/(ENR(IRIO-I)*ENR(IRIO-I))*DE
      CORA = CORA + DE
      IF (DABS(DE/CORA).LT.1.D-27) GO TO 27
   26 CONTINUE
   27 DL = (ENRC-ENR(IRIO))/(CORA+CORB)
      CL=DL+CL
      IF (DABS(DL/CL).LT.1.0D-24) GO TO 22
      IFC=IFC+1
      IF (IFC.LT.20) GO TO 17
   22 IF (INDEX.EQ.0) GO TO 28
      EIGL1(L+1) = CL
      GO TO 29
   28 EIGL2(L+1) = CL
   29 DO 118 I = 1,IBLIM
      ARR = IX + 2.D*I
      EA = ARR + ARR
  118 DND0(I) = (EA-1.D)*(EA+1.D)*ENR(I)/((EA-ARR)*(EA-ARR-1.D)*HSQ)
      DN(1) = DND0(1)
      DM(1) = DND0(1)*1.D150
      DO 119 J = 2,IBLIM
      DN(J) = DND0(J)*DN(J-1)
  119 DM(J) = DND0(J)*DM(J-1)
      IW21 = 2*IBLIM
      W28 = 0.D
      IWW = 0
  165 DO 166 J = ISC,IW21,2
      IWW = IWW + 1
  166 W28 = W28 + DN(IWW)
      DLIST(1) = 1.D/(W28+1.D)
      DLAST(1) = DLIST(1)*1.D150
      DO 121 J = 1, IBLIM
      DLAST(J+1) = DM(J)*DLIST(1)
  121 DLIST(J+1) = DN(J)*DLIST(1)
      IF (INDEX.EQ.0) GO TO 42
```

UNCLASSIFIED

```
      BUFFER OUT (1,1) (DLIST(1),DLIST(145))
      BUFFER OUT (3,1) (DLAST(1),DLAST(145))
      IRWD = 1
  792 IF (UNIT,3) 792,787
   42 BUFFER OUT (2,1) (DLIST(1),DLIST(145))
      BUFFER OUT (4,1) (DLAST(1),DLAST(145))
      IRWD = 2
  790 IF (UNIT,4)790,787
  783 PRINT 980
  980 FORMAT (10X*EOF SENSED*)
      STOP
  784 PRINT 981
  981 FORMAT (10X*PARITY ERROR*)
      STOP
  780 IF (IHDF.EQ.0) GO TO 771
      IF (JDNX.EQ.1) GO TO 771
      CL = FIGL2(L+1)
      IRWD = 2
      BUFFER IN (2,1) (DLIST(1),DLIST(145))
  781 IF (UNIT,2) 781,782,783,784
  782 BUFFER IN (4,1) (DLAST(1),DLAST(145))
  785 IF (UNIT,4) 785,787,783,784
  771 IRWD = 1
      BUFFER IN (1,1) (DLIST(1),DLIST(145))
  788 IF (UNIT,1) 788,779,783,784
  779 BUFFER IN (3,1) (DLAST(1),DLAST(145))
  789 IF (UNIT,3) 789,787,783,784
  787 IF (INDEX) 296,301,298
  301 XNML2(L+1) = DLIST(1)**2/(IX+.5)
      DO  43 J = 1,IBLIM
   43 XNML2(L+1) = XNML2(L+1)+DLIST(J+1)**2/(2.*J+IX+.5)
      DO 44 I = 1,ICM
   44 DDU2(L+1,I) = DLIST(I)
      GO TO 299
  296 XNML1(L+1) = DLIST(1)**2/(IX+.5)
      DO 122 J = 1,IBLIM
  122 XNML1(L+1) = XNML1(L+1)+DLIST(J+1)**2/(2.*J+IX+.5)
      DO 120 I = 1,IMPR
  120 DDL1(L+1,I) = DLIST(I)
      GO TO 299
  298 IF (INDEX*NFF.GT.0) GO TO 818
  299 IF ((X.GT.1.3D.AND.H.GE.10.D).OR.(X.GT.2.D.AND.H.GE.1.D).OR.(X.GT.
     15.D.AND.H.GE..1D)) GO TO 479
  300 DO 333 I = 1,100
      BAY(I) = 0.0D
  333 DR(I) = 0.0D
  305 DNEG (1) = DLIST (1)
  307 IS1 = 4
      DRFREE = 0.D
      IK = IX
      DO 319 K = IS1, 140, 2
      J = 140 - IK
      IY = 1 + (J-1)/2
      Q = -J
      UN = (Q+2.D)*(Q+1.D)/((Q+Q+5.D)*(Q+Q+3.D))
      VN = (2.D*Q*(Q+1.D)-1.D)/((2.D*Q+3.D)*(2.D*Q-1.D))
```

```
     $+(Q*(Q+1.D)-CL)/HSQ
      WN = Q*(Q-1.D)/((Q+Q-1.D)*(Q+Q-3.D))
      DRRATIO (IY) = -UN/(WN*DRFREE + VN)
      DRFREE = DRRATIO (IY)
  319 IK = IK + 2
      Q = IX - 2
      VN = (2.D*Q*(Q+1.D)-1.D)/((2.D*Q+3.D)*(2.D*Q-1.D))
     $+(Q*(Q+1.D)-CL)/HSQ
      WN = Q*(Q-1.D)/((Q+Q-1.D)*(Q+Q-3.D))
      IA = IY - 1
      DRRATIO(IA) = 1.D/((1-IX-IX)*(-1-IX-IX)*(WN*DRRATIO(IY)+VN))
      IF (IX.EQ.1) DRRATIO(IA) = -DRRATIO(IA)
      DNEGF = DNEG (1)
      IAA = 0
      DO 320 I = IA, 70
      IAA = IAA + 1
      DR (IAA) = DRRATIO (I) * DNEGF
      DNEGF = DR (IAA)
      IF (DABS(DNEGF).LE.1.D-290) 323,320
  320 CONTINUE
  323 SAIL 3 = Z * (Z + 2.D)
      SAIL4 = DSQRT (SAIL3)
      DO 407 I = 1, 141
  407 SHIP7 (I) = SHIP5 (I)/SAIL4
      DO 1414 I = 1, 141
 1414 SHIP12(I) = SAIL4*SHIP11(I)/SAIL3
      DO 1416 I = 1, 141
 1416 SILAND(I) = SHIP4(I)
      DO 418 I = 1, 70
  418 BAY(I) = DLIST(I)
      IS16 = IX - 1
      SAIL18 = 0.D
      IB = 1
      IFJ = 20
  471 DO 1422 I = IB,IFJ
      II = I
      IS16 = IS16 + 2
 1422 SAIL 18 = BAY(I)*SILAND(IS16) + SAIL18
      I = II
      IF (DABS(BAY(I)*SILAND(IS16)/SAIL18).LT.1.D-26) GO TO 470
      IB = I + 1
      IFJ = IFJ + 5
      GO TO 471
  470 SAIL19 = 0.D
      IS20 = 0
      IM1 = 2 - IX
      IFJ = 40
  474 DO 1426 I = IM1, IFJ, 2
      II = I
      IS20 = IS20 + 1
 1426 SAIL19 = SHIP10(I)*DR(IS20) + SAIL19
      I = II
      IF (DABS(SHIP10(I)*DR(IS20)/SAIL19).LT.1.D-26) GO TO 473
      IFJ = IFJ + 10
      IM1 = II + 2
      GO TO 474
```

UNCLASSIFIED

```
 473 SAIL22 = SAIL18 + SAIL 19
     DO 1429 I = 1, 141
1429 SHORE (I) = SHIP7 (I)
 440 IS16 = IX - 1
     SAIL181 = 0.D
     IB = 1
     IFJ = 20
 475 DO 1435 I = IB,IFJ
     II = I
     IS16 = IS16 + 2
1435 SAIL181 = BAY(I)*SHORE(IS16) + SAIL181
     I = II
     IF (DABS(BAY(I)*SHORE(IS16)/SAIL181).LT.1.D-26) GO TO 476
     IB = I + 1
     IFJ = IFJ + 5
     GO TO 475
 476 IS21 = 0
     SAIL191 = 0.D
     IM1 = 2 - IX
     IFJ = 40
 478 DO 1439 I = IM1, IFJ, 2
     II = I
     IS21 = IS21 +1
1439 SAIL191 = DR(IS21)*SHIP12(I) + SAIL191
     I = II
     IF (DABS(DR(IS21)*SHIP12(I)/SAIL191).LT.1.D-26) GO TO 477
     IFJ = IFJ + 10
     IM1 = II + 2
     GO TO 478
 477 SAIL23 = SAIL181 + SAIL191
     CFAC = -FACT(L+1+IX)*DNEG(1)*H**(1+IX)/FACT(L+1)
     IF (IX.EQ.0) 70,71
  70 DNUM = DLIST(1)
     DO 717 J = 1,48
 717 DNUM = FACT(J+J+1)/(4.D**J*FACT(J+1)*FACT(J+1))*DLIST(J+1)-DNUM
     CFAC = CFAC/DNUM
     IF (2*(IW6/2).NE.IW6) CFAC = -CFAC
     GO TO 72
  71 DNUM = DLIST(1)
     DO 718 J = 1,48
 718 DNUM = FACT(J+J+3)/(2.D**(J+J+1)*FACT(J+1)*FACT(J+2))*DLIST(J+1)
    $-DNUM
     CFAC = CFAC/(DNUM*3.D*(L+1.D))
     IF (2*(IW6/2).NE.IW6) CFAC = -CFAC
  72 R2 = SAIL22/CFAC
     R2D = SAIL23/CFAC
     IF ((X.LE.1.05D.OR.X.GT.1.3D).AND.H.GT..1D) GO TO 599
     RR2 = R2
     RR2D = R2D
 479 PL3 = (Z + 2.D) * Z
     X1SQ = X*X
     XTH = X*H
     NYMN = 2*IBLIM
     IF (LRK.EQ.1) GO TO 63
     DO 550 I = 1,300
 550 ARRAY(I) = 0.0D
```

```
      ARRAY(1) = -DCOS(XTH)/XTH*1.D-150
      ARRAY(2) = ARRAY(1)/XTH-DSIN(XTH)/XTH*1.D-150
      DO 62 K = 1,NYMN
      ARRAY(K+2) = (K+K+1.D)*ARRAY(K+1)/XTH-ARRAY(K)
      IF (DABS(ARRAY(K+2)).GT.1.D300) GO TO 63
   62 CONTINUE
   63 IA = 1 + IX
      IC = 41 + IX
      SUBSUM = 0.D
  503 DO 505 K = IA, IC, 2
      KK = K
      IBOX3 = IABS(K-L-1)/2
      BOOK(K) = DLAST((K+1)/2)*ARRAY(K)
      IF (2*(IBOX3/2).NE.IBOX3) BOOK(K) = -BOOK(K)
      SUBSUM = BOOK (K) + SUBSUM
  505 CONTINUE
      K = KK
      IF (DABS(BOOK(K)/SUBSUM).GE.1.D-26) 506,507
  506 IA = IC + 2
      IC = IC + 10
      GO TO 503
  507 R2 = SUBSUM
      PL17 = PL3/X1SQ
  513 IBB = 1 + IX
      ID = 41 + IX
      SSUM1 = 0.D
  517 DO 522 K = IBB, ID, 2
      KK = K
      IBOX3 = IABS(K-L-1)/2
      PL8 = 1.D
      IF (2*(IBOX3/2).NE.IBOX3)  PL8 = -PL8
      IF ((K-1).EQ.0) 518,519
  518 PAPER(K) = -DLAST((K+1)/2)*PL8*ARRAY(2)
      GO TO 521
  519 PAPER(K) = ((K-1.D)*ARRAY(K-1)-K*ARRAY(K+1))/(2.D*K-1.D)*DLAST((K+
     $1)/2)*PL8
  521 SSUM1 = PAPER (K) + SSUM1
  522 CONTINUE
      K = KK
      IF (DABS(PAPER(K)/SSUM1).GE.1.D-26) 523,524
  523 IBB = ID + 2
      ID = ID + 10
      GO TO 517
  524 R2D = H*SSUM1
      GO TO 597
  599 PL3 = (7 + 2.D) * 7
      X1SQ = X*X
      XTH = X*H
  597 DO 598 I = 1,300
      BOOK (I) = 0.D
  598 PAPER (I) = 0.D
      IF (LRK.EQ.1) GO TO 460
      CALL SBFSF(XTH, BRRAY)
  460 IA = 1 + IX
      IC = 41 + IX
      SUBSUM = 0.D
```

UNCLASSIFIED

```
603 DO 605 K = IA, IC, 2
    KK = K
    IBOX3 = IABS(K-L-1)/2
    BOOK(K) = DLIST((K+1)/2)*BRRAY(K)
    IF (2*(IBOX3/2).NE.IBOX3)  BOOK(K) = -BOOK(K)
    SUBSUM = BOOK (K) + SUBSUM
605 CONTINUE
    K = KK
    IF (DABS(BOOK(K)/SUBSUM).GE.1.D-26) 606,607
606 IA = IC + 2
    IC = IC + 10
    GO TO 603
607 R1 = SUBSUM
    PL17 = PL3/X1SQ
613 IBB = 1 + IX
    ID = 41 + IX
    SSUM1 = 0.D
617 DO 622 K = IBB, ID, 2
    KK = K
    IBOX3 = IABS(K-L-1)/2
    PL8 = 1.D
    IF (2*(IBOX3/2).NE.IBOX3)  PL8 = -PL8
    IF ((K-1).EQ.0) 618,619
618 PAPER(K) = -DLIST((K+1)/2)*PL8*BRRAY(2)
    GO TO 621
619 PAPER(K) = ((K-1.D)*BRRAY(K-1)-K*BRRAY(K+1))/(2.D*K-1.D)*DLIST((K+
   $1)/2)*PL8
621 SSUM1 = PAPER (K) + SSUM1
622 CONTINUE
    K = KK
    IF (DABS(PAPER(K)/SSUM1).GE.1.D-26) 623,624
623 IBB = ID + 2
    ID = ID + 10
    GO TO 617
624 R1D = H*SSUM1
    TWRON = 1.D/(H*Z*(Z+2.D))
    CWRON = R1*R2D - R1D*R2
    IAC = -DLOG10(DABS((TWRON-CWRON)/TWRON) + 1.D-26)
    IF (IAC.LT.0) IAC = 0
    IF ((X.LE.1.05D.OR.X.GT.1.3D).AND.H.GT..1D) GO TO 55
    CWRON = R1*RR2D - R1D*RR2
    IBC = -DLOG10(DABS((TWRON-CWRON)/TWRON) + 1.D-26)
    IF (IAC.GE.IBC) GO TO 55
    R2 = RR2
    R2D = RR2D
    IAC = IBC
 55 IF (IAC.LT.5) PRINT 933, L,R1,R1D,R2,R2D,CL,IAC
933 FORMAT (1X*RADIAL FUNCTION ACCURACY WARNING*5XI3,4D15.5,D20.10,I4)
    IF (INDEX.GT.0) 818,819
819 IF (ISM+IHDF.GT.0) GO TO 841
    R1F1(L+1) = R1
    R1F4(L+1) = CMPLX(R1,-R2)
    R1D1(L+1) = R1D
    R1D4(L+1) = CMPLX(R1D,-R2D)
    GO TO 842
841 R2D1(L+1) = R1D
```

```
      R2D4(L+1) = CMPLX(R1D,-R2D)
      R2F1(L+1) = R1
      R2F4(L+1) = CMPLX(R1,-R2)
      GO TO 842
  818 INX = 1
      IF (AAA.LT.0.D) INX = -1
      MIN = L + 10
      J = 1 + IX
      K = 1
      S = 0.D
      DO 80 I = J,MIN,2
      II = I + 2
      S = S + DLIST(K)*P(I)
   80 K = K + 1
   84 TERM = DLIST(K)*P(II)
      S = S + TERM
      IF (S.EQ.0.D) GO TO 81
      IF (DABS(TERM/S).LT.1.D-14) 81, 82
   82 K = K + 1
      II = II + 2
      GO TO 84
   81 IF (INX.EQ.-1.AND.2*(L/2).NE.L) S = -S
      IF (NFF.EQ.0) GO TO 744
      IF (THDF.EQ.0) GO TO 722
      IF (JDNX.EQ.1) 741,742
  744 IF (JDNX.LT.0) GO TO 820
      R11(L+1) = R1
      R41(L+1) = CMPLX(R1,-R2)
  741 S1(L+1) = S
      GO TO 842
  722 S1(L+1) = S2(L+1) = S
      GO TO 842
  820 R12(L+1) = R1
      R42(L+1) = CMPLX(R1,-R2)
  742 S2(L+1) = S
  842 LRK = 1
    1 LCK = 1
      IF (IRWD.EQ.2) GO TO 51
      REWIND 1
      REWIND 3
      GO TO 52
   51 REWIND 2
      REWIND 4
   52 IF (INDEX)810,748,747
  747 IF (JDNX.GT.0) 729,830
  851 DO 852 I = 1,IDEM
      XNML2(I) = XNML1(I)
      R2F1(I) = R1F1(I)
      R2F4(I) = R1F4(I)
      R2D1(I) = R1D1(I)
  852 R2D4(I) = R1D4(I)
      DO 581 I = 1,IDEM
      DO 581 J = 1,ICM
  581 DDU2(I,J) = DDL1(I,J)
  748 T1 = TIMELEFT(K)
      NZERO = 0
```

UNCLASSIFIED

```
      DO 650 MR = NZERO,ICXII
      DO 650 MS = MR,IMP
      MTB = MS - MR
      MTE = MR + MS
      MM = MTB+1
      NN = MTE+1
      MN = 1-MTB
      TSACKR = TSACKI = 0.
      INT = (-1)**((MS-MR+MTB-2)/2)
      DO 651 MT = MTB,MTE,2
      INT = -INT
      TTERM = (INT*(MT+MT+1.)*BACT(MM+MT)/((NN+MT)*BACT(NN+MT))*
     $BACT(NN-MT)*BACT(MN+MT))
      TSACKR = TSACKR + TTERM*SNGL(BSF(MT+1))
  651 TSACKI = TSACKI + TTERM*SNGL(YMF(MT+1))
      IF (MS.GT.NDR-1.OR.MR.GT.NDR-1) GO TO 650
      SACKR(MS+1+MR*NDR-((MR+1)*MR)/2) = TSACKR
  650 SACKI(MS+1+MR*NDI-((MR+1)*MR)/2) = -TSACKI
      T2 = TIMELEFT(K)
      T = T1 - T2
      PRINT 2000, T
 2000 FORMAT (//5X*TIME TO CALCULATE SACK IS*F8.3,8H SECONDS/)
      T2 = TIMELEFT(K)
      IND = MIN0(IDEMPR,ICXI)
      DO 652 LL = 1,IDEM
      LLL = LL
      LP = LL-1-2*((LL-1)/2)
      LL2 = (LL-LP)/2
      LS = 2*(LL/4)+8
      LS = MIN0(LS,ICXI)
      DO 653 JJ = 1,IDEM
      JJJ = JJ
      JP = JJ-1-2*((JJ-1)/2)
      JJ2 = (JJ-JP)/2
      JSN = JJ2 - 2*(JJ2/2)
      JS = 2*(JJ/4)+10
      JS = MIN0(JS,IDEMPI)
      MN = MIN0(LP+1,JP+1)
      MX = MAX0(LP+1,JP+1)
      SL = SACKI(MX+(MN-1)*NDI-(MN*(MN-1))/2)
      IF (MX.EQ.LP+1) GO TO 406
      IF (2*((MX-MN)/2).NE.MX-MN) SL = -SL
  406 RLIN = DDL1(JJ,1)*SL
      IB = 2
      IE = JS
  405 DO 654 K = IB,IE
      KK = K+K+JP-1
      RLI = RLIN
      KL = KK+LP*NDI-((LP+1)*LP)/2
  654 RLIN = DDL1(JJ,K)*(-1)**(LP+1-KK)*SACKI(KL) - RLI
      IF (ABSF((RLI+RLIN)/RLIN).LT.1.E-8) 401,402
  402 IF (IE.LT.IDEMPI) 403,404
  404 PRINT 499,LL,JJ
  499 FORMAT (//20X*NO CONVERGENCE FOR INITIAL RLI*3X4HLL =I3,3X4HJJ =
     $I3)
      STOP
```

```
403 IB = IE + 1
    IE = MIN0(IB+5,IDEMPI)
    GO TO 405
401 IF (JSN.EQ.0) RLIN = -RLIN
    CXLIN = DDU2(LL,1)*RLIN
    IBB = 2
    IEE = LS
    ISNCXL = 1
425 DO 420 MS = IBB,IEE
    MM = MS+MS+LP-1
    ML = MM+JP*NDI-((JP+1)*JP)/2
    RLIN = DDL1(JJ,1)*SACKI(ML)
    IB = 2
    IE = JS
415 DO 657 K = IB,IE
    KK = K+K+JP-1
    RLI = RLIN
    MN = MIN0(MM,KK)
    MX = MAX0(MM,KK)
    SL = SACKI(MX+(MN-1)*NDI-(MN*(MN-1))/2)
    IF (MX.EQ.MM) GO TO 657
    IF (2*((MX-MN)/2).NE.MX-MN) SL = -SL
657 RLIN = DDL1(JJ,K)*SL - RLI
    IF (ABSF((RLI+RLIN)/RLIN).LT.1.E-6) 411,412
412 IF (IE.LT.IDEMPI)413,414
414 IF (ABSF((RLI+RLIN)/RLIN).LT.1.E-2)411,424
413 IB = IE + 1
    IE = MIN0(IB+5,IDEMPI)
    GO TO 415
411 IF (JSN.EQ.0) RLIN = -RLIN
    CXLI = CXLIN
    ISNCXL = -ISNCXL
420 CXLIN = DDU2(LL,MS)*RLIN - CXLI
    IF (ABSF((CXLI+CXLIN)/CXLIN).LT.1.E-6)421,422
422 IF (IEE.LT.ICXI  )423,424
423 IBB = IEE + 1
    IEE = MIN0(IBB+3,ICXI  )
    GO TO 425
424 IF (ABSF((CXLI+CXLIN)/CXLIN).LT.1.E-2)421,426
421 IF (LL2.NE.2*(LL2/2))CXLIN = -CXLIN
    CXLIN = ISNCXL*CXLIN
653 STORI(JJ) = CXLIN
    GO TO 652
426 IF (LLL.GT.1) GO TO 430
    IDEM = JJJ - 1
    IDEM2 = 2*IDEM
    PRINT 495,LLL,JJJ,IDEM
495 FORMAT (45X18HCXI IS 0 FOR LLL =I3,2X5HJJJ = I3,2X
   $10HIDEM NOW =I3/)
    GO TO 652
430 DO 428 I = JJJ,IDEM
428 STORI(I) = 0.
    PRINT 497,LLL,JJJ,IDEM
497 FORMAT (50X*CXI IS 0 FOR LLL = *I3,2X5HJJJ =I3,3H TO I3)
652 WRITE(7) (STORI(JKL),JKL = 1  ,IDEM)
    IDEM2 = 2*IDEM
```

UNCLASSIFIED

```
      T1 = TIMELEFT(K)
      T = T2 - T1
      PRINT 2001,IDEM,T
 2001 FORMAT (/5X*TIME TO CALCULATE CXLIN FOR IDEM =*I3,3H ISF8.3,8H SEC
     $ONDS/)
      T2 = TIMELEFT(K)
      DO 462 LL = 1,IDEM
      LLL = LL
      LP = LL-1-2*((LL-1)/2)
      LL2 = (LL-LP)/2
      LS = 2*(LL/4)+8
      LS = MIN0(LS,IDEMPR)
      DO 463 JJ = 1,IDEM
      JJJ = JJ
      JP = JJ-1-2*((JJ-1)/2)
      JJ2 = (JJ-JP)/2
      JSN = JJ2 - 2*(JJ2/2)
      JS = 2*(JJ/4)+8
      JS = MIN0(JS,IDEMPR)
      MN = MIN0(LP+1,JP+1)
      MX = MAX0(LP+1,JP+1)
      SK = SACKR(MX+(MN-1)*NDR-(MN*(MN-1))/2)
      IF (MX.EQ.LP+1) GO TO 436
      IF (2*((MX-MN)/2).NE.MX-MN) SK = -SK
  436 RLRN = DDL1(JJ,1)*SK
      IB = 2
      IE = JS
  435 DO 464 K = IB,IE
      KK = K+K+JP-1
      RLR = RLRN
      KL = KK+LP*NDR-((LP+1)*LP)/2
  464 RLRN = DDL1(JJ,K)*(-1)**(LP+1-KK)*SACKR(KL) - RLR
      IF (ABSF((RLR+RLRN)/RLRN).LT.1.E-8) 431,432
  432 IF (IE.LT.IDEMPR) 433,434
  434 PRINT 498,LL,JJ
  498 FORMAT (//20X*NO CONVERGENCE FOR INITIAL RLR*3X4HLL =I3,3X4HJJ =
     $I3)
      STOP
  433 IB = IE + 1
      IE = MIN0(IB+5,IDEMPR)
      GO TO 435
  431 IF (JSN.EQ.0) RLRN = -RLRN
      CXLRN = DDU2(LL,1)*RLRN
      IBB = 2
      IEE = LS
      ISNCXL = 1
  455 DO 450 MS = IBB,IEE
      MM = MS+MS+LP-1
      ML = MM+JP*NDR-((JP+1)*JP)/2
      RLRN = DDL1(JJ,1)*SACKR(ML)
      IB = 2
      IE = JS
  445 DO 467 K = IB,IE
      KK = K+K+JP-1
      RLR = RLRN
      MN = MIN0(MM,KK)
```

```
      MX = MAX0(MM,KK)
      SK = SACKR(MX+(MN-1)*NDR-(MN*(MN-1))/2)
      IF (MX.EQ.MM) GO TO 467
      IF (2*((MX-MN)/2).NE.MX-MN) SK = -SK
  467 RLRN = DDL1(JJ,K)*SK - RLR
      IF (ABSF((RLR+RLRN)/RLRN).LT.1.E-6) 441,442
  442 IF (IE.LT.IDEMPR)443,444
  444 IF (ABSF((RLR+RLRN)/RLRN).LT.1.E-2)441,454
  443 IB = IE + 1
      IE = MIN0(IB+5,IDEMPR)
      GO TO 445
  441 IF (JSN.EQ.0) RLRN = -RLRN
      CXLR = CXLRN
      ISNCXL = -ISNCXL
  450 CXLRN = DDU2(LL,MS)*RLRN - CXLR
      IF (ABSF((CXLR+CXLRN)/CXLRN).LT.1.E-6) 451,452
  452 IF (IEE.LT.  IND )453,454
  453 IBB = IEE + 1
      IEE = MIN0(IBB+3,IND)
      GO TO 455
  454 IF (ABSF((CXLR+CXLRN)/CXLRN).LT.1.E-2)451,456
  451 IF (LL2.NE.2*(LL2/2))CXLRN = -CXLRN
      CXLRN = ISNCXL*CXLRN
  463 STORR(JJ) = CXLRN
      GO TO 462
  456 DO 458 I = JJJ,IDEM
  458 STORR(I) = 0.
      PRINT 496,LLL,JJJ,IDEM
  496 FORMAT (50X*CXR IS 0 FOR LLL = *I3,2X5HJJJ =I3,3H TO I3)
  462 WRITE(8) (STORR(JKL),JKL = 1   ,IDEM)
      T1 = TIMELEFT(K)
      T = T2 - T1
      PRINT 2006,IDEM,T
 2006 FORMAT (/5X*TIME TO CALCULATE CXLRN FOR IDEM =*I3,3H ISF8.3,8H SEC
     $ONDS/)
      CALL INTEGR(IDEM2,XS1,ETAL1,ETAU1,XIL,1,FBCT)
      DO 869 J = 1,IDEM
      SIRL1(J) = 0.
      KX = J-1 - 2*((J-1)/2)
      DO 869 I = 1,IDEM
  869 SIRL1(J) = SIRL1(J) + DDL1(J,I)*XIL(2*I-1+KX)
      CALL INTEGR(IDEM2,XS2,ETAL2,ETAU2,XIL,1,FBCT)
      DO 871 J = 1,IDEM
      SIRL2(J) = 0.
      KX = J-1 - 2*((J-1)/2)
      DO 871 I = 1,IDEM
  871 SIRL2(J) = SIRL2(J) + DDU2(J,I)*XIL(2*I-1+KX)
      REWIND 7
      REWIND 8
      DO 750 I = 1,IDEM
      DCS(I) = R1D1(I)*R1F4(I)/R1D4(I)/XNML1(I)
  750 DCS(IDEM+I) = R2D1(I)/R2D4(I)/XNML2(I)
      DO 649 I = 1,IDEM
      DO 649 J = 1,IDEM
  649 CX(I,J) = CX(I+IDEM,J+IDEM) = (0.,0.)
      DO 351 I = 1,IDEM
```

UNCLASSIFIED

```
      READ (7) (STORI(II),II=1,IDEM)
      READ (8) (STORR(II),II=1,IDEM)
      DO 351 J = 1,IDEM
  351 CX(I+IDEM,J) = CX(J,IDEM+I) = CMPLX(STORR(J),STORI(J))
      DO 353 I = 1,IDEM
      DO 353 J = 1,IDEM
      CX(I,J+IDEM) = CX(I,J+IDEM)*2./XNML1(I)
  353 CX(I+IDEM,J) = CX(I+IDEM,J)*2./XNML2(I)
      DO 751 I = 1,IDEM
      TRS1(I) = TRS2(I) = (0.,0.)
      DO 751 J = 1,IDEM
      TRS1(I) = TRS1(I) + CX(J,IDEM+I)*R1F1(J)*SIRL1(J)
  751 TRS2(I) = TRS2(I) + CX(IDEM+J,I)*R2F1(J)*SIRL2(J)
      DO 350 I = 1,IDEM
      DO 350 J = 1,IDEM
      CX(I,J+IDEM) = CX(I,J+IDEM)*R1D1(I)/R1D4(I)
  350 CX(I+IDEM,J) = CX(I+IDEM,J)*R2D1(I)/R2D4(I)
      DO 806 I = 1,IDEM2
  806 CX(I,I) = CMPLX(1.,0.)
      T2 = TIMELEFT(K)
      CALL CINV(CX,IDEM2,60)
      T1 = TIMELEFT(K)
      T = T2 - T1
      PRINT 915,IDEM2,T
  915 FORMAT (/5X*TIME TO INVERT MATRIX OF SIZE *I2,4H IS F8.3,8H SECOND
     $S)
      WRITE (10) ((CX(I,J),I=1,IDEM2),J=1,IDEM2)
      PRINT 5561
 5561 FORMAT (//38X*ACOUSTIC IMPEDANCE CONTRIBUTIONS (FOR CONVERGENCE CH
     $ECK)*/)
      SIMP1 = SIMP2 = SIMP3 = SIMP4 = SIMP5 = SIMP6 = (0.,0.)
      SIMP7 = SIMP8 = SIMP9 = (0.,0.)
      DO 752 I = 1,IDEM
      SIMP1 = SIMP1 + R1F1(I)*R1F4(I)*SIRL1(I)*SIRL1(I)/XNML1(I)
      SIMP4 = SIMP4 + R2F1(I)*SIRL2(I)*TRS1(I)/XNML2(I)
      SIMP7 = SIMP7 + R2F1(I)*R2F4(I)*SIRL2(I)*SIRL2(I)/XNML2(I)
  752 PRINT 940, I,SIMP1,SIMP4,SIMP7
  940 FORMAT (2XI3,5X7HSIMP1 =C(E15.7,E15.7),5X7HSIMP4 =C(E15.7,E15.7),
     $5X7HSIMP7 =C(E15.7,E15.7))
      PRINT 5056
 5056 FORMAT (//)
      DO 753 I = 1,IDEM
      SUM = SVM = (0.,0.)
      DO 754 J = 1,IDEM
      SUM = SUM + CX(I,J)*DCS(J)*SIRL1(J) + CX(I,IDEM+J)*DCS(IDEM+J)
     $*TRS1(J)
      SVM = SVM + CX(I,J)*DCS(J)/R1F4(J)*TRS2(J) + CX(I,IDEM+J)
     $*DCS(IDEM+J)*R2F4(J)*SIRL2(J)
  754 CONTINUE
      SIMP2 = SIMP2 + R1F4(I)*SIRL1(I)*SUM
      SIMP5 = SIMP5 + TRS2(I)*SUM
      SIMP8 = SIMP8 + TRS2(I)*SVM
  753 PRINT 941, I,SIMP2,SIMP5,SIMP8
  941 FORMAT (2XI3,5X7HSIMP2 =C(E15.7,E15.7),5X7HSIMP5 =C(E15.7,E15.7),
     $5X7HSIMP8 =C(E15.7,E15.7))
      PRINT 5056
```

```
      DO 755 I = 1,IDEM
      SUM = SVM = (0.,0.)
      DO 756 J = 1,IDEM
      SUM = SUM + CX(I+IDEM,J)*DCS(J)*SIRL1(J) + CX(IDEM+I,IDEM+J)
     $*DCS(IDEM+J)*TRS1(J)
      SVM = SVM + CX(I+IDEM,J)*DCS(J)/R1F4(J)*TRS2(J) + CX(IDEM+I,IDEM+J
     $)*DCS(IDEM+J)*R2F4(J)*SIRL2(J)
  756 CONTINUE
      SIMP3 = SIMP3 + TRS1(I)*SUM
      SIMP6 = SIMP6 + R2F4(I)*SIRL2(I)*SUM
      SIMP9 = SIMP9 + R2F4(I)*SIRL2(I)*SVM
  755 PRINT 942, I,SIMP3,SIMP6,SIMP9
  942 FORMAT (2XI3,5X7HSIMP3 =C(E15.7,E15.7),5X7HSIMP6 =C(E15.7,E15.7),
     $5X7HSIMP9 =C(E15.7,E15.7))
      SIMP3 = (SIMP1 - SIMP2 - SIMP3)*SQRTF(EXEX1)*HS1**2/SPDINT(1)
      SIMP6 = (SIMP4 - SIMP5 - SIMP6)*SQRTF(EXEX2)*HS2**2/SPDINT(1)
      SIMP9 = (SIMP7-SIMP8-SIMP9)*SQRTF(EXEX2)*HS2**2/SPDINT(2)
      SIMP7 = SIMP6*SPDINT(1)/SPDINT(2)*(HS1/HS2)**2*SQRTF(EXEX1/EXEX2)
      P1S = CABS(SIMP3)
      P1A = CANG(SIMP3)*180./PI
      P1SS = CABS(SIMP7)
      P1AA = CANG(SIMP7)*180./PI
      P2S = CABS(SIMP9)
      P2A = CANG(SIMP9)*180./PI
      P2SS = CABS(SIMP6)
      P2AA = CANG(SIMP6)*180./PI
      PRINT 5562
 5562 FORMAT (/47X*NORMALIZED ACOUSTIC RADIATION IMPEDANCES*/
     $/42X10HRESISTANCE5X9HREACTANCE8X9HMAGNITUDE6X*PHASE ANGLE (DEGREES
     $)*/)
      PRINT 943,SIMP3,P1S,P1A,SIMP7,P1SS,P1AA,SIMP9,P2S,P2A,SIMP6,
     $P2SS,P2AA
  943 FORMAT (34X5HZ11 =C(E13.5,E15.5),5XE12.5,6XF9.3/34X
     $5HZ21 =C(E13.5,E15.5),5XE12.5,6XF9.3/34X
     $5HZ22 =C(E13.5,E15.5),5XE12.5,6XF9.3/34X
     $5HZ12 =C(E13.5,E15.5),5XE12.5,6XF9.3)
      REWIND 10
      GO TO 810
  830 READ  (10) ((CX(I,J),I=1,IDEM2),J=1,IDEM2)
      DO 860 I = 1,IDEM
      DACS(I) = S1(I)*R41(I)*DCS(I)/R1F4(I)
  860 DACS(I+IDEM) = S2(I)*R42(I)*DCS(IDEM+I)
      DO 749 I = 1,IDEM2
      CTS(I) = (0.,0.)
      DO 749 J = 1,IDEM2
  749 CTS(I) = CTS(I) + CX(I,J)*DACS(J)
      SUM1 = SUM2 = STOR1 = STOR2 = (0.,0.)
      REWIND 10
      IF (IVEL1.EQ.0) GO TO 731
      ISP = 1
      PRINT 925,ISP
  925 FORMAT (//30X*ACOUSTIC PRESSURE CONTRIBUTIONS DUE TO SPHEROID*I2,
     $* (FOR CONVERGENCE CHECK)*/)
      SM1 = SM2 = SM3 = (0.,0.)
      DO 872 I = 1,IDEM
      SM1 = SM1 + S1(I)*R1F1(I)*R41(I)*SIRL1(I)/XNML1(I)
```

UNCLASSIFIED

```
872 PRINT 920, I,SM1
920 FORMAT (13XI3,5X5HSM1 =C(E15.7,F15.7))
    PRINT 5056
    DO 873 I = 1,IDEM
    SM2 = SM2 + CTS(I)*R1F4(I)*SIRL1(I)
873 PRINT 921, I,SM2
921 FORMAT (13XI3,5X5HSM2 =C(E15.7,F15.7))
    PRINT 5056
    DO 874 I = 1,IDEM
    SM3 = SM3 + CTS(IDEM+I)*TRS1(I)
874 PRINT 923,I,SM3
923 FORMAT (13XI3,5X5HSM3 =C(E15.7,F15.7))
    PRMLTR = HS1**2*CEE*RHO*SQRTF(EXEX1)*VEL1
    SUM1 = (SM1 - SM2 - SM3)*PRMLTR
    PRS = CABS(SUM1)
    PRA = CANG(SUM1)*180./PI
    IF (NFF.EQ.1) GO TO 760
    ISP = 1
    PRINT 924,ISP,SUM1,PRS,PRA
924 FORMAT (//15X*SPHEROID*I2,* CONTRIBUTION TO THE ACOUSTIC PRESSURE
   $(NTS/SQUARE METER) AT THE INPUT NEARFIELD POINT*//33X9HREAL PART3X
   $14HIMAGINARY PART11X9HMAGNITUDE5X*PHASE ANGLE (DEGREES)*//30X
   $C(E13.5,E14.5),10XE13.5,5XF9.3)
    GO TO 731
760 ISBT = ISBT + 1
    STOR1 = SUM1*CMPLX(COSF(.5*ART),-SINF(.5*ART))
    ANGLE(ISBT) = AAB
    ISBT = ISBT - 1
731 IF (IVEL2.EQ.0) GO TO 732
    ISP = 2
    PRINT 925,ISP
    SM1 = SM2 = SM3 = (0.,0.)
    DO 875 I = 1,IDEM
    SM1 = SM1 + S2(I)*R2F1(I)*R42(I)*SIRL2(I)/XNML2(I)
875 PRINT 920,I,SM1
    PRINT 5056
    DO 876 I = 1,IDEM
    SM2 = SM2 + CTS(I)*TRS2(I)
876 PRINT 921,I,SM2
    PRINT 5056
    DO 877 I = 1,IDEM
    SM3 = SM3 + CTS(I+IDEM)*R2F4(I)*SIRL2(I)
877 PRINT 923,I,SM3
    PRMLTR = HS2**2*CEE*RHO*SQRTF(EXEX2)*VEL2
    SUM2 = (SM1 - SM2 - SM3)*PRMLTR
    PRS = CABS(SUM2)
    PRA = CANG(SUM2)*180./PI
    IF (NFF.EQ.1) GO TO 761
    ISP = 2
    PRINT 924,ISP,SUM2,PRS,PRA
732 IF (NFF.EQ.1) GO TO 733
    SUMM = SUM1 + SUM2
    PRS = CABS(SUMM)
    PRA = CANG(SUMM)*180./PI
    PRINT 927,XCT,YCT,ZCT,SUMM,PRS,PRA
927 FORMAT (//12X*TOTAL ACOUSTIC PRESSURE (NTS/SQUARE METER) AT THE IN
```

```
      $PUT NEARFIELD POINT (*E12.5,1H,E12.5,1H,E12.5,1H)//33X9HREAL PART
      $3X14HIMAGINARY PART11X9HMAGNITUDE5X*PHASE ANGLE (DEGREES)*//30X
      $C(E13.5,E14.5),10XE13.5,5XF9.3)
       GO TO 810
  761  ISBT = ISBT + 1
       STOR2 = SUM2*CMPLX(COSF(.5*ART),-SINF(.5*ART))
       ANGLE(ISBT) = AAB
       ISBT = ISBT - 1
  733  ISBT = ISBT + 1
       FF(ISBT) = STOR1 + STOR2
       GO TO 810
  723  IF (NFFF.EQ.0) GO TO 811
       DO 725 I = 2,ISBT
       FF(I) = FF(I)/FF(1)
       FFA(I) = CANG(FF(I))*180./PI
  725  FFM(I) = CABS(FF(I))
       FF(1) = FFM(1) = 1.
        PRINT 929
  929  FORMAT (//53X*NORMALIZED FARFIELD PRESSURE*//16X*ANGLE (DEGREES)*
      $9X9HREAL PART3X14HIMAGINARY PART11X9HMAGNITUDE5X
      $*PHASE ANGLE (DEGREES)*/)
       PRINT 930, (ANGLE(I),FF(I),FFM(I),FFA(I),I=1,ISBT)
  930  FORMAT (20XF7.2,10XC(E13.5,E14.5),10XE13.5,5XF9.3)
       IF (NFF.EQ.2) GO TO 810
  811  CONTINUE
       STOP
       END
```

UNCLASSIFIED

```
      SUBROUTINE CALCPQ(M,LN,XX,P,Q)
      DIMENSION P(150),Q(150),AP( 2,150)
      COMMON/BLK1/FACT(300)
      TYPE DOUBLE P,Y,Y1,S,X,D2M,DJ,DM,DN,FACT,TERM,SUM,CWRON,DK,Z,ZZ,
     1XA,HAL,AL,BE,XY,COEF,GA,REALJ,REALJH,Q,FIRST, HM,  XZ,TWRON,B,
     2XX,RC1,RC2,RC3,RC4,PC1,PC2,PC3,PC4,AP,FACTR1,FACTR2,FACTR3
      INITIAL=0
      NN=0
   16 X=XX+1.D
      KAPPA=0
      IF(X.GT. 1.001D) KAPPA=M
      DO 65  MN=KAPPA,M
      DM = MN
      D2M=2.D**DM
      Z=XX**(DM/2.0D)
      ZZ=(X+1.D)**(DM/2.0D)
      Y=Z*ZZ
      DO 65,  N=NN,LN
      IF (MN.GT.N) 30,35
   30 AP(MN+1,N+1) = 0.0D
      GO TO 65
   35 DN = N
      NLESSM=N-MN
      TERM = FACT(N+MN+1)/FACT(N-MN+1)/(FACT(MN+1)*D2M)
      SUM = TERM
      DO 40 J = 1,NLESSM
      DJ = J
      TERM = TERM*(DN+DM+DJ)*XX*(DN-DM-DJ+1)/(DJ*(DM+DJ)*2.D)
      SUM = SUM + TERM
   40 IF (DABS(TERM/SUM).LT.1.D-30) GO TO 45
   45 AP(MN+1,N+1)=Y*SUM
   65 P(N+1) = AP(MN+1,N+1)
      IF (X.GT. 1.001D) 300,50
   50 FIRST= 0.5D * DLOG( (XX+2.0D)/XX)
      MONE=M+1
      DO 145 N=NN,LN
      DN = N
      Q(N+1) = FIRST*P(N+1)
      IF(N.EQ.0) GO TO70
      SUM=0.D
      KRAP=(N-1)/2
      DO 60 J=INITIAL,KRAP
       RJ=J
      DJ=DBLE(RJ)
      S=((2.D*DN-4.D*DJ-1.D)/(( (1.D) +2.D*DJ)*(DN-DJ)))*P(N-2*J)
   60 SUM=SUM+S
      Q(N+1) = Q(N+1)-SUM
   70 DO 140 J=2,MONE
      RJ=J-1
      REALJ=DBLE(RJ)
      REALJH= REALJ/2.0D
      Y= (XX**REALJ-((X+1.D)**REALJ))/(((XX*(X+1.D))**REALJH) *2.0D)
      S=FACT(M+1)/(REALJ  *FACT(M-J+2))
      TERM=Y*S*AP(M-J+2,N+1)
      IF (2*(J/2).NE.J) TERM = -TERM
  140 Q(N+1) = Q(N+1) + TERM
```

```
  145 CONTINUE
      GO TO 350
  300 EPS=1.0E-30
      XY=XX*(XX+2.0D)
      DM=M
      XZ=1.0D/(X+DSQRT(XY))
      Z=XZ*XZ
      HM=DM/2.0D
      XA=XY**HM
      DO 340 N=NN,LN
      DN = N
      AL=DN+DM+1.0D
      BE=DM+0.5D
      GA=DN+1.5D
      HAL=AL/2.0D
      SUM=1.0D
      TERM=1.0D
      K=0
  330 K=K+1
      DK = K-1
      IF(K.GT.2500)GO TO 335
  331 TERM=TERM*(AL+DK)*(BE+DK)*Z/(K*(GA+DK))
      SUM=SUM+TERM
      EPC=TERM/SUM
      IF(EPC-EPS)335,335,330
  335 IF(M+N .GE.131) 410,420
  410 FACTR1=2.D**(AL+DN)*(FACT(N+M+1)*1.D-300)
      GO TO 425
  420 FACTR1=2.0D**(AL+DN)*FACT(N+M+1)
  425 FACTR2= (Z**HAL)*XA
      FACTR3 = FACTR1*FACTR2
      COEF=FACTR3*(FACT(N+1)/FACT(2*N+2))
      IF(N.LT.85) GO TO 1338
      IF ( M+N.LT.131) COEF=COEF*(1.D-300)
 1338 JM=M/2
      IF(2*JM.EQ.M) 340,338
  338 COEF=-1.D*COEF
  340 Q(N+1) = COEF*SUM
  350 END
```

* * * * * * * *

```
      SUBROUTINE CINV(B,NS,ND)
      TYPE COMPLEX B,D
      DIMENSION B( ND,1),L(100),M(100)
      DO 1 J = 1,NS
      DO 1 I = 1,NS
    1 B(I+NS*(J-1)) = B(I+ND*(J-1))
      CALL MINC(B,NS,ND,D,L,M)
      DO 2 JJ = 1,NS
      J = NS+1-JJ
      DO 2 II = 1,NS
      I = NS+1-II
    2 B(I+ND*(J-1)) = B(I+NS*(J-1))
      RETURN
      END
```

UNCLASSIFIED

```
      SUBROUTINE MINC(A,N,ND,D,L,M)
      TYPE COMPLEX A,D,BIGA,BIGAI,HOLD,B
      EQUIVALENCE(B,C)
      DIMENSION C(2),C1(2)
      EQUIVALENCE(BIGA,C1)
      DIMENSION A(ND,1),L(1),M(1)
      D= (1.0,0.0)
      NK = -N
      DO 80 K=1,N
      NK = NK+N
      L(K) = K
      M(K) = K
      KK = NK+K
      BIGA = A(KK)
      BIGAI = (1.0,0.0)/BIGA
      J=L(K)
      IF(J-K) 35,35,25
   25 KI = K-N
      DO 30 I=1,N
      KI = KI+N
      HOLD = -A(KI)
      JI = KI -K+J
      A(KI)=A(JI)
   30 A(JI) = HOLD
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP= N*(I-1)
      DO 40 J=1,N
      JK = NK+J
      JI = JP+J
      HOLD = -A(JK)
      A(JK) = A(JI)
   40 A(JI) = HOLD
   45 IF(C1(1).EQ.0..AND.C1(2).EQ.0.)46,48
   46 D= 0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK = NK + I
      A(IK) = -BIGAI*A(IK)
   55 CONTINUE
      DO 65 I=1,N
      IK = NK+I
      HOLD = A(IK)
      IJ = I-N
      DO 65 J=1,N
      IJ = IJ+N
      IF(I.EQ.K.OR.J.EQ.K) GO TO 65
   62 KJ = IJ -I +K
      A(IJ) = HOLD*A(KJ) + A(IJ)
   65 CONTINUE
      KJ = K-N
      DO 75 J=1,N
      KJ = KJ + N
      IF(J-K) 70,75,70
   70 A(KJ) = BIGAI*A(KJ)
   75 CONTINUE
      D = D*BIGA
      A(KK) = BIGAI
   80 CONTINUE
  150 RETURN
      END
```

```
      SUBROUTINE EIGEN(A,VALU,N,NBL)
      DIMENSION  A(70,70),VALU(70),DIAG(70),Q(70),VALL(70)
      ANORM = A(1,1)*A(1,1) + A(2,2)*A(2,2)
      DO 6 I = 3,N
    6 ANORM = ANORM + A(I,I)*A(I,I)+ 2*A(I-2,I)*A(I-2,I)
      ANORM = SQRTF(ANORM)
      NN=N-2
      DO 160 I=1,NN
      II=I+2
      DO 160 J=II,N
      T1=A(I,I+1)
      T2=A(I,J)
      IF (T2.EQ.0.) GO TO 160
      T=1./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+1,K)+SIN*A(J,K)
      A(J,K)=COS*A(J,K)-SIN*A(I+1,K)
  125 A(I+1,K)=T2
  160 CONTINUE
      DO15I=1,N
      DIAG(I)=A(I,I)
      Q(I)=A(I,I-1)*A(I,I-1)
      VALL(I)=0.
   15 VALU(I) = ANORM
      I=1
   18 TAU=(VALL(I)+VALU(I))/2.
      MATCH=0
      T0 = 0.
      T1 = 1.
      DO20 J=1,N
      T2=(DIAG(J)-TAU)*T1-Q(J)*T0
      IF (ABS(T2).LT.1.E130) GO TO 26
      T2 = T2*1.E-200
      T1 = T1*1.E-200
   26 IF((T1.NE.0.).AND.((T2*T1).LE.0.))  MATCH=MATCH+1
      T0=T1
   20 T1=T2
      DO 25 K = I, MATCH
   25 VALU(K) = TAU
      NATCH = MATCH + 1
      DO 30 K = NATCH, NBL
   30 IF (VALL(K).LT.TAU) VALL(K) = TAU
   40 IF ((VALU(I)-VALL(I))/VALU(I).GT.1.E-7)  GO TO 18
      I=I+1
      IF(I.LE.NBL)  GO TO 40
      END
```

UNCLASSIFIED

```
 SUBROUTINE INTEGR(LN,TSI,ETA1,ETA2,A,INC,FBCT)
 TYPE DOUBLE A,ARG,BARG,BETA,CEF,ETA1,ETA2,FACT,FARG,FSCT,RTSUM1,
1RTSUM2,SUM,TSI,TSISQ,W,X,Y,YS,Z,FBCT
 DIMENSION A(1),ARG(96),BARG(96),CEF(96),FARG(96),SUM(202),W(96),
1X(96),Y(96),YS(96),Z(96),FBCT(300)
 COMMON/BLK1/FACT(300)
 DATA(X= -0.016276744849602969579D000,
*-0.048812985136049731112D,
*         -0.081297495464425558994D000,
*         -0.113695850110665920911D000,
*         -0.145973714654896941989D000,
*-0.178096882367618602759D000,-0.210031310460567203603D000,
*-0.241743156163840012328D000,-0.273198812591049141487D000,
*-0.304364944354496353024D000,-0.335208522892625422616D000,
*-0.365696861472313635031D000,-0.395797649828908603285D000,
*-0.425478988407300545365D000,-0.454709422167743008636D000,
*-0.483457973920596359768D000,-0.511694177154667673586D000,
*-0.539388108324357436227D000,-0.566510418561397168404D000,
*-0.593032364777572080684D000,-0.618925840125468570386D000,
*-0.644163403784967106798D000,
*-0.668718310043916153953D000,-0.692564536642171561344D000,
*-0.715676812348967626225D000,-0.738030643744400132851D000,
*-0.759602341176647498703D000,-0.780369043867433217604D000,
*-0.800308744139140817229D000,-0.819400310737931675539D000,
*-0.837623511228187121494D000,-0.854959033434601455463D000,
*-0.871388505909296502874D000,-0.886894517402420416057D000,
*-.901460635315852341319D000,-0.915071423120898074206D000,
*-0.927712456722308690965D000,-0.939370339752755216932D000,
*-0.950032717784437635756D000,-0.959688291448742539300D000,
*-0.968326828463264212174D000,-0.975939174585136466453D000,
*-0.982517263563014677447D000,-0.988054126329623799481D000,
*-0.992543900323762624572D000,-0.995981842987209290650D000,
*-0.998364375863181677724D000,-0.999689503883230766828D000)
 DATA(W=0.032550614492363166242D,0.032516118713868835987D,
*0.032447163714064269364D,0.032343822568575928429D,
*0.032206204794030250669D,0.032034456231992663218D,
*0.031828758894411006535D,0.031589330770727168558D,
*0.031316425596861355813D,0.031010332586313837423D,
*0.030671376123669149014D,0.030299915420827593794D,
*0.029896344136328385984D,0.029461089958167905970D,
*0.028994614150555236543D,0.028497411065085385646D,
*0.027970007616848334440D,0.027412962726029242823D,
*0.026826866725591762198D,0.026212340735672413913D,
*0.025570036005349361499D,0.024900633222483610288D,
*0.024204841792364691282D,0.023483399085926219842D,
*0.022737069658329374001D,0.021966644438744349195D,
*0.021172939892191298988D,.020356797154333324595D,
*0.019519081140145022410D, 0.018660679627411467385D,
*0.017782502316045260838D,0.016885479864245172450D,
*0.015970562902562291381D,0.015038721026994938006D,
*0.014090941772314860916D,0.013128229566961572637D,
*0.012151604671088319635D,0.011162102099838498591D,
*0.010160770535008415758D,0.009148671230783386633D,
*0.008126876925698759217D,0.007096470791153865269D,
*0.006058545504235961683D,0.005014202742927517693D,
*0.003964554338444686674D,0.002910731817934946408D,
```

```
   *0.00185396078894692173 2D,0.00079679206555201242 9D)
    LNM1 = LN - 1
    DO 5I=1,48
    X(48+I)=-X(I)
  5 W(I+48)=W(I)
    TSISQ=TSI*TSI
    RTSUM1=0.5D*(ETA2-ETA1)
    RTSUM2=0.5D*(ETA1+ETA2)
    A(1)=0.D
    A(2)=0.D
615 DO630 I=1,96
    Y(I)=RTSUM1*X(I)+RTSUM2
    YS(I)=Y(I)*Y(I)
    Z(I)=DSQRT((1.D)-YS(I))
    CEF(I)=W(I)*DSQRT(TSISQ-YS(I))*RTSUM1
    A(1)=A(1)+CEF(I)
    A(2)=A(2)+CEF(I)*Y(I)
    BARG(I)=0.D
    FARG(I)=1.D
630 ARG(I)=Y(I)
    SUM(1)=A(1)
    SUM(2)=A(2)
    DO 650 L=2,LN
    SUM(L+1)=0.D
    DO650 I=1,96
    BARG(I)=BARG(I)*Y(I)+FARG(I)*Z(I)
    FARG(I)=ARG(I)
    ARG(I)=ARG(I)*Y(I)-BARG(I)*Z(I)
650 SUM(L+1)=SUM(L+1)+ARG(I)*CEF(I)
632 FORMAT(1X,I4,D35.25)
631 DO 680 N = 2,LNM1,INC
    LIMIT=N/2
    A(N+1)=(FBCT(N+N)/FBCT(N))/(2.D*FACT(N+1))*SUM(N+1)
    DO 660 L=1,LIMIT
    BETA=(FACT(L+L)/(FACT(L)*FACT(L+1)*FACT(N-L+1)))
   1*(FBCT(N+N-L-L)/FBCT(N-L))
    IF((2*L).EQ.N)  BETA=.5D*BETA
660 A(N+1)=A(N+1)+BETA*SUM(N-L-L+1)
679 A(N+1)=0.5D**(N+N-3)*A(N+1)
680 CONTINUE
700 CONTINUE
    END
```

UNCLASSIFIED

```
      SUBROUTINE POLY(LN,ARG,P,FBCT)
      TYPE DOUBLE FACT,FBCT,P,ANGLE,TERM,SUM,ARG,S,Y
      DIMENSION P(125),Y(150),FBCT(300)
      COMMON/BLK1/FACT(300)
      II = 0
      M = 0
      IF (DABS(1.D0-ARG).LT.1.D-10) ARG = 1.D
      ANGLE = DATAN2(DSQRT(1.D0 - ARG*ARG),ARG)
      LNN=LN+1
      LDN = LN + 2
      DO 2 N = 1,LNN
      P(N) = 0.D
    2 Y(N) = DCOS((N-1)*ANGLE)
      MM = 1
      M2 = 2
      IF ((ANGLE.NE.0.).AND.((ANGLE.LT.3.1415926).OR.(ANGLE.GT.3.1415926
     19))) GO TO 6
      ARG = 1.
      DO 5 N=1,LNN
    5 P(N)= 1.D
      IF( ANGLE .EQ.0) GO TO 14
      DO 4 N=2,LNN,2
    4 P(N)=-P(N)
      GO TO 14
    6 IF ((ANGLE.LT.1.570796320).OR.(ANGLE.GT.1.570796329))GO TO 10
      ARG = 0.
    3 P(1) = 1.D
      DO 7 N = 2,LN,2
      P(N+1) = FACT(N)/(FACT(N/2)*FACT(N/2+1)*2.D**(N-1))
    7 IF (2*((N-M)/4).NE.(N-M)/2)  P(N+1)=-P(N+1)
      GO TO 14
   10 P(MM) = 1.D
   11 DO 13 N=MM,LN
      S = 0.D
      IN = (N-M-1)/2
      DO 12 I = II,IN
   12 S = S+FACT(2*M+2*I+1)*FBCT(2*N-2*I+1)*Y(N-M-2*I+1)/(2.D**
     1(M+N)*FBCT(N-I+1)*FACT(MM+I)*FACT(I+1)*FACT(N-M-I+1))
      P(N+1) = S/2.**(N-1)
   13 IF (2*((M+N)/2).EQ.M+N) P(N+1)=P(N+1)+(FACT(N+1)/
     1(FACT((N+M)/2+1)*FACT((N-M)/2+1)))**2/(FACT(2*M+1)*2.D**(2*N-M))
   14 RETURN
      END
```

```
      SUBROUTINE SBESF(XH,RAY)
      DIMENSION RAY(250)
      TYPE DOUBLE CP,FACT,RAY,SUM,TERM,TK,TM,XH,XI,Z2H
      COMMON /BLK1/FACT(300)
      L = 0
    1 IF (XH.GE..4D) GO TO 4
      Z2H=XH*XH/2.D
      DO 3 N = L,140
      TM=FACT(N+1)*(XH +XH)**N/FACT(N+N+2)
      IF(N.GT.84) TM=TM*1.D-300
      IF(TM.EQ.0.D) GO TO 8
      SUM=1.D
      TERM = 1.D
      DO 2 I=1,50
      XI=I*(N+N+I+I+1)
      TERM=-TERM*Z2H/XI
      SUM = SUM + TERM
    2 IF(DABS(TERM/SUM).LT.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
      NM1=N-1
      DO 11 K = 1,138
   11 RAY(K+2)=(K+K+1)*RAY(K+1)/XH-RAY(K)
      RETURN
   20 IF (XH.GT.10.0D) N = 200
      RAY(N+1) = 1.D-200
      RAY(N+2)=0.D
      I = -N
      M = -1
      DO 5 KK=I,M
      K = -KK
      TK=K+K+1
    5 RAY(K)=TK*RAY(K+1)/XH-RAY(K+2)
      CP=DSIN(XH)/(XH*RAY(1))
      IF(DSIN(XH).LT.1.D-2) CP=(DSIN(XH)/XH-DCOS(XH))/(XH*RAY(2))
      DO 6 L=1,140
    6 RAY(L)=CP*RAY(L)
    8 DO 9 J=N,139
    9 RAY(J+1)=0.D
      DO 12 I = 141,250
   12 RAY(I) = 0.D
      END
```

UNCLASSIFIED

Appendix B

LISTING IN FORTRAN OF OSPRDS, A COMPUTER PROGRAM TO CALCULATE THE ACOUSTIC RADIATION FROM TWO COAXIAL OBLATE SPHEROIDS

```
      PROGRAM OSPRDS
      TYPE COMPLEX CTS,CX,DACS,DCS,FF,R41,R42,R1D4,R2D4,R1F4,R2F4,
     $SIMP1,SIMP2,SIMP3,SIMP4,SIMP5,SIMP6,SIMP7,SIMP8,SIMP9,SM1,SM2,SM3,
     $STOR1,STOR2,SUM,SUMM,SUM1,SUM2,SVM,TRS1,TRS2,VEL1,VEL2
      TYPE DOUBLE AAA,ALPHA,BSF,CRT,CRZ,CRZZ,DTK,E,EIGL1,EIGL2,ETA1,
     $ETA2,ETAL1,ETAL2,ETAU1,ETAU2,FBCT,HS1,HS2,P,PI,S,TERM,XCT,XIL,
     $XS1,XS2,YCT,YMF,ZCT,THETAL1,THETAU1,THETAL2,THETAU2
      TYPE DOUBLE AA,AH,AR,ARATIO,ARG,ARRAY,ATERM,BLIST,BOOK,CL,COEFF1,
     1COEFF2,COEFF3,CORA,CORB,CRAD2,CRAD2D,CWRON,DE,DH,DL,DLIST,DN,DNEG,
     2DRATIO,DW,DX,E1,E2,E3,EJN,EM,ENR,ENRC,ESTORE,ESUM,EYE,FACT,FN1,
     3FN2,FNM,FSTRAT,GLIST,H,HHH,OUTPUT,PAPER,PCL,PEN,PL8,PL9,PLA,PLB,
     4PLC,PLD,Q,R1,R1D,R2,R2D,R3,RAD1,RAD1D,RAD2,RAD2D,RATIO,RR,RS,
     5RSTORE,RSUM,SSUM1,SUBSUM,TEMP,TERM1,TERM2,TERM3,TWRON,W,X,X1,XL,XX
      DIMENSION ANGLE(40),BACT(560),BSF(250),CTS(60),CX(60,60),DACS(60),
     $DCS(60),DDL1(30,110),DDU2(30, 30),DLAST(200),DM(200),EIGL1(30),
     $EIGL2(30),FBCT(300),FF(40),FFA(40),FFM(40),P(125),R11(30),
     $R41(30),R12(30),R42(30),R1D1(30),R1D4(30),R2D1(30),R2D4(30),
     $R1F1(30),R1F4(30),R2F1(30),R2F4(30),S1(30),S2(30),
     $SACKI( 8850),SACKR(2780),SIRL1(30),SIRL2(30),SPDINT(2),STORI(30),
     $STORR(30),TRS1(30),TRS2(30),XIL(60),XNML1(30),XNML2(30),YMF(300)
      DIMENSION A(80,80),AIG(80),ARATIO(250),ARRAY(250),BLIST(250),
     1BOOK(250),COEFF1(250),COEFF2(250),COEFF3(250),DLIST(200),DN(200),
     2DRATIO(30),ENR(250),FNM(250),GLIST(250),OUTPUT(130, 1),PAPER(250),
     3PCL(250),PEN(250),Q(200),RATIO(250)
      EQUIVALENCE (SACKI(1),CX(1)),(SACKI(8776),SACKR(1)),
     $(SACKI(1),BLIST(1)),(SACKI(501),GLIST(1)),(SACKI(1001),DN(1)),
     $(SACKI(1501),DM(1)),(SACKI(2001),ENR(1)),(SACKI(2501),BOOK(1)),
     $(SACKI(3001),COEFF1(1)),(SACKI(3501),COEFF2(1)),(SACKI(4001),COEFF
     $3(1)),(SACKI(4501),PAPER(1)),(SACKI(5001),PCL(1)),(SACKI(5501),PEN
     $(1)),(SACKI(6001),RATIO(1)),(SACKI(6501),ARATIO(1)),(CX(1),A(1))
      COMMON/5/SACKI,CX,A,SACK,BLIST,GLIST,DN,DM,ENR,BOOK,COEFF1,
     $COEFF2,COEFF3,PAPER,PCL,PEN,RATIO,ARATIO,SACKR
      COMMON/BLK1/FACT(300)
7777  CONTINUE
      REWIND 1$REWIND 2$REWIND 7$REWIND 8$REWIND 10
      RHO = 1000.
      CEE = 1500.
      FACT(1)=1.D
      DO 2945 J=2,171
2945  FACT(J)=(J-1)*FACT(J-1)
      FACT(171)=FACT(171)*1.D-300
      DO 2950 J=172,296
2950  FACT(J)=(J-1)* FACT(J-1)
      FBCT(1) = 1.D-250
      DO 2949 J = 1,275
2949  FBCT(J+1) = J*FBCT(J)
      BACT(1) = 1.
      DO 2939 I = 2,558,2
2939  BACT(I+1) = BACT(I-1)*(I-1.)*4./I
      L1 = 0
      IDL = 1
      READ 90,IDEM,IDEMPI,IDEMPR
   90 FORMAT (3I5)
      READ 91,XS1,HS1
   91 FORMAT (2D10.5)
      READ 91,XS2,HS2
```

```
      READ 96,ALPHA
   96 FORMAT (D10.5)
      READ 93,IVEL1,VEL1,THETAL1,THETAU1
   93 FORMAT (I1,C(F9.3,F10.3),2D10.5)
      READ 93,IVEL2,VEL2,THETAL2,THETAU2
      IDEMPI = IDEMPI/2*2
      IDEMPR = IDEMPR/2*2
      ICXI = 25
      ICXII = ICXI+ICXI-1
      IF (IDEMPI.LT.26) IDEMPI = 26
      NDR = IDEMPR + IDEMPR
      NDI = IDEMPI + IDEMPI
      NFFF = 0
      IDEMPII = NDI - 1
      IDEMPRR = NDR - 1
      IMP = MAX0(IDEMPII,IDEMPRR)
      IMPR = (IMP+1)/2
      ICM = MAX0(ICXI,IDEM)
      NL = IDEM
      INDEX = -2
      IDEM2 = 2*IDEM
      ISBT = 0
      PI = 3.14159265358979323846D
      PI2 = 1.5707963268
      IHDF = 0
      ISM = 0
      R41(1) = CMPLX(0.,1.)
      DO 727 I = 2,IDEM
  727 R41(I) = (0.,1.)*R41(I-1)
      PRINT 5555
 5555 FORMAT (1H145X*ACOUSTIC RADIATION FROM TWO OBLATE SPHEROIDS*)
      PRINT 5556,XS1,HS1,XS2,HS2
 5556 FORMAT (//16X*SHAPE PARAMETER XI   ACOUSTIC SIZE PARAMETER H*//5X
     $10HSPHEROID 1 5XF10.6,14XF10.6/ 5X10HSPHEROID 2 5XF10.6,14XF10.6)
      PRINT 5557,ALPHA
 5557 FORMAT (//5X*SEPARATION PARAMETER*//6X7HALPHA =F11.6)
      PRINT 5558,THETAL1,THETAU1,THETAL2,THETAU2
 5558 FORMAT (//16X*LOWER AND UPPER LIMITS (IN DEGREES) OF VIBRATING BAN
     $DS*//30X5HLOWER12X5HUPPER//5X10HSPHEROID 1 13XF8.3,9XF8.3/5X
     $10HSPHEROID 2 13XF8.3,9XF8.3)
      V1M = CABS(VEL1)
      V2M = CABS(VEL2)
      V1A = CANG(VEL1)*180./PI
      V2A = CANG(VEL2)*180./PI
      PRINT 5559,VEL1,V1M,V1A,VEL2,V2M,V2A
 5559 FORMAT (//16X*NORMAL VELOCITY (IN M/SEC) OF VIBRATING BANDS*
     $//23X4HREAL6X9HIMAGINARY5X9HMAGNITUDE2X*PHASE ANGLE (DEGREES)*
     $//5X10HSPHEROID 1 5XC(F8.3,F12.3),2F14.3
     $/ 5X10HSPHEROID 2 5XC(F8.3,F12.3),2F14.3)
      PRINT 5560,IDEM,IDEMPI,IDEMPR
 5560 FORMAT (//16X*SUMMATION PARAMETERS*//7X6HIDEM =I3,3X8HIDEMPI =I4,
     $3X8HIDEMPR =I3)
      ETAL1 = DCOS(PI*THETAL1/180.D)
      ETAU1 = DCOS(PI*THETAU1/180.D)
      ETAL2 = DCOS(PI*THETAL2/180.D)
      ETAU2 = DCOS(PI*THETAU2/180.D)
```

UNCLASSIFIED

```
      DTK = HS1*ALPHA
      CALL SBESF(DTK,140,BSF)
      YMF(1) = -DCOS(DTK)/DTK*1.E-295
      YMF(2) = YMF(1)/DTK - DSIN(DTK)/DTK*1.E-295
      KSR1 = ICXII + IDEMPII - 1
      DO 899 K = 1,KSR1
      KK = K
      YMF(K+2) = (K+K+1.D)*YMF(K+1)/DTK-YMF(K)
      IF (DABS(YMF(K+2)).GT.1.D300)GO TO 4501
  899 CONTINUE
      GO TO 4502
 4501 IDEMPI = (1+KK-ICXII)/4*2
      IF (IDEMPI.GE.ICXI) GO TO 4504
      IDEMPI = (IDEMPI + ICXI)/4*2
      ICXI = IDEMPI
 4504 NDI = IDEMPI + IDEMPI
      IDEMPII = NDI - 1
      IMP = MAX0(IDEMPII,IDEMPRR)
      IMPR = (IMP+1)/2
      PRINT 4503, IDEMPI
 4503 FORMAT (/10X8HIDEMPI =I4,1X24HDUE TO LARGE NEUMANN FNS)
      IF (ICXI.LT.25) PRINT 4505,ICXI
 4505 FORMAT (10X6HICXI = I4/)
 4502 M = 0
      EM = M
      LF = NL - 1
      N = N1 = NL
      LCK = 0
      JDNX = -1
  810 INDEX = INDEX + 1
      IF (INDEX) 814,815,816
  816 READ 719, NFF
  719 FORMAT (I1)
      IF (EOF,60) 723,724
  724 NFFF = NFF
      GO TO (729,720,723,7777) NFF + 1
  729 JDNX = -1*JDNX
      IF (NFF+JDNX.EQ.0) GO TO 1000
      IF (JDNX.GT.0) 825,826
  825 READ 817,XCT,YCT,ZCT
  817 FORMAT (3D10.5)
      CRT = XCT*XCT + YCT*YCT + ZCT*ZCT - 1.D
      X = DSQRT(.5D*(CRT+DSQRT(CRT*CRT+4.D*ZCT*ZCT)))
      E = ZCT/X
      XX = X*X + 1.D
      IF (XCT.EQ.0..AND.YCT.EQ.0.) GO TO 812
      PHI = ATAN2(YCT,XCT)*180./PI
      GO TO 813
  812 PHI = 0.
  813 PRINT 910,XCT,X,YCT,E,ZCT,PHI
  910 FORMAT (/////57X*INPUT NEARFIELD POINT*//37X*CARTESIAN COORDINATES
     $*19X*SPHEROIDAL COORDINATES*//36X10HX/(D1/2) =E13.5,18X4HXI =E13.5
     $/36X10HY/(D1/2) =E13.5,17X5HETA =E13.5/36X10HZ/(D1/2) =E13.5,17X
     $5HPHI =F13.3,8H DEGREES)
      AAA = E
      IF (E.LT.0.D) E = -E
```

```
      CALL POLY(90,E,P,FBCT)
      H = HS1
      HSQ = H*H
      AA=-H*H
      HHH = AA*AA
      GO TO 827
  826 CRZ = X*AAA - ALPHA
      CRZZ = CRZ*CRZ
      CRT = (CRZZ + (X*X+1.D)*(1.D0-E*F))*(HS1/HS2)**2 - 1.D
      X = DSQRT(.5D*(CRT+DSQRT(CRT*CRT+4.D*CRZZ*(HS1/HS2)**2)))
      F = CRZ/X*HS1/HS2
      XX = X*X + 1.D
      AAA = E
      IF (E.LT.0.D) E = -E
      CALL POLY(90,E,P,FBCT)
      H = HS2
      HSQ = H*H
      AA=-H*H
      HHH = AA*AA
  827 LRK = 0
      GO TO 1000
  720 READ 721, E
  721 FORMAT (D20.10)
      IF (IHDF.EQ.1) JDNX = -1*JDNX
      AAB = E
      E = DCOS(PI*E/180.D)
      PRINT 932,AAB
  932 FORMAT (/////10X17HFAR FIELD ANGLE =F7.2,1X7HDEGREES)
      AAA = E
      ART = AAA*DTK
      DO 730 I = 1,IDEM
  730 R42(I) = R41(I)*CMPLX(COSF(ART),SINF(ART))
      IF (E.LT.0.D) E = -E
      CALL POLY(90,E,P,FBCT)
      GO TO 1001
  814 X = XS1
      H = HS1
      LRK = 0
      HSQ = H*H
      AA=-H*H
      HHH = AA*AA
      IF(H.GT.20.D) N=N1+(H-20.D)/2.D
      IF(N.LT.50) N=50
      IF(N.GT.80) N=80
      XX = X*X + 1.D
      EXEX1 = XX
      SPDINT(1) = .5*(ETAU1*DABS(ETAU1) - ETAL1*DABS(ETAL1))
      IF (X.EQ.0.D) GO TO 6
      SPDINT(1) = .5*(ETAU1*DSQRT(X*X+ETAU1*ETAU1)+
     1X*X*DLOG(ETAU1+DSQRT(X*X+ETAU1*ETAU1))-ETAL1*DSQRT(X*X+ETAL1*ETAL1
     2)-X*X*DLOG(ETAL1+DSQRT(X*X+ETAL1*ETAL1)))
      GO TO 1000
  815 X = XS2
      H = HS2
      LRK = 0
      HSQ = H*H
```

UNCLASSIFIED

```
      AA=-H*H
      HHH = AA*AA
      IF(H.GT.20.D) N=N1+(H-20.D)/2.D
      IF(N.LT.50) N=50
      IF(N.GT.80) N=80
      XX = X*X + 1.D
      EXEX2 = XX
      SPDINT(2) = .5*(ETAU2*DABS(ETAU2) - ETAL2*DABS(ETAL2))
      IF (X.EQ.0.D) GO TO 1100
      SPDINT(2) = .5*(ETAU2*DSQRT(X*X+ETAU2*ETAU2)+
     1X*X*DLOG(ETAU2+DSQRT(X*X+ETAU2*ETAU2))-ETAL2*DSQRT(X*X+ETAL2*ETAL2
     2)-X*X*DLOG(ETAL2+DSQRT(X*X+ETAL2*ETAL2)))
 1100 IF (DABS(HS1-HS2).GT.1.D-7) IHDF = 1
      IF (DABS(XS2-XS1).GT.1.D-7) ISM = 1
      IF (IHDF.EQ.1) LCK = 0
      IF (ISM.EQ.INDEX.AND.IHDF.EQ.0) GO TO 851
 1000 IF (X.EQ.0.D) GO TO 6
      LQ = 125
      CALL QLFG(0,LQ,X,Q)
      DO 8I=1,LQ
    8 OUTPUT(I+1,1)=Q(I)
      OUTPUT(1,1) = 1.D
    6 ARG = X*H
 1001 DO 1 IL = 1,NL
      L = IL - 1
      IF (LCK.EQ.1) GO TO 4
      LCK = 1
      DO41 J = 1, N
      DO41 I = 1,N
   41 A(I,J) = 0.
      DO42 I = 1,N
      XL = I - 1
   42 A(I,I) =XL*(XL+1.)+AA*(2.*XL*(XL+1.)-1.)/((2.*XL-1.)*(2.*XL+3.))
      NM2 = N-2
      DO43 I = 1,NM2
      XL = I - 1
      A(I,I+2) = (AA/(2.*XL+3.))*SQRTF(((XL+2.)*(XL+1.)*
     1(XL+2.)*(XL+1.))/((2.*XL+5.)*(2.*XL+1.)))
   43 A(I+2,I) = A(I,I+2)
      ANORM1=AA
      ANORM2=LF*(LF+1.)
      CALL EIGEN(A,AIG,N,N1,ANORM1,ANORM2)
      IF (INDEX.EQ.0) GO TO 36
      DO 7I=1,N1
    7 EIGL1(I)=AIG(I)
      GO TO 38
   36 DO 33 I = 1,N1
   33 EIGL2(I) = AIG(I)
   38 PLB = 1.D
    4 CL = EIGL1(L+1)
      IFC=0
      IUCT = L/2
      IX = L - 2*IUCT
      IRIO=IUCT+1
      IR=IRIO+1
      IF (INDEX)774,775,780
```

```
775 IF (IHDF.EQ.1)57,780
 57 CL = EIGL2(L+1)
774 LIM = 240
    ID = 2 + IX
    IB = LIM/2 - IX
    GLIST(1) = AA/3.D
    IF (IX.EQ.1) GLIST(1) = (2.D) + .6D*AA
    IIB=IB-1
    DO 13I=ID,LIM,2
    EYE=I
    BLIST((I-ID+2)/2)=EYE*(EYE-1.D)*EYE*(EYE-1.D)*HHH/
   1((2.D*EYE-3.D)*(2.D*EYE+1.D)*(2.D*EYE-1.D)*(2.D*EYE-1.D))
 13 GLIST((I-ID+4)/2)=EYE*(EYE+1.D)+.5D*AA*((1.D)+1.D/((2.D*EYE-1.D)*
   1(2.D*EYE+3.D)))
 17 ENR(1)=CL-GLIST(1)
    DO 18I=1,IUCT
 18 ENR(I+1)=-BLIST(I)/ENR(I)-GLIST(I+1)+CL
    ENR(IB)=-BLIST(IB)/(GLIST(IB+1)-CL)
    IP=IIB+IR
    DO 19 I=IR,IIB
    IPI=IP-I
 19 ENR(IPI)=-BLIST(IPI)/(GLIST(IPI+1)-CL+ENR(IPI+1))
    ENRC=-BLIST(IRIO)/(GLIST(IR)-CL+ENR(IR))
    DE=ENRC*ENRC/BLIST(IRIO)
    CORB=DE
    DO 20 I=IR,IB
    DE=ENR(I)*ENR(I)/BLIST(I)*DE
    CORB=CORB+DE
 20 IF(DABS(DE/CORB).LT.1.D-27) GO TO 23
 23 CORA = 1.D
    DE=1.D
    DO 26 I=1,IUCT
    DE=BLIST(IRIO-I)/(ENR(IRIO-I)*ENR(IRIO-I))*DE
    CORA=CORA+DE
 26 IF(DABS(DE/CORA).LT.1.D-27) GO TO 27
 27 DL=(ENRC-ENR(IRIO))/(CORA+CORB)
    CL=CL+DL
    IF(DABS(DL/CL).LT.1.D-24) GO TO 22
    IFC=IFC+1
    IF(IFC.LT.50) GO TO 17
 22 IF (INDEX.EQ.0) GO TO 28
    EIGL1(L+1) = CL
    GO TO 31
 28 EIGL2(L+1) = CL
 31 AR=ID
    DN(1) = (2.D*AR-1.D)*(2.D*AR+1.D)*ENR(1)/(AR*(AR-1.D)*AA)
    DM(1) = DN(1)*1.E250
    W = DN(1)
    DO30J=2,IB
    AR=ID+2*(J-1)
    DN(J) = DN(J-1)*(2.D*AR-1.D)*(2.D*AR+1.D)*ENR(J)/(AR*(AR-1.D)*AA)
    DM(J) = DM(J-1)*(2.D*AR-1.D)*(2.D*AR+1.D)*ENR(J)/(AR*(AR-1.D)*AA)
    DW = DN(J)
 30 W=W+DW
    DLIST(1) = 1.D/(W+1.D)
    DLAST(1) = DLIST(1)*1.E250
```

UNCLASSIFIED

```
      DO32 J=1,113
      DLAST(J+1) = DM(J)*DLIST(1)
   32 DLIST(J+1)=DN(J)*DLIST(1)
      IF (INDEX.EQ.0) GO TO 46
      IRWD = 1
      BUFFER OUT (1,1) (DLIST(1),DLIST(72))
  792 IF (UNIT,1) 792,787
   46 BUFFER OUT (2,1) (DLIST(1),DLIST(72))
      IRWD = 2
  790 IF (UNIT,2) 790,787
  783 PRINT 980
  980 FORMAT (10X*EOF SENSED*)
      STOP
  784 PRINT 981
  981 FORMAT (10X*PARITY ERROR*)
      STOP
  780 IF (IHDF.EQ.0) GO TO 771
      IF (JDNX.EQ.1) GO TO 771
      CL = EIGL2(L+1)
      IRWD = 2
      BUFFER IN (2,1) (DLIST(1),DLIST(72))
  781 IF (UNIT,2) 781,787,783,784
  771 IRWD = 1
      BUFFER IN (1,1) (DLIST(1),DLIST(72))
  788 IF (UNIT,1)738,787,783,784
  787 IF (INDEX) 296,301,298
  301 XNML2(L+1) = DLIST(1)**2/(IX+.5)
      DO 45 J = 1,72
   45 XNML2(L+1) = XNML2(L+1)+DLIST(J+1)**2/(2.*J+IX+.5)
      DO 44 I = 1,ICM
   44 DDU2(L+1,I) = DLAST(I)
      GO TO 299
  296 XNML1(L+1) = DLIST(1)**2/(IX+.5)
      DO 122 J = 1,72
  122 XNML1(L+1) = XNML1(L+1)+DLIST(J+1)**2/(2.*J+IX+.5)
      DO 120 I = 1,IMPR
  120 DDL1(L+1,I) = DLAST(I)
      GO TO 299
  298 IF (INDEX*NFF.GT.0) GO TO 818
  299 IF(X.NE.0.D)  GO TO  200
      DRATIO(1)=0.D
      DNEG=DLIST(1)
      TERM = FACT(ID+ID-3)/(2.D**(ID-2)*FACT(ID-1))
      FSTRAT=TERM*DLIST(1)
      TERM2=DABS(FSTRAT)
      DO 34 I=2,71
      TERM = -TERM*(I+I+ID+ID-6)*(I+I+ID+ID-7)/(4.D*(I-1)*(I+ID-3.))
      TERM1=TERM*DLIST(I)
      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
      RR=DCOS(ARG)/H
      GO TO (201,203,205,206),L-(L/4)*4+1
```

```
201 E2 = - RR
    RS = E1
    GO TO 208
203 RS = - RR
    E2 = - F1
    GO TO 208
205 E2 = RR
    RS = - F1
    GO TO 208
206 RS = RR
    E2 = E1
208 E3 = RS
    R3 = - E2
    IF(L.NE.L1) GO TO 211
    LNE=145
    CALL SBESF(ARG,LNE,ARRAY)
211 PLA = 1.D
    IA=1
    IF(2*IUCT.NE.L)IA=IA+1
    IC=IA+142
    IF(X*H.GE.100.D) IC=IC-4
214 SUBSUM = 0.D
215 DO 217  K = IA, IC, 2
```

UNCLASSIFIED

```
242 RAD1=0.D
    RAD1D=PLC*H/3.D
    RAD2=-1.D/(H*RAD1D)
    RAD2D=3.D*PI*FSTRAT*FSTRAT/(2.D*H*H*DNEG)
    GO TO 326
244 RAD1=-PLC
    RAD1D=0.D
    RAD2 =  PI*FSTRAT*FSTRAT/(2.D*H*DNEG)
    RAD2D=1.D/(H*RAD1)
    GO TO 326
246 RAD1=0.D
    RAD1D=-PLC*H/3.D
    RAD2=-1.D/(H*RAD1D)
    RAD2D=-3.D*PI*FSTRAT*FSTRAT/(2.D*H*H*DNEG)
    GO TO 326
252 MA=123
    GO TO 256
254 MA=MA-20
256 JS=MA-2
    S=JS
    RATIO(MA)=0.D
    IMA = MA - 1
    RATIO (IMA) = 1.D
    DO 253    J = 1, IMA
    I=IMA-J+1
    COEFF1(I)=2.D*H*(S+1.D)*(S+1.D)/(2.D*S+3.D)
    COEFF2(I) = S*(S+1.D)-CL-H*H
    COEFF3(I) = 2.D*H*S*S/(2.D*S-1.D)
253 S = S - 1.D
    ARATIO (2) = 1. D
    ARATIO ( 1) = 0. D
269 DO 280 I=1,JS
    K=JS-I+1
    FN1 = COEFF1 (K + 1) * RATIO (K + 2)
    FN2 = COEFF2 (K + 1) * RATIO (K + 1)
270 RATIO (K) = (FN1 - FN2)/COEFF3 (K + 1)
    IF(DABS(RATIO(K)).GT.1.D+300) GO TO 254
    IF(DABS(RATIO(K+1)).GT.DABS(RATIO(K))) GO TO 272
280 CONTINUE
    IND=K
    GO TO 274
272 IND=K+1
274 IF(IND.LT.2) IND=2
281 DO 289 J=1,K
    IJ = J + 1
    FN1 = COEFF2 (J) * ARATIO (IJ)
    FN2 = COEFF3 (J) * ARATIO (IJ - 1)
    ARATIO (IJ + 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)/RATIO(1)
    RSUM=OUTPUT(3,1)*ARATIO(3)
    ESUM=OUTPUT(2,1)
    RSTORE=-OUTPUT(2,1)*(ARATIO(3)+H)
    ESTORE=1.D/XX+OUTPUT(3,1)*ARATIO(3)*H
```

```
      DO 306 K=4,MA
      AR=K
      TERM1=OUTPUT(K,1)*ARATIO(K)
      TERM3=-ARATIO(K)*(AR-2.D)*OUTPUT(K-1,1)/XX
      IF(2*(K/2).EQ.K) GO TO 304
      RSUM=RSUM+TERM1
      ESTORE=ESTORE+TERM1*H
      RSTORE=RSTORE+TERM3
      IF(ABS(TERM1/ESTORE).LT.1.D-27.AND.ABS(TERM3/RSTORE).LT.1.D-27
     1.AND.ABS(TERM1/RSUM).LT.1.D-27) GO TO 309
      GO TO 306
  304 TERM2=TERM1*X*(AR-2.D)/XX
      ESUM=ESUM+TERM1
      RSTORE=RSTORE-TERM1*H
      ESTORE=ESTORE+TERM2-TERM3
  306 CONTINUE
  309 RSUM = (RAD1 + E3 * ESUM)/R3
      RSTORE = (RAD1D + E3 * ESTORE)/R3
  311 CRAD2=R3*ESUM+E3*RSUM
      CRAD2D=R3*ESTORE+E3*RSTORE
      TWRON=1.D/(H*XX)
      CWRON=RAD1*CRAD2D-CRAD2*RAD1D
      NIAC=-DLOG10(DABS((TWRON-CWRON)/TWRON)+1.D-26)
      IF(X.LE.1.D.OR.(X*H).LE.10.D) GO TO 325
      IF(L.NE.L1) GO TO 313
      LNE=141+M
      CALL SPHYN(ARG,LNE,FNM)
  313 JN=1
      IF (2*(L/2).EQ.L)JN = 0
      RAD2=0.D
  315 K=1+JN/2
      TERM = DLIST(K)*FNM(JN+1)
      IF((JN+1).GT.170)  TERM=TERM*1.D+300
      IF (4*((JN-L)/4).NE.(JN-L)) TERM = -TERM
      RAD2=RAD2+TERM
      IF(K.LT.5) GO TO 316
      IF(DABS(TERM /RAD2).LT.(1.D-27)) GO TO 318
  316 JN=JN+2
      GO TO 315
  318 RAD2=RAD2*PLA
      RAD2D=0.D
      JN=JN-2*(JN/2)
  320 K=1+JN/2
      EJN=JN
      PL8=H/(2.D*EJN+1.D)
      TERM = DLIST(K)*PL8*(EJN*FNM(JN)-(EJN+1.D)*FNM(JN+2))
      IF((JN+1).GT.170)  TERM=TERM*1.D+300
      IF(4*((JN-L)/4).NE.(JN-L)) TERM =-TERM
      RAD2D=RAD2D+TERM
      IF(K.LT.5) GO TO 322
      IF(DABS(TERM /RAD2D).LT.(1.D-27)) GO TO 324
  322 JN=JN+2
      GO TO 320
  324 CWRON = RAD1 * RAD2D - RAD2 * RAD1D
      IAC=-DLOG10(DABS((TWRON-CWRON)/TWRON)+1.D-26)
      IF(IAC.GT.NIAC) GO TO 326
```

UNCLASSIFIED

```
325 RAD2=CRAD2
    RAD2D=CRAD2D
    IAC=NIAC
326 IF(IAC.LT.0) IAC=0
    R1 = RAD1
    R1D = RAD1D
    R2 = RAD2
    R2D = RAD2D
 55 IF (IAC.LT.5) PRINT 933, L,R1,R1D,R2,R2D,CL,IAC
933 FORMAT (1X*RADIAL FUNCTION ACCURACY WARNING*5XI3,4D15.5,D20.10,I4)
    IF (INDEX.GT.0) 818,819
819 IF (ISM+IHDF.GT.0) GO TO 841
    R1F1(L+1) = R1
    R1F4(L+1) = CMPLX(R1,-R2)
    R1D1(L+1) = R1D
    R1D4(L+1) = CMPLX(R1D,-R2D)
    GO TO 842
841 R2D1(L+1) = R1D
    R2D4(L+1) = CMPLX(R1D,-R2D)
    R2F1(L+1) = R1
    R2F4(L+1) = CMPLX(R1,-R2)
    GO TO 842
818 INX = 1
    IF (AAA.LT.0.D) INX = -1
    MIN = L + 10
    J = 1 + IX
    K = 1
    S = 0.D
    DO 80 I = J,MIN,2
    II = I + 2
    S = S + DLIST(K)*P(I)
 80 K = K + 1
 84 TERM = DLIST(K)*P(II)
    S = S + TERM
    IF (S.EQ.0.D) GO TO 81
    IF (DABS(TERM/S).LT.1.D-14) 81, 82
 82 K = K + 1
    II = II + 2
    GO TO 84
 81 IF (INX.EQ.-1.AND.2*(L/2).NE.L) S = -S
    IF (NFF.EQ.0) GO TO 744
    IF (IHDF.EQ.0) GO TO 722
    IF (JDNX.EQ.1) 741,742
744 IF (JDNX.LT.0) GO TO 820
    R11(L+1) = R1
    R41(L+1) = CMPLX(R1,-R2)
741 S1(L+1) = S
    GO TO 842
722 S1(L+1) = S2(L+1) = S
    GO TO 842
820 R12(L+1) = R1
    R42(L+1) = CMPLX(R1,-R2)
742 S2(L+1) = S
842 LRK = 1
  1 LCK = 1
    IF (IRWD.EQ.2) GO TO 51
```

```
      REWIND 1
      GO TO 52
   51 REWIND 2
   52 IF (INDEX)810,748,747
  747 IF (JDNX.GT.0) 729,830
  851 DO 852 I = 1,IDEM
      XNML2(I) = XNML1(I)
      R2F1(I) = R1F1(I)
      R2F4(I) = R1F4(I)
      R2D1(I) = R1D1(I)
  852 R2D4(I) = R1D4(I)
      DO 581 I = 1,IDEM
      DO 581 J = 1,ICM
  581 DDU2(I,J) = DDL1(I,J)
  748 T1 = TIMELEFT(K)
      NZERO = 0
      DO 650 MR = NZERO,ICXII
      DO 650 MS = MR,IMP
      MTB = MS - MR
      MTE = MR + MS
      MM = MTB+1
      NN = MTE+1
      MN = 1-MTB
      TSACKR = TSACKI = 0.
      INT = (-1)**((MS-MR+MTB-2)/2)
      DO 651 MT = MTB,MTE,2
      INT = -INT
      TTERM = (INT*(MT+MT+1.)*BACT(MM+MT)/((NN+MT)*BACT(NN+MT))*
     $BACT(NN-MT)*BACT(MN+MT))
      TSACKR = TSACKR + TTERM*SNGL(BSF(MT+1))
  651 TSACKI = TSACKI + TTERM*SNGL(YMF(MT+1))
      IF (MS.GT.NDR-1.OR.MR.GT.NDR-1) GO TO 650
      SACKR(MS+1+MR*NDR-((MR+1)*MR)/2) = TSACKR
  650 SACKI(MS+1+MR*NDI-((MR+1)*MR)/2) = -TSACKI
      T2 = TIMELEFT(K)
      T = T1 - T2
      PRINT 2000, T
 2000 FORMAT (//5X*TIME TO CALCULATE SACK IS*F8.3,8H SECONDS/)
      T2 = TIMELEFT(K)
      IND = MIN0(IDEMPR,ICXI)
      DO 652 LL = 1,IDEM
      LLL = LL
      LP = LL-1-2*((LL-1)/2)
      LL2 = (LL-LP)/2
      LS = 2*(LL/4)+8
      LS = MIN0(LS,ICXI)
      DO 653 JJ = 1,IDEM
      JJJ = JJ
      JP = JJ-1-2*((JJ-1)/2)
      JJ2 = (JJ-JP)/2
      JSN = JJ2 - 2*(JJ2/2)
      JS = 2*(JJ/4)+10
      JS = MIN0(JS,IDEMPI)
      MN = MIN0(LP+1,JP+1)
      MX = MAX0(LP+1,JP+1)
      SL = SACKI(MX+(MN-1)*NDI-(MN*(MN-1))/2)
```

UNCLASSIFIED

```
      IF (MX.EQ.LP+1) GO TO 406
      IF (2*((MX-MN)/2).NE.MX-MN) SL = -SL
  406 RLIN = DDL1(JJ,1)*SL
      IB = 2
      IE = JS
  405 DO 654 K = IB,IE
      KK = K+K+JP-1
      RLI = RLIN
      KL = KK+LP*NDI-((LP+1)*LP)/2
  654 RLIN = DDL1(JJ,K)*(-1)**(LP+1-KK)*SACKI(KL) - RLI
      IF (ABSF((RLI+RLIN)/RLIN).LT.1.E-8) 401,402
  402 IF (IE.LT.IDEMPI) 403,404
  404 PRINT 499,LL,JJ
  499 FORMAT (//20X*NO CONVERGENCE FOR INITIAL RLI*3X4HLL =I3,3X4HJJ =
     $I3)
      STOP
  403 IB = IE + 1
      IE = MIN0(IB+5,IDEMPI)
      GO TO 405
  401 IF (JSN.EQ.0) RLIN = -RLIN
      CXLIN = DDU2(LL,1)*RLIN
      IBB = 2
      IEE = LS
      ISNCXL = 1
  425 DO 420 MS = IBB,IEE
      MM = MS+MS+LP-1
      ML = MM+JP*NDI-((JP+1)*JP)/2
      RLIN = DDL1(JJ,1)*SACKI(ML)
      IB = 2
      IE = JS
  415 DO 657 K = IB,IE
      KK = K+K+JP-1
      RLI = RLIN
      MN = MIN0(MM,KK)
      MX = MAX0(MM,KK)
      SL = SACKI(MX+(MN-1)*NDI-(MN*(MN-1))/2)
      IF (MX.EQ.MM) GO TO 657
      IF (2*((MX-MN)/2).NE.MX-MN) SL = -SL
  657 RLIN = DDL1(JJ,K)*SL - RLI
      IF (ABSF((RLI+RLIN)/RLIN).LT.1.E-6) 411,412
  412 IF (IE.LT.IDEMPI)413,414
  414 IF (ABSF((RLI+RLIN)/RLIN).LT.1.E-2)411,424
  413 IB = IE + 1
      IE = MIN0(IB+5,IDEMPI)
      GO TO 415
  411 IF (JSN.EQ.0) RLIN = -RLIN
      CXLI = CXLIN
      ISNCXL = -ISNCXL
  420 CXLIN = DDU2(LL,MS)*RLIN - CXLI
      IF (ABSF((CXLI+CXLIN)/CXLIN).LT.1.E-6)421,422
  422 IF (IEE.LT.ICXI  )423,424
  423 IBB = IEE + 1
      IEE = MIN0(IBB+3,ICXI  )
      GO TO 425
  424 IF (ABSF((CXLI+CXLIN)/CXLIN).LT.1.E-2)421,426
  421 IF (LL2.NE.2*(LL2/2))CXLIN = -CXLIN
```

```
      CXLIN = ISNCXL*CXLIN
  653 STORI(JJ) = CXLIN*1.E-205
      GO TO 652
  426 IF (LLL.GT.1) GO TO 430
      IDEM = JJJ - 1
      IDEM2 = 2*IDEM
      PRINT 495,LLL,JJJ,IDEM
  495 FORMAT (45X18HCXI IS 0 FOR LLL =I3,2X5HJJJ = I3,2X
     $10HIDEM NOW =I3/)
      GO TO 652
  430 DO 428 I = JJJ,IDEM
  428 STORI(I) = 0.
      PRINT 497,LLL,JJJ,IDEM
  497 FORMAT (50X*CXI IS 0 FOR LLL = *I3,2X5HJJJ =I3,3H TO I3)
  652 WRITE(7) (STORI(JKL),JKL = 1  ,IDEM)
      IDEM2 = 2*IDEM
      T1 = TIMELEFT(K)
      T = T2 - T1
      PRINT 2001,IDEM,T
 2001 FORMAT (/5X*TIME TO CALCULATE CXLIN FOR IDEM =*I3,3H ISF8.3,8H SEC
     $ONDS/)
      T2 = TIMELEFT(K)
      DO 449 I = 1,IDEM
      DO 469 J = 1,ICM
  469 DDU2(I,J) = 1.E-250*DDU2(I,J)
      DO 449 J = 1,IMPR
  449 DDL1(I,J) = 1.E-250*DDL1(I,J)
      DO 462 LL = 1,IDEM
      LLL = LL
      LP = LL-1-2*((LL-1)/2)
      LL2 = (LL-LP)/2
      LS = 2*(LL/4)+8
      LS = MIN0(LS,IDEMPR)
      DO 463 JJ = 1,IDEM
      JJJ = JJ
      JP = JJ-1-2*((JJ-1)/2)
      JJ2 = (JJ-JP)/2
      JSN = JJ2 - 2*(JJ2/2)
      JS = 2*(JJ/4)+8
      JS = MIN0(JS,IDEMPR)
      MN = MIN0(LP+1,JP+1)
      MX = MAX0(LP+1,JP+1)
      SK = SACKR(MX+(MN-1)*NDR-(MN*(MN-1))/2)
      IF (MX.EQ.LP+1) GO TO 436
      IF (2*((MX-MN)/2).NE.MX-MN) SK = -SK
  436 RLRN = DDL1(JJ,1)*SK
      IB = 2
      IE = JS
  435 DO 464 K = IB,IE
      KK = K+K+JP-1
      RLR = RLRN
      KL = KK+LP*NDR-((LP+1)*LP)/2
  464 RLRN = DDL1(JJ,K)*(-1)**(LP+1-KK)*SACKR(KL) - RLR
      IF (ABSF((RLR+RLRN)/RLRN).LT.1.E-8) 431,432
  432 IF (IE.LT.IDEMPR) 433,434
  434 PRINT 498,LL,JJ
```

UNCLASSIFIED

```
498 FORMAT (//20X*NO CONVERGENCE FOR INITIAL RLR*3X4HLL =I3,3X4HJJ =
   $I3)
    STOP
433 IB = IE + 1
    IE = MIN0(IB+5,IDEMPR)
    GO TO 435
431 IF (JSN.EQ.0) RLRN = -RLRN
    CXLRN = DDU2(LL,1)*RLRN
    IBB = 2
    IEE = LS
    ISNCXL = 1
455 DO 450 MS = IBB,IEE
    MM = MS+MS+LP-1
    ML = MM+JP*NDR-((JP+1)*JP)/2
    RLRN = DDL1(JJ,1)*SACKR(ML)
    IB = 2
    IE = JS
445 DO 467 K = IB,IE
    KK = K+K+JP-1
    RLR = RLRN
    MN = MIN0(MM,KK)
    MX = MAX0(MM,KK)
    SK = SACKR(MX+(MN-1)*NDR-(MN*(MN-1))/2)
    IF (MX.EQ.MM) GO TO 467
    IF (2*((MX-MN)/2).NE.MX-MN) SK = -SK
467 RLRN = DDL1(JJ,K)*SK - RLR
    IF (ABSF((RLR+RLRN)/RLRN).LT.1.E-6) 441,442
442 IF (IE.LT.IDEMPR)443,444
444 IF (ABSF((RLR+RLRN)/RLRN).LT.1.E-2)441,454
443 IB = IE + 1
    IE = MIN0(IB+5,IDEMPR)
    GO TO 445
441 IF (JSN.EQ.0) RLRN = -RLRN
    CXLR = CXLRN
    ISNCXL = -ISNCXL
450 CXLRN = DDU2(LL,MS)*RLRN - CXLR
    IF (ABSF((CXLR+CXLRN)/CXLRN).LT.1.E-6) 451,452
452 IF (IEE.LT.  IND )453,454
453 IBB = IEE + 1
    IEE = MIN0(IBB+3,IND)
    GO TO 455
454 IF (ABSF((CXLR+CXLRN)/CXLRN).LT.1.E-2)451,456
451 IF (LL2.NE.2*(LL2/2))CXLRN = -CXLRN
    CXLRN = ISNCXL*CXLRN
463 STORR(JJ) = CXLRN
    GO TO 462
456 DO 458 I = JJJ,IDEM
458 STORR(I) = 0.
    PRINT 496,LLL,JJJ,IDEM
496 FORMAT (50X*CXR IS 0 FOR LLL = *I3,2X5HJJJ =I3,3H TO I3)
462 WRITE(8) (STORR(JKL),JKL = 1  ,IDEM)
    T1 = TIMELEFT(K)
    T = T2 - T1
    PRINT 2006,IDEM,T
2006 FORMAT (/5X*TIME TO CALCULATE CXLRN FOR IDEM =*I3,3H ISF8.3,8H SEC
   $ONDS/)
```

```
      CALL INTEGR(IDEM2,XS1,ETAL1,ETAU1,XIL,1,FBCT)
      DO 869 J = 1,IDEM
      SIRL1(J) = 0.
      KX = J-1 - 2*((J-1)/2)
      DO 869 I = 1,IDEM
  869 SIRL1(J) = SIRL1(J) + DDL1(J,I)*XIL(2*I-1+KX)
      CALL INTEGR(IDEM2,XS2,ETAL2,ETAU2,XIL,1,FBCT)
      DO 871 J = 1,IDEM
      SIRL2(J) = 0.
      KX = J-1 - 2*((J-1)/2)
      DO 871 I = 1,IDEM
  871 SIRL2(J) = SIRL2(J) + DDU2(J,I)*XIL(2*I-1+KX)
      REWIND 7
      REWIND 8
      DO 750 I = 1,IDEM
      DCS(I) = R1D1(I)*R1F4(I)/R1D4(I)/XNML1(I)
  750 DCS(IDEM+I) = R2D1(I)/R2D4(I)/XNML2(I)
      DO 649 I = 1,IDEM
      DO 649 J = 1,IDEM
  649 CX(I,J) = CX(I+IDEM,J+IDEM) = (0.,0.)
      DO 351 I = 1,IDEM
      READ (7) (STORI(II),II=1,IDEM)
      READ (8) (STORR(II),II=1,IDEM)
      DO 351 J = 1,IDEM
  351 CX(I+IDEM,J) = CX(J,IDEM+I) = CMPLX(STORR(J),STORI(J))
      DO 353 I = 1,IDEM
      DO 353 J = 1,IDEM
      CX(I,J+IDEM) = CX(I,J+IDEM)*2./XNML1(I)
  353 CX(I+IDEM,J) = CX(I+IDEM,J)*2./XNML2(I)
      DO 751 I = 1,IDEM
      TRS1(I) = TRS2(I) = (0.,0.)
      DO 751 J = 1,IDEM
      TRS1(I) = TRS1(I) + CX(J,IDEM+I)*R1F1(J)*SIRL1(J)
  751 TRS2(I) = TRS2(I) + CX(IDEM+J,I)*R2F1(J)*SIRL2(J)
      DO 350 I = 1,IDEM
      DO 350 J = 1,IDEM
      CX(I,J+IDEM) = CX(I,J+IDEM)*R1D1(I)/R1D4(I)
  350 CX(I+IDEM,J) = CX(I+IDEM,J)*R2D1(I)/R2D4(I)
      DO 806 I = 1,IDEM2
  806 CX(I,I) = CMPLX(1.,0.)
      T2 = TIMELEFT(K)
      CALL CINV(CX,IDEM2,60)
      T1 = TIMELEFT(K)
      T = T2 - T1
      PRINT 915,IDEM2,T
  915 FORMAT (/5X*TIME TO INVERT MATRIX OF SIZE *I2,4H IS F8.3,8H SECOND
     $S)
      WRITE (10) ((CX(I,J),I=1,IDEM2),J=1,IDEM2)
      PRINT 5561
 5561 FORMAT (//38X*ACOUSTIC IMPEDANCE CONTRIBUTIONS (FOR CONVERGENCE CH
     $ECK)*/)
      SIMP1 = SIMP2 = SIMP3 = SIMP4 = SIMP5 = SIMP6 = (0.,0.)
      SIMP7 = SIMP8 = SIMP9 = (0.,0.)
      DO 752 I = 1,IDEM
      SIMP1 = SIMP1 + R1F1(I)*R1F4(I)*SIRL1(I)*SIRL1(I)/XNML1(I)
      SIMP4 = SIMP4 + R2F1(I)*SIRL2(I)*TRS1(I)/XNML2(I)
```

UNCLASSIFIED

```
      SIMP7 = SIMP7 + R2F1(I)*R2F4(I)*SIRL2(I)*SIRL2(I)/XNML2(I)
  752 PRINT 940, I,SIMP1,SIMP4,SIMP7
  940 FORMAT (2XI3,5X7HSIMP1 =C(E15.7,F15.7),5X7HSIMP4 =C(E15.7,E15.7),
     $5X7HSIMP7 =C(E15.7,E15.7))
      PRINT 5056
 5056 FORMAT (//)
      DO 753 I = 1,IDEM
      SUM = SVM = (0.,0.)
      DO 754 J = 1,IDEM
      SUM = SUM + CX(I,J)*DCS(J)*SIRL1(J) + CX(I,IDEM+J)*DCS(IDEM+J)
     $*TRS1(J)
      SVM = SVM + CX(I,J)*DCS(J)/R1F4(J)*TRS2(J) + CX(I,IDEM+J)
     $*DCS(IDEM+J)*R2F4(J)*SIRL2(J)
  754 CONTINUE
      SIMP2 = SIMP2 + R1F4(I)*SIRL1(I)*SUM
      SIMP5 = SIMP5 + TRS2(I)*SUM
      SIMP8 = SIMP8 + TRS2(I)*SVM
  753 PRINT 941, I,SIMP2,SIMP5,SIMP8
  941 FORMAT (2XI3,5X7HSIMP2 =C(E15.7,E15.7),5X7HSIMP5 =C(E15.7,E15.7),
     $5X7HSIMP8 =C(E15.7,E15.7))
      PRINT 5056
      DO 755 I = 1,IDEM
      SUM = SVM = (0.,0.)
      DO 756 J = 1,IDEM
      SUM = SUM + CX(I+IDEM,J)*DCS(J)*SIRL1(J) + CX(IDEM+I,IDEM+J)
     $*DCS(IDEM+J)*TRS1(J)
      SVM = SVM + CX(I+IDEM,J)*DCS(J)/R1F4(J)*TRS2(J) + CX(IDEM+I,IDEM+J
     $)*DCS(IDEM+J)*R2F4(J)*SIRL2(J)
  756 CONTINUE
      SIMP3 = SIMP3 + TRS1(I)*SUM
      SIMP6 = SIMP6 + R2F4(I)*SIRL2(I)*SUM
      SIMP9 = SIMP9 + R2F4(I)*SIRL2(I)*SVM
  755 PRINT 942, I,SIMP3,SIMP6,SIMP9
  942 FORMAT (2XI3,5X7HSIMP3 =C(E15.7,E15.7),5X7HSIMP6 =C(E15.7,E15.7),
     $5X7HSIMP9 =C(E15.7,E15.7))
      SIMP3 = (SIMP1 - SIMP2 - SIMP3)*SQRTF(EXEX1)*HS1**2/SPDINT(1)
      SIMP6 = (SIMP4 - SIMP5 - SIMP6)*SQRTF(EXEX2)*HS2**2/SPDINT(1)
      SIMP9 = (SIMP7-SIMP8-SIMP9)*SQRTF(EXEX2)*HS2**2/SPDINT(2)
      SIMP7 = SIMP6*SPDINT(1)/SPDINT(2)*(HS1/HS2)**2*SQRTF(EXEX1/EXEX2)
      P1S = CABS(SIMP3)
      P1A = CANG(SIMP3)*180./PI
      P1SS = CABS(SIMP7)
      P1AA = CANG(SIMP7)*180./PI
      P2S = CABS(SIMP9)
      P2A = CANG(SIMP9)*180./PI
      P2SS = CABS(SIMP6)
      P2AA = CANG(SIMP6)*180./PI
      PRINT 5562
 5562 FORMAT (/47X*NORMALIZED ACOUSTIC RADIATION IMPEDANCES*/
     $/42X10HRESISTANCE5X9HREACTANCE8X9HMAGNITUDE6X*PHASE ANGLE (DEGREES
     $)*/)
      PRINT 943,SIMP3,P1S,P1A,SIMP7,P1SS,P1AA,SIMP9,P2S,P2A,SIMP6,
     $P2SS,P2AA
  943 FORMAT (34X5HZ11 =C(E13.5,E15.5),5XE12.5,6XF9.3/34X
     $5HZ21 =C(E13.5,E15.5),5XE12.5,6XF9.3/34X
     $5HZ22 =C(E13.5,E15.5),5XE12.5,6XF9.3/34X
```

```
   $5HZ12 =C(E13.5,E15.5),5XF12.5,6XF9.3)
    REWIND 10
    GO TO 810
830 READ  (10) ((CX(I,J),I=1,IDEM2),J=1,IDEM2)
    DO 860 I = 1,IDEM
    DACS(I) = S1(I)*R41(I)*DCS(I)/R1F4(I)
860 DACS(I+IDEM) = S2(I)*R42(I)*DCS(IDEM+I)
    DO 749 I = 1,IDEM2
    CTS(I) = (0.,0.)
    DO 749 J = 1,IDEM2
749 CTS(I) = CTS(I) + CX(I,J)*DACS(J)
    SUM1 = SUM2 = STOR1 = STOR2 = (0.,0.)
    REWIND 10
    IF (IVEL1.EQ.0) GO TO 731
    ISP = 1
    PRINT 925,ISP
925 FORMAT (//30X*ACOUSTIC PRESSURE CONTRIBUTIONS DUE TO SPHEROID*I2,
   $* (FOR CONVERGENCE CHECK)*/)
    SM1 = SM2 = SM3 = (0.,0.)
    DO 872 I = 1,IDEM
    SM1 = SM1 + S1(I)*R1F1(I)*R41(I)*SIRL1(I)/XNML1(I)
872 PRINT 920, I,SM1
920 FORMAT (13XI3,5X5HSM1 =C(E15.7,E15.7))
    PRINT 5056
    DO 873 I = 1,IDEM
    SM2 = SM2 + CTS(I)*R1F4(I)*SIRL1(I)
873 PRINT 921, I,SM2
921 FORMAT (13XI3,5X5HSM2 =C(E15.7,E15.7))
    PRINT 5056
    DO 874 I = 1,IDEM
    SM3 = SM3 + CTS(IDEM+I)*TRS1(I)
874 PRINT 923,I,SM3
923 FORMAT (13XI3,5X5HSM3 =C(E15.7,E15.7))
    PRMLTR = HS1**2*CEF*RHO*SQRTF(EXEX1)*VEL1
    SUM1 = (SM1 - SM2 - SM3)*PRMLTR
    PRS = CABS(SUM1)
    PRA = CANG(SUM1)*180./PI
    IF (NFF.EQ.1) GO TO 760
    ISP = 1
    PRINT 924,ISP,SUM1,PRS,PRA
924 FORMAT (//15X*SPHEROID*I2,* CONTRIBUTION TO THE ACOUSTIC PRESSURE
   $(NTS/SQUARE METER) AT THE INPUT NEARFIELD POINT*//33X9HREAL PART3X
   $14HIMAGINARY PART11X9HMAGNITUDE5X*PHASE ANGLE (DEGREES)*//30X
   $C(E13.5,E14.5),10XE13.5,5XF9.3)
    GO TO 731
760 ISBT = ISBT + 1
    STOR1 = SUM1*CMPLX(COSF(.5*ART),-SINF(.5*ART))
    ANGLE(ISBT) = AAR
    ISBT = ISBT - 1
731 IF (IVEL2.EQ.0) GO TO 732
    ISP = 2
    PRINT 925,ISP
    SM1 = SM2 = SM3 = (0.,0.)
    DO 875 I = 1,IDEM
    SM1 = SM1 + S2(I)*R2F1(I)*R42(I)*SIRL2(I)/XNML2(I)
875 PRINT 920,I,SM1
```

UNCLASSIFIED

```
      PRINT 5056
      DO 876 I = 1,IDEM
      SM2 = SM2 + CTS(I)*TRS2(I)
876   PRINT 921,I,SM2
      PRINT 5056
      DO 877 I = 1,IDEM
      SM3 = SM3 + CTS(I+IDEM)*R2F4(I)*SIRL2(I)
877   PRINT 923,I,SM3
      PRMLTR = HS2**2*CFF*RHO*SQRTF(EXEX2)*VEL2
      SUM2 = (SM1 - SM2 - SM3)*PRMLTR
      PRS = CABS(SUM2)
      PRA = CANG(SUM2)*180./PI
      IF (NFF.EQ.1) GO TO 761
      ISP = 2
      PRINT 924,ISP,SUM2,PRS,PRA
732   IF (NFF.EQ.1) GO TO 733
      SUMM = SUM1 + SUM2
      PRS = CABS(SUMM)
      PRA = CANG(SUMM)*180./PI
      PRINT 927,XCT,YCT,ZCT,SUMM,PRS,PRA
927   FORMAT (//12X*TOTAL ACOUSTIC PRESSURE (NTS/SQUARE METER) AT THE IN
     $PUT NEARFIELD POINT (*E12.5,1H,E12.5,1H,E12.5,1H)//33X9HREAL PART
     $3X14HIMAGINARY PART11X9HMAGNITUDE5X*PHASE ANGLE (DEGREES)*//30X
     $C(E13.5,E14.5),10XE13.5,5XF9.3)
      GO TO 810
761   ISBT = ISBT + 1
      STOR2 = SUM2*CMPLX(COSF(.5*ART),-SINF(.5*ART))
      ANGLE(ISBT) = AAB
      ISBT = ISBT - 1
733   ISBT = ISBT + 1
      FF(ISBT) = STOR1 + STOR2
      GO TO 810
723   IF (NFFF.EQ.0) GO TO 811
      DO 725 I = 2,ISBT
      FF(I) = FF(I)/FF(1)
      FFA(I) = CANG(FF(I))*180./PI
725   FFM(I) = CABS(FF(I))
      FF(1) = FFM(1) = 1.
      PRINT 929
929   FORMAT (//53X*NORMALIZED FARFIELD PRESSURE*//16X*ANGLE (DEGREES)*
     $9X9HREAL PART3X14HIMAGINARY PART11X9HMAGNITUDE5X
     $*PHASE ANGLE (DEGREES)*/)
930   FORMAT (20XF7.2,10XC(E13.5,E14.5),10XE13.5,5XF9.3)
      PRINT 930, (ANGLE(I),FF(I),FFM(I),FFA(I),I=1,ISBT)
      IF (NFF.EQ.2) GO TO 810
811   CONTINUE
      STOP
      END
```

```
      SUBROUTINE CINV(B,NS,ND)
      TYPE COMPLEX B,D
      DIMENSION B( ND,1),L(100),M(100)
      DO 1 J = 1,NS
      DO 1 I = 1,NS
    1 B(I+NS*(J-1)) = B(I+ND*(J-1))
      CALL MINC(B,NS,ND,D,L,M)
      DO 2 JJ = 1,NS
      J = NS+1-JJ
      DO 2 II = 1,NS
      I = NS+1-II
    2 B(I+ND*(J-1)) = B(I+NS*(J-1))
      RETURN
      END
```

UNCLASSIFIED

```
      SUBROUTINE MINC(A,N,ND,D,L,M)
      TYPE COMPLEX A,D,BIGA,BIGAI,HOLD,B
      EQUIVALENCE(B,C)
      DIMENSION C(2),C1(2)
      EQUIVALENCE(BIGA,C1)
      DIMENSION A(ND,1),L(1),M(1)
      D= (1.0,0.0)
      NK = -N
      DO 80 K=1,N
      NK = NK+N
      L(K) = K
      M(K) = K
      KK = NK+K
      BIGA = A(KK)
      BIGAI = (1.0,0.0)/BIGA
      J=L(K)
      IF(J-K) 35,35,25
   25 KI = K-N
      DO 30 I=1,N
      KI = KI+N
      HOLD = -A(KI)
      JI = KI -K+J
      A(KI)=A(JI)
   30 A(JI) = HOLD
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP= N*(I-1)
      DO 40 J=1,N
      JK = NK+J
      JI = JP+J
      HOLD = -A(JK)
      A(JK) = A(JI)
   40 A(JI) = HOLD
   45 IF(C1(1).EQ.0..AND.C1(2).EQ.0.)46,48
   46 D= 0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK = NK + I
      A(IK) = -BIGAI*A(IK)
   55 CONTINUE
      DO 65 I=1,N
      IK = NK+I
      HOLD = A(IK)
      IJ = I-N
      DO 65 J=1,N
      IJ = IJ+N
      IF(I.EQ.K.OR.J.EQ.K) GO TO 65
   62 KJ = IJ -I +K
      A(IJ) = HOLD*A(KJ) + A(IJ)
   65 CONTINUE
      KJ = K-N
      DO 75 J=1,N
      KJ = KJ + N
      IF(J-K) 70,75,70
   70 A(KJ) = BIGAI*A(KJ)
   75 CONTINUE
      D = D*BIGA
      A(KK) = BIGAI
   80 CONTINUE
  150 RETURN
      END
```

```
      SUBROUTINE EIGEN(A,VALU,N,N1,ANORM1,ANORM2)
      DIMENSION A(80,80),VALU(80),DIAG(80),Q(80),VALL(85)
      NN=N-2
      DO 160 I=1,NN
      II=I+2
      DO 160 J=II,N
      T1=A(I,I+1)
      T2=A(I,J)
      IF(T2.EQ.0.)  GO TO 160
      T=1./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+1,K)+SIN*A(J,K)
      A(J,K)=COS*A(J,K)-SIN*A(I+1,K)
  125 A(I+1,K)=T2
  160 CONTINUE
      DO15I=1,N
      DIAG(I)=A(I,I)
      Q(I)=A(I,I-1)*A(I,I-1)
      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
      T0=0.
      T1=1.E-100
      DO20J=1,N
      T2=(DIAG(J)-TAU)*T1-Q(J)*T0
      IF((T1.NE.0.).AND.((T2*T1).LE.0.))  MATCH=MATCH+1
      T0=T1
   20 T1=T2
      DO25K=I,MATCH
   25 VALU(K)=TAU
      NATCH=MATCH+1
      DO 30 K=NATCH,LATCH
   30 IF(TAU.GT.VALL(K))       VALL(K)=TAU
   40 IF((VALU(I)-VALL(I)).GT.(1.E-4))  GO TO 18
      I=I+1
      MATCH = N
      IF(I.LE.N1) GO TO 40
      END
```

UNCLASSIFIED

```
 SUBROUTINE INTEGR(LN,TSI,ETA1,ETA2,A,INC,FBCT)
 TYPE DOUBLE A,ARG,BARG,BETA,CEF,ETA1,ETA2,FACT,FARG,FSCT,RTSUM1,
1RTSUM2,SUM,TSI,TSISQ,W,X,Y,YS,Z,FBCT,SMM
 DIMENSION A(1),ARG(96),BARG(96),CEF(96),FARG(96),SUM(80),W(96),
1X(96),Y(96),YS(96),Z(96),FBCT(300),SMM(80)
 COMMON/BLK1/FACT(300)
 DATA(X= -0.016276744849602969579D000,
*-0.048812985136049731112D,
*         -0.081297495464425558994D000,
*         -0.113695850110665920911D000,
*         -0.145973714654896941989D000,
*-0.178096882367618602759D000,-0.210031310460567203603D000,
*-0.241743156163840012328D000,-0.273198812591049141487D000,
*-0.304364944354496353024D000,-0.335208522892625422616D000,
*-0.365696861472313635031D000,-0.395797649828908603285D000,
*-0.425478988407300545365D000,-0.454709422167743008636D000,
*-0.483457973920596359768D000,-0.511694177154667673586D000,
*-0.539388108324357436227D000,-0.566510418561397168404D000,
*-0.593032364777572080684D000,-0.618925840125468570386D000,
*-0.644163403784967106798D000,
*-0.668718310043916153953D000,-0.692564536642171561344D000,
*-0.715676812348967626225D000,-0.738030643744400132851D000,
*-0.759602341176647498703D000,-0.780369043867433217604D000,
*-0.800308744139140817229D000,-0.819400310737931675539D000,
*-0.837623511228187121494D000,-0.854959033434601455463D000,
*-0.871388505909296502874D000,-0.886894517402420416057D000,
*-.901460635315852341319D000,-0.915071423120898074206D000,
*-0.927712456722308690965D000,-0.939370339752755216932D000,
*-0.950032717784437635756D000,-0.959688291448742539300D000,
*-0.968326828463264212174D000,-0.975939174585136466453D000,
*-0.982517263563014677447D000,-0.988054126329623799481D000,
*-0.992543900323762624572D000,-0.995981842987209290650D000,
*-0.998364375863181677724D000,-0.999689503883230766828D000)
 DATA(W=0.032550614492363166242D,0.032516118713868835987D,
*0.032447163714064269364D,0.032343822568575928429D,
*0.032206204794030250669D,0.032034456231992663218D,
*0.031828758894411006535D,0.031589330770727168558D,
*0.031316425596861355813D,0.031010332586313837423D,
*0.030671376123669149014D,0.030299915420827593794D,
*0.029896344136328385984D,0.029461089958167905970D,
*0.028994614150555236543D,0.028497411065085385646D,
*0.027970007616848334440D,0.027412962726029242823D,
*0.026826866725591762198D,0.026212340735672413913D,
*0.025570036005349361499D,0.024900633222483610288D,
*0.024204841792364691282D,0.023483399085926219842D,
*0.022737069658329374001D,0.021966644438744349195D,
*0.021172939892191298988D,.020356797154333324595D,
*0.019519081140145022410D, 0.018660679627411467385D,
*0.017782502316045260838D,0.016885479864245172450D,
*0.015970562902562291381D,0.015038721026994938006D,
*0.014090941772314860916D,0.013128229566961572637D,
*0.012151604671088319635D,0.011162102099838498591D,
*0.010160770535008415758D,0.009148671230783386633D,
*0.008126876925698759217D,0.007096470791153865269D,
*0.006058545504235961683D,0.005014202742927517693D,
*0.003964554338444686674D,0.002910731817934946408D,
```

```
     *0.0018539607889469217 32D,0.00079679206555201242 9D)
      IFL = 0
      LNM1 = LN - 1
      DO 5I=1,48
      X(48+I)=-X(I)
    5 W(I+48)=W(I)
      DO 1 I = 1,80
    1 SMM(I) = 0.D
      TSISQ=TSI*TSI
      IF (ETA2*ETA1.LT.0.D.AND.TSI.EQ.0.D) GO TO 2
      RTSUM1=0.5D*(ETA2-ETA1)
      RTSUM2=0.5D*(ETA1+ETA2)
      GO TO 3
    2 RTSUM1 = -.5D*ETA1
      RTSUM2 = -RTSUM1
      IFL = 1
    3 A(1)=0.D
      A(2)=0.D
      DO630 I=1,96
      Y(I)=RTSUM1*X(I)+RTSUM2
      YS(I)=Y(I)*Y(I)
      Z(I)=DSQRT((1.D)-YS(I))
      CEF(I)=W(I)*DSQRT(TSISQ+YS(I))*RTSUM1
      A(1)=A(1)+CEF(I)
      A(2)=A(2)+CEF(I)*Y(I)
      BARG(I)=0.D
      FARG(I)=1.D
  630 ARG(I)=Y(I)
      SUM(1)=A(1)
      SUM(2)=A(2)
      SMM(1) = SMM(1) + SUM(1)
      SMM(2) = SMM(2) + SUM(2)
      A(1) = SMM(1)
      A(2) = SMM(2)
      DO 650 L = 3,LN
      SUM(L) = 0.D
      DO651 I=1,96
      BARG(I)=BARG(I)*Y(I)+FARG(I)*Z(I)
      FARG(I)=ARG(I)
      ARG(I)=ARG(I)*Y(I)-BARG(I)*Z(I)
  651 SUM(L) = SUM(L)+ARG(I)*CEF(I)
  650 SMM(L) = SMM(L) + SUM(L)
      IF (IFL.EQ.0) GO TO 631
      RTSUM1 = RTSUM2 = .5D*ETA2
      IFL = 0
      GO TO 3
  631 DO 680 N = 2,LNM1,INC
      LIMIT=N/2
      A(N+1)=(FBCT(N+N)/FBCT(N))/(2.D*FACT(N+1))*SMM(N+1)
      DO 660 L=1,LIMIT
      BETA=(FACT(L+L)/(FACT(L)*FACT(L+1)*FACT(N-L+1)))
     1*(FBCT(N+N-L-L)/FBCT(N-L))
      IF((2*L).EQ.N)  BETA=.5D*BETA
  660 A(N+1)=A(N+1)+BETA*SMM(N-L-L+1)
  680 A(N+1)=0.5D**(N+N-3)*A(N+1)
      END
```

UNCLASSIFIED

```
      SUBROUTINE POLY(LN,ARG,P,FBCT)
      TYPE DOUBLE FACT,FBCT,P,ANGLE,TERM,SUM,ARG,S,Y
      DIMENSION P(125),Y(150),FBCT(300)
      COMMON/BLK1/FACT(300)
      II = 0
      M = 0
      IF (DABS(1.D0-ARG).LT.1.D-10) ARG = 1.D
      ANGLE = DATAN2(DSQRT(1.D0 - ARG*ARG),ARG)
      LNN=LN+1
      LDN = LN + 2
      DO 2 N = 1,LNN
      P(N) = 0.D
    2 Y(N) = DCOS((N-1)*ANGLE)
      MM = 1
      M2 = 2
      IF ((ANGLE.NE.0.).AND.((ANGLE.LT.3.1415926).OR.(ANGLE.GT.3.1415926
     19))) GO TO 6
      ARG = 1.
      DO 5 N=1,LNN
    5 P(N)= 1.D
      IF( ANGLE .EQ.0) GO TO 14
      DO 4 N=2,LNN,2
    4 P(N)=-P(N)
      GO TO 14
    6 IF ((ANGLE.LT.1.570796320).OR.(ANGLE.GT.1.570796329))GO TO 10
      ARG = 0.
    3 P(1) = 1.D
      DO 7 N = 2,LN,2
      P(N+1) = FACT(N)/(FACT(N/2)*FACT(N/2+1)*2.D**(N-1))
    7 IF (2*((N-M)/4).NE.(N-M)/2)  P(N+1)=-P(N+1)
      GO TO 14
   10 P(MM) = 1.D
   11 DO 13 N=MM,LN
      S = 0.D
      IN = (N-M-1)/2
      DO 12 I = II,IN
   12 S = S+FACT(2*M+2*I+1)*FBCT(2*N-2*I+1)*Y(N-M-2*I+1)/(2.D**
     1(M+N)*FBCT(N-I+1)*FACT(MM+I)*FACT(I+1)*FACT(N-M-I+1))
      P(N+1) = S/2.**(N-1)
   13 IF (2*((M+N)/2).EQ.M+N) P(N+1)=P(N+1)+(FACT(N+1)/
     1(FACT((N+M)/2+1)*FACT((N-M)/2+1)))**2/(FACT(2*M+1)*2.D**(2*N-M))
   14 RETURN
      END
```

```
      SUBROUTINE QLEG(M,LN,X,Q)
      DIMENSION Q(200)
      COMMON/BLK1/FACT(300)
      TYPE DOUBLE BE,COEF,DI,DK,DM,DN,FACT,GA,Q,SUM,TERM,X,XA,YA,Z,ZA
      DM=M
      NN=0
      YA=DSQRT(X*X+1.D)
      ZA=(YA+X)*(YA+X)
      XA=0.25D*(YA+X)
      Z=2.D/(YA+X)
      LNM=LN+1
      DO 135 N=NN,LNM
      DN=N
      BE=DN+1.D
      GA=DN+1.5D
      IF(N.GT.84) GO TO 500
      COEF=Z*XA**(-N)*(FACT(N+1)   /FACT(N+N+2) *FACT(N+1))
      GO TO 510
  500 COEF=Z*XA**(-N)*(FACT(N+1)   *(1.D-300)*FACT(N+1)/FACT(N+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.1.E-23)  GO TO 130
  135 Q(N+1)=-COFF*SUM
      DO 30 I=1,M
      DI=I
      DO 30 N=NN,LN
      DN=N
   30 Q(N+1)=-((DI+DN)*X*Q(N+1)+(DN-DI+2.D)*Q(N+2))/YA
      DO 40 N=NN,LN
      IF((2*((N+1)/4)).NE.((N+1)/2)) Q(N+1)=-Q(N+1)
   40 IF(2*(N/2).NE.N) Q(N+1)=-Q(N+1)
      END
```

UNCLASSIFIED

```
      SUBROUTINE SBESF(XH,LJ,RAY)
      DIMENSION RAY(250)
      TYPE DOUBLE CP,FACT,RAY,SUM,TERM,TK,TM,XH,XI,Z2H
      COMMON/BLK1/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.84) TM=TM*1.D-300
      IF(TM.EQ.0.D) GO TO 8
      SUM=1.D
      TERM = 1.D
      DO 2 I=1,50
      XI=I*(N+N+I+I+1)
      TERM=-TERM*Z2H/XI
      SUM = SUM + TERM
    2 IF(DABS(TERM/SUM).LT.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
   11 RAY(K+2)=(K+K+1)*RAY(K+1)/XH-RAY(K)
      RETURN
   20 IF(XH.GT.10.D) N=210
      RAY(N+1)=1.D-250
      RAY(N+2)=0.D
      I = -N
      M = -1
      DO 5 KK=I,M
      K = -KK
      TK=K+K+1
    5 RAY(K)=TK*RAY(K+1)/XH-RAY(K+2)
      CP=DSIN(XH)/(XH*RAY(1))
      IF(DSIN(XH).LT.1.D-2) CP=(DSIN(XH)/XH-DCOS(XH))/(XH*RAY(2))
      DO 6 L=1,LJ
    6 RAY(L)=CP*RAY(L)
    8 DO 9 J=N,LJ
    9 RAY(J+1)=0.D
      END


      SUBROUTINE SPHYN(X,N,ARR)
      DIMENSION ARR(250)
      TYPE DOUBLE X,ARR,TKP1
      ARR(1)=-DCOS(X)/X
      ARR(2)=ARR(1)/X-DSIN(X)/X
      DO 2 K=1,N
      TKP1=K+K+1
    2 ARR(K+2)=TKP1*ARR(K+1)/X-ARR(K)
      END
```

Appendix C

SAMPLE INPUT AND OUTPUT FROM OSPRDS

INPUT

```
   10    40    20
1.             2.
.5             1.
4.
1 1.                    180.          0.
1 1.                    180.          0.
1
0.
1
90.
2
0
                        2.
```

OUTPUT

ACOUSTIC RADIATION FROM TWO OBLATE SPHEROIDS

	SHAPE PARAMETER XI	ACOUSTIC SIZE PARAMETER H
SPHEROID 1	1.000000	2.000000
SPHEROID 2	0.500000	1.000000

SEPARATION PARAMETER

ALPHA = 4.000000

LOWER AND UPPER LIMITS (IN DEGREES) OF VIBRATING BANDS

	LOWER	UPPER
SPHEROID 1	180.000	0.000
SPHEROID 2	180.000	0.000

NORMAL VELOCITY (IN M/SEC) OF VIBRATING BANDS

	REAL	IMAGINARY	MAGNITUDE	PHASE ANGLE (DEGREES)
SPHEROID 1	1.000	-0.000	1.000	0.000
SPHEROID 2	1.000	-0.000	1.000	0.000

SUMMATION PARAMETERS

IDEM = 10 IDEMPI = 40 IDEMPR = 20

TIME TO CALCULATE SACK IS 15.938 SECONDS

TIME TO CALCULATE CXLIN FOR IDEM = 10 IS 4.408 SECONDS

TIME TO CALCULATE CXLRN FOR IDEM = 10 IS 2.809 SECONDS

TIME TO INVERT MATRIX OF SIZE 20 IS 1.703 SECONDS

ACOUSTIC IMPEDANCE CONTRIBUTIONS (FOR CONVERGENCE CHECK)

1	SIMP1 =	1.4522280-001	-1.9119236-001	SIMP4 =	7.4569529-002	-2.9625104-002	SIMP7 =	8.2254194-001	6.6129655-001
2	SIMP1 =	1.4522280-001	-1.9119236-001	SIMP4 =	7.4569529-002	-2.9625104-002	SIMP7 =	8.2254194-001	6.6129655-001
3	SIMP1 =	1.4699060-001	-1.8786808-001	SIMP4 =	7.3537677-002	-2.8696971-002	SIMP7 =	8.2260930-001	6.7095088-001
4	SIMP1 =	1.4699060-001	-1.8786808-001	SIMP4 =	7.3537677-002	-2.8696971-002	SIMP7 =	8.2260930-001	6.7095088-001
5	SIMP1 =	1.4699072-001	-1.8785930-001	SIMP4 =	7.3537754-002	-2.8694595-002	SIMP7 =	8.2260930-001	6.7105187-001
6	SIMP1 =	1.4699072-001	-1.8785930-001	SIMP4 =	7.3537754-002	-2.8694595-002	SIMP7 =	8.2260930-001	6.7105187-001
7	SIMP1 =	1.4699072-001	-1.8785925-001	SIMP4 =	7.3537756-002	-2.8694595-002	SIMP7 =	8.2260930-001	6.7105400-001
8	SIMP1 =	1.4699072-001	-1.8785925-001	SIMP4 =	7.3537756-002	-2.8694595-002	SIMP7 =	8.2260930-001	6.7105400-001
9	SIMP1 =	1.4699072-001	-1.8785925-001	SIMP4 =	7.3537756-002	-2.8694595-002	SIMP7 =	8.2260930-001	6.7105409-001
10	SIMP1 =	1.4699072-001	-1.8785925-001	SIMP4 =	7.3537756-002	-2.8694595-002	SIMP7 =	8.2260930-001	6.7105409-001

UNCLASSIFIED

1	SIMP2 =	-1.9778769-001	-3.1727536-001	SIMP5 =	4.1120634-003	-1.0639421-001	SIMP8 =	1.8198716-002 -1.8803930-002
2	SIMP2 =	-1.9778769-001	-3.1727536-001	SIMP5 =	4.1787925-003	-1.0640959-001	SIMP8 =	1.7619066-002 -1.9549234-002
3	SIMP2 =	-2.0001192-001	-3.1819350-001	SIMP5 =	2.7119211-003	-9.9407364-002	SIMP8 =	3.3037704-002 -1.7936790-002
4	SIMP2 =	-2.0001192-001	-3.1819350-001	SIMP5 =	2.6764570-003	-1.0008250-001	SIMP8 =	2.4286124-002 -1.3377959-002
5	SIMP2 =	-2.0000645-001	-3.1819812-001	SIMP5 =	2.7600485-003	-1.0014799-001	SIMP8 =	2.4555585-002 -1.4906135-002
6	SIMP2 =	-2.0000645-001	-3.1819812-001	SIMP5 =	2.7536541-003	-1.0014255-001	SIMP8 =	2.4679058-002 -1.4844630-002
7	SIMP2 =	-2.0000644-001	-3.1819819-001	SIMP5 =	2.7527246-003	-1.0014297-001	SIMP8 =	2.4675591-002 -1.4837124-002
8	SIMP2 =	-2.0000644-001	-3.1819819-001	SIMP5 =	2.7527311-003	-1.0014299-001	SIMP8 =	2.4675218-002 -1.4837149-002
9	SIMP2 =	-2.0000644-001	-3.1819819-001	SIMP5 =	2.7527324-003	-1.0014299-001	SIMP8 =	2.4675209-002 -1.4837161-002
10	SIMP2 =	-2.0000644-001	-3.1819819-001	SIMP5 =	2.7527324-003	-1.0014299-001	SIMP8 =	2.4675209-002 -1.4837161-002

1	SIMP3 =	7.4292797-004	-1.2140386-003	SIMP6 =	1.8710779-002	6.3037572-004	SIMP9 =	1.9123912-001 -8.9916783-003
2	SIMP3 =	9.4291474-004	-4.9400009-003	SIMP6 =	1.8710779-002	6.3037572-004	SIMP9 =	1.9123912-001 -8.9916783-003
3	SIMP3 =	9.9487897-004	-4.7647486-003	SIMP6 =	1.9862718-002	1.1688905-003	SIMP9 =	1.9088660-001 -1.5527726-002
4	SIMP3 =	9.9014513-004	-4.7677790-003	SIMP6 =	1.9862718-002	1.1688905-003	SIMP9 =	1.9088660-001 -1.5527726-002
5	SIMP3 =	9.9020678-004	-4.7678052-003	SIMP6 =	1.9865372-002	1.1678646-003	SIMP9 =	1.9088684-001 -1.5616264-002
6	SIMP3 =	9.9020681-004	-4.7678047-003	SIMP6 =	1.9865372-002	1.1678646-003	SIMP9 =	1.9088684-001 -1.5616264-002
7	SIMP3 =	9.9020681-004	-4.7678047-003	SIMP6 =	1.9865372-002	1.1678620-003	SIMP9 =	1.9088684-001 -1.5618252-002
8	SIMP3 =	9.9020681-004	-4.7678047-003	SIMP6 =	1.9865372-002	1.1678620-003	SIMP9 =	1.9088684-001 -1.5618252-002
9	SIMP3 =	9.9020681-004	-4.7678047-003	SIMP6 =	1.9865372-002	1.1678620-003	SIMP9 =	1.9088684-001 -1.5618337-002
10	SIMP3 =	9.9020681-004	-4.7678047-003	SIMP6 =	1.9865372-002	1.1678620-003	SIMP9 =	1.9088684-001 -1.5618337-002

NORMALIZED ACOUSTIC RADIATION IMPEDANCES

	RESISTANCE	REACTANCE	MAGNITUDE	PHASE ANGLE (DEGREES)
Z11 =	8.52641-001	3.32934-001	9.15337-001	21.329
Z21 =	1.94764-001	2.68818-001	3.31958-001	54.076
Z22 =	4.58909-001	5.30319-001	7.01310-001	49.129
Z12 =	2.47997-002	3.42292-002	4.22689-002	54.076

FAR FIELD ANGLE = 0.00 DEGREES

ACOUSTIC PRESSURE CONTRIBUTIONS DUE TO SPHEROID 1 (FOR CONVERGENCE CHECK)

1	SM1 =	0.0000000+000	4.0141061-001
2	SM1 =	-7.6478522-026	4.0141061-001
3	SM1 =	-7.6478522-026	4.6643571-001
4	SM1 =	-5.2562131-026	4.6643571-001
5	SM1 =	-5.2562131-026	4.6570470-001
6	SM1 =	-5.6351147-026	4.6570470-001
7	SM1 =	-5.6351147-026	4.6570123-001
8	SM1 =	-5.6168851-026	4.6570123-001
9	SM1 =	-5.6168851-026	4.6570122-001
10	SM1 =	-5.6172360-026	4.6570122-001

1	SM2 =	6.1285108-001	2.0364278-001
2	SM2 =	6.1285108-001	2.0364278-001
3	SM2 =	5.8761348-001	1.7848963-001
4	SM2 =	5.8761348-001	1.7848963-001
5	SM2 =	5.8764433-001	1.7915424-001
6	SM2 =	5.8764433-001	1.7915424-001
7	SM2 =	5.8764423-001	1.7915737-001
8	SM2 =	5.8764423-001	1.7915737-001
9	SM2 =	5.8764423-001	1.7915738-001
10	SM2 =	5.8764423-001	1.7915738-001

```
 1     SM3 = -1.3003431-003  1.1650979-002
 2     SM3 = -1.6422078-002 -4.8935799-003
 3     SM3 = -1.7144656-002 -6.3524698-003
 4     SM3 = -1.7171689-002 -6.3680704-003
 5     SM3 = -1.7172342-002 -6.3681092-003
 6     SM3 = -1.7172345-002 -6.3681082-003
 7     SM3 = -1.7172345-002 -6.3681082-003
 8     SM3 = -1.7172345-002 -6.3681082-003
 9     SM3 = -1.7172345-002 -6.3681082-003
10     SM3 = -1.7172345-002 -6.3681082-003
```

ACOUSTIC PRESSURE CONTRIBUTIONS DUE TO SPHEROID 2 (FOR CONVERGENCE CHECK)

```
 1     SM1 = -7.0747621-001 -1.0404503-001
 2     SM1 = -7.0747621-001 -1.0404503-001
 3     SM1 = -6.9463353-001 -1.0215633-001
 4     SM1 = -6.9463353-001 -1.0215633-001
 5     SM1 = -6.9460877-001 -1.0215268-001
 6     SM1 = -6.9460877-001 -1.0215268-001
 7     SM1 = -6.9460875-001 -1.0215268-001
 8     SM1 = -6.9460875-001 -1.0215268-001
 9     SM1 = -6.9460875-001 -1.0215268-001
10     SM1 = -6.9460875-001 -1.0215268-001


 1     SM2 =  1.1187352-001  1.4597320-001
 2     SM2 =  1.1782502-001  1.4311626-001
 3     SM2 =  5.8118081-002  2.3062603-001
 4     SM2 =  8.2730662-002  1.7007108-001
 5     SM2 =  8.2818935-002  1.7993574-001
 6     SM2 =  8.2332969-002  1.7925083-001
 7     SM2 =  8.2380546-002  1.7925947-001
 8     SM2 =  8.2379106-002  1.7926094-001
 9     SM2 =  8.2379078-002  1.7926088-001
10     SM2 =  8.2379079-002  1.7926088-001


 1     SM3 = -1.4176702-001  6.0663438-002
 2     SM3 = -1.4176702-001  6.0663438-002
 3     SM3 = -1.5269497-001  5.7689831-002
 4     SM3 = -1.5269497-001  5.7689831-002
 5     SM3 = -1.5272265-001  5.7687277-002
 6     SM3 = -1.5272265-001  5.7687277-002
 7     SM3 = -1.5272268-001  5.7687280-002
 8     SM3 = -1.5272268-001  5.7687280-002
 9     SM3 = -1.5272268-001  5.7687280-002
10     SM3 = -1.5272268-001  5.7687280-002
```


UNCLASSIFIED

FAR FIELD ANGLE = 90.00 DEGREES

ACOUSTIC PRESSURE CONTRIBUTIONS DUE TO SPHEROID 1 (FOR CONVERGENCE CHECK)

1	SM1 =	0.0000000+000	2.0207722-001
2	SM1 =	0.0000000+000	2.0207722-001
3	SM1 =	0.0000000+000	1.6566730-001
4	SM1 =	0.0000000+000	1.6566730-001
5	SM1 =	0.0000000+000	1.6538560-001
6	SM1 =	0.0000000+000	1.6538560-001
7	SM1 =	0.0000000+000	1.6538670-001
8	SM1 =	0.0000000+000	1.6538670-001
9	SM1 =	0.0000000+000	1.6538670-001
10	SM1 =	0.0000000+000	1.6538670-001

1	SM2 =	2.9228819-001	1.2292213-001
2	SM2 =	2.9228819-001	1.2292213-001
3	SM2 =	3.0633277-001	1.3694430-001
4	SM2 =	3.0633277-001	1.3694430-001
5	SM2 =	3.0633230-001	1.3722096-001
6	SM2 =	3.0633230-001	1.3722096-001
7	SM2 =	3.0633236-001	1.3722000-001
8	SM2 =	3.0633236-001	1.3722000-001
9	SM2 =	3.0633236-001	1.3722000-001
10	SM2 =	3.0633236-001	1.3722000-001

1	SM3 =	8.0329421-003	-7.0261192-004
2	SM3 =	9.2160926-003	2.3492785-003
3	SM3 =	9.6193383-003	1.7729575-003
4	SM3 =	9.6237970-003	1.7739774-003
5	SM3 =	9.6237626-003	1.7742053-003
6	SM3 =	9.6237626-003	1.7742050-003
7	SM3 =	9.6237626-003	1.7742050-003
8	SM3 =	9.6237626-003	1.7742050-003
9	SM3 =	9.6237626-003	1.7742050-003
10	SM3 =	9.6237626-003	1.7742050-003

ACOUSTIC PRESSURE CONTRIBUTIONS DUE TO SPHEROID 2 (FOR CONVERGENCE CHECK)

1	SM1 =	-6.3894038-021	6.0426467-001
2	SM1 =	-6.3894038-021	6.0426467-001
3	SM1 =	-6.4598003-021	6.1092227-001
4	SM1 =	-6.4598003-021	6.1092227-001
5	SM1 =	-6.4597004-021	6.1091282-001
6	SM1 =	-6.4597004-021	6.1091282-001
7	SM1 =	-6.4597005-021	6.1091283-001
8	SM1 =	-6.4597005-021	6.1091283-001
9	SM1 =	-6.4597005-021	6.1091283-001
10	SM1 =	-6.4597005-021	6.1091283-001

```
 1     SM2 =  4.9231484-002  7.5699205-002
 2     SM2 =  4.9265361-002  7.5832698-002
 3     SM2 =  8.2563771-002  2.7120565-002
 4     SM2 =  8.3960162-002  2.7014302-002
 5     SM2 =  8.3799724-002  3.1113275-002
 6     SM2 =  8.3787345-002  3.1099310-002
 7     SM2 =  8.3772806-002  3.1096276-002
 8     SM2 =  8.3772850-002  3.1096295-002
 9     SM2 =  8.3772844-002  3.1096280-002
10     SM2 =  8.3772844-002  3.1096280-002


 1     SM3 =  6.0126269-002  8.7374510-002
 2     SM3 =  6.0126269-002  8.7374510-002
 3     SM3 =  5.8988252-002  8.2615614-002
 4     SM3 =  5.8988252-002  8.2615614-002
 5     SM3 =  5.8986478-002  8.2625245-002
 6     SM3 =  5.8986478-002  8.2625245-002
 7     SM3 =  5.8986478-002  8.2625240-002
 8     SM3 =  5.8986478-002  8.2625240-002
 9     SM3 =  5.8986478-002  8.2625240-002
10     SM3 =  5.8986478-002  8.2625240-002
```

NORMALIZED FARFIELD PRESSURE

ANGLE (DEGREES)	REAL PART	IMAGINARY PART	MAGNITUDE	PHASE ANGLE (DEGREES)
0.00	1.00000+000	0.00000+000	1.00000+000	0.000
90.00	-3.40159-001	-3.68704-001	5.01648-001	-132.694

INPUT NEARFIELD POINT

CARTESIAN COORDINATES	SPHEROIDAL COORDINATES
X/(D1/2) = 0.00000+000	XI = 2.00000+000
Y/(D1/2) = 0.00000+000	ETA = 1.00000+000
Z/(D1/2) = 2.00000+000	PHI = 0.000 DEGREES

ACOUSTIC PRESSURE CONTRIBUTIONS DUE TO SPHEROID 1 (FOR CONVERGENCE CHECK)

```
 1     SM1 = -8.4284239-002 -3.8620603-002
 2     SM1 = -8.4284239-002 -3.8620603-002
 3     SM1 = -1.0017512-001 -3.4744515-002
 4     SM1 = -1.0017512-001 -3.4744515-002
 5     SM1 = -1.0027612-001 -3.4981660-002
 6     SM1 = -1.0027612-001 -3.4981660-002
 7     SM1 = -1.0027604-001 -3.4978030-002
 8     SM1 = -1.0027604-001 -3.4978030-002
 9     SM1 = -1.0027604-001 -3.4978098-002
10     SM1 = -1.0027604-001 -3.4978098-002
```


UNCLASSIFIED

```
 1      SM2 = -1.1137050-001  9.9311624-002
 2      SM2 = -1.1137050-001  9.9311624-002
 3      SM2 = -1.0606027-001  9.2122994-002
 4      SM2 = -1.0606027-001  9.2122994-002
 5      SM2 = -1.0596758-001  9.2338863-002
 6      SM2 = -1.0596758-001  9.2338863-002
 7      SM2 = -1.0596767-001  9.2335576-002
 8      SM2 = -1.0596767-001  9.2335576-002
 9      SM2 = -1.0596767-001  9.2335639-002
10      SM2 = -1.0596767-001  9.2335639-002


 1      SM3 = -9.7614303-004  1.5427348-003
 2      SM3 = -5.9509139-004  6.4362580-003
 3      SM3 = -7.7920986-004  6.2485125-003
 4      SM3 = -7.7018480-004  6.2442046-003
 5      SM3 = -7.7009907-004  6.2443614-003
 6      SM3 = -7.7010066-004  6.2443633-003
 7      SM3 = -7.7010068-004  6.2443633-003
 8      SM3 = -7.7010068-004  6.2443633-003
 9      SM3 = -7.7010068-004  6.2443633-003
10      SM3 = -7.7010068-004  6.2443633-003

 SPHEROID 1 CONTRIBUTION TO THE ACOUSTIC PRESSURE (NTS/SQUARE METER) AT THE INPUT NEARFIELD POINT

                    REAL PART    IMAGINARY PART            MAGNITUDE      PHASE ANGLE (DEGREES)

                   5.48296+004  -1.13328+006              1.13460+006        -87.230

                  ACOUSTIC PRESSURE CONTRIBUTIONS DUE TO SPHEROID 2 (FOR CONVERGENCE CHECK)

 1      SM1 = -1.4128212-001 -1.0350546-001
 2      SM1 = -1.4128212-001 -1.0350546-001
 3      SM1 = -1.3779784-001 -1.0379287-001
 4      SM1 = -1.3779784-001 -1.0379287-001
 5      SM1 = -1.3780105-001 -1.0380220-001
 6      SM1 = -1.3780105-001 -1.0380220-001
 7      SM1 = -1.3780105-001 -1.0380216-001
 8      SM1 = -1.3780105-001 -1.0380216-001
 9      SM1 = -1.3780105-001 -1.0380216-001
10      SM1 = -1.3780105-001 -1.0380216-001


 1      SM2 = -4.2124395-002  5.5993395-003
 2      SM2 = -4.2045361-002  7.1441089-003
 3      SM2 = -6.5957435-002 -4.4435184-003
 4      SM2 = -4.9772379-002 -1.3216066-002
 5      SM2 = -4.8518657-002 -9.9661024-003
 6      SM2 = -4.8851574-002 -9.6624089-003
 7      SM2 = -4.8901873-002 -9.6684562-003
 8      SM2 = -4.8905808-002 -9.6720808-003
 9      SM2 = -4.8906020-002 -9.6725845-003
10      SM2 = -4.8906029-002 -9.6726405-003
```

```
 1      SM3 = -2.3984592-002 -1.1680643-003
 2      SM3 = -2.3984592-002 -1.1680643-003
 3      SM3 = -2.5811565-002 -1.0764698-003
 4      SM3 = -2.5811565-002 -1.0764698-003
 5      SM3 = -2.5808142-002 -1.0696896-003
 6      SM3 = -2.5808142-002 -1.0696896-003
 7      SM3 = -2.5808144-002 -1.0697197-003
 8      SM3 = -2.5808144-002 -1.0697197-003
 9      SM3 = -2.5808144-002 -1.0697195-003
10      SM3 = -2.5808144-002 -1.0697195-003

 SPHEROID 2 CONTRIBUTION TO THE ACOUSTIC PRESSURE (NTS/SQUARE METER) AT THE INPUT NEARFIELD POINT

                    REAL PART    IMAGINARY PART          MAGNITUDE      PHASE ANGLE (DEGREES)

                  -1.05800+005  -1.56066+005           1.88548+005        -124.134

TOTAL ACOUSTIC PRESSURE (NTS/SQUARE METER) AT THE INPUT NEARFIELD POINT ( 0.00000+000, 0.00000+000, 2.00000+000)

                    REAL PART    IMAGINARY PART          MAGNITUDE      PHASE ANGLE (DEGREES)

                  -5.09703+004  -1.28934+006           1.29035+006         -92.264
```


UNCLASSIFIED