SHIP
(Simplified-Helmholtz-Integral Program)

A Fast Computer Program for Calculating the Acoustic Radiation and Radiation Impedance for Free-Flooded-Ring and Finite-Circular-Cylinder Sources

PETER H. ROGERS

Transducer Branch
Acoustics Division

FR-7240

June 19, 1972

0745295

AD

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited. DDC-A

CONTENTS

Abstract	ii
Problem Status	ii
Authorization	ii
INTRODUCTION	1
THEORY	1
EVALUATION OF MATRIX ELEMENTS	7
EVALUATION OF THE SURFACE PRESSURE	9
FAR-FIELD AND IMPEDANCE CALCULATIONS	12
PROGRAM STRUCTURE	12
PROGRAM INPUT AND OUTPUT	14
Input	14
Output	16
ACCURACY AND SPEED	17
EXAMPLES	20
Magnetostrictive Ring	20
Circular Piston in a Finite Rigid Baffle	20
Circular Piston in an Infinite Rigid Baffle	23
Squirter	23
Band on a Prolate Spheroid	24
CHARACTERISTIC FREQUENCIES	26
REFERENCES	28
APPENDIX A — Program Listing	29

ABSTRACT

The Naval Research Laboratory SHIP is a special-purpose acoustic radiation program designed for rapidly determining acoustic surface pressures, radiation impedances, and far-field radiation patterns for acoustic sources in the form of finite cylinders and free-flooded rings and for acoustic sources which can be approximated by such forms. The program uses a finite-element method based on a one-dimensional version of the surface Helmholtz integral equation. Axial symmetry is assumed in all cases. It is the complete use of this axial symmetry which results in the high speed of this method relative to other formulations of this problem. The basic inputs to the program are the geometry of the ring or cylinder and the normal velocity on the surface of this source. The output is a representation of the complex acoustic pressure distribution on the surface of the source, the radiation impedance, and (if desired) the far-field radiation pattern. The program is presently limited to a single ring or single cylinder, but it may easily be modified to handle coaxial stacks of rings and cylinders.

PROBLEM STATUS

This is an interim report on the NRL Problem; work is continuing.

AUTHORIZATION

NRL Problem S02-19
Project RF 05-121-402-6200

Manuscript submitted March 2, 1972.

SHIP
(SIMPLIFIED-HELMHOLTZ-INTEGRAL PROGRAM)

A FAST COMPUTER PROGRAM FOR CALCULATING THE ACOUSTIC RADIATION AND RADIATION IMPEDANCE FOR FREE-FLOODED-RING AND FINITE-CIRCULAR-CYLINDER SOURCES

INTRODUCTION

The Simplified-Helmholtz-Integral Program (SHIP) is a special-purpose acoustic-radiation computer program designed for rapidly determining acoustic surface pressures, radiation impedances, and far-field radiation patterns for acoustic sources in the form of finite circular cylinders and free-flooded rings. The program uses a finite-element method based on a one-dimensional version of the surface Helmholtz integral equation. In all cases axial symmetry is assumed and it is the complete exploitation of this assumption which results in the high speed of this method relative to others. The basic inputs to SHIP are the geometric parameters of the ring or cylinder source and the normal velocity on the surface of the source. The output is a representation of the complex acoustic pressure distribution on the surface of the source, the radiation impedance, and (if desired) the far-field radiation pattern. SHIP can also handle the case in which a part of the surface of the transducer has zero pressure (pressure-release boundary condition) instead of a prescribed normal velocity. This would be the case for a ring operating in the so-called squirter mode (pressure-released on the outside surface so as to radiate primarily axially instead of radially). SHIP is limited to a single ring or cylinder, but an extended version of this program, the Multiple-Ring Simplified-Helmholtz-Integral Program (MR SHIP), capable of handling coaxial stacks of rings or cylinders has been completed and will be reported in another NRL Report.

THEORY

The theory used in the program begins with the Helmholtz integral equation for the velocity potential at a point $\mathbf{R}$, due to a surface S_0 (described by the position vector $\mathbf{R}_0$) which is vibrating harmonically with angular frequency ω:

$$\phi(\mathbf{R}) = \oiint \phi(\mathbf{R}_0) \frac{\partial g(\mathbf{R}_0|\mathbf{R})}{\partial n_0} \, dS_0 - \oiint \frac{\partial \phi(\mathbf{R}_0)}{\partial n_0} \, g(\mathbf{R}_0|\mathbf{R}) \, dS_0 \, , \tag{1}$$

where $\oiint dS_0$ represents integration over the surface of the radiator, $\partial/\partial n_0$ denotes a derivative normal to the surface, in particular $-\partial\phi/\partial n_0$ being the known normal velocity of the radiating surface, and $g(\mathbf{R}_0|\mathbf{R})$ is the free-space Green's function given by

$$g(\mathbf{R}_0|\mathbf{R}) = \frac{e^{-ik|\mathbf{R}-\mathbf{R}_0|}}{4\pi|\mathbf{R}-\mathbf{R}_0|} \, , \tag{2}$$

in which k is the wavenumber (ω/c). The acoustic field everywhere can be obtained from Eq. (1) once $\phi(\mathbf{R}_0)$, the velocity potential at the surface, is determined.

The limit of Eq. (1) as $\mathbf{R}$ approaches the surface of the radiator (allowing for the jump discontinuity in the first integral [1] is [2,3] the surface Helmholtz integral equation:

$$\phi(\mathbf{R}_s)/2 = \oiint \phi(\mathbf{R}_0) \frac{\partial g(\mathbf{R}_s|\mathbf{R}_0)}{\partial n_0} dS_0 - \oiint \frac{\partial \phi(\mathbf{R}_0)}{\partial n_0} g(\mathbf{R}_0|\mathbf{R}_s) dS_0 \tag{3}$$

The usual procedure for obtaining approximate solutions for Eq. (3) involves dividing the surface of the source into N subdivisions over which the velocity potential (and the velocity) may be considered constant. Equation (3) can then be reduced to a system of N complex linear equations in N unknowns, the N solutions to these equations representing some sort of mean velocity potential over each of the N subdivisions. If, for a cylindrical radiator, the surface were divided as shown in Fig. 1 and no advantage were taken of the symmetry, 7200 matrix elements would have to be calculated, each matrix element consisting of a surface integral which would have to be obtained numerically. In general, for M sliced segments and N bands $2(MN)^2$ surface integrals would have to be evaluated. If the bands are chosen to be equally spaced and if the symmetry of the problem is fully used, this number can be greatly reduced. For axially symmetric velocity distributions it is possible to reformulate the problem so as to eliminate the vertically sliced subdivisions entirely (with a consequent reduction in the minimum number of matrix elements to be calculated) and to convert the numerical integrations from surface to line integrals. The resulting method is much faster and (for a given number of bands) more accurate than the conventional method.

For simplicity the derivation of this method will be given for a solid cylinder of radius a and height $2h$. The extension of the method to ring transducers is straightforward. Cylindrical coordinates will be used with the z axis coinciding with the axis of symmetry of the cylinder and with the origin at the center of the cylinder. For the cylinder Eq. (3) becomes

$$\begin{aligned}
\phi(r_s, z)/2 = & \int_0^{2\pi}\int_0^a g(r_0, h, \theta_0|r_s, z_s)\, v(r_0, h)\, r_0\, dr_0 d\theta_0 \\
& + \int_0^{2\pi}\int_0^a g(r_0, -h, \theta_0|r_s, z_s)\, v(r_0, -h)\, r_0\, dr_0 d\theta_0 \\
& + \int_0^{2\pi}\int_{-h}^h g(a, z_0, \theta_0|r_s, z_s)\, v(a, z_0)\, a\, dz_0 d\theta_0 \\
& + \int_0^{2\pi}\int_0^a \left[\frac{\partial g(r_0, z_0, \theta_0|r_s, z_s)}{\partial z_0}\right]_{z_0=h} \phi(r_0, h)\, r_0\, dr_0 d\theta_0 \\
& - \int_0^{2\pi}\int_0^a \left[\frac{\partial g(r_0, z_0, \theta_0|r_s, z_s)}{\partial z_0}\right]_{z_0=-h} \phi(r_0, -h)\, r_0\, dr_0 d\theta_0 \\
& + \int_0^{2\pi}\int_{-h}^h \left[\frac{\partial g(r_0, z_0, \theta_0|r_s, z_s)}{\partial r_0}\right]_{r_0=a} \phi(a, z_0)\, a\, dz_0 d\theta_0 ,
\end{aligned} \tag{4}$$

where $v(r_0, z_0)$ is the normal velocity at the surface point (r_0, z_0).

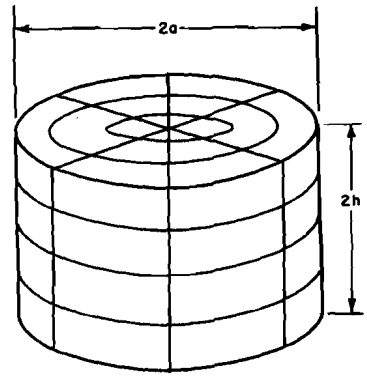


Fig. 1 - Subdivision of the surface of a cylinder into six sliced segments and ten bands (three bands on the bottom surface being hidden)

The Green's function is then replaced by its integral representation [4]

$$g(r_0, z_0, \theta_0 | r, z) = (1/4\pi) \sum_{m=0}^{\infty} \epsilon_m \cos m(\theta_0)$$

$$\times \int_0^{\infty} J_m(\ell r)\, J_m(\ell r_0)\, \frac{\ell e^{-(\ell^2-k^2)^{1/2}|z-z_0|}}{(\ell^2-k^2)^{1/2}}\, d\ell\,, \tag{5}$$

where ℓ is an integration variable and $\epsilon_m = 2$ for $m \neq 0$ and $\epsilon_0 = 1$. This integral representation enables exact integration over θ_0. Equation (4) then reduces to the one-dimensional integral equation

$$\begin{aligned}
\phi(r_s, z_s) = &\int_0^a \int_0^{\infty} J_0(\ell r_s)\, J_0(\ell r_0)\, \mu^{-1} \ell\, e^{-\mu|z_s-h|}\, v(r_0, h)\, r_0\, d\ell dr_0 \\
&+ \int_0^a \int_0^{\infty} J_0(\ell r_s)\, J_0(\ell r_0)\, \mu^{-1} \ell\, e^{-\mu|z_s+h|}\, v(r_0, -h)\, r_0\, d\ell dr_0 \\
&+ \int_{-h}^{h} \int_0^{\infty} J_0(\ell r_s)\, J_0(\ell a)\, \mu^{-1} \ell\, e^{-\mu|z_s-z_0|}\, v(a, z_0)\, a\, d\ell dz_0 \\
&+ \int_0^a \int_0^{\infty} J_0(\ell r_s)\, J_0(\ell r_0)\, \ell\, e^{-\mu|z_s-h|}\, \phi(r_0, h)\, r_0\, d\ell dr_0 \\
&+ \int_0^a \int_0^{\infty} J_0(\ell r_s)\, J_0(\ell r_0)\, \ell\, e^{-\mu|z_s+h|}\, \phi(r_0, -h)\, r_0\, d\ell dr_0 \\
&- \int_{-h}^{h} \int_0^{\infty} J_0(\ell r_s)\, J_0(\ell r_0)\, \mu^{-1} \ell^2\, e^{-\mu|z_s-z_0|}\, \phi(a, z_0)\, a\, d\ell dz_0\,,
\end{aligned} \tag{6}$$

where $\mu = (\ell^2 - k^2)^{1/2}$.

The surface of the cylinder is divided into IMAX bands on the top and bottom and JMAX bands on the side. The bands on the top and bottom surfaces are thus $D_t = a/\mathrm{IMAX}$ wide, and the bands on the side are $D_s = 2h/\mathrm{JMAX}$ wide. The centers of the top (and bottom) bands have radial coordinates

$$r_i = (i - 0.5)\, D_t \ , \qquad i = 1, \ldots, \mathrm{IMAX} \ , \tag{7}$$

and the centers of the side bands have z coordinates

$$z_j = h - (j - 0.5)\, D_s \ , \qquad j = 1, \ldots, \mathrm{JMAX} \ . \tag{8}$$

It is assumed that the bands have been chosen to be narrow enough for both the velocity and velocity potential to be considered constant over each band. With this assumption the remaining integrations over r_0 and z_0 can be performed analytically by using the relations

$$\int_a^b J_0(\ell r_0)\, r_0\, dr_0 = J_1(\ell a)\, a/\ell - J_1(\ell b)\, b/\ell \tag{9}$$

and

$$\begin{aligned} \int_a^b e^{-\mu|z-z_0|}\, dz_0 &= -\mu^{-1}\left[e^{-\mu(z-b)} - e^{-\mu(z-a)}\right] , & z > b > a \ , \\ &= -\mu^{-1}\left[2 - e^{-\mu(b-z)} - e^{-\mu(z-a)}\right] , & b > z > a \ , \\ &= -\mu^{-1}\left[e^{-\mu(a-z)} - e^{-\mu(b-z)}\right] , & b > a > z \ . \end{aligned} \tag{10}$$

Use of Eqs. (9) and (10) yields the following system of LCMAX (= 2IMAX + JMAX) equations in LCMAX unknowns:

$$\begin{aligned} \phi(r_i, h) = &\sum_{j=1}^{\mathrm{IMAX}} \left[G_{ij}^{tt}\, v(r_j, h) + G_{ij}^{tb}\, v(r_j, -h) + M_{ij}^{tb}\, \phi(r_j, -h)\right] \\ &+ \sum_{j=1}^{\mathrm{JMAX}} \left[G_{ij}^{ts}\, v(a, z_j) + M_{ij}^{ts}\, \phi(a, z_j)\right] , \qquad i = 1, \ldots, \mathrm{IMAX}, \end{aligned} \tag{11a}$$

$$\begin{aligned} \phi(r_i, -h) = &\sum_{j=1}^{\mathrm{IMAX}} \left[G_{ij}^{bb}\, v(r_j, -h) + G_{ij}^{bt}\, v(r_j, h) + M_{ij}^{bt}\, \phi(r_j, h)\right] \\ &+ \sum_{j=1}^{\mathrm{JMAX}} \left[G_{ij}^{bs}\, v(a, z_j) + M_{ij}^{bs}\, \phi(a, z_j)\right] , \qquad i = 1, \ldots, \mathrm{IMAX}, \end{aligned} \tag{11b}$$

$$\begin{aligned} \phi(a, z_i) = &\sum_{j=1}^{\mathrm{IMAX}} \left[G_{ij}^{st}\, v(r_j, h) + G_{ij}^{sb}\, v(r_j, -h) + M_{ij}^{st}\, \phi(r_j, h) + M_{ij}^{sb}\, \phi(r_j, -h)\right] \\ &+ \sum_{j=1}^{\mathrm{JMAX}} \left[G_{ij}^{ss}\, v(a, z_j) + M_{ij}^{ss}\, \phi(a, z_i)\right] , \qquad i = 1, \ldots, \mathrm{JMAX}, \end{aligned} \tag{11c}$$

where M and G are the influence matrices to be discussed in the next paragraph.

The superscripts t, b, and s designate the top, bottom, and side surfaces respectively; as examples, the symbol G_{ij}^{st} represents the contribution to the velocity potential on the ith band of the side surface due to the velocity on the jth band of the top surface and M_{ij}^{bs} would represent the contribution to the velocity potential on the ith band of the bottom surface due to the velocity potential on the jth band of the side surface. It is easy to show that all of these matrix elements are not independent; in particular

$$G_{ij}^{tt} = G_{ij}^{bb} \quad \text{and} \quad M_{ij}^{tt} = M_{ij}^{bb} = 0 , \qquad i, j = 1, \ldots, \text{IMAX}, \tag{12a}$$

$$G_{ij}^{tb} = G_{ij}^{bt} \quad \text{and} \quad M_{ij}^{tb} = M_{ij}^{bt} , \qquad i, j = 1, \ldots, \text{IMAX}, \tag{12b}$$

$$G_{ij}^{ts} = G_{i,\text{JMAX}+1-j}^{bs} \quad \text{and} \quad M_{ij}^{ts} = M_{i,\text{JMAX}+1-j}^{bs}, \qquad i = 1, \ldots, \text{IMAX}, \quad j = 1, \ldots, \text{JMAX}, \tag{12c}$$

$$G_{ij}^{st} = G_{\text{JMAX}+1-i,j}^{sb} \quad \text{and} \quad M_{ij}^{st} = M_{\text{JMAX}+1-i,j}^{sb}, \qquad i = 1, \ldots, \text{JMAX}, \quad j = 1, \ldots, \text{IMAX}, \tag{12d}$$

$$G_{ij}^{ss} = G_{ji}^{ss} \quad \text{and} \quad M_{ij}^{ss} = M_{ji}^{ss} , \qquad i, j = 1, \ldots, \text{JMAX}, \tag{12e}$$

and for all k and m such that $|i - j| = |k - m|$,

$$G_{ij}^{ss} = G_{km}^{ss} \quad \text{and} \quad M_{ij}^{ss} = M_{km}^{ss} , \qquad i, j = 1, \ldots, \text{JMAX}. \tag{12f}$$

The M and G matrices are all integrals from zero to infinity over the single variable ℓ. The explicit forms for these matrices as found from Eqs. (6), (9), and (10) are

$$M_{ij}^{ss} = -a \int_0^\infty J_0(\ell a)\, J_1(\ell a)\, \ell^2 \mu^{-2} \left[e^{-\mu |z_i - z_j - D_s/2|} - e^{-\mu |z_i - z_j + D_s/2|} \right] d\ell ,$$

$$i = 1, \ldots, \text{JMAX}, \quad j > i , \quad (\text{in particular, } i \neq j) \tag{13a}$$

$$M_{ii}^{ss} = -2a \int_0^\infty J_0(\ell a)\, J_1(\ell a)\, \ell^2 \mu^{-2} \left[e^{-\mu D_s/2} - 1 \right] d\ell , \qquad i = 1, \ldots, \text{JMAX}, \tag{13b}$$

$$M_{ij}^{ts} = -a \int_0^\infty J_0(\ell r_i)\, J_1(\ell a)\, \ell^2 \mu^{-2} \left[e^{-\mu(h - z_j - D_s/2)} - e^{-\mu(h - z_j + D_s/2)} \right] d\ell ,$$

$$i = 1, \ldots, \text{IMAX}, \quad j = 1, \ldots, \text{JMAX}, \tag{13c}$$

$$M_{ij}^{tb} = \int_0^\infty J_0(\ell r_i)\, e^{-2\mu h} \left[r_j^L J_1\left(\ell r_j^L\right) - r_j^U J_1\left(\ell r_j^U\right) \right] d\ell ,$$

$$i, j = 1, \ldots, \text{IMAX}, \quad r_i^L = r_j - D_t/2 , \quad r_j^U = r_j + D_t/2 , \tag{13d}$$

$$M_{ij}^{st} = -\int_0^\infty J_0(\ell a)\, e^{-\mu(h-z_i)}\left[r_j^U J_1\left(\ell r_j^U\right) - r_j^L J_1\left(\ell r_j^L\right)\right] d\ell \,,$$

$$i = 1, \ldots, \text{JMAX}, \quad j = 1, \ldots, \text{IMAX}, \tag{13e}$$

$$G_{ij}^{ss} = a\int_0^\infty [J_0(\ell a)]^2 \ell\mu^{-2}\left[e^{-\mu|z_i-z_j-D_s/2|} - e^{-\mu|z_i-z_j+D_s/2|}\right] d\ell \,,$$

$$i = 1, \ldots, \text{JMAX}, \quad j = 1, \ldots, i-1 \quad (\text{in particular, } i \neq j) \tag{14a}$$

$$G_{ii}^{ss} = 2a\int_0^\infty [J_0(\ell a)]^2 \ell\mu^{-2}\left[e^{-\mu D_s/2} - 1\right] d\ell \,, \quad i = 1, \ldots, \text{IMAX}, \tag{14b}$$

$$G_{ij}^{ts} = a\int_0^\infty J_0(\ell r_i)\, J_0(\ell a)\, \ell\mu^{-2}\left[e^{-\mu(h-z_j-D_s/2)} - e^{-\mu(h-z_j+D_s/2)}\right] d\ell \,,$$

$$i = 1, \ldots, \text{IMAX}, \quad j = 1, \ldots, \text{JMAX}, \tag{14c}$$

$$G_{ij}^{tt} = \int_0^\infty J_0(\ell r_i)\, \mu^{-1}\left[r_j^U J_1\left(\ell r_j^U\right) - r_j^L J_1\left(\ell r_j^L\right)\right] d\ell \,, \quad i, j = 1, \ldots, \text{IMAX}, \tag{14d}$$

$$G_{ij}^{tb} = \int_0^\infty J_0(\ell r_i)\, \mu^{-1} e^{-2\mu h}\left[r_j^U J_1\left(\ell r_j^U\right) - r_j^U J_1\left(\ell r_j^U\right)\right] d\ell \,, \quad i, j = 1, \ldots, \text{IMAX}, \tag{14e}$$

$$G_{ij}^{st} = \int_0^\infty J_0(\ell a)\, \mu^{-1} e^{-\mu(h-z_i)}\left[r_j^U J_1\left(\ell r_j^U\right) - r_j^L J_1\left(\ell r_j^L\right)\right] d\ell \,,$$

$$i = 1, \ldots, \text{JMAX}, \quad j = 1, \ldots, \text{IMAX}. \tag{14f}$$

The remaining M and G matrix elements can be determined from Eqs. (13) and (14) using the symmetry relationships expressed in Eq. (12). From Eq. (12) the matrix elements M_{ij}^{ss} and G_{ij}^{ss} need be calculated only for $i = 1$, $j = 1, \ldots, \text{JMAX}$.

For rings, the following additional matrix elements, homologous to those already derived, must be evaluated:

M_{ij}^{ii} and G_{ij}^{ii} relating inside surface points to inside surface bands

$(i, j = 1, \ldots, \text{JMAX})$,

M_{ij}^{is} and G_{ij}^{is} relating inside surface points to outside surface bands

$(i, j = 1, \ldots, \text{JMAX})$,

M_{ij}^{si} and G_{ij}^{si} relating outside surface points to inside surface bands

$(i, j = 1, \ldots, \text{JMAX})$,

M_{ij}^{it} and G_{ij}^{it} relating inside surface points to top surface bands

$$(i = 1, \ldots, \text{JMAX};\ j = 1, \ldots, \text{IMAX}),$$

M_{ij}^{ti} and G_{ij}^{ti} relating top surface points to inside surface bands

$$(i = 1, \ldots, \text{IMAX};\ j = 1, \ldots, \text{JMAX}).$$

EVALUATION OF MATRIX ELEMENTS

The quantity $\mu = (\ell^2 - k^2)^{1/2}$ in Eqs. (13) and (14) is pure imaginary for $\ell < k$. As a result the M and G matrix elements are complex quantities. The imaginary parts of the matrix elements are derived entirely from the integration from 0 to k, and contributions to the real parts of the matrix elements come from the entire range of integration. It is therefore convenient to divide the numerical integration into two parts, 0 to k and k to infinity, the latter integrals contributing to only the real part of the matrix elements. The numerical integrations are performed by Gaussian quadrature. A single ten-, 20-, or 32-point Gaussian quadrature is used for the integration from 0 to k. The integration from k to infinity is performed in steps of size FKFAST (first the integral from k to k + FKFAST is evaluated, then the integral from k + FKFAST to k + 2 FKFAST, and so forth) each step consisting of a single ten-, 20-, or 32-point Gaussian quadrature. After each step, a test for convergence is made. If the integral has converged, no further steps are taken; if it has not converged, the integral over the next step is evaluated. If the convergence test is not satisfied after ten steps, the integration is terminated.

There is an integrable singularity in μ^{-1} at $\ell = k$. This is eliminated by changing the integration variable from ℓ to μ. (At first glance the singularity appears to be of order μ^{-2}; this is not the case, however, because the trigonometric factors in the integrand are directly proportional to μ in the vicinity of $\ell = k$.)

To avoid repetitions of the same calculation, all of the Bessel functions used in the program are evaluated once and for all in subroutine CALBES and placed in COMMON for use by all of the integrating subroutines.

To further avoid redundant calculations, matrix elements which contain the same trigonometric functions are evaluated simultaneously whenever possible.

Only the independent matrix elements are evaluated.

All of the integrals in Eqs. (13) and (14) converge. For $\ell > k$, μ is real and positive, so that all integrals containing a factor $e^{-\mu\Gamma}$, where Γ is some real and positive quantity (such as $\Gamma = h - z_i$ in Eq. (14f)), converge rapidly. Several of the integrals (Eqs. (13b), (14b), (14d), and for some values of z_j (13c) and (14c)) contain terms which do not have an exponential factor. These integrals converge no faster than ℓ^{-2}, and special procedures had to be devised to achieve a satisfactory rate of convergence. These procedures are outlined in the discussion of the individual matrix elements which follows:

- The matrix elements M_{ij}^{ss}, M_{ij}^{is}, M_{ij}^{si}, M_{ij}^{ii}, G_{ij}^{ss}, G_{ij}^{is}, G_{ij}^{si}, and G_{ij}^{ii} are calculated simultaneously in subroutine CSSM for $i = 1$ and $j = 2$, JMAX. The matrix elements M_{ij}^{ss} etc. for $i = 2, \ldots,$ JMAX are evaluated from these using Eqs. (12e) and (12f).

- The matrix elements M_{ii}^{ss}, M_{ii}^{ii}, M_{ii}^{is}, M_{ii}^{si}, G_{ii}^{ss}, G_{ii}^{si}, G_{ii}^{is}, and G_{ii}^{ii} are calculated simultaneously in subroutine CSSM for $i = 1$. The remaining (i, i) elements are

determined from the (1, 1) elements using Eq. (12f). From Eq. (13b) the integral to be evaluated for the M_{11}^{ss} matrix element is

$$I = \int_0^\infty J_0(\ell a)\, J_1(\ell a)\, \ell^2 \mu^{-2}\left[e^{-\mu D_s/2} - 1\right] d\ell \,. \tag{15}$$

The part of the integral which contains the factor of unity converges only as fast as $(\sin \ell a)/\ell$ for large values of ℓ, and the other term may also converge slowly if D_s is small. The rate of convergence is improved by subtracting from I the integral I' given by

$$I' = \int_0^\infty J_0(\ell a)\, J_1(\ell a) \left[e^{-\ell D_s/2} - 1\right] d\ell \,. \tag{16}$$

The combination $I - I'$ converges as ℓ^{-3} instead of ℓ^{-1}. It is the integral $I - I'$ which is evaluated numerically using the technique outlined in the first paragraph of this section. I' by itself, however, can be evaluated exactly [5a]:

$$I' = D_s k_0 K(k_0)/(4\pi a^2) \,, \tag{17}$$

where $k_0^2 = 16a^2/(D_s^2 + 16a^2)$ and K is the complete elliptic integral of the first kind [6a]. $K(k_0)$ is evaluated by the FUNCTION subroutine ELLIPK. The matrix elements M_{11}^{ii}, M_{11}^{is}, and M_{11}^{si} are evaluated in a similar fashion. No convergence test is used for matrix elements in this section; all ten steps are always used.

A different method is used for the G_{11} matrix elements. The numerical integration is terminated after ten steps ($\mu \approx \ell = \mu_{max}$). The remainder of the integral ΔI (from μ_{max} to infinity) is approximated by integrating analytically an asymptotic expression for the integrand. For example, for G_{11}^{is}

$$\begin{aligned}
\Delta I &= 2r_{out} \int_{\mu_{max}}^\infty J_0(\ell r_{out})\, J_0(\ell r_{in})\, \mu^{-1}\left[e^{-\mu D_s/2} - 1\right] d\mu \\
&\approx -2r_{out} \int_{\mu_{max}}^\infty J_0(\ell r_{out})\, J_0(\ell r_{in})\, \ell^{-1}\, d\ell \\
&\approx -(2/\pi)(r_{out}/r_{in})^{1/2} \int_{\mu_{max}}^\infty \{\cos[\ell(r_{out} - r_{in})] + \sin[\ell(r_{out} + r_{in})]\}\, \ell^{-2}\, d\ell \\
&\approx (2/\pi)(r_{out}/r_{in})^{1/2} \Big(\{\cos[\mu_{max}(r_{out} - r_{in})] + \sin[\mu_{max}(r_{out} + r_{in})]\}\, \mu_{max}^{-1} \\
&\quad + (r_{out} - r_{in})\, \mathrm{si}[\mu_{max}(r_{out} - r_{in})] - (r_{out} + r_{in})\, \mathrm{Ci}[\mu_{max}(r_{in} + r_{out})]\Big) \,,
\end{aligned} \tag{18}$$

where r_{in} is the inner radius of the ring, r_{out} is the outer radius of the ring and Ci and si are the cosine and sine integrals [6b] respectively.

- The matrix elements M_{ij}^{tb}, G_{ij}^{tb}, and G_{ij}^{tt} are calculated simultaneously in subroutine CTBM for $i, j = 1, \ldots, \mathrm{IMAX}$. The integrals involved in the G^{tt} matrix elements (Eq. (14d)) are of the form

$$I = \int_0^\infty J_0(\ell x)\, J_1(\ell y)\, \mu^{-1}\, d\ell \,. \tag{19}$$

Since the integrand of I does not contain an exponential factor, a method similar to that used for the M_{11}^{ss} matrix element was used to obtain a satisfactory rate of convergence. In this case an integral $\mathcal{I}'$ given [5b] by

$$\mathcal{I}' = \int_0^\infty J_0(\ell x)\, J_1(\ell y)\, \ell^{-1}\, d\ell = (\pi y)^{-1}[(x+y)\, E(k_0) + (y-x)\, K(k_0)] \,, \tag{20}$$

where $k_0 = (4xy)^{1/2}/(x+y)$ is subtracted from I to yield a numerical integral which converges as ℓ^{-4}. The elliptic integral of the second kind $E(k_0)$ [6] is evaluated by the FUNCTION subroutine ELLIPE.

- The matrix elements M_{ij}^{st}, G_{ij}^{st}, M_{ij}^{it}, and G_{ij}^{it} are calculated simultaneously in subroutine CSTM for $i = 1$, JMAX and $j = 1, \ldots,$ IMAX. All integrals converge satisfactorily.

- The matrix elements M_{ij}^{ts}, G_{ij}^{ts}, M_{ij}^{ti}, and G_{ij}^{ti} are evaluated simultaneously in subroutine CTSM for $i = 1, \ldots,$ IMAX, $j = 2, \ldots,$ JMAX. For $j = 1$ the integrals involved contain terms which do not have an exponentially decaying factor. For the M_{i1}^{ts} and M_{i1}^{ti} matrix elements convergence is secured by a method similar to the method used for the M_{11}^{ss} matrix element, and for G_{i1}^{ts} and G_{i1}^{ti} a method was used similar to the one used for G_{11}^{is}.

EVALUATION OF THE SURFACE PRESSURE

For determining the acoustic pressure on the surface of the ring (or cylinder) a different system for numbering the bands is used. The total number of bands is designated LCMAX:

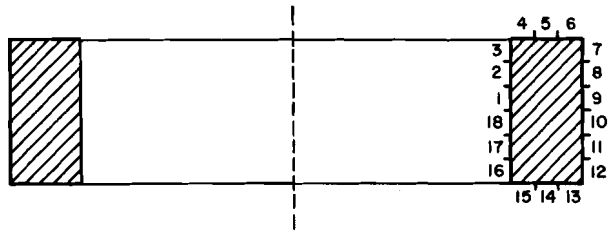
$$\text{LCMAX} = 2(\text{IMAX} + \text{JMAX}) \quad \text{for rings,} \tag{21a}$$

$$\text{LCMAX} = 2\ \text{IMAX} + \text{JMAX} \quad \text{for cylinders.} \tag{21b}$$

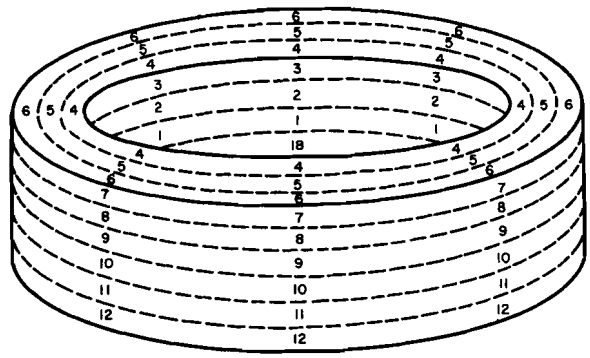
The bands are numbered as follows: For rings, band 1 is the band immediately above the median plane on the inside surface of the ring. Band 2 is above band 1 and so on to band JMAXH (= JMAX/2), which is the band nearest the top on the inside surface. The numbering of the bands continues across the top surface of the ring (from inside to outside), down the outside surface, across the bottom surface (from outside to inside), and then back up the inside surface. Band LCMAX is thus adjacent to band 1. Figure 2 illustrates the numbering of the bands for a ring when IMAX = 3 and JMAXH = 3 (LCMAX = 18).

For cylinders band 1 is at the center of the top surface. Band 2 is adjacent to band 1 and so on to band IMAX, which is on the top surface nearest to the edge. The numbering then continues down the outside surface and across the bottom (from the outside, in). Band LCMAX is thus the bottom center band. Figure 3 illustrates the numbering of the bands for a cylinder when IMAX = 3 and JMAXH = 3 (LCMAX = 12).

With the bands numbered in this way, the mirror image of band J with respect to the median plane is band LCMAX + 1 - J for all values of J (1 through LCMAX) for both rings and cylinders.



(a) Cross-sectional view

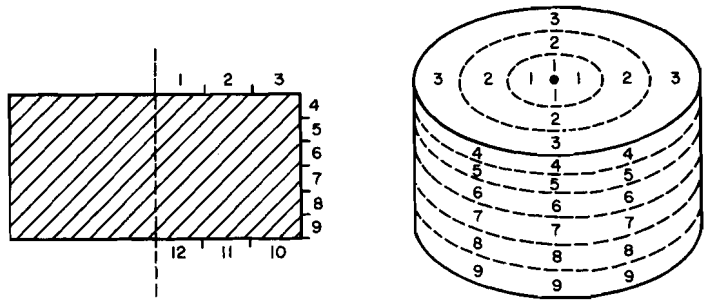


(b) Three-dimensional view

Fig. 2 - Numbering of the bands on a ring for IMAX = 3, JMAXH = 3

The pressure at the center of the Jth band is designated P(J), and VEL(J) is defined as the average normal velocity on the Jth band. Equations (11) and the equation $p = ic\rho_0 k\phi$ yield an equation of the form

$$\sum_{J=1}^{LCMAX} DM(I, J)P(J) = \sum_{J=1}^{LCMAX} GM(I, J)\,VEL(J)\ , \qquad I = 1, \ldots, LCMAX, \tag{22}$$



(a) Cross-sectional view

(b) Three-dimensional view

Fig. 3 - Numbering of the bands on a cylinder for IMAX = 3 and JMAXH = 3

relating the surface pressure and normal velocity. The DM and GM matrix elements are to be determined from the independent M and G matrix elements. Obtaining these matrix elements is a straightforward but tedious task involving the symmetry relations given by Eqs. (12) and the transformations necessary to convert from the numbering system used in the theory section to the numbering system just described. This procedure is carried out in subroutine ERING.

Since the ring is symmetric about the plane $z = 0$, whenever the normal velocity distribution is symmetric (or antisymmetric) with respect to this plane the resulting surface pressure distribution must also be symmetric (or antisymmetric). Any velocity distribution (VEL(J)) can be expressed as the sum of a symmetric ($V_S(J)$) and an antisymmetric ($V_A(J)$) velocity distribution, where

$$V_S(J) = [\mathrm{VEL}(J) + \mathrm{VEL}(\mathrm{LCMAX} + 1 - J)]/2 , \quad J = 1, \ldots, \mathrm{LCMAX}, \tag{23a}$$

$$V_A(J) = [\mathrm{VEL}(J) - \mathrm{VEL}(\mathrm{LCMAX} + 1 - J)]/2 , \quad J = 1, \ldots, \mathrm{LCMAX}, \tag{23b}$$

If we let LCMAXH = LCMAX/2 and let $P_S(J)$ and $P_A(J)$ be the symmetric and antisymmetric surface pressure distributions corresponding to the velocity distributions $V_S(J)$ and $V_A(J)$ respectively, then from Eq. (22) we obtain

$$\sum_{J=1}^{\mathrm{LCMAXH}} [\mathrm{DM}(I, J) + \mathrm{DM}(I, \mathrm{LCMAX} + 1 - J)]\, P_S(J)$$

$$= \sum_{J=1}^{\mathrm{LCMAXH}} [\mathrm{GM}(I, J) + \mathrm{GM}(I, \mathrm{LCMAX} + 1 - J)]\, V_S(J) , \quad I = 1, \ldots, \mathrm{LCMAXH} , \tag{24}$$

and

$$\sum_{J=1}^{\mathrm{LCMAXH}} [\mathrm{DM}(I, J) - \mathrm{DM}(I, \mathrm{LCMAX} + 1 - J)]\, P_A(J)$$

$$= \sum_{J=1}^{\mathrm{LCMAXH}} [\mathrm{GM}(I, J) - \mathrm{GM}(I, \mathrm{LCMAX} + 1 - J)]\, V_A(J) , \quad I = 1, \ldots, \mathrm{LXMAXH} . \tag{25}$$

Since the system is linear,

$$P(J) = P_S(J) + P_A(J) , \quad J = 1, \ldots, \mathrm{LCMAX} . \tag{26}$$

It is advantageous to solve Eqs. (24) and (25) for P_S and P_A rather than solving Eq. (22) for P directly, since it takes approximately 4 times longer to solve a single system of LCMAX equations in LCMAX unknowns than it takes to solve two systems of LCMAXH equations in LCMAXH unknowns. An additional factor of 2 in time may be saved by this method if (as it often turns out to be in practice) the initial velocity distribution is either symmetric or antisymmetric.

Equation (22) relates the surface pressure to the normal velocity. It makes no difference whether LCMAX pressures or LCMAX velocities or some combination of pressures and velocities totaling LCMAX are taken to be the independent parameters.

The SHIP program allows for a pressure release condition to be substituted for a prescribed velocity condition on any of the bands. When pressure release is assumed on band N, the Nth unknown becomes VEL(N) instead of P(N).

FAR-FIELD AND IMPEDANCE CALCULATIONS

The far-field pressures are obtained directly from Eq. (1) using asymptotic expressions for the Green's function. With the pressure and velocity taken to be constant over each band the integrations can be performed analytically, with the result that the far-field calculation involves only the summation of the contributions from each band and no numerical integration. If ICOR = 0, the distance from the source is taken to be infinite, and only the first term in the asymptotic expansion of the Green's function is used. If ICOR = 1, the distance from the source is taken to be DIST (an input parameter), and two terms are retained in the expansion of the Green's function. The calculations are performed in subroutine FARFLD.

The normalized radiation impedance is evaluated in subroutine SOLRING. The actual quantity which is evaluated is

$$Z = \int pv^* \, dS_0 \Big/ \left(\rho c A |v_{ref}|^2\right)$$

where the reference velocity v_{ref} is chosen to be 1 meter per second, A is the total surface area of the ring or cylinder, and ρc is 1.5×10^6 mks units. To obtain the normalized radiation impedance of the active part of the transducer, the result obtained from SHIP should be multiplied by the ratio of the total area to the active area. Since the pressure determined by SHIP is equal to the pressure at the center of the band and not the average pressure over the band, it may be desirable for the user of the program to calculate the impedance by fitting the surface pressure to a polynomial and integrating the polynomial analytically.

PROGRAM STRUCTURE

The SHIP program consists of a main program (SHIP) and 13 subroutines (Fig. 4). The program is written entirely in CDC 3600 Fortran. The following are the major computational steps, listed in the order of performance (with the subroutine in parentheses):

1. Input parameters are loaded (program SHIP) and Gaussian quadrature parameters are put in COMMON (GQC),

2. Band coordinates are calculated (ERING),

3. Bessel functions to be used in integrations are calculated and stored in COMMON (CALBES),

4. Independent matrix elements are evaluated (CSSM, CTBM, CSTM, and CTSM),

5. DM and GM matrix elements are determined from the independent matrix elements (ERING),

6. The symmetric and antisymmetric velocity distributions V_S and V_A are computed from the distribution of the velocity over the bands VEL(J) (SOLRING),

7. Simultaneous equations for $P_S(J)$ and $P_A(J)$ are set up (SOLRING),

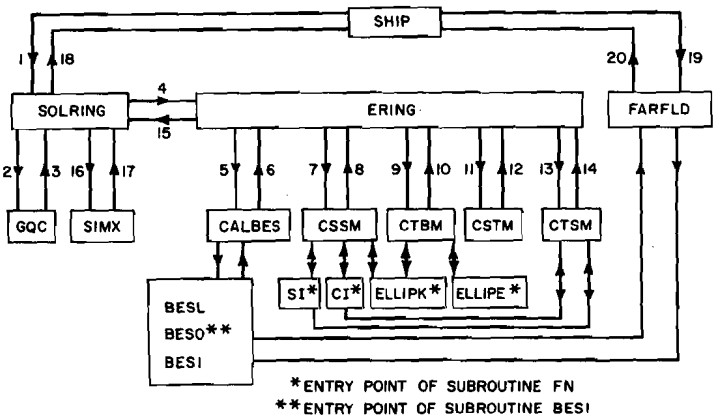


Fig. 4 - Structure of the SHIP program. The numbers refer to the order in which the calculations are performed. The main program is SHIP (Simplified-Helmholtz-Integral Program), and the 13 subroutines are SOLRING (solution of the ring), GQC (Gaussian-quadrature coefficients, ERING (evaluation of the ring's surface pressure), CALBES (calculate the Bessel functions), BESL (zeroth-order and first-order Bessel functions calculated simultaneously), CSSM (calculate side-side matrix), CTBM (calculate top-bottom matrix), CSTM (calculate side-top matrix), CTSM (calculate top-side matrix), FN (functions), FARFLD (far-field pressures), and BES1 (first-order Bessel function. Entry points of the subroutine FN are SI (sine integral), CI (cosine integral), ELLIPK (elliptic integral K of the first kind), and ELLIPE (elliptic integral E of the second kind). BES0 (zeroth-order Bessel function) is an entry point of subroutine BES1.

8. The simultaneous equations are solved for P_S and P_A (SIMX),

9. P(J) is evaluated and printed (SOLRING),

10. Impedance values are calculated (SOLRING),

11. Far-field pressure values are calculated and printed (FARFLD).

The special functions used by the SHIP program are evaluated using polynomial approximations found in Refs. 6a, 6b, and 6c. The elliptic integrals of the first and second kind are calculated in the function subroutines ELLIPK and ELLIPE respectively. The sine and cosine integrals (si and Ci) are calculated in the function subroutines SI and CI. ELLIPK, ELLIPE, SI, and CI are entry points of function FN(X). The zeroth-order and first-order Bessel functions ($J_0(x)$ and $J_1(x)$) are calculated separately in subroutine BES0 and subroutine BES1 and simultaneously in subroutine BESL.

PROGRAM INPUT AND OUTPUT

Input

The input parameters are summarized in Table 1. As illustrated in Fig. 5, ROUT is the outer radius of a ring, RIN is the inner radius, and 2H is the height. For a cylinder ROUT is the radius and RIN is set equal to 0. RIN, ROUT, and H may be given in any units which are consistent with the value chosen for the wavenumber FK (FK = ω/c). It is often convenient to choose FK equal to unity.

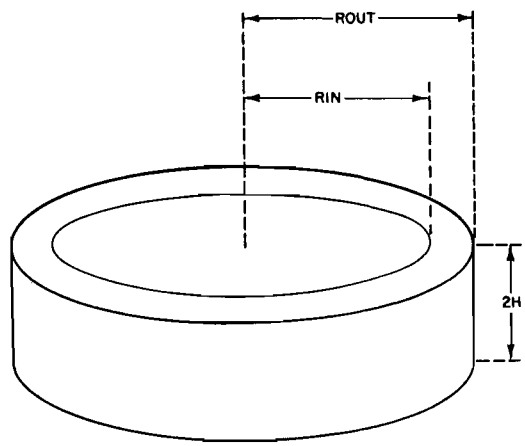


Fig. 5 - Geometry of the ring

The surface of the ring or cylinder is divided up into bands over which the surface pressure and normal velocity are considered to be constant. The number of bands is determined by the acoustic wavelength and the accuracy desired. IMAX is the number of bands on the top (or bottom) surface of the ring or cylinder. JMAXH (= JMAX/2) is half the number of bands on the outside and inside surfaces of the ring or on the outside surface of the cylinder. For a cylinder the number of bands (LCMAX) will be 2(IMAX) + JMAX, and for a ring LCMAX = 2(IMAX + JMAX). The bands on the top surfaces are circular annuluses of width (ROUT - RIN)/IMAX. The bands on the side surfaces are cylindrical shells of height H/JMAXH and radius either RIN or ROUT depending on whether the band is on the inside or outside surface. The bands are numbered as indicated in Figs. 2 and 3 and explained in the text accompanying those figures. Both JMAXH and IMAX must be less than or equal to 10 (LCMAX $\leq$ 60).

The average normal velocity on the Jth band is VEL(J) and is defined as being positive when it points outward toward the fluid. Phase shading may be accounted for by using complex values for VEL(J). The normal velocity distribution is thus approximated by the one-dimensional complex array VEL(J), J = 1, ..., LCMAX. The parameter ISYM expresses the symmetry of the velocity distribution. If the velocity distribution is symmetric with respect to the median plane, ISYM is set equal to +1; if it is antisymmetric, ISYM is set equal to -1; and if there is no symmetry, ISYM is set equal to 0. When ISYM is ±1, only the first LCMAXH values of VEL(J) are needed.

The Gaussian quadrature order is NQD1. NQD1 may be either 10, 20, or 32; in most cases 32 would be used.

Table 1
Input Parameters for Program SHIP

Parameter	Definition	Units	Comments
RIN	Inner radius of a ring	Any units consistent with FK	For a cylinder, set RIN equal to 0
ROUT	Outer radius of a ring or cylinder	Same as RIN	—
H	Half-height of a ring or cylinder	Same as RIN	—
ISYM	Symmetry parameter	Integer	ISYM = +1, if symmetric w.r.t. z = 0; = -1, if antisymmetric w.r.t. z = 0; = 0, if nonsymmetric
JMAXH	Half the number of bands on the inside (or outside) surface	Integer	Must be ≤ 10
IMAX	Number of bands on the top (or bottom) surface	Integer	Must be ≤ 10
NQD1	Gaussian quadrature order	Integer	Must be 10, 20, or 32
VEL(J)	Velocity on the Jth band (positive toward the fluid)	Meter-kilogram-second (mks) (complex)	J = 1, . . . , LCMAX (ISYM = 0), = 1, . . . , LCMAXH (ISYM = ±1)
FK	Wavenumber	Any units consistent with RIN, ROUT, and H	FK = 1.0 is convenient
ICOR	Far-field index	Integer	ICOR = 0 if the far field is calculated at infinity; ICOR = 1 if the far field is calculated at a distance DIST from the origin
DIST	Far-field distance	Same as RIN	DIST must be much greater than ROUT and the wavelength
NPTS	Number of far-field points between 0 and 90°	Integer	When ISYM = 0, 2 NPTS points are evaluated between ±90°
MXD	MXD = 1 means that one or more bands are pressure released	Integer	If not specified, MXD is assumed to be 0. ISYM must be +1.
JMXT(J)	JMXT(J) = 0 implies pressure release on the Jth band	Integer	If MXD ≠ 1, JMXT(J) is ignored

If a far-field radiation pattern is desired, the statement CALL FARFLD(NPTS) should appear in program SHIP following the statement CALL SOLRING. If the field is desired at infinity (true far field), then the parameter ICOR should be set equal to 0. If, on the other hand, one is trying to duplicate or predict the radiation pattern as measured at a specific distance from the source, ICOR should be set equal to 1 and the parameter DIST should be set equal to the distance from the source. (DIST must be much greater than ROUT, H, and the acoustic wavelength, because FARFLD is incapable of generating true near-field pressures.) DIST must have the same units as RIN, ROUT, and H. NPTS is the number of points between to be evaluated in the far field between 0° and 90° (in the SHIP program 0° corresponds to the plane of the ring, 90° to the axis of symmetry). If ISYM $\neq 1$, a number of points which is twice NPTS will be evaluated between -90° and +90°.

If pressure-release conditions are to be used over part of the surface, MXD would be set equal to 1 and JMXT(J) set equal to 0 for those bands which are pressure released. For the nonpressure-released bands JMXT(J) would be set equal to 1. For example if bands 1, 7, and 9 were pressure released, JMXT(1), JMXT(7), and JMXT(9) would be set equal to 0, and all other JMXT(J) would be set equal to 1. Pressure-release boundary conditions may be used only when ISYM = 1; thus JMXT(1) = 0 implies that both band 1 and band LCMAX are pressure released.

In the version of the program listed in Appendix A, RIN, ROUT, H, IMAX, JMAXH, and VEL(J) are read from cards. The remaining input parameters appear in program SHIP as Fortran statements. The first data card contains RIN, ROUT, and H in a (5X, 3F10.3) FORMAT, the second card contains JMAXH and IMAX in a (5X, 2I5) FORMAT, and the next LCMAXH cards (LCMAX if ISYM = 0) contain VEL(J) in a (5X, c(F10.3, F10.3)) FORMAT. If more than one case is being run, these cards would be followed by other similar sets. ISYM must be specified before VEL(J) is read in.

The input format given here can (and should) be altered to meet the specific requirements of the user. If, for example, many frequencies are to be used for a given ring (or cylinder) with the same velocity distribution, the statement 7 CONTINUE should be placed after the statement 604 CONTINUE followed by a READ statement for FK (or if desired a READ statement for a frequency may be used in conjunction with an arithmetic statement relating frequency to wavenumber). This would eliminate the necessity of reinputting RIN, ROUT, H, IMAX, JMAXH, and VEL(J) for each case. If many different velocity distributions are to be used for a single geometry and frequency, it is unnecessary to reevaluate the matrix elements each time. Reevaluating the matrix elements can be avoided by adding an ENTRY XXXX statement in subroutine SOLRING following the CALL ERING statement. After the first time through the program, one would CALL XXXX instead of calling SOLRING in program SHIP. If the number of velocity distributions is very large, it may be worthwhile to invert the DM matrix instead of solving the simultaneous equations each time through.

Output

The output of the SHIP program consists of:

- The original input — RIN, ROUT, H, FK, NQD1, IMAX, JMAX, and VEL(J) for J = 1, . . . , LCMAX.

- The real and imaginary parts of the surface pressure at the center of each band in mks units.

- The real and imaginary parts of the radiation impedance in units of ρcA, where ρ is the density of water, c is its sound speed, and A is the total surface area of the ring or cylinder. The impedance is normalized to a velocity of 1 m/sec.

- The far-field pressures in decibels relative to the value in the median plane (0°).

ACCURACY AND SPEED

For a given band structure SHIP and conventional finite-element Helmholtz integral methods should give identical results for the acoustic pressure at the center of each band. To test the speed and accuracy of the SHIP program, the program was compared with the NUC CHIEF [3] program (using CHIEF's axial symmetry option) for several problems. One such comparison is summarized in Table 2. In this case both SHIP and CHIEF used 12 bands (IMAX = 2; JMAXH = 2) for the ring described in the caption of Table 2. The SHIP program, which ran 4.8 seconds (execution time on a CDC 3800 computer) was compared with three CHIEF runs of progressively increasing computational accuracy. The first, which ran 182 seconds, had 12 vertically sliced subdivisions and used an eight-by-eight Gaussian quadrature to perform the numerical integrations. The second, which ran 331 seconds, had 24 vertically sliced subdivisions and used an eight-by-eight Gaussian quadrature. The third and most accurate, which ran 661 seconds, had 24 vertically sliced subdivisions and used an eight-by-16 Gaussian quadrature. In all cases the SHIP results were closest (within 0.13%) to the 661-second run on CHIEF. For this case the SHIP program was between 70 and 140 times faster than the CHIEF program for the same degree of computational accuracy.* The accuracy of the calculations (for a given number of bands) is probably better than 0.1%.

The preceding comparison tests only the precision of the calculations, not the actual accuracy of the results. That is, this comparison indicates that SHIP solves Eq. (11) to within 0.1%, but it in no way indicates whether this solution adequately represents the solution to Eq. (3). The overall accuracy of the method can be ascertained only by comparing SHIP results with known analytical solutions. Following Schenck [3], if the velocity input used is the average normal velocity over each band due to a point source at the center of the cylinder, and if a sufficient number of bands are used, the pressure calculated by SHIP should approach the pressure which would be present due to the point source at the center of the cylinder, and the far-field pattern should approach a perfect circle. The results of such a test are summarized in Table 3.

In each case ROUT = H = 1.0 and the surface of the cylinder was subdivided into 40 bands (JMAXH = IMAX = 10). The wavenumber FK varied between 0.5 and 8.0. The SHIP results for the surface pressures generally differed from the exact results by less than 1.0% for FK = 2 and by less than 3.0% for FK = 8. The agreement was best on the bands on the outside surface (bands 11 through 20) and worst on band 1 (because of its relatively small surface area), on band 10 (because of its proximity to the edge, where the Helmholtz integral equation is invalid), and on those bands where the surface pressure is relatively small. The far-field radiation patterns were never off by more than 0.5 dB. The execution times for these runs were approximately 31 seconds. Generally one can expect to obtain good results whenever the bands are less than 1/8 wavelength wide; far-field patterns and impedances may be satisfactory with bands as wide as 1/4

*There is, of course, no way to tell whether the CHIEF results were, as they seemed to be, converging toward the SHIP result or whether the 661-second CHIEF run was more accurate, without running even more time-consuming CHIEF runs. This did not seem worthwhile, since the inaccuracies due to the small number of bands are far more serious than the 0.065% difference between the most accurate CHIEF run and SHIP.

Table 2

Comparison of SHIP and CHIEF (With Axial Symmetry Option) Results for a Ring With 12 Bands (IMAX = 2, JMAXH = 2), RIN = 1.680, ROUT = 2.040, H = 0.600, Velocity of Inner Surface -1.029 m/sec, Velocity of Outer Surface 0.971 m/sec, Velocity of Top (and Bottom) Surface 0.097 m/sec, and FK = 1.000. The Times in Parentheses are the Execution Times of the Programs When Run on the NRL CDC 3800 Computer. The Percentages in Parentheses are the Percent Differences between the CHIEF and SHIP Results.

Real Part of Surface Pressure (10^{-5} mks units)					Imaginary Part of Surface Pressure (10^{-5} mks units)				
Band	SHIP (4.8 sec)	CHIEF (182 sec)	CHIEF (331 sec)	CHIEF (661 sec)	Band	SHIP (4.8 sec)	CHIEF (182 sec)	CHIEF (331 sec)	CHIEF (661 sec)
1	-19.0201	-18.6578 (1.9%)	-18.9888 (0.16%)	-19.0074 (0.07%	1	-14.4728	-14.5722 (0.7%)	-14.4655 (0.05%)	-14.4654 (0.05%)
2	-16.2642	-15.8962 (2.2%)	-16.2339 (0.19%)	-16.2532 (0.07%)	2	-10.3496	-10.3882 (0.37%)	-10.3430 (0.06%)	-10.3440 (0.05%)
3	-7.2124	-7.7303 (7.2%)	-7.2269 (0.2%)	-7.2097 (0.04%)	3	2.2604	1.8329 (18.9%)	2.2424 (0.80%)	2.2574 (0.13%)
4	-3.7169	-3.6343 (2.3%)	-3.7075 (0.25%)	-3.7147 (0.06%)	4	5.1875	5.7694 (11.2%)	5.2256 (0.73%)	5.1881 (0.01%)
5	2.1435	2.0238 (5.6%)	2.1370 (0.30%)	2.1417 (0.08%)	5	13.1613	12.9290 (1.8%)	13.1314 (0.23%)	13.1518 (0.07%)
6	3.4662	3.3689 (2.8%)	3.4589 (0.21%)	3.4635 (0.08%)	6	15.4715	15.2842 (1.2%)	15.4459 (0.17%)	15.4617 (0.06%)
Maximum deviation		7.2%	0.30%	0.08%	Maximum deviation		18.9%	0.80%	0.13%
Minimum deviation		1.9%	0.16%	0.04%	Minimum deviation		0.37%	0.05%	0.01%
Average deviation		3.6%	0.22%	0.07%	Average deviation		5.67%	0.34%	0.06%

Table 2

Comparison of SHIP and CHIEF (With Axial Symmetry Option) Results for a Ring With 12 Bands (IMAX = 2, JMAXH = 2), RIN = 1.680, ROUT = 2.040, H = 0.600, Velocity of Inner Surface -1.029 m/sec, Velocity of Outer Surface 0.971 m/sec, Velocity of Top (and Bottom) Surface 0.097 m/sec, and FK = 1.000. The Times in Parentheses are the Execution Times of the Programs When Run on the NRL CDC 3800 Computer. The Percentages in Parentheses are the Percent Differences between the CHIEF and SHIP Results.

Real Part of Surface Pressure (10^{-5} mks units)					Imaginary Part of Surface Pressure (10^{-5} mks units)				
Band	SHIP (4.8 sec)	CHIEF (182 sec)	CHIEF (331 sec)	CHIEF (661 sec)	Band	SHIP (4.8 sec)	CHIEF (182 sec)	CHIEF (331 sec)	CHIEF (661 sec)
1	-19.0201	-18.6578 (1.9%)	-18.9888 (0.16%)	-19.0074 (0.07%	1	-14.4728	-14.5722 (0.7%)	-14.4655 (0.05%)	-14.4654 (0.05%)
2	-16.2642	-15.8962 (2.2%)	-16.2339 (0.19%)	-16.2532 (0.07%)	2	-10.3496	-10.3882 (0.37%)	-10.3430 (0.06%)	-10.3440 (0.05%)
3	-7.2124	-7.7303 (7.2%)	-7.2269 (0.2%)	-7.2097 (0.04%)	3	2.2604	1.8329 (18.9%)	2.2424 (0.80%)	2.2574 (0.13%)
4	-3.7169	-3.6343 (2.3%)	-3.7075 (0.25%)	-3.7147 (0.06%)	4	5.1875	5.7694 (11.2%)	5.2256 (0.73%)	5.1881 (0.01%)
5	2.1435	2.0238 (5.6%)	2.1370 (0.30%)	2.1417 (0.08%)	5	13.1613	12.9290 (1.8%)	13.1314 (0.23%)	13.1518 (0.07%)
6	3.4662	3.3689 (2.8%)	3.4589 (0.21%)	3.4635 (0.08%)	6	15.4715	15.2842 (1.2%)	15.4459 (0.17%)	15.4617 (0.06%)
Maximum deviation		7.2%	0.30%	0.08%	Maximum deviation		18.9%	0.80%	0.13%
Minimum deviation		1.9%	0.16%	0.04%	Minimum deviation		0.37%	0.05%	0.01%
Average deviation		3.6%	0.22%	0.07%	Average deviation		5.67%	0.34%	0.06%

Table 3
Comparison of SHIP Results with the Exact Solution for a Point Source at the Center of a Cylinder with H = ROUT = 1 for Different Values of the Wavenumber FK. IMAX = JMAXH = 10 (Total Number of Bands LCMAX = 40). Surface Pressures are Given for Every Other Band on the Top Half of the Cylinder.

Method	Surface Pressure (10^{-6} mks units)									
	FK = 0.5		FK = 1.0		FK = 2.0		FK = 5.0		FK = 8.0	
	Real	Imaginary	Real	Imaginary	Real	Imaginary	Real	Imaginary	Real	Imaginary
Band 2										
SHIP	-1.433	-2.590	-1.254	-0.787	-0.672	0.3242	0.270	-0.102	-0.1743	0.0479
Exact	-1.437	-2.596	-1.257	-0.787	-0.667	0.3237	0.279	-0.100	-0.1803	0.0432
Band 4										
SHIP	-1.427	-2.438	-1.231	-0.693	-0.608	0.3688	0.229	-0.158	-0.1398	0.1021
Exact	-1.431	-2.443	-1.235	-0.693	-0.604	0.3689	0.236	-0.156	-0.1438	0.1031
Band 6										
SHIP	-1.415	-2.207	-1.191	-0.548	-0.501	0.4283	0.141	-0.2214	-0.0452	0.1525
Exact	-1.420	-2.212	-1.195	-0.547	-0.498	0.4292	0.144	-0.2203	-0.0477	0.1572
Band 8										
SHIP	-1.398	-1.939	-1.133	-0.378	-0.361	0.4788	0.0093	-0.2395	0.0814	0.1231
Exact	-1.404	-1.946	-1.139	-0.378	-0.359	0.4807	0.0080	-0.2399	0.0816	0.1259
Band 10										
SHIP	-1.371	-1.656	-1.055	-0.201	-0.199	0.5000	-0.123	-0.174	0.1317	-0.00465
Exact	-1.384	-1.678	-1.068	-0.207	-0.203	0.5044	-0.125	-0.178	0.1358	-0.00529
Band 12										
SHIP	-1.397	-1.811	-1.106	-0.291	-0.2812	0.4982	-0.0644	-0.2196	0.1252	0.0662
Exact	-1.395	-1.811	-1.105	-0.292	-0.2823	0.4969	-0.0630	-0.2197	0.1256	0.0680
Band 14										
SHIP	-1.414	-2.079	-1.170	-0.462	-0.4299	0.4592	0.0769	-0.240	0.0184	0.1544
Exact	-1.413	-2.081	-1.169	-0.464	-0.4315	0.4573	0.0791	-0.239	0.0183	0.1561
Band 16										
SHIP	-1.427	-2.331	-1.218	-0.621	-0.554	0.4012	0.195	-0.1931	-0.1053	0.1370
Exact	-1.426	-2.335	-1.217	-0.625	-0.556	0.3987	0.196	-0.1906	-0.1038	0.1359
Band 18										
SHIP	-1.436	-2.527	-1.249	-0.744	-0.639	0.3461	0.2641	-0.1271	-0.1685	0.0726
Exact	-1.434	-2.532	-1.248	-0.748	-0.642	0.3429	0.2631	-0.1244	-0.1681	0.0695
Band 20										
SHIP	-1.439	-2.622	-1.262	-0.803	-0.678	0.3170	0.289	-0.0894	-0.1829	0.0326
Exact	-1.438	-2.629	-1.262	-0.808	-0.680	0.3134	0.287	-0.0868	-0.1850	0.0291
Maximum Deviation of the Far Field From a Circle										
	0.003 dB		0.006 dB		0.073 dB		0.201 dB		0.418 dB	

wavelength. SHIP results will be further compared with both analytical and experimental results in the next section. The surface Helmholtz integral equation is known to fail at certain frequencies referred to as characteristic frequencies; these frequencies as they pertain to the SHIP program will be briefly discussed in the final section.

The running time for the SHIP program increases approximately quadratically with increasing values of IMAX and approximately linearly with JMAXH. The program was written primarily for ring sources and is thus considerably more efficient for rings than for cylinders. In fact, for given values of IMAX and JMAXH the ring case will run only slightly longer than the cylinder case, even though there are an additional JMAX bands for the ring.

EXAMPLES

Magnetostrictive Ring

Comparisons were made with experimental results obtained by J. Sinsky [7] of NRL for 6-inch-diameter magnetostrictively driven permendur rings. Using a single velocity distribution (determined by a combination of measurement, theory, and trial and error), good agreement with the experimentally determined far-field radiation patterns was achieved for frequencies between 6 and 15 kHz. Some representative far-field patterns are shown in Fig. 6. Using a simple lumped-element dynamical model (mechanical impedance parameters determined from the ring mass and resonant frequency in air), good agreement with the experimental axial and radial frequency-response curves was also achieved (Fig. 7).

For the far-field computations 50 bands were used with typical execution times of about 19 seconds plus 9 seconds to compute and print 45 far-field points.* The input parameters were: JMAXH = 10, IMAX = 5, ISYM = 1, NQD1 = 32, FK = 0.1612×(frequency in kilohertz), RIN = 1.720, ROUT = 2.120, H = 0.616, ICOR = 1, DIST = 80.234, NPTS = 45, and

$$\begin{aligned} \mathrm{VEL(I)} &= -1.000, & \mathrm{I} &= 1, \ldots, 10, \\ &= -0.250, & \mathrm{I} &= 11, \ldots, 15, \\ &= 1.000, & \mathrm{I} &= 16, \ldots, 25. \end{aligned}$$

The use of ICOR = 1 and a finite value of DIST conformed to the experimental far-field measurement. The most significant discrepancies between the SHIP results and the experimental results occur in the axial direction. This is no doubt due to our incomplete knowledge of the exact velocity distribution, particularly the difference between the velocity on the inside surface and the velocity on the outside surface.

Circular Piston in a Finite Rigid Baffle

Another example (Fig. 8) is a piston of radius B set in a rigid annular baffle of inner radius B and outer radius A. The problem considered was to determine the far-field radiation pattern and radiation impedance of the piston. The velocity of the piston was assumed to be $v_0 \cos \omega t$, and the velocity of the baffle was assumed to be zero. The underside of the system (a circle of radius A) was also assumed to have zero velocity.

*All execution times quoted in this section are for the CDC 3800 computer.

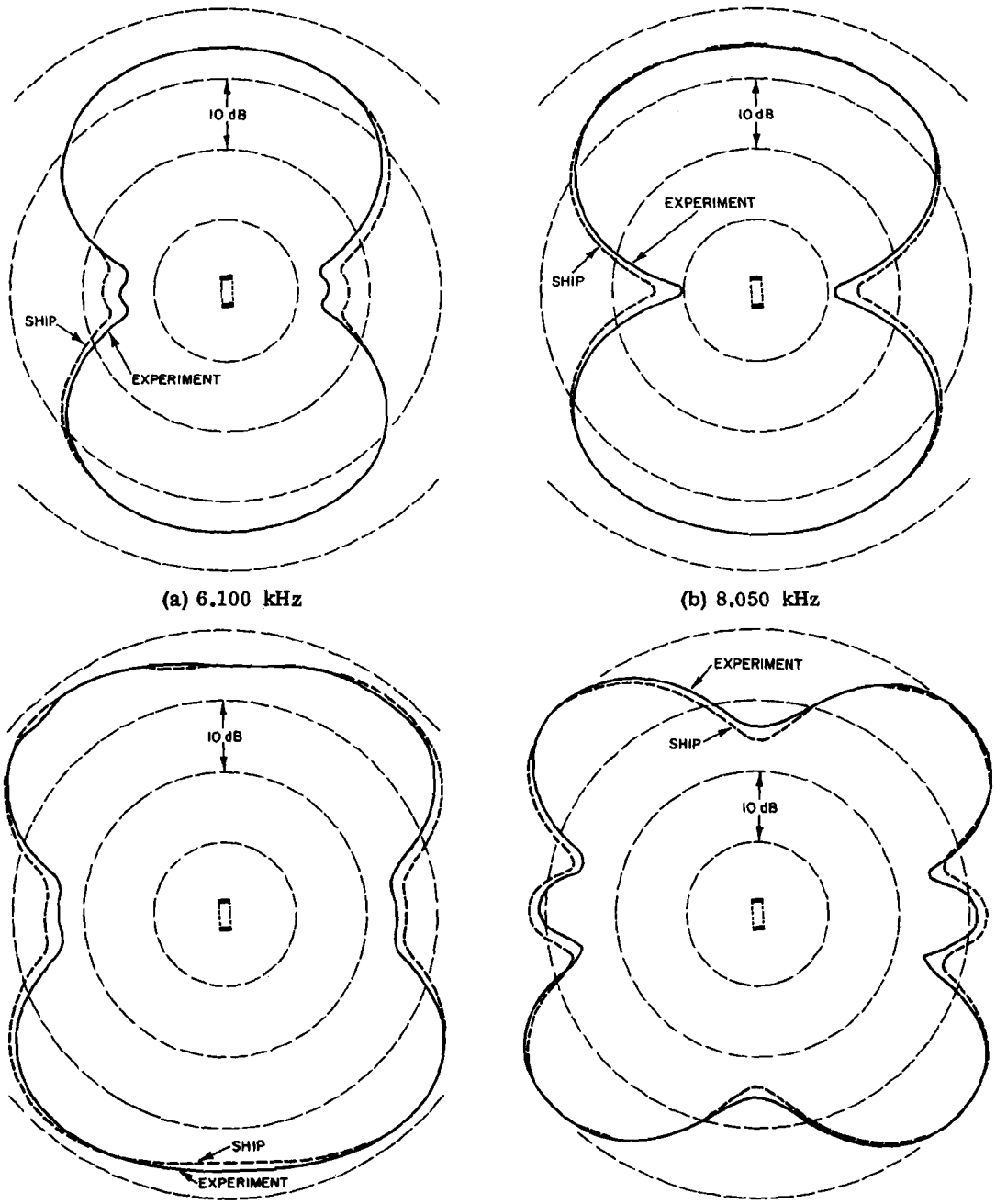


Fig. 6 - Far-field patterns for a magnetostrictive ring as measured [7] and calculated by the SHIP program. The four plots are not normalized to the same absolute pressure at like circles.

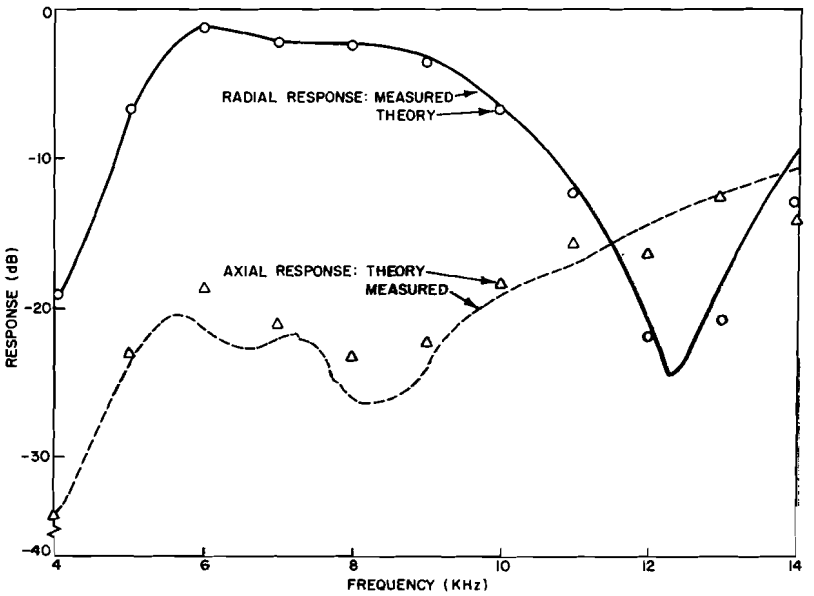


Fig. 7 - Frequency response at constant current for a magnetorestrictive ring. The best fit is to the radial response only. This plot could be used in drawing a normalized version of Fig. 6.

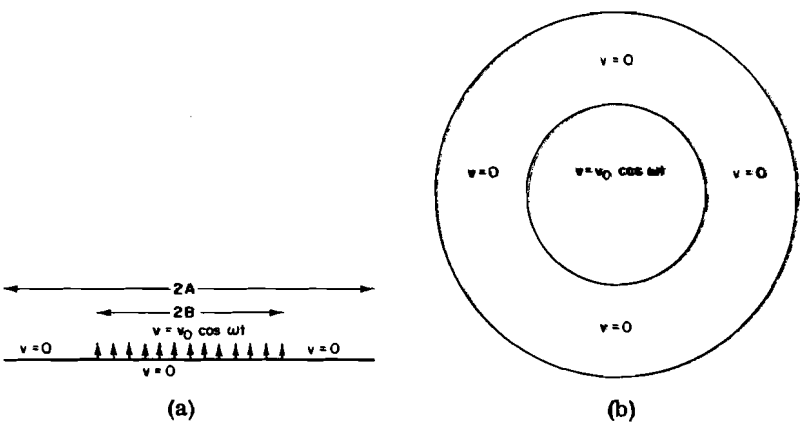


Fig. 8 - Piston in a finite rigid baffle

First the problem was handled exactly using oblate spheroidal wave functions. Then the disk was approximated as a finite cylinder with a large diameter-to-height ratio, the approximate system was solved using SHIP, and the results were compared with the exact answer. Specifically, the case was considered in which kA = 8.0, kB = 4.0, and v_0 = 1 m/sec. An arbitrary large diameter-to-height ratio of 160 was chosen. The input parameters for SHIP were: IMAX = 10, JMAXH = 1, NQD1 = 32, NPTS = 45, ICOR = 0, ISYM = 0 (since the system is not symmetric with respect to the median plane), RIN = 0.00 (indicating a cylinder), ROUT = 8.00, H = 0.05 (arbitrary small value), FK = 1.00, and

VEL(I) =	1.000,	I = 1, ..., 5	(velocity of the piston is unity),
=	0.000,	I = 6, ..., 10	(velocity of the baffle is zero),
=	0.000,	I = 13, ..., 22	(velocity of the underside is zero),
=	0.000,	I = 11, 12	(arbitrarily the velocity of the outside is zero).

The execution time for this run was 24 seconds. The resulting far-field pattern is plotted as the solid line in Fig. 9. The circles are the exact results as calculated by Van Buren and King [8] using oblate spheroidal wave functions. The agreement between SHIP and the exact results is good. The radiation impedance (in units of ρcA) as calculated by Van Buren and King [8] is 0.944 + 0.122i; the SHIP result is 0.936 + 0.128i.

Circular Piston in an Infinite Rigid Baffle

A circular piston of radius a in an infinite rigid baffle can be modeled as a finite cylinder of radius a and height $h \ll a$ with a symmetric velocity distribution. The symmetric velocity distribution assures that the normal velocity for $z = 0$, $r > a$ will be 0, which is precisely the boundary condition on the infinite baffle. The acoustic pressure on the surface of the piston can be evaluated by numerically integrating King's [9] integral for the surface pressure. It turns out that for this problem the integral equation effectively reduces to this same integral, enabling SHIP to accurately evaluate the surface pressures for acoustic sizes far larger than would ordinarily be possible.

Figure 10 shows SHIP results for the surface pressure compared with the exact results for a piston with $ka = 40$. The agreement is far better than one would ordinarily obtain with bands 0.63 wavelength wide. The input parameters for this run were: IMAX = 10, JMAXH = 2, NQD1 = 32, ISYM = 1, RIN = 0, ROUT = 40.0, H = 0.100, FK = 1.00, and

VEL(I) =	1.000,	I = 1, ..., 10	(velocity is unity on the piston),
=	0.0,	I = 11, 12	(velocity is zero on the edge).

Squirter

When an acoustically thin ring vibrates in its fundamental radial mode, little sound is radiated in the direction of the axis of symmetry. It is possible, however, to make the ring radiate primarily in this direction by pressure releasing the outside surface of the ring. A ring which is pressure released in this manner is referred to as a squirter. The squirter principle was easily demonstrated using the SHIP program.

First a thin ring was considered vibrating in its radial mode. The input parameters were: IMAX = 2, JMAXH = 10, NQD1 = 32, ISYM = 1, ICOR = 0, NPTS = 45, RIN = 1.800, ROUT = 2.000, H = 1.000, FK = 1.000, and

VEL(I) =	-1.000,	I = 1, ..., 10	(velocity of inside surface),
=	0.0,	I = 11, 12	(velocity of top surface),
=	1.000,	I = 13, ..., 22	(velocity of outside surface).

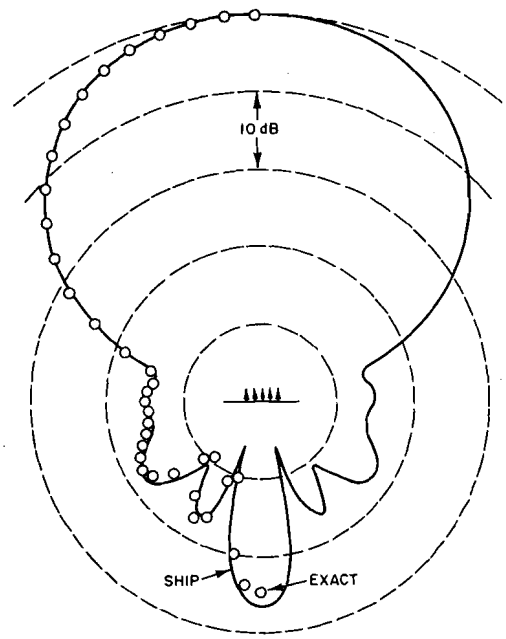


Fig. 9 - Far-field radiation pattern of a circular piston in a finite rigid baffle. The exact values were calculated by B. J. King using the computer program of Ref. 8.

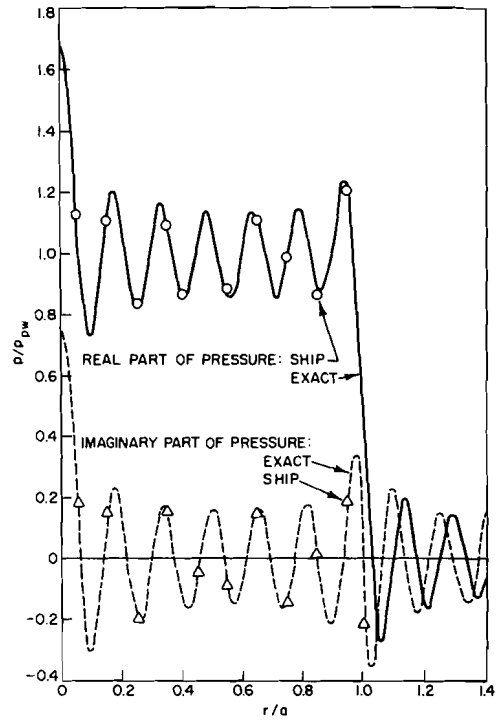


Fig. 10 - Surface pressure of a piston of radius a ($ka = 40$) in an infinite rigid baffle. p_{pw} is the plane-wave pressure.

The far-field radiation pattern for this ring is the solid line in Fig. 11. The radiation in the axial direction is 33 dB lower than the radiation in the radial direction. The same ring was operated as a squirter by pressure releasing the side (and top) of the ring. To do this all of the preceding parameters were used plus MXD = 1 (to indicate that pressure release is present) and

$$\begin{aligned} \text{JMXT(I)} &= 0, \quad I = 11, \ldots, 22 \quad \text{(these bands are pressure released),} \\ &= 1, \quad I = 1, \ldots, 10 \quad \text{(these bands are not pressure released).} \end{aligned}$$

The far-field radiation pattern for this squirter is a dashed line in Fig. 11. The radiation in the axial direction is now 6 dB higher than the radiation in the radial direction.

Band on a Prolate Spheroid

As another example of use of the SHIP program, a prolate spheroid of large eccentricity was approximately modeled as a finite cylinder, without concern for the effects of the ends. The half-height of the cylinder was chosen to be equal to the semimajor axis of the spheroid, and the radius of the cylinder was chosen to be equal to the semiminor axis of the spheroid. The example considered was a prolate spheroid whose acoustic length (2kH) is 14 and whose major-to-minor axis ratio is 10. The radiating portion of the spheroid is a band centered at $z = 0$ and 3/10 the length of the spheroid. This spheroid was modeled as a finite cylinder as shown in Fig. 12. The input parameters

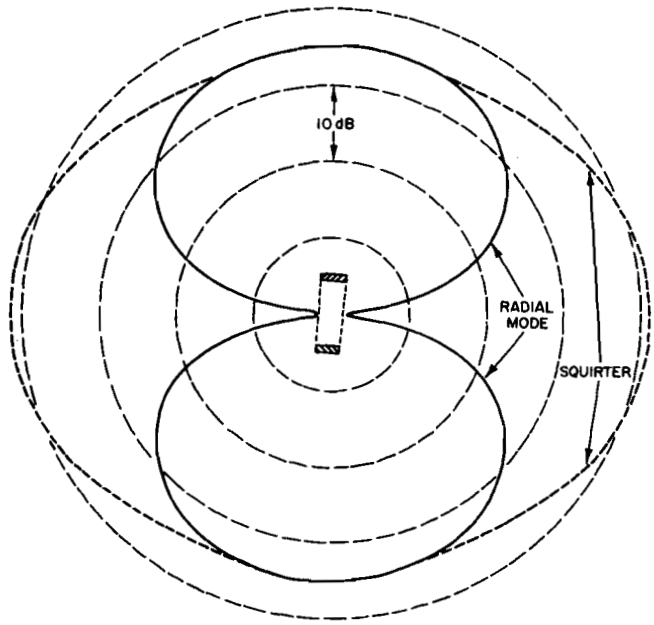


Fig. 11 - Comparison of a ring operating as a squirter with the same ring operating in the ordinary radial mode

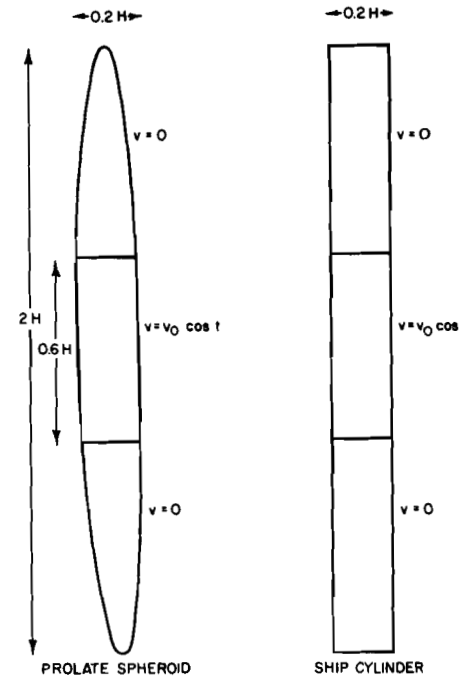


Fig. 12 - Approximation of a radiating band on a prolate spheroid by a band on a cylinder, so that the SHIP program can be used

were: IMAX = 3, JMAXH = 10, NQD1 = 32, ICOR = 0, ISYM = 1, NPTS = 45, RIN = 0.00 (indicating a cylinder), ROUT = 0.70, H = 7.00, FK = 1.000, and

$$\mathrm{VEL(I)} = 0.0, \quad I = 1, \ldots, 10$$

$$= 1.00, \quad I = 11, \ldots, 13 \quad \text{(velocity is unity on the band).}$$

The far-field radiation pattern is the solid line in Fig. 13; the circles are the far-field values for the prolate spheroid evaluated by Van Buren and King [8] using prolate spheroidal wave functions. The close agreement indicates that the effects due to the ends are negligible. The radiation impedance of the band as calculated by SHIP (in units of ρcA) is 0.646 + 0.478i, and the exact value as calculated by Van Buren [10] is 0.634 + 0.471i.

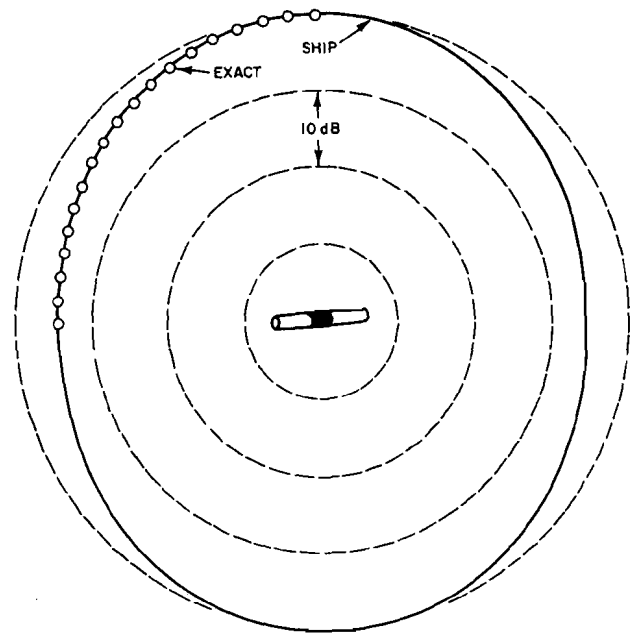


Fig. 13 - Far field of a prolate spheroid with a radiating band as calculated by King using the computer program of Ref. 8 and as approximated using SHIP in calculating the far field of a cylinder with a radiating band

CHARACTERISTIC FREQUENCIES

It is well known [2,3] that the surface Helmholtz integral method will give incorrect results at certain characteristic frequencies. These frequencies correspond to the eigenfrequencies of the interior Dirichlet problem. For a finite cylinder of height 2h and radius a the characteristic wavenumbers $(2\pi f/c)$ are given [3] by

$$k^*_{mnq} = \left[(m\pi/2h)^2 + (\alpha_{nq}/a)^2\right]^{1/2} ,$$

where m is a positive integer and α_{nq} is the qth root of the nth-order Bessel function J_n.

Due to the axial symmetry built into the SHIP program, it can be shown that the program will be affected only by those characteristic wavenumbers which have $n = 0$. This is because the interior eigenfunctions for $n \neq 0$ are not axially symmetric. Similarly, if ISYM = 1, SHIP will not be affected by characteristic wavenumbers which have even values of m; and when ISYM = -1, characteristic wavenumbers which have odd values of m will have no effect.

To demonstrate these conclusions as to which characteristic wavenumbers affect the SHIP program, the problem of the point source in a cylinder (Table 3) can be considered. For this problem ISYM = 1, so we should expect to have difficulty when $FK = k^*_{101}$ or when $FK = k^*_{102}$, but we should get correct results when $FK = k^*_{201}$ (since m is even) or when $FK = k^*_{111}$ (since $n \neq 0$). The far-field radiation patterns obtained from SHIP for these wavenumbers are shown in Fig. 14. The far-field patterns for $FK = k^*_{111}$ and $FK = k^*_{201}$ agree with the exact result (a perfect circle) correct to within 0.12 dB, whereas the far-field patterns for $FK = k^*_{101}$ and $FK = k^*_{102}$ differ drastically. Thus SHIP is seen to be affected by some but not all characteristic frequencies. The surface pressures for k^*_{111} and k^*_{201} were comparable in accuracy with those for noncharacteristic wavenumbers of similar magnitude. When $FK = k^*_{101}$ or k^*_{102}, the surface pressures were incorrect by an order of magnitude.

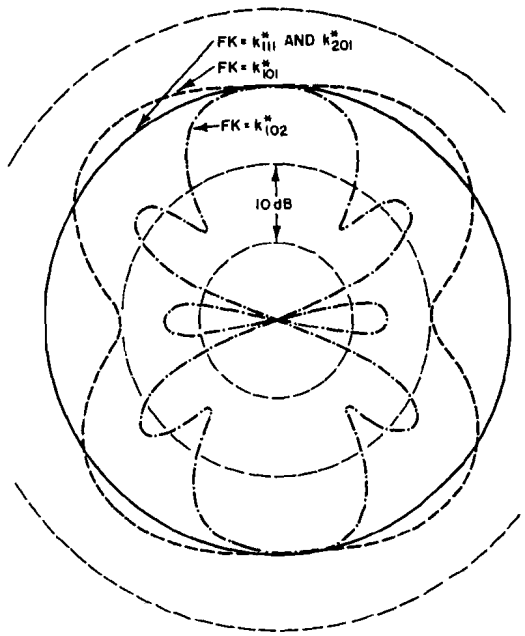


Fig. 14 - Far-field patterns for a point source in a cylinder for characteristic wavenumbers

It has been found possible to be quite near a characteristic frequency and still obtain satisfactory results. This is probably due to the relatively high accuracy to which the matrix elements are evaluated in SHIP.

Characteristic frequencies are usually not an important consideration in problems concerning rings. The lowest characteristic wavenumber is always greater than $\pi/(ROUT - RIN)$, so that there can be no characteristic wavenumbers unless the ring is more than 1/2 wavelength thick. In practice rings are almost never driven at such a high frequency.

REFERENCES

1. O.D. Kellogg, "Foundations of Potential Theory," Dover, New York, 1953, p. 164

2. L.G. Copley, J. Acoust. Soc. Am. 44, 28 (1968)

3. H.A. Schenck, J. Acoust. Soc. Am. 44, 41 (1968)

4. P.M. Morse and K.U. Ingard, "Theoretical Acoustics," McGraw-Hill, New York, 1968, p. 365

5. Y.L. Luke, "Integrals of Bessel Functions," McGraw-Hill, New York, 1962:

   a. p. 317

   b. p. 318

6. M. Abramowitz and I.A. Stegun, editors, "Handbook of Mathematical Functions," National Bureau of Standards, Applied Mathematical Series 55, Washington, D.C., U.S. Government Printing Office, 1966

   a. pp. 589-592

   b. pp. 231-233

   c. p. 369

7. J.A. Sinsky, "Acoustic Near-Field Measurements of a Free-Flooded Magnetostrictive Ring," NRL Report 7328, Dec. 30, 1971

8. A.L. Van Buren and B.J. King, "Acoustic Radiation From Two Spheroids," (to be published)

9. L.V King, Can. J. Res. 11, 135 (1934)

10. A.L. Van Buren, J. Acoust. Soc. Am. 50, 1343 (1971)

Appendix A

PROGRAM LISTING

12/22/71

```
      PROGRAM SHIP

      COMMON/TOY/NOD1,ISYM,ICOR/PIT/JMAX,JMAXH,IMAX/BLK1/H,A,FK,PI
      COMMON/VELC/VEL/LC/LCMAX,LCMAXH/DIST/DIST/RAD/RIN,ROUT/TIDY/FAST
      COMMON/MXD/JMXT,MXD
      DIMENSION VEL(60),JMXT(60)
      TYPE COMPLEX VEL
      CALL TIME
C     ROUT IS THE OUTER RADIUS OF THE RING RIN THE INNER RADIUS
C     FK IS THE WAVE NUMBER
C    IMAX IS THE NUMBER OF BANDS ON THE TOP AND BOTTOM OF THE RING
C     JMAXH IS HALF THE NUMBER OF BANDS ON THE INSIDE AND OUTSIDE SURFACES
C     JMAXH AND IMAX MUST BOTH BE LESS THAN OR EQUAL TO TEN
C     NOD1 SPECIFIES GAUSSIAN QUADRATURE ORDER - NOD1 MAY BE 10, 20 OR 32
C     FOR CYLINDER SET RIN = 0, AND ROUT EQUAL TO THE RADIUS
C     H IS THE HALF HEIGHT OF THE RING
C     VEL(I) SPECIFIES THE (COMPLEX) VELOCITY ON THE ITH BAND
C     ISYM=+1 | SYMMETRY ABOUT Z=0, -1 | ANTISYMMETRY AND ISYM=0 | NO SYMMETRY
      ISYM = 0
      FK=1
      NOD1 = 32
    7 CONTINUE
      READ 600, RIN,ROUT,H
      READ 601,JMAXH,IMAX
      LCMAXH = JMAXH + JMAXH + IMAX
      IF(RIN.EQ.0.)  LCMAXH = JMAXH + IMAX
      LCMAX = LCMAXH + LCMAXH
      NCT = LCMAXH
      IF(ISYM.EQ.0) NCT= LCMAX
      DO 604 J=1,NCT
      READ 922,VEL(J)
  604 CONTINUE
      IF(EOF,60)980,990
  990 CONTINUE
      A= .5*(RIN+ROUT)
      IF(RIN.EQ.0.) A = ROUT
      FKAH = FK*(A+H)
      FAST = 25.*NOD1/(32.*FKAH)
      IF(ISYM.EQ.0) GO TO 333
      DO 97 I=1,LCMAXH
   97 VEL(LCMAX + 1 - I) = ISYM*VEL(I)
  333 CONTINUE
      CALL SOLRING
C     NPTS IS THE NUMBER OF FARFIELD POINTS CALCULATED BETWEEN 0 AND 90 DEGREES
C     DIST IS THE FARFIELD DISTANCE (ICOR=1) IF ICOR=0 FIELD CALC, AT INFINITY
      ICOR = 0
      NPTS = 90
      CALL FARFLD(NPTS)
      GO TO 7
  600 FORMAT(5X,3F10.3)
  922 FORMAT(5X,C(F10.3,F10.3)  )
  601 FORMAT(5X,2I5)
  980 END
```

12/22/71

```
      SUBROUTINE SOLRING

      TYPE COMPLEX V,SUMJ,RHOC,ANS,VEL,DM,GM,VEL1,ANS1,ANS2
      DIMENSION REST(2640),ANS(60),V(30),VEL(60),GM(30,60),DM(60,60)
     1,VEL1(30),PZ(20),PLZ(20),PR(20),PLR(20),ANS1(30),ANS2(30),AREA(60)
      DIMENSION JMXT(60)
      COMMON/LC/LCMAX,LCMAXH/BLKA/PR,PLR,PZ,PLZ/RCC/JTOP
      COMMON/TOY/NOD1,ISYM,NOD3/PIT/JMAX,JMAXH,IMAX
      COMMON/TSUP/ISUP/BLK1/H,A,FK,PI/DEL/DELR,DELZ
      COMMON/RAD/RIN,ROUT/MXD/JMXT,MXD/5/DM,GM,REST/ANS/ANS/VELO/VEL
      JTOP = 10
      JMAX = 2*JMAXH
      CALL GQC
      CALL ERING
      RHOC = -FK*(0.0,1500000.)
      DO 14 J=1,LCMAXH
   11 ANS1(J)= ANS2(J) = (0.0,0.0)
      DO 14 I=1,LCMAXH
      DM(I,J) = DM(I,J) + DM(I, LCMAX + 1 -J)
      GM(I,J) = GM(I,J) + GM(I, LCMAX + 1 -J)
   14 CONTINUE
   88 IF(RIN.NE.0.) PRINT 744
      IF(RIN.EQ.0.) PRINT 746
  746  FORMAT(1H1 60X19HCYLINDER TRANSDUCER///)
  744 FORMAT (1H1 60X15HRING TRANSDUCER///)
      PRINT 701,RIN,ROUT,H,FK,NOD1,IMAX,JMAX
  701 FORMAT(* RIN=*F7.3*  ROUT=*F7.3*  H=*F7.3* FK=*F7.3*  NOD1=*I3
     3*  IMAX=*I3*  JMAX=*I3//)
      PRINT 365
  365 FORMAT (//9X3HNUM8X7HSP REAL11X12HSP IMAGINARY9X8HVEL REAL10X13HVE
     SL IMAGINARY/)
      IF(ISYM.EQ.-1) GO TO 99
      DO 7 J=1,LCMAXH
      VEL1(J) = .5*(VEL(J) + VEL(LCMAX + 1 - J) )
    7 CONTINUE
      IF(MXD.NE.1) GO TO 806
      DO 805 J=1,LCMAXH
      IF(JMXT(J).NE.0 ) GO TO 805
      DO 807 I=1,LCMAXH
      DM(I,J) = - GM(I,J)
  807 GM(I,J) = (0.0,0.0)
  805 CONTINUE
  806 CONTINUE
      DO 246 I=1,LCMAXH
      V(I) = (0.0,0.0)
      DO 246 J=1,LCMAXH
      DM(I+LCMAXH,J) = DM(I,J)
      V(I) = V(I) + GM(I,J)*VEL1(J)
  246 CONTINUE
      CALL SIMX(DM,LCMAXH,V,ANS1)
      IF(ISYM.EQ.1) GO TO 98
   99 CONTINUE
      DO 15 I=1,LCMAXH
      DO 15 J=1,LCMAXH
      DM(I,J) = DM(I+LCMAXH,J)
```

12/22/71

```
      DM(I,J) = DM(I,J) - DM(I,LCMAX + 1 - J) - DM(I,LCMAX + 1 -J)
   15 GM(I,J) = GM(I,J) - GM(I,LCMAX + 1 - J) - GM(I,LCMAX + 1 -J)
      DO 8 J=1,LCMAXH
      VEL1(J) = .5*(VEL(J) - VEL(LCMAX + 1 - J) )
    8 CONTINUE
      DO 248 I=1,LCMAXH
      V(I) = (0,0,0,0)
      DO 248 J=1,LCMAXH
      V(I) = V(I) + GM(I,J)*VEL1(J)
  248 CONTINUE
      CALL SIMX(DM,LCMAXH,V,ANS2)
   98 DO 10 J=1,LCMAXH
      ANS(J) = ANS1(J) + ANS2(J)
   10 ANS(LCMAX + 1 - J) = ANS1(J) - ANS2(J)
      IF(MXD.NE.1) GO TO 809
      DO 808 J=1,LCMAXH
      IF(JMXT(J).NE.0 ) GO TO 808
      VEL(J) = VEL(LCMAX+1-J) = ANS(J)
      ANS(J) = ANS(LCMAX+1-J) = (0,0,0,0)
  808 CONTINUE
  809 CONTINUE
      SUMJ = (0,0,0,0)
      DO 537 J=1,JMAXH
      IF(RIN.EQ.0.) GO TO 600
      AREA(J) = DELZ*RIN
      AREA(JMAXH+IMAX+J) = DELZ*ROUT
      GO TO 537
  600 AREA(IMAX+J) = ROUT*DELZ
  537 CONTINUE
      JRING = JMAXH
      IF(RIN.EQ.0.) JRING=0
      DO 538 J=1,IMAX
      AREA(JRING+J)= PR(J)*DELR
  538 CONTINUE
      DO 539 J=1,LCMAXH
  539 AREA(LCMAX + 1 - J) = AREA(J)
      DO 401 J=1,LCMAX
      SUMJ= SUMJ + CONJG(VEL(J))*ANS(J)*AREA(J)
      ANS(J) = RHOC*ANS(J)
      PRINT 400, J,ANS(J),VEL(J)
  400 FORMAT(1X,I10,2C(E20.8,E20.8)/)
  401 CONTINUE
      A1 = RIN+ROUT
      AREAT =A1*( H + H + ROUT - RIN)
      SUMJ = -FK*(0.0,1.0)*SUMJ/AREAT
      CALL TIME
      PRINT 506, SUMJ
  506 FORMAT(//* THE REAL AND IMAGINARY PARTS OF THE IMPEDANCE IN UNITS
     * OF RHO C A ARE*/C(E25.8,E25.8)//)
  403 FORMAT(3X,2I5,C(E20.8,E20.8)/)
  402 CONTINUE
      END
```

12/22/71

```
      SUBROUTINE ERING

      TYPE COMPLEX GM,ISM,SIM,ITM,TIM,IIM,SSM,TSM,STM,SBM,TBM,DM,RHOC,
     $ANS,VT,V,VS,VI,ISV,SIV,ITV,TIV,IIV,SSV,TSV,STV,SBV,TBV,SUMJ,TTV
      DIMENSION GM(30,60),DM(60,60),SSM(10,20),SSV(10,20),REST(2640),
     1TSM(10,20),TBM(10,10),STM(20,10),TTV(10,10),TSV(10,20),TBV(10,10),
     2 STV(20,10),ISM(10,20),ITM(20,10),SIM(10,20),TIM(10,20),IIM(10,20)
     3            ,ISV(10,20),SIV(10,20),ITV(20,10),TIV(10,20)
     4 ,IIV(10,20),PLZ(20),PR(20),PLR(20),PZ(20)
      COMMON/DEL/DELR,DELZ/BLKA/PR,PLR,PZ,PLZ/LC/LCMAX,LCMAXH
      COMMON/SSV/SSV/TSV/TSV/STV/STV/SBV/SBV/TBV/TBV/TTV/TTV
      COMMON/ISV/ISV/SIV/SIV/ITV/ITV/TIV/TIV/IIV/IIV
      COMMON/TIM/TIM/SIM/SIM/ISM/ISM/ITM/ITM/IIM/IIM
      COMMON/BLKC/TSM/UJ/TBM/PJ/STM/BLKB/SSM/VEL/VS,VI/VELT/VT
      COMMON/PIT/JMAX,JMAXH,IMAX/RAD/RIN,ROUT/5/DM,GM,REST
      COMMON/TIDY/FAST/RCC/JTOP/BLK1/H,A,FK,PI
      FKA = FK*A $ DELZ = H/JMAXH $ DELR = (ROUT-RIN)/IMAX
      DO 1 J=1,JMAXH
      KC = JMAX + 1 - J
      PZ(J) = H + .5*DELZ - J*DELZ
      PZ(KC) = -PZ(J)
      PLZ(J) = PZ(J)-.5*DELZ
    1 PLZ(KC)=PZ(KC) - .5*DELZ
      DO 2 I=1,IMAX
    2 PR(I) = RIN - .5*DELR + I*DELR
      CALL TIME
      CALL CALRES
      CALL CSSM
      CALL CTBM
      CALL CSTM
      CALL CTSM
      CALL TIME
      DO 6000 I=1,LCMAXH
      DO 6000 J=1,LCMAXH
      GM(I,J) = (0.0,0.0)
 6000 DM(I,J) = (0.0,0.0)
      IF(RIN.EQ.0.) GO TO 80
      DO 64 I=1,IMAX
      DO 64 J= 1,IMAX
      GM(JMAXH+I,JMAXH +J) = TTV(I,J)
   64 CONTINUE
      DO 31, I=1,JMAXH
      KITE = JMAXH + 1 - I
      DO 32 J=1,JMAXH
      DM(I,J)  = IIM(KITE,JMAXH +1 - J)
      GM(I,J)  = IIV(KITE,JMAXH +1 - J)
      GM(I,LCMAX + 1 - J) = IIV(KITE,JMAXH + J)
   32 DM(I,LCMAX + 1 - J) = IIM(KITE,JMAXH + J)
      DO 33 J=1,IMAX
      GM(I,JMAXH+J) = ITV(KITE,J)
   33 DM(I,JMAXH+J) = ITM(KITE,J)
      DO 34 J=1,JMAX
      GM(I,JMAXH+IMAX+J)= ISV(KITE,J)
   34 DM(I,JMAXH+IMAX+J)= ISM(KITE,J)
      DO 35 J=1,IMAX
```

12/22/71

```
      GM(I,JMAXH+LCMAXH+J)= ITV(JMAX +1-KITE,IMAX+1-J)
35    DM(I,JMAXH+LCMAXH+J)= ITM(JMAX +1-KITE,IMAX+1-J)
31    CONTINUE
      DO 36 I=1,IMAX
      KOOT = JMAXH+I
      DM(KOOT,KOOT) = (0.5,0.0)
      DO 37 J=1,JMAXH
      KITE = JMAXH + 1 -J
      GM(KOOT,J)= TIV(I,KITE)
      DM(KOOT,J)= TIM(I,KITE)
      GM(KOOT,LCMAX +1-J)= TIV(I,JMAXH+J)
37    DM(KOOT,LCMAX +1-J)= TIM(I,JMAXH+J)
      DO 38 J=1,JMAX
      GM(KOOT,JMAXH+IMAX+J)= TSV(I,J)
38    DM(KOOT,JMAXH+IMAX+J)= TSM(I,J)
      DO 39 J=1,IMAX
      GM(KOOT,LCMAXH+JMAXH+J)= TBV(I,IMAX+1-J)
39    DM(KOOT,LCMAXH+JMAXH+J)= TBM(I,IMAX+1-J)
36    CONTINUE
      DO 40 I=1,JMAXH
      KOOT = JMAXH + IMAX + I
      DO 41 J=1,JMAXH
      KITE = JMAXH+1-J
      GM(KOOT,J) = SIV(I,KITE)
      DM(KOOT,J) = SIM(I,KITE)
      DM(KOOT,LCMAX+1-J)= SIM(I,JMAXH+J)
      GM(KOOT,LCMAX+1-J)= SIV(I,JMAXH+J)
41    CONTINUE
      DO 42 J=1,IMAX
      DM(KOOT,JMAXH+LCMAXH+J)=STM(JMAX+1-I,IMAX+1-J)
      GM(KOOT,JMAXH+LCMAXH+J)=STV(JMAX+1-I,IMAX+1-J)
      GM(KOOT,JMAXH+J)= STV(I,J)
42    DM(KOOT,JMAXH+J)= STM(I,J)
      DO 43 J=1,JMAX
      GM(KOOT,JMAXH+IMAX+J) = SSV(I,J)
43    DM(KOOT,JMAXH+IMAX+J) = SSM(I,J)
40    CONTINUE
      RETURN
80    DO 84 I=1,IMAX
      DO 84 J= 1,IMAX
84    GM(I,J) = TTV(I,J)
      DO 71 I=1,IMAX
      DM(I,I)= (0.5,0.0)
      DO 72 J=1,JMAX
      GM(I,IMAX+J) = TSV(I,J)
72    DM(I,IMAX+J) = TSM(I,J)
      DO 71 J=1,IMAX
       GM(I,JMAX + IMAX + J) = TBV(I,IMAX + 1 - J)
71     DM(I,JMAX + IMAX + J) = TBM(I,IMAX + 1 - J)
      DO 73 I=1,JMAXH
      K= IMAX + I
      DO 74 J=1,IMAX
      GM(K,J) = STV(I,J)
74    DM(K,J) = STM(I,J)
      DO 75 J=1,JMAX
      GM(K,IMAX + J) = SSV(I,J)
```

12/22/71

```
75 DM(K,IMAX + J) = SSM(I,J)
   DO 73 J=1,IMAX
   GM(K,JMAX+IMAX+J)= + STV(JMAX +1-I,IMAX + 1 -J)
73 DM(K,JMAX+IMAX+J)= + STM(JMAX +1-I,IMAX + 1 -J)
   END
```

12/22/71

```
      SUBROUTINE CTBM

      COMMON/5/BT0(10,10,32),BT1(11,10,32),GT0(10,10,32),GT1(11,10,32)
      COMMON/BLK1/H,A,FK,PI/PIT/JMAX,JMAXH,IMAX/TTV/TTV/TIDY/FAST
      COMMON/BLK2/XI,WF/TGY/NQD1,N2,N3/BLK3B/BI0,BI1/BLK3/BS0,BS1
      COMMON/BLKA/PR,PLR,PZ,PLZ/DEL/DELR,DELZ/AOLD/GS0,GS1/RCC/JTOP
      COMMON/FS/FKFK,FKFAST,TFK/TBV/TBV/UJ/TBM
      DIMENSION PZ(20),PLZ(20),PR(20),PLR(20),TTV(10,10),TBM(10,10),
     1 TBV(10,10)
      TYPE COMPLEX TTV,TBM,TBV
      DIMENSION XI(32),WF(32),SUMT(10),GS0(10,32),GS1(10,32),UMT(10),
     1BUMT(10), BS0(10,32),BS1(10,32),QZ(32),BI0(10,32),BI1(10,32)
      N=NQD1
      HD = .5*DELR
      EPS = .000001
      PFK = .5*FK
      ARGP = 2.*H
      DO 7 JR=1,IMAX
      DO 7 IR=1,IMAX
      R1 = PR(IR) $ R2 = PR(JR)
      RL = R2 - HD          $ RU = R2 + HD
      UMR=UMI=SUMR=SUMI=SUMRB=SUMIB=0.
      DO 1 L =1,N                                                       00009400
      TRAP = PFK*XI(L)
      SIGMA = TRAP + PFK
      QZ(L)= FAST*(2.*TRAP - FK)
      QQ = SIGMA*SIGMA                                                  00009800
      D = SQRT(FKFK - QQ)                                               00009900
      SIGMAP = ARGP*SIGMA                                               00010000
      TRIGFI = SIN(SIGMAP) $ TRIGFR =  COS(SIGMAP)
      BS = BT0(IR,1,L)
      BU = RU*BT1(JR+1,1,L) - RL*BT1(JR,1,L)
      YAK = BS*BU*WF(L)/D
      AI = YAK*SIGMA*TRIGFI
      VI = YAK*TRIGFR
      AR = YAK*SIGMA*TRIGFR
      VR = YAK*TRIGFI
      SUMIB= SUMIB -  VI
      SUMRB = SUMRB +  VR
      SUMR = SUMR - AR
      UMI = UMI + YAK
    1 SUMI = SUMI + AI
      UMI = UMI*TFK
      SUMI = SUMI*TFK
      SUMR = SUMR*TFK
      SUMRB = SUMRB*TFK   $ SUMIB = SUMIB*TFK
      R = R1
      IF(R2.LE.R1) R = R2
      IF(R.EQ.0.) R = DELR
      FACT = 2./(R*PI*ARGP)
      RM = R1
      IF(R.EQ.R1) RM = R2
      FACT1 = PI*.5/RM
      UMT(1) = 0. $ SUMT(1) = -SUMR $ BUMT(1) = - SUMRB
      ICE = 1
```

12/22/71

```
  2 ICE = ICE + 1
    UMMIT = SUMMIT = SUMMITB = 0.
    DO 375 M=1,N                                                    00011300
    QZ(M) = QZ(M) + FKFAST
    RHO = QZ(M)      $ QQR = RHO*RHO
    GMS = SQRT(QQR + FKFK)                                          00011700
    BS =  BTO(IR,ICE,M)     $      GS =  GTO(IR,ICE,M)
    BU = RU*BT1(JR+1,ICE,M)  +  RL*BT1(JR,ICE,M)
    GU = RU*GT1(JR+1,ICE,M)  +  RL*GT1(JR,ICE,M)
    YAKR = BS*BU*WF(M)/GMS
    VAKR = YAKR + GU*GS*WF(M)/RHO
    UMMIT = UMMIT + VAKR
    EXPF1 = EXPF(-ARGP*RHO)
    ARTISTB = YAKR*EXPF1
    ARTIST = ARTISTB*RHO
    SUMMIT = SUMMIT + ARTIST                                        00012300
375 SUMMITB = SUMMITB + ARTISTB
    SUMT(ICE) = SUMT(ICE-1) + .25*FKFAST*SUMMIT
    BUMT(ICE) = BUMT(ICE-1)  + .25*FKFAST*SUMMITB
    UMT(ICE) = UMT(ICE-1) + .25*FKFAST*UMMIT
    FL = (ICE-1)*FKFAST
    FLT = FL + FACT1
    ERFC = (FACT/FL)*(EXPF(-ARGP*FL) - EXPF(-ARGP*FLT))
    ERFC = RM*ERFC
    ERFC = ERFC/SUMT(ICE)
    TESTER = 1.  - SUMT(ICE)/SUMT(ICE-1)
    TESTER = ABS(TESTER)
    IF(TESTER.LT.EPS.AND.ERFC.LT.EPS) GO TO 3
    IF(ICE.GE.NTOP) GO TO 3
    GO TO 2
  3 ERFC = TESTER*SUMT(ICE)
    ANSR = SUMT(ICE)              $ ANSI = -SUMI
    VNSI = -SUMIB  $   VNSR = - BUMT(ICE)
    TBM(IR,JR) = CMPLX(ANSR,ANSI)   $ TBV(IR,JR) = CMPLX(VNSR,VNSI)
          RARA = (R1-RU)*(R1+RU)
    DK = 4.*R1*RU/RARA
    PII = 1./(PI*RU)
    CU= PII*((R1+RU)*ELLIPE(DK) + (RU-R1)*ELLIPK(DK))
    CU = RU*CU
    RL = RL
          RARA = (R1+RU)*(R1+RU)
    DK = 4.*R1*RU/RARA
    PII = 1./(PI*RU)
    CL= PII*((R1+RU)*ELLIPE(DK) + (RU-R1)*ELLIPK(DK))
    CL = RL*CL
    CF = CU - CL
    CF = .5*CF
    ANSR = - UMT(ICE) - CF
  7 TTV(IR,JR) = CMPLX(ANSR,UMI)
    END
```

12/22/71

```
      SUBROUTINE CSSM

      DIMENSION SIV(10,20),IIV(10,20),SSV(10,20),ISV(10,20)
      DIMENSION SIM(10,20),IIM(10,20),SSM(10,20),ISM(10,20)
      TYPE COMPLEX SSV,SIV,ISV,IIV,SSM,SIM,ISM,IIM
      DIMENSION PZ(20),PLZ(20),PR(20),PLR(20)
      TYPE REAL IIIV,IIRV,ISIV,ISRV,ISB,IIB,IIT,IST,III,IIR,ISI,ISR
      DIMENSION SSR(10),IIR(10),SIB(10),ISB(10)
      DIMENSION GS0(10,32),GS1(10,32),XI(32),WF(32),SUMT(10)
      DIMENSION BS0(10,32),BS1(10,32),QZ(32)
      DIMENSION BI0(10,32),BI1(10,32),GI0(10,32),GI1(10,32)
      TYPE COMPLEX SS,II,YAKC,UMII,UMSS,UMIS
      DIMENSION SST(10),IIT(10),SIT(10),IST(10)
      DIMENSION BUMT(10),DUMT(10),EXPF1(32)
      COMMON/RAD/RIN,ROUT/TOY/NQD1,NQD2,NQD3/RCC/JTOP/BOLD/GS0,GS1
      COMMON/PIT/JMAX,JMAXH,IMAX/BLKA/PR,PLR,PZ,PLZ
      COMMON/SSV/SSV/IIV/IIV/SIV/SIV/ISV/ISV
      COMMON/BLKB/SSM/SIM/SIM/IIM/IIM/ISM/ISM/FS/FKFK,FKFAST,TFK
      COMMON/BLK1/H,A,FK,PI/BLK2/XI,WF/BLK3/BS0,BS1/TIDY/FAST
      COMMON/BBOLD/GI0,GI1/BLK3B/BI0,BI1/DEL/DELR,DELZ
      N=NQD1 $ FKFK = FK*FK$ TFK= .25*FK $ FKFAST = 2.*FAST*FK
      ARGP = .5*DELZ
      EPS = .00D001
      Z0 = H- .5*DELZ
      SSI=SSR=III=IIR=SII=SIR=ISI=ISR=0.
      UMII = UMIS = UMSS = (0.0,0.0)
      PFK = .5*FK
      DO 1 L =1,N                                                       00009400
      TRAP = PFK*XI(L)
      SIGMA = TRAP + PFK
      QZ(L)= FAST*(2.*TRAP - FK)
      QQ = SIGMA*SIGMA                                                  00009800
      D = SQRT(FKFK - QQ)                                               00009900
      SIGMAP = ARGP*SIGMA                                               00010000
      EXPF1(L) = EXPF(-QZ(L)*ARGP)
      TRIGFR  = 2.*COS(SIGMAP) - 2.
      TRIGFI = 2.*SIN(SIGMAP)
      YAK = TFK*WF(L)/SIGMA
      YAKR =-YAK*TRIGFR
      YAKI = YAK*TRIGFI
      YAKC = CMPLX(YAKR,-YAKI)
      YAKR = YAKR*D  $ YAKI = YAKI*D
      SSI = SSI + BS0(1,L)*BS1(1,L)*YAKI
      SSR = SSR + BS0(1,L)*BS1(1,L)*YAKR
      IIR = IIR + BI0(1,L)*BI1(1,L)*YAKR
      III = III + BI0(1,L)*BI1(1,L)*YAKI
      SII = SII + BS0(1,L)*BI1(1,L)*YAKI
      SIR = SIR + BS0(1,L)*BI1(1,L)*YAKR
      ISI = ISI + BI0(1,L)*BS1(1,L)*YAKI
      ISR = ISR + BI0(1,L)*BS1(1,L)*YAKR
       UMII =  UMII + YAKC*BI0(1,L)*BI0(1,L)
       UMIS =  UMIS + YAKC*BI0(1,L)*BS0(1,L)
       UMSS =  UMSS + YAKC*BS0(1,L)*BS0(1,L)
    1 CONTINUE                                                          00010900
      SUMT(1) = REAL(UMSS) $ BUMT(1) = REAL(UMII) $ DUMT(1) = REAL(UMIS)
```

12/22/71

```
      SST(1) = SSR $ IIT(1)= IIR $ SIT(1)  = SIR $ IST(1)= ISR
      EXFACT = EXPF(-FKFAST*ARGP)
      TPK = .25*FKFAST
      DO 501 ICE=2,JTOP                                                 00011100
      SUMSS=SUMII=SUMSI = SUMIS = 0.
      SUMMIT = SUMMITB = SUMMITD = 0.
      DO 375 M=1,N                                                      00011300
      QZ(M) = QZ(M) + FKFAST
      RHO = QZ(M)
      QQR = RHO*RHO                                                     00011600
      GMS = SQRT(QQR + FKFK)                                            00011700
      GMQ = 1./RHO
      GMR = GMS*GMQ
      EXPF1(M) = EXPF1(M)*EXFACT
      EXPF2 = 2.*(EXPF1(M) - 1.)
      YAK = -TPK*WF(M)
      YAKR = -EXPF2*YAK
      YAK = YAK + YAK
      SUMSS = SUMSS + BS0(ICE,M)*BS1(ICE,M)*YAKR*GMR
      SUMSS = SUMSS - GS0(ICE,M)*GS1(ICE,M)*YAKR
      SUMII = SUMII - BI0(ICE,M)*BI1(ICE,M)*YAKR*GMR
      SUMII = SUMII + GI0(ICE,M)*GI1(ICE,M)*YAKR
      SUMSI = SUMSI - BS0(ICE,M)*BI1(ICE,M)*YAKR*GMR
      SUMSI = SUMSI + GS0(ICE,M)*GI1(ICE,M)*YAK
      SUMIS = SUMIS + BI0(ICE,M)*BS1(ICE,M)*YAKR*GMR
      SUMIS = SUMIS - GI0(ICE,M)*GS1(ICE,M)*YAK
      BSBS = BS0(ICE,M)*BS0(ICE,M)
      BIBS = BI0(ICE,M)*BS0(ICE,M)
      BIBI = BI0(ICE,M)*BI0(ICE,M)
      YAKR = YAKR*GMQ
      SUMMIT = SUMMIT + BSBS*YAKR
      SUMMITB = SUMMITB + BIBI*YAKR
      SUMMITD = SUMMITD + BIBS*YAKR
  375 CONTINUE                                                          00012400
      SST(ICE) = SST(ICE-1) + SUMSS
      IIT(ICE) = IIT(ICE-1) + SUMII
      SIT(ICE) = SIT(ICE-1) + SUMSI
      IST(ICE) = IST(ICE -1) + SUMIS
      SUMT(ICE) = SUMT(ICE - 1)  +  SUMMIT
      BUMT(ICE) = BUMT(ICE - 1) + SUMMITB
  501 DUMT(ICE) = DUMT(ICE-1) + SUMMITD
      AA = 4.*ROUT*ROUT
      P = ARGP                                                          00013300
      PP = P*P                                                          00013400
      DK = AA/(PP + AA)
      SDK = SQRT(DK)                                                    00013600
      CF2 = -P*SDK*ELLIPK(DK)/(PI*AA)
      SSR= ROUT*(SST(JTOP) + CF2 + CF2)
      AA = RIN*RIN*4.
      DK = AA/(PP + AA)                                                 00013500
      SDK = SQRT(DK)
      CF2 = -P*SDK*ELLIPK(DK)/(PI*AA)                                   00013700
      IIR=  RIN*(IIT(JTOP) - CF2 - CF2)
      SSI = ROUT*SSI $ III= RIN*III $ SII= RIN*SII $ ISI=ROUT*ISI
      ICE = JTOP
      SIR = RIN*SIT(ICE) $ ISR = ROUT*IST(ICE)
```

12/22/71

```
      ISR = ISR - 1.0
      JM = JTOP-1
      FLAM = FKFAST*JM
      B= RIN $ R =  ROUT
      RB = ABS(R-B)
      ARG1 = RB*FLAM
      ARG2 = (R+B)*FLAM
      CF1 = 1./(PI*SQRT(R*B))
      CF3 =      CF1*(COS(ARG1)/FLAM + SIN(ARG2)/FLAM
     1 + RB*SI(ARG1)  - (R+B)*CI(ARG2) )
      FLAM = FLAM*ROUT
      FLAMB = FKFAST*RIN*JM
      CF2 = - 1./(PI*FLAMB)
      CF1 = - 1./(PI*FLAM)
      ANSR = SUMT(ICE) + CF1
      ANSRB = DUMT(ICE) + CF2
            ANSRD = DUMT(ICE) - CF3
      QI =-AIMAG( UMII)
      IIV(1,1) = RIN*CMPLX(ANSRB,QI)
      QI =-AIMAG( UMIS)
      ISV(1,1) = ROUT*CMPLX(ANSRD,QI)
      SIV(1,1) = RIN*CMPLX(ANSRD,QI)
      QI =-AIMAG( UMSS)
      SSV(1,1) = ROUT*CMPLX(ANSR,QI)
      IIM(1,1) =-CMPLX(IIR,III) + (0.5,0.0)
      SSM(1,1) = -CMPLX(SSR,SSI)  + (0.5,0.0)
      SIM(1,1) = - CMPLX(SIR,SII)
      ISM(1,1) = -CMPLX(ISR,ISI)
      DO 66 J=2,JMAX
      Z = PLZ(J)
      ARGP = Z0 - Z - DELZ
      ARGM = Z0 - Z
      SSI=SSR=III=IIR=SII=SIR=ISI=ISR=0,
      SSIV=SSRV=IIIV=IIRV=SIIV=SIRV=ISIV=ISRV= 0,
      DO 61 L=1,N
      TRAP = PFK*XI(L)
      SIGMA = TRAP + PFK
      QZ(L)= FAST*(2.*TRAP - FK)
      QQ = SIGMA*SIGMA                                                  00009800
      D = SQRT(FKFK - QQ)                                               00009900
      SIGMAP = ARGP*SIGMA                                               00010000
      SIGMAM = SIGMA*ARGM                                               00010100
      TRIGFR = COS(SIGMAP) - COS(SIGMAM)
      TRIGFI = SIN(SIGMAP) - SIN(SIGMAM)
      YAK =  TFK*WF(L)*D/SIGMA
      YAKR =-YAK*TRIGFR
      YAKI = YAK*TRIGFI
      VAKI = YAKI/D $ VAKR = YAKR/D
      ISR = ISR + BI0(1,L)*BS1(1,L)*YAKR
      ISI = ISI + BI0(1,L)*BS1(1,L)*YAKI
      SIR = SIR - BS0(1,L)*BI1(1,L)*YAKR
      SII = SII - BS0(1,L)*BI1(1,L)*YAKI
      III = III - BI0(1,L)*BI1(1,L)*YAKI
      IIR = IIR - BI0(1,L)*BI1(1,L)*YAKR
      SSR = SSR + BS0(1,L)*BS1(1,L)*YAKR
      SSI = SSI + BS0(1,L)*BS1(1,L)*YAKI
```

12/22/71

```
      IIIV= IIIV+ BI0(1,L)*BI0(1,L)*VAKI
      IIRV= IIRV+ BI0(1,L)*BI0(1,L)*VAKR
      ISRV= ISRV+ BI0(1,L)*BS0(1,L)*VAKR
      ISIV= ISIV+ BI0(1,L)*BS0(1,L)*VAKI
      SSRV= SSRV+ BS0(1,L)*BS0(1,L)*VAKR
   61 SSIV= SSIV+ BS0(1,L)*BS0(1,L)*VAKI
      SIIV = ISIV $ SIRV = ISRV
      FACT1 = PI*.5/A
      FACT = 2./(A*PI*ARGM)
      SST(1) = SSR $ IIT(1)= IIR $ SIT(1)  = SIR $ IST(1)= ISR
      SSB(1)= SSRV $ IIB(1)=  IIRV $ SIB(1) = SIRV $ ISB(1) = ISRV
      ICE = 1
    2 ICE = ICE  + 1
      SUMSS=SUMII=SUMSI = SUMIS = 0.
      VUMSS = VUMII=VUMSI=VUMIS  = 0.
      DO 376 M=1,N
      QZ(M) = QZ(M) + FKFAST
      RHO = QZ(M)                                                       00011500
      QQR = RHO*RHO                                                     00011600
      GMS = SQRT(QQR + FKFK)                                            00011700
      EXPF1 = EXPF(-ARGP*RHO) - EXPF(-ARGM*RHO)
            YAKR = .25*FKFAST*WF(M)*EXPF1*GMS/RHO
      VAKR = YAKR/GMS
      VUMSS = VUMSS + BS0(ICE,M)*BS0(ICE,M)*VAKR
      VUMIS = VUMIS + BI0(ICE,M)*BS0(ICE,M)*VAKR
      VUMII = VUMII + BI0(ICE,M)*BI0(ICE,M)*VAKR
      SUMSI = SUMSI - BS0(ICE,M)*BI1(ICE,M)*YAKR
      SUMIS = SUMIS + BI0(ICE,M)*BS1(ICE,M)*YAKR
      SUMSS = SUMSS + BS0(ICE,M)*BS1(ICE,M)*YAKR
  376 SUMII = SUMII - BI0(ICE,M)*BI1(ICE,M)*YAKR
      VUMSI = VUMIS
      SST(ICE) = SST(ICE-1) + SUMSS
      IIT(ICE) = IIT(ICE-1) + SUMII
      IST(ICE) = IST(ICE -1) + SUMIS
      SIT(ICE) = SIT(ICE-1) + SUMSI
      SSB(ICE) = SSB(ICE-1) + VUMSS
      IIB(ICE) = IIB(ICE-1) + VUMII
      ISB(ICE) = ISB(ICE -1) + VUMIS
      SIB(ICE) = SIB(ICE-1) + VUMSI
      FL = (ICE - 1)*FKFAST
      FLT = FL + FACT1
      ERFC = (FACT/FL)*(EXPF(-ARGP*FL) - EXPF(- ARGP*FLT))
      ERFC = ERFC/SIT(ICE)
      ERFC = ABS(ERFC)
      IF(ERFC.LE.EPS) GO TO 3
      IF(ICE.GE.10) GO TO 3
      GO TO 2
    3 SSI = ROUT*SSI $ III= RIN*III $ SII= RIN*SII $ ISI=ROUT*ISI
      SSR = ROUT*SST(ICE) $ IIR = RIN*IIT(ICE)
      SIR = RIN*SIT(ICE) $ ISR = ROUT*IST(ICE)
      SSIV = ROUT*SSIV $ IIIV = RIN*IIIV $ SIIV = RIN*SIIV
      ISIV = ROUT*ISIV
      SIRV = RIN*SIB(ICE) $ ISRV = ROUT*ISB(ICE)
      SSRV = ROUT*SSB(ICE) $ IIRV= RIN*IIB(ICE)
      SS = CMPLX(SSR,SSI) $ II = CMPLX(IIR,III)
      SSM(1,J)=SS
```

12/22/71

```
      ISM(1,J)=            CMPLX(ISR,ISI)
      SIM(1,J)=             CMPLX(SIR,SII)
      IIM(1,J)=II
      SSV(1,J)=           -CMPLX(SSRV,SSIV)
      IIV(1,J)=           -CMPLX(IIRV,IIIV)
      ISV(1,J)=           -CMPLX(ISRV,ISIV)
      SIV(1,J)=           -CMPLX(SIRV,SIIV)
      IF(J.GT.JMAXH) GO TO 66
      SSM(J,1)=SSM(1,J) $ SSV(J,1)=SSV(1,J) $ ISM(J,1)=ISM(1,J)
      SIM(J,1)=SIM(1,J) $ ISV(J,1)=ISV(1,J) $ IIM(J,1)=IIM(1,J)
      IIV(J,1)=IIV(1,J) $ SIV(J,1)=SIV(1,J)
   66 CONTINUE
      DO 12 J=1,JMAX
      JILT = JMAX-J+1
      DO 11 M=2,JILT
      IF(M.GT.JMAXH) GO TO 601
      SSM(M,M+J-1) = SSM(1,J)
      SSV(M,M+J-1) = SSV(1,J)
      ISM(M,M+J-1) = ISM(1,J)
      ISV(M,M+J-1) = ISV(1,J)
      IIM(M,M+J-1) = IIM(1,J)
      IIV(M,M+J-1) = IIV(1,J)
      SIM(M,M+J-1) = SIM(1,J)
      SIV(M,M+J-1) = SIV(1,J)
  601 CONTINUE
      IF(M+J-1.GT.JMAXH) GO TO 11
      SSM(M+J-1,M) = SSM(1,J)
      SSV(M+J-1,M) = SSV(1,J)
      ISM(M+J-1,M) = ISM(1,J)
      ISV(M+J-1,M) = ISV(1,J)
      IIM(M+J-1,M) = IIM(1,J)
      IIV(M+J-1,M) = IIV(1,J)
      SIM(M+J-1,M) = SIM(1,J)
      SIV(M+J-1,M) = SIV(1,J)
   11 CONTINUE
   12 CONTINUE
      END
```

12/22/71

```
      SUBROUTINE CSTM


      TYPE COMPLEX STV,ITV,STM,ITM
      COMMON/STV/STV/ITV/ITV/PJ/STM/ITM/ITM
      DIMENSION ITM(20,10),ITV(20,10),STM(20,10),STV(20,10)
      COMMON/BLKA/PR,PLR,PZ,PLZ/PIT/JMAX,JMAXH,IMAX
      DIMENSION PZ(20),PLZ(20),PR(20),PLR(20)
      DIMENSION BS0(10,32),BS1(10,32),QZ(32),BI0(10,32),BI1(10,32)
      DIMENSION XI(32),WF(32),SUMV(10)
      DIMENSION GS0(10,32),GS1(10,32),BUMT(10),SUMT(10),BUMV(10)
      TYPE COMPLEX ANS,ANSB,VNS,VNSB
      COMMON/FS/FKFK,FKFAST,TFK
      COMMON/DEL/DELR,DELZ/BOLD/GS0,GS1/BLK1/H,A,FK,PI/BLK2/XI,WF
      COMMON/5/BT0(10,10,32),BT1(11,10,32),GT0(10,10,32),GT1(11,10,32)
      COMMON/TOY/N,NN,NNN/RCC/JTOP/BLK3B/BI0,BI1/BLK3/BS0,BS1/TIDY/FAST
      DO 1000 JR=1,IMAX
      DO 1000 IR=1,JMAX
      R1 = A
      ZZ = PZ(IR) $ R2 = PR(JR)
      ARGP = H - ZZ
      HD = .5*DELR
      RL = R2 - HD
      RU = R2 + HD
      EPS = .000001
      SUMR=SUMI=SUMRB=SUMIB = VUMR=VUMI=VUMRB=VUMIB= 0.
      PFK = .5*FK
      DO 1 L =1,N                                                       00009400
      TRAP = PFK*XI(L)
      SIGMA = TRAP + PFK
      QZ(L)= FAST*(2.*TRAP - FK)
      QQ = SIGMA*SIGMA                                                  00009800
      D = SQRT(FKFK - QQ)                                               00009900
      SIGMAP = ARGP*SIGMA                                               00010000
      TRIGFI = SIN(SIGMAP) $ TRIGFR =  COS(SIGMAP)
      BS = BS0(1,L)    $ BI =  BI0(1,L)
      BU = RU*BT1(JR+1,1,L) - RL*BT1(JR,1,L)
      VAK = BS*BU*WF(L)/D    $ YAK = VAK*SIGMA
      VAKB = BI*BU*WF(L)/D $ YAKB = VAKB*SIGMA
      AI = YAK*TRIGFI    $  VR = VAK*TRIGFI
      AR = YAK*TRIGFR  $ VI = VAK*TRIGFR
      AIB = YAKB*TRIGFI  $ VRB = VAKB*TRIGFI
      ARB = YAKB*TRIGFR $ VIB = VAKB*TRIGFR
      VUMI = VUMI + VI $  VUMR = VUMR + VR
      SUMR = SUMR - AR
      SUMI = SUMI + AI                                                  00010800
      SUMRB =  SUMRB + ARB  $ SUMIB = SUMIB + AIB
      VUMRB =  VUMRB - VRB  $ VUMIB = VUMIB + VIB
    1 CONTINUE                                                          00010900
      TFK = .25*FK
      SUMI = SUMI*TFK
      SUMR = SUMR*TFK
      SUMRB = SUMRB*TFK       $   SUMIB = SUMIB*TFK
      VUMI = VUMI*TFK $ VUMR = VUMR*TFK
      VUMRB = VUMRB*TFK $ VUMIB = TFK*VUMIB
      R = R1
```

12/22/71

```
      IF(R2.LE.R1) R = R2
      IF(R.EQ.0.) R = DELR
      FACT = 2./(R*PI*ARGP)
      RM = R1
      IF(R.EQ.R1) RM = R2
      FACT1 = PI*.5/RM
      SUMT(1) = - SUMR
      BUMT(1) = - SUMRB
      SUMV(1) =   VUMR  $  BUMV(1) =   VUMRB
      ICE = 1
    2 ICE = ICE + 1
      VUMMIT = VLMMITB = 0.
      SUMMIT = SLMMITB = 0.
      DO 375 M=1,N                                                      00011300
      QZ(M) = QZ(M) + FKFAST
      RHO = QZ(M)                                                       00011500
      QQR = RHO*RHO                                                     00011600
      GMS = SQRT(QQR + FKFK)                                            00011700
      BS = BSO(ICE,M)   $  BI = BIO(ICE,M)
      BU = RU*BT1(JR-1,ICE,M)  -  RL*BT1(JR,ICE,M)
      VAKR = WF(M)*BS*BU/GMS $  YAKR = RHO*VAKR
      VAKRB = BI*BU*WF(M)/GMS  $ YAKRB = RHO*VAKRB
      EXPF1 = EXPF(-ARGP*RHO)
      SUMMIT = SLMMIT + YAKR*EXPF1
      SUMMITB = SUMMITB + YAKRB*EXPF1
      VUMMIT = VLMMIT +  VAKR*EXPF1
      VUMMITB = VUMMITB + VAKRB*EXPF1
  375 CONTINUE                                                          00012400
      BUMV(ICE) = BUMV(ICE-1) + .25*FKFAST*VUMMITB
      SUMV(ICE) = SUMV(ICE-1) + .25*FKFAST*VUMMIT
      SUMT(ICE) = SUMT(ICE-1) + .25*FKFAST*SUMMIT
      BUMT(ICE)  = BUMT(ICE-1) + .25*FKFAST*SUMMITB
      FL = (ICE-1)*FKFAST
      FLT = FL + FACT1
      ERFC = (FACT/FL)*(EXPF(-ARGP*FL) - EXPF(-ARGP*FLT))
      ERFC = RM*ERFC
      ERFC = ERFC/SUMT(ICE)
      IF(ERFC.LT.EPS) GO TO 3
      IF(ICE.GE.JTOP) GO TO 3
      GO TO 2
    3 ERFC = SUMT(ICE)*ERFC
      ANSI =-SUMI
      ANSR = SUMT(ICE)
      ANSIB = - SUMIB   $  ANSRB = BUMT(ICE)
      STM(IR,JR) = CMPLX(ANSR,ANSI)
      ITM(IR,JR) = CMPLX(ANSRB,ANSIB)
      VNS = CMPLX(-SUMV(ICE),VUMI) $ VNSB = CMPLX(-BUMV(ICE),VUMIB)
      STV(IR,JR) = VNS  $ ITV(IR,JR) = VNSB
 1000 CONTINUE
      END
```

12/22/71

```
      SUBROUTINE CTSM

      TYPE COMPLEX TIV,TSV,TIM,TSM,ANS,ANSB,VNS,VNSB
      DIMENSION TSM(10,20),TIM(10,20),TSV(10,20),TIV(10,20)
      DIMENSION PZ(20),PLZ(20),PR(20),PLR(20)
      COMMON/BLK1/H,A,FK,PI/RAD/RIN,ROUT/PIT/JMAX,JMAXH,IMAX/IJK/I,J
      COMMON/TIM/TIM/BLKD/TSM/TIV/TIV/TSV/TSV/BLKA/PR,PLR,PZ,PLZ
      DIMENSION BS0(10,32),BS1(10,32),QZ(32),EX(32)
      DIMENSION BI0(10,32),BI1(10,32),GI0(10,32),GI1(10,32)
      DIMENSION   SUMTV(10),BUMTV(10),XI(32),WF(32),SUMT(10),BUMT(10)
      DIMENSION GS0(10,32),GS1(10,32)
      COMMON/5/BT0(10,10,32),BT1(11,10,32),GT0(10,10,32),GT1(11,10,32)
      COMMON/BBOLD/GI0,GI1/DEL/DELR,DELZ/BOLD/GS0,GS1/RCC/JTOP
      COMMON/BLK3/BS0,BS1/TIDY/FAST/FS/FKFK,FKFAST,TFK
      COMMON/BLK3B/BI0,BI1/TOY/N,NN,NNN/BLK2/XI,WF
      DO 20 I=1,IMAX
      IR = I
      DO 20 J= 1,JMAX
      R = PR(I) $ Z = PLZ(J)
      EPS = .000001          $ EPS1 = .0001
      ZO = H
      ARGP = ZO - Z - DELZ
      ARGM = ZO - Z
      SUMR=SUMI=SUMRB=SUMIB=SUMRV=SUMIV=SUMRBV=SUMIBV=0.
      PFK = .5*FK
      DO 1 L =1,N                                                       00009400
      TRAP = PFK*XI(L)
      SIGMA = TRAP + PFK
      QZ(L)= FAST*(2.*TRAP - FK)
      QQ = SIGMA*SIGMA                                                  00009800
      D = SQRT(FKFK - QQ)                                               00009900
      SIGMAP = ARGP*SIGMA                                               00010000
      SIGMAM = SIGMA*ARGM                                               00010100
      IF(J.EQ.1) GO TO 90
      TRIGFI = SIN(SIGMAP) - SIN(SIGMAM)
      TRIGFR = COS(SIGMAP) - COS(SIGMAM)
      GO TO 91
   90 TRIGFR = 1. - COS(SIGMAM)
      TRIGFI = - SIN(SIGMAM)
   91 BS = BT0(IR,1,L)
      BSOS = WF(L)*BS/SIGMA
      YAK = BSOS*BS1(1,L)*D
      YAKB = - BSOS*BI1(1,L)*D
      SUMR = SUMR - YAK*TRIGFR
      SUMI = SUMI + YAK*TRIGFI
      SUMRB = SUMRB - YAKB*TRIGFR
      SUMIB = SUMIB + YAKB*TRIGFI
      IF(J.EQ.1) GO TO 1
      VAK = BSOS*BS0(1,L)
      VAKB= BSOS*BI0(1,L)
      SUMRV = SUMRV - VAK*TRIGFR
      SUMIV = SUMIV + VAK*TRIGFI
      SUMRBV = SUMRBV - VAKB*TRIGFR
      SUMIBV = SUMIBV + VAKB*TRIGFI
    1 CONTINUE                                                          00010900
```

12/22/71

```
      SUMI = SUMI*TFK $ SUMR = SUMR*TFK
      SUMIB = SUMIB*TFK $ SUMRB = SUMRB*TFK
      SUMIV =-SUMIV*TFK $ SUMRV = SUMRV*TFK
      SUMRBV = SUMRBV*TFK  $ SUMIBV =-SUMIBV*TFK
      SUMT(1) = SUMR $ BUMT(1) = SUMRB
      SUMTV(1) = SUMRV  $ BUMTV(1)  = SUMRBV
      FACT = 2./(R*PI*ARGP)
      FACT1 = PI*.5/A
      ANSTB = RIN*SUMRB
      ANST = ROUT*SUMR
      ICE = 1
    2 ICE = ICE + 1
      SUMMIT=SUMMITB=VUMMIT=VUMMITB=0.
      DO 375 M=1,N                                                      00011300
      QZ(M) = QZ(M) + FKFAST
      RHO = QZ(M)                                                       00011500
      QQR = RHO*RHO                                                     00011600
      GMS = SQRT(QQR + FKFK)                                            00011700
      BS = BTO(IR,ICE,M)
      EXPF1 = EXPF(-ARGP*RHO) - EXPF(-ARGM*RHO)
      GMR = BS*EXPF1*WF(M)/RHO
      ARTIST = GMR*GMS*BS1(ICE,M)
      ARTISTB= GMR*GMS*BI1(ICE,M)
      IF(J.EQ.1) GO TO 93
      ARTISTV = GMR*BSO(ICE,M)
      VRTISTB = GMR*BIO(ICE,M)
      VUMMIT = VUMMIT + ARTISTV
      VUMMITB = VUMMITB + VRTISTB
      GO TO 94
   93 GS = GTO(IR,ICE,M)*WF(M)
      GSGS = GS*GS1(ICE,M)
      GSGSI = GS*GI1(ICE,M)
      ARTIST = ARTIST + GSGS $ ARTISTB = ARTISTB + GSGSI
   94 SUMMIT = SUMMIT + ARTIST
      SUMMITB= SUMMITB + ARTISTB
  375 CONTINUE                                                          00012400
      BUMT(ICE) = BUMT(ICE-1) + .25*FKFAST*SUMMITB
      SUMT(ICE) = SUMT(ICE-1) + .25*FKFAST*SUMMIT
      IF(J.EQ.1) GO TO 95
      SUMTV(ICE) = SUMTV(ICE-1) + .25*FKFAST*VUMMIT
      BUMTV(ICE) = BUMTV(ICE-1)   + .25*FKFAST*VUMMITB
      FL = (ICE-1)*FKFAST
      FLT = FL + FACT1
      ERFC = (FACT/FL)*(EXPF(-ARGP*FL) - EXPF(-ARGP*FLT))
      ERFC = ERFC/SUMT(ICE)
      IF(ERFC.LT.EPS) GO TO 3
      IF(ICE.GE.NTOP) GO TO 3
      GO TO 2
   95 ANSR = ROUT*SUMT(ICE) + .50
      ANSRB = RIN*BUMT(ICE)
      TESTER = 1. - ANST/ANSR
      TESTER = ABS(TESTER)
      TESTERB = 1. - ANSTB/ANSRB
      TESTERB =  ABS(TESTERB)
      IF(TESTER.LT.EPS1.AND.TESTERB.LT.EPS1) GO TO 3
      IF(ICE.GE.NTOP) GO TO 3
```

12/22/71

```
      ANST = ANSR
      ANSTB = ANSRB
      GO TO 2
    3 CONTINUE
      ANS = CMPLX(SUMT(ICE),SUMI)
      ANSB = CMPLX(SUMT(ICE),SUMIB)
      VNS =CMPLX(-SUMTV(ICE),SUMIV)
      VNSB =CMPLX(-SUMTV(ICE),SUMIBV)
      TSM(I,J) = ANS*ROUT  $ TIM(I,J) = RIN*ANSB
      TSV(I,J) = VNS*ROUT  $ TIV(I,J) = RIN*VNSB
      IF(J.EQ.1) TSM(I,J) = TSM(I,J) + 0.500
   20 CONTINUE
      EPS = .0001
      ARGP =  DELZ
      DO 10 L=1,N
      TRAP =PFK*XI(L)
      SIGMA = TRAP + PFK
      QZ(L)= FAST*(2.*TRAP - FK)
      QQ = SIGMA*SIGMA                                                  00009800
      EX(L) = EXPF(-ARGP*QZ(L))
      SIGMAP = ARGP*SIGMA                                               00010000
      TRIGFI = SIN(SIGMAP)
      TRIGFR = COS(SIGMAP) - 1.
      GSO(1,L) = TRIGFI   $  GS1(1,L) = TRIGFR
   10 CONTINUE
      EXFACT = EXPF(-ARGP*FKFAST)
      DO 11 ICE= 2,JTOP
      DO 11 M=1,N
      EX(M) = EX(M)*EXFACT
      EXPF1 = EX(M) - 1.
   11 GS1(ICE,M) = EXPF1
      TPK = .25*FKFAST
      DO 99 J=1,IMAX
      SUMRB=SUMIB=SUMR=SUMI=0.
      IMP=IR=J
      R = PR(J)
      DO 61 L=1,N
      TRAP = PFK*XI(L)
      SIGMA = TRAP + PFK
      QZ(L)= FAST*(2.*TRAP - FK)
      QQ = SIGMA*SIGMA
      D = SQRT(FKFK - QQ)
      TRIGFI = GSO(1,L)   $ TRIGFR = GS1(1,L)
      YAM = TFK*WF(L)/SIGMA
      YAKI = YAM*TRIGFI
      YAKR = - YAM*TRIGFR
      VTAB = BSO(1,L)*BTO(IMP,1,L)*ROUT
      VTABI = BIO(1,L)*BTO(IMP,1,L)*RIN
      SUMI = SUMI + YAKI*VTAB
      SUMR = SUMR + YAKR*VTAB
      SUMIB = SUMIB + YAKI*VTABI
   61 SUMRB = SUMRB + VTABI*YAKR
      SUMT(1) = SUMR $ BUMT(1) = SUMRB
      ANST=ANSTB= 0.
      DO 501 ICB=2,JTOP
      SUMMIT=SUMMITB = 0.
```

12/22/71

```
      DO 376 M=1,N
      QZ(M) = QZ(M) + FKFAST
      RHO = QZ(M)
      QOR = RHO*RHO
      QHS = SQRT(QOR + FKFK)
      EXPF1 = GS1(ICE,M)
      YAKR = EXPF1*TPK*WF(M)/RHO
      VTABI = BIO(ICE,M)*BTO(IMP,ICE,M)*RIN
      VTAB = BSO(ICE,M)*BTO(IMP,ICE,M)*ROUT
      ARTIST = YAKR*VTAB $ ARTISTB = YAKR*VTABI
      SUMMITB = SUMMITB + ARTISTB
  376 SUMMIT = SUMMIT + ARTIST
      SUMT(ICE) = SUMT(ICE - 1)  +  SUMMIT
      BUMT(ICE) = BUMT(ICE - 1) + SUMMITB
      FLAM = FKFAST*(ICE - 1)
      B = ROUT
      RB = ABS(R-B)
      ARG1 = RB*FLAM
      ARG2 = (R+B)*FLAM
      CF1 = .5/(PI*SQRT(R*B))
      CF3 =CF1*ROUT*(COS(ARG1)/FLAM + SIN(ARG2)/FLAM
     1 + RB*SI(ARG1)  - (R+B)*CI(ARG2) )
      B= RIN
      RB = ABS(R-B)
      ARG1 = RB*FLAM
      ARG2 = (R+B)*FLAM
      CF1 = .5/(PI*SQRT(R*B))
      CF2 = CF1*RIN*(COS(ARG1)/FLAM + SIN(ARG2)/FLAM
     1 + RB*SI(ARG1)  - (R+B)*CI(ARG2) )
      ANSR = SUMT(ICE) - CF3
      ANSRB = BUMT(ICE) - CF2
      TESTER = 1. - ANST/ANSR
      TESTER = ABS(TESTER)
      TESTERB = 1. - ANSTB/ANSRB
      TESTERB = ABS(TESTERB)
      IF(TESTER.LE.EPS.AND.TESTERB.LE.EPS) GO TO 934
      ANSTB = ANSRB
      ANST = ANSR
  501 CONTINUE
  934 CONTINUE
      TSV(J,1)= CMPLX(ANSR,SUMI) $ TIV(J,1) = CMPLX(ANSRB,SUMIB)
   99 CONTINUE
      END
```

12/22/71

```
      SUBROUTINE GQC

      TYPE DOUBLE W,X                                                   00000200
      COMMON/BLK1/H,A,FK,PI/TOY/NQD1,LL,MM/BLK2/XI,WF
      DIMENSION XI(32),WF(32),W(31),X(31)
      DATA (PI = 3.141592653589)                                        00000500
      DATA(X = 0.148874338981631D000, 0.433395394129247D000,
     * 0.679409568299024D000,0.865063366688985D000,
     *0.973906528517172D000,
     *       0.076526521133497333755D000,0.227785851141645078080D000,
     * 0.373706088715419560673D000,0.510867001950827098004D000,
     * 0.636053680726515025453D000,0.746331906460150792614D000,
     * 0.839116971822218823395D000, 0.912234428251325905868D000,
     * 0.963971927277913791268D000,0.993128599185094924786D000,
     *        0.997263861849481563540D000, 0.985611511545268335400D000,
     *0.964762255587506430770D000, 0.934906075937739689170D000,          00000700
     * 0.896321155766052123960D000, 0.849367613732569970130D000,         00000800
     * 0.794483795967942406960D000, 0.732182118740289680380D000,         00000900
     * 0.663044266930215200970D000, 0.587715757240762329040D000,         00001000
     * 0.506899908932229390020D000, 0.421351276130635345360D000,         00001100
     *0.331868602282127649770D000, 0.239287362252137074540D000,          00001200
     * 0.144471961582796493480D000, 0.483076656877383162340D-001)        00001300
      DATA(W= 0.295524224714753D000, 0.269266719309996D000,
     *0.219086362515982D000, 0.149451349150581D000,
     *0.066671344308688D000,
     *       0.152753387130725850698D000,0.149172986472603746788D000,
     * 0.142096109318382051329D000,0.131688638449176626898D000,
     * 0.118194531961518417312D000, 0.101930119817240435037D000,
     * 0.083276741576704748725D000,0.062672048334109063570D000,
     *0.040601429800386941331D000, 0.017614007139152118312D000,
     *        0.701861000947009660040D-002, 0.162743947309056706050D-001, 00001400
     * 0.253920653092620594550D-001, 0.342738629130214331020D-001,       00001500
     * 0.428358980222266806560D-001, 0.509980592623761762960D-001,       00001600
     *0.586840934785355471450D-001, 0.658222227763618468370D-001,        00001700
     * 0.723457941088485062250D-001, 0.781938957870703064710D-001,       00001800
     * 0.833119242269467552220D-001, 0.876520930044038111420D-001,       00001900
     * 0.911738786957638847120D-001, 0.938443990808045656390D-001,       00002000
     * 0.956387200792748594190D-001, 0.965400885147278005660D-001)       00002100
      NNL = NQD1/2   $ NBL = 0
      IF(NQD1.EQ.20) NBL = 5
      IF(NQD1.EQ.32) NBL = 15
      DO 111 I=1,NNL                                                    00002400
      K = I + NBL
      XI(I) = X(K)
      WF(I) = W(K)
      XI(I + NNL) = -X(K)
  111 WF(I + NNL)= W(K)
      RETURN
      ENTRY TIME
      TIMER = TIMELEFT(K)
      PRINT 1, TIMER
    1 FORMAT(* TIME LEFT *F10.3* SECONDS*//)
      RETURN
      END
```

12/22/71

```
      SUBROUTINE CALBES

      DIMENSION PZ(20),PLZ(20),PR(20),PLR(20),XI(32),WF(32)
      DIMENSION BI0(10,32),BI1(10,32),GI0(10,32),GI1(10,32)
      DIMENSION GS0(10,32),GS1(10,32),BS0(10,32),BS1(10,32),QZ(32)
      COMMON/BLKA/PR,PLR,PZ,PLZ/PIT/JMAX,JMAXH,IMAX/RCC/JTOP
      COMMON/5/BT0(10,10,32),BT1(11,10,32),GT0(10,10,32),GT1(11,10,32)
      COMMON/TOY/NQD1,NQD2,NQD3/BOLD/GS0,GS1/BLK1/H,A,FK,PI/BLK2/XI,WF
      COMMON/BLK3B/BI0,BI1/BBOLD/GI0,GI1/TIDY/FAST/BLK3/BS0,BS1
      COMMON/RAD/RIN,ROUT
      N=NQD1
      FKFK = FK*FK                                                      00015100
      FKFAST = 2.*FAST*FK
      DO 9 L=1,N
      TRAP = FK*.5*XI(L)                                                00015300
      SIGMA = TRAP  + .5*FK                                             00015400
      QZ(L)= FAST*(2.*TRAP + FK)
      QQ = SIGMA*SIGMA                                                  00015600
      D = SQRT(FKFK - QQ)                                               00015700
      T=D*ROUT $ TI = D*RIN
      CALL BESL(T,BSJ0,BSJ1,Y0,YU)                                      00016000
      CALL BESL(TI,BIJ0,BIJ1,Y0,YU)
      BI0(1,L) = BIJ0 $ BI1(1,L) = BIJ1
      BS0(1,L) = BSJ0                                                   00016400
      BS1(1,L) = BSJ1                                                   00016500
    9 CONTINUE
      DO15 ICE=2,JTOP
      DO16 M=1,N
      QZ(M) = QZ(M) + FKFAST
      RHO = QZ(M)                                                       00017000
      QQR = RHO*RHO                                                     00017100
      GMS = SQRT(QQR + FKFK)                                            00017200
      TM = GMS*ROUT $ SM = RHO*ROUT
      TIM = GMS*RIN $ SIM = RHO*RIN
      CALL BESL(SM,GST0,GST1,Y1,Y2)                                     00017700
      CALL BESL(TM,BST0,BST1,YZ,YJ)                                     00017800
      CALL BESL(SIM,GIT0,GIT1,YI,Y0)
      CALL BESL(TIM,BIT0,BIT1,YK,Y0)
      BI0(ICE,M) = BIT0 $ BI1(ICE,M) = BIT1 $ GI0(ICE,M) = GIT0
      GI1(ICE,M) = GIT1
      BS0(ICE,M) = BST0                                                 00018300
      BS1(ICE,M) = BST1                                                 00018400
      GS0(ICE,M) = GST0                                                 00018500
      GS1(ICE,M) = GST1                                                 00018600
   16 CONTINUE
   15 CONTINUE
      DO 8 ICE=1,JTOP
      DO 8 M=1,N
      GT1(IMAX+1,ICE,M) = GS1(ICE,M)
      BT1(IMAX+1,ICE,M) = BS1(ICE,M)
      BT1(1,ICE,M) = BI1(ICE,M)
      GT1(1,ICE,M) = GI1(ICE,M)
    8 CONTINUE
      DELR = (ROUT - RIN)/IMAX
      DELR = .5*DELR
```

12/22/71

```
      DO 5 IMP=1,IMAX
      DO 1 L=1,N
      TRAP = FK*.5*XI(L)                                                00015300
      SIGMA = TRAP  + .5*FK                                             00015400
      QZ(L)= FAST*(2.*TRAP - FK)
      QQ = SIGMA*SIGMA                                                  00015600
      D = SQRT(FKFK - QQ)                                               00015700
      T= D*PR(IMP)
      T1= D*(PR(IMP) - DELR)
      CALL BESO(T,BTO(IMP,1,L))
      IF(IMP.EQ.1) GO TO 1
      CALL BES1(T1,BT1(IMP,1,L))
    1 CONTINUE                                                          00016600
      DO 5 ICE=2,JTOP                                                   00016700
      DO 6 M=1,N
      QZ(M) = QZ(M) + FKFAST
      RHO = QZ(M)                                                       00017000
      QQR = RHO*RHO                                                     00017100
      GMS = SQRT(QQR + FKFK)                                            00017200
      SM = RHO*PR(IMP) $  SM1 = RHO*(PR(IMP) - DELR)
      TM = GMS*PR(IMP) $ TM1 = GMS*(PR(IMP) - DELR)
      CALL BESO(TM,BTO(IMP,ICE,M))
      CALL BESO(SM,GTO(IMP,ICE,M))
      IF(IMP.EQ.1) GO TO 6
      CALL BES1(TM1,BT1(IMP,ICE,M))
      CALL BES1(SM1,GT1(IMP,ICE,M))
    6 CONTINUE                                                          00018700
    5 CONTINUE                                                          00018800
      RETURN                                                            00018900
      END
```

12/22/71

```
      SUBROUTINE FARFLD(NPTS)

      DIMENSION PTOP(10),PINS(20),POUT(20),ANS(60)
      COMMON/LC/LCMAX,LCMAXH/DIST/DIST/RAD/RIN,ROUT/ANS/ANS
      COMMON/BLK1/H,A,FK,PI/DEL/DELR,DELZ/BLKA/PR,PLR,PZ,PLZ
      COMMON/PIT/JMAX,JMAXH,IMAX/VELO/VEL/TOY/NOD1,ISYM,ICOR
      TYPE COMPLEX FFI,FFT,FFTB,RHOC,ADDIT1,SUM1,FFVT,FFVS,BJI,BJOUT,CEP
     1,CEM,PTOP,PINS,POUT,ANS,I,SUM,ADDIT,ARGI,VEL,PBOT,VINS,VOUT,VTOP,
     2 VBOT
      DIMENSION VEL(60),PBOT(10),VINS(20),VOUT(20),VTOP(10),VBOT(10)
      DIMENSION PZ(20),PLZ(20),PR(20),PLR(20)
      DATA(ALICE= 8,685889638)
      PRINT 40
   40 FORMAT(1H1*          FARFIELD PATTERN  */7X*                ANGLE
     *            DB  */)
      I= (0,0,1,0)     $ RHOC = 1500000,*I
      DD = 1,/(2,*NPTS) $ DELTH = DD*PI $ DELTHD = 180,*DD
      NPT = NPTS-1    $ JMAXT = JMAXH
      IF(RIN.EQ.0,)        JMAXH = 0
      DO 2 J=1,IMAX
      PBOT(J) = ANS(LCMAX - JMAXH + 1 - J)
      VBOT(J) = VEL(LCMAX - JMAXH + 1 - J)
      VTOP(J) = VEL(JMAXH + J)
    2 PTOP(J) = ANS(JMAXH+J)
      DO 3 J=1,JMAXT
      POUT(J)= ANS(IMAX+JMAXH+J)
      VOUT(J)= VEL(IMAX+JMAXH+J)
      VOUT(JMAX +1-J) = VEL( LCMAX +1 -JMAXH-IMAX - J)
    3 POUT(JMAX +1-J) = ANS( LCMAX +1 -JMAXH-IMAX - J)
      JMAXH = JMAXT
      IF(RIN.EQ.0,) GO TO 966
      DO 965 J=1,JMAXH
      PINS(J) = ANS(JMAXH+1-J)
      VINS(J) = VEL(JMAXH+1-J)
      VINS(JMAX+1 - J) = VEL(LCMAX -JMAXH + J)
  965 PINS(JMAX+1 - J) = ANS(LCMAX -JMAXH + J)
      GO TO 970
  966 DO 969 J=1,JMAX
      VINS(J) = (0,0,0,0)
  969 PINS(J) = (0,0,0,0)
  970 FFI= ,5*FK*DELZ
      RO = FK*ROUT $ RI = FK*RIN
      CALL BESL(RO,BSO,BJO,YO,Y1) $ CALL BESL(RI,BSI,BJ1,Y1,YO)
      BO = BSO*ROUT    $  BI = BSI*RIN
      BJI = RIN*BJ1 $ BJOUT = ROUT*BJO
      IF(ICOR.EQ.0) GO TO 202
      BSI = RIN*RIN*BSI/DIST $ BSO = ROUT*ROUT*BSO/DIST
      GO TO 203
  202 BSI=BSO=0,
  203 BJOUT = BJOUT + I*BSO $ BJI = BJI + I*BSI
      SUM = (0,0,0,0)
      DO 37 K=1,JMAX
      ADDIT = (BJOUT*POUT(K) - BJI*PINS(K))
      ADDIT1= RHOC*(VINS(K)*BI + VOUT(K)*BO)
      ADDIT = ADDIT - ADDIT1
```

12/22/71

```
 37 SUM = SUM + ADDIT
    FFI = FFI*SUM          $ FFVT = (0.0,0.0)
    DO 38 K=1,IMAX
    RL = PR(K) - .5*DELR   $   RU = RL + DELR
    TU = FK*RU  $  TL = FK*RL
    CALL BES1(TU,BU) $  CALL BES1(TL,BL)
 38 FFVT = FFVT +          (VTOP(K) + VBOT(K))*(RU*BU - RL*BL)
    FFO = CABS(FFI + FFVT*.5*RHOC)
902 DEG = 0.0 $ FFOM = 0.0
    PRINT 33, DEG,FFOM
    DO 1 J=1,NPT
    THETA = J*DELTH
    COSTH = COS(THETA) $ SINTH = SIN(THETA)
    ARGI = -I*FK*SINTH
    TT = RI*COSTH
    CALL BESL(TT,BS1,BJ1,Y0,Y1)
    BI = RIN*BS1
    TT = RO*COSTH
    CALL BESL(TT,BS2,BJ2,Y0,Y1)
    BJI= RIN*BJ1   $ BJOUT = ROUT*BJ2  $ BO = ROUT*BS2
    IF(ICOR.EQ.0)     GO TO 205
    BSI = RIN*RIN*BS1/DIST  $  BSO = ROUT*ROUT*BS2/DIST
    BJOUT = BJOUT*COSTH + I*BSO
    BJI = BJI*COSTH + I*BSI
    GO TO 207
205 BJOUT = BJOUT*COSTH
    BJI = BJI*COSTH
207 FFI = .5*I*(CEXP(ARGI*DELZ) - 1.)/SINTH
    SUM = (0.0,0.0)
    DO 4 K=1,JMAX
    CEM = CEXP(ARGI*PLZ(K))
    ADDIT = (BJOUT*ROUT(K) - BJI*PINS(K))*CEM
    ADDIT1 = (VOUT(K)*BO + VINS(K)*BI)*RHOC*CEM
  4 SUM = SUM + ADDIT +ADDIT1
    FFI= FFI*SUM
    SUM = (0.0,0.0)
    FFTB = .5/COSTH
    ARGI = I*FK*H*SINTH
    CEM = CEXP(-ARGI)
    CEP = CONJG(CEM)
    DO 5 K=1,IMAX
    RU = PR(K) + .5*DELR  $ RL = PR(K) - .5*DELR
    TT = FK*RU*COSTH   $ TTL = FK*RL*COSTH
    CALL BES1(TT,BU) $ CALL BES1(TTL,BL)
    RUBL = RU*BU - RL*BL
    ADDIT = -I*SINTH*(PBOT(K)*CEP - PTOP(K)*CEM)
    ADDIT1 = RHOC*(VTOP(K)*CEM + VBOT(K)*CEP    )
  5 SUM = SUM + (ADDIT + ADDIT1)*RUBL
    FFTB = SUM*FFTB
    FFM = CABS(FFI + FFTB)
500 FORMAT(2X,4C(E10.2,E10.2)/)
    DEG = J*DELTHD
    FFM = ALICE*LOGF(FFM/FFO)
  1 PRINT 33, DEG,FFM
 33 FORMAT(5X,F20.3,F20.3/)
    IF(DEG.GE.0..AND.ISYM.NE.1) GO TO 900
```

12/22/71

```
      GO TO 901
900   DELTH = -DELTH $ DELTHD = -DELTHD
      PRINT 905
905   FORMAT(2X///)
      GO TO 902
901   CONTINUE
      END
```

12/22/71

```
      SUBROUTINE SIMX( MAT,MCT,RHS,ANS)

      DIMENSION OHOLD(60),MAT(60,60),RHS(30),ANS(MCT)
      TYPE COMPLEX CC,CC2,OHOLD,RHS
      DIMENSION C(2),CX(2),CX2(2)
      TYPE COMPLEX MAT,ORIG,ANS,B0,B2,B4,B6,B8,B10,B11,B13,B15                020
      TYPE INTEGER V
      EQUIVALENCE  (B2,C),(CC,CX),(CC2,CX2)
      DO 100 J=1,MCT
      OHOLD(J) =  MAT(J,MCT+1)
100    MAT(J,MCT + 1) = - RHS(J)
 10   FORMAT(1X,C(E17.10,E17.10))                                            040
 15   FORMAT(25H  THIS MATRIX IS SINGULAR/)                                  050
 28   FORMAT(1H1)                                                            091
      NCT=MCT+1
      JSING=JFIN*MCT                                                         100
      IX=0$ B4=(0.0,0.0)$ B11=(1.0,0.0)                                      120
      JCT=MCT-1                                                              159
      DO 3 J=1,JCT                                                           160
      KK=J+1                                                                 170
      GOTO 25                                                                180
 24   DO 4 K=KK,MCT                                                          190
      B8=MAT(K,J)/MAT(J,J)                                                   200
      DO 5 L=J,NCT                                                           210
      B10=B8*MAT(J,L)                                                        220
  5   MAT(K,L)=MAT(K,L)-B10                                                  230
  4   CONTINUE                                                               240
  3   B11=B11*MAT(J,J)                                                       250
      B11=B11*MAT(MCT,MCT)                                              00000255
      LOW=-MCT$ MO=-1$                                                       260
      DO 6 INM=LOW,MO                                                        261
      M=IABS(INM)                                                            262
      B0=-MAT(M,NCT)                                                         270
      B2=MAT(M,M)                                                            280
      B4=(0.0,0.0)                                                           290
      IF(IX) 7,22,7                                                          300
 22   IX=IX+1                                                                310
      GOTO 8                                                                 320
  7   MO2=-JFIN                                                              330
      DO 9 INN=LOW,MO2                                                       331
      N=IABS(INN)                                                            332
  9   B4=B4+MAT(M,N)*ANS(N)                                                  340
      B0=B0-B4$ JFIN=JFIN-1                                                  350
  8   IF( C(1).EQ.0.).AND.C(2).EQ.0.)  13,29                                 360
 29   ANS(M)=B0/B2                                                           361
  6   CONTINUE                                                               370
      GO TO 27
 25   V=J                                                                    410
      CC=MAT(J,J)                                                            415
      IF( CX(1).EQ.0. .AND.CX(2).EQ.0.) 11,12                                420
 11   IF(V.EQ.JSING)13,14                                                    430
 13   PRINT28$ PRINT15                                                       440
      RETURN                                                                 450
 14   V=V+1                                                                  460
      CC2=MAT(V,J)                                                           465
```

12/22/71

```
      IF(CX2(1).EQ.0.,AND.CX2(2).EQ.0.) 11,16
   16 DO 17 JJ=J,NCT
      B6=MAT(J,JJ)
      MAT(J,JJ)=MAT(V,JJ)
   17 MAT(V,JJ)=B6
      B11=-B11
   12 JSING=JSING-1
      GOTO 24
   27 DO 200 J=1,NCT
  200  MAT(J,NCT+1) = OHOLD(J)
      END
```

12/22/71

```
      SUBROUTINE BES1(X,BJ1)


      IF(X.GT.3.)GO TO 1
      XT = (X*X)/9.
      BJ1= X*(.9 +XT*(-.56249985 + XT*(.21093573 + XT*(-.03954289 +
     1 XT*(.00443319 + XT*(-.00031761 + XT*.00001109))))))
      RETURN
    1 XT = 3./X
      F1 =  .79788456 +XT*(.00000156 + XT*(.01659667 + XT*(.00017105 +
     1  XT*(- .00249511 + XT*(.00113653 - .00020033*XT)))))
      T1= X - 2.35619449 + XT*(.12499612 + XT*(.00005650 +XT*(-.00637879
     1    + XT*(.00074348 + XT*(.00079824 -.00029166*XT)))))
      SX = SQRT(X)
      SX = 1./SX
      BJ1 = SX*F1*COS(T1)
      RETURN
      ENTRY BES0
      IF(X.GT.3.) GO TO 5
      XT = (X*X)/9.
      BJ1=(1. + XT*(-2.2499997 + XT*(1.2656208 + XT*( -.3163866 +
     1 XT*(.0444479 + XT*(- .0039444 + XT*.0002100))))))
      RETURN
    5 XT = 3./X
      F0 = .79788456 +XT*(-.00000077 + XT*(-.00552740 + XT*(-.00009512
     1 + XT*(.00137237 + XT*(-.00072805 + XT*.00014476)))))
      T0 = X-.78539816 + XT*(-.04166397 + XT*(-.00003954 + XT*(.00262573
     1  + XT*(-.00054125 + XT*(-.00029333 + .00013558*XT)))))
      SX = SQRT(X)
      SX = 1./SX
      BJ1 = SX*F0*COS(T0)
      RETURN
      END
```

12/22/71

```
      SUBROUTINE BESL(X,BJ0,BJ1,Y0,Y1)

      IF(X.GT.3.)GO TO 1
      XT = (X*X)/9.
      BJ0=(1. + XT*(-2.2499997 + XT*(1.2656208 + XT*( -.3163866 +
     1 XT*(.0444479 + XT*(- .0039444 + XT*.0002100))))))
      BJ1= X*(.5 +XT*(-.56249985 + XT*(.21093573 + XT*(-.03954289 +
     1 XT*(.00443319 + XT*(-.00031761 + XT*.00001109))))))
      RETURN
    1 XT = 3./X
      F0 = .79788456 +XT*(-.00000077 + XT*(-.00552740 + XT*(-.00009512
     1 + XT*(.00137237 + XT*(-.00072805 + XT*.00014476)))))
      F1 =  .79788456 +XT*(.00000156 + XT*(.01659667 + XT*(.00017105 +
     1  XT*(- .00249511 + XT*(.00113653 - .00020033*XT)))))
      T0 = X-.78539816 + XT*(-.04166397 + XT*(-.00003954 + XT*(.00262573
     1  + XT*(-.00054125 + XT*(-.00029333 + .00013558*XT)))))
      T1= X - 2.35619449 + XT*(.12499612 + XT*(.00005650 +XT*(-.00637879
     1   + XT*(.00074348 + XT*(.00079824 -.00029166*XT)))))
      SX = SQRT(X)
      SX = 1./SX
      BJ0 = SX*F0*COS(T0)
      BJ1 = SX*F1*COS(T1)
      RETURN
      END
```

12/22/71

```
      FUNCTION FN(X)

      DIMENSION AK(4),BK(4),AE(4),BE(4),A(4),B(4)
      DIMENSION FA(4),FB(4),GA(4),GB(4)
      DATA (FA=38,027264,265,187033,335,677320,38,102495)
      DATA (FB=40,021433,322,624911,570,236280,157,105423)
      DATA (GA = 42,242855,302,757865,352,018498,21,821899)
      DATA (GB=48,196927,482,485984,1114,978885,449,690326)
      DATA(F2=0,250000000),(F4=0,010416667),(F6=0,000231481)
      DATA(F8=0,000003100),(F10=0,000000028),(F3=0,055555556)
      DATA(F5=0,001666667),(F7=0,000028345),(F9=0,000000306),(F11=
     1 ,000000002)
      DATA (AK= 0,09666344259,0,03590092383,0,03742563713,0,01451196212)
      DATA(BK=0,12498593597,0,06880248576,0,03328355346,0,00441787012)
      DATA(AE=0,44325141463,0,06260601220,0,04757383546,0,01736506451)
      DATA(BE=0,24998368310,0,09200180037,0,04069967526,0,00526449639)
         ENTRY CI
      IFLAG = 1
      IF(X.LT.1.) GO TO 1
   27 XX=X*X
      FDEN = FB(4) + XX*(FB(3) + XX*(FB(2) + XX*(FB(1) + XX)))
      FNUM = FA(4) + XX*(FA(3) + XX*(FA(2) + XX*(FA(1) + XX)))
      F = FNUM/(X*FDEN)
      GNUM = GA(4) + XX*(GA(3) + XX*(GA(2) + XX*(GA(1) + XX)))
      GDEN = GB(4) + XX*(GB(3) + XX*(GB(2) + XX*(GB(1) + XX)))
      G = GNUM/(XX*GDEN)
      IF(IFLAG.EQ.0) GO TO 28
      FN = F*SIN(X) - G*COS(X)
      RETURN
    1 XX = X*X
      CH = XX*(F2 + XX*(F4 + XX*(F6 + XX*(F8 + XX*(F10 )))))
      GAMMA = .577215664
      FN = GAMMA + LOGF(X) + CH
      RETURN
      ENTRY SI
      IFLAG = 0
      IF(X.LT.1.) GO TO 29
      GO TO 27
   28 FN = - F*COS(X) - G*SIN(X)
      RETURN
   29 XX = X*X
      PI = 3.1415926536
      SH = X*(1. + XX*(F3 + XX*(F5 + XX*(F7 + XX*(F9  + XX*F11)))))
      FN = SH - .5*PI
      RETURN
      ENTRY ELLIPK
      HOLDA=1.38629436112 $ HOLDB = 0.5
      DO 2 J=1,4
      A(J) = AK(J)
    2 B(J) = BK(J)
      GO TO 4
      ENTRY ELLIPE
      HOLDA = 1.0
      HOLDB = 0.0
      DO 3 J=1,4
```

12/22/71

```
      A(J) = AE(J)
    3 B(J) = BE(J)
    4 Q = QM = 1. - X
      FACT = LOGF(1./Q)
      DO 11 J=1,4
      HOLDA = HOLDA + A(J)*Q
      HOLDB = HOLDB + B(J)*Q
   11 Q = Q*QM
      FN = HOLDA + HOLDB*FACT
      END
```

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Research Laboratory Washington, D.C. 20390	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

SHIP (SIMPLIFIED-HELMHOLTZ-INTEGRAL PROGRAM): A FAST COMPUTER PROGRAM FOR CALCULATING THE ACOUSTIC RADIATION AND RADIATION IMPEDANCE FOR FREE-FLOODED-RING AND FINITE-CIRCULAR-CYLINDER SOURCES

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

An interim report on a continuing NRL Problem.

5. AUTHOR(S) *(First name, middle initial, last name)*

Peter H. Rogers

6. REPORT DATE June 19, 1972	7a. TOTAL NO. OF PAGES 64	7b. NO. OF REFS 10
8a. CONTRACT OR GRANT NO. NRL Problem S02-19 b. PROJECT NO. RF 05-121-402-6200 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) NRL Report 7240 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Office of Naval Research Department of the Navy Arlington, Va. 22217

13. ABSTRACT

The Naval Research Laboratory SHIP is a special-purpose acoustic radiation program designed for rapidly determining acoustic surface pressures, radiation impedances, and far-field radiation patterns for acoustic sources in the form of finite cylinders and free-flooded rings and for acoustic sources which can be approximated by such forms. The program uses a finite-element method based on a one-dimensional version of the surface Helmholtz integral equation. Axial symmetry is assumed in all cases. It is the complete use of this axial symmetry which results in the high speed of this method relative to other formulations of this problem. The basic inputs to the program are the geometry of the ring or cylinder and the normal velocity on the surface of this source. The output is a representation of the complex acoustic pressure distribution on the surface of the source, the radiation impedance, and (if desired) the far-field radiation pattern. The program is presently limited to a single ring or single cylinder, but it may easily be modified to handle coaxial stacks of rings and cylinders.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

Security Classification

Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Acoustic radiation						
Computer programs						
Simplified-Helmholtz-Integral Program						
Underwater sound sources						
Sound transducers						
Free-flooded rings						
Circular cylinders						
Impedance (acoustical)						
Acoustic fields						
Far-field patterns						
Acoustic surface pressures						
Integral equations						
Helmholtz integral equations						

DD FORM 1 NOV 65 1473 (BACK)
(PAGE 2)

Security Classification