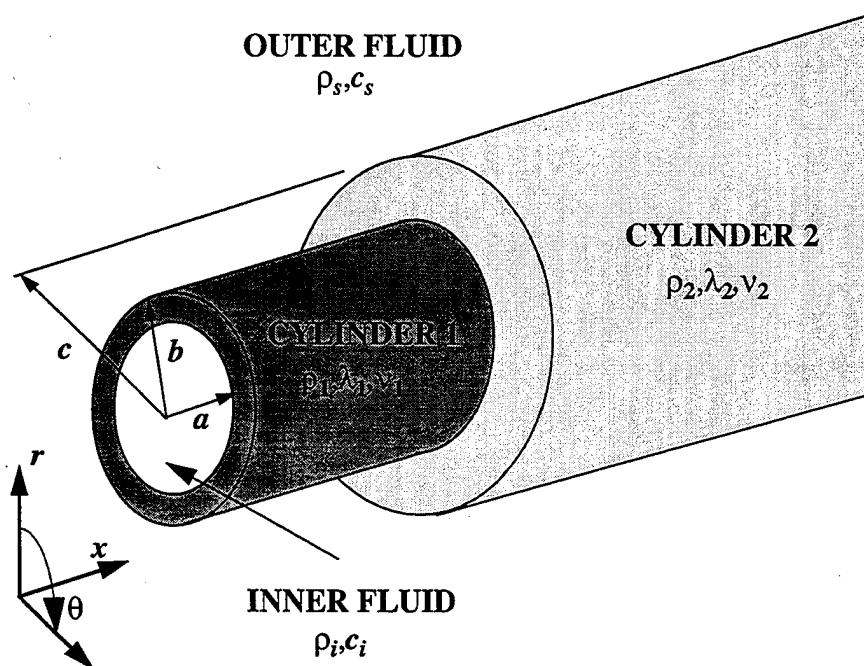


FORTRAN Algorithms for the Three-Dimensional Solution of Two-Layer Solid and Hollow Cylinder Dynamic Elasticity Problems With and Without Fluids

Mark S. Peloquin
Submarine Sonar Department



19960304 100

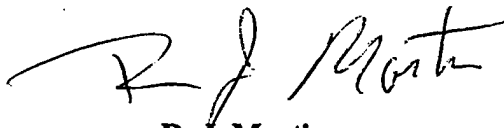
Naval Undersea Warfare Center Division
Newport, Rhode Island

PREFACE

The research presented in this document was sponsored by the Ocean, Atmosphere, and Space S&T Department, Sensing and Systems Division, of the Office of Naval Research (ONR), Program Element 0602314N. The ONR Program Manager for Sensors, Sources, and Arrays is K. G. Dial (ONR 321SS).

The author wishes to thank K. A. Holt for her help with the editing of the manuscript.

Reviewed and Approved: 29 November 1995

A handwritten signature in black ink, appearing to read 'R. J. Martin', with a stylized, cursive script.

R. J. Martin
Acting Head, Submarine Sonar Department

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE 29 November 1995	3. REPORT TYPE AND DATES COVERED Final		
4. TITLE AND SUBTITLE FORTRAN Algorithms for the Three-Dimensional Solution of Two-Layer Solid and Hollow Cylinder Dynamic Elasticity Problems With and Without Fluids		5. FUNDING NUMBERS		
6. AUTHOR(S) Mark S. Peloquin				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Undersea Warfare Center Detachment New London New London, Connecticut 06320		8. PERFORMING ORGANIZATION REPORT NUMBER TD 11,065		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research 800 No. Quincy Street Arlington, VA 22217-5000		10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) This document contains FORTRAN algorithms for the numerical evaluation of exact closed form solutions to four fluid/structure interaction mathematical problems that are modeled as cylinders and fluids. The cylinders (infinitely long in the longitudinal x-direction) are modeled as three-dimensional linear viscoelastic solids, and the fluids are treated as inviscid. The outer fluid extends to infinity in the radial r-direction; the inner fluid (contained within the hollow cylinder) is bounded by the inner cylinder at radius $r = a$. The cylinders are subject to forced harmonic vibration at the solid/fluid interface. The response to both axisymmetric and nonaxisymmetric excitations is calculated, with nonaxisymmetric excitation described in terms of circumferential order number n. The first model is comprised of a single hollow cylinder, filled with an inner fluid and immersed in an outer fluid. The second model consists of a single solid cylinder immersed in an outer fluid. The third model, an extension of the second one, adds a second cylinder in contact with the first cylinder. The fourth model consists of the two-layered cylinder with outer fluid (model three); however, the inner cylinder is now hollow and contains an inner fluid. Structural damping requires that the material properties be represented as complex numbers; therefore, series solutions to the Bessel functions are required. Native algorithms are provided for the Bessel functions, with certain parameter restrictions, and a subroutine is given to allow use of the external FORTRAN Bessel function libraries provided by MathWorks, Inc., without parameter restriction.				
14. SUBJECT TERMS Closed Form Solution, Cylinder, Dynamic Elasticity, Excitation, Fluid/Structure Interaction, Forced Harmonic Vibration, FORTRAN Algorithms, Three-Dimensional Solution			15. NUMBER OF PAGES 260	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT SAR	

TABLE OF CONTENTS

	Page
LIST OF ILLUSTRATIONS	iv
LIST OF TABLES	iv
INTRODUCTION	1
INPUT/OUTPUT VARIABLES	5
Solid Cylinder	5
Cylinder 1	5
Cylinder 2	6
Outer Fluid	6
Inner Fluid	6
General Variables	7
Output Variables	8
SUBPROGRAM DESCRIPTION	9
SUBPROGRAM mr2cf.f	9
SUBPROGRAM cbessl.f	9
Function gamma(n)	9
Function fac(n)	9
Function psi(n)	10
Double Complex Function cbessj(n,a,r)	10
Double Complex Function cbessy(n,a,r)	10
Double Complex Function cbessk(n,a,r)	10
Double Complex Function d1cbessk(n,a,r)	10
Double Complex Function d2cbessk(n,a,r)	10
Double Complex Function d1cbessj(n,a,r)	10
Double Complex Function d2cbessj(n,a,r)	10
Double Complex Function d1cbessy(n,a,r)	10
Double Complex Function d2cbessy(n,a,r)	10
Double Complex Function d1cbessi(n,a,r)	11
Double Complex Function d2cbessi(n,a,r)	11

TABLE OF CONTENTS (CONT'D)

Double Complex Function cbessh1(n,a,r)	11
Double Complex Function cbessh2(n,a,r)	11
Double Complex Function d1cbessh1(n,a,r)	11
Double Complex Function d1cbessh2(n,a,r)	11
SUBPROGRAM rf.f	11
Subroutine ROD_POT(r,k,Om,n,cl_rod,ct_rod)	11
Subroutine SYS_MATRIX_ROD(n,k,ao_rod,Om,ro,co,sm)	12
Subroutine ABC_ROD_SOLVE(exctype,sm)	12
Subroutine ABC_ROD_INVERT(exctype,sm)	12
Subroutine OUTPUT(tft,n,k,r,value)	12
Subroutine MINV(c,cinv,work,n,iflag)	13
SUBPROGRAM c1.f	13
Subroutine C1A_POT(r,k,Om,n)	13
Subroutine C1B_POT(r,k,Om,n)	13
Subroutine SYS_MATRIX_RC1(n,k,ao_rod,ao_1cyl, bo_c1,Om,ro,co,smrc1)	14
Subroutine ABC_RC1_INVERT(n,exctype,smrc1,b)	14
Subroutine OUTPUT_RC1(tft,n,k,r,value)	14
SUBPROGRAM c2.f	15
Subroutine C2B_POT(r,k,Om,n)	15
Subroutine C2C_POT(r,k,Om,n)	15
Subroutine SYS_MATRIX_RC2(n,k,ao_1cyl,bo_c1,co_c2, Om,ro,co,ri,ci,smrc2)	15
Subroutine ABC_RC2_INVERT(n,exctype,smrc2,a,c)	16
Subroutine OUTPUT_RC2(tft,n,k,r,value)	16
SUBPROGRAM fluids.f	16
Subroutine IFL_POT(n,r,k,Om,ci)	16
Subroutine OFL_POT(n,r,k,Om,co)	17
Subroutine OUTPUT_IF(tft,n,k,r,value,Om,ri)	17
Subroutine OUTPUT_OF(tft,n,k,r,value,Om,ro)	17
SUBPROGRAM smc1.f	17

TABLE OF CONTENTS (CONT'D)

Subroutine SYS_MATRIX_C1(n,k,ao_1cyl,bo_c1,Om,ro,co, ri,ci,smc1)	17
Subroutine ABC_C1_INVERT(n,exctype,smc1,a,b)	17
FLOW CHARTS.....	20
Inner Fluid/Cylinder 1/Outer Fluid.....	20
SolidCylinder/Outer Fluid	22
Solid Cylinder/Cylinder 2/Outer Fluid	24
Inner Fluid/Cylinder 1/Cylinder 2/Outer Fluid.....	26
SUBPROGRAM LISTINGS	29
Listing for mr2cf.f.....	29
Listing for cbessl.f	42
Listing for rf.f.....	56
Listing for c1.f	70
Listing for c2.f	94
Listing for fluids.f	126
Listing for smc1.f.....	131
CROSS REFERENCE	141
Subprograms	142
Variables.....	147
Labels.....	203
Strings	204
Index	205
APPENDIX A—INTERMEDIATE VARIABLES FOR NUWC-NPT TR 11,043.....	A - 1
APPENDIX B—INTERMEDIATE VARIABLES FOR NUWC-NPT TR 11,067.....	B - 1
APPENDIX C—MAKEFILE.....	C - 1
APPENDIX D—SUBROUTINE cbessl.f_matlab.....	D - 1
APPENDIX E—MODIFIED SUBROUTINES FOR CIRCUMFERENTIAL EXCITATION.....	E - 1

LIST OF ILLUSTRATIONS

Figure	Page
1 Model Types $emt = 1$ and $emt = 2$	3
2 Model Types $emt = 3$ and $emt = 4$	4
3 Flow Chart 1—Inner Fluid, Cylinder 1, and Outer Fluid for $emt = 1$	21
4 Flow Chart 2—Solid Cylinder and Outer Fluid for $emt = 2$	23
5 Flow Chart 3—Solid Cylinder, Cylinder 2, and Outer Fluid for $emt = 3$	25
6 Flow Chart 4—Inner Fluid, Cylinder 1, Cylinder 2, and Outer Fluid for $emt = 4$	27

LIST OF TABLES

Table	Page
1 INPUTS FOR THE SOLID CYLINDER.....	5
2 INPUTS FOR CYLINDER 1.....	5
3 INPUTS FOR CYLINDER 2.....	6
4 INPUTS FOR THE OUTER FLUID.....	6
5 INPUTS FOR THE INNER FLUID	6
6 GENERAL VARIABLES	7
7 OUTPUT SETUP VARIABLES.....	8
8 OUTPUT VARIABLE	8

FORTRAN ALGORITHMS FOR THE THREE-DIMENSIONAL SOLUTION OF TWO-LAYER SOLID AND HOLLOW CYLINDER DYNAMIC ELASTICITY PROBLEMS WITH AND WITHOUT FLUIDS

INTRODUCTION

Closed form solutions for wave propagation in a two-layered cylinder with outer fluid loading were analyzed in NUWC-NPT Technical Report 11,043.¹ The solution for the subproblem consisting of a single solid cylinder in contact with an outer fluid was also presented in this reference.

Additionally, a closed form solution for wave propagation in a two-layered cylinder with inner and outer fluid loading was derived in NUWC-NPT Technical Report 11,067.² The subproblem of a single solid cylinder in contact with inner and outer fluids was also addressed.

In both of these previous derivations, the cylinders were infinite in the longitudinal direction. Damping was modeled using a complex modulus of elasticity, which required a series solution for the requisite Bessel functions. Three excitation states were considered: the normal pressure P_o , the longitudinal shear stress P_x , and the circumferential shear stress P_θ . The response was derived for nonaxisymmetric excitation in terms of circumferential order number n . The axisymmetric response was obtained by the degenerate case of $n = 0$.

The closed form solutions described above have been programmed in FORTRAN to obtain numerical results. This document details the FORTRAN algorithms developed to describe the cases of wave propagation presented in references 1 and 2. Variable lists, subroutine

1. M. S. Peloquin, "A Three-Dimensional Elasticity Solution for Wave Propagation in a Two-Layered Infinite Viscoelastic Solid Cylinder With Outer Fluid Loading," NUWC-NPT Technical Report 11,043, Naval Undersea Warfare Center Detachment, New London, CT, 31 August 1995.

2. M. S. Peloquin, "A Closed Form Dynamic Elasticity Solution to the Fluid/Structure Interaction Problem of a Two-Layered Infinite Viscoelastic Cylinder With Inner and Outer Fluid Loading Subject to Forced Harmonic Excitation," NUWC-NPT Technical Report 11,067, Naval Undersea Warfare Center Detachment, New London, CT, 30 December 1995.

descriptions, flow charts, and a cross reference of the FORTRAN program are presented.

The input variables are contained in tables 1 through 6. These tables relate the FORTRAN variables to the corresponding symbols used in the references. Intermediate variables used in the FORTRAN algorithms and the references are listed in appendixes A and B. Table 7 lists the output variables necessary to specify the quantity desired for output and the excitation type. It also lists the particular model to be used in the calculation:

- inner fluid/cylinder 1/outer fluid
- solid cylinder/outer fluid
- solid cylinder/cylinder 2/outer fluid
- inner fluid/cylinder 1/cylinder 2/outer fluid.

These four model types are identified by variable *emt* and are depicted in figures 1 and 2. Table 8 describes the output variable.

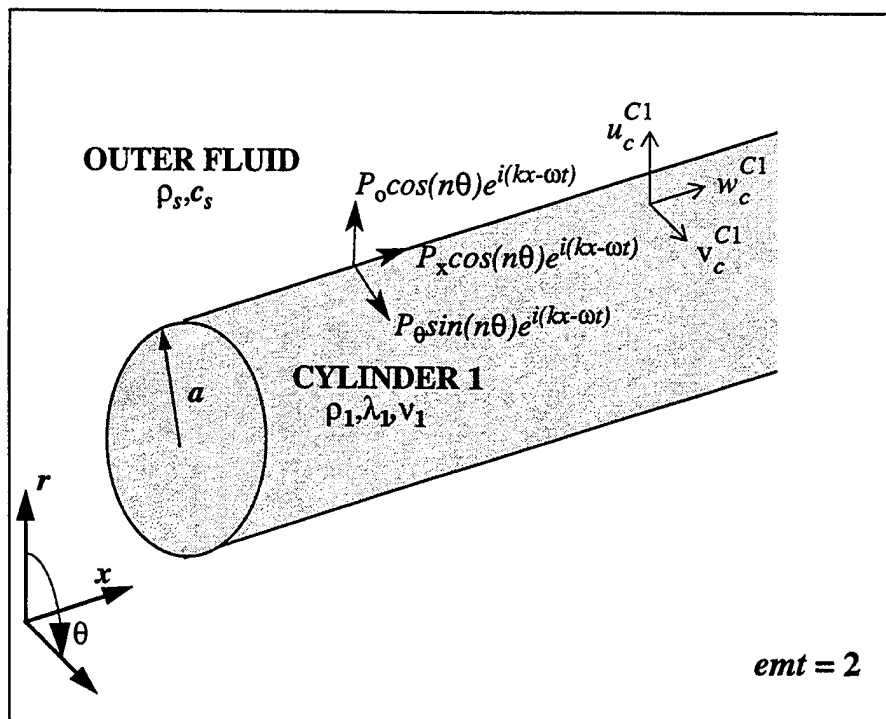
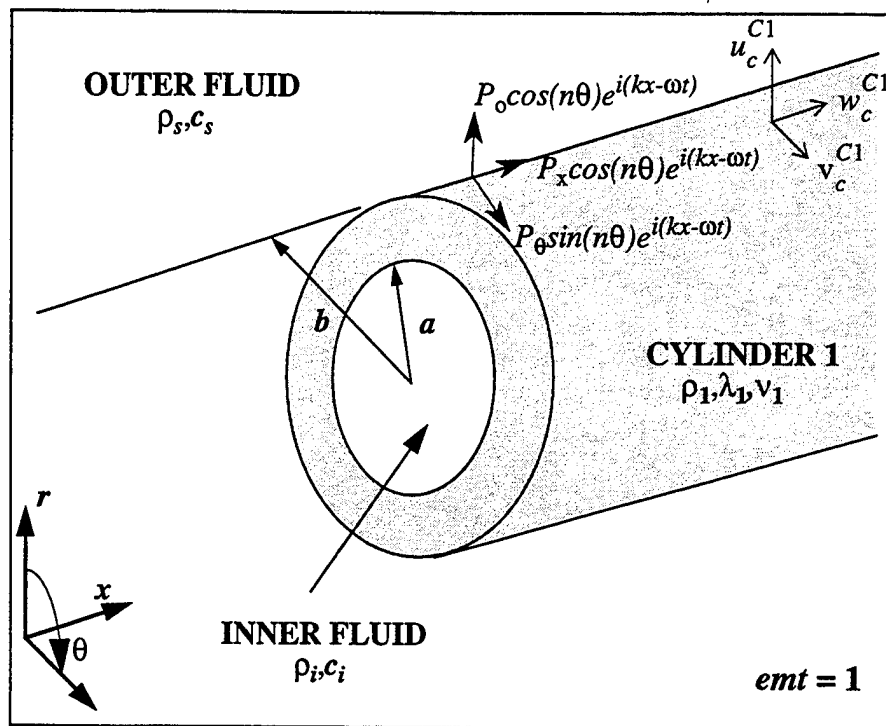


Figure 1. Model Types $emt = 1$ and $emt = 2$

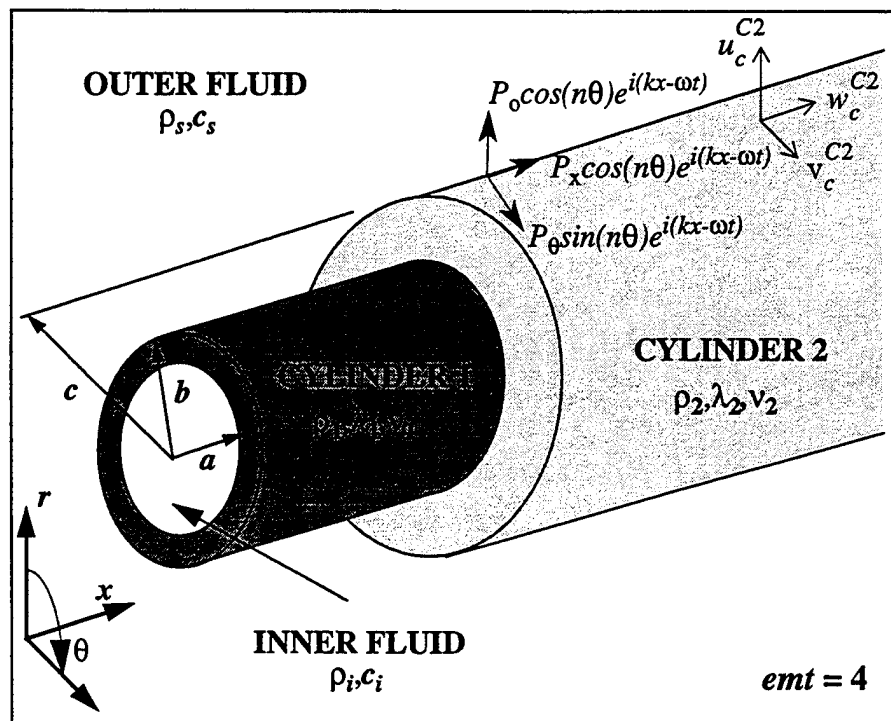
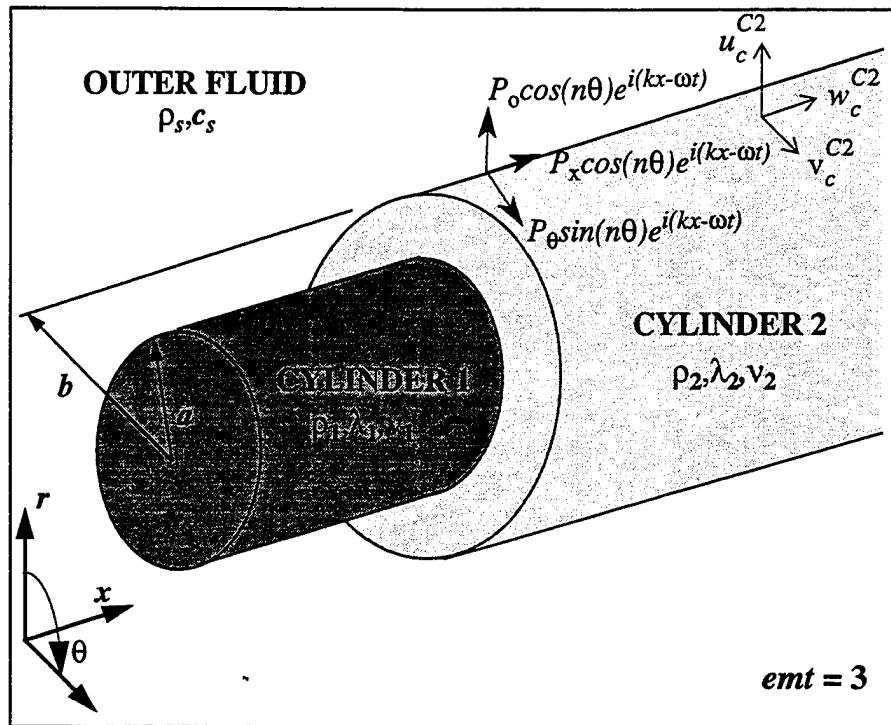


Figure 2. Model Types $emt = 3$ and $emt = 4$

INPUT/OUTPUT/VARIABLES

SOLID CYLINDER

Table 1. Inputs for the Solid Cylinder

FORTTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION
E_rod	E_1	Young's modulus
zeta_rod	ζ_1	Structural loss factor
mu_rod	ν_1	Poisson ratio
r_rod	ρ_1	Density
ao_rod	a	Outer radius

CYLINDER 1

Table 2. Inputs for Cylinder 1

FORTTRAN VARIABLE	TR 11,043 VARIABLE	TR11,067 VARIABLE	DESCRIPTION
E_1cyl	E_2	E_1	Young's modulus
zeta_1cyl	ζ_2	ζ_1	Structural loss factor
mu_1cyl	ν_2	ν_1	Poisson ratio
r_1cyl	ρ_2	ρ_1	Density
ao_1cyl	a	a	Inner radius
h_1cyl	No equivalent	No equivalent	Cylinder wall thickness

CYLINDER 2**Table 3. Inputs for Cylinder 2**

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
E_2cyl	E_2	Young's modulus
zeta_2cyl	ζ_2	Structural loss factor
mu_2cyl	ν_2	Poisson ratio
r_2cyl	ρ_2	Density
ao_2cyl	b	Inner radius
h_2cyl	No equivalent	Cylinder wall thickness

OUTER FLUID**Table 4. Inputs for the Outer Fluid**

FORTTRAN VARIABLE	TR 11,043 TR11067 VARIABLE	DESCRIPTION
co	c_s	Speed of sound in the outer fluid
ro	ρ_s	Density of the outer fluid

INNER FLUID**Table 5. Inputs for the Inner Fluid**

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
ci	c_i	Speed of sound in the inner fluid
ri	ρ_i	Density of the inner fluid

GENERAL VARIABLES**Table 6. General Variables**

FORTTRAN VARIABLE	TR 11,043 TR 11,067 VARIABLES	DESCRIPTION
k	k	Calculation wavenumber
Ωm	ω	Calculation frequency
r	r_1	Calculation radius
n	n	Circumferential order number

OUTPUT VARIABLES

Table 7. Output Setup Variables

FORTTRAN VARIABLE	VALUE	DESCRIPTION
tft	0	RADIAL STRESS or FLUID PRESSURE
	1	LONGITUDINAL STRESS
	2	AXIAL DISPLACEMENT
	3	THETA DISPLACEMENT
	4	RADIAL DISPLACEMENT
	5	LONGITUDINAL DISPLACEMENT
	6	CIRCUMFERENTIAL STRAIN
	7	RADIAL STRAIN
	8	
	9	OPTICAL PHASE SENSITIVITY
	10	OPTICAL PHASE SENSITIVITY
exctype	0	LONGITUDINAL EXCITATION
	1	RADIAL EXCITATION
	2	CIRCUMFERENTIAL EXCITATION
emt	1	INNER FLUID/CYLINDER 1/OUTER FLUID
	2	SOLID CYLINDER/OUTER FLUID
	3	SOLID CYLINDER/CYLINDER 2/FLUID
	4	INNER FLUID/CYLINDER 1/CYLINDER 2/OUTER FLUID

Table 8. Output Variable

FORTTRAN VARIABLE	VALUE	DESCRIPTION
g(jk)	Calculated	Contains the calculated quantity, specified by variables <i>tft</i> and <i>exctype</i> .

SUBPROGRAM DESCRIPTION

SUBPROGRAM *mr2cf.f*

This is a miniature main program that is used to evaluate the various wave propagation models described earlier. It is an interactive program in which the user enters the wavenumber and frequency at which the calculation will be performed. All the input and output variables are specified as well. Appendix C contains the makefile that was used to generate the executable.

The logical flow in the flow charts can be embedded in a routine that will generate an entire response surface over a range of wavenumbers and frequencies. This is the method used to generate the images in references 1 and 2.

SUBPROGRAM *cbessl.f*

The series solutions for the Bessel functions of complex argument used in reference 1 were generated with the functions contained in *cbessl.f* ($emt = 2$ and $emt = 3$). This group of functions is adequate for use when $n = 0$ or 1 for model $emt = 1$ or $emt = 4$. When n is greater than 1 , the subroutine *cbessl.f_matlab* (appendix D) is used to call series solutions to the Bessel functions provided by MATLAB in its user libraries. The MATLAB user libraries are not provided in this document. Any Bessel function library could be used by making appropriate changes in the *cbessl.f_matlab* file and linking the Bessel function library of choice when compiling.

The following functions are contained in the FORTRAN subprogram *cbessl.f*:

Function gamma(n)

Gamma function of argument n .

Function fac(n)

Factorial of argument n .

Function $\psi(n)$

Psi function of argument n .

Double Complex Function $\text{cbessj}(n,a,r)$

Bessel function of the first kind of integer order n and argument ar , where a is a complex number.

Double Complex Function $\text{cbessy}(n,a,r)$

Bessel function of the second kind of integer order n and argument ar , where a is a complex number.

Double Complex Function $\text{cbessk}(n,a,r)$

Modified Bessel function of integer order n and argument ar , where a is a complex number.

Double Complex Function $d1\text{cbessk}(n,a,r)$

First derivative with respect to r of the modified Bessel function $\text{cbessk}(n,a,r)$.

Double Complex Function $d2\text{cbessk}(n,a,r)$

Second derivative with respect to r of the modified Bessel function $\text{cbessk}(n,a,r)$.

Double Complex Function $d1\text{cbessj}(n,a,r)$

First derivative with respect to r of the Bessel function of the first kind $\text{bessj}(n,a,r)$.

Double Complex Function $d2\text{cbessj}(n,a,r)$

Second derivative with respect to r of the Bessel function of the first kind $\text{bessj}(n,a,r)$.

Double Complex Function $d1\text{cbessy}(n,a,r)$

First derivative with respect to r of the Bessel function of the second kind $\text{bessy}(n,a,r)$.

Double Complex Function $d2\text{cbessy}(n,a,r)$

Second derivative with respect to r of the Bessel function of the second kind $\text{bessy}(n,a,r)$.

Double Complex Function $d1cbessi(n,a,r)$

First derivative with respect to r of the modified Bessel function $cbessi(n,a,r)$.

Double Complex Function $d2cbessi(n,a,r)$

Second derivative with respect to r of the modified Bessel function $cbessi(n,a,r)$.

Double Complex Function $cbessh1(n,a,r)$

Hankel function of the first kind of integer order n and argument ar , where a is a complex number.

Double Complex Function $cbessh2(n,a,r)$

Hankel function of the second kind of integer order n and argument ar , where a is a complex number.

Double Complex Function $d1cbessh1(n,a,r)$

First derivative with respect to r of the Hankel function of the first kind $cbessh1(n,a,r)$.

Double Complex Function $d1cbessh2(n,a,r)$

First derivative with respect to r of the Hankel function of the second kind $cbessh2(n,a,r)$.

SUBPROGRAM rf.f**Subroutine $ROD_POT(r,k,Om,n,cl_rod,ct_rod)$**

Displacement potentials for the solid cylinder are calculated. This subroutine is called once at $r = a$ in order to facilitate the solution of the undetermined coefficients. It is called a second time in cases $emt = 2$ and 3 when the calculation radius falls within the envelope of the solid cylinder. The following variables are calculated and returned by this subroutine in Common Block /ROD/:

```

65 C DEFINITION FOR COMMON BLOCK /ROD/
67     complex*16 SP_rod,d1_SP_rod,d2_SP_rod
68     complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
69     complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod

```

Subroutine SYS_MATRIX_ROD(n,k,ao_rod,Om,ro,co,sm)

Coefficients of the system matrix, sm , for the solid cylinder/outer fluid combination ($emt = 2$) are calculated. A fully populated sm matrix is returned.

Subroutine ABC_ROD_SOLVE($exctype,sm$)

This subroutine solves for the undetermined coefficients of the solid cylinder/outer fluid combination ($emt = 2$) by using the method of determinants to invert the system matrix sm . This subroutine returns the following variables in Common Blocks /ROD/ and /OFLUID/:

```

65 C DEFINITION FOR COMMON BLOCK /ROD/
70      complex*16 A1_rod,B1_rod,C1_rod

80 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
81
82      complex*16 M_OF
```

Subroutine ABC_ROD_INVERT($exctype,sm$)

Undetermined coefficients are solved for through the inversion of matrix sm ($emt = 2$). The subroutine *MINV* is called from *ABC_ROD_INVERT* to perform the complex matrix inversion on matrix sm . This subroutine returns the following variables in Common Block /ROD/ and /OFLUID/:

```

65 C DEFINITION FOR COMMON BLOCK /ROD/
70      complex*16 A1_rod,B1_rod,C1_rod

80 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
81
82      complex*16 M_OF
```

Subroutine OUTPUT($tft,n,k,r,value$)

The particular output quantity of interest for the solid cylinder at radius r is calculated in this subroutine based on the value of the variable tft . Cases of $emt = 2$ and 3 are applicable. The calculated quantity is returned by the variable $value$.

Subroutine MINV(*c,cinv,work,n,iflag*)

With this subroutine, complex system matrices are inverted for all permissible values of the variable *emt*. Matrix *c* is passed to *MINV* for inversion and *cinv* is the inverted matrix that is returned from the subroutine.

SUBPROGRAM c1.f**Subroutine C1A_POT(*r,k,Om,n*)**

The displacement potentials for the first (inner) cylinder, evaluated at radius $r = a$, are calculated in this subroutine. This subroutine is also called when the calculation radius falls within the inner cylinder ($emt = 1, 3$, and 4); in this case, r is passed to the subroutine. The following variables are calculated and returned in Common Block /CYLINDER1/:

```

90 C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
92     complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
93     complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
97     complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
98     complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
102    complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
103    complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2

```

Subroutine C1B_POT(*r,k,Om,n*)

The displacement potentials for cylinder 2, evaluated at radius $r = b$, are calculated in this subroutine. The following variables are calculated and returned in Common Block /CYLINDER1/:

```

90 C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
94     complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
95     complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2
99     complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
100    complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
104    complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
105    complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2

```

Subroutine SYS_MATRIX_RC1(n,k,ao_rod,ao_1cyl,bo_c1,Om,ro,co,smrc1)

The system matrix for $emt = 3$ is calculated in this subroutine and is returned as *smrc1*.

Subroutine ABC_RC1_INVERT(n,exctype,smrc1,b)

Undetermined coefficients are solved for through the inversion of matrix *smrc1* ($emt = 3$). The subroutine *MINV* is called from *ABC_RC1_INVERT* to perform the complex matrix inversion. The following variables are calculated and returned in Common Blocks /ROD/, /OFLUID/, and /CYLINDER1/:

```

65 C DEFINITION FOR COMMON BLOCK /ROD/
70      complex*16 A1_rod,B1_rod,C1_rod

80 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
81
82      complex*16 M_OF

90 C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
108     complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
```

Subroutine OUTPUT_RC1(tft,n,k,r,value)

When the desired radius for output falls within the first, or inner, cylinder, this subroutine is used. Cases of $emt = 1, 3$, and 4 are applicable. The calculated quantity is returned by the variable *value*.

SUBPROGRAM c2.f***Subroutine C2B_POT(r,k,Om,n)***

The displacement potentials for the second (outer) cylinder, evaluated at radius $r = b$, are calculated with this subroutine. This subroutine is also called when the calculation radius falls within the outer cylinder ($emt = 4$); in this case, r is passed to the subroutine. The following variables are calculated and returned in Common Block /CYLINDER2/:

```

130 C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/
131
132     complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
133     complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
137     complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
138     complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
142     complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1
143     complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2

```

Subroutine C2C_POT(r,k,Om,n)

The displacement potentials for cylinder 2, evaluated at radius $r = c$, are calculated with this subroutine. The following variables are calculated and returned in Common Block /CYLINDER2/:

```

130 C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/
131
134     complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
135     complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2
139     complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
140     complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2
144     complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1
145     complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2

```

Subroutine SYS_MATRIX_RC2(n,k,ao_1cyl,bo_c1,co_c2,Om,ro,co,ri,ci,smrc2)

The system matrix *smrc2* for $emt = 4$ is calculated and returned with this subroutine.

Subroutine ABC_RC2_INVERT(n,exctype,smrc2,a,c)

Undetermined coefficients are solved for through the inversion of matrix *smrc2* (*emt* = 4). The subroutine *MINV* is called from *ABC_RC2_INVERT* to perform the complex matrix inversion. The following variables are returned in Common Blocks */OFLUID/*, */CYLINDER1/*, */CYLINDER2/*, and */IFLUID/*:

80 C DEFINITIONS FOR COMMON BLOCK */OFLUID/*

82 complex*16 M_OF

90 C DEFINITIONS FOR COMMON BLOCK */CYLINDER1/*

108 complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

130 C DEFINITIONS FOR COMMON BLOCK */CYLINDER2/*

149 complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

170 C DEFINITIONS FOR COMMON BLOCK */IFLUID/*

171

172 complex*16 D_IF

Subroutine OUTPUT_RC2(tft,n,k,r,value)

When the desired radius for output falls within the second or outer cylinder, this subroutine is used. Cases of *emt* = 3 and 4 are applicable. The calculated quantity is returned by the variable *value*.

SUBPROGRAM fluids.f

Subroutine IFL_POT(n,r,k,Om,ci)

Inner fluid displacement potentials are calculated with this subroutine. This subroutine is called once at $r = a$ in order to facilitate the solution of the undetermined coefficients. It is called a second time for *emt* = 1 and 4 when the desired calculation radius corresponds to the region of the inner fluid. The following variables are calculated and returned in Common Block */IFLUID/*:

170 C DEFINITIONS FOR COMMON BLOCK */IFLUID/*

171

172 complex*16 IFSC,d1_IFSC

Subroutine OFL_POT(n,r,k,Om,co)

Outer fluid displacement potentials are calculated with this subroutine. This subroutine is called once at $r = b$ for $emt = 1, 2$, and 3 and once at $r = c$ for $emt = 4$. It is called a second time for all model cases where the desired calculation radius corresponds to the region of the outer fluid. The following variables are returned in the Common Block /OFLUID/:

```

80 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
81
82      complex*16 OFSC,d1_OFSC

```

Subroutine OUTPUT_IF(tft,n,k,r,value,Om,ri)

If the radius at which output is desired corresponds to the region occupied by the inner fluid, this subroutine is used to calculate either the pressure or velocity transfer function based on the value of tft . The calculated quantity is returned by the variable $value$.

Subroutine OUTPUT_OF(tft,n,k,r,value,Om,ro)

If the radius at which output is desired corresponds to the region occupied by the outer fluid, this subroutine is used to calculate either the pressure or velocity transfer function based on the value of tft . The calculated quantity is returned by the variable $value$.

SUBPROGRAM smc1.f***Subroutine SYS_MATRIX_C1(n,k,ao_1cyl,bo_c1,Om,ro,co,ri,ci,smc1)***

The system matrix $smc1$ for $emt = 1$ is calculated and returned with this subroutine.

Subroutine ABC_C1_INVERT(n,exctype,smc1,a,b)

Undetermined coefficients are solved for through the inversion of matrix $smrc1$ ($emt = 1$). The subroutine *MINV* is called from *ABC_C1_INVERT* to perform the complex matrix inversion. The following variables are calculated and returned in Common Blocks

/OFLUID/, /CYLINDER1/, and /IFLUID:

80 C DEFINITIONS FOR COMMON BLOCK /OFLUID/

81

82 complex*16 M_OF

90 C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

108 complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

170 C DEFINITIONS FOR COMMON BLOCK /IFLUID/

171

172 complex*16 D_IF

FLOW CHARTS

INNER FLUID/CYLINDER 1/OUTER FLUID

This case, $emt = 1$, involves three media: the inner fluid, cylinder 1, and the outer fluid. Flow chart 1 in figure 3 displays the necessary execution sequence for this model. Subroutines C1A_POT and C1B_POT calculate the Bessel functions needed for the cylinder displacement potentials at the inner and outer radii of the cylinder, respectively. OFL_POT and IFL_POT calculate the Bessel functions needed for the displacement potentials of the outer and inner fluids, respectively. The results of these four subroutine calls are passed to subroutine SYS_MATRIX_C1 via common blocks, and then the components of the system matrix, $smc1$, are calculated. The undetermined coefficients are solved for after system matrix $smc1$ is inverted in subroutine ABC_C1_INVERT.

At this point, the undetermined coefficients are known for the given problem, and final output can be calculated at a radius corresponding to any one of the three media mentioned above. The output calculations are partitioned by the three *if* statements, and the output quantity is calculated based on the value of the radius r_1 specified for the output calculation. If r_1 is less than or equal to the inner radius of the cylinder, the IFL_POT subroutine is used to calculate the Bessel functions needed for the inner fluid displacement potential and then OUTPUT_IF is used to calculate the desired output quantity according to the value of variable tft . If r_1 is less than or equal to the outer radius of the cylinder and greater than the inner radius, then C1A_POT is used to calculate the Bessel functions needed for the displacement potentials and OUTPUT_RC1 is used to calculate the desired output quantity according to the value of the variable tft . When r_1 is greater than the outer radius of the cylinder, subroutine OFL_POT is called, and the Bessel functions needed for the outer fluid displacement potential at radius r_1 are calculated. Subroutine OUTPUT_OF is used to calculate the desired output quantity according to the value of the variable tft .

The subroutines listed in appendix D have been modified to account for the case of circumferential excitation. Substitution of the appendix D subroutines into subprograms $rf.f$, $smc1.f$, $c1.f$, and $c2.f$ is required in order to calculate the response from circumferential excitation P_θ .

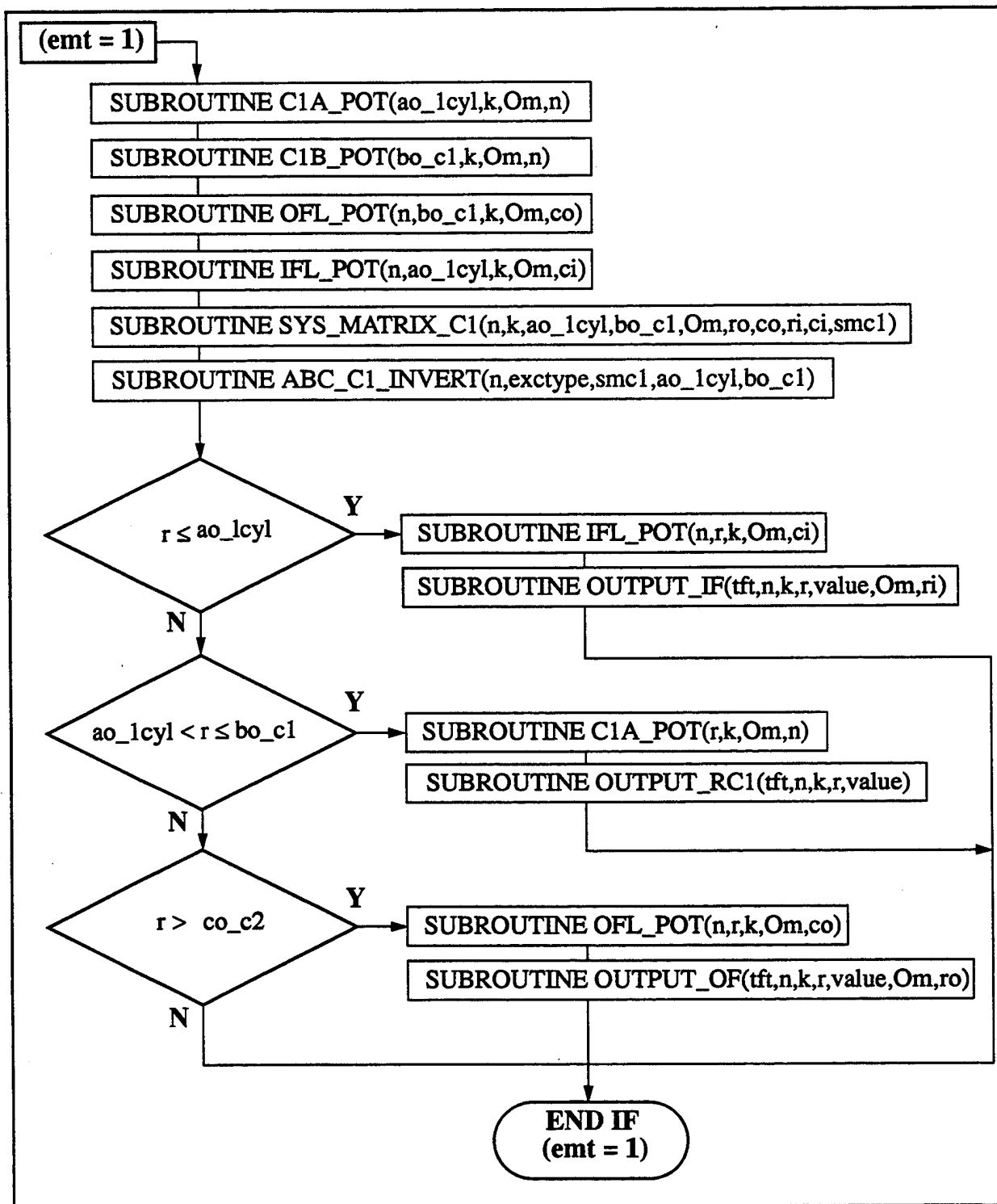


Figure 3. Flow Chart 1—Inner Fluid, Cylinder 1, and Outer Fluid for $emt = 1$

SOLID CYLINDER/OUTER FLUID

This case, $emt = 2$, involves two media: the solid cylinder and the outer fluid. In figure 4, flow chart 2 displays the necessary execution sequence for this model. Subroutine ROD_POT calculates the Bessel functions needed for the solid cylinder displacement potentials evaluated at the outer radius of the cylinder. Subroutine OFL_POT calculates the Bessel functions needed for the outer fluid displacement potential. The results of these two subroutine calls are passed to subroutine SYS_MATRIX_ROD via common blocks, and the components of the system matrix sm are calculated. The undetermined coefficients are solved for after system matrix sm is inverted in subroutine ABC_ROD_INVERT.

At this point, the undetermined coefficients are known for the given problem, and final output can be calculated at a radius corresponding to any one of the two media mentioned above. The output calculations are partitioned by the two *if* statements, and the output quantity is calculated based on the value of the radius (r_1) specified for output calculation. If r_1 is less than the outer radius of the cylinder, then ROD_POT is used to calculate the Bessel functions needed for the displacement potentials and OUTPUT is used to calculate the desired output quantity according to the value of the variable tft . When r_1 is greater than the outer radius of the cylinder, subroutine OFL_POT is called and the Bessel functions needed for the outer fluid displacement potential at radius r_1 are calculated. Subroutine OUTPUT_OF is used to calculate the desired output quantity according to the value of the variable tft .

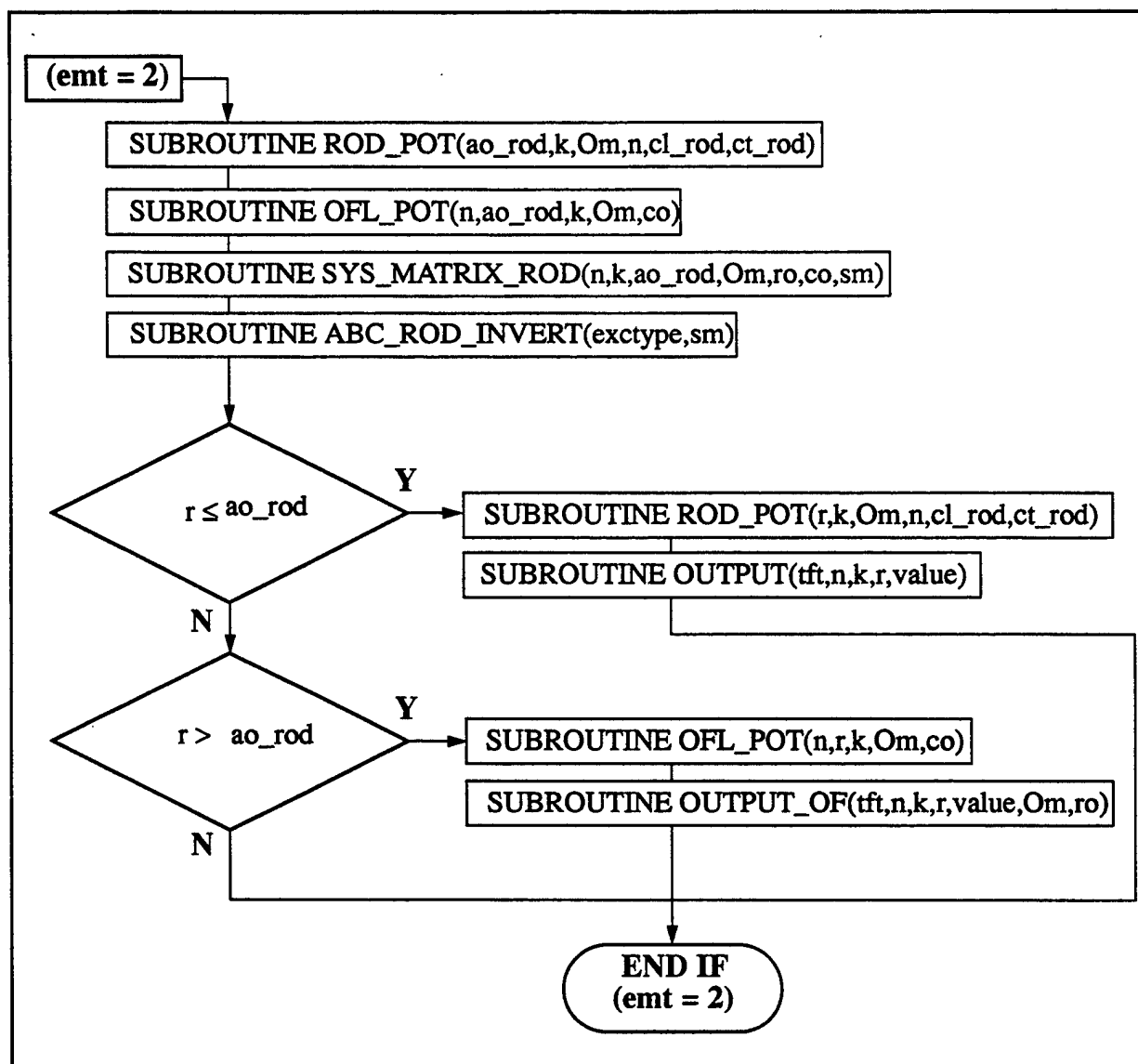


Figure 4. Flow Chart 2—Solid Cylinder and Outer Fluid for $emt = 2$

SOLID CYLINDER/CYLINDER 2/OUTER FLUID

This case, $emt = 3$, involves three media: the solid cylinder, cylinder 2, and the outer fluid. In figure 5, flow chart 3 displays the necessary execution sequence for this model. Subroutines C1A_POT and C1B_POT calculate the Bessel functions needed for the displacement potentials of cylinder 2 at the inner and outer radii of the cylinder, respectively. Subroutine ROD_POT calculates the Bessel functions needed for the solid cylinder displacement potentials at $r = a$. OFL_POT calculates the Bessel functions needed for the outer fluid displacement potential at $r = b$. The results of these four subroutine calls are passed to subroutine SYS_MATRIX_RC1 via common blocks, and then the components of the system matrix, $smrc1$, are calculated. The undetermined coefficients are solved for after system matrix $smrc1$ is inverted in subroutine ABC_RC1_INVERT.

At this point, the undetermined coefficients are known for the given problem, and final output can be calculated at a radius corresponding to any one of the three media mentioned above. The output calculations are partitioned by the three *if* statements, and the output quantity is calculated based on the value of the radius (r_1) specified for output calculation. If r_1 is less than or equal to the outer radius of the solid cylinder, the ROD_POT subroutine is used to calculate the Bessel functions needed for the displacement potentials at radius r_1 and then OUTPUT is used to calculate the desired output quantity according to the value of variable tft . If r_1 is less than or equal to the outer radius of cylinder 2 and greater than the inner radius, then C1A_POT is used to calculate the Bessel functions needed for the displacement potentials and OUTPUT_RC1 is used to calculate the desired output quantity according to the value of the variable tft . When r_1 is greater than the outer radius of cylinder 2, subroutine OFL_POT is called and the Bessel functions needed for the outer fluid displacement potential at radius r_1 are calculated. Subroutine OUTPUT_OF is used to calculate the desired output quantity according to the value of the variable tft .

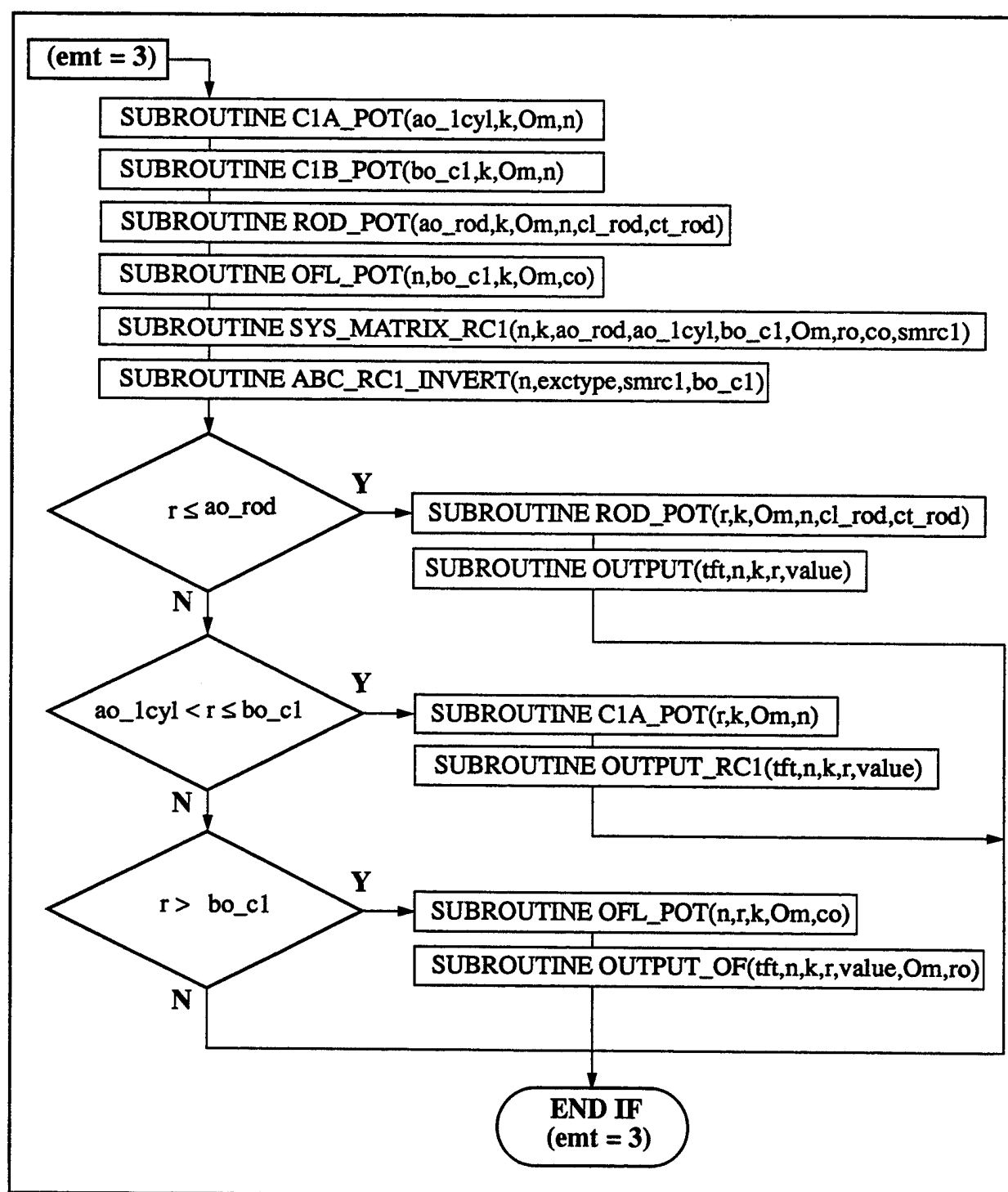


Figure 5. Flow Chart 3—Solid Cylinder, Cylinder 2, and Outer Fluid for $emt = 3$

INNER FLUID/CYLINDER 1/CYLINDER 2/OUTER FLUID

This case, $emt = 4$, involves four media: the inner fluid, cylinder 1, cylinder 2, and the outer fluid. In figure 6, flow chart 4 displays the necessary execution sequence for this model. Subroutines C1A_POT and C1B_POT calculate the Bessel functions needed for the displacement potentials of cylinder 1 at the inner and outer radii of the cylinder, respectively. Subroutines C2B_POT and C2C_POT calculate the Bessel functions needed for the displacement potentials of cylinder 2 at $r = b$ and $r = c$, respectively. OFL_POT calculates the Bessel functions needed for the displacement potential of the outer fluid at $r = c$. The results of these six subroutine calls are passed to subroutine SYS_MATRIX_RC2 via common blocks, and then the components of the system matrix, $smrc2$, are calculated. The undetermined coefficients are solved for after system matrix $smrc2$ is inverted in subroutine ABC_RC2_INVERT.

At this point, the undetermined coefficients are known for the given problem, and final output can be calculated at a radius corresponding to any one of the four media mentioned above. The output calculation is partitioned by the four *if* statements, and the output quantity is calculated based on the value of the radius (r_1) specified for output calculation. If r_1 is less than or equal to the inner radius of cylinder 1, IFL_POT is called to evaluate the Bessel functions needed for the fluid displacement potential. Subroutine OUTPUT_IF is used to calculate the desired output quantity according to the value of the variable tft . If r_1 is less than or equal to the outer radius of cylinder 1 and greater than the inner radius, subroutine C1A_POT is used to calculate the Bessel functions needed for the displacement potentials of cylinder 1 at radius r_1 and then subroutine OUTPUT is used to calculate the desired output quantity according to the value of the variable tft . If r_1 is less than or equal to the outer radius of cylinder 2 and greater than the inner radius, then C2B_POT is used to calculate the Bessel functions needed for the displacement potentials at radius r_1 and OUTPUT_RC2 is used to calculate the desired output quantity according to the value of the variable tft . When r_1 is greater than the outer radius of cylinder 2, subroutine OFL_POT is called and the Bessel functions needed for the displacement potential of the outer fluid at radius r_1 are calculated. Subroutine OUTPUT_OF is used to calculate the desired output quantity according to the value of the variable tft .

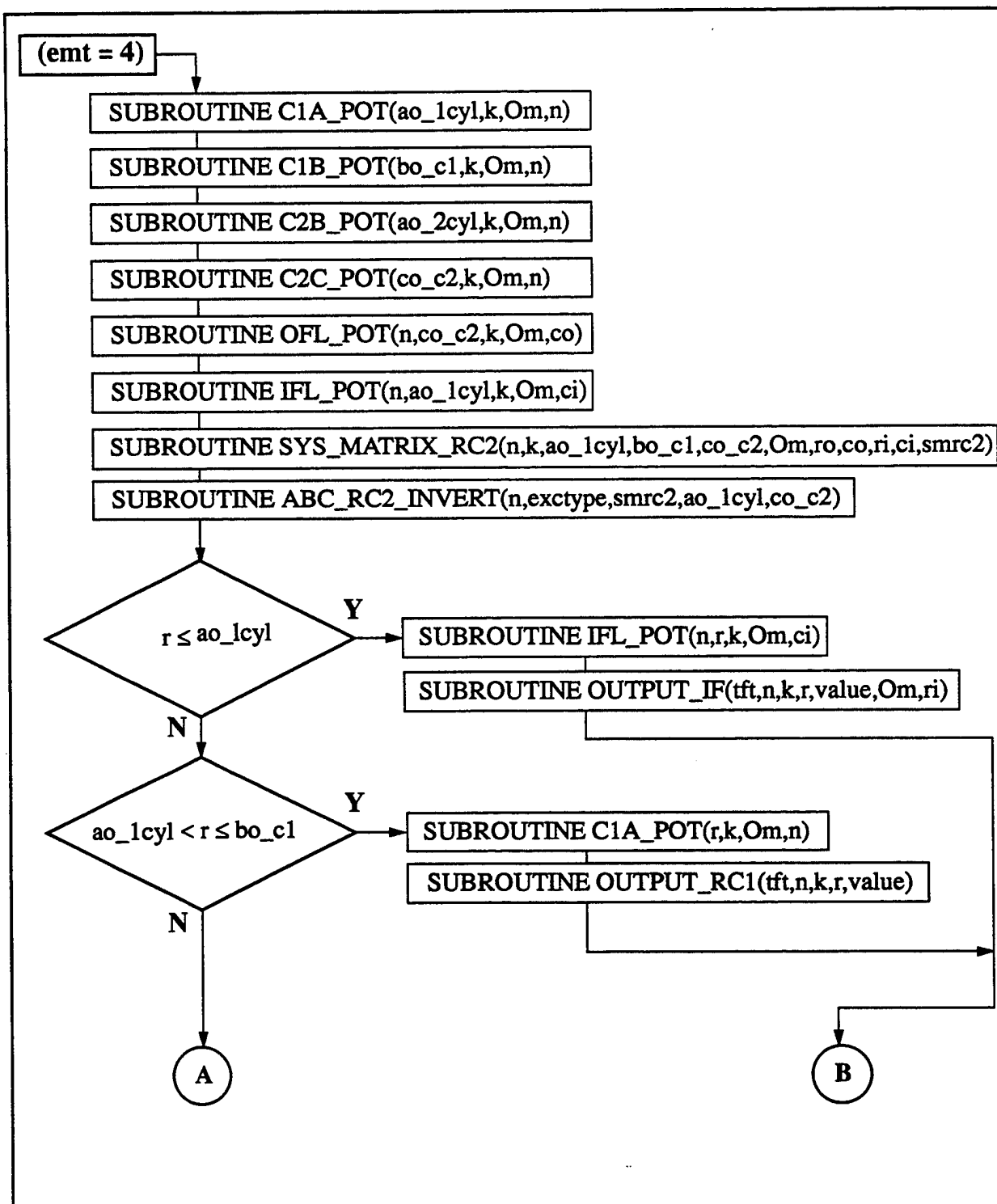


Figure 6. Flow Chart 4—Inner Fluid, Cylinder 1, Cylinder 2, and Outer Fluid for $emt = 4$

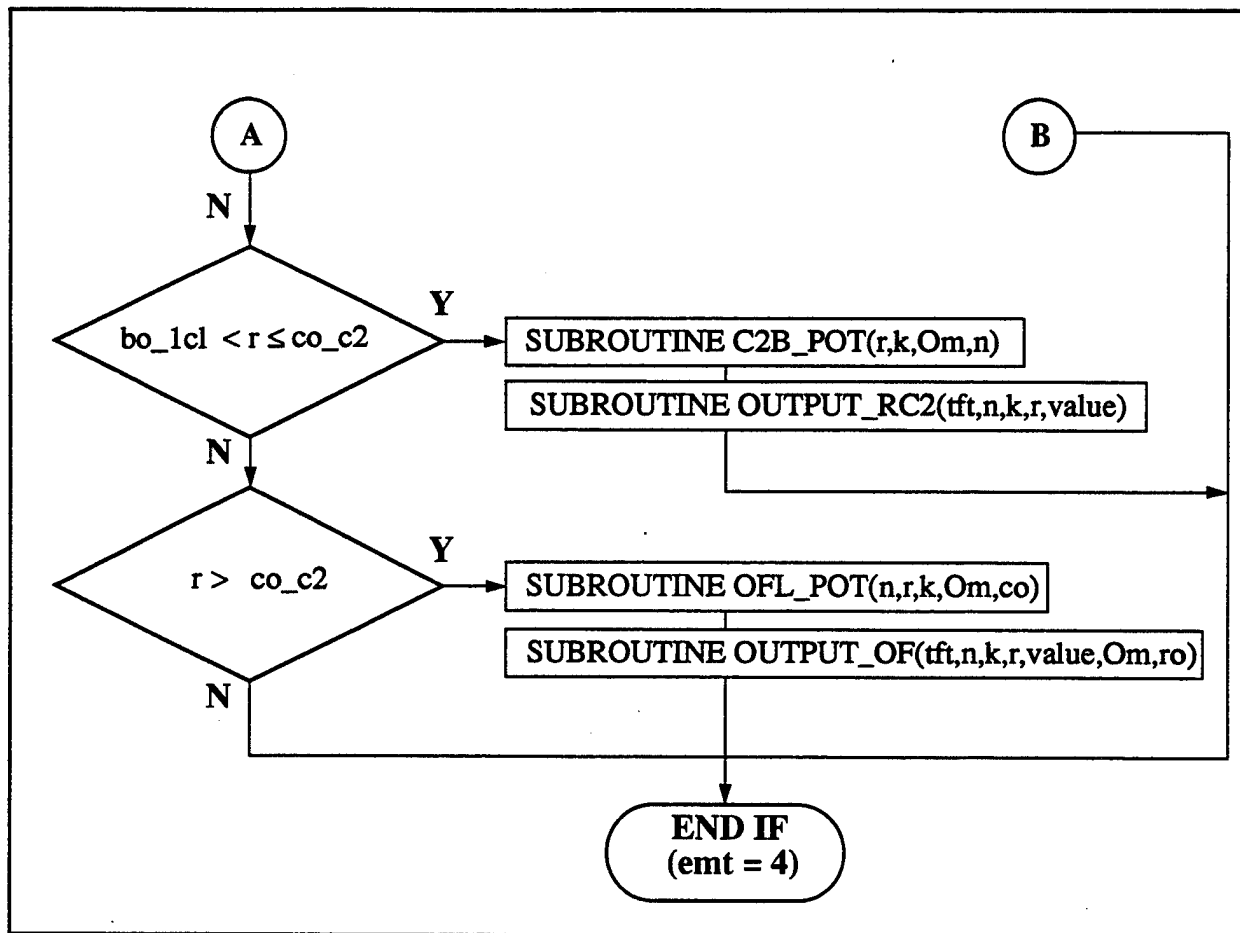


Figure 6. Flow Chart 4—Inner Fluid, Cylinder 1, Cylinder 2, and Outer Fluid for $emt = 4$ (Cont'd)

SUBPROGRAM LISTINGS

LISTING FOR mr2cf.f

mr2cf.f

Thu Oct 19 16:34:21 1995

```

1 C TEST PROGRAM "mr2cf.f FOR TESTING THE TWO CYLINDER INNER/OUTER
  FLUID
2 C SIMULATION 5/16/95
3
4 C TEST PROGRAM VARIABLES
5
6
7 C      character*1 msg
8      complex*16 a,z,Jn,In,Yn,Kn,mains
9      real*8 SI,bb
10     integer FA,msg,f
11
12
13 C *****
14
15 C VARIABLES NEEDED TO SIMULATE MAIN PROGRAM Xtota
16
17     integer iptmax
18     parameter (iptmax = 80000)
19     integer n,tft,exctype,jk,emt
20     real*8 r,ao_rod,mu_rod,r_rod
21     real*8 k,Om,co,ro,zeta_rod
22     complex*16 g(iptmax),E_rod
23     real*8 ri,ci
24
25     real*8 E_1cyl,zeta_1cyl,mu_1cyl,ao_1cyl,r_1cyl,h_1cyl
26     real*8 E_2cyl,zeta_2cyl,mu_2cyl,ao_2cyl,r_2cyl,h_2cyl
27
28 C *****
29
30 C NEW ADDITIONS TO MAIN PROGRAM
31
32     integer gamma,fac,iflag,size
33     real*8 psi
34     complex*16 cl_rod,ct_rod,sm(4,4)
35     complex*16 work(4,8),sminv(4,4)
36     complex*16 value
37     complex*16 Ec_rod
38     double complex cbessj,cbessi,cbessy,cbessk,cbessh1,cbessh2
39     double complex d1cbessj,d2cbessj
40     double complex d1cbessy,d2cbessy
41     double complex d1cbessk,d2cbessk
42     double complex d1cbessi,d2cbessi
43     double complex d1cbessh1,d1cbessh2

```

```

44
45 C ADDITIONS NEEDED FOR THE FIRST CYLINDER BEYOND THE ROD/FLUID
46
47     real*8 bo_c1
48     complex*16 smrc1(10,10),smrc1inv(10,10),workrc1(10,20)
49     complex*16 Ec_c1
50
51 C ADDITIONS NEEDED FOR THE SECOND CYLINDER BEYOND THE ROD/
    CYLINDER
52
53     real*8 co_c2
54     complex*16 smrc2(13,13),smrc2inv(13,13),workrc2(13,26)
55     complex*16 Ec_c2
56
57 C ADDITIONS NEEDED FOR CYLINDER WITH INNER AND OUTER FLUIDS
    ONLY
58
59     complex*16 smc1(7,7),smc1inv(7,7),workc1(7,14)
60
61
62
63 C *****
64
65 C DEFINITION FOR COMMON BLOCK /ROD/
66
67     complex*16 SP_rod,d1_SP_rod,d2_SP_rod
68     complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
69     complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
70     complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
71
72     common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
73     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
74     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
75
76 C *****
77
78 C *****
79
80 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
81
82     complex*16 OFSC,d1_OFSC,M_OF
83
84     common /OFLUID/ OFSC,d1_OFSC,M_OF
85
86 C *****
87

```

```

88 C *****
89
90 C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
91
92     complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
93     complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
94     complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
95     complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2
96
97     complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
98     complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
99     complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
100    complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
101
102    complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
103    complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
104    complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
105    complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2
106
107    complex*16 lame_c1,shear_c1,cl_c1,ct_c1
108    complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
109
110    common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
111    1      SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
112    1      SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
113    1      SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
114    1      VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
115    1      VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
116    1      VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
117    1      VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
118    1      VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
119    1      VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
120    1      VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
121    1      VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
122    1      lame_c1,shear_c1,cl_c1,ct_c1,
123    1      A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
124
125 C *****
126
127

```

128 C "CYLINDER 2 VARIABLES NEEDED TO EXPAND BEYOND ROD/CYLINDER
CASE **

129

130 C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/

131

132 complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1

133 complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2

134 complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1

135 complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2

136

137 complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1

138 complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2

139 complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1

140 complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2

141

142 complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1

143 complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2

144 complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1

145 complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2

146

147 complex*16 lame_c2,shear_c2,cl_c2,ct_c2

148

149 complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

150

151 common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1,

152 1 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,

153 1 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,

154 1 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,

155 1 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,

156 1 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,

157 1 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,

158 1 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,

159 1 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,

160 1 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,

161 1 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,

162 1 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,

163 1 lame_c2,shear_c2,cl_c2,ct_c2,

164 1 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

165

166 C *****

167

168 C *****

169

```

170 C DEFINITIONS FOR COMMON BLOCK /IFLUID/
171
172     complex*16 IFSC,d1_IFSC,D_IF
173
174     common /IFLUID/ IFSC,d1_IFSC,D_IF
175
176
177 C *****
178
179
180 C "ROD CASE" VARIABLES NEEDED TO SIMULATE MAIN PROGRAM
181
182     E_rod = 7.20D2
183     mu_rod = 0.17D0
184     r_rod = 2600.0D0
185     zeta_rod = 0.0D0
186
187 C "ROD CASE" MAIN PROGRAM CALCULATION SECTION ADDITIONS *****
188
189     Ec_rod = dcmplx(1.0D0,zeta_rod)*E_rod
190     lame_rod = Ec_rod*mu_rod/((1.0D0+mu_rod)*(1.0D0-2.0D0*mu_rod))
191     shear_rod = Ec_rod/(2.0D0*(1.0D0 + mu_rod))
192     cl_rod = zsqrt((lame_rod + 2.0D0* shear_rod)/r_rod)
193     ct_rod = zsqrt(shear_rod / r_rod)
194
195 C *****
196
197 C "ROD/CYLINDER" VARIABLES NEEDED TO SIMULATE MAIN PROGRAM
198
199     E_1cyl = 7.1D10
200     zeta_1cyl = 0.0D0
201     mu_1cyl = 0.33D0
202     ao_1cyl = 125.0D-6
203     r_1cyl = 2700.0D0
204     h_1cyl = 375.0D-6
205
206
207 C "ROD/CYL1 CASE" MAIN PROGRAM CALCULATION SECTION ADDITIONS **
208
209     Ec_c1 = dcmplx(1.0D0,zeta_1cyl)*E_1cyl
210     lame_c1 = Ec_c1*mu_1cyl/((1.0D0+mu_1cyl)*(1.0D0-2.0D0*mu_1cyl))
211     shear_c1 = Ec_c1/(2.0D0*(1.0D0+mu_1cyl))
212     cl_c1 = zsqrt((lame_c1 + 2.0D0*shear_c1)/r_1cyl)
213     ct_c1 = zsqrt(shear_c1/r_1cyl)
214     bo_c1 = ao_1cyl + h_1cyl
215

```

```

216 C *****
217
218 C *****
219
220 C "TWO CYLINDER" VARIABLES NEEDED TO SIMULATE MAIN PROGRAM
221
222     E_2cyl = 7.1D10
223     zeta_2cyl = .0D0
224     mu_2cyl = .33D0
225     ao_2cyl = 400.0D-6
226     r_2cyl = 2700.0D0
227     h_2cyl = 100.0D-6
228
229
230 C "ROD/CYL1 CASE" MAIN PROGRAM CALCULATION SECTION ADDITIONS *
231
232     Ec_c2 = dcplx(1.0D0,zeta_2cyl)*E_2cyl
233     lame_c2 = Ec_c2*mu_2cyl/((1.0D0+mu_2cyl)*(1.0D0-2.0D0*mu_2cyl))
234     shear_c2 = Ec_c2/(2.0D0*(1.0D0+mu_2cyl))
235     cl_c2 = zsqrt((lame_c2 + 2.0D0*shear_c2)/r_2cyl)
236     ct_c2 = zsqrt(shear_c2/r_2cyl)
237     co_c2 = ao_2cyl + h_2cyl
238
239 C *****
240
241
242
243 C RADIAL STRESS/(Pr or Px)          tft = 0
244 C LONGITUDINAL STRESS              tft = 1
245 C AXIAL DISPLACEMENT              tft = 2
246 C THETA DISPLACEMENT              tft = 3
247 C RADIAL DISPLACEMENT             tft = 4
248 C LONGITUDINAL STRAIN e11/(Pr or Px) tft = 5
249 C THETA STRAIN ett/(Pr or Px)      tft = 6
250 C RADIAL STRAIN err/(Pr or Px)     tft = 7
251 C                                  tft = 8
252 C OPTIC exx=0@k=0 ((dp/p)(r))/(Pr or Px) tft = 9
253 C OPTIC exx=const@k=0 ((dp/p)(r))/(Pr or Px) tft = 10
254
255 C RADIAL EXCITATION      exctype = 1
256 C AXIAL EXCITATION      exctype = 0
257
258 C INNER FLUID/CYLINDER/OUTER FLUID      emt = 1
259 C ROD AND OUTER FLUID                  emt = 2
260 C ROD/CYLINDER/FLUID                  emt = 3
261 C INNER FLUID/TWO CYLINDERS/OUTER FLUID emt = 4

```

```

262
263
264      k    = 0.1D0
265      Om   = 6000.0D0
266      ao_rod = 125.0D-6
267      r    = 150.0D-6
268      n    = 0
269      tft   = 5
270      exctype = 1
271      emt   = 3
272      jk    = 1
273
274      co    = 1500.0D0
275      ro    = 1000.0D0
276
277      ci          = 150.0D0
278      ri          = 1.0D0
279
280      g(jk) = 3.0D0
281      size  = 3
282      iflag = 0
283
284      mains = zsqrt(k**2 - (Om**2/cl_rod**2))*ao_rod
285
286      msg = 0
287
288      do while (msg .ne. 1)
289
290          print *, 'Enter tft,k,f,emt: (where emt=1 1CYLINDER,
291 1 emt=2 ROD, emt=3 ROD&CYL, emt=4 2CYLINDERS,
292 1 using mr2cf.f)'
293
294          read(*,20)tft,k,f,emt
295 20      format(i2,f10.4,i6,i1)
296
297          Om = 2.0D0*3.14D0*f
298
299          write(*,10)n,r,ao_1cyl,bo_c1,co_c2,tft
300 10      format(/,'MAIN PROGRAM n = ',i2,' r = ',E15.7,
301 1' ao_1cyl = ',E15.7,' bo_c1 = ',E15.7,' co_c2 = ',E15.7,/
302 1,'tft = ',i2)
303
304
305

```

```

306 C INNER FLUID/CYLINDER 1/OUTER FLUID (emt=1) *****
307     if (emt .eq. 1) then
308
309
310         CALL C1A_POT(ao_1cyl,k,Om,n)
311         CALL C1B_POT(bo_c1,k,Om,n)
312         CALL OFL_POT(n,bo_c1,k,Om,co)
313         CALL IFL_POT(n,ao_1cyl,k,Om,ci)
314         CALL SYS_MATRIX_C1(n,k,ao_1cyl,bo_c1,
315     1Om,ro,co,ri,ci,smc1)
316         CALL ABC_C1_INVERT(n,exctype,smc1,ao_1cyl,bo_c1)
317
318         if (r .le. ao_1cyl) then
319
320 C     PRESSURE AND VELOCITY TRANSFER FUNCTIONS (INNER FLUID)
321         CALL IFL_POT(n,r,k,Om,ci)
322         CALL OUTPUT_IF(tft,n,k,r,value,Om,ri)
323
324         elseif (r .le. bo_c1 .and. r .gt. ao_1cyl) then
325
326         CALL C1A_POT(r,k,Om,n)
327         CALL OUTPUT_RC1(tft,n,k,r,value)
328
329 C     PRESSURE AND VELOCITY TRANSFER FUNCTIONS (OUTER FLUID)
330         elseif (r .gt. co_c2) then
331
332         CALL OFL_POT(n,r,k,Om,co)
333
334         write(*,400)d1_OFSC,OFSC,M_OF
335 400     format(/,'d1_OFSC = '2E15.7,' OFSC = '2E15.7,' M_OF = ',
336     12E15.7,/)
337
338         CALL OUTPUT_OF(tft,n,k,r,value,Om,ro)
339
340         endif
341
342         if(tft .gt. 1)then
343             value = value*(1.0D-6)
344         endif
345
346         g(jk) = value
347 C *****
348
349

```

```

350 C SOLID CYLINDER/OUTER FLUID (emt=2) *****
351     elseif (emt .eq. 2) then
352
353         CALL ROD_POT(ao_rod,k,Om,n,cl_rod,ct_rod)
354         CALL OFL_POT(n,ao_rod,k,Om,co)
355         CALL SYS_MATRIX_ROD(n,k,ao_rod,Om,ro,co,sm)
356         CALL ABC_ROD_INVERT(exctype,sm)
357
358         if (r .le. ao_rod) then
359
360             CALL ROD_POT(r,k,Om,n,cl_rod,ct_rod)
361             CALL OUTPUT(tft,n,k,r,value)
362
363 C     PRESSURE AND VELOCITY TRANSFER FUNCTIONS
364     elseif (r .gt. ao_rod) then
365
366         CALL OFL_POT(n,r,k,Om,co)
367         write(*,400)d1_OFSC,OFSC,M_OF
368         CALL OUTPUT_OF(tft,n,k,r,value,Om,ro)
369
370     endif
371
372     if(tft .gt. 1)then
373         value = value*(1.0D-6)
374     endif
375
376     g(jk) = value
377 C *****
378

```

```

379 C SOLID CYLINDER/CYLINDER 1/ OUTER FLUID (emt=3) *****
380     elseif (emt .eq. 3)then
381
382         CALL C1A_POT(ao_1cyl,k,Om,n)
383         CALL C1B_POT(bo_c1,k,Om,n)
384         CALL ROD_POT(ao_rod,k,Om,n,cl_rod,ct_rod)
385         CALL OFL_POT(n,bo_c1,k,Om,co)
386         CALL SYS_MATRIX_RC1(n,k,ao_rod,ao_1cyl,bo_c1,Om,ro,co
387 1,smrc1)
388         CALL ABC_RC1_INVERT(n,exctype,smrc1,bo_c1)
389
390     if (r .le. ao_rod) then
391
392         CALL ROD_POT(r,k,Om,n,cl_rod,ct_rod)
393         CALL OUTPUT(tft,n,k,r,value)
394
395     elseif (r .le. bo_c1 .and. r .gt. ao_rod) then
396
397         CALL C1A_POT(r,k,Om,n)
398         CALL OUTPUT_RC1(tft,n,k,r,value)
399
400 C     PRESSURE AND VELOCITY TRANSFER FUNCTIONS
401     elseif (r .gt. bo_c1) then
402
403         CALL OFL_POT(n,r,k,Om,co)
404         write(*,400)d1_OFSC,OFSC,M_OF
405
406         CALL OUTPUT_OF(tft,n,k,r,value,Om,ro)
407
408     endif
409
410     if(tft .gt. 1)then
411         value = value*(1.0D-6)
412     endif
413
414     g(jk) = value
415
416 C *****
417

```

```

418 C INNER FLUID/CYLINDER 1/CYLINDER 2/OUTER FLUID *****
419     elseif (emt .eq. 4)then
420
421         CALL C1A_POT(ao_1cyl,k,Om,n)
422         CALL C1B_POT(bo_c1,k,Om,n)
423         CALL C2B_POT(ao_2cyl,k,Om,n)
424         CALL C2C_POT(co_c2,k,Om,n)
425         CALL IFL_POT(n,ao_1cyl,k,Om,ci)
426
427         write(*,410)d1_IFSC,IFSC,D_IF
428 410     format(/,'d1_IFSC = ',2E15.7,' IFSC = ',2E15.7,' D_IF = ',
429     12E15.7,/)
430         CALL OFL_POT(n,co_c2,k,Om,co)
431         CALL SYS_MATRIX_RC2(n,k,ao_1cyl,bo_c1,co_c2,Om,ro,co
432     1,ri,ci,smrc2)
433         CALL ABC_RC2_INVERT(n,exctype,smrc2,ao_1cyl,co_c2)
434
435         write(*,400)d1_OFSC,OFSC,M_OF
436
437         if (r .le. ao_1cyl) then
438
439 C         PRESSURE AND VELOCITY TRANSFER FUNCTIONS (INNER FLUID)
440             CALL IFL_POT(n,r,k,Om,ci)
441             CALL OUTPUT_IF(tft,n,k,r,value,Om,ri)
442
443         elseif (r .le. bo_c1 .and. r .gt. ao_1cyl) then
444
445             CALL C1A_POT(r,k,Om,n)
446             CALL OUTPUT_RC1(tft,n,k,r,value)
447
448         elseif (r .le. co_c2 .and. r .gt. bo_c1) then
449
450             CALL C2B_POT(r,k,Om,n)
451             CALL OUTPUT_RC2(tft,n,k,r,value)
452
453 C         PRESSURE AND VELOCITY TRANSFER FUNCTIONS (OUTER FLUID)
454             elseif (r .gt. co_c2) then
455
456                 CALL OFL_POT(n,r,k,Om,co)
457
458             write(*,400)d1_OFSC,OFSC,M_OF
459
460                 CALL OUTPUT_OF(tft,n,k,r,value,Om,ro)
461
462             endif
463

```

```

464      if(tft .gt. 1)then
465          value = value*(1.0D-6)
466      endif
467
468      g(jk) = value
469
470      endif
471
472      g(jk) = 10.0D0*log10(cdabs((value)**2))
473
474 C *****
475
476      write (*,51)value
477 51      format('SUB OUTPUT THE ANSWER IS value = '2e15.7)
478      write (*,60)g(jk)
479 60      format('SUB OUTPUT THE ANSWER IS g(jk) = '2e15.7)
480      write (*,70)k,Om
481 70      format('SUB OUTPUT          k = 'e15.7,
482 1' Om = 'f7.2,/)
483
484
485
486
487      print *, 'Type 1 if you wish to quit'
488      read(*,4) msg
489 4      format(i1)
490
491      end do
492      stop
493      end
494
495
496

```

LISTING FOR cbessl.f

cbessl.f

Thu Oct 19 16:47:22 1995

```
1
2
3
4     function gamma(n)
5     integer i,n,sum,gamma
6         if(n .eq. 0 .or. n .eq. 1) then
7             gamma = 1
8         else
9             sum=n
10            do 10, i=1, n-1
11                sum = sum*(n-i)
12    10        continue
13            gamma = sum
14        endif
15    return
16    end
17
18
19
20
21     function fac(n)
22     integer n,sum,fac,i
23         if(n .eq. 0 .or. n .eq. 1) then
24             fac = 1
25         else
26             sum=n
27             do 10, i=1, n-1
28                 sum = sum*(n-i)
29    10        continue
30            fac = sum
31        endif
32    return
33    end
34
35
36
37
```

```

38      function psi(n)
39      integer n,na,i
40      real*8 sum,psi,euler
41      parameter (euler=.5772156649015328606)
42      sum = 0.0
43      na = iabs(n)
44      if (na .eq. 1) then
45          psi = - euler
46      else
47          do 10, i=1, na-1
48              sum = sum + (1.0/i)
49 10      continue
50          psi = -euler + sum
51      endif
52      return
53      end

```

```

54
55
56
57
58      double complex function cbessj(n,a,r)
59
60      integer n,limit,j,k,na,fac
61      real*8 r,pi,zm
62      complex*16 a,z,sum,total,Am,Bm
63      double complex cbessj
64
65      real*8 fn2
66      complex*16 a1,a2,a3,a4,b1,b2,b3,b4,ez,ez2
67
68      parameter (pi=3.1415926535897932384D0)
69
70      z = a*r
71      na = iabs(n)
72      zm = cdabs(z)
73
74
75      fn2 = 4.0D0*n**2
76      ez = 8.0D0*z
77      ez2 = (8.0D0*z)**2
78
79      if (zm .le. 3.0) then
80
81          limit = 10
82          total = (0.0D0,0.0D0)
83

```

```

84 C Abramowitz and Stegun Equation 9.1.10
85
86     do 10, k=0, limit
87         j = na + k
88         sum = ((-0.25D0*z**2)**k)/(fac(k)*fac(j))
89         total = total + sum
90 10    continue
91         cbessj = ((z/2.0D0)**na)*total
92
93         if (n .lt. 0) then
94             cbessj = ((-1.0D0)**na)*cbessj
95
96
97         end if
98
99     else
100
101 C Large Argument Calculation Korn and Korn 21.8-44 and 45
102
103
104     a1 = (fn2 - 1.0D0)*(fn2 - 9.0D0)/(2.0D0*ez2)
105
106     a2 = a1*(fn2 - 25.0D0)*(fn2 - 49.0D0)/(ez2*12.0D0)
107
108     a3 = a2*(fn2 - 81.0D0)*(fn2 - 121.0D0)/(ez2*30.0D0)
109
110     a4 = a3*(fn2 - 169.0D0)*(fn2 - 225.0D0)/(ez2*56.0D0)
111
112
113     Am = 1.0D0 - a1 + a2 - a3 + a4
114
115     b1 = (fn2 - 1.0D0)/ez
116
117     b2 = b1*(fn2 - 9.0D0)*(fn2 - 25.0D0)/(ez2*6.0D0)
118
119     b3 = b2*(fn2 - 49.0D0)*(fn2 - 81.0D0)/(ez2*20.0D0)
120
121     b4 = b3*(fn2 - 121.0D0)*(fn2 - 169.0D0)/(ez2*42.0D0)
122
123
124     Bm = b1 - b2 + b3 - b4
125
126     cbessj = cdsqrt(2.0D0/(pi*z))*(Am*cdcos(z - n*pi/
127 1      2.0D0 - pi/4.0D0) - Bm*cdsin(z - n*pi/2.0D0 - pi/
128 1      4.0D0))
129

```

```

130
131     endif
132
133
134
135     return
136     end
137
138
139
140
141     double complex function cbessi(n,a,r)
142
143     integer n,limit,j,k,na,fac
144     real*8 r
145     complex*16 a,z,sum,total
146     double complex cbessi
147
148     z = a*r
149     na=iabs(n)
150     limit = 10
151     total = (0.0,0.0)
152
153     do 10, k=0, limit
154         j = na + k
155         sum = ((.25*z**2)**k)/(fac(k)*fac(j))
156         total = total + sum
157 10    continue
158     cbessi = ((z/2)**na)*total
159
160
161     return
162     end
163
164
165
166

```

```

167      double complex function cbessy(n,a,r)
168
169      integer n,na,limit,j,k,l,m,fac
170      real*8 r,pi,psi,zm
171      complex*16 a,z,z2,part1,part2,part3,sum1,sum2
172      complex*16 total1,total2,Am,Bm
173      double complex cbessj,cbessy
174
175      parameter (pi=3.1415926535897932384)
176
177      z = a*r
178      z2 = .5*z
179      na = iabs(n)
180      zm = cdabs(z)
181
182      IF (zm .le. 3.0) THEN
183
184      C Abramowitz and Stegun Equation 9.1.11
185
186      part1 = (0.,0.)
187      part2 = (0.,0.)
188      part3 = (0.,0.)
189      total1 = (0.,0.)
190      total2 = (0.,0.)
191      limit = 10
192
193      do 10, k = 0, na-1
194          j = na-k-1
195          sum1 = (fac(j)/fac(k))*(.25*z**2)**k
196          total1 = total1 + sum1
197 10      continue
198
199      part1 = (-((.5*z)**-na)/pi)*total1
200
201      part2 = (2.0/pi)*cdlog(z2)*cbessj(na,a,r)
202
203      do 20, k = 0, limit
204          j = k+1
205          l = na+k+1
206          m = na+k
207          sum2 = (psi(j)+psi(l))*(-.25*z**2)**k/(fac(k)*fac(m))
208          total2 = total2+sum2
209 20      continue
210
211      part3 = ((-(.5*z)**na)/pi)*total2
212

```

```

213      cbessy = part1+part2+part3
214
215
216      if (n .lt. 0) then
217          cbessy = ((-1)**na)*cbessy
218      end if
219
220
221      ELSE
222
223      C Large Argument Calculation Korn and Korn 21.8-44 & 45
224
225          Am = 1.0 - (4.0*n**2 - 1.0)*(4*n**2 - 9.0)/(2.0*(8*z)**2)
226      1      + (4.0*n**2-1.0)*(4.0*n**2-9.0)*(4.0*n**2-25.0)*
227      1      (4.0*n**2-49.0)/(24.0*(8.0*z)**4)
228
229          Bm = (4.0*n**2-1.0)/(8.0*z) - (4.0*n**2-1.0)*(4.0*n**2-9.0)*
230      1      (4.0*n**2-25.0)/(6.0*(8.0*z)**3)
231
232          cbessy = cdsqrt(2.0/(pi*z))*(Am*cdsin(z - n*pi/2.0 - pi/4.0) +
233      1      Bm*cdcos(z - n*pi/2.0 - pi/4.0))
234
235      ENDIF
236
237
238      return
239      end
240
241
242
243

```

```

244      double complex function cbessk(n,a,r)
245
246      integer n,na,limit,j,k,l,m,fac
247      real*8 r,pi,psi
248      complex*16 a,z,z2,part1,part2,part3,sum1,sum2,total1,total2
249      double complex cbessi,cbessk
250
251      parameter (pi=3.1415926535897932384)
252
253      z = a*r
254      z2 = .5*z
255      na = iabs(n)
256
257
258      part1 = (0.,0.)
259      part2 = (0.,0.)
260      part3 = (0.,0.)
261      total1 = (0.,0.)
262      total2 = (0.,0.)
263      limit = 10
264
265      do 10, k = 0, na-1
266          j = na-k-1
267          sum1 = (fac(j)/fac(k))*(-.25*z**2)**k
268          total1 = total1 + sum1
269 10    continue
270
271      part1 = (((.5*z)**-na)/2.)*total1
272
273      part2 = ((-1)**(na+1))*cdlog(z2)*cbessi(na,a,r)
274
275      do 20, k = 0, limit
276          j = k+1
277          l = na+k+1
278          m = na+k
279          sum2 = (psi(j)+psi(l))*(.25*z**2)**k/(fac(k)*fac(m))
280          total2 = total2+sum2
281 20    continue
282
283      part3 = ((-1)**na)*((.5*z)**na)*.5*total2
284
285      cbessk = part1+part2+part3
286
287      return
288      end
289

```

```
290
291
292
293     double complex function d1cbessk(n,a,r)
294
295     real*8 r
296     complex*16 a
297     integer n
298     double complex cbessk,d1cbessk
299
300     d1cbessk = -a*cbessk(n+1,a,r)+(n/r)*cbessk(n,a,r)
301
302     return
303     end
304
305
306
307
308     double complex function d2cbessk(n,a,r)
309
310     real*8 r
311     complex*16 a
312     integer n
313     double complex d2cbessk
314
315     d2cbessk = (1.0,1.0)
316
317     return
318     end
319
320
321
322
323     double complex function d1cbessj(n,a,r)
324
325     real*8 r
326     complex*16 a
327     integer n
328     double complex cbessj,d1cbessj
329
330     d1cbessj = -a*cbessj(n+1,a,r)+(n/r)*cbessj(n,a,r)
331
332     return
333     end
334
335
```

```

336
337
338     double complex function d2cbessj(n,a,r)
339
340     real*8 r
341     complex*16 a
342     integer n
343     double complex cbessj,d2cbessj
344
345
346     d2cbessj = ((a**2)/4.0D0)*(cbessj(n-2,a,r) - 2.0D0*
347 1      cbessj(n,a,r) + cbessj(n+2,a,r))
348
349
350     return
351     end
352
353
354
355
356     double complex function d1cbessy(n,a,r)
357
358     real*8 r
359     complex*16 a
360     integer n
361     double complex cbessy,d1cbessy
362     complex*16 Y2,Y3
363
364
365     IF (n .eq. 0) THEN
366
367         d1cbessy = - a*cbessy(1,a,r)
368
369     ELSEIF (n .eq. 1) THEN
370
371         Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
372         d1cbessy = - a*Y2 + cbessy(1,a,r)/r
373
374     ELSEIF (n .eq. 2) THEN
375
376         Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
377         Y3 = (8.0D0/((a*r)**2) - 1.0D0)*cbessy(1,a,r) -
378 1      4.0D0*cbessy(0,a,r)/(a*r)
379         d1cbessy = - a*Y3 + 2.0D0*Y2/r
380
381

```

```

382      ELSEIF (n .eq. -1) THEN
383
384          d1cbessy = -a*cbessy(0,a,r) - cbessy(-1,a,r)/r
385
386      ELSEIF (n .eq. -2) THEN
387
388          Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
389          d1cbessy = -a*cbessy(-1,a,r) - 2.0D0*Y2/r
390
391
392      ENDIF
393
394
395      return
396      end
397
398
399
400
401      double complex function d2cbessy(n,a,r)
402
403      real*8 r
404      complex*16 a
405      integer n
406      double complex cbessy,d2cbessy
407      complex*16 Y2,Y3,Y4
408
409
410
411      IF (n .eq. 0) THEN
412
413          Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
414          d2cbessy = a**2*Y2 - a*cbessy(1,a,r)/r
415
416      ELSEIF (n .eq. 1) THEN
417
418          Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
419          Y3 = (8.0D0/((a*r)**2) - 1.0D0)*cbessy(1,a,r) -
420      1          4.0D0*cbessy(0,a,r)/(a*r)
421          d2cbessy = a**2*Y3 - 3.0D0*a*Y2/r
422
423      ELSEIF (n .eq. 2) THEN
424
425          Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
426          Y3 = (8.0D0/((a*r)**2) - 1.0D0)*cbessy(1,a,r) -
427      1          4.0D0*cbessy(0,a,r)/(a*r)

```

```

428      Y4 = (1.0D0 - 24.0D0/((a*r)**2))*cbessy(0,a,r)+
429      1      (48.0D0/((a*r)**3) - 8.0D0/(a*r))*cbessy(1,a,r)
430      d2cbessy = a*Y3/r + (2.0D0/r**2 - a**2)*Y2
431
432
433      ELSEIF (n .eq. -1) THEN
434
435      d2cbessy = a**2*cbessy(1,a,r) + a*cbessy(0,a,r)/r +
436      1      2.0D0*cbessy(-1,a,r)/(r**2)
437
438      ELSEIF (n .eq. -2) THEN
439
440      Y2 = -cbessy(0,a,r) + 2.0D0*cbessy(1,a,r)/(a*r)
441      d2cbessy = a**2*cbessy(0,a,r) + 3.0D0*a*cbessy(-1,a,r)/r +
442      1      6.0D0*Y2/(r**2)
443
444
445      ENDIF
446
447      return
448      end
449
450
451
452
453      double complex function d1cbessi(n,a,r)
454
455      real*8 r
456      complex*16 a
457      integer n
458      double complex cbessi,d1cbessi
459
460      d1cbessi = a*cbessi(n+1,a,r)+(n/r)*cbessi(n,a,r)
461
462      return
463      end
464
465
466
467

```

```
468      double complex function d2cbessi(n,a,r)
469
470      real*8 r
471      complex*16 a
472      integer n
473      double complex d2cbessi
474
475      d2cbessi = (1.0,1.0)
476
477      return
478      end
479
480
481
482
483      double complex function cbessh1(n,a,r)
484      real*8 r
485      complex*16 a
486      integer n
487      double complex cbessj,cbessy,cbessh1
488
489      cbessh1 = cbessj(n,a,r)+(0.0, 1.0)*cbessy(n,a,r)
490
491      return
492      end
493
494
495
496
497      double complex function cbessh2(n,a,r)
498      real*8 r
499      integer n
500      complex*16 a
501      double complex cbessj,cbessy,cbessh2
502
503      cbessh2 = cbessj(n,a,r)-(0.0, 1.0)*cbessy(n,a,r)
504
505      return
506      end
507
508
509
510
```

```
511      double complex function d1cbessh1(n,a,r)
512      integer n
513      real*8 r
514      complex a
515      double complex d1cbessj,d1cbessy,d1cbessh1
516
517          d1cbessh1 = d1cbessj(n,a,r) + (0.0,1.0)*d1cbessy
518 1          (n,a,r)
519
520      return
521      end
522
523
524
525
526      double complex function d1cbessh2(n,a,r)
527      integer n
528      real*8 r
529      complex a
530      double complex d1cbessj,d1cbessy,d1cbessh2
531
532          d1cbessh2 = d1cbessj(n,a,r) - (0.0,1.0)*d1cbessy
533 1          (n,a,r)
534
535      return
536      end
537
```

LISTING FOR rf.f

rf.f

Sat Jun 10 14:42:17 1995

```

1  C *****
2  C
3  C          subprogram "rf.f"
4  C
5  C This subprogram was written and developed by Mark S. Peloquin
6  C
7  C at NUWCDETNLON 6/10/95. As of 6/10/95, there are no known bugs.
8  C
9  C Please notify the author if bugs are found (203) 440-5433.
10 C
11 C *****
12
13
14     SUBROUTINE ROD_POT(r,k,Om,n,cl_rod,ct_rod)
15
16
17 C EXTERNAL VARIABLES
18
19     integer n
20     real*8 r,k,Om
21     complex*16 cl_rod,ct_rod
22
23     common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
24     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
25     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
26
27     complex*16 SP_rod,d1_SP_rod,d2_SP_rod
28     complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
29     complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
30     complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
31
32
33     double complex cbessj
34     double complex d1cbessj,d2cbessj
35
36 C INTERNAL VARIABLES
37
38     complex*16 p,q
39
40
41     p = zsqrt((Om**2/cl_rod**2) - k**2)
42     d2_SP_rod = d2cbessj(n,p,r)
43     d1_SP_rod = d1cbessj(n,p,r)
44     SP_rod = cbessj(n,p,r)

```

```

45
46
47     q = zsqrt((Om**2/ct_rod**2) - k**2)
48     d2_VXP_rod = d2cbessj(n,q,r)
49     d1_VXP_rod = d1cbessj(n,q,r)
50     VXP_rod = cbessj(n,q,r)
51
52     d2_VRTP_rod = d2cbessj(n+1,q,r)
53     d1_VRTP_rod = d1cbessj(n+1,q,r)
54     VRTP_rod = cbessj(n+1,q,r)
55
56
57     return
58     end
59
60
61
62
63
64     SUBROUTINE SYS_MATRIX_ROD(n,k,ao_rod,Om,ro,co,sm)
65
66 C EXTERNAL VARIABLES
67
68     integer n
69     real*8 k,ao_rod,ro,co,Om
70     complex*16 sm(4,4)
71
72     common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
73     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
74     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
75
76
77     complex*16 SP_rod,d1_SP_rod,d2_SP_rod
78     complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
79     complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
80     complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
81
82 C *****
83
84 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
85
86     common /OFLUID/ OFSC,d1_OFSC,M_OF
87
88     complex*16 OFSC,d1_OFSC,M_OF
89
90 C *****

```

```

91
92 C INTERNAL VARIABLES
93
94     real*8 b2,k2
95     integer n2
96
97     b2 = ao_rod**2
98     n2 =  n**2
99     k2 =  k**2
100
101
102     sm(1,1) = (lame_rod + 2.0D0*shear_rod)*d2_SP_rod +
103 1         (lame_rod/ao_rod)*d1_SP_rod -
104 1         (n2/b2 + k2)*lame_rod*SP_rod
105
106     sm(1,2) = -2.0D0*n*shear_rod*VXP_rod/b2 +
107 1         2.0D0*shear_rod*n*d1_VXP_rod/ao_rod
108
109     sm(1,3) = (0.0D0,1.0D0)*2.0D0*shear_rod*k*d1_VRTP_rod
110
111     sm(2,1) = (0.0D0,1.0D0)*2.0D0*shear_rod*k*d1_SP_rod
112
113     sm(2,2) = (0.0D0,1.0D0)*k*n*VXP_rod*shear_rod/ao_rod
114
115     sm(2,3) = shear_rod*(VRTP_rod*(n/b2 - k2 + 1.0D0/b2) -
116 1         d1_VRTP_rod*(n/ao_rod + 1.0D0/ao_rod) - d2_VRTP_rod)
117
118     sm(3,1) = 2.0D0*shear_rod*((n/b2)*SP_rod - (n/ao_rod)*
119 1         d1_SP_rod)
120
121     sm(3,2) = shear_rod*(-d2_VXP_rod + (1.0D0/ao_rod)*
122 1         d1_VXP_rod - (n2/b2)*VXP_rod)
123
124     sm(3,3) = (0.0D0,1.0D0)*(d1_VRTP_rod - VRTP_rod*
125 1         (1.0D0/ao_rod + n/ao_rod))*k*shear_rod
126
127 C *****
128 C THIS TERM WILL CHANGE SIGN IF THE PRESSURE CONDITION IS
    REVERSED
129
130     sm(1,4) = +Om**2*ro*OFSC
131
132 C *****
133     sm(2,4) = (0.0D0,0.0D0)
134
135     sm(3,4) = (0.0D0,0.0D0)

```

```

136
137      sm(4,1) = d1_SP_rod
138
139      sm(4,2) = n*VXP_rod/ao_rod
140
141      sm(4,3) = (0.0D0,1.0D0)*k*VRTP_rod
142
143      sm(4,4) = -d1_OFSC
144
145
146      return
147      end
148
149
150
151
152      SUBROUTINE ABC_ROD_SOLVE(exctype,sm)
153
154 C EXTERNAL VARIABLES
155
156      integer exctype
157      complex*16 sm(3,3)
158
159      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
160      1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
161      1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
162
163
164      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
165      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
166      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
167      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
168
169 C INTERNAL VARIABLES
170
171      complex*16 detA
172
173 C RADIAL EXCITATION exctype = 1
174 C AXIAL EXCITATION exctype = 0
175
176      if (exctype .eq. 1) then
177          A1_rod = sm(2,2)*sm(3,3) - sm(3,2)*sm(2,3)
178          B1_rod = -sm(2,1)*sm(3,3) + sm(3,1)*sm(2,3)
179          C1_rod = sm(2,1)*sm(3,2) - sm(3,1)*sm(2,2)
180      elseif (exctype .eq. 0) then
181          A1_rod = -sm(1,2)*sm(3,3) - sm(3,2)*sm(1,3)

```

```

182      B1_rod = sm(1,1)*sm(3,3) + sm(3,1)*sm(1,3)
183      C1_rod = -sm(1,1)*sm(3,2) - sm(3,1)*sm(1,2)
184  endif
185
186      detA = (sm(1,1)*sm(2,2)*sm(3,3) +
187      1sm(1,2)*sm(2,3)*sm(3,1) + sm(2,1)*sm(3,2)*sm(1,3))-
188      1(sm(3,1)*sm(2,2)*sm(1,3) + sm(1,1)*sm(3,2)*sm(2,3)+
189      1sm(1,2)*sm(2,1)*sm(3,3))
190
191      A1_rod = A1_rod/detA
192      B1_rod = B1_rod/detA
193      C1_rod = C1_rod/detA
194
195  return
196  end
197
198
199
200
201  SUBROUTINE ABC_ROD_INVERT(exctype,sm)
202
203
204  C EXTERNAL VARIABLES
205
206      integer exctype
207      complex*16 sm(4,4)
208
209      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
210      1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
211      1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
212
213
214      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
215      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
216      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
217      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
218
219  C *****
220
221  C DEFINITIONS FOR COMMON BLOCK /OFLUID/
222
223      complex*16 OFSC,d1_OFSC,M_OF
224
225      common /OFLUID/ OFSC,d1_OFSC,M_OF
226
227  C *****

```

```
228
229 C INTERNAL VARIABLES
230
231     integer n,iflag
232     complex*16 sminv(4,4),work(4,8)
233
234     n = 4
235     iflag = 0
236
237     CALL MINV(sm,sminv,work,n,iflag)
238
239
240
241 C RADIAL EXCITATION exctype = 1
242 C AXIAL EXCITATION exctype = 0
243
244     if (exctype .eq. 1) then
245         A1_rod = -sminv(1,1)
246         B1_rod = -sminv(2,1)
247         C1_rod = -sminv(3,1)
248         M_OF = -sminv(4,1)
249     elseif (exctype .eq. 0) then
250         A1_rod = -sminv(1,2)
251         B1_rod = -sminv(2,2)
252         C1_rod = -sminv(3,2)
253         M_OF = -sminv(4,2)
254     endif
255
256
257     return
258     end
259
260
261
262
263
```

```

264      SUBROUTINE OUTPUT(tft,n,k,r,value)
265
266 C EXTERNAL VARIABLES
267
268      integer n,tft
269      real*8 k,r,zero
270      complex*16 value
271
272      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
273      1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
274      1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
275
276
277      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
278      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
279      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
280      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
281
282 C INTERNAL VARIABLES
283
284      integer n2
285      real*8 k2,r2,no,p11,p12
286      complex*16 dill,dpp,exx,err,Srr,Sxx,ett,ur,vr,wr
287
288      n2 = n**2
289      r2 = r**2
290      k2 = k**2
291
292      zero = 1.D-20
293      no = 1.46D0
294      p11 = .126D0
295      p12 = .27D0
296
297
298 C RADIAL STRESS/(Pr or Px)          tft = 0
299 C LONGITUDINAL STRESS              tft = 1
300 C AXIAL DISPLACEMENT              tft = 2
301 C THETA DISPLACEMENT              tft = 3
302 C RADIAL DISPLACEMENT              tft = 4
303 C LONGITUDINAL STRAIN      e11/(Pr or Px)    tft = 5
304 C THETA STRAIN      ett/(Pr or Px)          tft = 6
305 C RADIAL STRAIN      err/(Pr or Px)         tft = 7
306 C                                tft = 8
307 C OPTIC exx=0@k=0      ((dp/p)(r))/(Pr or Px) tft = 9
308 C OPTIC exx=const@k=0 ((dp/p)(r))/(Pr or Px) tft = 10
309

```

```

310      if (tft .eq. 0) then
311
312      dill = A1_rod*(d2_SP_rod+d1_SP_rod/r -
313 1      n2/r2*SP_rod - k2*SP_rod)
314      Srr = lame_rod*dill + 2.0D0*shear_rod*
315 1      (A1_rod*d2_SP_rod - n/r2*B1_rod*VXP_rod +
316 1      n/r*B1_rod*d1_VXP_rod + (0.0D0,1.0D0)*
317 1      k*C1_rod*d1_VRTP_rod)
318
319      value = Srr
320
321
322 C LONGITUDINAL STRESS USING (EQ 96)
323      elseif(tft .eq. 1) then
324
325      dill = A1_rod*(d2_SP_rod+d1_SP_rod/r -
326 1      n2/r2*SP_rod - k2*SP_rod)
327
328      Sxx = lame_rod*dill + 2.0D0*shear_rod*(
329 1      -k2*A1_rod*SP_rod - (0.0D0,1.0D0)*k*C1_rod*
330 1      (d1_VRTP_rod + VRTP_rod*(n + 1.0D0)/r))
331
332      value = Sxx
333
334      elseif(tft .eq. 2) then
335
336      wr = (0.0D0,1.0D0)*k*A1_rod*SP_rod -
337 1      C1_rod*(d1_VRTP_rod + 1.0D0/r*VRTP_rod + n/r*VRTP_rod)
338      value = wr
339
340      elseif(tft .eq. 3) then
341
342      vr = -n*A1_rod*SP_rod/r - B1_rod*d1_VXP_rod +
343 1      (0.0D0,1.0D0)*k*C1_rod*VRTP_rod
344      value = vr
345
346      elseif(tft .eq. 4) then
347
348      ur = A1_rod*d1_SP_rod + n*B1_rod*VXP_rod/r +
349 1      (0.0D0,1.0D0)*k*C1_rod*VRTP_rod
350      value = ur
351
352      elseif(tft .eq. 5) then
353
354      exx = -k2*A1_rod*SP_rod -
355 1      (0.0D0,1.0D0)*k*C1_rod*d1_VRTP_rod -

```

```

356 1      (0.0D0,1.0D0)*k*C1_rod*VRTP_rod*(1.0D0 + n)/r
357
358      if (zabs(exx) .lt. dabs(zero)) then
359          value = zero
360      endif
361
362      value = exx
363
364      elseif(tft .eq. 6) then
365
366          ett = A1_rod*(1.0D0/r*d1_SP_rod - n2/r2 * SP_rod) +
367 1          B1_rod*(n/r2*VXP_rod - n/r * d1_VXP_rod) +
368 1          (0.0D0,1.0D0)*C1_rod*k*VRTP_rod/r*(1.0D0 + n)
369
370      if (zabs(ett) .lt. dabs(zero)) then
371          value = zero
372      endif
373      value = ett
374
375      elseif(tft .eq. 7) then
376
377          err = A1_rod*d2_SP_rod +
378 1          B1_rod*((-n/r2)*VXP_rod +
379 1          (n/r)*d1_VXP_rod) +
380 1          (0.0D0,1.0D0)*k*C1_rod*d1_VRTP_rod
381
382      if (zabs(err) .lt. dabs(zero)) then
383          value = zero
384      endif
385
386      value = err
387
388      elseif(tft .eq. 8) then
389          value = 1.0D0
390      elseif(tft .eq. 9) then
391
392          exx = -k2*A1_rod*SP_rod -
393 1          (0.0D0,1.0D0)*k*C1_rod*d1_VRTP_rod -
394 1          (0.0D0,1.0D0)*k*C1_rod*VRTP_rod*(1.0D0 + n)/r
395
396      if (zabs(exx) .lt. dabs(zero)) then
397          exx = zero
398      endif
399
400      err = A1_rod*d2_SP_rod +
401 1          B1_rod*((-n/r2)*VXP_rod +

```

```

402      1      (n/r)*d1_VXP_rod) +
403      1      (0.0D0,1.0D0)*k*C1_rod*d1_VRTP_rod
404
405      if (zabs(err) .lt. dabs(zero)) then
406          err = zero
407      endif
408
409      dpp = exx - (no**2/2.0D0)*((p11+p12)*err + p12*exx)
410
411      value = dpp
412
413      elseif(tft .eq. 10) then
414
415          ett = A1_rod*(1.0D0/r*d1_SP_rod - n2/r2 * SP_rod) +
416      1          B1_rod*(n/r2*VXP_rod - n/r * d1_VXP_rod) +
417      1          (0.0D0,1.0D0)*C1_rod*k*VRTP_rod/r*(1.0D0 + n)
418
419          if (zabs(ett) .lt. dabs(zero)) then
420              value = zero
421          endif
422
423          err = A1_rod*d2_SP_rod +
424      1          B1_rod*((-n/r2)*VXP_rod +
425      1          (n/r)*d1_VXP_rod) +
426      1          (0.0D0,1.0D0)*k*C1_rod*d1_VRTP_rod
427
428          if (zabs(err) .lt. dabs(zero)) then
429              err = zero
430          endif
431
432          exx = -(lame_rod/(lame_rod + 2.0D0 * shear_rod))*(ett + err)
433          dpp = exx - (no**2/2.0D0)*((p11+p12)*err + p12*exx)
434          value = dpp
435
436      endif
437
438
439
440      return
441      end
442
443
444
445

```

```

446 C CMINV--Complex Matrix Inversion
447 C
448     SUBROUTINE MINV (C,CINV,WORK,N,IFLAG)
449 C
450 C--Notes
451 C If IFLAG = 1, the matrix is singular
452 C Working precision (artificial zero) = 1D-12
453 C
454 C--External variables
455     INTEGER N,IFLAG
456     COMPLEX*16 C(N,N),CINV(N,N),WORK(N,2*N)
457 C
458 C--Internal variables
459     INTEGER I,J,IP,IROW,JROW,JCOL,K
460     COMPLEX*16 MAXPIV,S1,C1,SWITCH
461     REAL*8 BMAG,T
462 C
463     DO 110 I = 1, N, 1
464         DO 100 J = 1, N, 1
465             WORK(I,J) = C(I,J)
466 100     CONTINUE
467 110     CONTINUE
468 C
469     DO 130 I = 1, N, 1
470         DO 120 J = 1, N, 1
471             WORK(I,J+N) = ( 0.0D0, 0.0D0 )
472             IF (I.EQ.J) WORK(I,J+N) = ( 1.0D0, 0.0D0 )
473 120     CONTINUE
474 130     CONTINUE
475 C
476     J = 1
477     I = 1
478 140     IP = I
479     MAXPIV = WORK(I,J)
480 C
481     DO 150 IROW = I+1, N, 1
482         S1 = WORK(IROW,J)
483         IF (ZABS(S1) .LT. ZABS(MAXPIV)) GOTO 150
484         IP = IROW
485         MAXPIV = WORK(IROW,J)
486 150     CONTINUE
487 C
488     IF (I.EQ.1) GOTO 170
489     DO 160 JROW = 1, 2*N, 1
490         SWITCH = WORK(IP,JROW)
491         WORK(IP,JROW) = WORK(I,JROW)

```

```

492     WORK(I,JROW) = SWITCH
493 160  CONTINUE
494 C
495 170  BMAG = ZABS(MAXPIV)
496
497 C  WRITE(*,*)BMAG
498 C
499     IF (BMAG .LT. 1.0D-12) GOTO 900
500     DO 190 IROW = I+1, N, 1
501         DO 180 JROW = J+1, 2*N, 1
502             C1 = WORK(IROW,J)*WORK(I,JROW)/WORK(I,J)
503             WORK(IROW,JROW) = WORK(IROW,JROW)-C1
504 180  CONTINUE
505     WORK(IROW,J) = ( 0., 0. )
506 190  CONTINUE
507     I = I+1
508     J = J+1
509 C
510     IF ((I.LT.N) .AND. (J.LT.N)) GOTO 140
511 C
512     DO 210 I = 1, N, 1
513         C1 = WORK(I,I)
514         DO 200 J = I, 2*N, 1
515             WORK(I,J) = WORK(I,J)/C1
516 200  CONTINUE
517 210  CONTINUE
518 C
519     DO 240 I = N, 2, -1
520         DO 230 J = I-1, 1, -1
521             C1 = WORK(J,I)
522             DO 220 JCOL = J+1, 2*N, 1
523                 WORK(J,JCOL) = WORK(J,JCOL)-C1*WORK(I,JCOL)
524 220  CONTINUE
525 230  CONTINUE
526 240  CONTINUE
527 C
528     DO 260 I = 1, N, 1
529         DO 250 J = 1, N, 1
530             CINV(I,J) = WORK(I,J+N)
531 250  CONTINUE
532 260  CONTINUE
533     IFLAG = 0
534 C
535     T = 0.0D0
536     K = 0
537 CC--The norm of the matrix inversion is T

```

```
538 CC    DO 190 I = 1, N, 1
539 CC      DO 190 J = 1, N, 1
540 CC        C1 = ( 0., 0. )
541 CC          DO 180 K = 1, N, 1
542 CC            C1 = C1 + C(I,K)*WORK(K,J)
543 CC180      CONTINUE
544 CC        S1 = ( -1., 0. ) * C1
545 CC          IF (I.EQ.J) S1 = S1 + ( 1., 0. )
546 CC            T = T + (REAL(S1)**2) + (IMAG(S1)**2)
547 CC190      CONTINUE
548 CC        T = SQRT(T)
549 CC
550 CC      RETURN
551 CC
552 900 IFLAG = 1
553
554
555      WRITE(*,*) 'BMAG < ARTIFICIAL ZERO (1.0D-12) RETURNING FROM MINV'
556      RETURN
557      END
558
559
560
```

LISTING FOR c1.f

```

1  C *****
2  C
3  C          subprogram "c1.f"
4  C
5  C This subprogram was written and developed by Mark S. Peloquin
6  C
7  C at NUWCDETNLON 6/10/95. As of 6/10/95, there are no known bugs.
8  C
9  C Please notify the author if bugs are found (203) 440-5433.
10 C
11 C *****
12
13
14      SUBROUTINE C1A_POT(r,k,Om,n)
15
16
17 C EXTERNAL VARIABLES
18
19      integer n
20      real*8 r,k,Om
21
22
23 C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
24
25      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
26      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
27      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
28      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2
29
30      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
31      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
32      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
33      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
34
35      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
36      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
37      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
38      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2
39
40      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
41
42      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
43
44      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
45 1      SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
46 1      SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,

```

```

47 1      SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
48 1      VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
49 1      VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
50 1      VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
51 1      VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
52 1      VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
53 1      VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
54 1      VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
55 1      VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
56 1      lame_c1,shear_c1,cl_c1,ct_c1,
57 1      A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
58
59 C *****
60
61      double complex cbessj,cbessy
62      double complex d1cbessj,d2cbessj
63      double complex d1cbessy,d2cbessy
64
65 C INTERNAL VARIABLES
66
67      complex*16 p,q
68
69 C DISPLACEMENT SCALAR POTENTIALS ARE EVALUATED
70
71      p = zsqrt((Om**2/cl_c1**2) - k**2)
72      d2_SP_CY1_a1 = d2cbessj(n,p,r)
73      d1_SP_CY1_a1 = d1cbessj(n,p,r)
74      SP_CY1_a1 = cbessj(n,p,r)
75      d2_SP_CY1_a2 = d2cbessy(n,p,r)
76      d1_SP_CY1_a2 = d1cbessy(n,p,r)
77      SP_CY1_a2 = cbessy(n,p,r)
78
79      q = zsqrt((Om**2/ct_c1**2) - k**2)
80      d2_VXP_CY1_a1 = d2cbessj(n,q,r)
81      d1_VXP_CY1_a1 = d1cbessj(n,q,r)
82      VXP_CY1_a1 = cbessj(n,q,r)
83      d2_VXP_CY1_a2 = d2cbessy(n,q,r)
84      d1_VXP_CY1_a2 = d1cbessy(n,q,r)
85      VXP_CY1_a2 = cbessy(n,q,r)
86
87      d2_VRTP_CY1_a1 = d2cbessj(n+1,q,r)
88      d1_VRTP_CY1_a1 = d1cbessj(n+1,q,r)
89      VRTP_CY1_a1 = cbessj(n+1,q,r)
90      d2_VRTP_CY1_a2 = d2cbessy(n+1,q,r)
91      d1_VRTP_CY1_a2 = d1cbessy(n+1,q,r)
92      VRTP_CY1_a2 = cbessy(n+1,q,r)

```

```

93
94
95     return
96     end
97
98
99
100    SUBROUTINE C1B_POT(r,k,Om,n)
101
102
103    C EXTERNAL VARIABLES
104
105        integer n
106        real*8 r,k,Om
107
108
109    C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
110
111        complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
112        complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
113        complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
114        complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2
115
116        complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
117        complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
118        complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
119        complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
120
121        complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
122        complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
123        complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
124        complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2
125
126        complex*16 lame_c1,shear_c1,cl_c1,ct_c1
127
128        complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
129
130    common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
131    1      SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
132    1      SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
133    1      SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
134    1      VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
135    1      VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
136    1      VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
137    1      VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
138    1      VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,

```

```

139 1          VRTP_CY1_a2,d1_V RTP_CY1_a2,d2_V RTP_CY1_a2,
140 1          VRTP_CY1_b1,d1_V RTP_CY1_b1,d2_V RTP_CY1_b1,
141 1          VRTP_CY1_b2,d1_V RTP_CY1_b2,d2_V RTP_CY1_b2,
142 1          lame_c1,shear_c1,cl_c1,ct_c1,
143 1          A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
144
145 C *****
146
147         double complex cbessj,cbessy
148         double complex d1cbessj,d2cbessj
149         double complex d1cbessy,d2cbessy
150
151 C INTERNAL VARIABLES
152
153         complex*16 p,q
154
155 C DISPLACEMENT SCALAR POTENTIALS ARE EVALUATED
156
157         p = zsqrt((Om**2/cl_c1**2) - k**2)
158         d2_SP_CY1_b1 = d2cbessj(n,p,r)
159         d1_SP_CY1_b1 = d1cbessj(n,p,r)
160         SP_CY1_b1 = cbessj(n,p,r)
161         d2_SP_CY1_b2 = d2cbessy(n,p,r)
162         d1_SP_CY1_b2 = d1cbessy(n,p,r)
163         SP_CY1_b2 = cbessy(n,p,r)
164
165
166         q = zsqrt((Om**2/ct_c1**2) - k**2)
167         d2_VXP_CY1_b1 = d2cbessj(n,q,r)
168         d1_VXP_CY1_b1 = d1cbessj(n,q,r)
169         VXP_CY1_b1 = cbessj(n,q,r)
170         d2_VXP_CY1_b2 = d2cbessy(n,q,r)
171         d1_VXP_CY1_b2 = d1cbessy(n,q,r)
172         VXP_CY1_b2 = cbessy(n,q,r)
173
174         d2_V RTP_CY1_b1 = d2cbessj(n+1,q,r)
175         d1_V RTP_CY1_b1 = d1cbessj(n+1,q,r)
176         VRTP_CY1_b1 = cbessj(n+1,q,r)
177         d2_V RTP_CY1_b2 = d2cbessy(n+1,q,r)
178         d1_V RTP_CY1_b2 = d1cbessy(n+1,q,r)
179         VRTP_CY1_b2 = cbessy(n+1,q,r)
180
181
182
183         return
184         end

```

```

185
186
187
188
189
190
191     SUBROUTINE SYS_MATRIX_RC1(n,k,ao_rod,ao_1cyl,bo_c1,
192 1Om,ro,co,smrc1)
193
194 C EXTERNAL VARIABLES
195
196     integer n
197     real*8 k,ao_rod,ao_1cyl,bo_c1,Om,ro,co
198     complex*16 smrc1(10,10)
199
200 C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
201
202     complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
203     complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
204     complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
205     complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2
206
207     complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
208     complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
209     complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
210     complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
211
212     complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
213     complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
214     complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
215     complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2
216
217     complex*16 lame_c1,shear_c1,cl_c1,ct_c1
218
219     complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
220
221     common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
222 1 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
223 1 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
224 1 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
225 1 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
226 1 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
227 1 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
228 1 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
229 1 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
230 1 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,

```

```

231 1      VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
232 1      VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
233 1      lame_c1,shear_c1,cl_c1,ct_c1,
234 1      A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
235
236 C *****
237
238 C DEFINITION FOR COMMON BLOCK /ROD/
239
240      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
241      1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
242      1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
243
244
245      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
246      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
247      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
248      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
249
250 C *****
251
252 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
253
254      common /OFLUID/ OFSC,d1_OFSC,M_OF
255
256      complex*16 OFSC,d1_OFSC,M_OF
257
258 C *****
259
260 C INTERNAL VARIABLES
261
262      complex*16 L2GC1,L2GR
263      real*8 b2,k2,a2
264      integer n2
265
266      a2  = ao_rod**2
267      b2  = bo_c1**2
268      n2  = n**2
269      k2  = k**2
270      L2GC1 = lame_c1 + 2.0D0 * shear_c1
271      L2GR  = lame_rod + 2.0D0 * shear_rod
272
273 C BOUNDARY CONDITION #1 (EQ 113)
274
275      smrc1(1,1) = L2GC1*d2_SP_CY1_b1 + (lame_c1/bo_c1)*
276      1d1_SP_CY1_b1 - lame_c1*SP_CY1_b1*(n2/b2 + k2)

```

```

277
278      smrc1(1,2) = L2GC1*d2_SP_CY1_b2 + (lame_c1/bo_c1)*
279      1d1_SP_CY1_b2 - lame_c1*SP_CY1_b2*(n2/b2 + k2)
280
281      smrc1(1,3) = -2.0D0*shear_c1*n/b2*(VXP_CY1_b1-bo_c1*
282      1d1_VXP_CY1_b1)
283
284      smrc1(1,4) = -2.0D0*shear_c1*n/b2*(VXP_CY1_b2-bo_c1*
285      1d1_VXP_CY1_b2)
286
287      smrc1(1,5) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_b1
288
289      smrc1(1,6) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_b2
290
291      smrc1(1,7) = (0.0D0,0.0D0)
292
293      smrc1(1,8) = (0.0D0,0.0D0)
294
295      smrc1(1,9) = (0.0D0,0.0D0)
296
297      smrc1(1,10) = +ro*Om**2*OFSC
298
299 C BOUNDARY CONDITION #2 (EQ 116)
300
301      smrc1(2,1) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_b1
302
303      smrc1(2,2) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_b2
304
305      smrc1(2,3) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_b1/bo_c1
306
307      smrc1(2,4) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_b2/bo_c1
308
309      smrc1(2,5) = shear_c1*(VRTP_CY1_b1*(n/b2 - k2 +
310      1 1.0D0/b2) - d1_VRTP_CY1_b1*(n + 1.0D0)/bo_c1 - d2_VRTP_CY1_b1)
311
312      smrc1(2,6) = shear_c1*(VRTP_CY1_b2*(n/b2 - k2 +
313      1 1.0D0/b2) - d1_VRTP_CY1_b2*(n + 1.0D0)/bo_c1 - d2_VRTP_CY1_b2)
314
315      smrc1(2,7) = (0.0D0,0.0D0)
316
317      smrc1(2,8) = (0.0D0,0.0D0)
318
319      smrc1(2,9) = (0.0D0,0.0D0)
320
321      smrc1(2,10) = (0.0D0,0.0D0)
322

```

323 C BOUNDARY CONDITION #3 (EQ 120)

324

325 $\text{smrc1}(3,1) = (\text{shear_c1} * 2.0D0 * n / \text{bo_c1}) * ((1.0D0 / \text{bo_c1}) * \\$
 326 $1\text{SP_CY1_b1} - d1_SP_CY1_b1)$

327

328 $\text{smrc1}(3,2) = (\text{shear_c1} * 2.0D0 * n / \text{bo_c1}) * ((1.0D0 / \text{bo_c1}) * \\$
 329 $1\text{SP_CY1_b2} - d1_SP_CY1_b2)$

330

331 $\text{smrc1}(3,3) = \text{shear_c1} * (-d2_VXP_CY1_b1 + 1.0D0 / \text{bo_c1} * \\$
 332 $1d1_VXP_CY1_b1 - n2 / b2 * VXP_CY1_b1)$

333

334 $\text{smrc1}(3,4) = \text{shear_c1} * (-d2_VXP_CY1_b2 + 1.0D0 / \text{bo_c1} * \\$
 335 $1d1_VXP_CY1_b2 - n2 / b2 * VXP_CY1_b2)$

336

337 $\text{smrc1}(3,5) = (0.0D0, 1.0D0) * k * \text{shear_c1} * (d1_VRTP_CY1_b1 \\$
 338 $1 - VRTP_CY1_b1 * (1.0D0 + n) / \text{bo_c1})$

339

340 $\text{smrc1}(3,6) = (0.0D0, 1.0D0) * k * \text{shear_c1} * (d1_VRTP_CY1_b2 \\$
 341 $1 - VRTP_CY1_b2 * (1.0D0 + n) / \text{bo_c1})$

342

343 $\text{smrc1}(3,7) = (0.0D0, 0.0D0)$

344

345 $\text{smrc1}(3,8) = (0.0D0, 0.0D0)$

346

347 $\text{smrc1}(3,9) = (0.0D0, 0.0D0)$

348

349 $\text{smrc1}(3,10) = (0.0D0, 0.0D0)$

350

351

352 C BOUNDARY CONDITION #4 (EQ 123)

353

354 $\text{smrc1}(4,1) = d1_SP_CY1_b1$

355

356 $\text{smrc1}(4,2) = d1_SP_CY1_b2$

357

358 $\text{smrc1}(4,3) = n / \text{bo_c1} * VXP_CY1_b1$

359

360 $\text{smrc1}(4,4) = n / \text{bo_c1} * VXP_CY1_b2$

361

362 $\text{smrc1}(4,5) = (0.0D0, 1.0D0) * k * VRTP_CY1_b1$

363

364 $\text{smrc1}(4,6) = (0.0D0, 1.0D0) * k * VRTP_CY1_b2$

365

366 $\text{smrc1}(4,7) = (0.0D0, 0.0D0)$

367

368 $\text{smrc1}(4,8) = (0.0D0, 0.0D0)$

```

369
370      smrc1(4,9) = (0.0D0,0.0D0)
371
372      smrc1(4,10)= -d1_OFSC
373
374 C BOUNDARY CONDITION #5 (EQ 125)
375
376      smrc1(5,1) = L2GC1*d2_SP_CY1_a1 + lame_c1/ao_1cyl
377      1*d1_SP_CY1_a1 - lame_c1*SP_CY1_a1*(n2/a2 + k2)
378
379      smrc1(5,2) = L2GC1*d2_SP_CY1_a2 + lame_c1/ao_1cyl
380      1*d1_SP_CY1_a2 - lame_c1*SP_CY1_a2*(n2/a2 + k2)
381
382      smrc1(5,3) = -2.0D0*shear_c1*n/a2*(VXP_CY1_a1 - ao_1cyl
383      1*d1_VXP_CY1_a1)
384
385      smrc1(5,4) = -2.0D0*shear_c1*n/a2*(VXP_CY1_a2 - ao_1cyl
386      1*d1_VXP_CY1_a2)
387
388      smrc1(5,5) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_a1
389
390      smrc1(5,6) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_a2
391
392      smrc1(5,7) = -L2GR*d2_SP_rod - lame_rod/ao_rod*
393      1d1_SP_rod + lame_rod*SP_rod*(n2/a2 + k2)
394
395      smrc1(5,8) = -2.0D0*shear_rod*n/ao_rod*(d1_VXP_rod - 1.0D0/
396      1ao_rod*VXP_rod)
397
398      smrc1(5,9) = (0.0D0,-1.0D0)*k*2.0D0*shear_rod*d1_VRTP_rod
399
400      smrc1(5,10)= (0.0D0,0.0D0)
401
402 C BOUNDARY CONDITION #6 (EQ 127)
403
404      smrc1(6,1) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_a1
405
406      smrc1(6,2) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_a2
407
408      smrc1(6,3) = (0.0D0,1.0D0)*k*n*shear_c1/ao_rod*VXP_CY1_a1
409
410      smrc1(6,4) = (0.0D0,1.0D0)*k*n*shear_c1/ao_rod*VXP_CY1_a2
411
412      smrc1(6,5) = shear_c1*(VRTP_CY1_a1*((n+1.0D0)/a2 - k2)
413      1- d1_VRTP_CY1_a1*(n+1.0D0)/ao_1cyl - d2_VRTP_CY1_a1)
414

```

```

415      smrc1(6,6) = shear_c1*(VRTP_CY1_a2*((n+1.0D0)/a2 - k2)
416 1- d1_VRTP_CY1_a2*(n+1.0D0)/ao_1cyl - d2_VRTP_CY1_a2)
417
418      smrc1(6,7) = (0.0D0,-1.0D0)*k*2.0*shear_rod*d1_SP_rod
419
420      smrc1(6,8) = (0.0D0,-1.0D0)*k*n*shear_rod/ao_rod*VXP_rod
421
422      smrc1(6,9) = -shear_rod*(VRTP_rod*((n+1.0D0)/a2 - k2)
423 1- d1_VRTP_rod*(n+1.0D0)/ao_rod - d2_VRTP_rod)
424
425      smrc1(6,10) = (0.0D0,0.0D0)
426
427 C BOUNDARY CONDITION #7 (EQ 129)
428
429      smrc1(7,1) = shear_c1*2.0D0*n/ao_1cyl*(SP_CY1_a1/ao_1cyl
430 1- d1_SP_CY1_a1)
431
432      smrc1(7,2) = shear_c1*2.0D0*n/ao_1cyl*(SP_CY1_a2/ao_1cyl
433 1- d1_SP_CY1_a2)
434
435      smrc1(7,3) = shear_c1*(-d2_VXP_CY1_a1 + d1_VXP_CY1_a1/
436 1 ao_1cyl - n2/a2*VXP_CY1_a1)
437
438      smrc1(7,4) = shear_c1*(-d2_VXP_CY1_a2 + d1_VXP_CY1_a2/
439 1 ao_1cyl - n2/a2*VXP_CY1_a2)
440
441      smrc1(7,5) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_a1 -
442 1 VRTP_CY1_a1*(1.0D0+n)/ao_1cyl)
443
444      smrc1(7,6) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_a2 -
445 1 VRTP_CY1_a2*(1.0D0+n)/ao_1cyl)
446
447      smrc1(7,7) = -2.0D0*n*shear_rod/a2*(SP_rod - ao_rod*d1_SP_rod)
448
449      smrc1(7,8) = -shear_rod*(-d2_VXP_rod + d1_VXP_rod/ao_rod
450 1 - n2/a2*VXP_rod)
451
452      smrc1(7,9) = (0.0D0,-1.0D0)*k*shear_rod*(d1_VRTP_rod
453 1 - VRTP_rod*(1.0D0 + n)/ao_rod)
454
455      smrc1(7,10) = (0.0D0,0.0D0)
456
457 C BOUNDARY CONDITION #8 (EQ 132)
458
459      smrc1(8,1) = d1_SP_cyl_a1
460

```

```

461      smrc1(8,2) = d1_SP_cyl_a2
462
463      smrc1(8,3) = n*VXP_CY1_a1/ao_1cyl
464
465      smrc1(8,4) = n*VXP_CY1_a2/ao_1cyl
466
467      smrc1(8,5) = (0.0D0,1.0D0)*k*VRTP_CY1_a1
468
469      smrc1(8,6) = (0.0D0,1.0D0)*k*VRTP_CY1_a2
470
471      smrc1(8,7) = -d1_SP_rod
472
473      smrc1(8,8) = -n*VXP_rod/ao_rod
474
475      smrc1(8,9) = (0.0D0,-1.0D0)*k*VRTP_rod
476
477      smrc1(8,10) = (0.0D0,0.0D0)
478
479 C BOUNDARY CONDITION #9 (EQ 135)
480
481      smrc1(9,1) = -n*SP_CY1_a1/ao_1cyl
482
483      smrc1(9,2) = -n*SP_CY1_a2/ao_1cyl
484
485      smrc1(9,3) = -d1_VXP_CY1_a1
486
487      smrc1(9,4) = -d1_VXP_CY1_a2
488
489      smrc1(9,5) = (0.0D0,1.0D0)*k*VRTP_CY1_a1
490
491      smrc1(9,6) = (0.0D0,1.0D0)*k*VRTP_CY1_a2
492
493      smrc1(9,7) = n*SP_rod/ao_rod
494
495      smrc1(9,8) = d1_VXP_rod
496
497      smrc1(9,9) = (0.0D0,-1.0D0)*k*VRTP_rod
498
499      smrc1(9,10) = (0.0D0,0.0D0)
500
501 C BOUNDARY CONDITION #10 (EQ 138)
502
503      smrc1(10,1) = (0.0D0,1.0D0)*k*SP_CY1_a1
504
505      smrc1(10,2) = (0.0D0,1.0D0)*k*SP_CY1_a2
506

```

```

507      smrc1(10,3) = (0.0D0,0.0D0)
508
509      smrc1(10,4) = (0.0D0,0.0D0)
510
511      smrc1(10,5) = -d1_VRTP_CY1_a1 - (n + 1.0D0)*VRTP_CY1_a1
512 1/ao_1cyl
513
514      smrc1(10,6) = -d1_VRTP_CY1_a2 - (n + 1.0D0)*VRTP_CY1_a2
515 1/ao_1cyl
516
517      smrc1(10,7) = (0.0D0,-1.0D0)*k*SP_rod
518
519      smrc1(10,8) = (0.0D0,0.0D0)
520
521      smrc1(10,9) = d1_VRTP_rod + VRTP_rod*(n + 1.0D0)
522 1/ao_rod
523
524      smrc1(10,10) = (0.0D0,0.0D0)
525
526
527      return
528      end
529
530
531
532
533      SUBROUTINE ABC_RC1_INVERT(n,exctype,smrc1,b)
534
535
536 C EXTERNAL VARIABLES
537
538      integer n,exctype
539      real*8 b
540      complex*16 smrc1(10,10)
541
542 C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
543
544      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
545      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
546      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
547      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2
548
549      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
550      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
551      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
552      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2

```

```

553
554      complex*16 VRTP_CY1_a1,d1_V RTP_CY1_a1,d2_V RTP_CY1_a1
555      complex*16 VRTP_CY1_a2,d1_V RTP_CY1_a2,d2_V RTP_CY1_a2
556      complex*16 VRTP_CY1_b1,d1_V RTP_CY1_b1,d2_V RTP_CY1_b1
557      complex*16 VRTP_CY1_b2,d1_V RTP_CY1_b2,d2_V RTP_CY1_b2
558
559      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
560      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
561
562      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
563      1      SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
564      1      SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
565      1      SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
566      1      VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
567      1      VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
568      1      VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
569      1      VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
570      1      VRTP_CY1_a1,d1_V RTP_CY1_a1,d2_V RTP_CY1_a1,
571      1      VRTP_CY1_a2,d1_V RTP_CY1_a2,d2_V RTP_CY1_a2,
572      1      VRTP_CY1_b1,d1_V RTP_CY1_b1,d2_V RTP_CY1_b1,
573      1      VRTP_CY1_b2,d1_V RTP_CY1_b2,d2_V RTP_CY1_b2,
574      1      lame_c1,shear_c1,cl_c1,ct_c1,
575      1      A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
576
577 C *****
578
579 C DEFINITIONS FOR COMMON BLOCK /ROD/
580
581      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
582      1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_V RTP_rod,d2_V RTP_rod
583      1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
584
585
586      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
587      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
588      complex*16 VRTP_rod,d1_V RTP_rod,d2_V RTP_rod
589      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
590
591 C *****
592
593 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
594
595      complex*16 OFSC,d1_OFSC,M_OF
596
597      common /OFLUID/ OFSC,d1_OFSC,M_OF
598

```

```

599 C *****
600
601 C INTERNAL VARIABLES
602
603     integer size,iflag
604     complex*16 smrc1inv(10,10),workrc1(10,20)
605
606     size = 10
607     iflag = 0
608
609
610     CALL MINV(smrc1,smrc1inv,workrc1,size,iflag)
611
612
613
614 C RADIAL EXCITATION exctype = 1
615 C AXIAL EXCITATION exctype = 0
616
617     if (exctype .eq. 1) then
618         A1_C1 = -smrc1inv(1,1)
619         A2_C1 = -smrc1inv(2,1)
620         B1_C1 = -smrc1inv(3,1)
621         B2_C1 = -smrc1inv(4,1)
622         C1_C1 = -smrc1inv(5,1)
623         C2_C1 = -smrc1inv(6,1)
624         A1_rod = -smrc1inv(7,1)
625         B1_rod = -smrc1inv(8,1)
626         C1_rod = -smrc1inv(9,1)
627         M_OF = -smrc1inv(10,1)
628     elseif (exctype .eq. 0) then
629         A1_C1 = -smrc1inv(1,2)
630         A2_C1 = -smrc1inv(2,2)
631         B1_C1 = -smrc1inv(3,2)
632         B2_C1 = -smrc1inv(4,2)
633         C1_C1 = -smrc1inv(5,2)
634         C2_C1 = -smrc1inv(6,2)
635         A1_rod = -smrc1inv(7,2)
636         B1_rod = -smrc1inv(8,2)
637         C1_rod = -smrc1inv(9,2)
638         M_OF = -smrc1inv(10,2)
639     endif
640
641
642     return
643     end
644

```

```

645
646
647
648
649      SUBROUTINE OUTPUT_RC1(tft,n,k,r,value)
650
651 C EXTERNAL VARIABLES
652
653      integer n,tft
654      real*8 k,r,zero
655      complex*16 value
656
657 C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
658
659      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
660      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
661      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
662      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2
663
664      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
665      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
666      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
667      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
668
669      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
670      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
671      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
672      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2
673
674      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
675      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
676
677      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
678 1      SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
679 1      SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
680 1      SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
681 1      VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
682 1      VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
683 1      VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
684 1      VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
685 1      VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
686 1      VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
687 1      VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
688 1      VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
689 1      lame_c1,shear_c1,cl_c1,ct_c1,
690 1      A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

```

```

691
692 C *****
693
694 C DEFINITIONS FOR COMMON BLOCK /ROD/
695
696     common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
697     1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
698     1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
699
700
701     complex*16 SP_rod,d1_SP_rod,d2_SP_rod
702     complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
703     complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
704     complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
705
706 C *****
707
708 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
709
710     common /OFLUID/ OFSC,d1_OFSC,M_OF
711
712     complex*16 OFSC,d1_OFSC,M_OF
713
714 C *****
715
716 C INTERNAL VARIABLES
717
718     integer n2
719     real*8 k2,r2,no,p11,p12
720     complex*16 dpp,exx,err,Srr,Sxx,ett,uc,vc,wc
721     complex*16 A1C1,A2C1,B1C1,B2C1,C1C1,C2C1,L2GC1
722
723     n2 = n**2
724     r2 = r**2
725     k2 = k**2
726
727     zero = 1.000D-30
728     no = 1.460D0
729     p11 = 0.126D0
730     p12 = 0.270D0
731     L2GC1 = lame_c1 + 2.0D0*shear_c1
732
733

```

```

734 C RADIAL STRESS/(Pr or Px)          tft = 0
735 C                                  tft = 1
736 C AXIAL DISPLACEMENT              tft = 2
737 C THETA DISPLACEMENT              tft = 3
738 C RADIAL DISPLACEMENT             tft = 4
739 C LONGITUDINAL STRAIN      e11/(Pr or Px)    tft = 5
740 C THETA STRAIN              ett/(Pr or Px)    tft = 6
741 C RADIAL STRAIN             err/(Pr or Px)    tft = 7
742 C                                  tft = 8
743 C OPTIC exx=0@k=0            ((dp/p)(r))/(Pr or Px)  tft = 9
744 C OPTIC exx=const@k=0       ((dp/p)(r))/(Pr or Px)  tft = 10
745
746
747
748
749     if (tft .eq. 0) then
750
751         A1C1 = A1_C1*(L2GC1*d2_SP_CY1_a1 + lame_c1*
752     1         d1_SP_CY1_a1/r - lame_c1*SP_CY1_a1*(n2/
753     1         r2 + k2))
754
755         A2C1 = A2_C1*(L2GC1*d2_SP_CY1_a2 + lame_c1*
756     1         d1_SP_CY1_a2/r - lame_c1* SP_CY1_a2*(n2/
757     1         r2 + k2))
758
759         B1C1 = B1_C1*-2.0*shear_c1*n/r**2*(VXP_CY1_a1 -
760     1         r*d1_VXP_CY1_a1)
761
762         B2C1 = B2_C1*-2.0*shear_c1*n/r**2*(VXP_CY1_a2 -
763     1         r*d1_VXP_CY1_a2)
764
765         C1C1 = (0.0D0,1.0D0)*C1_C1*2.0D0*shear_c1*k*d1_VRTP_CY1_a1
766
767         C2C1 = (0.0D0,1.0D0)*C2_C1*2.0D0*shear_c1*k*d1_VRTP_CY1_a2
768
769         Srr = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1
770
771         value = Srr
772
773
774 C LONGITUDINAL STRESS USING (EQ 153)
775     elseif(tft .eq. 1) then
776
777         A1C1 = A1_C1*(lame_c1*(d2_SP_CY1_a1 + d1_SP_CY1_a1/r)
778     1         - SP_CY1_a1*(lame_c1*n2/r2 + k2*(lame_c1
779     1         + 2.0D0*shear_c1)))

```

```

780
781      A2C1 = A2_C1*(lame_c1*(d2_SP_CY1_a2 + d1_SP_CY1_a2/r)
782 1      - SP_CY1_a2*(lame_c1*n2/r2 + k2*(lame_c1
783 1      + 2.0D0*shear_c1)))
784
785      C1C1 = C1_C1*((0.0D0,-1.0D0)*k*(d1_VRTP_CY1_a1 +
786 1      VRTP_CY1_a1*(n + 1.0D0)/r))*2.0D0*shear_c1
787
788      C2C1 = C2_C1*((0.0D0,-1.0D0)*k*(d1_VRTP_CY1_a2 +
789 1      VRTP_CY1_a2*(n + 1.0D0)/r))*2.0D0*shear_c1
790
791      Sxx = A1C1 + A2C1 + C1C1 + C2C1
792
793      value = Sxx
794
795
796 C AXIAL DISPLACEMENT USING (EQ 149)
797     elseif(tft .eq. 2) then
798
799         A1C1 = A1_C1*(0.0D0,1.0D0)*k*SP_CY1_a1
800
801         A2C1 = A2_C1*(0.0D0,1.0D0)*k*SP_CY1_a2
802
803         C1C1 = C1_C1*(-d1_VRTP_CY1_a1 - VRTP_CY1_a1*
804 1         (n + 1.0D0)/r)
805
806         C2C1 = C2_C1*(-d1_VRTP_CY1_a2 - VRTP_CY1_a2*
807 1         (n + 1.0D0)/r)
808
809         wc = A1C1 + A2C1 + C1C1 + C2C1
810
811         value = wc
812
813 C THETA DISPLACEMENT USING (EQ 150)
814     elseif(tft .eq. 3) then
815
816
817         A1C1 = A1_C1*-SP_CY1_a1*n/r
818
819         A2C1 = A2_C1*-SP_CY1_a2*n/r
820
821         B1C1 = -B1_C1*d1_VXP_CY1_a1
822
823         B2C1 = -B2_C1*d1_VXP_CY1_a2
824
825         C1C1 = C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a1

```

```

826
827      C2C1 = C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a2
828
829      vc = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1
830
831      value = vc
832
833 C RADIAL DISPLACEMENT USING (EQ 148)
834      elseif(tft .eq. 4) then
835
836          A1C1 = A1_C1*d1_SP_CY1_a1
837
838          A2C1 = A2_C1*d1_SP_CY1_a2
839
840          B1C1 = B1_C1*VXP_CY1_a1*n/r
841
842          B2C1 = B2_C1*VXP_CY1_a2*n/r
843
844          C1C1 = C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a1
845
846          C2C1 = C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a2
847
848          uc = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1
849
850          value = uc
851
852 C LONGITUDINAL STRAIN USING (EQ 147)
853      elseif(tft .eq. 5) then
854
855          A1C1 = -A1_C1*k2*SP_CY1_a1
856
857          A2C1 = -A2_C1*k2_SP_CY1_a2
858
859          C1C1 = C1_C1*(0.0D0,-1.0D0)*k*(d1_VRTP_CY1_a1 +
860 1          VRTP_CY1_a1*(n + 1.0D0)/r)
861
862          C2C1 = C2_C1*(0.0D0,-1.0D0)*k*(d1_VRTP_CY1_a2 +
863 1          VRTP_CY1_a2*(n + 1.0D0)/r)
864
865          exx = A1C1 + A2C1 + C1C1 + C2C1
866
867
868          if (zabs(exx) .lt. dabs(zero)) then
869              value = zero
870          endif
871

```

```

872         value = exx
873
874 C THETA STRAIN USING (EQ 145)
875     elseif(tft .eq. 6) then
876
877         A1C1 = A1_C1*(d1_SP_CY1_a1/r - SP_CY1_a1*n2/r2)
878
879         A2C1 = A2_C1*(d1_SP_CY1_a2/r - SP_CY1_a2*n2/r2)
880
881         B1C1 = B1_C1*(-d1_VXP_CY1_a1*n/r + VXP_CY1_a1*n/r2)
882
883         B2C1 = B2_C1*(-d1_VXP_CY1_a2*n/r + VXP_CY1_a2*n/r2)
884
885         C1C1 = C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a1*(n + 1.0D0)/r
886
887         C2C1 = C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a2*(n + 1.0D0)/r
888
889         ett = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1
890
891         if (zabs(ett) .lt. dabs(zero)) then
892             value = zero
893         endif
894
895         value = ett
896
897 C RADIAL STRAIN USING (eq 143)
898     elseif(tft .eq. 7) then
899
900         A1C1 = A1_C1*d2_SP_CY1_a1
901
902         A2C1 = A2_C1*d2_SP_CY1_a2
903
904         B1C1 = B1_C1*(d1_VXP_CY1_a1 - VXP_CY1_a1/r)*n/r
905
906         B2C1 = B2_C1*(d1_VXP_CY1_a2 - VXP_CY1_a2/r)*n/r
907
908         C1C1 = C1_C1*(0.0D0,1.0D0)*k*d1_VRTP_CY1_a1
909
910         C2C1 = C2_C1*(0.0D0,1.0D0)*k*d1_VRTP_CY1_a2
911
912         err = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1
913
914         if (zabs(err) .lt. dabs(zero)) then
915             value = zero
916         endif
917

```

```

918         value = err
919
920 C UNUSED AT PRESENT
921         elseif(tft .eq. 8) then
922             value = 1.0
923
924 C OPTIC dp/p/uPa USING (EQ 147 & 143)
925 C exx = 0 @ k = 0
926         elseif(tft .eq. 9) then
927
928             A1C1 = -A1_C1*k2*SP_CY1_a1
929
930             A2C1 = -A2_C1*k2_SP_CY1_a2
931
932             C1C1 = C1_C1*(0.0D0,-1.0D0)*k*(d1_VRTP_CY1_a1 +
933 1             VRTP_CY1_a1*(n + 1.0D0)/r)
934
935             C2C1 = C2_C1*(0.0D0,-1.0D0)*k*(d1_VRTP_CY1_a2 +
936 1             VRTP_CY1_a2*(n + 1.0D0)/r)
937
938             exx = A1C1 + A2C1 + C1C1 + C2C1
939
940             if (zabs(exx) .lt. dabs(zero)) then
941                 exx = zero
942             endif
943
944             A1C1 = A1_C1*d2_SP_CY1_a1
945
946             A2C1 = A2_C1*d2_SP_CY1_a2
947
948             B1C1 = B1_C1*(d1_VXP_CY1_a1 - VXP_CY1_a1/r)*n/r
949
950             B2C1 = B2_C1*(d1_VXP_CY1_a2 - VXP_CY1_a2/r)*n/r
951
952             C1C1 = C1_C1*(0.0D0,1.0D0)*k*d1_VRTP_CY1_a1
953
954             C2C1 = C2_C1*(0.0D0,1.0D0)*k*d1_VRTP_CY1_a2
955
956             err = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1
957
958             if (zabs(err) .lt. dabs(zero)) then
959                 err = zero
960             endif
961
962             dpp = exx - (no**2/2.0D0)*((p11+p12)*err + p12*exx)
963

```

```

964         value = dpp
965
966 C OPTIC dp/p/uPa USING (EQ 145 & 143)
967 C exx = constant @ k = 0
968     elseif(tft.eq. 10) then
969
970         A1C1 = A1_C1*(d1_SP_CY1_a1/r - SP_CY1_a1*n2/r2)
971
972         A2C1 = A2_C1*(d1_SP_CY1_a2/r - SP_CY1_a2*n2/r2)
973
974         B1C1 = B1_C1*(-d1_VXP_CY1_a1*n/r + VXP_CY1_a1*n/r2)
975
976         B2C1 = B2_C1*(-d1_VXP_CY1_a2*n/r + VXP_CY1_a2*n/r2)
977
978         C1C1 = C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a1*(n + 1.0D0)/r
979
980         C2C1 = C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_a2*(n + 1.0D0)/r
981
982         ett = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1
983
984         if (zabs(ett) .lt. dabs(zero)) then
985             value = zero
986         endif
987
988         A1C1 = A1_C1*d2_SP_CY1_a1
989
990         A2C1 = A2_C1*d2_SP_CY1_a2
991
992         B1C1 = B1_C1*(d1_VXP_CY1_a1 - VXP_CY1_a1/r)*n/r
993
994         B2C1 = B2_C1*(d1_VXP_CY1_a2 - VXP_CY1_a2/r)*n/r
995
996         C1C1 = C1_C1*(0.0D0,1.0D0)*k*d1_VRTP_CY1_a1
997
998         C2C1 = C2_C1*(0.0D0,1.0D0)*k*d1_VRTP_CY1_a2
999
1000        err = A1C1 + A2C1 + B1C1 + B2C1 + C1C1 + C2C1
1001
1002        if (zabs(err) .lt. dabs(zero)) then
1003            err = zero
1004        endif
1005
1006        exx = -(lame_rod/(lame_rod + 2.0D0 * shear_rod))*(ett + err)
1007        dpp = exx - (no**2/2.0D0)*((p11+p12)*err + p12*exx)
1008        value = dpp
1009

```

```
1010      endif
1011
1012
1013      return
1014      end
1015
1016
```

LISTING FOR c2.f

c2.f

Sat Jun 10 14:39:41 1995

```

1  C *****
2  C
3  C          subprogram "c2.f"
4  C
5  C This subprogram was written and developed by Mark S. Peloquin
6  C
7  C at NUWCDETNLON 6/10/95. As of 6/10/95, there are no known bugs.
8  C
9  C Please notify the author if bugs are found (203) 440-5433.
10 C
11 C *****
12 C Modifications on 5-14-95; Removed K & I Bessel functions.
13 C The routine is now based on J & Y only, using series to handle
14 C complex arguments. C2B_POT, C2C_POT
15
16
17          SUBROUTINE C2B_POT(r,k,Om,n)
18
19
20 C EXTERNAL VARIABLES
21
22          integer n
23          real*8 r,k,Om
24
25
26 C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/
27
28          complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
29          complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
30          complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
31          complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2
32
33          complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
34          complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
35          complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
36          complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2
37
38          complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1
39          complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2
40          complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1
41          complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2
42
43          complex*16 lame_c2,shear_c2,cl_c2,ct_c2
44

```

```

45      complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2
46
47      common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1,
48 1      SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,
49 1      SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,
50 1      SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,
51 1      VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,
52 1      VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,
53 1      VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,
54 1      VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,
55 1      VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,
56 1      VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,
57 1      VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,
58 1      VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,
59 1      lame_c2,shear_c2,cl_c2,ct_c2,
60 1      A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2
61
62 C *****
63
64      double complex cbessj,cbessy
65      double complex d1cbessj,d2cbessj
66      double complex d1cbessy,d2cbessy
67
68
69 C INTERNAL VARIABLES
70
71      complex*16 p,q
72
73 C DISPLACEMENT SCALAR POTENTIALS ARE EVALUATED
74
75      p = zsqrt((Om**2/cl_c2**2) - k**2)
76      d2_SP_CY2_b1 = d2cbessj(n,p,r)
77      d1_SP_CY2_b1 = d1cbessj(n,p,r)
78      SP_CY2_b1 = cbessj(n,p,r)
79      d2_SP_CY2_b2 = d2cbessy(n,p,r)
80      d1_SP_CY2_b2 = d1cbessy(n,p,r)
81      SP_CY2_b2 = cbessy(n,p,r)
82
83      q = zsqrt((Om**2/ct_c2**2) - k**2)
84      d2_VXP_CY2_b1 = d2cbessj(n,q,r)
85      d1_VXP_CY2_b1 = d1cbessj(n,q,r)
86      VXP_CY2_b1 = cbessj(n,q,r)
87      d2_VXP_CY2_b2 = d2cbessy(n,q,r)
88      d1_VXP_CY2_b2 = d1cbessy(n,q,r)
89      VXP_CY2_b2 = cbessy(n,q,r)
90

```

```

91      d2_V RTP_CY2_b1 = d2cbessj(n+1,q,r)
92      d1_V RTP_CY2_b1 = d1cbessj(n+1,q,r)
93      VRTP_CY2_b1 = cbessj(n+1,q,r)
94      d2_V RTP_CY2_b2 = d2cbessy(n+1,q,r)
95      d1_V RTP_CY2_b2 = d1cbessy(n+1,q,r)
96      VRTP_CY2_b2 = cbessy(n+1,q,r)
97
98
99
100
101
102      return
103      end
104
105
106
107      SUBROUTINE C2C_POT(r,k,Om,n)
108
109
110      C EXTERNAL VARIABLES
111
112      integer n
113      real*8 r,k,Om
114
115
116      C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/
117
118
119      complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
120      complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
121      complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
122      complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2
123
124      complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
125      complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
126      complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
127      complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2
128
129      complex*16 VRTP_CY2_b1,d1_V RTP_CY2_b1,d2_V RTP_CY2_b1
130      complex*16 VRTP_CY2_b2,d1_V RTP_CY2_b2,d2_V RTP_CY2_b2
131      complex*16 VRTP_CY2_c1,d1_V RTP_CY2_c1,d2_V RTP_CY2_c1
132      complex*16 VRTP_CY2_c2,d1_V RTP_CY2_c2,d2_V RTP_CY2_c2
133
134      complex*16 lame_c2,shear_c2,cl_c2,ct_c2
135
136      complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

```

```

137
138
139      common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1,
140 1      SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,
141 1      SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,
142 1      SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,
143 1      VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,
144 1      VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,
145 1      VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,
146 1      VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,
147 1      VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,
148 1      VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,
149 1      VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,
150 1      VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,
151 1      lame_c2,shear_c2,cl_c2,ct_c2,
152 1      A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2
153
154 C *****
155
156
157      double complex cbessj,cbessy
158      double complex d1cbessj,d2cbessj
159      double complex d1cbessy,d2cbessy
160
161 C INTERNAL VARIABLES
162
163      complex*16 p,q
164
165 C DISPLACEMENT SCALAR POTENTIALS ARE EVALUATED
166
167      p = zsqrt((Om**2/cl_c2**2) - k**2)
168      d2_SP_CY2_c1 = d2cbessj(n,p,r)
169      d1_SP_CY2_c1 = d1cbessj(n,p,r)
170      SP_CY2_c1 = cbessj(n,p,r)
171      d2_SP_CY2_c2 = d2cbessy(n,p,r)
172      d1_SP_CY2_c2 = d1cbessy(n,p,r)
173      SP_CY2_c2 = cbessy(n,p,r)
174
175
176      q = zsqrt((Om**2/ct_c2**2) - k**2)
177      d2_VXP_CY2_c1 = d2cbessj(n,q,r)
178      d1_VXP_CY2_c1 = d1cbessj(n,q,r)
179      VXP_CY2_c1 = cbessj(n,q,r)
180      d2_VXP_CY2_c2 = d2cbessy(n,q,r)
181      d1_VXP_CY2_c2 = d1cbessy(n,q,r)
182      VXP_CY2_c2 = cbessy(n,q,r)

```

```

183
184      d2_V RTP_CY2_c1 = d2cbessj(n+1,q,r)
185      d1_V RTP_CY2_c1 = d1cbessj(n+1,q,r)
186      VRTP_CY2_c1 = cbessj(n+1,q,r)
187      d2_V RTP_CY2_c2 = d2cbessy(n+1,q,r)
188      d1_V RTP_CY2_c2 = d1cbessy(n+1,q,r)
189      VRTP_CY2_c2 = cbessy(n+1,q,r)
190
191
192
193      return
194      end
195
196
197
198      SUBROUTINE SYS_MATRIX_RC2(n,k,ao_1cyl,bo_c1,co_c2,
199      1Om,ro,co,ri,ci,smrc2)
200
201  C EXTERNAL VARIABLES
202
203      integer n
204      real*8 k,ao_1cyl,bo_c1,co_c2,Om,ro,co,ri,ci
205      complex*16 smrc2(13,13)
206
207  C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
208
209      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
210      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
211      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
212      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2
213
214      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
215      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
216      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
217      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
218
219      complex*16 VRTP_CY1_a1,d1_V RTP_CY1_a1,d2_V RTP_CY1_a1
220      complex*16 VRTP_CY1_a2,d1_V RTP_CY1_a2,d2_V RTP_CY1_a2
221      complex*16 VRTP_CY1_b1,d1_V RTP_CY1_b1,d2_V RTP_CY1_b1
222      complex*16 VRTP_CY1_b2,d1_V RTP_CY1_b2,d2_V RTP_CY1_b2
223
224      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
225      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
226
227
228      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,

```

```

229 1      SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
230 1      SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
231 1      SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
232 1      VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
233 1      VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
234 1      VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
235 1      VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
236 1      VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
237 1      VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
238 1      VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
239 1      VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
240 1      lame_c1,shear_c1,cl_c1,ct_c1,
241 1      A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
242
243 C *****
244
245
246
247 C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/
248
249
250      complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
251      complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
252      complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
253      complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2
254
255      complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
256      complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
257      complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
258      complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2
259
260      complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1
261      complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2
262      complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1
263      complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2
264
265      complex*16 lame_c2,shear_c2,cl_c2,ct_c2
266
267      complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2
268
269
270      common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1,
271 1      SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,
272 1      SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,
273 1      SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,
274 1      VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,

```

```

275 1      VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,
276 1      VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,
277 1      VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,
278 1      VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,
279 1      VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,
280 1      VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,
281 1      VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,
282 1      lame_c2,shear_c2,cl_c2,ct_c2,
283 1      A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2
284
285 C *****
286
287
288
289 C DEFINITION FOR COMMON BLOCK /ROD/
290
291      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
292      1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
293      1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
294
295
296      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
297      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
298      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
299      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
300
301 C *****
302
303 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
304
305      common /OFLUID/ OFSC,d1_OFSC,M_OF
306
307      complex*16 OFSC,d1_OFSC,M_OF
308
309 C *****
310
311 C *****
312
313 C DEFINITIONS FOR COMMON BLOCK /IFLUID/
314
315      common /IFLUID/ IFSC,d1_IFSC,D_IF
316
317      complex*16 IFSC,d1_IFSC,D_IF
318
319 C *****
320

```

```

321
322 C INTERNAL VARIABLES
323
324     complex*16 L2GC1,L2GC2
325     real*8 a,b,c,a2,b2,c2,k2
326     integer n2
327
328     a      = ao_1cyl
329     b      = bo_c1
330     c      = co_c2
331     a2     = ao_1cyl**2
332     b2     = bo_c1**2
333     c2     = co_c2**2
334     n2     = n**2
335     k2     = k**2
336     L2GC1  = lame_c1 + 2.0D0 * shear_c1
337     L2GC2  = lame_c2 + 2.0D0 * shear_c2
338
339
340
341 C BOUNDARY CONDITION #1 (EQ 229)
342
343     smrc2(1,1) = L2GC2*d2_SP_CY2_c1 + (lame_c2/c)*
344     1d1_SP_CY2_c1 - lame_c2*SP_CY2_c1*(n2/c2 + k2) +
345     1ro*Om**2*OFSC/d1_OFSC*d1_SP_CY2_c1
346
347     smrc2(1,2) = L2GC2*d2_SP_CY2_c2 + (lame_c2/c)*
348     1d1_SP_CY2_c2 - lame_c2*SP_CY2_c2*(n2/c2 + k2) +
349     1ro*Om**2*OFSC/d1_OFSC*d1_SP_CY2_c2
350
351     smrc2(1,3) = -2.0D0*shear_c2*n/c2*(VXP_CY2_c1-c*
352     1d1_VXP_CY2_c1) +
353     1n*ro*Om**2*OFSC*VXP_CY2_c1/(d1_OFSC*c)
354
355     smrc2(1,4) = -2.0D0*shear_c2*n/c2*(VXP_CY2_c2-c*
356     1d1_VXP_CY2_c2) +
357     1n*ro*Om**2*OFSC*VXP_CY2_c2/(d1_OFSC*c)
358
359     smrc2(1,5) = (0.0D0,1.0D0)*2.0D0*shear_c2*k*d1_VRTP_CY2_c1
360     1 + (0.0D0,1.0D0)*k*ro*Om**2*OFSC*VRTP_CY2_c1/d1_OFSC
361
362     smrc2(1,6) = (0.0D0,1.0D0)*2.0D0*shear_c2*k*d1_VRTP_CY2_c2
363     1 + (0.0D0,1.0D0)*k*ro*Om**2*OFSC*VRTP_CY2_c2/d1_OFSC
364
365     smrc2(1,7) = (0.0D0,0.0D0)
366

```

```

367      smrc2(1,8) = (0.0D0,0.0D0)
368
369      smrc2(1,9) = (0.0D0,0.0D0)
370
371      smrc2(1,10)= (0.0D0,0.0D0)
372
373      smrc2(1,11)= (0.0D0,0.0D0)
374
375      smrc2(1,12)= (0.0D0,0.0D0)
376
377      smrc2(1,13)= (0.0D0,0.0D0)
378
379
380
381 C BOUNDARY CONDITION #2 (EQ 215)
382
383      smrc2(2,1) = (0.0D0,1.0D0)*k*2.0D0*shear_c2*d1_SP_CY2_c1
384
385      smrc2(2,2) = (0.0D0,1.0D0)*k*2.0D0*shear_c2*d1_SP_CY2_c2
386
387      smrc2(2,3) = (0.0D0,1.0D0)*k*n*shear_c2*VXP_CY2_c1/c
388
389      smrc2(2,4) = (0.0D0,1.0D0)*k*n*shear_c2*VXP_CY2_c2/c
390
391      smrc2(2,5) = shear_c2*(VRTP_CY2_c1*(n/c2 - k2 +
392 1 1.0D0/c2) - d1_VRTP_CY2_c1*(n + 1.0D0)/c - d2_VRTP_CY2_c1)
393
394      smrc2(2,6) = shear_c2*(VRTP_CY2_c2*(n/c2 - k2 +
395 1 1.0D0/c2) - d1_VRTP_CY2_c2*(n + 1.0D0)/c - d2_VRTP_CY2_c2)
396
397      smrc2(2,7) = (0.0D0,0.0D0)
398
399      smrc2(2,8) = (0.0D0,0.0D0)
400
401      smrc2(2,9) = (0.0D0,0.0D0)
402
403      smrc2(2,10) = (0.0D0,0.0D0)
404
405      smrc2(2,11) = (0.0D0,0.0D0)
406
407      smrc2(2,12) = (0.0D0,0.0D0)
408
409      smrc2(2,13) = (0.0D0,0.0D0)
410
411
412

```

413 C BOUNDARY CONDITION #3 (EQ 216)

414

415 $\text{smrc2}(3,1) = (\text{shear_c2} * 2.0\text{D0} * \text{n}/\text{c}) * ((1.0\text{D0}/\text{c}) *$

416 $1\text{SP_CY2_c1} - \text{d1_SP_CY2_c1})$

417

418 $\text{smrc2}(3,2) = (\text{shear_c2} * 2.0\text{D0} * \text{n}/\text{c}) * ((1.0\text{D0}/\text{c}) *$

419 $1\text{SP_CY2_c2} - \text{d1_SP_CY2_c2})$

420

421 $\text{smrc2}(3,3) = \text{shear_c2} * (-\text{d2_VXP_CY2_c1} + 1.0\text{D0}/\text{c} *$

422 $1\text{d1_VXP_CY2_c1} - \text{n2}/\text{c2} * \text{VXP_CY2_c1})$

423

424 $\text{smrc2}(3,4) = \text{shear_c2} * (-\text{d2_VXP_CY2_c2} + 1.0\text{D0}/\text{c} *$

425 $1\text{d1_VXP_CY2_c2} - \text{n2}/\text{c2} * \text{VXP_CY2_c2})$

426

427 $\text{smrc2}(3,5) = (0.0\text{D0}, 1.0\text{D0}) * \text{k} * \text{shear_c2} * (\text{d1_VRTP_CY2_c1}$

428 $1 - \text{VRTP_CY2_c1} * (1.0\text{D0} + \text{n})/\text{c})$

429

430 $\text{smrc2}(3,6) = (0.0\text{D0}, 1.0\text{D0}) * \text{k} * \text{shear_c2} * (\text{d1_VRTP_CY2_c2}$

431 $1 - \text{VRTP_CY2_c2} * (1.0\text{D0} + \text{n})/\text{c})$

432

433 $\text{smrc2}(3,7) = (0.0\text{D0}, 0.0\text{D0})$

434

435 $\text{smrc2}(3,8) = (0.0\text{D0}, 0.0\text{D0})$

436

437 $\text{smrc2}(3,9) = (0.0\text{D0}, 0.0\text{D0})$

438

439 $\text{smrc2}(3,10) = (0.0\text{D0}, 0.0\text{D0})$

440

441 $\text{smrc2}(3,11) = (0.0\text{D0}, 0.0\text{D0})$

442

443 $\text{smrc2}(3,12) = (0.0\text{D0}, 0.0\text{D0})$

444

445 $\text{smrc2}(3,13) = (0.0\text{D0}, 0.0\text{D0})$

446

447

448

449 C BOUNDARY CONDITION #5 (EQ 218)

450

451 $\text{smrc2}(4,1) = \text{L2GC2} * \text{d2_SP_CY2_b1} + \text{lame_c2}/\text{b}$

452 $1 * \text{d1_SP_CY2_b1} - \text{lame_c2} * \text{SP_CY2_b1} * (\text{n2}/\text{b2} + \text{k2})$

453

454 $\text{smrc2}(4,2) = \text{L2GC2} * \text{d2_SP_CY2_b2} + \text{lame_c2}/\text{b}$

455 $1 * \text{d1_SP_CY2_b2} - \text{lame_c2} * \text{SP_CY2_b2} * (\text{n2}/\text{b2} + \text{k2})$

456

457 $\text{smrc2}(4,3) = -2.0\text{D0} * \text{shear_c2} * \text{n}/\text{b2} * (\text{VXP_CY2_b1} - \text{b}$

458 $1 * \text{d1_VXP_CY2_b1})$

```

459
460     smrc2(4,4) = -2.0D0*shear_c2*n/b2*(VXP_CY2_b2 - b
461 1*d1_VXP_CY2_b2)
462
463     smrc2(4,5) = (0.0D0,1.0D0)*2.0D0*shear_c2*k*d1_VRTP_CY2_b1
464
465     smrc2(4,6) = (0.0D0,1.0D0)*2.0D0*shear_c2*k*d1_VRTP_CY2_b2
466
467     smrc2(4,7) = -L2GC1*d2_SP_CY1_b1 - lame_c1/b*
468 1d1_SP_CY1_b1 + lame_c1*SP_CY1_b1*(n2/b2 + k2)
469
470     smrc2(4,8) = -L2GC1*d2_SP_CY1_b2 - lame_c1/b*
471 1d1_SP_CY1_b2 + lame_c1*SP_CY1_b2*(n2/b2 + k2)
472
473     smrc2(4,9) = 2.0D0*shear_c1*n/b2*(VXP_CY1_b1 -
474 1b*d1_VXP_CY1_b1)
475
476     smrc2(4,10) = 2.0D0*shear_c1*n/b2*(VXP_CY1_b2 -
477 1b*d1_VXP_CY1_b2)
478
479     smrc2(4,11) = -(0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_b1
480
481     smrc2(4,12) = -(0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_b2
482
483     smrc2(4,13) = (0.0D0,0.0D0)
484
485
486
487 C BOUNDARY CONDITION #6 (EQ 219)
488
489     smrc2(5,1) = (0.0D0,1.0D0)*k*2.0D0*shear_c2*d1_SP_CY2_b1
490
491     smrc2(5,2) = (0.0D0,1.0D0)*k*2.0D0*shear_c2*d1_SP_CY2_b2
492
493     smrc2(5,3) = (0.0D0,1.0D0)*k*n*shear_c2/b*VXP_CY2_b1
494
495     smrc2(5,4) = (0.0D0,1.0D0)*k*n*shear_c2/b*VXP_CY2_b2
496
497     smrc2(5,5) = shear_c2*(VRTP_CY2_b1*((n+1.0D0)/b2 - k2)
498 1- d1_VRTP_CY2_b1*(n+1.0D0)/b - d2_VRTP_CY2_b1)
499
500     smrc2(5,6) = shear_c2*(VRTP_CY2_b2*((n+1.0D0)/b2 - k2)
501 1- d1_VRTP_CY2_b2*(n+1.0D0)/b - d2_VRTP_CY2_b2)
502
503     smrc2(5,7) = (0.0D0,-1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_b1
504

```

```

505      smrc2(5,8) = (0.0D0,-1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_b2
506
507      smrc2(5,9) = (0.0D0,-1.0D0)*k*n*shear_c1/b*VXP_CY1_b1
508
509      smrc2(5,10) = (0.0D0,-1.0D0)*k*n*shear_c1/b*VXP_CY1_b2
510
511      smrc2(5,11) = -(shear_c1*(VRTP_CY1_b1*(n/b2 - k2 + 1.0D0/b2) -
512 1d1_VRTP_CY1_b1*(n/b+1.0D0/b) - d2_VRTP_CY1_b1))
513
514      smrc2(5,12) = -(shear_c1*(VRTP_CY1_b2*(n/b2 - k2 + 1.0D0/b2) -
515 1d1_VRTP_CY1_b2*(n/b+1.0D0/b) - d2_VRTP_CY1_b2))
516
517      smrc2(5,13) = (0.0D0,0.0D0)
518
519
520
521 C BOUNDARY CONDITION #7 (EQ 220)
522
523      smrc2(6,1) = shear_c2*2.0D0*n/b*(1.0D0/b*SP_CY2_b1
524 1- d1_SP_CY2_b1)
525
526      smrc2(6,2) = shear_c2*2.0D0*n/b*(1.0D0/b*SP_CY2_b2
527 1- d1_SP_CY2_b2)
528
529      smrc2(6,3) = shear_c2*(-d2_VXP_CY2_b1 + d1_VXP_CY2_b1/b
530 1 - n2/b2*VXP_CY2_b1)
531
532      smrc2(6,4) = shear_c2*(-d2_VXP_CY2_b2 + d1_VXP_CY2_b2/b
533 1 - n2/b2*VXP_CY2_b2)
534
535      smrc2(6,5) = (0.0D0,1.0D0)*k*shear_c2*(d1_VRTP_CY2_b1 -
536 1 VRTP_CY2_b1*(1.0D0+n)/b)
537
538      smrc2(6,6) = (0.0D0,1.0D0)*k*shear_c2*(d1_VRTP_CY2_b2 -
539 1 VRTP_CY2_b2*(1.0D0+n)/b)
540
541      smrc2(6,7) = -2.0D0*n*shear_c1/b*(1.0D0/b*SP_CY1_b1
542 1 - d1_SP_CY1_b1)
543
544      smrc2(6,8) = -2.0D0*n*shear_c1/b*(1.0D0/b*SP_CY1_b2
545 1 - d1_SP_CY1_b2)
546
547      smrc2(6,9) = -shear_c1*(-d2_VXP_CY1_b1 + d1_VXP_CY1_b1/b
548 1 - n2/b2*VXP_CY1_b1)
549
550      smrc2(6,10) = -shear_c1*(-d2_VXP_CY1_b2 + d1_VXP_CY1_b2/b

```

```

551 1 - n2/b2*VXP_CY1_b2)
552
553 smrc2(6,11) = (0.0D0,-1.0D0)*k*shear_c1*(d1_VRTP_CY1_b1 -
554 1VRTP_CY1_b1*(1.0/b + n/b))
555
556 smrc2(6,12) = (0.0D0,-1.0D0)*k*shear_c1*(d1_VRTP_CY1_b2 -
557 1VRTP_CY1_b2*(1.0D0/b + n/b))
558
559 smrc2(6,13) = (0.0D0,0.0D0)
560
561
562
563 C BOUNDARY CONDITION #8 (EQ 221)
564
565 smrc2(7,1) = d1_SP_cy2_b1
566
567 smrc2(7,2) = d1_SP_cy2_b2
568
569 smrc2(7,3) = n*VXP_CY2_b1/b
570
571 smrc2(7,4) = n*VXP_CY2_b2/b
572
573 smrc2(7,5) = (0.0D0,1.0D0)*k*VRTP_CY2_b1
574
575 smrc2(7,6) = (0.0D0,1.0D0)*k*VRTP_CY2_b2
576
577 smrc2(7,7) = -d1_SP_CY1_b1
578
579 smrc2(7,8) = -d1_SP_CY1_b2
580
581 smrc2(7,9) = -n*VXP_CY1_b1/b
582
583 smrc2(7,10) = -n*VXP_CY1_b2/b
584
585 smrc2(7,11) = (0.0D0,-1.0D0)*k*VRTP_CY1_b1
586
587 smrc2(7,12) = (0.0D0,-1.0D0)*k*VRTP_CY1_b2
588
589 smrc2(7,13) = (0.0D0,0.0D0)
590
591
592
593 C BOUNDARY CONDITION #9 (EQ 222)
594
595 smrc2(8,1) = -n*SP_CY2_b1/b
596

```

```

597      smrc2(8,2) = -n*SP_CY2_b2/b
598
599      smrc2(8,3) = -d1_VXP_CY2_b1
600
601      smrc2(8,4) = -d1_VXP_CY2_b2
602
603      smrc2(8,5) = (0.0D0,1.0D0)*k*VRTP_CY2_b1
604
605      smrc2(8,6) = (0.0D0,1.0D0)*k*VRTP_CY2_b2
606
607      smrc2(8,7) = n*SP_CY1_b1/b
608
609      smrc2(8,8) = n/b*SP_CY1_b2
610
611      smrc2(8,9) = d1_VXP_CY1_b1
612
613      smrc2(8,10) = d1_VXP_CY1_b2
614
615      smrc2(8,11) = (0.0D0,-1.0D0)*k*VRTP_CY1_b1
616
617      smrc2(8,12) = (0.0D0,-1.0D0)*k*VRTP_CY1_b2
618
619      smrc2(8,13) = (0.0D0,0.0D0)
620
621
622
623 C BOUNDARY CONDITION #10 (EQ 223)
624
625      smrc2(9,1) = (0.0D0,1.0D0)*k*SP_CY2_b1
626
627      smrc2(9,2) = (0.0D0,1.0D0)*k*SP_CY2_b2
628
629      smrc2(9,3) = (0.0D0,0.0D0)
630
631      smrc2(9,4) = (0.0D0,0.0D0)
632
633      smrc2(9,5) = -d1_VRTP_CY2_b1 - (n + 1.0D0)*VRTP_CY2_b1
634 1/b
635
636      smrc2(9,6) = -d1_VRTP_CY2_b2 - (n + 1.0D0)*VRTP_CY2_b2
637 1/b
638
639      smrc2(9,7) = (0.0D0,-1.0D0)*k*SP_CY1_b1
640
641      smrc2(9,8) = (0.0D0,-1.0D0)*k*SP_CY1_b2
642

```

```

643      smrc2(9,9) = (0.0D0,0.0D0)
644
645      smrc2(9,10) = (0.0D0,0.0D0)
646
647      smrc2(9,11) = d1_VRTP_CY1_b1 + VRTP_CY1_b1*(n+1.0D0)/b
648
649      smrc2(9,12) = d1_VRTP_CY1_b2 + VRTP_CY1_b2*(n+1.0D0)/b
650
651      smrc2(9,13) = (0.0D0,0.0D0)
652
653
654
655 C BOUNDARY CONDITION #11 (EQ 224)
656
657
658      smrc2(10,1) = (0.0D0,0.0D0)
659
660      smrc2(10,2) = (0.0D0,0.0D0)
661
662      smrc2(10,3) = (0.0D0,0.0D0)
663
664      smrc2(10,4) = (0.0D0,0.0D0)
665
666      smrc2(10,5) = (0.0D0,0.0D0)
667
668      smrc2(10,6) = (0.0D0,0.0D0)
669
670      smrc2(10,7) = L2GC1*d2_SP_CY1_a1 + (lame_c1/a)*
671 1d1_SP_CY1_a1 - lame_c1*SP_CY1_a1*(n2/a2 + k2)
672
673      smrc2(10,8) = L2GC1*d2_SP_CY1_a2 + (lame_c1/a)*
674 1d1_SP_CY1_a2 - lame_c1*SP_CY1_a2*(n2/a2 + k2)
675
676      smrc2(10,9) = -2.0D0*shear_c1*n/a2*(VXP_CY1_a1
677 1- a*d1_VXP_CY1_a1)
678
679      smrc2(10,10) = -2.0D0*shear_c1*n/a2*(VXP_CY1_a2
680 1- a*d1_VXP_CY1_a2)
681
682      smrc2(10,11) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_a1
683
684      smrc2(10,12) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_a2
685
686      smrc2(10,13) = +ri*Om**2*IFSC
687
688

```

```

689
690 C BOUNDARY CONDITION #12 (EQ 225)
691
692
693      smrc2(11,1) = (0.0D0,0.0D0)
694
695      smrc2(11,2) = (0.0D0,0.0D0)
696
697      smrc2(11,3) = (0.0D0,0.0D0)
698
699      smrc2(11,4) = (0.0D0,0.0D0)
700
701      smrc2(11,5) = (0.0D0,0.0D0)
702
703      smrc2(11,6) = (0.0D0,0.0D0)
704
705      smrc2(11,7) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_a1
706
707      smrc2(11,8) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_a2
708
709      smrc2(11,9) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_a1/a
710
711      smrc2(11,10) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_a2/a
712
713      smrc2(11,11) = shear_c1*(VRTP_CY1_a1*(n/a2 - k2 +
714 1 1.0D0/a2) - d1_VRTP_CY1_a1*(n + 1.0D0)/a - d2_VRTP_CY1_a1)
715
716      smrc2(11,12) = shear_c1*(VRTP_CY1_a2*(n/a2 - k2 +
717 1 1.0D0/a2) - d1_VRTP_CY1_a2*(n + 1.0D0)/a - d2_VRTP_CY1_a2)
718
719      smrc2(11,13) = (0.0D0,0.0D0)
720
721
722
723 C BOUNDARY CONDITION #13 (EQ 226)
724
725
726      smrc2(12,1) = (0.0D0,0.0D0)
727
728      smrc2(12,2) = (0.0D0,0.0D0)
729
730      smrc2(12,3) = (0.0D0,0.0D0)
731
732      smrc2(12,4) = (0.0D0,0.0D0)
733
734      smrc2(12,5) = (0.0D0,0.0D0)

```

```

735
736      smrc2(12,6) = (0.0D0,0.0D0)
737
738      smrc2(12,7) = (shear_c1*2.0D0*n/a)*((1.0D0/a)*
739 1SP_CY1_a1 - d1_SP_CY1_a1)
740
741      smrc2(12,8) = (shear_c1*2.0D0*n/a)*((1.0D0/a)*
742 1SP_CY1_a2 - d1_SP_CY1_a2)
743
744      smrc2(12,9) = shear_c1*(-d2_VXP_CY1_a1 + 1.0D0/a*
745 1d1_VXP_CY1_a1 - n2/a2*VXP_CY1_a1)
746
747      smrc2(12,10) = shear_c1*(-d2_VXP_CY1_a2 + 1.0D0/a*
748 1d1_VXP_CY1_a2 - n2/a2*VXP_CY1_a2)
749
750      smrc2(12,11) = (0.0D0,1.0D0)*k*shear_c1*(d1_V RTP_CY1_a1
751 1- VRTP_CY1_a1*(1.0D0 + n)/a)
752
753      smrc2(12,12) = (0.0D0,1.0D0)*k*shear_c1*(d1_V RTP_CY1_a2
754 1- VRTP_CY1_a2*(1.0D0 + n)/a)
755
756
757      smrc2(12,13) = (0.0D0,0.0D0)
758
759
760 C BOUNDARY CONDITION #14 (EQ 227)
761
762
763      smrc2(13,1) = (0.0D0,0.0D0)
764
765      smrc2(13,2) = (0.0D0,0.0D0)
766
767      smrc2(13,3) = (0.0D0,0.0D0)
768
769      smrc2(13,4) = (0.0D0,0.0D0)
770
771      smrc2(13,5) = (0.0D0,0.0D0)
772
773      smrc2(13,6) = (0.0D0,0.0D0)
774
775      smrc2(13,7) = d1_SP_CY1_a1
776
777      smrc2(13,8) = d1_SP_CY1_a2
778
779      smrc2(13,9) = n/a*VXP_CY1_a1
780

```

```

781      smrc2(13,10) = n/a*VXP_CY1_a2
782
783      smrc2(13,11) = (0.0D0,1.0D0)*k*VRTP_CY1_a1
784
785      smrc2(13,12) = (0.0D0,1.0D0)*k*VRTP_CY1_a2
786
787      smrc2(13,13) = -d1_IFSC
788
789
790 C END BOUNDARY CONDITIONS FOR MATRIX "smrc2"
791
792
793
794      return
795      end
796
797
798
799
800      SUBROUTINE ABC_RC2_INVERT(n,exctype,smrc2,a,c)
801
802
803 C EXTERNAL VARIABLES
804
805      integer n,exctype
806      real*8 a,c
807      complex*16 smrc2(13,13)
808
809 C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
810
811      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
812      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
813      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
814      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2
815
816      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
817      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
818      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
819      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
820
821      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
822      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
823      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
824      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2
825
826      complex*16 lame_c1,shear_c1,cl_c1,ct_c1

```

```

827      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
828
829      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
830 1      SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
831 1      SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
832 1      SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
833 1      VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
834 1      VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
835 1      VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
836 1      VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
837 1      VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
838 1      VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
839 1      VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
840 1      VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
841 1      lame_c1,shear_c1,cl_c1,ct_c1,
842 1      A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
843
844 C *****
845
846
847 C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/
848
849      complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
850      complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
851      complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
852      complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2
853
854      complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
855      complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
856      complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
857      complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2
858
859      complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1
860      complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2
861      complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1
862      complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2
863
864      complex*16 lame_c2,shear_c2,cl_c2,ct_c2
865
866      complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2
867
868
869      common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1,
870 1      SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,
871 1      SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,
872 1      SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,

```

```

873 1      VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,
874 1      VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,
875 1      VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,
876 1      VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,
877 1      VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,
878 1      VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,
879 1      VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,
880 1      VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,
881 1      lame_c2,shear_c2,cl_c2,ct_c2,
882 1      A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2
883
884 C *****
885
886
887
888 C DEFINITIONS FOR COMMON BLOCK /ROD/
889
890      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
891      1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
892      1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
893
894
895      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
896      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
897      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
898      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
899
900 C *****
901
902 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
903
904      complex*16 OFSC,d1_OFSC,M_OF
905
906      common /OFLUID/ OFSC,d1_OFSC,M_OF
907
908 C *****
909
910
911 C *****
912
913 C DEFINITIONS FOR COMMON BLOCK /IFLUID/
914
915      common /IFLUID/ IFSC,d1_IFSC,D_IF
916
917      complex*16 IFSC,d1_IFSC,D_IF
918

```

```

919 C *****
920
921
922 C INTERNAL VARIABLES
923
924     integer size,iflag
925     complex*16 smrc2inv(13,13),workrc2(13,26)
926
927     size = 13
928     iflag = 0
929
930
931     CALL MINV(smrc2,smrc2inv,workrc2,size,iflag)
932
933
934
935 C RADIAL EXCITATION exctype = 1
936 C AXIAL EXCITATION exctype = 0
937
938     if (exctype .eq. 1) then
939         A1_C2 = -smrc2inv(1,1)
940         A2_C2 = -smrc2inv(2,1)
941         B1_C2 = -smrc2inv(3,1)
942         B2_C2 = -smrc2inv(4,1)
943         C1_C2 = -smrc2inv(5,1)
944         C2_C2 = -smrc2inv(6,1)
945         A1_C1 = -smrc2inv(7,1)
946         A2_C1 = -smrc2inv(8,1)
947         B1_C1 = -smrc2inv(9,1)
948         B2_C1 = -smrc2inv(10,1)
949         C1_C1 = -smrc2inv(11,1)
950         C2_C1 = -smrc2inv(12,1)
951         D_IF = -smrc2inv(13,1)
952         M_OF = (A1_C2*d1_SP_CY2_c1 +
953 1           A2_C2*d1_SP_CY2_c2 +
954 1           B1_C2*n/c*VXP_CY2_c1 +
955 1           B2_C2*n/c*VXP_CY2_c2 +
956 1           C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c1 +
957 1           C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c2)/d1_OFSC
958
959     elseif (exctype .eq. 0) then
960         A1_C2 = -smrc2inv(1,2)
961         A2_C2 = -smrc2inv(2,2)
962         B1_C2 = -smrc2inv(3,2)
963         B2_C2 = -smrc2inv(4,2)
964         C1_C2 = -smrc2inv(5,2)

```

```

965      C2_C2 = -smrc2inv(6,2)
966      A1_C1 = -smrc2inv(7,2)
967      A2_C1 = -smrc2inv(8,2)
968      B1_C1 = -smrc2inv(9,2)
969      B2_C1 = -smrc2inv(10,2)
970      C1_C1 = -smrc2inv(11,2)
971      C2_C1 = -smrc2inv(12,2)
972      D_IF = -smrc2inv(13,2)
973      M_OF = (A1_C2*d1_SP_CY2_c1 +
974 1      A2_C2*d1_SP_CY2_c2 +
975 1      B1_C2*n/c*VXP_CY2_c1 +
976 1      B2_C2*n/c*VXP_CY2_c2 +
977 1      C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c1 +
978 1      C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c2)/d1_OFSC
979
980      endif
981
982
983
984      return
985      end
986
987
988
989
990
991      SUBROUTINE OUTPUT_RC2(tft,n,k,r,value)
992
993 C EXTERNAL VARIABLES
994
995      integer n,tft
996      real*8 k,r,zero
997      complex*16 value
998
999 C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
1000
1001      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
1002      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
1003      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
1004      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2
1005
1006      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
1007      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
1008      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
1009      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
1010

```

```

1011      complex*16 VRTP_CY1_a1,d1_V RTP_CY1_a1,d2_V RTP_CY1_a1
1012      complex*16 VRTP_CY1_a2,d1_V RTP_CY1_a2,d2_V RTP_CY1_a2
1013      complex*16 VRTP_CY1_b1,d1_V RTP_CY1_b1,d2_V RTP_CY1_b1
1014      complex*16 VRTP_CY1_b2,d1_V RTP_CY1_b2,d2_V RTP_CY1_b2
1015
1016      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
1017      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
1018
1019      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
1020      1      SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
1021      1      SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
1022      1      SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
1023      1      VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
1024      1      VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
1025      1      VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
1026      1      VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
1027      1      VRTP_CY1_a1,d1_V RTP_CY1_a1,d2_V RTP_CY1_a1,
1028      1      VRTP_CY1_a2,d1_V RTP_CY1_a2,d2_V RTP_CY1_a2,
1029      1      VRTP_CY1_b1,d1_V RTP_CY1_b1,d2_V RTP_CY1_b1,
1030      1      VRTP_CY1_b2,d1_V RTP_CY1_b2,d2_V RTP_CY1_b2,
1031      1      lame_c1,shear_c1,cl_c1,ct_c1,
1032      1      A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
1033
1034 C *****
1035
1036
1037 C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/
1038
1039      complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
1040      complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
1041      complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
1042      complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2
1043
1044      complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
1045      complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
1046      complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
1047      complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2
1048
1049      complex*16 VRTP_CY2_b1,d1_V RTP_CY2_b1,d2_V RTP_CY2_b1
1050      complex*16 VRTP_CY2_b2,d1_V RTP_CY2_b2,d2_V RTP_CY2_b2
1051      complex*16 VRTP_CY2_c1,d1_V RTP_CY2_c1,d2_V RTP_CY2_c1
1052      complex*16 VRTP_CY2_c2,d1_V RTP_CY2_c2,d2_V RTP_CY2_c2
1053
1054      complex*16 lame_c2,shear_c2,cl_c2,ct_c2
1055
1056      complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2

```

```

1057
1058
1059      common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1, .
1060 1      SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,
1061 1      SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,
1062 1      SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,
1063 1      VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,
1064 1      VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,
1065 1      VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,
1066 1      VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,
1067 1      VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,
1068 1      VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,
1069 1      VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,
1070 1      VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,
1071 1      lame_c2,shear_c2,c1_c2,ct_c2,
1072 1      A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2
1073
1074 C *****
1075
1076
1077 C DEFINITIONS FOR COMMON BLOCK /ROD/
1078
1079      common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
1080 1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
1081 1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
1082
1083
1084      complex*16 SP_rod,d1_SP_rod,d2_SP_rod
1085      complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
1086      complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
1087      complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
1088
1089 C *****
1090
1091 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
1092
1093      complex*16 OFSC,d1_OFSC,M_OF
1094
1095      common /OFLUID/ OFSC,d1_OFSC,M_OF
1096
1097 C *****
1098
1099
1100 C *****
1101
1102 C DEFINITIONS FOR COMMON BLOCK /IFLUID/

```

```

1103
1104      common /IFLUID/ IFSC,d1_IFSC,D_IF
1105
1106      complex*16 IFSC,d1_IFSC,D_IF
1107
1108 C *****
1109
1110
1111 C INTERNAL VARIABLES
1112
1113      integer n2
1114      real*8 k2,r2,no,p11,p12
1115      complex*16 dpp,exx,err,Srr,Sxx,ett,uc,vc,wc
1116      complex*16 A1C2,A2C2,B1C2,B2C2,C1C2,C2C2
1117      complex*16 L2GC1,L2GC2
1118
1119      n2 = n**2
1120      r2 = r**2
1121      k2 = k**2
1122
1123      zero = 1.000D-30
1124      no  = 1.460D0
1125      p11 = 0.126D0
1126      p12 = 0.270D0
1127      L2GC1 = lame_c1 + 2.0D0*shear_c1
1128      L2GC2 = lame_c2 + 2.0D0*shear_c2
1129
1130
1131 C RADIAL STRESS/(Pr or Px)          tft = 0
1132 C LONGITUDINAL STRESS              tft = 1
1133 C AXIAL DISPLACEMENT              tft = 2
1134 C THETA DISPLACEMENT              tft = 3
1135 C RADIAL DISPLACEMENT             tft = 4
1136 C LONGITUDINAL STRAIN  e11/(Pr or Px)  tft = 5
1137 C THETA STRAIN          ett/(Pr or Px)  tft = 6
1138 C RADIAL STRAIN         err/(Pr or Px)  tft = 7
1139 C                      tft = 8
1140 C OPTIC exx=0@k=0        ((dp/p)(r))/(Pr or Px)  tft = 9
1141 C OPTIC exx=const@k=0   ((dp/p)(r))/(Pr or Px)  tft = 10
1142
1143
1144
1145
1146
1147
1148      if (tft .eq. 0) then

```

```

1149
1150      A1C2 = A1_C2*(L2GC2*d2_SP_CY2_b1 + lame_c2*
1151 1      d1_SP_CY2_b1/r - lame_c2*SP_CY2_b1*(n2/
1152 1      r2 + k2))
1153
1154      A2C2 = A2_C2*(L2GC2*d2_SP_CY2_b2 + lame_c2*
1155 1      d1_SP_CY2_b2/r - lame_c2* SP_CY2_b2*(n2/
1156 1      r2 + k2))
1157
1158      B1C2 = B1_C2*-2.0D0*shear_c2*n/r**2*(VXP_CY2_b1 -
1159 1      r*d1_VXP_CY2_b1)
1160
1161      B2C2 = B2_C2*-2.0D0*shear_c2*n/r**2*(VXP_CY2_b2 -
1162 1      r*d1_VXP_CY2_b2)
1163
1164      C1C2 = (0.0D0,1.0D0)*C1_C2*2.0D0*shear_c2*k*d1_VRTP_CY2_b1
1165
1166      C2C2 = (0.0D0,1.0D0)*C2_C2*2.0D0*shear_c2*k*d1_VRTP_CY2_b2
1167
1168      Srr = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2
1169
1170      value = Srr
1171
1172
1173 C LONGITUDINAL STRESS USING (EQ 153)
1174      elseif(tft.eq. 1) then
1175
1176      A1C2 = A1_C2*(lame_c2*(d2_SP_CY2_b1 + d1_SP_CY2_b1/r)
1177 1      - SP_CY2_b1*(lame_c2*n2/r2 + k2*(lame_c2
1178 1      + 2.0D0*shear_c2)))
1179
1180      A2C2 = A2_C2*(lame_c2*(d2_SP_CY2_b2 + d1_SP_CY2_b2/r)
1181 1      - SP_CY2_b2*(lame_c2*n2/r2 + k2*(lame_c2
1182 1      + 2.0D0*shear_c2)))
1183
1184      C1C2 = C1_C2*((0.0D0,-1.0D0)*k*(d1_VRTP_CY2_b1 +
1185 1      VRTP_CY2_b1*(n + 1.0D0)/r))*2.0D0*shear_c2
1186
1187      C2C2 = C2_C2*((0.0D0,-1.0D0)*k*(d1_VRTP_CY2_b2 +
1188 1      VRTP_CY2_b2*(n + 1.0D0)/r))*2.0D0*shear_c2
1189
1190      Sxx = A1C2 + A2C2 + C1C2 + C2C2
1191
1192      value = Sxx
1193
1194

```

```

1195 C AXIAL DISPLACEMENT USING (EQ 149)
1196     elseif(tft .eq. 2) then
1197
1198          $A1C2 = A1\_C2 * (0.0D0, 1.0D0) * k * SP\_CY2\_b1$ 
1199
1200          $A2C2 = A2\_C2 * (0.0D0, 1.0D0) * k * SP\_CY2\_b2$ 
1201
1202          $C1C2 = C1\_C2 * (-d1\_VRTP\_CY2\_b1 - VRTP\_CY2\_b1 * (n + 1.0D0) / r)$ 
1203     1
1204
1205          $C2C2 = C2\_C2 * (-d1\_VRTP\_CY2\_b2 - VRTP\_CY2\_b2 * (n + 1.0D0) / r)$ 
1206     1
1207
1208          $wc = A1C2 + A2C2 + C1C2 + C2C2$ 
1209
1210         value = wc
1211
1212 C THETA DISPLACEMENT USING (EQ 150)
1213     elseif(tft .eq. 3) then
1214
1215
1216          $A1C2 = A1\_C2 * -SP\_CY2\_b1 * n / r$ 
1217
1218          $A2C2 = A2\_C2 * -SP\_CY2\_b2 * n / r$ 
1219
1220          $B1C2 = -B1\_C2 * d1\_VXP\_CY2\_b1$ 
1221
1222          $B2C2 = -B2\_C2 * d1\_VXP\_CY2\_b2$ 
1223
1224          $C1C2 = C1\_C2 * (0.0D0, 1.0D0) * k * VRTP\_CY2\_b1$ 
1225
1226          $C2C2 = C2\_C2 * (0.0D0, 1.0D0) * k * VRTP\_CY2\_b2$ 
1227
1228          $vc = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2$ 
1229
1230         value = vc
1231
1232 C RADIAL DISPLACEMENT USING (EQ 148)
1233     elseif(tft .eq. 4) then
1234
1235          $A1C2 = A1\_C2 * d1\_SP\_CY2\_b1$ 
1236
1237          $A2C2 = A2\_C2 * d1\_SP\_CY2\_b2$ 
1238
1239          $B1C2 = B1\_C2 * VXP\_CY2\_b1 * n / r$ 
1240

```

```

1241      B2C2 = B2_C2*VXP_CY2_b2*n/r
1242
1243      C1C2 = C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b1
1244
1245      C2C2 = C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b2
1246
1247      uc = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2
1248
1249      value = uc
1250
1251 C LONGITUDINAL STRAIN USING (EQ 147)
1252     elseif(tft .eq. 5) then
1253
1254         A1C2 = -A1_C2*k2*SP_CY2_b1
1255
1256         A2C2 = -A2_C2*k2*SP_CY2_b2
1257
1258         C1C2 = C1_C2*(0.0D0,-1.0D0)*k*(d1_VRTP_CY2_b1 +
1259 1      VRTP_CY2_b1*(n + 1.0D0)/r)
1260
1261         C2C2 = C2_C2*(0.0D0,-1.0D0)*k*(d1_VRTP_CY2_b2 +
1262 1      VRTP_CY2_b2*(n + 1.0D0)/r)
1263
1264         exx = A1C2 + A2C2 + C1C2 + C2C2
1265
1266         if (zabs(exx) .lt. dabs(zero)) then
1267             value = zero
1268         endif
1269
1270         value = exx
1271
1272 C THETA STRAIN USING (EQ 145)
1273     elseif(tft .eq. 6) then
1274
1275         A1C2 = A1_C2*(d1_SP_CY2_b1/r - SP_CY2_b1*n2/r2)
1276
1277         A2C2 = A2_C2*(d1_SP_CY2_b2/r - SP_CY2_b2*n2/r2)
1278
1279         B1C2 = B1_C2*(-d1_VXP_CY2_b1*n/r + VXP_CY2_b1*n/r2)
1280
1281         B2C2 = B2_C2*(-d1_VXP_CY2_b2*n/r + VXP_CY2_b2*n/r2)
1282
1283         C1C2 = C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b1*(n + 1.0D0)/r
1284
1285         C2C2 = C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b2*(n + 1.0D0)/r
1286

```

```

1287     ett = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2
1288
1289     if (zabs(ett) .lt. dabs(zero)) then
1290         value = zero
1291     endif
1292
1293     value = ett
1294
1295 C RADIAL STRAIN USING (eq 143)
1296     elseif(tft .eq. 7) then
1297
1298         A1C2 = A1_C2*d2_SP_CY2_b1
1299
1300         A2C2 = A2_C2*d2_SP_CY2_b2
1301
1302         B1C2 = B1_C2*(d1_VXP_CY2_b1 - VXP_CY2_b1/r)*n/r
1303
1304         B2C2 = B2_C2*(d1_VXP_CY2_b2 - VXP_CY2_b2/r)*n/r
1305
1306         C1C2 = C1_C2*(0.0D0,1.0D0)*k*d1_VRTP_CY2_b1
1307
1308         C2C2 = C2_C2*(0.0D0,1.0D0)*k*d1_VRTP_CY2_b2
1309
1310         err = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2
1311
1312         if (zabs(err) .lt. dabs(zero)) then
1313             value = zero
1314         endif
1315
1316         value = err
1317
1318 C UNUSED AT PRESENT
1319     elseif(tft .eq. 8) then
1320         value = 1.0D0
1321
1322 C OPTIC dP/P/uPa USING (EQ 147 & 143)
1323 C exx = 0 @ k = 0
1324     elseif(tft .eq. 9) then
1325
1326         A1C2 = -A1_C2*k2*SP_CY2_b1
1327
1328         A2C2 = -A2_C2*k2*SP_CY2_b2
1329
1330         C1C2 = C1_C2*(0.0D0,-1.0D0)*k*(d1_VRTP_CY2_b1 +
1331 1 VRTP_CY2_b1*(n + 1.0D0)/r)
1332

```

```

1333      C2C2 = C2_C2*(0.0D0,-1.0D0)*k*(d1_V RTP_CY2_b2 +
1334 1      VRTP_CY2_b2*(n + 1.0D0)/r)
1335
1336      exx = A1C2 + A2C2 + C1C2 + C2C2
1337
1338      if (zabs(exx) .lt. dabs(zero)) then
1339          exx = zero
1340      endif
1341
1342      A1C2 = A1_C2*d2_SP_CY2_b1
1343
1344      A2C2 = A2_C2*d2_SP_CY2_b2
1345
1346      B1C2 = B1_C2*(d1_VXP_CY2_b1 - VXP_CY2_b1/r)*n/r
1347
1348      B2C2 = B2_C2*(d1_VXP_CY2_b2 - VXP_CY2_b2/r)*n/r
1349
1350      C1C2 = C1_C2*(0.0D0,1.0D0)*k*d1_V RTP_CY2_b1
1351
1352      C2C2 = C2_C2*(0.0D0,1.0D0)*k*d1_V RTP_CY2_b2
1353
1354      err = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2
1355
1356      if (zabs(err) .lt. dabs(zero)) then
1357          err = zero
1358      endif
1359
1360      dpp = exx - (no**2/2.0D0)*((p11+p12)*err + p12*exx)
1361
1362      value = dpp
1363
1364  C OPTIC dp/p/uPa USING (EQ 145 & 143)
1365  C exx = constant @ k = 0
1366      elseif(tft .eq. 10) then
1367
1368      A1C2 = A1_C2*(d1_SP_CY2_b1/r - SP_CY2_b1*n2/r2)
1369
1370      A2C2 = A2_C2*(d1_SP_CY2_b2/r - SP_CY2_b2*n2/r2)
1371
1372      B1C2 = B1_C2*(-d1_VXP_CY2_b1*n/r + VXP_CY2_b1*n/r2)
1373
1374      B2C2 = B2_C2*(-d1_VXP_CY2_b2*n/r + VXP_CY2_b2*n/r2)
1375
1376      C1C2 = C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b1*(n + 1.0D0)/r
1377
1378      C2C2 = C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_b2*(n + 1.0D0)/r

```

```

1379
1380     ett = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2
1381
1382     if (zabs(ett) .lt. dabs(zero)) then
1383         value = zero
1384     endif
1385
1386     A1C2 = A1_C2*d2_SP_CY2_b1
1387
1388     A2C2 = A2_C2*d2_SP_CY2_b2
1389
1390     B1C2 = B1_C2*(d1_VXP_CY2_b1 - VXP_CY2_b1/r)*n/r
1391
1392     B2C2 = B2_C2*(d1_VXP_CY2_b2 - VXP_CY2_b2/r)*n/r
1393
1394     C1C2 = C1_C2*(0.0D0,1.0D0)*k*d1_VRTP_CY2_b1
1395
1396     C2C2 = C2_C2*(0.0D0,1.0D0)*k*d1_VRTP_CY2_b2
1397
1398     err = A1C2 + A2C2 + B1C2 + B2C2 + C1C2 + C2C2
1399
1400     if (zabs(err) .lt. dabs(zero)) then
1401         err = zero
1402     endif
1403
1404     exx = -(lame_rod/(lame_rod + 2.0D0 * shear_rod))*(ett + err)
1405     dpp = exx - (no**2/2.0D0)*((p11+p12)*err + p12*exx)
1406     value = dpp
1407
1408 endif
1409
1410
1411
1412 return
1413 end
1414
1415

```

LISTING FOR fluids.f

fluids.f

Sat Jun 10 14:41:58 1995

```

1  C *****
2  C
3  C          subprogram "fluids.f"
4  C
5  C This subprogram was written and developed by Mark S. Peloquin
6  C
7  C at NUWCDETNLON 6/10/95. As of 6/10/95, there are no known bugs.
8  C
9  C Please notify the author if bugs are found (203) 440-5433.
10 C
11 C *****
12
13
14     SUBROUTINE IFL_POT(n,r,k,Om,ci)
15
16 C EXTERNAL VARIABLES
17
18     integer n
19     real*8 r,k,Om,ci
20
21 C *****
22
23 C DEFINITIONS FOR COMMON BLOCK /IFLUID/
24
25     complex*16 IFSC,d1_IFSC,D_IF
26
27     common /IFLUID/ IFSC,d1_IFSC,D_IF
28
29
30 C *****
31
32     double complex cbessj,d1cbessj
33
34
35 C INTERNAL VARIABLES
36
37     complex*16 g1,arg
38
39     arg = DCMLPX(((Om/ci)**2 - k**2), 0.0D0)
40     g1 = zsqrtr(arg)
41
42     d1_IFSC = d1cbessj(n,g1,r)
43     IFSC = cbessj(n,g1,r)
44

```

```

45      return
46      end
47
48
49
50
51
52      SUBROUTINE OFL_POT(n,r,k,Om,co)
53
54      integer n
55      real*8 r,k,Om,co
56
57 C *****
58
59 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
60
61      common /OFLUID/ OFSC,d1_OFSC,M_OF
62
63      complex*16 OFSC,d1_OFSC,M_OF
64
65 C *****
66
67      complex*16 g2,f2
68      double complex cbessh1,d1cbessh1,cbessk,d1cbessk
69
70      if (dabs(k) .le. dabs(Om/co)) then
71
72          g2 = dsqrt((Om/co)**2 - k**2)
73          d1_OFSC = d1cbessh1(n,g2,r)
74          OFSC = cbessh1(n,g2,r)
75
76      else
77
78          f2 = dsqrt(k**2 - (Om/co)**2)
79          d1_OFSC = d1cbessk(n,f2,r)
80          OFSC = cbessk(n,f2,r)
81
82      endif
83
84      return
85      end
86
87
88
89
90

```

```

91      SUBROUTINE OUTPUT_IF(tft,n,k,r,value,Om,ri)
92
93 C EXTERNAL VARIABLES
94
95      integer n,tft
96      real*8 k,r,Om,ri
97      complex*16 value
98 C *****
99
100 C DEFINITIONS FOR COMMON BLOCK /IFLUID/
101
102      complex*16 IFSC,d1_IFSC,D_IF
103
104      common /IFLUID/ IFSC,d1_IFSC,D_IF
105
106
107 C *****
108
109 C INTERNAL VARIABLES
110
111      complex*16 pi,vui
112
113 C OUTER FLUID PRESSURE TRANSFER FUNCTION USING (EQ F.14)
114      if (tft .eq. 0)then
115
116          pi = Om**2*ri*D_IF*IFSC
117
118          value = pi
119
120 C OUTER FLUID VELOCITY USING (EQ F.16)
121      elseif(tft .eq. 8)then
122
123          vui = (0.0D0,-1.0D0)*Om*D_IF*d1_IFSC
124
125          value = vui
126
127      endif
128
129      return
130      end
131
132
133

```

```

134      SUBROUTINE OUTPUT_OF(tft,n,k,r,value,Om,ro)
135
136 C EXTERNAL VARIABLES
137
138      integer n,tft
139      real*8 k,r,Om,ro
140      complex*16 value
141 C *****
142
143 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
144
145      common /OFLUID/ OFSC,d1_OFSC,M_OF
146
147      complex*16 OFSC,d1_OFSC,M_OF
148
149 C *****
150
151 C INTERNAL VARIABLES
152
153      complex*16 ps,vws
154
155 C OUTER FLUID PRESSURE TRANSFER FUNCTION USING (EQ F.7)
156      if (tft .eq. 0)then
157
158          ps = Om**2*ro*M_OF*OFSC
159
160          value = ps
161
162 C OUTER FLUID VELOCITY USING (EQ F.8a)
163      elseif(tft .eq. 8)then
164
165          vws = (0.0D0,-1.0D0)*Om*M_OF*d1_OFSC
166
167          value = vws
168
169      endif
170
171      return
172      end
173
174
175

```

LISTING FOR smc1.f

```

1  C *****
2  C
3  C          subprogram "smc1.f"
4  C
5  C This subprogram was written and developed by Mark S. Peloquin
6  C
7  C at NUWCDETNLON 6/10/95. As of 6/10/95, there are no known bugs.
8  C
9  C Please notify the author if bugs are found (203) 440-5433.
10 C
11 C *****
12
13
14      SUBROUTINE SYS_MATRIX_C1(n,k,ao_1cyl,bo_c1,
15 1Om,ro,co,ri,ci,smc1)
16
17 C EXTERNAL VARIABLES
18
19      integer n
20      real*8 k,ao_1cyl,bo_c1,Om,ro,co,ri,ci
21      complex*16 smc1(7,7)
22
23 C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
24
25      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
26      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
27      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
28      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2
29
30      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
31      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
32      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
33      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
34
35      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
36      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
37      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
38      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2
39
40      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
41
42      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
43
44      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
45 1      SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
46 1      SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,

```

```

47 1      SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
48 1      VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
49 1      VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
50 1      VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
51 1      VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
52 1      VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
53 1      VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
54 1      VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
55 1      VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
56 1      lame_c1,shear_c1,cl_c1,ct_c1,
57 1      A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
58
59 C *****
60
61
62 C *****
63
64 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
65
66      common /OFLUID/ OFSC,d1_OFSC,M_OF
67
68      complex*16 OFSC,d1_OFSC,M_OF
69
70 C *****
71
72
73 C *****
74
75 C DEFINITIONS FOR COMMON BLOCK /IFLUID/
76
77      common /IFLUID/ IFSC,d1_IFSC,D_IF
78
79      complex*16 IFSC,d1_IFSC,D_IF
80
81 C *****
82
83
84 C INTERNAL VARIABLES
85
86      complex*16 L2GC1
87      real*8 a,b,b2,k2,a2
88      integer n2
89
90      a   = ao_1cyl
91      b   = bo_c1
92      a2  = ao_1cyl**2

```

```

93      b2  = bo_c1**2
94      n2  = n**2
95      k2  = k**2
96      L2GC1 = lame_c1 + 2.0D0 * shear_c1
97
98
99 C BOUNDARY CONDITION #1 (EQ 113)
100
101      smc1(1,1) = L2GC1*d2_SP_CY1_b1 + (lame_c1/bo_c1)*
102      1d1_SP_CY1_b1 - lame_c1*SP_CY1_b1*(n2/b2 + k2) +
103      1ro*Om**2*OFSC/d1_OFSC*d1_SP_CY1_b1
104
105      smc1(1,2) = L2GC1*d2_SP_CY1_b2 + (lame_c1/bo_c1)*
106      1d1_SP_CY1_b2 - lame_c1*SP_CY1_b2*(n2/b2 + k2) +
107      1ro*Om**2*OFSC/d1_OFSC*d1_SP_CY1_b2
108
109      smc1(1,3) = -2.0D0*shear_c1*n/b2*(VXP_CY1_b1-bo_c1*
110      1d1_VXP_CY1_b1) +
111      1n*ro*Om**2*OFSC*VXP_CY1_b1/(d1_OFSC*b)
112
113      smc1(1,4) = -2.0D0*shear_c1*n/b2*(VXP_CY1_b2-bo_c1*
114      1d1_VXP_CY1_b2) +
115      1n*ro*Om**2*OFSC*VXP_CY1_b2/(d1_OFSC*b)
116
117      smc1(1,5) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_b1
118      1 + (0.0D0,1.0D0)*k*ro*Om**2*OFSC*VRTP_CY1_b1/d1_OFSC
119
120      smc1(1,6) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_b2
121      1 + (0.0D0,1.0D0)*k*ro*Om**2*OFSC*VRTP_CY1_b2/d1_OFSC
122
123      smc1(1,7) = (0.0D0,0.0D0)
124
125
126
127 C BOUNDARY CONDITION #2 (EQ 116)
128
129      smc1(2,1) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_b1
130
131      smc1(2,2) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_b2
132
133      smc1(2,3) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_b1/bo_c1
134
135      smc1(2,4) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_b2/bo_c1
136
137      smc1(2,5) = shear_c1*(VRTP_CY1_b1*(n/b2 - k2 +
138      1 1.0D0/b2) - d1_VRTP_CY1_b1*(n + 1.0D0)/bo_c1 - d2_VRTP_CY1_b1)

```

```

139
140      smc1(2,6) = shear_c1*(VRTP_CY1_b2*(n/b2 - k2 +
141 1 1.0D0/b2) - d1_VRTP_CY1_b2*(n + 1.0D0)/bo_c1 - d2_VRTP_CY1_b2)
142
143      smc1(2,7) = (0.0D0,0.0D0)
144
145
146
147
148 C BOUNDARY CONDITION #3 (EQ 120)
149
150      smc1(3,1) = (shear_c1*2.0D0*n/bo_c1)*((1.0D0/bo_c1)*
151 1SP_CY1_b1 - d1_SP_CY1_b1)
152
153      smc1(3,2) = (shear_c1*2.0D0*n/bo_c1)*((1.0D0/bo_c1)*
154 1SP_CY1_b2 - d1_SP_CY1_b2)
155
156      smc1(3,3) = shear_c1*(-d2_VXP_CY1_b1 + 1.0D0/bo_c1*
157 1d1_VXP_CY1_b1 - n2/b2*VXP_CY1_b1)
158
159      smc1(3,4) = shear_c1*(-d2_VXP_CY1_b2 + 1.0D0/bo_c1*
160 1d1_VXP_CY1_b2 - n2/b2*VXP_CY1_b2)
161
162      smc1(3,5) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_b1
163 1- VRTP_CY1_b1*(1.0D0 + n)/bo_c1)
164
165      smc1(3,6) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_b2
166 1- VRTP_CY1_b2*(1.0D0 + n)/bo_c1)
167
168      smc1(3,7) = (0.0D0,0.0D0)
169
170
171
172
173 C BOUNDARY CONDITION #11 (EQ 224)
174
175
176
177
178      smc1(4,1) = L2GC1*d2_SP_CY1_a1 + (lame_c1/a)*
179 1d1_SP_CY1_a1 - lame_c1*SP_CY1_a1*(n2/a2 + k2)
180
181      smc1(4,2) = L2GC1*d2_SP_CY1_a2 + (lame_c1/a)*
182 1d1_SP_CY1_a2 - lame_c1*SP_CY1_a2*(n2/a2 + k2)
183
184      smc1(4,3) = -2.0D0*shear_c1*n/a2*(VXP_CY1_a1-a*

```

```

185   1d1_VXP_CY1_a1)
186
187       smc1(4,4) = -2.0D0*shear_c1*n/a2*(VXP_CY1_a2-a*
188   1d1_VXP_CY1_a2)
189
190       smc1(4,5) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_a1
191
192       smc1(4,6) = (0.0D0,1.0D0)*2.0D0*shear_c1*k*d1_VRTP_CY1_a2
193
194       smc1(4,7) = + ri*Om**2*IFSC
195
196
197 C BOUNDARY CONDITION #12 (EQ 225)
198
199
200
201       smc1(5,1) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_a1
202
203       smc1(5,2) = (0.0D0,1.0D0)*k*2.0D0*shear_c1*d1_SP_CY1_a2
204
205       smc1(5,3) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_a1/a
206
207       smc1(5,4) = (0.0D0,1.0D0)*k*n*shear_c1*VXP_CY1_a2/a
208
209       smc1(5,5) = shear_c1*(VRTP_CY1_a1*((n+1.0D0)/a2 - k2)
210   1- d1_VRTP_CY1_a1*(n+1.0D0)/ao_1cyl - d2_VRTP_CY1_a1)
211
212       smc1(5,6) = shear_c1*(VRTP_CY1_a2*((n+1.0D0)/a2 - k2)
213   1- d1_VRTP_CY1_a2*(n+1.0D0)/ao_1cyl - d2_VRTP_CY1_a2)
214
215       smc1(5,7) = (0.0D0,0.0D0)
216
217
218
219
220
221 C BOUNDARY CONDITION #13 (EQ 226)
222
223
224
225       smc1(6,1) = (shear_c1*2.0D0*n/a)*((1.0D0/a)*
226   1SP_CY1_a1 - d1_SP_CY1_a1)
227
228       smc1(6,2) = (shear_c1*2.0D0*n/a)*((1.0D0/a)*
229   1SP_CY1_a2 - d1_SP_CY1_a2)
230

```

```

231      smc1(6,3) = shear_c1*(-d2_VXP_CY1_a1 + 1.0D0/a*
232      1d1_VXP_CY1_a1 - n2/a2*VXP_CY1_a1)
233
234      smc1(6,4) = shear_c1*(-d2_VXP_CY1_a2 + 1.0D0/a*
235      1d1_VXP_CY1_a2 - n2/a2*VXP_CY1_a2)
236
237      smc1(6,5) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_a1
238      1- VRTP_CY1_a1*(1.0D0 + n)/a)
239
240      smc1(6,6) = (0.0D0,1.0D0)*k*shear_c1*(d1_VRTP_CY1_a2
241      1- VRTP_CY1_a2*(1.0D0 + n)/a)
242
243      smc1(6,7) = (0.0D0,0.0D0)
244
245
246
247
248 C BOUNDARY CONDITION #14 (EQ 227)
249
250
251
252      smc1(7,1) = d1_SP_CY1_a1
253
254      smc1(7,2) = d1_SP_CY1_a2
255
256      smc1(7,3) = n/a*VXP_CY1_a1
257
258      smc1(7,4) = n/a*VXP_CY1_a2
259
260      smc1(7,5) = (0.0D0,1.0D0)*k*VRTP_CY1_a1
261
262      smc1(7,6) = (0.0D0,1.0D0)*k*VRTP_CY1_a2
263
264      smc1(7,7) = -d1_IFSC
265
266
267
268 C END BOUNDARY CONDITIONS FOR MATRIX "smc1"
269
270
271
272      return
273      end
274
275
276

```

```

277
278
279      SUBROUTINE ABC_C1_INVERT(n,exctype,smc1,a,b)
280
281
282 C EXTERNAL VARIABLES
283
284      integer n,exctype
285      real*8 a,b
286      complex*16 smc1(7,7)
287
288 C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/
289
290      complex*16 SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1
291      complex*16 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2
292      complex*16 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1
293      complex*16 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2
294
295      complex*16 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1
296      complex*16 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2
297      complex*16 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1
298      complex*16 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2
299
300      complex*16 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1
301      complex*16 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2
302      complex*16 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1
303      complex*16 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2
304
305      complex*16 lame_c1,shear_c1,cl_c1,ct_c1
306      complex*16 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
307
308      common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
309 1      SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
310 1      SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
311 1      SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
312 1      VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
313 1      VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
314 1      VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
315 1      VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
316 1      VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
317 1      VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
318 1      VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
319 1      VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
320 1      lame_c1,shear_c1,cl_c1,ct_c1,
321 1      A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1
322

```

```

323 C *****
324
325
326 C *****
327
328 C DEFINITIONS FOR COMMON BLOCK /OFLUID/
329
330     complex*16 OFSC,d1_OFSC,M_OF
331
332     common /OFLUID/ OFSC,d1_OFSC,M_OF
333
334 C *****
335
336
337 C *****
338
339 C DEFINITIONS FOR COMMON BLOCK /IFLUID/
340
341     common /IFLUID/ IFSC,d1_IFSC,D_IF
342
343     complex*16 IFSC,d1_IFSC,D_IF
344
345 C *****
346
347
348 C INTERNAL VARIABLES
349
350     integer size,iflag
351     complex*16 smc1inv(7,7),workc1(7,14)
352
353     size = 7
354     iflag = 0
355
356
357     CALL MINV(smc1,smc1inv,workc1,size,iflag)
358
359
360
361
362
363 C RADIAL EXCITATION exctype = 1
364 C AXIAL EXCITATION exctype = 0
365
366     if (exctype .eq. 1) then
367         A1_C1 = -smc1inv(1,1)
368         A2_C1 = -smc1inv(2,1)

```

```

369      B1_C1 = -smc1inv(3,1)
370      B2_C1 = -smc1inv(4,1)
371      C1_C1 = -smc1inv(5,1)
372      C2_C1 = -smc1inv(6,1)
373      M_OF = (A1_C1*d1_SP_CY1_b1 +
374 1          A2_C1*d1_SP_CY1_b2 +
375 1          B1_C1*n/b*VXP_CY1_b1 +
376 1          B2_C1*n/b*VXP_CY1_b2 +
377 1          C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b1 +
378 1          C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b2)/d1_OFSC
379      D_IF = -smc1inv(7,1)
380      elseif (exctype .eq. 0) then
381          A1_C1 = -smc1inv(1,2)
382          A2_C1 = -smc1inv(2,2)
383          B1_C1 = -smc1inv(3,2)
384          B2_C1 = -smc1inv(4,2)
385          C1_C1 = -smc1inv(5,2)
386          C2_C1 = -smc1inv(6,2)
387          M_OF = (A1_C1*d1_SP_CY1_b1 +
388 1              A2_C1*d1_SP_CY1_b2 +
389 1              B1_C1*n/b*VXP_CY1_b1 +
390 1              B2_C1*n/b*VXP_CY1_b2 +
391 1              C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b1 +
392 1              C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b2)/d1_OFSC
393          D_IF = -smc1inv(7,2)
394      endif
395
396
397
398      return
399      end
400
401
402

```

CROSS REFERENCE

SUBPROGRAMS

Cross Reference Thu Oct 19 16:48:44 1995

Subprograms :

Symbol	File/Subprogram	Line
-----	-----	----
abc_c1_invert		
	mr2cf.f/MAIN	316
	smc1.f/abc_c1_invert	279D
abc_rc1_invert		
	c1.f/abc_rc1_invert	533D
	mr2cf.f/MAIN	388
abc_rc2_invert		
	c2.f/abc_rc2_invert	800D
	mr2cf.f/MAIN	433
abc_rod_invert		
	" / "	356
	rf.f/abc_rod_invert	201D
abc_rod_solve		
	" /abc_rod_solve	152D
c1a_pot	c1.f/c1a_pot	14D
	mr2cf.f/MAIN	310, 326, 382, 397, 421, 445
c1b_pot	c1.f/c1b_pot	100D
	mr2cf.f/MAIN	311, 383, 422
c2b_pot	c2.f/c2b_pot	17D
	mr2cf.f/MAIN	423, 450
c2c_pot	c2.f/c2c_pot	107D
	mr2cf.f/MAIN	424
cbessh1	cbessl.f/cbessh1	483D
	fluids.f/ofl_pot	74
cbessh2	cbessl.f/cbessh2	497D
cbessi	" /cbessi	141D
	" /cbessk	273
	" /d1cbessi	460, 460
cbessj	c1.f/c1a_pot	74, 82, 89
	" /c1b_pot	160, 169, 176
	c2.f/c2b_pot	78, 86, 93
	" /c2c_pot	170, 179, 186
	cbessl.f/cbessh1	489
	" /cbessh2	503
	" /cbessj	58D
	" /cbessy	201
	" /d1cbessj	330, 330
	" /d2cbessj	346, 347, 347

Subprograms :

Symbol	File/Subprogram	Line
	fluids.f/ifi_pot	43
	rf.f/rod_pot	44, 50, 54
cbessk	cbessl.f/cbessk	244D
	“ /dlcbessk	300, 300
	fluids.f/ofl_pot	80
cbessy	c1.f/c1a_pot	77, 85, 92
	“ /c1b_pot	163, 172, 179
	c2.f/c2b_pot	81, 89, 96
	“ /c2c_pot	173, 182, 189
	cbessl.f/cbessh1	489
	“ /cbessh2	503
	“ /cbessy	167D
	“ /dlcbessy	367, 371, 371, 372, 376, 376, 377, 378, 384, 384, 388, 388, 389
	“ /d2cbessy	413, 413, 414, 418, 418, 419, 420, 425, 425, 426, 427, 428, 429, 435, 435, 436, 440, 440, 441, 441
cdabs	“ /cbessj	72
	“ /cbessy	180
	mr2cf.f/MAIN	472
cdcos	cbessl.f/cbessj	126
	“ /cbessy	233
cdlog	cbessl.f/cbessk	273
	“ /cbessy	201
cdsin	“ /cbessj	127
	“ /cbessy	232
cdsqr	“ /cbessj	126
	“ /cbessy	232
dlcbessh1	“ /dlcbessh1	511D
	fluids.f/ofl_pot	73
dlcbessh2	cbessl.f/dlcbessh2	526D
dlcbessi	“ /dlcbessi	453D
dlcbessj	c1.f/c1a_pot	73, 81, 88
	“ /c1b_pot	159, 168, 175
	c2.f/c2b_pot	77, 85, 92
	“ /c2c_pot	169, 178, 185
	cbessl.f/dlcbessh1	517
	“ /dlcbessh2	532
	“ /dlcbessj	323D
	fluids.f/ifi_pot	42
	rf.f/rod_pot	43, 49, 53

Subprograms :

Symbol	File/Subprogram	Line
-----	-----	----
d1cbessk	cbessl.f/d1cbessk	293D
	fluids.f/ofl_pot	79
d1cbessy	c1.f/c1a_pot	76, 84, 91
	" /c1b_pot	162, 171, 178
	c2.f/c2b_pot	80, 88, 95
	" /c2c_pot	172, 181, 188
	cbessl.f/d1cbessh1	517
	" /d1cbessh2	532
	" /d1cbessy	356D
d2cbessi	" /d2cbessi	468D
d2cbessj	c1.f/c1a_pot	72, 80, 87
	" /c1b_pot	158, 167, 174
	c2.f/c2b_pot	76, 84, 91
	" /c2c_pot	168, 177, 184
	cbessl.f/d2cbessj	338D
	rf.f/rod_pot	42, 48, 52
d2cbessk	cbessl.f/d2cbessk	308D
d2cbessy	c1.f/c1a_pot	75, 83, 90
	" /c1b_pot	161, 170, 177
	c2.f/c2b_pot	79, 87, 94
	" /c2c_pot	171, 180, 187
	cbessl.f/d2cbessy	401D
dabs	c1.f/output_rc1	868, 891, 914, 940, 958, 984, 1002
	c2.f/output_rc2	1266, 1289, 1312, 1338, 1356, 1382,
		1400
	fluids.f/ofl_pot	70, 70
	rf.f/output	358, 370, 382, 396, 405, 419, 428
dcmplx	fluids.f/ifl_pot	39
	mr2cf.f/MAIN	189, 209, 232
dsqrt	fluids.f/ofl_pot	72, 78
fac	cbessl.f/cbessi	155, 155
	" /cbessj	88, 88
	" /cbessk	267, 267, 279, 279
	" /cbessy	195, 195, 207, 207
	" /fac	21D
gamma	" /gamma	4D
iabs	" /cbessi	149
	" /cbessj	71
	" /cbessk	255
	" /cbessy	179
	" /psi	43

Subprograms :

Symbol	File/Subprogram	Line
ifl_pot	fluids.f/ifl_pot	14D
	mr2cf.f/MAIN	313, 321, 425, 440
log10	" / "	472
minv	c1.f/abc_rc1_invert	610
	c2.f/abc_rc2_invert	931
	rf.f/abc_rod_invert	237
	" / minv	448D
	smc1.f/abc_c1_invert	357
ofl_pot	fluids.f/ofl_pot	52D
	mr2cf.f/MAIN	312, 332, 354, 366, 385, 403, 430, 456
output	" / "	361, 393
	rf.f/output	264D
output_if	fluids.f/output_if	91D
	mr2cf.f/MAIN	322, 441
output_of	fluids.f/output_of	134D
	mr2cf.f/MAIN	338, 368, 406, 460
output_rc1	c1.f/output_rc1	649D
	mr2cf.f/MAIN	327, 398, 446
output_rc2	c2.f/output_rc2	991D
	mr2cf.f/MAIN	451
psi	cbessl.f/cbessk	279, 279
	" / cbessy	207, 207
	" / psi	38D
rod_pot	mr2cf.f/MAIN	353, 360, 384, 392
	rf.f/rod_pot	14D
sys_matrix_c1		
	mr2cf.f/MAIN	314
	smc1.f/sys_matrix_c1	14D
sys_matrix_rc1		
	c1.f/sys_matrix_rc1	191D
	mr2cf.f/MAIN	386
sys_matrix_rc2		
	c2.f/sys_matrix_rc2	198D
	mr2cf.f/MAIN	431
sys_matrix_rod		
	" / "	355
	rf.f/sys_matrix_rod	64D

Subprograms :

Symbol	File/Subprogram	Line
-----	-----	----
zabs	c1.f/output_rc1	868, 891, 914, 940, 958, 984, 1002
	c2.f/output_rc2	1266, 1289, 1312, 1338, 1356, 1382,
		1400
	rf.f/minv	483, 483, 495
	" /output	358, 370, 382, 396, 405, 419, 428
zsqr	c1.f/c1a_pot	71, 79
	" /c1b_pot	157, 166
	c2.f/c2b_pot	75, 83
	" /c2c_pot	167, 176
	fluids.f/ifl_pot	40
	mr2cf.f/MAIN	192, 193, 212, 213, 235, 236, 284
	rf.f/rod_pot	41, 47

VARIABLES

Variables :

Symbol	File/Subprogram	Line
a	c2.f/abc_rc2_invert	800D/*, 806D
	“ /sys_matrix_rc2	325D, 328=, 670, 673, 677, 680, 709, 711, 714, 717, 738, 738, 741, 741, 744, 747, 751, 754, 779, 781
	cbessl.f/cbessh1	483D/*, 485D, 489, 489
	“ /cbessh2	497D/*, 500D, 503, 503
	“ /cbessi	141D/*, 145D, 148
	“ /cbessj	58D/*, 62D, 70
	“ /cbessk	244D/*, 248D, 253, 273
	“ /cbessy	167D/*, 171D, 177, 201
	“ /dlcbessh1	511D/*, 514D, 517, 518
	“ /dlcbessh2	526D/*, 529D, 532, 533
	“ /dlcbessi	453D/*, 456D, 460, 460, 460
	“ /dlcbessj	323D/*, 326D, 330, 330, 330
	“ /dlcbessk	293D/*, 296D, 300, 300, 300
	“ /dlcbessy	356D/*, 359D, 367, 367, 371, 371, 371, 372, 372, 376, 376, 376, 377, 377, 378, 378, 379, 384, 384, 384, 388, 388, 388, 389, 389
	“ /d2cbessi	468D/*, 471D
	“ /d2cbessj	338D/*, 341D, 346, 346, 347, 347
	“ /d2cbessk	308D/*, 311D
	“ /d2cbessy	401D/*, 404D, 413, 413, 413, 414, 414, 414, 418, 418, 418, 419, 419, 420, 420, 421, 421, 425, 425, 425, 426, 426, 427, 427, 428, 428, 429, 429, 429, 430, 430, 435, 435, 435, 435, 436, 440, 440, 440, 441, 441, 441, 441
	mr2cf.f/MAIN	8D
	smc1.f/abc_c1_invert	279D/*, 285D
	“ /sys_matrix_c1	87D, 90=, 178, 181, 184, 187, 205, 207, 225, 225, 228, 228, 231, 234, 238, 241, 256, 258
a1	cbessl.f/cbessj	66D, 104=, 106, 113

Variables :

Symbol	File/Subprogram	Line
a1_c1	c1.f/abc_rc1_invert	560D, 575D, 618=, 629=
	“/c1a_pot	42D, 57D
	“/c1b_pot	128D, 143D
	“/output_rc1	675D, 690D, 751, 777, 799, 817, 836, 855, 877, 900, 928, 944, 970, 988
	“/sys_matrix_rc1	219D, 234D
	c2.f/abc_rc2_invert	827D, 842D, 945=, 966=
	“/output_rc2	1017D, 1032D
	“/sys_matrix_rc2	225D, 241D
	mr2cf.f/MAIN	108D, 123D
	smc1.f/abc_c1_invert	306D, 321D, 367=, 373, 381=, 387
	“/sys_matrix_c1	42D, 57D
a1_c2	c2.f/abc_rc2_invert	866D, 882D, 939=, 952, 960=, 973
	“/c2b_pot	45D, 60D
	“/c2c_pot	136D, 152D
	“/output_rc2	1056D, 1072D, 1150, 1176, 1198, 1216, 1235, 1254, 1275, 1298, 1326, 1342, 1368, 1386
	“/sys_matrix_rc2	267D, 283D
	mr2cf.f/MAIN	149D, 164D
a1_rod	c1.f/abc_rc1_invert	583D, 589D, 624=, 635=
	“/output_rc1	698D, 704D
	“/sys_matrix_rc1	242D, 248D
	c2.f/abc_rc2_invert	892D, 898D
	“/output_rc2	1081D, 1087D
	“/sys_matrix_rc2	293D, 299D
	mr2cf.f/MAIN	70D, 74D
	rf.f/abc_rod_invert	211D, 217D, 245=, 250=
	“/abc_rod_solve	161D, 167D, 177=, 181=, 191, 191=
	“/output	274D, 280D, 312, 315, 325, 329, 336, 342, 348, 354, 366, 377, 392, 400, 415, 423
	“/rod_pot	25D, 30D
	“/sys_matrix_rod	74D, 80D
a1c1	c1.f/output_rc1	721D, 751=, 769, 777=, 791, 799=, 809, 817=, 829, 836=, 848, 855=, 865, 877=, 889, 900=, 912, 928=, 938, 944=, 956, 970=, 982, 988=, 1000

Variables :

Symbol	File/Subprogram	Line
a1c2	c2.f/output_rc2	1116D, 1150=, 1168, 1176=, 1190, 1198=, 1208, 1216=, 1228, 1235=, 1247, 1254=, 1264, 1275=, 1287, 1298=, 1310, 1326=, 1336, 1342=, 1354, 1368=, 1380, 1386=, 1398
a2	c1.f/sys_matrix_rc1	263D, 266=, 377, 380, 382, 385, 393, 412, 415, 422, 436, 439, 447, 450
	c2.f/sys_matrix_rc2	325D, 331=, 671, 674, 676, 679, 713, 714, 716, 717, 745, 748
	cbessl.f/cbessj	66D, 106=, 108, 113
	smc1.f/sys_matrix_c1	87D, 92=, 179, 182, 184, 187, 209, 212, 232, 235
a2_c1	c1.f/abc_rc1_invert	560D, 575D, 619=, 630=
	"/c1a_pot	42D, 57D
	"/c1b_pot	128D, 143D
	"/output_rc1	675D, 690D, 755, 781, 801, 819, 838, 857, 879, 902, 930, 946, 972, 990
	"/sys_matrix_rc1	219D, 234D
	c2.f/abc_rc2_invert	827D, 842D, 946=, 967=
	"/output_rc2	1017D, 1032D
	"/sys_matrix_rc2	225D, 241D
	mr2cf.f/MAIN	108D, 123D
	smc1.f/abc_c1_invert	306D, 321D, 368=, 374, 382=, 388
	"/sys_matrix_c1	42D, 57D
a2_c2	c2.f/abc_rc2_invert	866D, 882D, 940=, 953, 961=, 974
	"/c2b_pot	45D, 60D
	"/c2c_pot	136D, 152D
	"/output_rc2	1056D, 1072D, 1154, 1180, 1200, 1218, 1237, 1256, 1277, 1300, 1328, 1344, 1370, 1388
	"/sys_matrix_rc2	267D, 283D
	mr2cf.f/MAIN	149D, 164D
a2c1	c1.f/output_rc1	721D, 755=, 769, 781=, 791, 801=, 809, 819=, 829, 838=, 848, 857=, 865, 879=, 889, 902=, 912, 930=, 938, 946=, 956, 972=, 982, 990=, 1000
a2c2	c2.f/output_rc2	1116D, 1154=, 1168, 1180=, 1190, 1200=, 1208, 1218=, 1228, 1237=, 1247, 1256=, 1264, 1277=, 1287, 1300=, 1310, 1328=, 1336, 1344=, 1354, 1370=, 1380, 1388=, 1398

Variables :

Symbol	File/Subprogram	Line
a3	cbessl.f/cbessj	66D, 108=, 110, 113
a4	" / "	66D, 110=, 113
am	" / "	62D, 113=, 126
	" /cbessy	172D, 225=, 232
ao_1cyl	c1.f/sys_matrix_rc1	191D/*, 197D, 376, 379, 382, 385, 413, 416, 429, 429, 432, 432, 436, 439, 442, 445, 463, 465, 481, 483, 512, 515
	c2.f/sys_matrix_rc2	198D/*, 204D, 328, 331
	mr2cf.f/MAIN	25D, 202=, 214, 299, 310=, 313=, 314=, 316=, 318, 324, 382=, 386=, 421=, 425=, 431=, 433=, 437, 443
	smc1.f/sys_matrix_c1	14D/*, 20D, 90, 92, 210, 213
ao_2cyl	mr2cf.f/MAIN	26D, 225=, 237, 423=
ao_rod	c1.f/sys_matrix_rc1	191D/*, 197D, 266, 392, 395, 396, 408, 410, 420, 423, 447, 449, 453, 473, 493, 522
	mr2cf.f/MAIN	20D, 266=, 284, 353=, 354=, 355=, 358, 364, 384=, 386=, 390, 395
	rf.f/sys_matrix_rod	64D/*, 69D, 97, 103, 107, 113, 116, 116, 118, 121, 125, 125, 139
arg	fluids.f/ifi_pot	37D, 39=, 40
b	c1.f/abc_rc1_invert	533D/*, 539D
	c2.f/sys_matrix_rc2	325D, 329=, 451, 454, 457, 460, 467, 470, 474, 477, 493, 495, 498, 501, 507, 509, 512, 512, 515, 515, 523, 523, 526, 526, 529, 532, 536, 539, 541, 541, 544, 544, 547, 550, 554, 554, 557, 557, 569, 571, 581, 583, 595, 597, 607, 609, 634, 637, 647, 649
	smc1.f/abc_c1_invert	279D/*, 285D, 375, 376, 389, 390
	" /sys_matrix_c1	87D, 91=, 111, 115
b1	cbessl.f/cbessj	66D, 115=, 117, 124

Variables :

Symbol	File/Subprogram	Line
b1_c1	c1.f/abc_rc1_invert	560D, 575D, 620=, 631=
	"/c1a_pot	42D, 57D
	"/c1b_pot	128D, 143D
	"/output_rc1	675D, 690D, 759, 821, 840, 881, 904, 948, 974, 992
	"/sys_matrix_rc1	219D, 234D
	c2.f/abc_rc2_invert	827D, 842D, 947=, 968=
	"/output_rc2	1017D, 1032D
	"/sys_matrix_rc2	225D, 241D
	mr2cf.f/MAIN	108D, 123D
	smc1.f/abc_c1_invert	306D, 321D, 369=, 375, 383=, 389
	"/sys_matrix_c1	42D, 57D
b1_c2	c2.f/abc_rc2_invert	866D, 882D, 941=, 954, 962=, 975
	"/c2b_pot	45D, 60D
	"/c2c_pot	136D, 152D
	"/output_rc2	1056D, 1072D, 1158, 1220, 1239, 1279, 1302, 1346, 1372, 1390
	"/sys_matrix_rc2	267D, 283D
	mr2cf.f/MAIN	149D, 164D
b1_rod	c1.f/abc_rc1_invert	583D, 589D, 625=, 636=
	"/output_rc1	698D, 704D
	"/sys_matrix_rc1	242D, 248D
	c2.f/abc_rc2_invert	892D, 898D
	"/output_rc2	1081D, 1087D
	"/sys_matrix_rc2	293D, 299D
	mr2cf.f/MAIN	70D, 74D
	rf.f/abc_rod_invert	211D, 217D, 246=, 251=
	"/abc_rod_solve	161D, 167D, 178=, 182=, 192, 192=
	"/output	274D, 280D, 315, 316, 342, 348, 367, 378, 401, 416, 424
	"/rod_pot	25D, 30D
	"/sys_matrix_rod	74D, 80D
b1c1	c1.f/output_rc1	721D, 759=, 769, 821=, 829, 840=, 848, 881=, 889, 904=, 912, 948=, 956, 974=, 982, 992=, 1000
b1c2	c2.f/output_rc2	1116D, 1158=, 1168, 1220=, 1228, 1239=, 1247, 1279=, 1287, 1302=, 1310, 1346=, 1354, 1372=, 1380, 1390=, 1398
b2	c1.f/sys_matrix_rc1	263D, 267=, 276, 279, 281, 284, 309, 310, 312, 313, 332, 335

Variables :

Symbol	File/Subprogram	Line
	c2.f/sys_matrix_rc2	325D, 332=, 452, 455, 457, 460, 468, 471, 473, 476, 497, 500, 511, 511, 514, 514, 530, 533, 548, 551
	cbessl.f/cbessj	66D, 117=, 119, 124
	rf.f/sys_matrix_rod	94D, 97=, 104, 106, 115, 115, 118, 122
	smc1.f/sys_matrix_c1	87D, 93=, 102, 106, 109, 113, 137, 138, 140, 141, 157, 160
b2_c1	c1.f/abc_rc1_invert	560D, 575D, 621=, 632=
	“/c1a_pot	42D, 57D
	“/c1b_pot	128D, 143D
	“/output_rc1	675D, 690D, 762, 823, 842, 883, 906, 950, 976, 994
	“/sys_matrix_rc1	219D, 234D
	c2.f/abc_rc2_invert	827D, 842D, 948=, 969=
	“/output_rc2	1017D, 1032D
	“/sys_matrix_rc2	225D, 241D
	mr2cf.f/MAIN	108D, 123D
	smc1.f/abc_c1_invert	306D, 321D, 370=, 376, 384=, 390
	“/sys_matrix_c1	42D, 57D
b2_c2	c2.f/abc_rc2_invert	866D, 882D, 942=, 955, 963=, 976
	“/c2b_pot	45D, 60D
	“/c2c_pot	136D, 152D
	“/output_rc2	1056D, 1072D, 1161, 1222, 1241, 1281, 1304, 1348, 1374, 1392
	“/sys_matrix_rc2	267D, 283D
	mr2cf.f/MAIN	149D, 164D
b2c1	c1.f/output_rc1	721D, 762=, 769, 823=, 829, 842=, 848, 883=, 889, 906=, 912, 950=, 956, 976=, 982, 994=, 1000
b2c2	c2.f/output_rc2	1116D, 1161=, 1168, 1222=, 1228, 1241=, 1247, 1281=, 1287, 1304=, 1310, 1348=, 1354, 1374=, 1380, 1392=, 1398
b3	cbessl.f/cbessj	66D, 119=, 121, 124
b4	“ / “	66D, 121=, 124
bb	mr2cf.f/MAIN	9D
bm	cbessl.f/cbessj	62D, 124=, 127
	“/cbessy	172D, 229=, 233
bmag	rf.f/minv	461D, 495=, 499

Variables :

Symbol	File/Subprogram	Line
bo_c1	c1.f/sys_matrix_rc1	191D/*, 197D, 267, 275, 278, 281, 284, 305, 307, 310, 313, 325, 325, 328, 328, 331, 334, 338, 341, 358, 360
	c2.f/sys_matrix_rc2	198D/*, 204D, 329, 332
	mr2cf.f/MAIN	47D, 214=, 299, 311=, 312=, 314=, 316=, 324, 383=, 385=, 386=, 388=, 395, 401, 422=, 431=, 443, 448
	smc1.f/sys_matrix_c1	14D/*, 20D, 91, 93, 101, 105, 109, 113, 133, 135, 138, 141, 150, 150, 153, 153, 156, 159, 163, 166
c	c2.f/abc_rc2_invert	800D/*, 806D, 954, 955, 975, 976
	"/sys_matrix_rc2	325D, 330=, 343, 347, 351, 353, 355, 357, 387, 389, 392, 395, 415, 415, 418, 418, 421, 424, 428, 431
	rf.f/minv	448D/*, 456D, 465
c1	"/	460D, 502=, 503, 513=, 515, 521=, 523
c1_c1	c1.f/abc_rc1_invert	560D, 575D, 622=, 633=
	"/c1a_pot	42D, 57D
	"/c1b_pot	128D, 143D
	"/output_rc1	675D, 690D, 765, 785, 803, 825, 844, 859, 885, 908, 932, 952, 978, 996
	"/sys_matrix_rc1	219D, 234D
	c2.f/abc_rc2_invert	827D, 842D, 949=, 970=
	"/output_rc2	1017D, 1032D
	"/sys_matrix_rc2	225D, 241D
	mr2cf.f/MAIN	108D, 123D
	smc1.f/abc_c1_invert	306D, 321D, 371=, 377, 385=, 391
	"/sys_matrix_c1	42D, 57D
c1_c2	c2.f/abc_rc2_invert	866D, 882D, 943=, 956, 964=, 977
	c2.f/c2b_pot	45D, 60D
	"/c2c_pot	136D, 152D
	"/output_rc2	1056D, 1072D, 1164, 1184, 1202, 1224, 1243, 1258, 1283, 1306, 1330, 1350, 1376, 1394
	"/sys_matrix_rc2	267D, 283D
	mr2cf.f/MAIN	149D, 164D

Variables :

Symbol	File/Subprogram	Line
c1_rod	c1.f/abc_rc1_invert	583D, 589D, 626=, 637=
	“/output_rc1	698D, 704D
	“/sys_matrix_rc1	242D, 248D
c2.f/abc_rc2_invert		892D, 898D
	“/output_rc2	1081D, 1087D
	“/sys_matrix_rc2	293D, 299D
mr2cf.f/MAIN		70D, 74D
rf.f/abc_rod_invert		211D, 217D, 247=, 252=
	“/abc_rod_solve	161D, 167D, 179=, 183=, 193, 193=
	“/output	274D, 280D, 317, 329, 337, 343, 349, 355, 356, 368, 380, 393, 394, 403, 417, 426
	“/rod_pot	25D, 30D
	“/sys_matrix_rod	74D, 80D
clc1	c1.f/output_rc1	721D, 765=, 769, 785=, 791, 803=, 809, 825=, 829, 844=, 848, 859=, 865, 885=, 889, 908=, 912, 932=, 938, 952=, 956, 978=, 982, 996=, 1000
clc2	c2.f/output_rc2	1116D, 1164=, 1168, 1184=, 1190, 1202=, 1208, 1224=, 1228, 1243=, 1247, 1258=, 1264, 1283=, 1287, 1306=, 1310, 1330=, 1336, 1350=, 1354, 1376=, 1380, 1394=, 1398
c2	“/sys_matrix_rc2	325D, 333=, 344, 348, 351, 355, 391, 392, 394, 395, 422, 425
c2_c1	c1.f/abc_rc1_invert	560D, 575D, 623=, 634=
	“/cla_pot	42D, 57D
	“/clb_pot	128D, 143D
	“/output_rc1	675D, 690D, 767, 788, 806, 827, 846, 862, 887, 910, 935, 954, 980, 998
	“/sys_matrix_rc1	219D, 234D
c2.f/abc_rc2_invert		827D, 842D, 950=, 971=
	“/output_rc2	1017D, 1032D
	“/sys_matrix_rc2	225D, 241D
mr2cf.f/MAIN		108D, 123D
smc1.f/abc_c1_invert		306D, 321D, 372=, 378, 386=, 392
	“/sys_matrix_c1	42D, 57D

Variables :

Symbol	File/Subprogram	Line
-----	-----	----
c2_c2	c2.f/abc_rc2_invert	866D, 882D, 944=, 957, 965=, 978
	“ /c2b_pot	45D, 60D
	“ /c2c_pot	136D, 152D
	“ /output_rc2	1056D, 1072D, 1166, 1187, 1205, 1226, 1245, 1261, 1285, 1308, 1333, 1352, 1378, 1396
	“ /sys_matrix_rc2	267D, 283D
	mr2cf.f/MAIN	149D, 164D
c2c1	c1.f/output_rc1	721D, 767=, 769, 788=, 791, 806=, 809, 827=, 829, 846=, 848, 862=, 865, 887=, 889, 910=, 912, 935=, 938, 954=, 956, 980=, 982, 998=, 1000
c2c2	c2.f/output_rc2	1116D, 1166=, 1168, 1187=, 1190, 1205=, 1208, 1226=, 1228, 1245=, 1247, 1261=, 1264, 1285=, 1287, 1308=, 1310, 1333=, 1336, 1352=, 1354, 1378=, 1380, 1396=, 1398
cbessh1	cbessl.f/cbessh1	487D, 489=
	fluids.f/ofl_pot	68D
	mr2cf.f/MAIN	38D
cbessh2	cbessl.f/cbessh2	501D, 503=
	mr2cf.f/MAIN	38D
cbessi	cbessl.f/cbessi	146D, 158=
	“ /cbessk	249D
	“ /d1cbessi	458D
	mr2cf.f/MAIN	38D
cbessj	c1.f/c1a_pot	61D
	“ /c1b_pot	147D
	c2.f/c2b_pot	64D
	“ /c2c_pot	157D
	cbessl.f/cbessh1	487D
	“ /cbessh2	501D
	“ /cbessj	63D, 91=, 94, 94=, 126=
	“ /cbessy	173D
	“ /d1cbessj	328D
	“ /d2cbessj	343D
	fluids.f/ifl_pot	32D
	mr2cf.f/MAIN	38D
	rf.f/rod_pot	33D

Variables :

Symbol	File/Subprogram	Line
cbessk	cbessl.f/cbessk	249D, 285=
	“ /d1cbessk	298D
	fluids.f/ofl_pot	68D
	mr2cf.f/MAIN	38D
cbessy	c1.f/c1a_pot	61D
	“ /c1b_pot	147D
	c2.f/c2b_pot	64D
	“ /c2c_pot	157D
	cbessl.f/cbessh1	487D
	“ /cbessh2	501D
	“ /cbessy	173D, 213=, 217, 217=, 232=
	“ /d1cbessy	361D
	“ /d2cbessy	406D
	mr2cf.f/MAIN	38D
ci	c2.f/sys_matrix_rc2	199D/*, 204D
	fluids.f/ifl_pot	14D/*, 19D, 39
	mr2cf.f/MAIN	23D, 277=, 313=, 315=, 321=, 425=, 432=, 440=
	smc1.f/sys_matrix_c1	15D/*, 20D
cin	rf.f/minv	448D/*, 456D, 530=
cl_c1	c1.f/abc_rc1_invert	559D, 574D
	“ /c1a_pot	40D, 56D, 71
	“ /c1b_pot	126D, 142D, 157
	“ /output_rc1	674D, 689D
	“ /sys_matrix_rc1	217D, 233D
	c2.f/abc_rc2_invert	826D, 841D
	“ /output_rc2	1016D, 1031D
	“ /sys_matrix_rc2	224D, 240D
	mr2cf.f/MAIN	107D, 122D, 212=
	smc1.f/abc_c1_invert	305D, 320D
	“ /sys_matrix_c1	40D, 56D
cl_c2	c2.f/abc_rc2_invert	864D, 881D
	“ /c2b_pot	43D, 59D, 75
	“ /c2c_pot	134D, 151D, 167
	“ /output_rc2	1054D, 1071D
	“ /sys_matrix_rc2	265D, 282D
	mr2cf.f/MAIN	147D, 163D, 235=

Variables :

Symbol	File/Subprogram	Line
cl_rod	" / "	34D, 192=, 284, 353=, 360=, 384=, 392=
	rf.f/rod_pot	14D/*, 21D, 41
co	c1.f/sys_matrix_rc1	192D/*, 197D
	c2.f/sys_matrix_rc2	199D/*, 204D
	fluids.f/ofl_pot	52D/*, 55D, 70, 72, 78
	mr2cf.f/MAIN	21D, 274=, 312=, 315=, 332=, 354=, 355=, 366=, 385=, 386=, 403=, 430=, 431=, 456=
	rf.f/sys_matrix_rod	64D/*, 69D
	smc1.f/sys_matrix_c1	15D/*, 20D
co_c2	c2.f/sys_matrix_rc2	198D/*, 204D, 330, 333
	mr2cf.f/MAIN	53D, 237=, 299, 330, 424=, 430=, 431=, 433=, 448, 454
ct_c1	c1.f/abc_rc1_invert	559D, 574D
	" /c1a_pot	40D, 56D, 79
	" /c1b_pot	126D, 142D, 166
	" /output_rc1	674D, 689D
	" /sys_matrix_rc1	217D, 233D
	c2.f/abc_rc2_invert	826D, 841D
	" /output_rc2	1016D, 1031D
	" /sys_matrix_rc2	224D, 240D
	mr2cf.f/MAIN	107D, 122D, 213=
	smc1.f/abc_c1_invert	305D, 320D
	" /sys_matrix_c1	40D, 56D
ct_c2	c2.f/abc_rc2_invert	864D, 881D
	" /c2b_pot	43D, 59D, 83
	" /c2c_pot	134D, 151D, 176
	" /output_rc2	1054D, 1071D
	" /sys_matrix_rc2	265D, 282D
	mr2cf.f/MAIN	147D, 163D, 236=
ct_rod	" / "	34D, 193=, 353=, 360=, 384=, 392=
	rf.f/rod_pot	14D/*, 21D, 47

Variables :

Symbol	File/Subprogram	Line
cylinder1	c1.f/abc_rc1_invert	562D
	"/c1a_pot	44D
	"/c1b_pot	130D
	"/output_rc1	677D
	"/sys_matrix_rc1	221D
	c2.f/abc_rc2_invert	829D
	"/output_rc2	1019D
	"/sys_matrix_rc2	228D
	mr2cf.f/MAIN	110D
	smc1.f/abc_c1_invert	308D
	"/sys_matrix_c1	44D
cylinder2	c2.f/abc_rc2_invert	869D
	"/c2b_pot	47D
	"/c2c_pot	139D
	"/output_rc2	1059D
	"/sys_matrix_rc2	270D
	mr2cf.f/MAIN	151D
d1_ifsc	c2.f/abc_rc2_invert	915D, 917D
	"/output_rc2	1104D, 1106D
	"/sys_matrix_rc2	315D, 317D, 787
	fluids.f/ifl_pot	25D, 27D, 42=
	"/output_if	102D, 104D, 123
	mr2cf.f/MAIN	172D, 174D, 427
	smc1.f/abc_c1_invert	341D, 343D
	"/sys_matrix_c1	77D, 79D, 264
d1_ofsc	c1.f/abc_rc1_invert	595D, 597D
	"/output_rc1	710D, 712D
	"/sys_matrix_rc1	254D, 256D, 372
	c2.f/abc_rc2_invert	904D, 906D, 957, 978
	"/output_rc2	1093D, 1095D
	"/sys_matrix_rc2	305D, 307D, 345, 349, 353, 357, 360, 363
	fluids.f/off_pot	61D, 63D, 73=, 79=
	"/output_of	145D, 147D, 165
	mr2cf.f/MAIN	82D, 84D, 334, 367, 404, 435, 458
	rf.f/abc_rod_invert	223D, 225D
	"/sys_matrix_rod	86D, 88D, 143
	smc1.f/abc_c1_invert	330D, 332D, 378, 392
	"/sys_matrix_c1	66D, 68D, 103, 107, 111, 115, 118, 121

Variables :

Symbol	File/Subprogram	Line
d1_sp_cyl_a1	c1.f/abc_rc1_invert	544D, 562D
	c1.f/c1a_pot	25D, 44D, 73=
	" /c1b_pot	111D, 130D
	" /output_rc1	659D, 677D, 752, 777, 836, 877, 970
	" /sys_matrix_rc1	202D, 221D, 377, 404, 430, 459
c2.f/abc_rc2_invert		811D, 829D
	" /output_rc2	1001D, 1019D
	" /sys_matrix_rc2	209D, 228D, 671, 705, 739, 775
mr2cf.f/MAIN		92D, 110D
smc1.f/abc_c1_invert		290D, 308D
	" /sys_matrix_c1	25D, 44D, 179, 201, 226, 252
d1_sp_cyl_a2	c1.f/abc_rc1_invert	545D, 563D
	" /c1a_pot	26D, 45D, 76=
	" /c1b_pot	112D, 131D
	" /output_rc1	660D, 678D, 756, 781, 838, 879, 972
	" /sys_matrix_rc1	203D, 222D, 380, 406, 433, 461
c2.f/abc_rc2_invert		812D, 830D
	" /output_rc2	1002D, 1020D
	" /sys_matrix_rc2	210D, 229D, 674, 707, 742, 777
mr2cf.f/MAIN		93D, 111D
smc1.f/abc_c1_invert		291D, 309D
	" /sys_matrix_c1	26D, 45D, 182, 203, 229, 254
d1_sp_cyl_b1	c1.f/abc_rc1_invert	546D, 564D
	" /c1a_pot	27D, 46D
	" /c1b_pot	113D, 132D, 159=
	" /output_rc1	661D, 679D
	" /sys_matrix_rc1	204D, 223D, 276, 301, 326, 354
c2.f/abc_rc2_invert		813D, 831D
	" /output_rc2	1003D, 1021D
	" /sys_matrix_rc2	211D, 230D, 468, 503, 542, 577
mr2cf.f/MAIN		94D, 112D
smc1.f/abc_c1_invert		292D, 310D, 373, 387
	" /sys_matrix_c1	27D, 46D, 102, 103, 129, 151

Variables :

Symbol	File/Subprogram	Line
d1_sp_cy1_b2	c1.f/abc_rc1_invert	547D, 565D
	"/c1a_pot	28D, 47D
	"/c1b_pot	114D, 133D, 162=
	"/output_rc1	662D, 680D
	"/sys_matrix_rc1	205D, 224D, 279, 303, 329, 356
c2.f/abc_rc2_invert	814D, 832D	
	"/output_rc2	1004D, 1022D
	"/sys_matrix_rc2	212D, 231D, 471, 505, 545, 579
mr2cf.f/MAIN	95D, 113D	
smc1.f/abc_c1_invert	293D, 311D, 374, 388	
	"/sys_matrix_c1	28D, 47D, 106, 107, 131, 154
d1_sp_cy2_b1	c2.f/abc_rc2_invert	849D, 869D
	"/c2b_pot	28D, 47D, 77=
	"/c2c_pot	119D, 139D
	"/output_rc2	1039D, 1059D, 1151, 1176, 1235, 1275, 1368
	"/sys_matrix_rc2	250D, 270D, 452, 489, 524, 565
mr2cf.f/MAIN	132D, 151D	
d1_sp_cy2_b2	c2.f/abc_rc2_invert	850D, 870D
	"/c2b_pot	29D, 48D, 80=
	"/c2c_pot	120D, 140D
	"/output_rc2	1040D, 1060D, 1155, 1180, 1237, 1277, 1370
	"/sys_matrix_rc2	251D, 271D, 455, 491, 527, 567
mr2cf.f/MAIN	133D, 152D	
d1_sp_cy2_c1	c2.f/abc_rc2_invert	851D, 871D, 952, 973
	"/c2b_pot	30D, 49D
	"/c2c_pot	121D, 141D, 169=
	"/output_rc2	1041D, 1061D
	"/sys_matrix_rc2	252D, 272D, 344, 345, 383, 416
mr2cf.f/MAIN	134D, 153D	
d1_sp_cy2_c2	c2.f/abc_rc2_invert	852D, 872D, 953, 974
	c2.f/c2b_pot	31D, 50D
	"/c2c_pot	122D, 142D, 172=
	"/output_rc2	1042D, 1062D
	"/sys_matrix_rc2	253D, 273D, 348, 349, 385, 419
mr2cf.f/MAIN	135D, 154D	

Variables :

Symbol	File/Subprogram	Line
d1_sp_rod	c1.f/abc_rc1_invert	581D, 586D
	"/output_rc1	696D, 701D
	"/sys_matrix_rc1	240D, 245D, 393, 418, 447, 471
	c2.f/abc_rc2_invert	890D, 895D
	"/output_rc2	1079D, 1084D
	"/sys_matrix_rc2	291D, 296D
	mr2cf.f/MAIN	67D, 72D
	rf.f/abc_rod_invert	209D, 214D
	"/abc_rod_solve	159D, 164D
	"/output	272D, 277D, 312, 325, 348, 366, 415
	"/rod_pot	23D, 27D, 43=
	"/sys_matrix_rod	72D, 77D, 103, 111, 119, 137
d1_vrtp_cyl_a1	c1.f/abc_rc1_invert	554D, 570D
	"/cla_pot	35D, 52D, 88=
	"/clb_pot	121D, 138D
	"/output_rc1	669D, 685D, 765, 785, 803, 859, 908, 932, 952, 996
	"/sys_matrix_rc1	212D, 229D, 388, 413, 441, 511
	c2.f/abc_rc2_invert	821D, 837D
	"/output_rc2	1011D, 1027D
	"/sys_matrix_rc2	219D, 236D, 682, 714, 750
	mr2cf.f/MAIN	102D, 118D
	smc1.f/abc_c1_invert	300D, 316D
	"/sys_matrix_c1	35D, 52D, 190, 210, 237
d1_vrtp_cyl_a2	c1.f/abc_rc1_invert	555D, 571D
	"/cla_pot	36D, 53D, 91=
	"/clb_pot	122D, 139D
	"/output_rc1	670D, 686D, 767, 788, 806, 862, 910, 935, 954, 998
	"/sys_matrix_rc1	213D, 230D, 390, 416, 444, 514
	c2.f/abc_rc2_invert	822D, 838D
	"/output_rc2	1012D, 1028D
	"/sys_matrix_rc2	220D, 237D, 684, 717, 753
	mr2cf.f/MAIN	103D, 119D
	smc1.f/abc_c1_invert	301D, 317D
	"/sys_matrix_c1	36D, 53D, 192, 213, 240
d1_vrtp_cyl_b1	c1.f/abc_rc1_invert	556D, 572D
	"/cla_pot	37D, 54D

Variables :

Symbol	File/Subprogram	Line
-----	-----	----
	"/c1b_pot	123D, 140D, 175=
	"/output_rc1	671D, 687D
	"/sys_matrix_rc1	214D, 231D, 287, 310, 337
	c2.f/abc_rc2_invert	823D, 839D
	"/output_rc2	1013D, 1029D
	"/sys_matrix_rc2	221D, 238D, 479, 512, 553, 647
	mr2cf.f/MAIN	104D, 120D
	smc1.f/abc_c1_invert	302D, 318D
	"/sys_matrix_c1	37D, 54D, 117, 138, 162
d1_vrtp_cy1_b2		
	c1.f/abc_rc1_invert	557D, 573D
	"/c1a_pot	38D, 55D
	"/c1b_pot	124D, 141D, 178=
	"/output_rc1	672D, 688D
	"/sys_matrix_rc1	215D, 232D, 289, 313, 340
	c2.f/abc_rc2_invert	824D, 840D
	"/output_rc2	1014D, 1030D
	"/sys_matrix_rc2	222D, 239D, 481, 515, 556, 649
	mr2cf.f/MAIN	105D, 121D
	smc1.f/abc_c1_invert	303D, 319D
	"/sys_matrix_c1	38D, 55D, 120, 141, 165
d1_vrtp_cy2_b1		
	c2.f/abc_rc2_invert	859D, 877D
	"/c2b_pot	38D, 55D, 92=
	"/c2c_pot	129D, 147D
	"/output_rc2	1049D, 1067D, 1164, 1184, 1202, 1258, 1306, 1330, 1350, 1394
	"/sys_matrix_rc2	260D, 278D, 463, 498, 535, 633
	mr2cf.f/MAIN	142D, 159D
d1_vrtp_cy2_b2		
	c2.f/abc_rc2_invert	860D, 878D
	"/c2b_pot	39D, 56D, 95=
	"/c2c_pot	130D, 148D
	"/output_rc2	1050D, 1068D, 1166, 1187, 1205, 1261, 1308, 1333, 1352, 1396
	"/sys_matrix_rc2	261D, 279D, 465, 501, 538, 636
	mr2cf.f/MAIN	143D, 160D

Variables :

Symbol	File/Subprogram	Line
d1_vrtp_cy2_c1		
	c2.f/abc_rc2_invert	861D, 879D
	"/c2b_pot	40D, 57D
	"/c2c_pot	131D, 149D, 185=
	"/output_rc2	1051D, 1069D
	"/sys_matrix_rc2	262D, 280D, 359, 392, 427
	mr2cf.f/MAIN	144D, 161D
d1_vrtp_cy2_c2		
	c2.f/abc_rc2_invert	862D, 880D
	"/c2b_pot	41D, 58D
	"/c2c_pot	132D, 150D, 188=
	"/output_rc2	1052D, 1070D
	"/sys_matrix_rc2	263D, 281D, 362, 395, 430
	mr2cf.f/MAIN	145D, 162D
d1_vrtp_rod	c1.f/abc_rc1_invert	582D, 588D
	"/output_rc1	697D, 703D
	"/sys_matrix_rc1	241D, 247D, 398, 423, 452, 521
	c2.f/abc_rc2_invert	891D, 897D
	"/output_rc2	1080D, 1086D
	"/sys_matrix_rc2	292D, 298D
	mr2cf.f/MAIN	69D, 73D
	rf.f/abc_rod_invert	210D, 216D
	"/abc_rod_solve	160D, 166D
	"/output	273D, 279D, 317, 330, 337, 355, 380, 393, 403, 426
	"/rod_pot	24D, 29D, 53=
	"/sys_matrix_rod	73D, 79D, 109, 116, 124
d1_vxp_cyl_a1		
	c1.f/abc_rc1_invert	549D, 566D
	"/c1a_pot	30D, 48D, 81=
	"/c1b_pot	116D, 134D
	"/output_rc1	664D, 681D, 760, 821, 881, 904, 948, 974, 992
	"/sys_matrix_rc1	207D, 225D, 383, 435, 485
	c2.f/abc_rc2_invert	816D, 833D
	"/output_rc2	1006D, 1023D
	"/sys_matrix_rc2	214D, 232D, 677, 745
	mr2cf.f/MAIN	97D, 114D
	smc1.f/abc_c1_invert	295D, 312D
	"/sys_matrix_c1	30D, 48D, 185, 232

Variables :

Symbol	File/Subprogram	Line

d1_vxp_cyl_a2		
c1.f/abc_rc1_invert		550D, 567D
"/c1a_pot		31D, 49D, 84=
"/c1b_pot		117D, 135D
"/output_rc1		665D, 682D, 763, 823, 883, 906, 950, 976, 994
"/sys_matrix_rc1		208D, 226D, 386, 438, 487
c2.f/abc_rc2_invert		817D, 834D
"/output_rc2		1007D, 1024D
"/sys_matrix_rc2		215D, 233D, 680, 748
mr2cf.f/MAIN		98D, 115D
smc1.f/abc_c1_invert		296D, 313D
"/sys_matrix_c1		31D, 49D, 188, 235
d1_vxp_cyl_b1		
c1.f/abc_rc1_invert		551D, 568D
"/c1a_pot		32D, 50D
"/c1b_pot		118D, 136D, 168=
"/output_rc1		666D, 683D
"/sys_matrix_rc1		209D, 227D, 282, 332
c2.f/abc_rc2_invert		818D, 835D
"/output_rc2		1008D, 1025D
"/sys_matrix_rc2		216D, 234D, 474, 547, 611
mr2cf.f/MAIN		99D, 116D
smc1.f/abc_c1_invert		297D, 314D
"/sys_matrix_c1		32D, 50D, 110, 157
d1_vxp_cyl_b2		
c1.f/abc_rc1_invert		552D, 569D
"/c1a_pot		33D, 51D
"/c1b_pot		119D, 137D, 171=
"/output_rc1		667D, 684D
"/sys_matrix_rc1		210D, 228D, 285, 335
c2.f/abc_rc2_invert		819D, 836D
"/output_rc2		1009D, 1026D
"/sys_matrix_rc2		217D, 235D, 477, 550, 613
mr2cf.f/MAIN		100D, 117D
smc1.f/abc_c1_invert		298D, 315D
"/sys_matrix_c1		33D, 51D, 114, 160

Variables :

Symbol	File/Subprogram	Line
d1_vxp_cy2_b1		
	c2.f/abc_rc2_invert	854D, 873D
	“ /c2b_pot	33D, 51D, 85=
	“ /c2c_pot	124D, 143D
	“ /output_rc2	1044D, 1063D, 1159, 1220, 1279, 1302, 1346, 1372, 1390
	“ /sys_matrix_rc2	255D, 274D, 458, 529, 599
	mr2cf.f/MAIN	137D, 155D
d1_vxp_cy2_b2		
	c2.f/abc_rc2_invert	855D, 874D
	“ /c2b_pot	34D, 52D, 88=
	“ /c2c_pot	125D, 144D
	“ /output_rc2	1045D, 1064D, 1162, 1222, 1281, 1304, 1348, 1374, 1392
	“ /sys_matrix_rc2	256D, 275D, 461, 532, 601
	mr2cf.f/MAIN	138D, 156D
d1_vxp_cy2_c1		
	c2.f/abc_rc2_invert	856D, 875D
	“ /c2b_pot	35D, 53D
	“ /c2c_pot	126D, 145D, 178=
	“ /output_rc2	1046D, 1065D
	“ /sys_matrix_rc2	257D, 276D, 352, 422
	mr2cf.f/MAIN	139D, 157D
d1_vxp_cy2_c2		
	c2.f/abc_rc2_invert	857D, 876D
	“ /c2b_pot	36D, 54D
	“ /c2c_pot	127D, 146D, 181=
	“ /output_rc2	1047D, 1066D
	“ /sys_matrix_rc2	258D, 277D, 356, 425
	mr2cf.f/MAIN	140D, 158D

Variables :

Symbol	File/Subprogram	Line
d1_vxp_rod	c1.f/abc_rc1_invert	582D, 587D
	"/output_rc1	697D, 702D
	c1.f/sys_matrix_rc1	241D, 246D, 395, 449, 495
	c2.f/abc_rc2_invert	891D, 896D
	"/output_rc2	1080D, 1085D
	"/sys_matrix_rc2	292D, 297D
	mr2cf.f/MAIN	68D, 73D
	rf.f/abc_rod_invert	210D, 215D
	"/abc_rod_solve	160D, 165D
	"/output	273D, 278D, 316, 342, 367, 379, 402, 416, 425
	"/rod_pot	24D, 28D, 49=
	"/sys_matrix_rod	73D, 78D, 107, 122
d1cbessh1	cbessl.f/d1cbessh1	515D, 517=
	fluids.f/offl_pot	68D
	mr2cf.f/MAIN	43D
d1cbessh2	cbessl.f/d1cbessh2	530D, 532=
	mr2cf.f/MAIN	43D
d1cbessi	cbessl.f/d1cbessi	458D, 460=
	mr2cf.f/MAIN	42D
d1cbessj	c1.f/c1a_pot	62D
	"/c1b_pot	148D
	c2.f/c2b_pot	65D
	"/c2c_pot	158D
	cbessl.f/d1cbessh1	515D
	"/d1cbessh2	530D
	"/d1cbessj	328D, 330=
	fluids.f/ifl_pot	32D
	mr2cf.f/MAIN	39D
	rf.f/rod_pot	34D
d1cbessk	cbessl.f/d1cbessk	298D, 300=
	fluids.f/offl_pot	68D
	mr2cf.f/MAIN	41D
d1cbessy	c1.f/c1a_pot	63D
	"/c1b_pot	149D
	c2.f/c2b_pot	66D
	"/c2c_pot	159D
	cbessl.f/d1cbessh1	515D
	"/d1cbessh2	530D
	"/d1cbessy	361D, 367=, 372=, 379=, 384=, 389=
	mr2cf.f/MAIN	40D

Variables :

Symbol	File/Subprogram	Line
d2_sp_cyl_a1	c1.f/abc_rc1_invert	544D, 562D
" /cla_pot		25D, 44D, 72=
" /clb_pot		111D, 130D
" /output_rc1		659D, 677D, 751, 777, 900, 944, 988
" /sys_matrix_rc1		202D, 221D, 376
c2.f/abc_rc2_invert		811D, 829D
" /output_rc2		1001D, 1019D
" /sys_matrix_rc2		209D, 228D, 670
mr2cf.f/MAIN		92D, 110D
smc1.f/abc_c1_invert		290D, 308D
" /sys_matrix_c1		25D, 44D, 178
d2_sp_cyl_a2	c1.f/abc_rc1_invert	545D, 563D
" /cla_pot		26D, 45D, 75=
" /clb_pot		112D, 131D
" /output_rc1		660D, 678D, 755, 781, 902, 946, 990
" /sys_matrix_rc1		203D, 222D, 379
c2.f/abc_rc2_invert		812D, 830D
" /output_rc2		1002D, 1020D
" /sys_matrix_rc2		210D, 229D, 673
mr2cf.f/MAIN		93D, 111D
smc1.f/abc_c1_invert		291D, 309D
" /sys_matrix_c1		26D, 45D, 181
d2_sp_cyl_b1	c1.f/abc_rc1_invert	546D, 564D
" /cla_pot		27D, 46D
" /clb_pot		113D, 132D, 158=
c1.f/output_rc1		661D, 679D
" /sys_matrix_rc1		204D, 223D, 275
c2.f/abc_rc2_invert		813D, 831D
" /output_rc2		1003D, 1021D
" /sys_matrix_rc2		211D, 230D, 467
mr2cf.f/MAIN		94D, 112D
smc1.f/abc_c1_invert		292D, 310D
" /sys_matrix_c1		27D, 46D, 101

Variables :

Symbol	File/Subprogram	Line
d2_sp_cy1_b2	c1.f/abc_rc1_invert	547D, 565D
	"/c1a_pot	28D, 47D
	"/c1b_pot	114D, 133D, 161=
	"/output_rc1	662D, 680D
	"/sys_matrix_rc1	205D, 224D, 278
c2.f/abc_rc2_invert		814D, 832D
	"/output_rc2	1004D, 1022D
	"/sys_matrix_rc2	212D, 231D, 470
mr2cf.f/MAIN		95D, 113D
smc1.f/abc_c1_invert		293D, 311D
	"/sys_matrix_c1	28D, 47D, 105
d2_sp_cy2_b1	c2.f/abc_rc2_invert	849D, 869D
	"/c2b_pot	28D, 47D, 76=
	"/c2c_pot	119D, 139D
	"/output_rc2	1039D, 1059D, 1150, 1176, 1298, 1342, 1386
	"/sys_matrix_rc2	250D, 270D, 451
mr2cf.f/MAIN		132D, 151D
d2_sp_cy2_b2	c2.f/abc_rc2_invert	850D, 870D
	"/c2b_pot	29D, 48D, 79=
	"/c2c_pot	120D, 140D
	"/output_rc2	1040D, 1060D, 1154, 1180, 1300, 1344, 1388
	"/sys_matrix_rc2	251D, 271D, 454
mr2cf.f/MAIN		133D, 152D
d2_sp_cy2_c1	c2.f/abc_rc2_invert	851D, 871D
	"/c2b_pot	30D, 49D
	"/c2c_pot	121D, 141D, 168=
	"/output_rc2	1041D, 1061D
	"/sys_matrix_rc2	252D, 272D, 343
mr2cf.f/MAIN		134D, 153D
d2_sp_cy2_c2	c2.f/abc_rc2_invert	852D, 872D
	"/c2b_pot	31D, 50D
	"/c2c_pot	122D, 142D, 171=
	"/output_rc2	1042D, 1062D
	"/sys_matrix_rc2	253D, 273D, 347
mr2cf.f/MAIN		135D, 154D

Variables :

Symbol	File/Subprogram	Line
d2_sp_rod	c1.f/abc_rc1_invert	581D, 586D
	"/output_rc1	696D, 701D
	"/sys_matrix_rc1	240D, 245D, 392
c2.f/abc_rc2_invert		890D, 895D
	"/output_rc2	1079D, 1084D
	"/sys_matrix_rc2	291D, 296D
mr2cf.f/MAIN		67D, 72D
rf.f/abc_rod_invert		209D, 214D
	"/abc_rod_solve	159D, 164D
	"/output	272D, 277D, 312, 315, 325, 377, 400, 423
	"/rod_pot	23D, 27D, 42=
	"/sys_matrix_rod	72D, 77D, 102
d2_vrtp_cyl_a1		
	c1.f/abc_rc1_invert	554D, 570D
	"/cla_pot	35D, 52D, 87=
	"/clb_pot	121D, 138D
	"/output_rc1	669D, 685D
	"/sys_matrix_rc1	212D, 229D, 413
c2.f/abc_rc2_invert		821D, 837D
	"/output_rc2	1011D, 1027D
	"/sys_matrix_rc2	219D, 236D, 714
mr2cf.f/MAIN		102D, 118D
smc1.f/abc_c1_invert		300D, 316D
	"/sys_matrix_c1	35D, 52D, 210
d2_vrtp_cyl_a2		
	c1.f/abc_rc1_invert	555D, 571D
	"/cla_pot	36D, 53D, 90=
	"/clb_pot	122D, 139D
	"/output_rc1	670D, 686D
	"/sys_matrix_rc1	213D, 230D, 416
c2.f/abc_rc2_invert		822D, 838D
	"/output_rc2	1012D, 1028D
	"/sys_matrix_rc2	220D, 237D, 717
mr2cf.f/MAIN		103D, 119D
smc1.f/abc_c1_invert		301D, 317D
	"/sys_matrix_c1	36D, 53D, 213

Variables :

Symbol	File/Subprogram	Line

d2_vrtp_cy1_b1		
c1.f/abc_rc1_invert		556D, 572D
"/c1a_pot		37D, 54D
"/c1b_pot		123D, 140D, 174=
"/output_rc1		671D, 687D
"/sys_matrix_rc1		214D, 231D, 310
c2.f/abc_rc2_invert		823D, 839D
"/output_rc2		1013D, 1029D
"/sys_matrix_rc2		221D, 238D, 512
mr2cf.f/MAIN		104D, 120D
smc1.f/abc_c1_invert		302D, 318D
"/sys_matrix_c1		37D, 54D, 138
d2_vrtp_cy1_b2		
c1.f/abc_rc1_invert		557D, 573D
"/c1a_pot		38D, 55D
"/c1b_pot		124D, 141D, 177=
"/output_rc1		672D, 688D
"/sys_matrix_rc1		215D, 232D, 313
c2.f/abc_rc2_invert		824D, 840D
"/output_rc2		1014D, 1030D
"/sys_matrix_rc2		222D, 239D, 515
mr2cf.f/MAIN		105D, 121D
smc1.f/abc_c1_invert		303D, 319D
"/sys_matrix_c1		38D, 55D, 141
d2_vrtp_cy2_b1		
c2.f/abc_rc2_invert		859D, 877D
"/c2b_pot		38D, 55D, 91=
"/c2c_pot		129D, 147D
"/output_rc2		1049D, 1067D
"/sys_matrix_rc2		260D, 278D, 498
mr2cf.f/MAIN		142D, 159D
d2_vrtp_cy2_b2		
c2.f/abc_rc2_invert		860D, 878D
"/c2b_pot		39D, 56D, 94=
"/c2c_pot		130D, 148D
"/output_rc2		1050D, 1068D
"/sys_matrix_rc2		261D, 279D, 501
mr2cf.f/MAIN		143D, 160D

Variables :

Symbol	File/Subprogram	Line
d2_vrtp_cy2_c1		
	c2.f/abc_rc2_invert	861D, 879D
	"/c2b_pot	40D, 57D
	"/c2c_pot	131D, 149D, 184=
	"/output_rc2	1051D, 1069D
	"/sys_matrix_rc2	262D, 280D, 392
	mr2cf.f/MAIN	144D, 161D
d2_vrtp_cy2_c2		
	c2.f/abc_rc2_invert	862D, 880D
	"/c2b_pot	41D, 58D
	"/c2c_pot	132D, 150D, 187=
	"/output_rc2	1052D, 1070D
	"/sys_matrix_rc2	263D, 281D, 395
	mr2cf.f/MAIN	145D, 162D
d2_vrtp_rod	c1.f/abc_rc1_invert	582D, 588D
	"/output_rc1	697D, 703D
	"/sys_matrix_rc1	241D, 247D, 423
	c2.f/abc_rc2_invert	891D, 897D
	"/output_rc2	1080D, 1086D
	"/sys_matrix_rc2	292D, 298D
	mr2cf.f/MAIN	69D, 73D
	rf.f/abc_rod_invert	210D, 216D
	"/abc_rod_solve	160D, 166D
	"/output	273D, 279D
	"/rod_pot	24D, 29D, 52=
	"/sys_matrix_rod	73D, 79D, 116
d2_vxp_cyl_a1		
	c1.f/abc_rc1_invert	549D, 566D
	"/c1a_pot	30D, 48D, 80=
	"/c1b_pot	116D, 134D
	"/output_rc1	664D, 681D
	"/sys_matrix_rc1	207D, 225D, 435
	c2.f/abc_rc2_invert	816D, 833D
	"/output_rc2	1006D, 1023D
	"/sys_matrix_rc2	214D, 232D, 744
	mr2cf.f/MAIN	97D, 114D
	smc1.f/abc_c1_invert	295D, 312D
	"/sys_matrix_c1	30D, 48D, 231

Variables :

Symbol	File/Subprogram	Line

d2_vxp_cyl_a2		
c1.f/abc_rc1_invert		550D, 567D
"/c1a_pot		31D, 49D, 83=
"/c1b_pot		117D, 135D
"/output_rc1		665D, 682D
"/sys_matrix_rc1		208D, 226D, 438
c2.f/abc_rc2_invert		817D, 834D
"/output_rc2		1007D, 1024D
"/sys_matrix_rc2		215D, 233D, 747
mr2cf.f/MAIN		98D, 115D
smc1.f/abc_c1_invert		296D, 313D
"/sys_matrix_c1		31D, 49D, 234
d2_vxp_cyl_b1		
c1.f/abc_rc1_invert		551D, 568D
"/c1a_pot		32D, 50D
"/c1b_pot		118D, 136D, 167=
"/output_rc1		666D, 683D
"/sys_matrix_rc1		209D, 227D, 331
c2.f/abc_rc2_invert		818D, 835D
"/output_rc2		1008D, 1025D
"/sys_matrix_rc2		216D, 234D, 547
mr2cf.f/MAIN		99D, 116D
smc1.f/abc_c1_invert		297D, 314D
"/sys_matrix_c1		32D, 50D, 156
d2_vxp_cyl_b2		
c1.f/abc_rc1_invert		552D, 569D
"/c1a_pot		33D, 51D
"/c1b_pot		119D, 137D, 170=
"/output_rc1		667D, 684D
"/sys_matrix_rc1		210D, 228D, 334
c2.f/abc_rc2_invert		819D, 836D
"/output_rc2		1009D, 1026D
"/sys_matrix_rc2		217D, 235D, 550
mr2cf.f/MAIN		100D, 117D
smc1.f/abc_c1_invert		298D, 315D
"/sys_matrix_c1		33D, 51D, 159

Variables :

Symbol	File/Subprogram	Line
d2_vxp_cy2_b1		
	c2.f/abc_rc2_invert	854D, 873D
	" /c2b_pot	33D, 51D, 84=
	" /c2c_pot	124D, 143D
	" /output_rc2	1044D, 1063D
	" /sys_matrix_rc2	255D, 274D, 529
	mr2cf.f/MAIN	137D, 155D
d2_vxp_cy2_b2		
	c2.f/abc_rc2_invert	855D, 874D
	" /c2b_pot	34D, 52D, 87=
	" /c2c_pot	125D, 144D
	" /output_rc2	1045D, 1064D
	" /sys_matrix_rc2	256D, 275D, 532
	mr2cf.f/MAIN	138D, 156D
d2_vxp_cy2_c1		
	c2.f/abc_rc2_invert	856D, 875D
	" /c2b_pot	35D, 53D
	" /c2c_pot	126D, 145D, 177=
	" /output_rc2	1046D, 1065D
	" /sys_matrix_rc2	257D, 276D, 421
	mr2cf.f/MAIN	139D, 157D
d2_vxp_cy2_c2		
	c2.f/abc_rc2_invert	857D, 876D
	" /c2b_pot	36D, 54D
	" /c2c_pot	127D, 146D, 180=
	" /output_rc2	1047D, 1066D
	" /sys_matrix_rc2	258D, 277D, 424
	mr2cf.f/MAIN	140D, 158D
d2_vxp_rod	c1.f/abc_rc1_invert	582D, 587D
	" /output_rc1	697D, 702D
	" /sys_matrix_rc1	241D, 246D, 449
	c2.f/abc_rc2_invert	891D, 896D
	" /output_rc2	1080D, 1085D
	" /sys_matrix_rc2	292D, 297D
	mr2cf.f/MAIN	68D, 73D
	rf.f/abc_rod_invert	210D, 215D
	" /abc_rod_solve	160D, 165D
	" /output	273D, 278D
	" /rod_pot	24D, 28D, 48=
	" /sys_matrix_rod	73D, 78D, 121

Variables :

Symbol	File/Subprogram	Line
d2cbessi	cbessl.f/d2cbessi	473D, 475=
	mr2cf.f/MAIN	42D
d2cbessj	c1.f/c1a_pot	62D
	" /c1b_pot	148D
	c2.f/c2b_pot	65D
	" /c2c_pot	158D
	cbessl.f/d2cbessj	343D, 346=
	mr2cf.f/MAIN	39D
	rf.f/rod_pot	34D
d2cbessk	cbessl.f/d2cbessk	313D, 315=
	mr2cf.f/MAIN	41D
d2cbessy	c1.f/c1a_pot	63D
	" /c1b_pot	149D
	c2.f/c2b_pot	66D
	" /c2c_pot	159D
	cbessl.f/d2cbessy	406D, 414=, 421=, 430=, 435=, 441=
	mr2cf.f/MAIN	40D
d_if	c2.f/abc_rc2_invert	915D, 917D, 951=, 972=
	" /output_rc2	1104D, 1106D
	" /sys_matrix_rc2	315D, 317D
	fluids.f/ifl_pot	25D, 27D
	" /output_if	102D, 104D, 116, 123
	mr2cf.f/MAIN	172D, 174D, 427
	smc1.f/abc_c1_invert	341D, 343D, 379=, 393=
	" /sys_matrix_c1	77D, 79D
deta	rf.f/abc_rod_solve	171D, 186=, 191, 192, 193
dill	" /output	286D, 312=, 314, 325=, 328
dpp	c1.f/output_rc1	720D, 962=, 964, 1007=, 1008
	c2.f/output_rc2	1115D, 1360=, 1362, 1405=, 1406
	rf.f/output	286D, 409=, 411, 433=, 434
e_1cyl	mr2cf.f/MAIN	25D, 199=, 209
e_2cyl	" / "	26D, 222=, 232
e_rod	" / "	22D, 182=, 189
ec_c1	" / "	49D, 209=, 210, 211
ec_c2	" / "	55D, 232=, 233, 234
ec_rod	" / "	37D, 189=, 190, 191
emt	" / "	19D, 271=, 294, 307, 351, 380, 419

Variables :

Symbol	File/Subprogram	Line
err	c1.f/output_rc1	720D, 912=, 914, 918, 956=, 958, 959=, 962, 1000=, 1002, 1003=, 1006, 1007
	c2.f/output_rc2	1115D, 1310=, 1312, 1316, 1354=, 1356, 1357=, 1360, 1398=, 1400, 1401=, 1404, 1405
	rf.f/output	286D, 377=, 382, 386, 400=, 405, 406=, 409, 423=, 428, 429=, 432, 433
ett	c1.f/output_rc1	720D, 889=, 891, 895, 982=, 984, 1006
	c2.f/output_rc2	1115D, 1287=, 1289, 1293, 1380=, 1382, 1404
	rf.f/output	286D, 366=, 370, 373, 415=, 419, 432
euler	cbessl.f/psi	40D, 41D, 45, 50
exctype	c1.f/abc_rc1_invert	533D/*, 538D, 617, 628
	c2.f/abc_rc2_invert	800D/*, 805D, 938, 959
	mr2cf.f/MAIN	19D, 270=, 316=, 356=, 388=, 433=
	rf.f/abc_rod_invert	201D/*, 206D, 244, 249
	“ /abc_rod_solve	152D/*, 156D, 176, 180
	smc1.f/abc_c1_invert	279D/*, 284D, 366, 380
exx	c1.f/output_rc1	720D, 865=, 868, 872, 938=, 940, 941=, 962, 962, 1006=, 1007, 1007
	c2.f/output_rc2	1115D, 1264=, 1266, 1270, 1336=, 1338, 1339=, 1360, 1360, 1404=, 1405, 1405
	rf.f/output	286D, 354=, 358, 362, 392=, 396, 397=, 409, 409, 432=, 433, 433
ez	cbessl.f/cbessj	66D, 76=, 115
ez2	“ / “	66D, 77=, 104, 106, 108, 110, 117, 119, 121
f	mr2cf.f/MAIN	10D, 294, 297
f2	fluids.f/ofl_pot	67D, 78=, 79, 80
fa	mr2cf.f/MAIN	10D
fac	cbessl.f/cbessi	143D
	“ /cbessj	60D
	“ /cbessk	246D
	“ /cbessy	169D
	“ /fac	22D, 24=, 30=
	mr2cf.f/MAIN	32D
fn2	cbessl.f/cbessj	65D, 75=, 104, 104, 106, 106, 108, 108, 110, 110, 115, 117, 117, 119, 119, 121, 121

Variables :

Symbol	File/Subprogram	Line
-----	-----	----
g	mr2cf.f/MAIN	22D, 280=, 346=, 376=, 414=, 468=, 472=, 478
g1	fluids.f/ifi_pot	37D, 40=, 42, 43
g2	" /ofi_pot	67D, 72=, 73, 74
gamma	cbessl.f/gamma	5D, 7=, 13=
	mr2cf.f/MAIN	32D
h_1cyl	" / "	25D, 204=, 214
h_2cyl	" / "	26D, 227=, 237
i	cbessl.f/fac	22D, 27=, 28
	" /gamma	5D, 10=, 11
	cbessl.f/psi	39D, 47=, 48
	rf.f/minv	459D, 463=, 465, 465, 469=, 471, 472, 472, 477=, 478, 479, 481, 491, 492, 500, 502, 502, 507, 507=, 510, 512=, 513, 513, 514, 515, 515, 519=, 520, 521, 523, 528=, 530, 530
iflag	c1.f/abc_rc1_invert	603D, 607=, 610=
	c2.f/abc_rc2_invert	924D, 928=, 931=
	mr2cf.f/MAIN	32D, 282=
	rf.f/abc_rod_invert	231D, 235=, 237=
	" /minv	448D/*, 455D, 533=, 552=
	smc1.f/abc_c1_invert	350D, 354=, 357=
ifluid	c2.f/abc_rc2_invert	915D
	" /output_rc2	1104D
	" /sys_matrix_rc2	315D
	fluids.f/ifi_pot	27D
	" /output_if	104D
	mr2cf.f/MAIN	174D
	smc1.f/abc_c1_invert	341D
	" /sys_matrix_c1	77D
ifsc	c2.f/abc_rc2_invert	915D, 917D
	" /output_rc2	1104D, 1106D
	" /sys_matrix_rc2	315D, 317D, 686
	fluids.f/ifi_pot	25D, 27D, 43=
	" /output_if	102D, 104D, 116
	mr2cf.f/MAIN	172D, 174D, 427
	smc1.f/abc_c1_invert	341D, 343D
	" /sys_matrix_c1	77D, 79D, 194
in	mr2cf.f/MAIN	8D
ip	rf.f/minv	459D, 478=, 484=, 488, 490, 491

Variables :

Symbol	File/Subprogram	Line
iptmax	mr2cf.f/MAIN	17D, 18D, 22
irow	rf.f/minv	459D, 481=, 482, 484, 485, 500=, 502, 503, 503, 505
j	cbessl.f/cbessi	143D, 154=, 155
	“ /cbessj	60D, 87=, 88
	“ /cbessk	246D, 266=, 267, 276=, 279
	“ /cbessy	169D, 194=, 195, 204=, 207
	rf.f/minv	459D, 464=, 465, 465, 470=, 471, 472, 472, 476=, 479, 482, 485, 501, 502, 502, 505, 508, 508=, 510, 514=, 515, 515, 520=, 521, 522, 523, 523, 529=, 530, 530
jcol	“ / “	459D, 522=, 523, 523, 523
jk	mr2cf.f/MAIN	19D, 272=, 280, 346, 376, 414, 468, 472, 478
jn	“ / “	8D
jrow	rf.f/minv	459D, 489=, 490, 491, 491, 492, 501=, 502, 503, 503
k	c1.f/c1a_pot	14D/*, 20D, 71, 79
	“ /c1b_pot	100D/*, 106D, 157, 166
	“ /output_rc1	649D/*, 654D, 725, 765, 767, 785, 788, 799, 801, 825, 827, 844, 846, 859, 862, 885, 887, 908, 910, 932, 935, 952, 954, 978, 980, 996, 998
	“ /sys_matrix_rc1	191D/*, 197D, 269, 287, 289, 301, 303, 305, 307, 337, 340, 362, 364, 388, 390, 398, 404, 406, 408, 410, 418, 420, 441, 444, 452, 467, 469, 475, 489, 491, 497, 503, 505, 517
	c2.f/abc_rc2_invert	956, 957, 977, 978
	“ /c2b_pot	17D/*, 23D, 75, 83
	“ /c2c_pot	107D/*, 113D, 167, 176
	“ /output_rc2	991D/*, 996D, 1121, 1164, 1166, 1184, 1187, 1198, 1200, 1224, 1226, 1243, 1245, 1258, 1261, 1283, 1285, 1306, 1308, 1330, 1333, 1350, 1352, 1376, 1378, 1394, 1396

Variables :

Symbol	File/Subprogram	Line
-----	-----	----
	"/sys_matrix_rc2	198D/*, 204D, 335, 359, 360, 362, 363, 383, 385, 387, 389, 427, 430, 463, 465, 479, 481, 489, 491, 493, 495, 503, 505, 507, 509, 535, 538, 553, 556, 573, 575, 585, 587, 603, 605, 615, 617, 625, 627, 639, 641, 682, 684, 705, 707, 709, 711, 750, 753, 783, 785
cbessl.f/cbessi		143D, 153=, 154, 155, 155
	"/cbessj	60D, 86=, 87, 88, 88
	"/cbessk	246D, 265=, 266, 267, 267, 275=, 276, 277, 278, 279, 279
	"/cbessy	169D, 193=, 194, 195, 195, 203=, 204, 205, 206, 207, 207
fluids.f/ifi_pot		14D/*, 19D, 39
	"/ofi_pot	52D/*, 55D, 70, 72, 78
	"/output_if	91D/*, 96D
	"/output_of	134D/*, 139D
mr2cf.f/MAIN		21D, 264=, 284, 294, 310=, 311=, 312=, 313=, 314=, 321=, 322=, 326=, 327=, 332=, 338=, 353=, 354=, 355=, 360=, 361=, 366=, 368=, 382=, 383=, 384=, 385=, 386=, 392=, 393=, 397=, 398=, 403=, 406=, 421=, 422=, 423=, 424=, 425=, 430=, 431=, 440=, 441=, 445=, 446=, 450=, 451=, 456=, 460=, 480
rf.f/minv		459D, 536=
	"/output	264D/*, 269D, 290, 317, 329, 336, 343, 349, 355, 356, 368, 380, 393, 394, 403, 417, 426
	"/rod_pot	14D/*, 20D, 41, 47
	"/sys_matrix_rod	64D/*, 69D, 99, 109, 111, 113, 125, 141
smc1.f/abc_c1_invert		377, 378, 391, 392
	"/sys_matrix_c1	14D/*, 20D, 95, 117, 118, 120, 121, 129, 131, 133, 135, 162, 165, 190, 192, 201, 203, 205, 207, 237, 240, 260, 262

Variables :

Symbol	File/Subprogram	Line
k2	c1.f/output_rc1	719D, 725=, 753, 757, 778, 782, 855, 928
	"/sys_matrix_rc1	263D, 269=, 276, 279, 309, 312, 377, 380, 393, 412, 415, 422
	c2.f/output_rc2	1114D, 1121=, 1152, 1156, 1177, 1181, 1254, 1326
	"/sys_matrix_rc2	325D, 335=, 344, 348, 391, 394, 452, 455, 468, 471, 497, 500, 511, 514, 671, 674, 713, 716
	rf.f/output	285D, 290=, 313, 326, 329, 354, 392
	"/sys_matrix_rod	94D, 99=, 104, 115
	smc1.f/sys_matrix_c1	87D, 95=, 102, 106, 137, 140, 179, 182, 209, 212
k2_sp_cy1_a2	c1.f/output_rc1	857, 930
k2_sp_cy2_b2	c2.f/output_rc2	1256, 1328
kn	mr2cf.f/MAIN	8D
l	cbessl.f/cbessk	246D, 277=, 279
	"/cbessy	169D, 205=, 207
l2gc1	c1.f/output_rc1	721D, 731=, 751, 755
	"/sys_matrix_rc1	262D, 270=, 275, 278, 376, 379
	c2.f/output_rc2	1117D, 1127=
	"/sys_matrix_rc2	324D, 336=, 467, 470, 670, 673
	smc1.f/sys_matrix_c1	86D, 96=, 101, 105, 178, 181
l2gc2	c2.f/output_rc2	1117D, 1128=, 1150, 1154
	"/sys_matrix_rc2	324D, 337=, 343, 347, 451, 454
l2gr	c1.f/sys_matrix_rc1	262D, 271=, 392
lame_c1	c1.f/abc_rc1_invert	559D, 574D
	"/cla_pot	40D, 56D
	"/clb_pot	126D, 142D
	"/output_rc1	674D, 689D, 731, 751, 752, 755, 756, 777, 778, 778, 781, 782, 782
	"/sys_matrix_rc1	217D, 233D, 270, 275, 276, 278, 279, 376, 377, 379, 380
	c2.f/abc_rc2_invert	826D, 841D
	"/output_rc2	1016D, 1031D, 1127
	"/sys_matrix_rc2	224D, 240D, 336, 467, 468, 470, 471, 670, 671, 673, 674
	mr2cf.f/MAIN	107D, 122D, 210=, 212
	smc1.f/abc_c1_invert	305D, 320D
	"/sys_matrix_c1	40D, 56D, 96, 101, 102, 105, 106, 178, 179, 181, 182

Variables :

Symbol	File/Subprogram	Line
lame_c2	c2.f/abc_rc2_invert	864D, 881D
	“ /c2b_pot	43D, 59D
	“ /c2c_pot	134D, 151D
	“ /output_rc2	1054D, 1071D, 1128, 1150, 1151, 1154, 1155, 1176, 1177, 1177, 1180, 1181, 1181
	“ /sys_matrix_rc2	265D, 282D, 337, 343, 344, 347, 348, 451, 452, 454, 455
	mr2cf.f/MAIN	147D, 163D, 233=, 235
lame_rod	c1.f/abc_rc1_invert	583D, 589D
	“ /output_rc1	698D, 704D, 1006, 1006
	“ /sys_matrix_rc1	242D, 248D, 271, 392, 393
	c2.f/abc_rc2_invert	892D, 898D
	“ /output_rc2	1081D, 1087D, 1404, 1404
	“ /sys_matrix_rc2	293D, 299D
	mr2cf.f/MAIN	70D, 74D, 190=, 192
	rf.f/abc_rod_invert	211D, 217D
	“ /abc_rod_solve	161D, 167D
	“ /output	274D, 280D, 314, 328, 432, 432
	“ /rod_pot	25D, 30D
	“ /sys_matrix_rod	74D, 80D, 102, 103, 104
limit	cbessl.f/cbessi	143D, 150=, 153
	“ /cbessj	60D, 81=, 86
	“ /cbessk	246D, 263=, 275
	“ /cbessy	169D, 191=, 203
m	“ /cbessk	246D, 278=, 279
	“ /cbessy	169D, 206=, 207
m_of	c1.f/abc_rc1_invert	595D, 597D, 627=, 638=
	“ /output_rc1	710D, 712D
	“ /sys_matrix_rc1	254D, 256D
	c2.f/abc_rc2_invert	904D, 906D, 952=, 973=
	“ /output_rc2	1093D, 1095D
	“ /sys_matrix_rc2	305D, 307D
	fluids.f/ofl_pot	61D, 63D
	“ /output_of	145D, 147D, 158, 165
	mr2cf.f/MAIN	82D, 84D, 334, 367, 404, 435, 458
	rf.f/abc_rod_invert	223D, 225D, 248=, 253=
	“ /sys_matrix_rod	86D, 88D
	smc1.f/abc_c1_invert	330D, 332D, 373=, 387=
	“ /sys_matrix_c1	66D, 68D

Variables :

Symbol	File/Subprogram	Line
mains	mr2cf.f/MAIN	8D, 284=
maxpiv	rf.f/minv	460D, 479=, 483, 485=, 495
msg	mr2cf.f/MAIN	10D, 286=, 288, 488
mu_1cyl	" / "	25D, 201=, 210, 210, 210, 211
mu_2cyl	" / "	26D, 224=, 233, 233, 233, 234
mu_rod	" / "	20D, 183=, 190, 190, 190, 191
n	c1.f/abc_rc1_invert	533D/*, 538D
	"/cla_pot	14D/*, 19D, 72, 73, 74, 75, 76, 77, 80, 81, 82, 83, 84, 85, 87, 88, 89, 90, 91, 92
	"/c1b_pot	100D/*, 105D, 158, 159, 160, 161, 162, 163, 167, 168, 169, 170, 171, 172, 174, 175, 176, 177, 178, 179
	"/output_rc1	649D/*, 653D, 723, 759, 762, 786, 789, 804, 807, 817, 819, 840, 842, 860, 863, 881, 881, 883, 883, 885, 887, 904, 906, 933, 936, 948, 950, 974, 974, 976, 976, 978, 980, 992, 994
	"/sys_matrix_rc1	191D/*, 196D, 268, 281, 284, 305, 307, 309, 310, 312, 313, 325, 328, 338, 341, 358, 360, 382, 385, 395, 408, 410, 412, 413, 415, 416, 420, 422, 423, 429, 432, 442, 445, 447, 453, 463, 465, 473, 481, 483, 493, 511, 514, 521
	c2.f/abc_rc2_invert	800D/*, 805D, 954, 955, 975, 976
	"/c2b_pot	17D/*, 22D, 76, 77, 78, 79, 80, 81, 84, 85, 86, 87, 88, 89, 91, 92, 93, 94, 95, 96
	"/c2c_pot	107D/*, 112D, 168, 169, 170, 171, 172, 173, 177, 178, 179, 180, 181, 182, 184, 185, 186, 187, 188, 189
	"/output_rc2	991D/*, 995D, 1119, 1158, 1161, 1185, 1188, 1203, 1206, 1216, 1218, 1239, 1241, 1259, 1262, 1279, 1279, 1281, 1281, 1283, 1285, 1302, 1304, 1331, 1334, 1346, 1348, 1372, 1372, 1374, 1374, 1376, 1378, 1390, 1392

Variables :

Symbol	File/Subprogram	Line
“ /sys_matrix_rc2	198D/*, 203D, 334, 351, 353, 355, 357,	
	387, 389, 391, 392, 394, 395, 415, 418,	
	428, 431, 457, 460, 473, 476, 493, 495,	
	497, 498, 500, 501, 507, 509, 511, 512,	
	514, 515, 523, 526, 536, 539, 541, 544,	
	554, 557, 569, 571, 581, 583, 595, 597,	
	607, 609, 633, 636, 647, 649, 676, 679,	
	709, 711, 713, 714, 716, 717, 738, 741,	
	751, 754, 779, 781	
cbessl.f/cbessh1	483D/*, 486D, 489, 489	
“ /cbessh2	497D/*, 499D, 503, 503	
“ /cbessi	141D/*, 143D, 149	
“ /cbessj	58D/*, 60D, 71, 75, 93, 126, 127	
“ /cbessk	244D/*, 246D, 255	
“ /cbessy	167D/*, 169D, 179, 216, 225, 225, 226,	
	226, 226, 227, 229, 229, 229, 230, 232,	
	233	
“ /d1cbessh1	511D/*, 512D, 517, 518	
“ /d1cbessh2	526D/*, 527D, 532, 533	
“ /d1cbessi	453D/*, 457D, 460, 460, 460	
“ /d1cbessj	323D/*, 327D, 330, 330, 330	
“ /d1cbessk	293D/*, 297D, 300, 300, 300	
“ /d1cbessy	356D/*, 360D, 365, 369, 374, 382, 386	
“ /d2cbessi	468D/*, 472D	
“ /d2cbessj	338D/*, 342D, 346, 347, 347	
“ /d2cbessk	308D/*, 312D	
“ /d2cbessy	401D/*, 405D, 411, 416, 423, 433, 438	
“ /fac	21D/*, 22D, 23, 23, 26, 27, 28	
“ /gamma	4D/*, 5D, 6, 6, 9, 10, 11	
“ /psi	38D/*, 39D, 43	
fluids.f/ifl_pot	14D/*, 18D, 42, 43	
“ /ofl_pot	52D/*, 54D, 73, 74, 79, 80	
“ /output_if	91D/*, 95D	
“ /output_of	134D/*, 138D	

Variables :

Symbol	File/Subprogram	Line
	mr2cf.f/MAIN	19D, 268=, 299, 310=, 311=, 312=, 313=, 314=, 316=, 321=, 322=, 326=, 327=, 332=, 338=, 353=, 354=, 355=, 360=, 361=, 366=, 368=, 382=, 383=, 384=, 385=, 386=, 388=, 392=, 393=, 397=, 398=, 403=, 406=, 421=, 422=, 423=, 424=, 425=, 430=, 431=, 433=, 440=, 441=, 445=, 446=, 450=, 451=, 456=, 460=
	rf.f/abc_rod_invert	231D, 234=, 237=
	“ /minv	448D/*, 455D, 456, 456, 456, 456, 456, 456, 463, 464, 469, 470, 471, 472, 481, 489, 500, 501, 510, 510, 512, 514, 519, 522, 528, 529, 530
	“ /output	264D/*, 268D, 288, 315, 316, 330, 337, 342, 348, 356, 367, 367, 368, 378, 379, 394, 401, 402, 416, 416, 417, 424, 425
	“ /rod_pot	14D/*, 19D, 42, 43, 44, 48, 49, 50, 52, 53, 54
	“ /sys_matrix_rod	64D/*, 68D, 98, 106, 107, 113, 115, 116, 118, 118, 125, 139
	smc1.f/abc_c1_invert	279D/*, 284D, 375, 376, 389, 390
	“ /sys_matrix_c1	14D/*, 19D, 94, 109, 111, 113, 115, 133, 135, 137, 138, 140, 141, 150, 153, 163, 166, 184, 187, 205, 207, 209, 210, 212, 213, 225, 228, 238, 241, 256, 258
n2	c1.f/output_rc1	718D, 723=, 752, 756, 778, 782, 877, 879, 970, 972
	“ /sys_matrix_rc1	264D, 268=, 276, 279, 332, 335, 377, 380, 393, 436, 439, 450
	c2.f/output_rc2	1113D, 1119=, 1151, 1155, 1177, 1181, 1275, 1277, 1368, 1370
	“ /sys_matrix_rc2	326D, 334=, 344, 348, 422, 425, 452, 455, 468, 471, 530, 533, 548, 551, 671, 674, 745, 748
	rf.f/output	284D, 288=, 313, 326, 366, 415
	“ /sys_matrix_rod	95D, 98=, 104, 122
	smc1.f/sys_matrix_c1	88D, 94=, 102, 106, 157, 160, 179, 182, 232, 235

Variables :

Symbol	File/Subprogram	Line
-----	-----	----
na	cbessl.f/cbessi	143D, 149=, 154, 158
	“ /cbessj	60D, 71=, 87, 91, 94
	“ /cbessk	246D, 255=, 265, 266, 271, 273, 273, 277, 278, 283, 283
	“ /cbessy	169D, 179=, 193, 194, 199, 201, 205, 206, 211, 217
	“ /psi	39D, 43=, 44, 47
no	c1.f/output_rc1	719D, 728=, 962, 1007
	c2.f/output_rc2	1114D, 1124=, 1360, 1405
	rf.f/output	285D, 293=, 409, 433
offluid	c1.f/abc_rc1_invert	597D
	“ /output_rc1	710D
	“ /sys_matrix_rc1	254D
	c2.f/abc_rc2_invert	906D
	“ /output_rc2	1095D
	“ /sys_matrix_rc2	305D
	fluids.f/ofl_pot	61D
	“ /output_of	145D
	mr2cf.f/MAIN	84D
	rf.f/abc_rod_invert	225D
	“ /sys_matrix_rod	86D
	smc1.f/abc_c1_invert	332D
	“ /sys_matrix_c1	66D
ofsc	c1.f/abc_rc1_invert	595D, 597D
	“ /output_rc1	710D, 712D
	“ /sys_matrix_rc1	254D, 256D, 297
	c2.f/abc_rc2_invert	904D, 906D
	c2.f/output_rc2	1093D, 1095D
	“ /sys_matrix_rc2	305D, 307D, 345, 349, 353, 357, 360, 363
	fluids.f/ofl_pot	61D, 63D, 74=, 80=
	“ /output_of	145D, 147D, 158
	mr2cf.f/MAIN	82D, 84D, 334, 367, 404, 435, 458
	rf.f/abc_rod_invert	223D, 225D
	“ /sys_matrix_rod	86D, 88D, 130
	smc1.f/abc_c1_invert	330D, 332D
	“ /sys_matrix_c1	66D, 68D, 103, 107, 111, 115, 118, 121

Variables :

Symbol	File/Subprogram	Line
-----	-----	----
om	c1.f/c1a_pot	14D/*, 20D, 71, 79
	“ /c1b_pot	100D/*, 106D, 157, 166
	“ /sys_matrix_rc1	192D/*, 197D, 297
	c2.f/c2b_pot	17D/*, 23D, 75, 83
	“ /c2c_pot	107D/*, 113D, 167, 176
	“ /sys_matrix_rc2	199D/*, 204D, 345, 349, 353, 357, 360, 363, 686
	fluids.f/ifl_pot	14D/*, 19D, 39
	“ /ofl_pot	52D/*, 55D, 70, 72, 78
	“ /output_if	91D/*, 96D, 116, 123
	“ /output_of	134D/*, 139D, 158, 165
	mr2cf.f/MAIN	21D, 265=, 284, 297=, 310=, 311=, 312=, 313=, 315=, 321=, 322=, 326=, 332=, 338=, 353=, 354=, 355=, 360=, 366=, 368=, 382=, 383=, 384=, 385=, 386=, 392=, 397=, 403=, 406=, 421=, 422=, 423=, 424=, 425=, 430=, 431=, 440=, 441=, 445=, 450=, 456=, 460=, 480
	rf.f/rod_pot	14D/*, 20D, 41, 47
	“ /sys_matrix_rod	64D/*, 69D, 130
	smc1.f/sys_matrix_c1	15D/*, 20D, 103, 107, 111, 115, 118, 121, 194
p	c1.f/c1a_pot	67D, 71=, 72, 73, 74, 75, 76, 77
	“ /c1b_pot	153D, 157=, 158, 159, 160, 161, 162, 163
	c2.f/c2b_pot	71D, 75=, 76, 77, 78, 79, 80, 81
	“ /c2c_pot	163D, 167=, 168, 169, 170, 171, 172, 173
	rf.f/rod_pot	38D, 41=, 42, 43, 44
p11	c1.f/output_rc1	719D, 729=, 962, 1007
	c2.f/output_rc2	1114D, 1125=, 1360, 1405
	rf.f/output	285D, 294=, 409, 433
p12	c1.f/output_rc1	719D, 730=, 962, 962, 1007, 1007
	c2.f/output_rc2	1114D, 1126=, 1360, 1360, 1405, 1405
	rf.f/output	285D, 295=, 409, 409, 433, 433
part1	cbessl.f/cbessk	248D, 258=, 271=, 285
	“ /cbessy	171D, 186=, 199=, 213
part2	“ /cbessk	248D, 259=, 273=, 285
	“ /cbessy	171D, 187=, 201=, 213
part3	“ /cbessk	248D, 260=, 283=, 285
	“ /cbessy	171D, 188=, 211=, 213

Variables :

Symbol	File/Subprogram	Line
-----	-----	----
pi	“ /cbessj	61D, 68D, 126, 126, 127, 127, 127
	“ /cbessk	247D, 251D
	“ /cbessy	170D, 175D, 199, 201, 211, 232, 232, 232, 233, 233
	fluids.f/output_if	111D, 116=, 118
ps	“ /output_of	153D, 158=, 160
psi	cbessl.f/cbessk	247D
	“ /cbessy	170D
	“ /psi	40D, 45=, 50=
	mr2cf.f/MAIN	33D
q	c1.f/c1a_pot	67D, 79=, 80, 81, 82, 83, 84, 85, 87, 88, 89, 90, 91, 92
	“ /c1b_pot	153D, 166=, 167, 168, 169, 170, 171, 172, 174, 175, 176, 177, 178, 179
	c2.f/c2b_pot	71D, 83=, 84, 85, 86, 87, 88, 89, 91, 92, 93, 94, 95, 96
	“ /c2c_pot	163D, 176=, 177, 178, 179, 180, 181, 182, 184, 185, 186, 187, 188, 189
	rf.f/rod_pot	38D, 47=, 48, 49, 50, 52, 53, 54
r	c1.f/c1a_pot	14D/*, 20D, 72, 73, 74, 75, 76, 77, 80, 81, 82, 83, 84, 85, 87, 88, 89, 90, 91, 92
	“ /c1b_pot	100D/*, 106D, 158, 159, 160, 161, 162, 163, 167, 168, 169, 170, 171, 172, 174, 175, 176, 177, 178, 179
	“ /output_rc1	649D/*, 654D, 724, 752, 756, 759, 760, 762, 763, 777, 781, 786, 789, 804, 807, 817, 819, 840, 842, 860, 863, 877, 879, 881, 883, 885, 887, 904, 904, 906, 906, 933, 936, 948, 948, 950, 950, 970, 972, 974, 976, 978, 980, 992, 992, 994, 994
	c2.f/c2b_pot	17D/*, 23D, 76, 77, 78, 79, 80, 81, 84, 85, 86, 87, 88, 89, 91, 92, 93, 94, 95, 96
	“ /c2c_pot	107D/*, 113D, 168, 169, 170, 171, 172, 173, 177, 178, 179, 180, 181, 182, 184, 185, 186, 187, 188, 189

Variables :

Symbol	File/Subprogram	Line
“ /output_rc2	991D/*, 996D, 1120, 1151, 1155, 1158,	
	1159, 1161, 1162, 1176, 1180, 1185,	
	1188, 1203, 1206, 1216, 1218, 1239,	
	1241, 1259, 1262, 1275, 1277, 1279,	
	1281, 1283, 1285, 1302, 1302, 1304,	
	1304, 1331, 1334, 1346, 1346, 1348,	
	1348, 1368, 1370, 1372, 1374, 1376,	
	1378, 1390, 1390, 1392, 1392	
cbessl.f/cbessh1	483D/*, 484D, 489, 489	
“ /cbessh2	497D/*, 498D, 503, 503	
“ /cbessi	141D/*, 144D, 148	
“ /cbessj	58D/*, 61D, 70	
“ /cbessk	244D/*, 247D, 253, 273	
“ /cbessy	167D/*, 170D, 177, 201	
“ /dlcbessh1	511D/*, 513D, 517, 518	
“ /dlcbessh2	526D/*, 528D, 532, 533	
“ /dlcbessi	453D/*, 455D, 460, 460, 460	
“ /dlcbessj	323D/*, 325D, 330, 330, 330	
“ /dlcbessk	293D/*, 295D, 300, 300, 300	
“ /dlcbessy	356D/*, 358D, 367, 371, 371, 371, 372,	
	372, 376, 376, 376, 377, 377, 378, 378,	
	379, 384, 384, 384, 388, 388, 388, 389,	
	389	
“ /d2cbessi	468D/*, 470D	
“ /d2cbessj	338D/*, 340D, 346, 347, 347	
“ /d2cbessk	308D/*, 310D	
“ /d2cbessy	401D/*, 403D, 413, 413, 413, 414, 414,	
	418, 418, 418, 419, 419, 420, 420, 421,	
	425, 425, 425, 426, 426, 427, 427, 428,	
	428, 429, 429, 429, 430, 430, 435, 435,	
	435, 436, 436, 440, 440, 440, 441, 441,	
	441, 442	
fluids.f/ifl_pot	14D/*, 19D, 42, 43	
“ /ofl_pot	52D/*, 55D, 73, 74, 79, 80	
“ /output_if	91D/*, 96D	
“ /output_of	134D/*, 139D	
mr2cf.f/MAIN	20D, 267=, 299, 318, 321=, 322=, 324,	
	324, 326=, 327=, 330, 332=, 338=, 358,	
	360=, 361=, 364, 366=, 368=, 390, 392=,	
	393=, 395, 395, 397=, 398=, 401, 403=,	

Variables :

Symbol	File/Subprogram	Line
		406=, 437, 440=, 441=, 443, 443, 445=, 446=, 448, 448, 450=, 451=, 454, 456=, 460=
	rf.f/output	264D/*, 269D, 289, 312, 316, 325, 330, 337, 337, 342, 348, 356, 366, 367, 368, 379, 394, 402, 415, 416, 417, 425
	“ /rod_pot	14D/*, 20D, 42, 43, 44, 48, 49, 50, 52, 53, 54
r2	c1.f/output_rc1	719D, 724=, 753, 757, 778, 782, 877, 879, 881, 883, 970, 972, 974, 976
	c2.f/output_rc2	1114D, 1120=, 1152, 1156, 1177, 1181, 1275, 1277, 1279, 1281, 1368, 1370, 1372, 1374
	rf.f/output	285D, 289=, 313, 315, 326, 366, 367, 378, 401, 415, 416, 424
r_1cyl	mr2cf.f/MAIN	25D, 203=, 212, 213
r_2cyl	“ / “	26D, 226=, 235, 236
r_rod	“ / “	20D, 184=, 192, 193
ri	c2.f/sys_matrix_rc2	199D/*, 204D, 686
	fluids.f/output_if	91D/*, 96D, 116
	mr2cf.f/MAIN	23D, 278=, 315=, 322=, 432=, 441=
	smc1.f/sys_matrix_c1	15D/*, 20D, 194
ro	c1.f/sys_matrix_rc1	192D/*, 197D, 297
	c2.f/sys_matrix_rc2	199D/*, 204D, 345, 349, 353, 357, 360, 363
	fluids.f/output_of	134D/*, 139D, 158
	mr2cf.f/MAIN	21D, 275=, 315=, 338=, 355=, 368=, 386=, 406=, 431=, 460=
	rf.f/sys_matrix_rod	64D/*, 69D, 130
	smc1.f/sys_matrix_c1	15D/*, 20D, 103, 107, 111, 115, 118, 121

Variables :

Symbol	File/Subprogram	Line
rod	c1.f/abc_rc1_invert	581D
	"/output_rc1	696D
	"/sys_matrix_rc1	240D
	c2.f/abc_rc2_invert	890D
	"/output_rc2	1079D
	"/sys_matrix_rc2	291D
	mr2cf.f/MAIN	72D
	rf.f/abc_rod_invert	209D
	"/abc_rod_solve	159D
	"/output	272D
	"/rod_pot	23D
	"/sys_matrix_rod	72D
s1	"/minv	460D, 482=, 483
shear_c1	c1.f/abc_rc1_invert	559D, 574D
	"/cla_pot	40D, 56D
	"/clb_pot	126D, 142D
	"/output_rc1	674D, 689D, 731, 759, 762, 765, 767, 779, 783, 786, 789
	"/sys_matrix_rc1	217D, 233D, 270, 281, 284, 287, 289, 301, 303, 305, 307, 309, 312, 325, 328, 331, 334, 337, 340, 382, 385, 388, 390, 404, 406, 408, 410, 412, 415, 429, 432, 435, 438, 441, 444
	c2.f/abc_rc2_invert	826D, 841D
	"/output_rc2	1016D, 1031D, 1127
	"/sys_matrix_rc2	224D, 240D, 336, 473, 476, 479, 481, 503, 505, 507, 509, 511, 514, 541, 544, 547, 550, 553, 556, 676, 679, 682, 684, 705, 707, 709, 711, 713, 716, 738, 741, 744, 747, 750, 753
	mr2cf.f/MAIN	107D, 122D, 211=, 212, 213
	smc1.f/abc_c1_invert	305D, 320D
	"/sys_matrix_c1	40D, 56D, 96, 109, 113, 117, 120, 129, 131, 133, 135, 137, 140, 150, 153, 156, 159, 162, 165, 184, 187, 190, 192, 201, 203, 205, 207, 209, 212, 225, 228, 231, 234, 237, 240

Variables :

Symbol	File/Subprogram	Line
shear_c2	c2.f/abc_rc2_invert	864D, 881D
	"/c2b_pot	43D, 59D
	"/c2c_pot	134D, 151D
	"/output_rc2	1054D, 1071D, 1128, 1158, 1161, 1164, 1166, 1178, 1182, 1185, 1188
	"/sys_matrix_rc2	265D, 282D, 337, 351, 355, 359, 362, 383, 385, 387, 389, 391, 394, 415, 418, 421, 424, 427, 430, 457, 460, 463, 465, 489, 491, 493, 495, 497, 500, 523, 526, 529, 532, 535, 538
	mr2cf.f/MAIN	147D, 163D, 234=, 235, 236
shear_rod	c1.f/abc_rc1_invert	583D, 589D
	"/output_rc1	698D, 704D, 1006
	"/sys_matrix_rc1	242D, 248D, 271, 395, 398, 418, 420, 422, 447, 449, 452
	c2.f/abc_rc2_invert	892D, 898D
	"/output_rc2	1081D, 1087D, 1404
	"/sys_matrix_rc2	293D, 299D
	mr2cf.f/MAIN	70D, 74D, 191=, 192, 193
	rf.f/abc_rod_invert	211D, 217D
	"/abc_rod_solve	161D, 167D
	"/output	274D, 280D, 314, 328, 432
	"/rod_pot	25D, 30D
	"/sys_matrix_rod	74D, 80D, 102, 106, 107, 109, 111, 113, 115, 118, 121, 125
si	mr2cf.f/MAIN	9D
size	c1.f/abc_rc1_invert	603D, 606=, 610=
	c2.f/abc_rc2_invert	924D, 927=, 931=
	mr2cf.f/MAIN	32D, 281=
	smc1.f/abc_c1_invert	350D, 353=, 357=
sm	mr2cf.f/MAIN	34D, 355=, 356=
	rf.f/abc_rod_invert	201D/*, 207D, 237=
	"/abc_rod_solve	152D/*, 157D, 177, 177, 177, 177, 178, 178, 178, 178, 179, 179, 179, 179, 181, 181, 181, 181, 182, 182, 182, 182, 183, 183, 183, 183, 186, 186, 186, 187, 187, 187, 187, 187, 187, 188, 188, 188, 188, 188, 188, 189, 189, 189
	"/sys_matrix_rod	64D/*, 70D, 102=, 106=, 109=, 111=, 113=, 115=, 118=, 121=, 124=, 130=, 133=, 135=, 137=, 139=, 141=, 143=

Variables :

Symbol	File/Subprogram	Line
smc1	mr2cf.f/MAIN	59D, 315=, 316=
	smc1.f/abc_c1_invert	279D/*, 286D, 357=
	" /sys_matrix_c1	15D/*, 21D, 101=, 105=, 109=, 113=,
		117=, 120=, 123=, 129=, 131=, 133=,
		135=, 137=, 140=, 143=, 150=, 153=,
		156=, 159=, 162=, 165=, 168=, 178=,
		181=, 184=, 187=, 190=, 192=, 194=,
		201=, 203=, 205=, 207=, 209=, 212=,
		215=, 225=, 228=, 231=, 234=, 237=,
		240=, 243=, 252=, 254=, 256=, 258=,
		260=, 262=, 264=
smc1inv	mr2cf.f/MAIN	59D
	smc1.f/abc_c1_invert	351D, 357=, 367, 368, 369, 370, 371,
		372, 379, 381, 382, 383, 384, 385, 386,
		393
sminv	mr2cf.f/MAIN	35D
	rf.f/abc_rod_invert	232D, 237=, 245, 246, 247, 248, 250,
		251, 252, 253
smrc1	c1.f/abc_rc1_invert	533D/*, 540D, 610=
	c1.f/sys_matrix_rc1	192D/*, 198D, 275=, 278=, 281=, 284=,
		287=, 289=, 291=, 293=, 295=, 297=,
		301=, 303=, 305=, 307=, 309=, 312=,
		315=, 317=, 319=, 321=, 325=, 328=,
		331=, 334=, 337=, 340=, 343=, 345=,
		347=, 349=, 354=, 356=, 358=, 360=,
		362=, 364=, 366=, 368=, 370=, 372=,
		376=, 379=, 382=, 385=, 388=, 390=,
		392=, 395=, 398=, 400=, 404=, 406=,
		408=, 410=, 412=, 415=, 418=, 420=,
		422=, 425=, 429=, 432=, 435=, 438=,
		441=, 444=, 447=, 449=, 452=, 455=,
		459=, 461=, 463=, 465=, 467=, 469=,
		471=, 473=, 475=, 477=, 481=, 483=,
		485=, 487=, 489=, 491=, 493=, 495=,
		497=, 499=, 503=, 505=, 507=, 509=,
		511=, 514=, 517=, 519=, 521=, 524=
	mr2cf.f/MAIN	48D, 387=, 388=
smrc1inv	c1.f/abc_rc1_invert	604D, 610=, 618, 619, 620, 621, 622,
		623, 624, 625, 626, 627, 629, 630, 631,
		632, 633, 634, 635, 636, 637, 638
	mr2cf.f/MAIN	48D

Variables :

Symbol	File/Subprogram	Line
smrc2	c2.f/abc_rc2_invert	800D/*, 807D, 931=
	“ /sys_matrix_rc2	199D/*, 205D, 343=, 347=, 351=, 355=,
		359=, 362=, 365=, 367=, 369=, 371=,
		373=, 375=, 377=, 383=, 385=, 387=,
		389=, 391=, 394=, 397=, 399=, 401=,
		403=, 405=, 407=, 409=, 415=, 418=,
		421=, 424=, 427=, 430=, 433=, 435=,
		437=, 439=, 441=, 443=, 445=, 451=,
		454=, 457=, 460=, 463=, 465=, 467=,
		470=, 473=, 476=, 479=, 481=, 483=,
		489=, 491=, 493=, 495=, 497=, 500=,
		503=, 505=, 507=, 509=, 511=, 514=,
		517=, 523=, 526=, 529=, 532=, 535=,
		538=, 541=, 544=, 547=, 550=, 553=,
		556=, 559=, 565=, 567=, 569=, 571=,
		573=, 575=, 577=, 579=, 581=, 583=,
		585=, 587=, 589=, 595=, 597=, 599=,
		601=, 603=, 605=, 607=, 609=, 611=,
		613=, 615=, 617=, 619=, 625=, 627=,
		629=, 631=, 633=, 636=, 639=, 641=,
		643=, 645=, 647=, 649=, 651=, 658=,
		660=, 662=, 664=, 666=, 668=, 670=,
		673=, 676=, 679=, 682=, 684=, 686=,
		693=, 695=, 697=, 699=, 701=, 703=,
		705=, 707=, 709=, 711=, 713=, 716=,
		719=, 726=, 728=, 730=, 732=, 734=,
		736=, 738=, 741=, 744=, 747=, 750=,
		753=, 763=, 765=, 767=, 769=, 771=,
		773=, 775=, 777=, 779=, 781=, 783=,
		785=, 787=
	mr2cf.f/MAIN	54D, 432=, 433=
smrc2inv	c2.f/abc_rc2_invert	925D, 931=, 939, 940, 941, 942, 943,
		944, 945, 946, 947, 948, 949, 950, 951,
		960, 961, 962, 963, 964, 965, 966, 967,
		968, 969, 970, 971, 972
	mr2cf.f/MAIN	54D

Variables :

Symbol	File/Subprogram	Line
sp_cyl_a1	c1.f/abc_rc1_invert	544D, 562D
	“/cla_pot	25D, 44D, 74=
	“/clb_pot	111D, 130D
	“/output_rc1	659D, 677D, 752, 778, 799, 817, 855, 877, 928, 970
	“/sys_matrix_rc1	202D, 221D, 377, 429, 481, 503
	c2.f/abc_rc2_invert	811D, 829D
	“/output_rc2	1001D, 1019D
	“/sys_matrix_rc2	209D, 228D, 671, 739
	mr2cf.f/MAIN	92D, 110D
	smc1.f/abc_c1_invert	290D, 308D
	“/sys_matrix_c1	25D, 44D, 179, 226
sp_cyl_a2	c1.f/abc_rc1_invert	545D, 563D
	“/cla_pot	26D, 45D, 77=
	“/clb_pot	112D, 131D
	“/output_rc1	660D, 678D, 756, 782, 801, 819, 879, 972
	“/sys_matrix_rc1	203D, 222D, 380, 432, 483, 505
	c2.f/abc_rc2_invert	812D, 830D
	“/output_rc2	1002D, 1020D
	“/sys_matrix_rc2	210D, 229D, 674, 742
	mr2cf.f/MAIN	93D, 111D
	smc1.f/abc_c1_invert	291D, 309D
	“/sys_matrix_c1	26D, 45D, 182, 229
sp_cyl_b1	c1.f/abc_rc1_invert	546D, 564D
	“/cla_pot	27D, 46D
	“/clb_pot	113D, 132D, 160=
	“/output_rc1	661D, 679D
	“/sys_matrix_rc1	204D, 223D, 276, 326
	c2.f/abc_rc2_invert	813D, 831D
	“/output_rc2	1003D, 1021D
	“/sys_matrix_rc2	211D, 230D, 468, 541, 607, 639
	mr2cf.f/MAIN	94D, 112D
	smc1.f/abc_c1_invert	292D, 310D
	“/sys_matrix_c1	27D, 46D, 102, 151

Variables :

Symbol	File/Subprogram	Line
sp_cy1_b2	c1.f/abc_rc1_invert	547D, 565D
	"/c1a_pot	28D, 47D
	"/c1b_pot	114D, 133D, 163=
	"/output_rc1	662D, 680D
	"/sys_matrix_rc1	205D, 224D, 279, 329
	c2.f/abc_rc2_invert	814D, 832D
	"/output_rc2	1004D, 1022D
	"/sys_matrix_rc2	212D, 231D, 471, 544, 609, 641
	mr2cf.f/MAIN	95D, 113D
	smc1.f/abc_c1_invert	293D, 311D
	"/sys_matrix_c1	28D, 47D, 106, 154
sp_cy2_b1	c2.f/abc_rc2_invert	849D, 869D
	"/c2b_pot	28D, 47D, 78=
	"/c2c_pot	119D, 139D
	"/output_rc2	1039D, 1059D, 1151, 1177, 1198, 1216, 1254, 1275, 1326, 1368
	"/sys_matrix_rc2	250D, 270D, 452, 523, 595, 625
	mr2cf.f/MAIN	132D, 151D
sp_cy2_b2	c2.f/abc_rc2_invert	850D, 870D
	"/c2b_pot	29D, 48D, 81=
	"/c2c_pot	120D, 140D
	"/output_rc2	1040D, 1060D, 1155, 1181, 1200, 1218, 1277, 1370
	"/sys_matrix_rc2	251D, 271D, 455, 526, 597, 627
	mr2cf.f/MAIN	133D, 152D
sp_cy2_c1	c2.f/abc_rc2_invert	851D, 871D
	"/c2b_pot	30D, 49D
	"/c2c_pot	121D, 141D, 170=
	"/output_rc2	1041D, 1061D
	"/sys_matrix_rc2	252D, 272D, 344, 416
	mr2cf.f/MAIN	134D, 153D
sp_cy2_c2	c2.f/abc_rc2_invert	852D, 872D
	"/c2b_pot	31D, 50D
	"/c2c_pot	122D, 142D, 173=
	"/output_rc2	1042D, 1062D
	c2.f/sys_matrix_rc2	253D, 273D, 348, 419
	mr2cf.f/MAIN	135D, 154D

Variables :

Symbol	File/Subprogram	Line
-----	-----	----
sp_rod	c1.f/abc_rc1_invert	581D, 586D
	"/output_rc1	696D, 701D
	"/sys_matrix_rc1	240D, 245D, 393, 447, 493, 517
	c2.f/abc_rc2_invert	890D, 895D
	"/output_rc2	1079D, 1084D
	"/sys_matrix_rc2	291D, 296D
	mr2cf.f/MAIN	67D, 72D
	rf.f/abc_rod_invert	209D, 214D
	"/abc_rod_solve	159D, 164D
	"/output	272D, 277D, 313, 313, 326, 326, 329, 336, 342, 354, 366, 392, 415
	"/rod_pot	23D, 27D, 44=
	"/sys_matrix_rod	72D, 77D, 104, 118
str	c1.f/output_rc1	720D, 769=, 771
	c2.f/output_rc2	1115D, 1168=, 1170
	rf.f/output	286D, 314=, 319
sum	cbessl.f/cbessi	145D, 155=, 156
	"/cbessj	62D, 88=, 89
	"/fac	22D, 26=, 28, 28=, 30
	"/gamma	5D, 9=, 11, 11=, 13
	"/psi	40D, 42=, 48, 48=, 50
sum1	"/cbessk	248D, 267=, 268
	"/cbessy	171D, 195=, 196
sum2	"/cbessk	248D, 279=, 280
	"/cbessy	171D, 207=, 208
switch	rf.f/minv	460D, 490=, 492
sxx	c1.f/output_rc1	720D, 791=, 793
	c2.f/output_rc2	1115D, 1190=, 1192
	rf.f/output	286D, 328=, 332
t	"/minv	461D, 535=
tft	c1.f/output_rc1	649D/*, 653D, 749, 775, 797, 814, 834, 853, 875, 898, 921, 926, 968
	c2.f/output_rc2	991D/*, 995D, 1148, 1174, 1196, 1213, 1233, 1252, 1273, 1296, 1319, 1324, 1366
	fluids.f/output_if	91D/*, 95D, 114, 121
	"/output_of	134D/*, 138D, 156, 163
	mr2cf.f/MAIN	19D, 269=, 294, 299, 322=, 327=, 338=, 342, 361=, 368=, 372, 393=, 398=, 406=, 410, 441=, 446=, 451=, 460=, 464

Variables :

Symbol	File/Subprogram	Line
	rf.f/output	264D/*, 268D, 310, 323, 334, 340, 346, 352, 364, 375, 388, 390, 413
total	cbessl.f/cbessi	145D, 151=, 156, 156=, 158
	“ /cbessj	62D, 82=, 89, 89=, 91
total1	“ /cbessk	248D, 261=, 268, 268=, 271
	“ /cbessy	172D, 189=, 196, 196=, 199
total2	“ /cbessk	248D, 262=, 280, 280=, 283
	“ /cbessy	172D, 190=, 208, 208=, 211
uc	c1.f/output_rc1	720D, 848=, 850
	c2.f/output_rc2	1115D, 1247=, 1249
ur	rf.f/output	286D, 348=, 350
value	c1.f/output_rc1	649D/*, 655D, 771=, 793=, 811=, 831=, 850=, 869=, 872=, 892=, 895=, 915=, 918=, 922=, 964=, 985=, 1008=
	c2.f/output_rc2	991D/*, 997D, 1170=, 1192=, 1210=, 1230=, 1249=, 1267=, 1270=, 1290=, 1293=, 1313=, 1316=, 1320=, 1362=, 1383=, 1406=
	fluids.f/output_if	91D/*, 97D, 118=, 125=
	“ /output_of	134D/*, 140D, 160=, 167=
	mr2cf.f/MAIN	36D, 322=, 327=, 338=, 343, 343=, 346, 361=, 368=, 373, 373=, 376, 393=, 398=, 406=, 411, 411=, 414, 441=, 446=, 451=, 460=, 465, 465=, 468, 472, 476
	rf.f/output	264D/*, 270D, 319=, 332=, 338=, 344=, 350=, 359=, 362=, 371=, 373=, 383=, 386=, 389=, 411=, 420=, 434=
vc	c1.f/output_rc1	720D, 829=, 831
	c2.f/output_rc2	1115D, 1228=, 1230
vr	rf.f/output	286D, 342=, 344

Variables :

Symbol	File/Subprogram	Line
vrtp_cyl_a1	c1.f/abc_rc1_invert	554D, 570D
" /c1a_pot		35D, 52D, 89=
" /c1b_pot		121D, 138D
" /output_rc1		669D, 685D, 786, 803, 825, 844, 860, 885, 933, 978
" /sys_matrix_rc1		212D, 229D, 412, 442, 467, 489, 511
c2.f/abc_rc2_invert		821D, 837D
" /output_rc2		1011D, 1027D
" /sys_matrix_rc2		219D, 236D, 713, 751, 783
mr2cf.f/MAIN		102D, 118D
smc1.f/abc_c1_invert		300D, 316D
" /sys_matrix_c1		35D, 52D, 209, 238, 260
vrtp_cyl_a2	c1.f/abc_rc1_invert	555D, 571D
" /c1a_pot		36D, 53D, 92=
" /c1b_pot		122D, 139D
" /output_rc1		670D, 686D, 789, 806, 827, 846, 863, 887, 936, 980
" /sys_matrix_rc1		213D, 230D, 415, 445, 469, 491, 514
c2.f/abc_rc2_invert		822D, 838D
" /output_rc2		1012D, 1028D
" /sys_matrix_rc2		220D, 237D, 716, 754, 785
mr2cf.f/MAIN		103D, 119D
smc1.f/abc_c1_invert		301D, 317D
" /sys_matrix_c1		36D, 53D, 212, 241, 262
vrtp_cyl_b1	c1.f/abc_rc1_invert	556D, 572D
" /c1a_pot		37D, 54D
" /c1b_pot		123D, 140D, 176=
" /output_rc1		671D, 687D
" /sys_matrix_rc1		214D, 231D, 309, 338, 362
c2.f/abc_rc2_invert		823D, 839D
" /output_rc2		1013D, 1029D
" /sys_matrix_rc2		221D, 238D, 511, 554, 585, 615, 647
mr2cf.f/MAIN		104D, 120D
smc1.f/abc_c1_invert		302D, 318D, 377, 391
" /sys_matrix_c1		37D, 54D, 118, 137, 163

Variables :

Symbol	File/Subprogram	Line
vrtp_cy1_b2	c1.f/abc_rc1_invert	557D, 573D
“/c1a_pot		38D, 55D
“/c1b_pot		124D, 141D, 179=
“/output_rc1		672D, 688D
“/sys_matrix_rc1		215D, 232D, 312, 341, 364
c2.f/abc_rc2_invert		824D, 840D
“/output_rc2		1014D, 1030D
“/sys_matrix_rc2		222D, 239D, 514, 557, 587, 617, 649
mr2cf.f/MAIN		105D, 121D
smc1.f/abc_c1_invert		303D, 319D, 378, 392
“/sys_matrix_c1		38D, 55D, 121, 140, 166
vrtp_cy2_b1	c2.f/abc_rc2_invert	859D, 877D
“/c2b_pot		38D, 55D, 93=
“/c2c_pot		129D, 147D
“/output_rc2		1049D, 1067D, 1185, 1202, 1224, 1243, 1259, 1283, 1331, 1376
“/sys_matrix_rc2		260D, 278D, 497, 536, 573, 603, 633
mr2cf.f/MAIN		142D, 159D
vrtp_cy2_b2	c2.f/abc_rc2_invert	860D, 878D
“/c2b_pot		39D, 56D, 96=
“/c2c_pot		130D, 148D
c2.f/output_rc2		1050D, 1068D, 1188, 1205, 1226, 1245, 1262, 1285, 1334, 1378
“/sys_matrix_rc2		261D, 279D, 500, 539, 575, 605, 636
mr2cf.f/MAIN		143D, 160D
vrtp_cy2_c1	c2.f/abc_rc2_invert	861D, 879D, 956, 977
“/c2b_pot		40D, 57D
“/c2c_pot		131D, 149D, 186=
“/output_rc2		1051D, 1069D
“/sys_matrix_rc2		262D, 280D, 360, 391, 428
mr2cf.f/MAIN		144D, 161D
vrtp_cy2_c2	c2.f/abc_rc2_invert	862D, 880D, 957, 978
“/c2b_pot		41D, 58D
“/c2c_pot		132D, 150D, 189=
“/output_rc2		1052D, 1070D
“/sys_matrix_rc2		263D, 281D, 363, 394, 431
mr2cf.f/MAIN		145D, 162D

Variables :

Symbol	File/Subprogram	Line
vrtp_rod	c1.f/abc_rc1_invert	582D, 588D
	“/output_rc1	697D, 703D
	“/sys_matrix_rc1	241D, 247D, 422, 453, 475, 497, 521
	c2.f/abc_rc2_invert	891D, 897D
	“/output_rc2	1080D, 1086D
	“/sys_matrix_rc2	292D, 298D
	mr2cf.f/MAIN	69D, 73D
	rf.f/abc_rod_invert	210D, 216D
	“/abc_rod_solve	160D, 166D
	“/output	273D, 279D, 330, 337, 337, 343, 349, 356, 368, 394, 417
	“/rod_pot	24D, 29D, 54=
	“/sys_matrix_rod	73D, 79D, 115, 124, 141
vui	fluids.f/output_if	111D, 123=, 125
vws	“/output_of	153D, 165=, 167
vxp_cyl_a1	c1.f/abc_rc1_invert	549D, 566D
	“/cla_pot	30D, 48D, 82=
	“/clb_pot	116D, 134D
	“/output_rc1	664D, 681D, 759, 840, 881, 904, 948, 974, 992
	“/sys_matrix_rc1	207D, 225D, 382, 408, 436, 463
	c2.f/abc_rc2_invert	816D, 833D
	“/output_rc2	1006D, 1023D
	“/sys_matrix_rc2	214D, 232D, 676, 709, 745, 779
	mr2cf.f/MAIN	97D, 114D
	smc1.f/abc_c1_invert	295D, 312D
	“/sys_matrix_c1	30D, 48D, 184, 205, 232, 256
vxp_cyl_a2	c1.f/abc_rc1_invert	550D, 567D
	“/cla_pot	31D, 49D, 85=
	“/clb_pot	117D, 135D
	“/output_rc1	665D, 682D, 762, 842, 883, 906, 950, 976, 994
	“/sys_matrix_rc1	208D, 226D, 385, 410, 439, 465
	c2.f/abc_rc2_invert	817D, 834D
	“/output_rc2	1007D, 1024D
	“/sys_matrix_rc2	215D, 233D, 679, 711, 748, 781
	mr2cf.f/MAIN	98D, 115D
	smc1.f/abc_c1_invert	296D, 313D
	“/sys_matrix_c1	31D, 49D, 187, 207, 235, 258

Variables :

Symbol	File/Subprogram	Line
vxp_cyl_b1	c1.f/abc_rc1_invert	551D, 568D
	“/c1a_pot	32D, 50D
	“/c1b_pot	118D, 136D, 169=
	“/output_rc1	666D, 683D
	“/sys_matrix_rc1	209D, 227D, 281, 305, 332, 358
c2.f/abc_rc2_invert		818D, 835D
	“/output_rc2	1008D, 1025D
	“/sys_matrix_rc2	216D, 234D, 473, 507, 548, 581
mr2cf.f/MAIN		99D, 116D
smc1.f/abc_c1_invert		297D, 314D, 375, 389
	“/sys_matrix_c1	32D, 50D, 109, 111, 133, 157
vxp_cyl_b2	c1.f/abc_rc1_invert	552D, 569D
	“/c1a_pot	33D, 51D
	“/c1b_pot	119D, 137D, 172=
	“/output_rc1	667D, 684D
	“/sys_matrix_rc1	210D, 228D, 284, 307, 335, 360
c2.f/abc_rc2_invert		819D, 836D
	“/output_rc2	1009D, 1026D
	“/sys_matrix_rc2	217D, 235D, 476, 509, 551, 583
mr2cf.f/MAIN		100D, 117D
smc1.f/abc_c1_invert		298D, 315D, 376, 390
	“/sys_matrix_c1	33D, 51D, 113, 115, 135, 160
vxp_cy2_b1	c2.f/abc_rc2_invert	854D, 873D
	“/c2b_pot	33D, 51D, 86=
	“/c2c_pot	124D, 143D
	“/output_rc2	1044D, 1063D, 1158, 1239, 1279, 1302, 1346, 1372, 1390
	“/sys_matrix_rc2	255D, 274D, 457, 493, 530, 569
mr2cf.f/MAIN		137D, 155D
vxp_cy2_b2	c2.f/abc_rc2_invert	855D, 874D
	“/c2b_pot	34D, 52D, 89=
	“/c2c_pot	125D, 144D
	“/output_rc2	1045D, 1064D, 1161, 1241, 1281, 1304, 1348, 1374, 1392
	“/sys_matrix_rc2	256D, 275D, 460, 495, 533, 571
mr2cf.f/MAIN		138D, 156D

Variables :

Symbol	File/Subprogram	Line
vxp_cy2_c1	c2.f/abc_rc2_invert	856D, 875D, 954, 975
	"/c2b_pot	35D, 53D
	"/c2c_pot	126D, 145D, 179=
	"/output_rc2	1046D, 1065D
	"/sys_matrix_rc2	257D, 276D, 351, 353, 387, 422
	mr2cf.f/MAIN	139D, 157D
vxp_cy2_c2	c2.f/abc_rc2_invert	857D, 876D, 955, 976
	"/c2b_pot	36D, 54D
	"/c2c_pot	127D, 146D, 182=
	"/output_rc2	1047D, 1066D
	"/sys_matrix_rc2	258D, 277D, 355, 357, 389, 425
	mr2cf.f/MAIN	140D, 158D
vxp_rod	c1.f/abc_rc1_invert	581D, 587D
	"/output_rc1	696D, 702D
	"/sys_matrix_rc1	240D, 246D, 396, 420, 450, 473
	c2.f/abc_rc2_invert	890D, 896D
	"/output_rc2	1079D, 1085D
	"/sys_matrix_rc2	291D, 297D
	mr2cf.f/MAIN	68D, 72D
	rf.f/abc_rod_invert	209D, 215D
	"/abc_rod_solve	159D, 165D
	"/output	272D, 278D, 315, 348, 367, 378, 401, 416, 424
	"/rod_pot	23D, 28D, 50=
	"/sys_matrix_rod	72D, 78D, 106, 113, 122, 139
wc	c1.f/output_rc1	720D, 809=, 811
	c2.f/output_rc2	1115D, 1208=, 1210
work	mr2cf.f/MAIN	35D
	rf.f/abc_rod_invert	232D, 237=
	"/minv	448D/*, 456D, 465=, 471=, 472=, 479, 482, 485, 490, 491, 491=, 492=, 502, 502, 502, 503, 503=, 505=, 513, 515, 515=, 521, 523, 523, 523=, 530
workc1	mr2cf.f/MAIN	59D
	smc1.f/abc_c1_invert	351D, 357=
workrc1	c1.f/abc_rc1_invert	604D, 610=
	mr2cf.f/MAIN	48D
workrc2	c2.f/abc_rc2_invert	925D, 931=
	mr2cf.f/MAIN	54D
wr	rf.f/output	286D, 336=, 338

Variables :

Symbol	File/Subprogram	Line
y2	cbessl.f/d1cbessy	362D, 371=, 372, 376=, 379, 388=, 389
	" /d2cbessy	407D, 413=, 414, 418=, 421, 425=, 430, 440=, 442
y3	" /d1cbessy	362D, 377=, 379
	" /d2cbessy	407D, 419=, 421, 426=, 430
y4	" / "	407D, 428=
yn	mr2cf.f/MAIN	8D
z	cbessl.f/cbessi	145D, 148=, 155, 158
	" /cbessj	62D, 70=, 72, 76, 77, 88, 91, 126, 126, 127
	" /cbessk	248D, 253=, 254, 267, 271, 279, 283
	" /cbessy	171D, 177=, 178, 180, 195, 199, 207, 211, 225, 227, 229, 230, 232, 232, 233
	mr2cf.f/MAIN	8D
z2	cbessl.f/cbessk	248D, 254=, 273
	" /cbessy	171D, 178=, 201
zero	c1.f/output_rc1	654D, 727=, 868, 869, 891, 892, 914, 915, 940, 941, 958, 959, 984, 985, 1002, 1003
	c2.f/output_rc2	996D, 1123=, 1266, 1267, 1289, 1290, 1312, 1313, 1338, 1339, 1356, 1357, 1382, 1383, 1400, 1401
	rf.f/output	269D, 292=, 358, 359, 370, 371, 382, 383, 396, 397, 405, 406, 419, 420, 428, 429
zeta_1cyl	mr2cf.f/MAIN	25D, 200=, 209
zeta_2cyl	" / "	26D, 223=, 232
zeta_rod	" / "	21D, 185=, 189
zm	cbessl.f/cbessj	61D, 72=, 79
	" /cbessy	170D, 180=, 182

LABELS

Labels :

Symbol	File/Subprogram	Line
-----	-----	----
4	mr2cf.f/MAIN	488, 489D
10	cbessl.f/cbessi	153, 157D
	“ /cbessj	86, 90D
	“ /cbessk	265, 269D
	“ /cbessy	193, 197D
	“ /fac	27, 29D
	“ /gamma	10, 12D
	“ /psi	47, 49D
	mr2cf.f/MAIN	299, 300D
20	cbessl.f/cbessk	275, 281D
	“ /cbessy	203, 209D
	mr2cf.f/MAIN	294, 295D
51	“ / “	476, 477D
60	“ / “	478, 479D
70	“ / “	480, 481D
100	rf.f/minv	464, 466D
110	“ / “	463, 467D
120	“ / “	470, 473D
130	“ / “	469, 474D
140	“ / “	478D, 510
150	“ / “	481, 483, 486D
160	“ / “	489, 493D
170	“ / “	488, 495D
180	“ / “	501, 504D
190	“ / “	500, 506D
200	“ / “	514, 516D
210	“ / “	512, 517D
220	“ / “	522, 524D
230	“ / “	520, 525D
240	“ / “	519, 526D
250	“ / “	529, 531D
260	“ / “	528, 532D
400	mr2cf.f/MAIN	334, 335D, 367, 404, 435, 458
410	“ / “	427, 428D
900	rf.f/minv	499, 552D

STRINGS

Strings :

Symbol	File/Subprogram	Line
--------	-----------------	------

' r = ' mr2cf.f/MAIN		300
' D_IF = ' " / "		428
' IFSC = ' " / "		428
' M_OF = ' " / "		335
' OFSC = ' " / "		335
' Om = ' " / "		482
' ao_1cyl = ' " / "		301
' bo_c1 = ' " / "		301
' co_c2 = ' " / "		301
'BMAG < ARTIFICIAL ZERO (1.0D-12) RETURNING FROM MINV'		
rf.f/minv		555
'Enter tft,k,f,emt: (where emt=1 1CYLINDER, emt=2 ROD, emt=3 ROD&CYL, emt=4 2CYLINDERS, using mr2cf.f)'		
mr2cf.f/MAIN		290
'MAIN PROGRAM n = ' " / "		300
'SUB OUTPUT k = ' " / "		481
'SUB OUTPUT THE ANSWER IS g(jk) = ' " / "		479
'SUB OUTPUT THE ANSWER IS value = ' " / "		477
'Type 1 if you wish to quit' " / "		487
'd1_IFSC = ' " / "		428
'd1_OFSC = ' " / "		335
'tft = ' " / "		302

INDEX

Index Thu Oct 19 16:48:44 1995

General options in effect:

-o mr2cf.lst

Files:

/home/kwcode/analysis/tota/RODCY2/mr2cf.f.....	27
/home/kwcode/analysis/tota/RODCY2/cbessl.f.....	40
/home/kwcode/analysis/tota/RODCY2/rf.f.....	54
/home/kwcode/analysis/tota/RODCY2/c1.f.....	68
/home/kwcode/analysis/tota/RODCY2/c2.f.....	92
/home/kwcode/analysis/tota/RODCY2/fluids.f	124
/home/kwcode/analysis/tota/RODCY2/smc1.f.....	130

Cross Reference:

Subprograms	142
Variables.....	147
Labels.....	203
Strings	204

Legend:

D	Definition/declaration
=	Possibly modified occurrence
*	Dummy argument

APPENDIX A

INTERMEDIATE VARIABLES FOR NUWC-NPT TR 11,043

Table A-1. Common Block /ROD/

FORTTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Solid Cylinder Variables)
SP_rod	$J_n(p_1 a)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again for $(p_1 r_1)$.
d1_SP_rod	$\frac{\partial}{\partial r} J_n(p_1 a)$	First derivative of $J_n(p_1 r)$ with respect to r . Can be evaluated again for $(p_1 r_1)$.
d2_SP_rod	$\frac{\partial^2}{\partial r^2} J_n(p_1 a)$	Second derivative of $J_n(p_1 r)$ with respect to r . Can be evaluated again for $(p_1 r_1)$.
VXP_rod	$J_n(q_1 a)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again for $(q_1 r_1)$.
d1_VXP_rod	$\frac{\partial}{\partial r} J_n(q_1 a)$	First derivative of $J_n(q_1 r)$ with respect to r . Can be evaluated again for $(q_1 r_1)$.
d2_VXP_rod	$\frac{\partial^2}{\partial r^2} J_n(q_1 a)$	Second derivative of $J_n(q_1 r)$ with respect to r . Can be evaluated again for $(q_1 r_1)$.
VRTP_rod	$J_{n+1}(q_1 a)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again for $(q_1 r_1)$.
d1_VRTP_rod	$\frac{\partial}{\partial r} J_{n+1}(q_1 a)$	First derivative of $J_{n+1}(q_1 r)$ with respect to r . Can be evaluated again for $(q_1 r_1)$.
d2_VRTP_rod	$\frac{\partial^2}{\partial r^2} J_{n+1}(q_1 a)$	Second derivative of $J_{n+1}(q_1 r)$ with respect to r . Can be evaluated again for $(q_1 r_1)$.

Table A-1. Common Block /ROD/ (Cont'd)

FORTTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Solid Cylinder Variables)
A1_rod	A_1^{C1}	Undetermined constant.
B1_rod	B_1^{C1}	Undetermined constant.
C1_rod	C_1^{C1}	Undetermined constant.
lame_rod	λ_1	Lame constant for the solid cylinder.
shear_rod	μ_1	Shear modulus for the solid cylinder.

Table A-2. Common Block /CYLINDER1/

FORTTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Outer Cylinder Variables)
SP_CY1_a1	$J_n(p_2a)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again for (p_2r_1) .
d1_SP_CY1_a1	$\frac{\partial}{\partial r} J_n(p_2a)$	First derivative of $J_n(p_2r)$ with respect to r . Can be evaluated again for (p_2r_1) .
d2_SP_CY1_a1	$\frac{\partial^2}{\partial r^2} J_n(p_2a)$	Second derivative of $J_n(p_2r)$ with respect to r . Can be evaluated again for (p_2r_1) .
VXP_CY1_a1	$J_n(q_2a)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again for (q_2r_1) .
d1_VXP_CY1_a1	$\frac{\partial}{\partial r} J_n(q_2a)$	First derivative of $J_n(q_2r)$ with respect to r . Can be evaluated again for (q_2r_1) .
d2_VXP_CY1_a1	$\frac{\partial^2}{\partial r^2} J_n(q_2a)$	Second derivative of $J_n(q_2r)$ with respect to r . Can be evaluated again for (q_2r_1) .

Table A-2. Common Block /CYLINDER1/ (Cont'd)

FORTTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Outer Cylinder Variables)
VRTP_CY1_a1	$J_{n+1}(q_2a)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again for (q_2r_1) .
d1_VRTP_CY1_a1	$\frac{\partial}{\partial r} J_{n+1}(q_2a)$	First derivative of $J_{n+1}(q_2r)$ with respect to r . Can be evaluated again for (q_2r_1) .
d2_VRTP_CY1_a1	$\frac{\partial^2}{\partial r^2} J_{n+1}(q_2a)$	Second derivative of $J_{n+1}(q_2r)$ with respect to r . Can be evaluated again for (q_2r_1) .
SP_CY1_b1	$J_n(p_2b)$	Bessel function of the first kind, used for the scalar displacement potential.
d1_SP_CY1_b1	$\frac{\partial}{\partial r} J_n(p_2b)$	First derivative of $J_n(p_2r)$ with respect to r .
d2_SP_CY1_b1	$\frac{\partial^2}{\partial r^2} J_n(p_2b)$	Second derivative of $J_n(p_2r)$ with respect to r .
VXP_CY1_b1	$J_n(q_2b)$	Bessel function of the first kind, used for the vector x displacement potential.
d1_VXP_CY1_b1	$\frac{\partial}{\partial r} J_n(q_2b)$	First derivative of $J_n(q_2r)$ with respect to r .
d2_VXP_CY1_b1	$\frac{\partial^2}{\partial r^2} J_n(q_2b)$	Second derivative of $J_n(q_2r)$ with respect to r .
VRTP_CY1_b1	$J_{n+1}(q_2b)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$.

Table A-2. Common Block /CYLINDER1/ (Cont'd)

FORTTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Outer Cylinder Variables)
d1_VRTP_CY1_b1	$\frac{\partial}{\partial r} J_{n+1}(q_2 b)$	First derivative of $J_{n+1}(q_2 r)$ with respect to r .
d2_VRTP_CY1_b1	$\frac{\partial^2}{\partial r^2} J_{n+1}(q_2 b)$	Second derivative of $J_{n+1}(q_2 r)$ with respect to r .
SP_CY1_a2	$Y_n(p_2 a)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again at $(p_2 r_1)$.
d1_SP_CY1_a2	$\frac{\partial}{\partial r} Y_n(p_2 a)$	First derivative of $Y_n(p_2 r)$ with respect to r . Can be evaluated again at $(p_2 r_1)$.
d2_SP_CY1_a2	$\frac{\partial^2}{\partial r^2} Y_n(p_2 a)$	Second derivative of $Y_n(p_2 r)$ with respect to r . Can be evaluated again at $(p_2 r_1)$.
VXP_CY1_a2	$Y_n(q_2 a)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again at $(q_2 r_1)$.
d1_VXP_CY1_a2	$\frac{\partial}{\partial r} Y_n(q_2 a)$	First derivative of $Y_n(q_2 r)$ with respect to r . Can be evaluated again at $(q_2 r_1)$.
d2_VXP_CY1_a2	$\frac{\partial^2}{\partial r^2} Y_n(q_2 a)$	Second derivative of $Y_n(q_2 r)$ with respect to r . Can be evaluated again at $(q_2 r_1)$.
VRTP_CY1_a2	$Y_{n+1}(q_2 a)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again at $(q_2 r_1)$.
d1_VRTP_CY1_a2	$\frac{\partial}{\partial r} Y_{n+1}(q_2 a)$	First derivative of $Y_{n+1}(q_2 r)$ with respect to r . Can be evaluated again at $(q_2 r_1)$.

Table A-2. Common Block /CYLINDER1/ (Cont'd)

FORTTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Outer Cylinder Variables)
d2_V RTP_CY1_a2	$\frac{\partial^2}{\partial r^2} Y_{n+1}(q_2 a)$	Second derivative of $Y_{n+1}(q_2 r)$ with respect to r . Can be evaluated again at $(q_2 r_1)$.
SP_CY1_b2	$Y_n(p_2 b)$	Bessel function of the first kind, used for the scalar displacement potential.
d1_SP_CY1_b2	$\frac{\partial}{\partial r} Y_n(p_2 b)$	First derivative of $Y_n(p_2 r)$ with respect to r .
d2_SP_CY1_b2	$\frac{\partial^2}{\partial r^2} Y_n(p_2 b)$	Second derivative of $Y_n(p_2 r)$ with respect to r .
VXP_CY1_b2	$Y_n(q_2 b)$	Bessel function of the first kind, used for the vector x displacement potential.
d1_VXP_CY1_b2	$\frac{\partial}{\partial r} Y_n(q_2 b)$	First derivative of $Y_n(q_2 r)$ with respect to r .
d2_VXP_CY1_b2	$\frac{\partial^2}{\partial r^2} Y_n(q_2 b)$	Second derivative of $Y_n(q_2 r)$ with respect to r .
V RTP_CY1_b2	$Y_{n+1}(q_2 b)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$.
d1_V RTP_CY1_b2	$\frac{\partial}{\partial r} Y_{n+1}(q_2 b)$	First derivative of $Y_{n+1}(q_2 r)$ with respect to r .
d2_V RTP_CY1_b2	$\frac{\partial^2}{\partial r^2} Y_{n+1}(q_2 b)$	Second derivative of $Y_{n+1}(q_2 r)$ with respect to r .
A1_C1	A_1^{C2}	Undetermined constant.
A2_C1	A_2^{C2}	Undetermined constant.

Table A-2. Common Block /CYLINDER1/ (Cont'd)

FORTTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION (Outer Cylinder Variables)
B1_C1	B_1^{C2}	Undetermined constant.
B2_C1	B_2^{C2}	Undetermined constant.
C1_C1	C_1^{C2}	Undetermined constant.
C2_C1	C_2^{C2}	Undetermined constant.
lame_c1	λ_2	Lame constant for the second cylinder.
shear_c1	μ_2	Shear modulus for the second cylinder.

Table A-3. Calculated Solid Cylinder Variable

FORTTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION
Ec_rod	E_1^*	Complex Young's modulus.

Table A-4. Calculated Outer Cylinder Variables

FORTTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION
Ec_c1	E_2^*	Complex Young's modulus.
bo_c1	b	Outer radius.

Table A-5. Common Block /OFLUID/

FORTTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION
OFSC	$H_n^{(1)}(g_2 b)$ or $K_n(f_2 b)$	Modified Bessel function, used for the scalar displacement potential. Can also be evaluated at $(g_2 r_1)$ and $(f_2 r_1)$.
d1_OFSC	$\frac{\partial}{\partial r} H_n^{(1)}(g_2 b)$ or $\frac{\partial}{\partial r} K_n(f_2 b)$	First derivative of "OFSC" with respect to r . Can also be evaluated at $(g_2 r_1)$ and $(f_2 r_1)$.
M_OF	M or H	Undetermined constant.

Table A-6. System Matrix Variables

FORTTRAN VARIABLE	TR 11,043 VARIABLE	DESCRIPTION
sm	SR	System matrix for the single solid cylinder and outer fluid.
smrc1	S	System matrix for the double cylinder and outer fluid.

APPENDIX B

INTERMEDIATE VARIABLES FOR NUWC-NPT TR 11,067

Table B-1. Common Block /CYLINDER1/

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
SP_CY1_a1	$J_n(p_1 a)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again for $(p_1 r_1)$.
d1_SP_CY1_a1	$\frac{\partial}{\partial r} J_n(p_1 a)$	First derivative of $J_n(p_1 r)$ with respect to r . Can be evaluated again for $(p_1 r_1)$.
d2_SP_CY1_a1	$\frac{\partial^2}{\partial r^2} J_n(p_1 a)$	Second derivative of $J_n(p_1 r)$ with respect to r . Can be evaluated again for $(p_1 r_1)$.
VXP_CY1_a1	$J_n(q_1 a)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again for $(q_1 r_1)$.
d1_VXP_CY1_a1	$\frac{\partial}{\partial r} J_n(q_1 a)$	First derivative of $J_n(q_1 r)$ with respect to r . Can be evaluated again for $(q_1 r_1)$.
d2_VXP_CY1_a1	$\frac{\partial^2}{\partial r^2} J_n(q_1 a)$	Second derivative of $J_n(q_1 r)$ with respect to r . Can be evaluated again for $(q_1 r_1)$.
VRTP_CY1_a1	$J_{n+1}(q_1 a)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again for $(q_1 r_1)$.
d1_VRTP_CY1_a1	$\frac{\partial}{\partial r} J_{n+1}(q_1 a)$	First derivative of $J_{n+1}(q_1 r)$ with respect to r . Can be evaluated again for $(q_1 r_1)$.
d2_VRTP_CY1_a1	$\frac{\partial^2}{\partial r^2} J_{n+1}(q_1 a)$	Second derivative of $J_{n+1}(q_1 r)$ with respect to r . Can be evaluated again for $(q_1 r_1)$.

Table B-1. Common Block /CYLINDER1/ (Cont'd)

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
SP_CY1_b1	$J_n(p_1b)$	Bessel function of the first kind, used for the scalar displacement potential.
d1_SP_CY1_b1	$\frac{\partial}{\partial r} J_n(p_1b)$	First derivative of $J_n(p_1r)$ with respect to r .
d2_SP_CY1_b1	$\frac{\partial^2}{\partial r^2} J_n(p_1b)$	Second derivative of $J_n(p_1r)$ with respect to r .
VXP_CY1_b1	$J_n(q_1b)$	Bessel function of the first kind, used for the vector x displacement potential.
d1_VXP_CY1_b1	$\frac{\partial}{\partial r} J_n(q_1b)$	First derivative of $J_n(q_1r)$ with respect to r .
d2_VXP_CY1_b1	$\frac{\partial^2}{\partial r^2} J_n(q_1b)$	Second derivative of $J_n(q_1r)$ with respect to r .
VRTP_CY1_b1	$J_{n+1}(q_1b)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$.
d1_VRTP_CY1_b1	$\frac{\partial}{\partial r} J_{n+1}(q_1b)$	First derivative of $J_{n+1}(q_1r)$ with respect to r .
d2_VRTP_CY1_b1	$\frac{\partial^2}{\partial r^2} J_{n+1}(q_1b)$	Second derivative of $J_{n+1}(q_1r)$ with respect to r .
SP_CY1_a2	$Y_n(p_1a)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again for (p_1r_1) .
d1_SP_CY1_a2	$\frac{\partial}{\partial r} Y_n(p_1a)$	First derivative of $Y_n(p_1r)$ with respect to r . Can be evaluated again for (p_1r_1) .

Table B-1. Common Block /CYLINDER1/ (Cont'd)

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
d2_SP_CY1_a2	$\frac{\partial^2}{\partial r^2} Y_n(p_1 a)$	Second derivative of $Y_n(p_1 r)$ with respect to r . Can be evaluated again for $(p_1 r_1)$.
VXP_CY1_a2	$Y_n(q_1 a)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again for $(q_1 r_1)$.
d1_VXP_CY1_a2	$\frac{\partial}{\partial r} Y_n(q_1 a)$	First derivative of $Y_n(q_1 r)$ with respect to r . Can be evaluated again for $(q_1 r_1)$.
d2_VXP_CY1_a2	$\frac{\partial^2}{\partial r^2} Y_n(q_1 a)$	Second derivative of $Y_n(q_1 r)$ with respect to r . Can be evaluated again for $(q_1 r_1)$.
VRTP_CY1_a2	$Y_{n+1}(q_1 a)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again for $(q_1 r_1)$.
d1_VRTP_CY1_a2	$\frac{\partial}{\partial r} Y_{n+1}(q_1 a)$	First derivative of $Y_{n+1}(q_1 r)$ with respect to r . Can be evaluated again for $(q_1 r_1)$.
d2_VRTP_CY1_a2	$\frac{\partial^2}{\partial r^2} Y_{n+1}(q_1 a)$	Second derivative of $Y_{n+1}(q_1 r)$ with respect to r . Can be evaluated again for $(q_1 r_1)$.
SP_CY1_b2	$Y_n(p_1 b)$	Bessel function of the first kind, used for the scalar displacement potential.
d1_SP_CY1_b2	$\frac{\partial}{\partial r} Y_n(p_1 b)$	First derivative of $Y_n(p_1 r)$ with respect to r .
d2_SP_CY1_b2	$\frac{\partial^2}{\partial r^2} Y_n(p_1 b)$	Second derivative of $Y_n(p_1 r)$ with respect to r .
VXP_CY1_b2	$Y_n(q_1 b)$	Bessel function of the first kind, used for the vector x displacement potential.

Table B-1. Common Block /CYLINDER1/ (Cont'd)

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
d1_VXP_CY1_b2	$\frac{\partial}{\partial r} Y_n(q_1 b)$	First derivative of $Y_n(q_1 r)$ with respect to r .
d2_VXP_CY1_b2	$\frac{\partial^2}{\partial r^2} Y_n(q_1 b)$	Second derivative of $Y_n(q_1 r)$ with respect to r .
VRTP_CY1_b2	$Y_{n+1}(q_1 b)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$.
d1_VRTP_CY1_b2	$\frac{\partial}{\partial r} Y_{n+1}(q_1 b)$	First derivative of $Y_{n+1}(q_1 r)$ with respect to r .
d2_VRTP_CY1_b2	$\frac{\partial^2}{\partial r^2} Y_{n+1}(q_1 b)$	Second derivative of $Y_{n+1}(q_1 r)$ with respect to r .
A1_C1	A_1^{C1}	Undetermined constant.
A2_C1	A_2^{C1}	Undetermined constant.
B1_C1	B_1^{C1}	Undetermined constant.
B2_C1	B_2^{C1}	Undetermined constant.
C1_C1	C_1^{C1}	Undetermined constant.
C2_C1	C_2^{C1}	Undetermined constant.
lame_c1	λ_1	Lame constant for cylinder 1.
shear_c1	μ_1	Shear modulus for cylinder 1.

Table B-2. Common Block /CYLINDER2/

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
SP_CY2_b1	$J_n(p_2b)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again for (p_2r_1) .
d1_SP_CY2_b1	$\frac{\partial}{\partial r} J_n(p_2b)$	First derivative of $J_n(p_2r)$ with respect to r . Can be evaluated again for (p_2r_1) .
d2_SP_CY2_b1	$\frac{\partial^2}{\partial r^2} J_n(p_2b)$	Second derivative of $J_n(p_2r)$ with respect to r . Can be evaluated again for (p_2r_1) .
VXP_CY2_b1	$J_n(q_2b)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again for (q_2r_1) .
d1_VXP_CY2_b1	$\frac{\partial}{\partial r} J_n(q_2b)$	First derivative of $J_n(q_2r)$ with respect to r . Can be evaluated again for (q_2r_1) .
d2_VXP_CY2_b1	$\frac{\partial^2}{\partial r^2} J_n(q_2b)$	Second derivative of $J_n(q_2r)$ with respect to r . Can be evaluated again for (q_2r_1) .
VRTP_CY2_b1	$J_{n+1}(q_2b)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again for (q_2r_1) .
d1_VRTP_CY2_b1	$\frac{\partial}{\partial r} J_{n+1}(q_2b)$	First derivative of $J_{n+1}(q_2r)$ with respect to r . Can be evaluated again for (q_2r_1) .
d2_VRTP_CY2_b1	$\frac{\partial^2}{\partial r^2} J_{n+1}(q_2b)$	Second derivative of $J_{n+1}(q_2r)$ with respect to r . Can be evaluated again for (q_2r_1) .
SP_CY2_c1	$J_n(p_2c)$	Bessel function of the first kind, used for the scalar displacement potential.
d1_SP_CY2_c1	$\frac{\partial}{\partial r} J_n(p_2c)$	First derivative of $J_n(p_2r)$ with respect to r .

Table B-2. Common Block /CYLINDER2/ (Cont'd)

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
d2_SP_CY2_c1	$\frac{\partial^2}{\partial r^2} J_n(p_2 c)$	Second derivative of $J_n(p_2 r)$ with respect to r .
VXP_CY2_c1	$J_n(q_2 c)$	Bessel function of the first kind, used for the vector x displacement potential.
d1_VXP_CY2_c1	$\frac{\partial}{\partial r} J_n(q_2 c)$	First derivative of $J_n(q_2 r)$ with respect to r .
d2_VXP_CY2_c1	$\frac{\partial^2}{\partial r^2} J_n(q_2 c)$	Second derivative of $J_n(q_2 r)$ with respect to r .
VRTP_CY2_c1	$J_{n+1}(q_2 c)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$.
d1_VRTP_CY2_c1	$\frac{\partial}{\partial r} J_{n+1}(q_2 c)$	First derivative of $J_{n+1}(q_2 r)$ with respect to r .
d2_VRTP_CY2_c1	$\frac{\partial^2}{\partial r^2} J_{n+1}(q_2 c)$	Second derivative of $J_{n+1}(q_2 r)$ with respect to r .
SP_CY2_b2	$Y_n(p_2 b)$	Bessel function of the first kind, used for the scalar displacement potential. Can be evaluated again for $(p_2 r_1)$.
d1_SP_CY2_b2	$\frac{\partial}{\partial r} Y_n(p_2 b)$	First derivative of $Y_n(p_2 r)$ with respect to r . Can be evaluated again for $(p_2 r_1)$.
d2_SP_CY2_b2	$\frac{\partial^2}{\partial r^2} Y_n(p_2 b)$	Second derivative of $Y_n(p_2 r)$ with respect to r . Can be evaluated again for $(p_2 r_1)$.
VXP_CY2_b2	$Y_n(q_2 b)$	Bessel function of the first kind, used for the vector x displacement potential. Can be evaluated again for $(q_2 r_1)$.

Table B-2. Common Block /CYLINDER2/ (Cont'd)

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
d1_VXP_CY2_b2	$\frac{\partial}{\partial r} Y_n(q_2 b)$	First derivative of $Y_n(q_2 r)$ with respect to r . Can be evaluated again for $(q_2 r_1)$.
d2_VXP_CY2_b2	$\frac{\partial^2}{\partial r^2} Y_n(q_2 b)$	Second derivative of $Y_n(q_2 r)$ with respect to r . Can be evaluated again for $(q_2 r_1)$.
VRTP_CY2_b2	$Y_{n+1}(q_2 b)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$. Can be evaluated again for $(q_2 r_1)$.
d1_VRTP_CY2_b2	$\frac{\partial}{\partial r} Y_{n+1}(q_2 b)$	First derivative of $Y_{n+1}(q_2 r)$ with respect to r . Can be evaluated again for $(q_2 r_1)$.
d2_VRTP_CY2_b2	$\frac{\partial^2}{\partial r^2} Y_{n+1}(q_2 b)$	Second derivative of $Y_{n+1}(q_2 r)$ with respect to r . Can be evaluated again for $(q_2 r_1)$.
SP_CY2_c2	$Y_n(p_2 c)$	Bessel function of the first kind, used for the scalar displacement potential.
d1_SP_CY2_c2	$\frac{\partial}{\partial r} Y_n(p_2 c)$	First derivative of $Y_n(p_2 r)$ with respect to r .
d2_SP_CY2_c2	$\frac{\partial^2}{\partial r^2} Y_n(p_2 c)$	Second derivative of $Y_n(p_2 r)$ with respect to r .
VXP_CY2_c2	$Y_n(q_2 c)$	Bessel function of the first kind, used for the vector x displacement potential.
d1_VXP_CY2_c2	$\frac{\partial}{\partial r} Y_n(q_2 c)$	First derivative of $Y_n(q_2 r)$ with respect to r .
d2_VXP_CY2_c2	$\frac{\partial^2}{\partial r^2} Y_n(q_2 c)$	Second derivative of $Y_n(q_2 r)$ with respect to r .

Table B-2. Common Block /CYLINDER2/ (Cont'd)

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
VRTP_CY2_c2	$Y_{n+1}(q_2c)$	Bessel function of the first kind, used for the vector r and θ displacement potential of order $n + 1$.
d1_VRTP_CY2_c2	$\frac{\partial}{\partial r} Y_{n+1}(q_2c)$	First derivative of $Y_{n+1}(q_2r)$ with respect to r .
d2_VRTP_CY2_c2	$\frac{\partial^2}{\partial r^2} Y_{n+1}(q_2c)$	Second derivative of $Y_{n+1}(q_2r)$ with respect to r .
A1_C2	A_1^{C2}	Undetermined constant.
A2_C2	A_2^{C2}	Undetermined constant.
B1_C2	B_1^{C2}	Undetermined constant.
B2_C2	B_2^{C2}	Undetermined constant.
C1_C2	C_1^{C2}	Undetermined constant.
C2_C2	C_2^{C2}	Undetermined constant.
lame_c2	λ_2	Lame constant for cylinder 2.
shear_c2	μ_2	Shear modulus for cylinder 2.

Table B-3. Common Block /IFLUID/

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
IFSC	$J_n(g_1 a)$	Bessel function of the first kind, used for the scalar displacement potential. Can also be evaluated at $(g_1 r_1)$.
d1_IFSC	$\frac{\partial}{\partial r} J_n(g_1 a)$	First derivative of $J_n(g_1 r)$ with respect to r . Can also be evaluated at $(g_1 r_1)$.
D_IF	D	Undetermined constant.

Table B-4. Common Block /OFLUID/

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
OFSC	$H_n^{(1)}(g_2 b)$ or $K_n(f_2 b)$	Modified Bessel function, used for the scalar displacement potential. Can also be evaluated at $(g_2 r_1)$ and $(f_2 r_1)$.
d1_OFSC	$\frac{\partial}{\partial r} H_n^{(1)}(g_2 b)$ or $\frac{\partial}{\partial r} K_n(f_2 b)$	First derivative of "OFSC" with respect to r . Can also be evaluated at $(g_2 r_1)$ and $(f_2 r_1)$.
M_OF	M or H	Undetermined constant.

Table B-5. Calculated Cylinder 1 Variables

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
Ec_c1	E_1^*	Complex Young's modulus.
bo_c1	b	Outer radius.

Table B-6. Calculated Cylinder 2 Variables

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
Ec_c2	E_2^*	Complex Young's modulus.
co_c2	c	Outer radius.

Table B-7. System Matrix Variables

FORTTRAN VARIABLE	TR 11,067 VARIABLE	DESCRIPTION
smrc1	$smc1$	System matrix for the single cylinder with inner and outer fluids.
smrc2	C	System matrix for the double cylinder with inner and outer fluids.

APPENDIX C

MAKEFILE

```
# Makefile for mr2cf program
# PATH /home/kwcode/analysis/tota/RODCY2
#
#
all:mr2cf
#
mr2cf:mr2cf.o zbessel.o cbessl.o rf.o c1.o c2.o fluids.o smc1.o
    f77 -g -u mr2cf.o zbessel.o cbessl.o rf.o c1.o c2.o fluids.o smc1.o -Nl20 -lm -o mr2cf
#
zbessel:zbessel.o
    f77 -g -u zbessel.f
#
#
cbessl: cbessl.o
    f77 -g -u cbessl.f
#
#
rf: rf.o
    f77 -g -u rf.f
#
#
c1: c1.o
    f77 -g -u c1.f
#
c2: c2.o
    f77 -g -u c2.f
#
fluids: fluids.o
    f77 -g -u fluids.f

smc1:smc1.o
    f77 -g -u smc1.f
#
clean:
    rm -f *.o *%
#
```

APPENDIX D

SUBROUTINE cbessel.f_matlab

```

C *****
C
C      subroutine "cbessel.f_matlab"
C
C This subroutine was written and developed by Mark S. Peloquin
C
C at NUWCDETNLON 6/10/95. As of 6/10/95, there are no known bugs.
C
C Please notify the author if bugs are found (203) 440-5433.
C
C *****
C This subroutine "cbessel.f_matlab" uses the Matlab library for Bessel Functions
C of complex argument z. The source file needed is zbessel.f. The following
C Bessel functions are used from Matlab: J, Y, and K.
C The formula for the first derivative of the Bessel functions is taken from
C Conduction Heat Transfer by Vedat S. Arpaci. They are also listed in
C Field Theory Handbook by Moon and Spencer.
C The formula for the second derivative was derived by Mark S. Peloquin and checked
C by Roy Streit. It is listed as B-9 in the notes.
C This code was written by Mark S. Peloquin 6/10/95 at NUWCDETNLON.

```

```

      function gamma(n)
      integer i,n,sum,gamma
      if(n .eq. 0 .or. n .eq. 1) then
         gamma = 1
      else
         sum=n
         do 10, i=1, n-1
            sum = sum*(n-i)
10      continue
         gamma = sum
      endif
      return
      end

```

```

function fac(n)
  integer n,sum,fac,i
  if(n .eq. 0 .or. n .eq. 1) then
    fac = 1
  else
    sum=n
    do 10, i=1, n-1
      sum = sum*(n-i)
10    continue
    fac = sum
  endif
  return
end

```

```

function psi(n)
  integer n,na,i
  real*8 sum,psi,euler
  parameter (euler=.5772156649015328606)
  sum = 0.0
  na = iabs(n)
  if (na .eq. 1) then
    psi = - euler
  else
    do 10, i=1, na-1
      sum = sum + (1.0/i)
10  continue
    psi = -euler + sum
  endif
  return
end

```

```
double complex function cbessj(n,a,r)
```

```
complex*16 a
```

```
real*8 r
```

```
integer n
```

```
COMPLEX*16 Z,W,W1,W2,WRK
```

```
REAL*8 ZR,ZI,NU,PI,D1MACH
```

```
INTEGER KODE,UNFL,IERR
```

```
DATA PI /3.1415926535897932385D0/
```

```
C
```

```
C Pick up input arguments and allocate output array
```

```
C
```

```
nu = n
```

```
Z = a*r
```

```
KODE = 1
```

```
C
```

```
C J_nu(z)
```

```
C
```

```
IF (NU .GE. 0.0D0) THEN
```

```
CALL CBESJ(Z,NU,KODE,1,W,UNFL,IERR)
```

```
ELSE IF (NU .EQ. DINT(NU)) THEN
```

```
CALL CBESJ(Z,-NU,KODE,1,W,UNFL,IERR)
```

```
IF (MOD(IDINT(NU),2) .NE. 0) W = -W
```

```
ELSE IF (ZR .EQ. 0.0D0 .AND. ZI .EQ. 0.0D0) THEN
```

```
W = -D1MACH(6)
```

```
ELSE
```

```
CALL CBESJ(Z,-NU,KODE,1,W1,UNFL,IERR)
```

```
CALL CBESY(Z,-NU,KODE,1,W2,UNFL,WRK,IERR)
```

```
W = DCOS(NU*PI)*W1 + SIN(NU*PI)*W2
```

```
ENDIF
```

```
cbessj = W
```

```
return
```

```
end
```

double complex function cbessy(n,a,r)

complex*16 a

real*8 r

integer n

COMPLEX*16 Z,W,W1,W2,WRK

REAL*8 ZR,ZI,NU,PI,D1MACH

INTEGER KODE,UNFL,IERR

DATA PI /3.1415926535897932385D0/

C

C Pick up input arguments and allocate output array

C

NU = n

Z = a*r

KODE = 1

ZR = DREAL(Z)

ZI = DIMAG(Z)

C

C Y_nu(z)

C Y(nu,0) = -Infinity

C

IF (ZR .EQ. 0.0D0 .AND. ZI .EQ. 0.0D0) THEN

W = -D1MACH(6)

ELSE IF (NU .GE. 0.0D0) THEN

CALL CBESY(Z,NU,KODE,1,W,UNFL,WRK,IERR)

ELSE IF (NU .EQ. DINT(NU)) THEN

CALL CBESY(Z,-NU,KODE,1,W,UNFL,WRK,IERR)

IF (MOD(DINT(NU),2) .NE. 0) W = -W

ELSE

CALL CBESJ(Z,-NU,KODE,1,W1,UNFL,IERR)

CALL CBESY(Z,-NU,KODE,1,W2,UNFL,WRK,IERR)

W = DCOS(NU*PI)*W2 - SIN(NU*PI)*W1

ENDIF

C

cbessy = W

return

end

- C Argument limit is approximately (3.0,3.0) for cbessi
 double complex function cbessi(n,a,r)

```
integer n,limit,j,k,na,fac
real*8 r
complex*16 a,z,sum,total
double complex cbessi
```

```
z = a*r
na=iabs(n)
limit = 10
total = (0.0,0.0)
```

- ```
do 10, k=0, limit
 j = na + k
 sum = ((.25*z**2)**k)/(fac(k)*fac(j))
 total = total + sum
10 continue
cbessi = ((z/2)**na)*total
```

```
return
end
```

double complex function cbessk(n,a,r)

```
complex*16 a
real*8 r
integer n
```

```
COMPLEX*16 Z,W
REAL*8 ZR,ZI,NU,PI,D1MACH
INTEGER FUN,KODE,UNFL,IERR
DATA PI /3.1415926535897932385D0/
```

- C  
 C Pick up input arguments and allocate output array  
 C

```
NU = n
Z = a*r
KODE = 1
ZR = DREAL(Z)
ZI = DIMAG(Z)
FUN = 75
```

```

C
C K_nu(z)
C K(nu,0) = Infinity
C
 IF (FUN .EQ. 75) THEN
 IF (ZR .EQ. 0.0D0 .AND. ZI .EQ. 0.0D0) THEN
 W = D1MACH(6)
 ELSE
 CALL CBESK(Z,DABS(NU),KODE,1,W,UNFL,IERR)
 ENDIF
 ENDIF
C
 cbessk = W

 return
 end

```

double complex function d1cbessk(n,a,r)

```

 real*8 r
 complex*16 a
 integer n
 double complex cbessk,d1cbessk

 d1cbessk = -a*cbessk(n+1,a,r)+(n/r)*cbessk(n,a,r)

 return
 end

```

double complex function d2cbessk(n,a,r)

```

 real*8 r
 complex*16 a
 integer n
 double complex d2cbessk

```

C Need the proper equation here, not being used 6/10/95.

```

 d2cbessk = (1.0,1.0)

```

```

return
end

```

```

double complex function d1cbessj(n,a,r)

```

```

real*8 r
complex*16 a
integer n
double complex cbessj,d1cbessj

```

```

d1cbessj = -a*cbessj(n+1,a,r)+(n/r)*cbessj(n,a,r)

```

```

return
end

```

```

double complex function d2cbessj(n,a,r)

```

```

real*8 r
complex*16 a
integer n
double complex cbessj,d2cbessj

```

```

d2cbessj = ((a**2)/4.0D0)*(cbessj(n-2,a,r) - 2.0D0*
1 cbessj(n,a,r) + cbessj(n+2,a,r))

```

```

return
end

```

double complex function d1cbessy(n,a,r)

real\*8 r

complex\*16 a

integer n

double complex cbessy,d1cbessy

d1cbessy = -a\*cbessy(n+1,a,r)+(n/r)\*cbessy(n,a,r)

return

end

double complex function d2cbessy(n,a,r)

real\*8 r

complex\*16 a

integer n

double complex cbessy,d2cbessy

d2cbessy = ((a\*\*2)/4.0D0)\*(cbessy(n-2,a,r) - 2.0D0\*  
1       cbessy(n,a,r) + cbessy(n+2,a,r))

return

end

double complex function d1cbessi(n,a,r)

real\*8 r

complex\*16 a

integer n

double complex cbessi,d1cbessi

d1cbessi = a\*cbessi(n+1,a,r)+(n/r)\*cbessi(n,a,r)

return

end

```
double complex function d2cbessi(n,a,r)
```

```
real*8 r
```

```
complex*16 a
```

```
integer n
```

```
double complex d2cbessi
```

C Need the proper equation here, not being used 6/10/95.

```
d2cbessi = (1.0,1.0)
```

```
return
```

```
end
```

```
double complex function cbessh1(n,a,r)
```

```
real*8 r
```

```
complex*16 a
```

```
integer n
```

```
double complex cbessj,cbessy,cbessh1
```

```
cbessh1 = cbessj(n,a,r)+(0.0, 1.0)*cbessy(n,a,r)
```

```
return
```

```
end
```

```
double complex function cbessh2(n,a,r)
```

```
real*8 r
```

```
integer n
```

```
complex*16 a
```

```
double complex cbessj,cbessy,cbessh2
```

```
cbessh2 = cbessj(n,a,r)-(0.0, 1.0)*cbessy(n,a,r)
```

```
return
```

```
end
```

```
double complex function d1cbessh1(n,a,r)
integer n
real*8 r
complex a
double complex d1cbessj,d1cbessy,d1cbessh1

d1cbessh1 = d1cbessj(n,a,r) + (0.0,1.0)*d1cbessy
1(n,a,r)

return
end
```

```
double complex function d1cbessh2(n,a,r)
integer n
real*8 r
complex a
double complex d1cbessj,d1cbessy,d1cbessh2

d1cbessh2 = d1cbessj(n,a,r) - (0.0,1.0)*d1cbessy
1(n,a,r)

return
end
```

## APPENDIX E

## MODIFIED SUBROUTINES FOR CIRCUMFERENTIAL EXCITATION

SUBROUTINE ABC\_ROD\_INVERT(exctype,sm)

## C EXTERNAL VARIABLES

```
integer exctype
complex*16 sm(4,4)
```

```
common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
```

```
complex*16 SP_rod,d1_SP_rod,d2_SP_rod
complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod
```

C \*\*\*\*\*

## C DEFINITIONS FOR COMMON BLOCK /OFLUID/

```
complex*16 OFSC,d1_OFSC,M_OF

common /OFLUID/ OFSC,d1_OFSC,M_OF
```

C \*\*\*\*\*

## C INTERNAL VARIABLES

```
integer n,iflag
complex*16 sminv(4,4),work(4,8)
```

```
n = 4
iflag = 0
```

CALL MINV(sm,sminv,work,n,iflag)

|                   |            |             |
|-------------------|------------|-------------|
| C CIRCUMFERENTIAL | EXCITATION | exctype = 2 |
| C RADIAL          | EXCITATION | exctype = 1 |
| C AXIAL           | EXCITATION | exctype = 0 |

```

if (exctype .eq. 1) then
 A1_rod = -sminv(1,1)
 B1_rod = -sminv(2,1)
 C1_rod = -sminv(3,1)
 M_OF = -sminv(4,1)
elseif (exctype .eq. 0) then
 A1_rod = -sminv(1,2)
 B1_rod = -sminv(2,2)
 C1_rod = -sminv(3,2)
 M_OF = -sminv(4,2)
elseif (exctype .eq. 2) then
 A1_rod = -sminv(1,3)
 B1_rod = -sminv(2,3)
 C1_rod = -sminv(3,3)
 M_OF = -sminv(4,3)
endif

```

```

return
end

```

SUBROUTINE ABC\_C1\_INVERT(n,exctype,smc1,a,b)

# C EXTERNAL VARIABLES

integer n,exctype  
real\*8 a,b  
complex\*16 smc1(7,7)

# C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

complex\*16 SP\_CY1\_a1,d1\_SP\_CY1\_a1,d2\_SP\_CY1\_a1  
complex\*16 SP\_CY1\_a2,d1\_SP\_CY1\_a2,d2\_SP\_CY1\_a2  
complex\*16 SP\_CY1\_b1,d1\_SP\_CY1\_b1,d2\_SP\_CY1\_b1  
complex\*16 SP\_CY1\_b2,d1\_SP\_CY1\_b2,d2\_SP\_CY1\_b2

complex\*16 VXP\_CY1\_a1,d1\_VXP\_CY1\_a1,d2\_VXP\_CY1\_a1  
complex\*16 VXP\_CY1\_a2,d1\_VXP\_CY1\_a2,d2\_VXP\_CY1\_a2  
complex\*16 VXP\_CY1\_b1,d1\_VXP\_CY1\_b1,d2\_VXP\_CY1\_b1  
complex\*16 VXP\_CY1\_b2,d1\_VXP\_CY1\_b2,d2\_VXP\_CY1\_b2

complex\*16 VRTP\_CY1\_a1,d1\_VRTP\_CY1\_a1,d2\_VRTP\_CY1\_a1  
complex\*16 VRTP\_CY1\_a2,d1\_VRTP\_CY1\_a2,d2\_VRTP\_CY1\_a2  
complex\*16 VRTP\_CY1\_b1,d1\_VRTP\_CY1\_b1,d2\_VRTP\_CY1\_b1  
complex\*16 VRTP\_CY1\_b2,d1\_VRTP\_CY1\_b2,d2\_VRTP\_CY1\_b2

complex\*16 lame\_c1,shear\_c1,cl\_c1,ct\_c1  
complex\*16 A1\_C1,A2\_C1,B1\_C1,B2\_C1,C1\_C1,C2\_C1

common /CYLINDER1/ SP\_CY1\_a1,d1\_SP\_CY1\_a1,d2\_SP\_CY1\_a1,  
1 SP\_CY1\_a2,d1\_SP\_CY1\_a2,d2\_SP\_CY1\_a2,  
1 SP\_CY1\_b1,d1\_SP\_CY1\_b1,d2\_SP\_CY1\_b1,  
1 SP\_CY1\_b2,d1\_SP\_CY1\_b2,d2\_SP\_CY1\_b2,  
1 VXP\_CY1\_a1,d1\_VXP\_CY1\_a1,d2\_VXP\_CY1\_a1,  
1 VXP\_CY1\_a2,d1\_VXP\_CY1\_a2,d2\_VXP\_CY1\_a2,  
1 VXP\_CY1\_b1,d1\_VXP\_CY1\_b1,d2\_VXP\_CY1\_b1,  
1 VXP\_CY1\_b2,d1\_VXP\_CY1\_b2,d2\_VXP\_CY1\_b2,  
1 VRTP\_CY1\_a1,d1\_VRTP\_CY1\_a1,d2\_VRTP\_CY1\_a1,  
1 VRTP\_CY1\_a2,d1\_VRTP\_CY1\_a2,d2\_VRTP\_CY1\_a2,  
1 VRTP\_CY1\_b1,d1\_VRTP\_CY1\_b1,d2\_VRTP\_CY1\_b1,  
1 VRTP\_CY1\_b2,d1\_VRTP\_CY1\_b2,d2\_VRTP\_CY1\_b2,  
1 lame\_c1,shear\_c1,cl\_c1,ct\_c1,  
1 A1\_C1,A2\_C1,B1\_C1,B2\_C1,C1\_C1,C2\_C1

C \*\*\*\*\*

C \*\*\*\*\*

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

complex\*16 OFSC,d1\_OFSC,M\_OF

common /OFLUID/ OFSC,d1\_OFSC,M\_OF

C \*\*\*\*\*

C \*\*\*\*\*

C DEFINITIONS FOR COMMON BLOCK /IFLUID/

common /IFLUID/ IFSC,d1\_IFSC,D\_IF

complex\*16 IFSC,d1\_IFSC,D\_IF

C \*\*\*\*\*

C INTERNAL VARIABLES

integer size,iflag

complex\*16 smc1inv(7,7),workc1(7,14)

size = 7

iflag = 0

CALL MINV(smc1,smc1inv,workc1,size,iflag)

|                   |            |             |
|-------------------|------------|-------------|
| C CIRCUMFERENTIAL | EXCITATION | exctype = 2 |
| C RADIAL          | EXCITATION | exctype = 1 |
| C AXIAL           | EXCITATION | exctype = 0 |

```

if (exctype .eq. 1) then
 A1_C1 = -smc1inv(1,1)
 A2_C1 = -smc1inv(2,1)
 B1_C1 = -smc1inv(3,1)
 B2_C1 = -smc1inv(4,1)
 C1_C1 = -smc1inv(5,1)
 C2_C1 = -smc1inv(6,1)
 M_OF = (A1_C1*d1_SP_CY1_b1 +
1 A2_C1*d1_SP_CY1_b2 +
1 B1_C1*n/b*VXP_CY1_b1 +
1 B2_C1*n/b*VXP_CY1_b2 +
1 C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b1 +
1 C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b2)/d1_OFSC
 D_IF = -smc1inv(7,1)
elseif (exctype .eq. 0) then
 A1_C1 = -smc1inv(1,2)
 A2_C1 = -smc1inv(2,2)
 B1_C1 = -smc1inv(3,2)
 B2_C1 = -smc1inv(4,2)
 C1_C1 = -smc1inv(5,2)
 C2_C1 = -smc1inv(6,2)
 M_OF = (A1_C1*d1_SP_CY1_b1 +
1 A2_C1*d1_SP_CY1_b2 +
1 B1_C1*n/b*VXP_CY1_b1 +
1 B2_C1*n/b*VXP_CY1_b2 +
1 C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b1 +
1 C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b2)/d1_OFSC
 D_IF = -smc1inv(7,2)
elseif (exctype .eq. 2) then
 A1_C1 = -smc1inv(1,3)
 A2_C1 = -smc1inv(2,3)
 B1_C1 = -smc1inv(3,3)
 B2_C1 = -smc1inv(4,3)
 C1_C1 = -smc1inv(5,3)
 C2_C1 = -smc1inv(6,3)
 M_OF = (A1_C1*d1_SP_CY1_b1 +
1 A2_C1*d1_SP_CY1_b2 +
1 B1_C1*n/b*VXP_CY1_b1 +
1 B2_C1*n/b*VXP_CY1_b2 +
1 C1_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b1 +
1 C2_C1*(0.0D0,1.0D0)*k*VRTP_CY1_b2)/d1_OFSC
 D_IF = -smc1inv(7,3)

```

endif

return  
end

SUBROUTINE ABC\_RC1\_INVERT(n,exctype,smrc1,b)

# C EXTERNAL VARIABLES

integer n,exctype  
real\*8 b  
complex\*16 smrc1(10,10)

# C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

complex\*16 SP\_CY1\_a1,d1\_SP\_CY1\_a1,d2\_SP\_CY1\_a1  
complex\*16 SP\_CY1\_a2,d1\_SP\_CY1\_a2,d2\_SP\_CY1\_a2  
complex\*16 SP\_CY1\_b1,d1\_SP\_CY1\_b1,d2\_SP\_CY1\_b1  
complex\*16 SP\_CY1\_b2,d1\_SP\_CY1\_b2,d2\_SP\_CY1\_b2

complex\*16 VXP\_CY1\_a1,d1\_VXP\_CY1\_a1,d2\_VXP\_CY1\_a1  
complex\*16 VXP\_CY1\_a2,d1\_VXP\_CY1\_a2,d2\_VXP\_CY1\_a2  
complex\*16 VXP\_CY1\_b1,d1\_VXP\_CY1\_b1,d2\_VXP\_CY1\_b1  
complex\*16 VXP\_CY1\_b2,d1\_VXP\_CY1\_b2,d2\_VXP\_CY1\_b2

complex\*16 VRTP\_CY1\_a1,d1\_VRTP\_CY1\_a1,d2\_VRTP\_CY1\_a1  
complex\*16 VRTP\_CY1\_a2,d1\_VRTP\_CY1\_a2,d2\_VRTP\_CY1\_a2  
complex\*16 VRTP\_CY1\_b1,d1\_VRTP\_CY1\_b1,d2\_VRTP\_CY1\_b1  
complex\*16 VRTP\_CY1\_b2,d1\_VRTP\_CY1\_b2,d2\_VRTP\_CY1\_b2

complex\*16 lame\_c1,shear\_c1,cl\_c1,ct\_c1  
complex\*16 A1\_C1,A2\_C1,B1\_C1,B2\_C1,C1\_C1,C2\_C1

```

common /CYLINDER1/ SP_CY1_a1,d1_SP_CY1_a1,d2_SP_CY1_a1,
1 SP_CY1_a2,d1_SP_CY1_a2,d2_SP_CY1_a2,
1 SP_CY1_b1,d1_SP_CY1_b1,d2_SP_CY1_b1,
1 SP_CY1_b2,d1_SP_CY1_b2,d2_SP_CY1_b2,
1 VXP_CY1_a1,d1_VXP_CY1_a1,d2_VXP_CY1_a1,
1 VXP_CY1_a2,d1_VXP_CY1_a2,d2_VXP_CY1_a2,
1 VXP_CY1_b1,d1_VXP_CY1_b1,d2_VXP_CY1_b1,
1 VXP_CY1_b2,d1_VXP_CY1_b2,d2_VXP_CY1_b2,
1 VRTP_CY1_a1,d1_VRTP_CY1_a1,d2_VRTP_CY1_a1,
1 VRTP_CY1_a2,d1_VRTP_CY1_a2,d2_VRTP_CY1_a2,
1 VRTP_CY1_b1,d1_VRTP_CY1_b1,d2_VRTP_CY1_b1,
1 VRTP_CY1_b2,d1_VRTP_CY1_b2,d2_VRTP_CY1_b2,
1 lame_c1,shear_c1,cl_c1,ct_c1,
1 A1_C1,A2_C1,B1_C1,B2_C1,C1_C1,C2_C1

```

C \*\*\*\*\*

## C DEFINITIONS FOR COMMON BLOCK /ROD/

```

common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
1,A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

```

```

complex*16 SP_rod,d1_SP_rod,d2_SP_rod
complex*16 VXP_rod,d1_VXP_rod,d2_VXP_rod
complex*16 VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
complex*16 A1_rod,B1_rod,C1_rod,lame_rod,shear_rod

```

C \*\*\*\*\*

## C DEFINITIONS FOR COMMON BLOCK /OFLUID/

```

complex*16 OFSC,d1_OFSC,M_OF

```

```

common /OFLUID/ OFSC,d1_OFSC,M_OF

```

C \*\*\*\*\*

## C INTERNAL VARIABLES

```

integer size,iflag
complex*16 smrc1inv(10,10),workrc1(10,20)

```

```

size = 10
iflag = 0

```

```

CALL MINV(smrc1,smrc1inv,workrc1,size,iflag)

```

|                   |                        |
|-------------------|------------------------|
| C CIRCUMFERENTIAL | EXCITATION exctype = 2 |
| C RADIAL          | EXCITATION exctype = 1 |
| C AXIAL           | EXCITATION exctype = 0 |

```

if (exctype .eq. 1) then
 A1_C1 = -smrc1inv(1,1)
 A2_C1 = -smrc1inv(2,1)
 B1_C1 = -smrc1inv(3,1)
 B2_C1 = -smrc1inv(4,1)
 C1_C1 = -smrc1inv(5,1)
 C2_C1 = -smrc1inv(6,1)
 A1_rod = -smrc1inv(7,1)
 B1_rod = -smrc1inv(8,1)
 C1_rod = -smrc1inv(9,1)
 M_OF = -smrc1inv(10,1)
elseif (exctype .eq. 0) then
 A1_C1 = -smrc1inv(1,2)
 A2_C1 = -smrc1inv(2,2)
 B1_C1 = -smrc1inv(3,2)
 B2_C1 = -smrc1inv(4,2)
 C1_C1 = -smrc1inv(5,2)
 C2_C1 = -smrc1inv(6,2)
 A1_rod = -smrc1inv(7,2)
 B1_rod = -smrc1inv(8,2)
 C1_rod = -smrc1inv(9,2)
 M_OF = -smrc1inv(10,2)
elseif (exctype .eq. 2) then
 A1_C1 = -smrc1inv(1,3)
 A2_C1 = -smrc1inv(2,3)
 B1_C1 = -smrc1inv(3,3)
 B2_C1 = -smrc1inv(4,3)
 C1_C1 = -smrc1inv(5,3)
 C2_C1 = -smrc1inv(6,3)
 A1_rod = -smrc1inv(7,3)
 B1_rod = -smrc1inv(8,3)
 C1_rod = -smrc1inv(9,3)
 M_OF = -smrc1inv(10,3)
endif

```

```

return
end

```

SUBROUTINE ABC\_RC2\_INVERT(n,exctype,smrc2,a,c)

# C EXTERNAL VARIABLES

integer n,exctype  
real\*8 a,c  
complex\*16 smrc2(13,13)

# C DEFINITIONS FOR COMMON BLOCK /CYLINDER1/

complex\*16 SP\_CY1\_a1,d1\_SP\_CY1\_a1,d2\_SP\_CY1\_a1  
complex\*16 SP\_CY1\_a2,d1\_SP\_CY1\_a2,d2\_SP\_CY1\_a2  
complex\*16 SP\_CY1\_b1,d1\_SP\_CY1\_b1,d2\_SP\_CY1\_b1  
complex\*16 SP\_CY1\_b2,d1\_SP\_CY1\_b2,d2\_SP\_CY1\_b2

complex\*16 VXP\_CY1\_a1,d1\_VXP\_CY1\_a1,d2\_VXP\_CY1\_a1  
complex\*16 VXP\_CY1\_a2,d1\_VXP\_CY1\_a2,d2\_VXP\_CY1\_a2  
complex\*16 VXP\_CY1\_b1,d1\_VXP\_CY1\_b1,d2\_VXP\_CY1\_b1  
complex\*16 VXP\_CY1\_b2,d1\_VXP\_CY1\_b2,d2\_VXP\_CY1\_b2

complex\*16 VRTP\_CY1\_a1,d1\_VRTP\_CY1\_a1,d2\_VRTP\_CY1\_a1  
complex\*16 VRTP\_CY1\_a2,d1\_VRTP\_CY1\_a2,d2\_VRTP\_CY1\_a2  
complex\*16 VRTP\_CY1\_b1,d1\_VRTP\_CY1\_b1,d2\_VRTP\_CY1\_b1  
complex\*16 VRTP\_CY1\_b2,d1\_VRTP\_CY1\_b2,d2\_VRTP\_CY1\_b2

complex\*16 lame\_c1,shear\_c1,cl\_c1,ct\_c1  
complex\*16 A1\_C1,A2\_C1,B1\_C1,B2\_C1,C1\_C1,C2\_C1

common /CYLINDER1/ SP\_CY1\_a1,d1\_SP\_CY1\_a1,d2\_SP\_CY1\_a1,  
1 SP\_CY1\_a2,d1\_SP\_CY1\_a2,d2\_SP\_CY1\_a2,  
1 SP\_CY1\_b1,d1\_SP\_CY1\_b1,d2\_SP\_CY1\_b1,  
1 SP\_CY1\_b2,d1\_SP\_CY1\_b2,d2\_SP\_CY1\_b2,  
1 VXP\_CY1\_a1,d1\_VXP\_CY1\_a1,d2\_VXP\_CY1\_a1,  
1 VXP\_CY1\_a2,d1\_VXP\_CY1\_a2,d2\_VXP\_CY1\_a2,  
1 VXP\_CY1\_b1,d1\_VXP\_CY1\_b1,d2\_VXP\_CY1\_b1,  
1 VXP\_CY1\_b2,d1\_VXP\_CY1\_b2,d2\_VXP\_CY1\_b2,  
1 VRTP\_CY1\_a1,d1\_VRTP\_CY1\_a1,d2\_VRTP\_CY1\_a1,  
1 VRTP\_CY1\_a2,d1\_VRTP\_CY1\_a2,d2\_VRTP\_CY1\_a2,  
1 VRTP\_CY1\_b1,d1\_VRTP\_CY1\_b1,d2\_VRTP\_CY1\_b1,  
1 VRTP\_CY1\_b2,d1\_VRTP\_CY1\_b2,d2\_VRTP\_CY1\_b2,  
1 lame\_c1,shear\_c1,cl\_c1,ct\_c1,  
1 A1\_C1,A2\_C1,B1\_C1,B2\_C1,C1\_C1,C2\_C1

C \*\*\*\*\*

## C DEFINITIONS FOR COMMON BLOCK /CYLINDER2/

```
complex*16 SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1
complex*16 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2
complex*16 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1
complex*16 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2
```

```
complex*16 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1
complex*16 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2
complex*16 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1
complex*16 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2
```

```
complex*16 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1
complex*16 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2
complex*16 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1
complex*16 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2
```

```
complex*16 lame_c2,shear_c2,cl_c2,ct_c2
```

```
complex*16 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2
```

```
common /CYLINDER2/ SP_CY2_b1,d1_SP_CY2_b1,d2_SP_CY2_b1,
1 SP_CY2_b2,d1_SP_CY2_b2,d2_SP_CY2_b2,
1 SP_CY2_c1,d1_SP_CY2_c1,d2_SP_CY2_c1,
1 SP_CY2_c2,d1_SP_CY2_c2,d2_SP_CY2_c2,
1 VXP_CY2_b1,d1_VXP_CY2_b1,d2_VXP_CY2_b1,
1 VXP_CY2_b2,d1_VXP_CY2_b2,d2_VXP_CY2_b2,
1 VXP_CY2_c1,d1_VXP_CY2_c1,d2_VXP_CY2_c1,
1 VXP_CY2_c2,d1_VXP_CY2_c2,d2_VXP_CY2_c2,
1 VRTP_CY2_b1,d1_VRTP_CY2_b1,d2_VRTP_CY2_b1,
1 VRTP_CY2_b2,d1_VRTP_CY2_b2,d2_VRTP_CY2_b2,
1 VRTP_CY2_c1,d1_VRTP_CY2_c1,d2_VRTP_CY2_c1,
1 VRTP_CY2_c2,d1_VRTP_CY2_c2,d2_VRTP_CY2_c2,
1 lame_c2,shear_c2,cl_c2,ct_c2,
1 A1_C2,A2_C2,B1_C2,B2_C2,C1_C2,C2_C2
```

C \*\*\*\*\*

## C DEFINITIONS FOR COMMON BLOCK /ROD/

```
common /ROD/ SP_rod,d1_SP_rod,d2_SP_rod,VXP_rod,
1d1_VXP_rod,d2_VXP_rod,VRTP_rod,d1_VRTP_rod,d2_VRTP_rod
```

1,A1\_rod,B1\_rod,C1\_rod,lame\_rod,shear\_rod

complex\*16 SP\_rod,d1\_SP\_rod,d2\_SP\_rod  
 complex\*16 VXP\_rod,d1\_VXP\_rod,d2\_VXP\_rod  
 complex\*16 VRTP\_rod,d1\_VRTP\_rod,d2\_VRTP\_rod  
 complex\*16 A1\_rod,B1\_rod,C1\_rod,lame\_rod,shear\_rod

C \*\*\*\*\*

C DEFINITIONS FOR COMMON BLOCK /OFLUID/

complex\*16 OFSC,d1\_OFSC,M\_OF

common /OFLUID/ OFSC,d1\_OFSC,M\_OF

C \*\*\*\*\*

C \*\*\*\*\*

C DEFINITIONS FOR COMMON BLOCK /IFLUID/

common /IFLUID/ IFSC,d1\_IFSC,D\_IF

complex\*16 IFSC,d1\_IFSC,D\_IF

C \*\*\*\*\*

C INTERNAL VARIABLES

integer size,iflag  
 complex\*16 smrc2inv(13,13),workrc2(13,26)

size = 13  
 iflag = 0

CALL MINV(smrc2,smrc2inv,workrc2,size,iflag)

|                   |                        |
|-------------------|------------------------|
| C CIRCUMFERENTIAL | EXCITATION exctype = 2 |
| C RADIAL          | EXCITATION exctype = 1 |
| C AXIAL           | EXCITATION exctype = 0 |

if (exctype .eq. 1) then

```

 A1_C2 = -smrc2inv(1,1)
 A2_C2 = -smrc2inv(2,1)
 B1_C2 = -smrc2inv(3,1)
 B2_C2 = -smrc2inv(4,1)
 C1_C2 = -smrc2inv(5,1)
 C2_C2 = -smrc2inv(6,1)
 A1_C1 = -smrc2inv(7,1)
 A2_C1 = -smrc2inv(8,1)
 B1_C1 = -smrc2inv(9,1)
 B2_C1 = -smrc2inv(10,1)
 C1_C1 = -smrc2inv(11,1)
 C2_C1 = -smrc2inv(12,1)
 D_IF = -smrc2inv(13,1)
 M_OF = (A1_C2*d1_SP_CY2_c1 +
1 A2_C2*d1_SP_CY2_c2 +
1 B1_C2*n/c*VXP_CY2_c1 +
1 B2_C2*n/c*VXP_CY2_c2 +
1 C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c1 +
1 C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c2)/d1_OFSC

```

elseif (exctype .eq. 0) then

```

 A1_C2 = -smrc2inv(1,2)
 A2_C2 = -smrc2inv(2,2)
 B1_C2 = -smrc2inv(3,2)
 B2_C2 = -smrc2inv(4,2)
 C1_C2 = -smrc2inv(5,2)
 C2_C2 = -smrc2inv(6,2)
 A1_C1 = -smrc2inv(7,2)
 A2_C1 = -smrc2inv(8,2)
 B1_C1 = -smrc2inv(9,2)
 B2_C1 = -smrc2inv(10,2)
 C1_C1 = -smrc2inv(11,2)
 C2_C1 = -smrc2inv(12,2)
 D_IF = -smrc2inv(13,2)
 M_OF = (A1_C2*d1_SP_CY2_c1 +
1 A2_C2*d1_SP_CY2_c2 +
1 B1_C2*n/c*VXP_CY2_c1 +
1 B2_C2*n/c*VXP_CY2_c2 +
1 C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c1 +
1 C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c2)/d1_OFSC

```

```

elseif (exctype .eq. 2) then
 A1_C2 = -smrc2inv(1,3)
 A2_C2 = -smrc2inv(2,3)
 B1_C2 = -smrc2inv(3,3)
 B2_C2 = -smrc2inv(4,3)
 C1_C2 = -smrc2inv(5,3)
 C2_C2 = -smrc2inv(6,3)
 A1_C1 = -smrc2inv(7,3)
 A2_C1 = -smrc2inv(8,3)
 B1_C1 = -smrc2inv(9,3)
 B2_C1 = -smrc2inv(10,3)
 C1_C1 = -smrc2inv(11,3)
 C2_C1 = -smrc2inv(12,3)
 D_IF = -smrc2inv(13,3)
 M_OF = (A1_C2*d1_SP_CY2_c1 +
1 A2_C2*d1_SP_CY2_c2 +
1 B1_C2*n/c*VXP_CY2_c1 +
1 B2_C2*n/c*VXP_CY2_c2 +
1 C1_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c1 +
1 C2_C2*(0.0D0,1.0D0)*k*VRTP_CY2_c2)/d1_OFSC

endif

return
end

```

**INITIAL DISTRIBUTION LIST**

| <b>Addressee</b>                                                          | <b>No. of Copies</b> |
|---------------------------------------------------------------------------|----------------------|
| Defense Technical Information Center                                      | 12                   |
| Office of Naval Research (Code 321SS: K. Dial, S. Littlefield, R. Varley) | 3                    |
| Naval Research Laboratory (A. Dandridge, S. Vohra)                        | 2                    |
| Cambridge Acoustical Associates (J. Garrelick, J. Cole III)               | 2                    |
| Applied Measurement Systems, Inc., New London (J. Diggs)                  | 2                    |