TECHNICAL DOCUMENT 3180
April 2004

CHIEF 2004 Users Manual

G. W. Benthien
D. Barach
S. L. Hobbs

SSC San Diego
San Diego, CA 92152-5001

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
00 APR 2004	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
CHIEF 2004 Users Manual	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **SSC San Diego San Diego, CA 92152-5001**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM201682, Report No. SSC/SD-TD 2980, Handbook of Acoustic Projector Technology Revision 2. , The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**129**	

SSC SAN DIEGO
San Diego, California 92152-5001

T. V. Flynn, CAPT, USN
Commanding Officer

R. F. Smith
Executive Director

ADMINISTRATIVE INFORMATION

This report was prepared for the Office of Naval Research, Sensors, Sources, and Arrays (ONR 321) by the Maritime Surveillance Division (Code 271) SSC San Diego.

Released by
S. M. Adams, Head
Active Acoustics Branch

Under authority of
G. L. Davis, Head
Maritime Surveillance
Division

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This user guide explains how to use the CHIEF2004 computer program. CHIEF2004 is an update of the CHIEF2000 program [Benthien, Barach, and Hobbs, 2000]. The program was tested on machines running Windows 95/98®, Windows NT®, and COMPACT UNIX. The CHIEF program was originally developed in the 1960s to compute the acoustic radiation from an arbitrary shaped radiating body. The program was extensively updated in 1988 and a users manual was published [Benthien, Barach, and Gillette, 1988]. This CHIEF manual is still an excellent reference for those interested in the details of the CHIEF program and the underlying theory. The original CHIEF manual was included in the CD-ROM document [Benthien and Barach, 1998]. The CHIEF2000 manual was published as an Adobe Acrobat document [Benthien,Barach, and Hobbs, 2000].

The original CHIEF program and the 1988 update were essentially a library of Fortran subroutines that were called from a user-generated control routine. This approach gave the user a lot of flexibility, but required a Fortran compiler to compile and link the control routine with the library routines. The presence of a Fortran compiler was not a big problem on most main frame computers. However, with the increasing use of PCs and workstations for numerical computation, the presence of a Fortran compiler can no longer be assumed. Thus, both CHIEF 2000 and CHIEF2004 are compiled executable programs that rely on a user-generated ASCII text file for input. While this approach gives up some flexibility, we have included as many features of the previous versions of CHIEF as possible.

This user guide describes the setup of the user-generated input file and provides a number of examples. Section 2 describes the additions and changes to CHIEF2000 introduced by CHIEF2004. Section 3 describes the use of the program and the general organization of the input file. Sections 4–6 describe the main subdivisions of the input file. Section 4 describes the parameter section in which the user can define parameters for use in the remaining sections of the input file. Section 5 describes the geometry input section in which the user specifies the radiating surface and its subdivision. Section 6 describes the control section in which the user specifies the quantities to be computed and the form of the output. Section 7 describes the format of the various possible output files. Section 8 contains several example problems along with the corresponding input files.

Much of the material in this user-guide is reference material; therefore this document is not designed to be read from start to finish as one would read a novel. The authors feel that the program can be best learned by studying the example problems near the end of the manual. A suggested reading plan is given below:

1. Read Introduction.
2. Read Changes from CHIEF2000
3. Read General Organization section.
4. Read Parameter section.

5. Read introductory portions of the Geometry and Control sections.
6. Work through example problems.
7. Refer to the command descriptions in the Geometry and the Control sections as needed.

2. CHANGES FROM CHIEF2000

Although CHIEF2004 and CHIEF2000 have the same basic function, there have been a number of additions and changes introduced since the release of CHIEF2000. Some of the changes are transparent to the user and some require user interaction. I will attempt to outline below the major changes and what improvements they are intended to provide.

1. CHIEF2000 relies extensively on disk I/O. For large problems a significant amount of time can be spent in moving large matrices in and out of RAM memory. CHIEF2004 attempts to minimize this transfer of data by keeping as much data as possible in core at any given time. A large portion of the disk I/O was involved with combining the symmetry blocks of the CHIEF A,B-matrices. In the new storage scheme a predetermined number of rows of all the symmetry blocks are kept in RAM at any one time. In this way the symmetry combinations of the blocks can be performed for the rows in core. The number of rows present in RAM at any time is determined based on the amount of RAM available. If all the rows of all the blocks will fit in core, then the whole process is accomplished in core. The matrix area in RAM is also used to hold the combined blocks during the solution procedure. In this case there is only one symmetry block of the matrix in core at any given time. It is hoped that this revised storage scheme will significantly speed up larger CHIEF problems.

2. The revised storage scheme required almost a complete recoding of CHIEF. In the process the code was made more modular. This should make it easier to make future changes.

3. In CHIEF2004 virtually all of the array dimensioning is done dynamically in the subroutines where the arrays are used. Therefore, there is very little wasted storage space and it is not necessary to redimension and recompile CHIEF in order to handle larger problems.

4. The linear equation solver in CHIEF2004 now employs routines from LAPACK. For overdetermined systems a QR algorithm with column pivoting is used. For square systems an LU algorithm is used. In both cases a condition number estimate is obtained. The maximum condition number at each frequency is printed to the screen and to the output file so that frequencies where nonuniqueness occurs can be detected. The LU algorithm is about four times faster than the QR algorithm. In CHIEF2000 a QR algorithm was used in all cases.

5. In CHIEF2004 the interior points used in overdetermining the equations near frequencies where the solution is nonunique can be determined automatically. The points are chosen along the inner normal to the surface at the center of randomly selected CHIEF subdivisions. The distance off the surface is chosen based on the size of the corresponding subdivision. We want to avoid having all interior points lie on a nodal surface of an interior mode since such points do not help the nonuniqueness problem. Our strategy is to keep the points fairly close to the boundary surface (which also is a

nodal surface), but not so close that there are problems with the singular integrands in the matrix coefficients. We use a distance from the surface equal to the square root of the corresponding subdivision area. After a potential interior point is chosen it is tested to make sure that it lies interior to the surface. The test uses a numerical approximation to the integral relation

$$-\int_S \frac{\partial}{\partial n_\xi}\left(\frac{1}{|x-\xi|}\right) dS(\xi) = \begin{cases} 0 & x \text{ exterior to } S \\ 2\pi & x \text{ on } S \\ 4\pi & x \text{ interior to } S. \end{cases}$$

6. In the point source check the program now automatically determines if the point source is interior to the surface or exterior to the surface using the integral relation above.

7. CHIEF2004 now contains triangular elements as well as quadrilateral elements. These elements can use either linear or quadratic interpolation.

8. The user can now specify which planes are to be used as planes of symmetry. In CHIEF2000 a single plane of symmetry had to be the $x_1 = 0$ plane and two planes of symmetry had to be the $x_1 = 0$, $x_2 = 0$ planes.

9. There is now an additional symmetry option that allows for a symmetry plane $x_3 = 0$ along with rotational symmetry.

10. In addition to the point source check there is now also a plane wave check. In this check the actual surface is considered to be transparent. The surface velocity distribution is determined from the normal derivative of the incident plane wave. This velocity distribution is used along with the incident pressure field to form the right-hand-side of the CHIEF equations. The CHIEF equations are solved for the surface pressures that are then compared with the incident plane wave pressures.

11. The user can now use mathematical expressions in any argument of the commands contained in the input file that are looking for a real or integer value. These expressions can involve any previously defined parameters. In CHIEF2000 mathematical expressions were restricted to the parameter section.

12. The first record of binary output files (.zmx, .pmf, .pmn, .abs, .abf, .abn) now contains the lowest frequency, the frequency increment, and the number of frequencies used in the frequency sweep. This allows faster search of these files for information associated with a desired frequency.

13. The user can now specify a surface acceleration distribution or the normal derivative of pressure in place of a velocity distribution.

14. The program can now generate a coupling matrix that can be used in the coupling of the finite element equations to the CHIEF equations. The user specifies a list of nodal coordinates on the surface and and a list of surface elements specifying the surface nodes belonging to each element. The program outputs a coupling matrix relating

the finite element force degrees of freedom to the CHIEF surface pressures and a list relating the finite element node numbers to the sequential node numbers (1,2,...) used by CHIEF. It should be emphasized that CHIEF does not solve the coupled problem. It only provides certain matrices that can be used by a knowledgeable user in solving the coupled problem.

15. The output file (.out) now contains the time spent in various subroutines as well as the RAM memory usage of the larger arrays.

16. Frequency independent quantities such as quadrature points, surface field points, surface normals, far-field points, etc. are now computed only once and stored in arrays.

17. An executable program convert.exe is now included with CHIEF. This program can convert all or part of a CHIEF binary output file to an ASCII data file. The user can specify which frequencies and which symmetry blocks are desired.

This page intentionally left blank

3. GENERAL ORGANIZATION

The program can be run by typing the command

```
chief <input file name>.inp
```

at the command prompt. In Windows 95/98® or Windows NT®, the user should go to an MS-DOS® window to enter the command. The input file name can be any legal DOS file name (maximum of eight characters with no spaces). The .inp extension can be omitted in the command line. If the run completes normally, the message **Normal Termination** will appear on the screen.

The input file (`<input file name>.inp`) is an ASCII text file generated by the user. It must not contain any control characters. The file is divided into the following three major sections:

1. Parameter definition
2. Geometry specification
3. Control logic

In the parameter section, the user can assign parameter names to values required for input in the following sections. The parameter definitions can involve arithmetic expressions as well as previously defined parameters. It is not necessary to define any parameters since all required input values can be either parameters or numerical values. The parameter section is provided for the user's convenience. In the geometry section, the user specifies the radiating surface, the subdivision of the surface, and any symmetry conditions. In the control logic section, the user can specify the quantities to be computed, provide necessary input data, and specify the type of output. A sample input file is shown below.

```
! Parameter section

nblks=8
rotlim=pi/nblks
radius=1.2e-1    ! radius of sphere
freql=100
frequ=200
freqinc=10
end*parameters

! Geometry section

rotational,nblks
sphere,radius
0,pi,-rotlim,rotlim
12,1,4,16
3,0,0,0
end*geometry

! Control logic section

frequency,freql,frequ,freqinc
fluid,995,1489
rhs symmetry
velocities,sample.vel
surface pressures,1
far-field,1,3
0,90,5,0,0,0
```

A detailed explanation of each input-file section will be given later. Note that the parameter definition section must be terminated with a statement beginning with `end*p` and the geometry section must be terminated with a statement beginning with `end*g`. The control logic section is terminated by an end of file. Blank lines are ignored in the input file, as are any comments. Comments must begin with an exclamation point (!). All commands and parameters are case insensitive. Thus, `NBLKS` is the same parameter as `nblks`, and `Sphere` is the same command as `sphere`. It is also important that consistent units be used for all input quantities. The program by default takes the fluid density and sound speed to be 1000 Kg/m^3 and 1500 m/sec. If the user desires to use a system of units other than mks, then these quantities must be changed to values appropriate for that system of units. Data read in from files must also be in the chosen consistent set of units.

Unlike the original version of CHIEF, this version has virtually no predetermined size limits. Most arrays used by the program are allocated dynamically within the program based on inputs in the input file. The only major exception is the limit of 64 on the number of Gaussian quadrature points in each direction. This limit is imposed by the method used to calculate the Gaussian roots and weights.

In addition to any user-specified output files, the program always creates a file with the name <`input file name`>`.out` that contains an echo of all input file lines, error diagnostics, and user-specified output. In the parameter and geometry sections, the program finds as many input errors as it can before stopping. However, the program does not go on to the next section if there are any errors found in the current section. In the control section, the program terminates if any input error is found. Termination helps prevent calculations from being performed with inconsistent or missing input.

The program can generate many output files. These files all have the same root file name as the input file. Table 2 lists the extensions of these output files along with a brief description of their contents.

Table 1. Possible output files.

File Extension	File Type	File Contents
.sps	ASCII	Surface pressures on each subdivision
.ffp	ASCII	Far-field angles and pressures
.nfp	ASCII	Near-field coordinates and pressures
.zmx	Binary	Radiation impedance matrix
.pmf	Binary	Far-field pattern matrix
.pmn	Binary	Near-field pattern matrix
.abs	Binary	A and B surface matrices
.abf	Binary	A and B far-field matrices
.abn	Binary	A and B near-field matrices

4. PARAMETER SECTION

In the parameter section, the user can define parameters that can be used elsewhere in the CHIEF ASCII input file in place of numerical inputs. Only a single parameter may be defined on each line of the parameter section, and the parameter definition must have the form

```
<parameter name>=<expression>.
```

The parameter name may be any sequence of alpha-numeric characters or underscore characters (_) subject to the following restrictions:

- The first character must be a letter.
- The length of the parameter name can not exceed 24 characters.
- The parameter name cannot contain any spaces.

Case is not important in parameter names (e.g., `Nblks` is the same as `nblks`).

All parameter values are interpreted by the program as real numbers. However, the nearest integer is used in places where an integer input is expected. The user may insert comments on a separate line or at the end of a line by preceding the comment with an exclamation point (!).

The expression on the right-hand side is subject to the following conditions:

- The four arithmetic operations (+, −, *, /) and exponentiation (^) are allowed.
- The mathematical functions sin, cos, tan, sqrt, abs, ln, log, and exp are allowed (e.g., sin(x^2) or sqrt(x+y)).
- The operands can be numerical values or previously defined parameters.
- Parenthetical grouping (...) and nested operations may be used.
- An exponent may be any real number.
- Scientific notation is allowed (i.e., 1.6e-6).
- The entire line must not exceed 132 characters.

Expressions satisfying the above conditions are also allowed in the geometry and the control sections wherever a numeric input is expected.

Spaces may be freely used in parameter definitions except within parameter names. The program ignores them. There is one parameter, `pi`, that has been predefined by the program and may be used anywhere in the input file, including the definition of other parameters. The usual rules for precedence of operations apply, for example,

- Operators of equal precedence are evaluated in the usual left to right order.
- Exponentiation has precedence over addition, subtraction, multiplication, and division.
- Multiplication and division have precedence over addition and subtraction.
- Unary plus and minus operations (i.e., -1.0 or +5.6) have the same precedence as exponentiation.

The parameter section must end with a line whose first five nonblank characters are `end*p`. The following is an example of a valid parameter section.

```
rho=1000
c=1500.

diam       =   1.5e2      ! This is a comment
height     =   4.67
length1    =   ((height*.5)^2+(diam/2)^2)^.5
length2    =   length1^(1/2)

theta_upper=pi/4
costu=cos(theta_upper)
theta_lower=-theta_upper

freq_upper=3e3
freq_lower=freq_upper*1e-1

near_field_r=length1*c/freq_lower
near_field_theta=pi*5E-1

end*parameters
```

The following example contains some invalid parameter definitions.

```
rho fluid=1000                        (space in name)
c(1)=1500                           (parenthesis in name)
height=4.67  width=2.0      (more than one parameter on line)
length1=((height*.5)^2+(diam/2)^2)^.5
```

```
  (parameter used on the right side before its definition)
1diam=1.5,     (name begins with a numeral and comma illegal)
inner*radius=.5                (* illegal symbol in name)
outer-radius=2.5               (- illegal symbol in name)
end*param
```

A line specifying the end of the parameter section (i.e., `end*params`) must be included even if no parameters are defined.

This page intentionally left blank

5. GEOMETRY SECTION

CHIEF calculates radiation from one or more closed surfaces in 3-space, or from one or more planar regions in an infinite rigid baffle. In this section of the manual, we describe how the user specifies the geometry of the radiating surface. If the surface has one, two, or three planes of reflective symmetry or a finite order of rotational symmetry, then the user only needs to input the portion of the surface corresponding to one symmetry block.

The commands that specify the surface geometry may be given in any order, except the command specifying the symmetry type (NONE, REFLECTIVE, ROTATIONAL) must be the first command in the geometry section. If there are no underscore characters in the command word, the program uses only the first three characters. If there is an underscore character, the program uses the first letter, the underscore character, and the first character after the underscore to specify the command. Other forms of the command words are perfectly valid as long as the three characters specifying the command are the same.

Each command line consists of a command or a command followed by one or more arguments. Some command lines must be followed by additional lines of data. Commands, arguments, and data items are separated by commas. Parameters defined in the parameter section and expressions involving these parameters may be used for any numerical input in this section. As always, blank lines, comments, and spaces are ignored.

CHIEF has many predefined surfaces such as spherical, cylindrical, ellipsoidal, etc. The radiating surface can be defined in terms of portions of several of these predefined surfaces. CHIEF also allows the surface to be defined in terms of nodes and elements as in a finite element problem. The user must supply inputs specifying the various nonoverlapping surface regions that make up the total radiating surface. In the case of finite element type input, each element is considered as a separate surface region. Each surface region is defined by a one-to-one smooth mapping from a rectangular planar region into 3-dimensional space. The points on the surface as well as any field points are defined in terms of a global rectangular Cartesian coordinate system, x_1, x_2, x_3. Points in the base rectangular region are defined in terms a pair of coordinates, u, v. The values u and v are sometimes called surface coordinates. Each surface region is defined by three relations of the form:

$$x_1 = x_1(u, v) \qquad x_2 = x_2(u, v) \qquad x_3 = x_3(u, v).$$

Each predefined geometry is actually defined in terms of a local x, y, z Cartesian coordinate system that is related to the global coordinate system by a translation and/or a cyclic permutation of the coordinate axes. The translation is specified by a vector, $\vec{\boldsymbol{t}}$, from the global origin to the local origin. The cyclic permutation of the coordinate axes is specified by giving the axis (I_z) in the global system that corresponds to the z-axis in the local system. The user input for this specification is the integer I_z. Figures 1–3 show the relation between the global and local coordinate systems for $I_z = 1, 2, 3$. Note that $I_z = 3$ corresponds to no rotation at all.

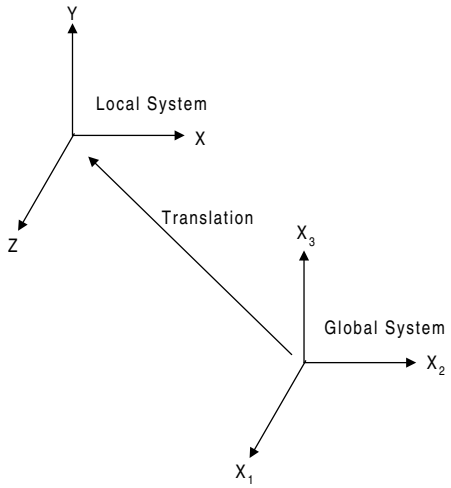

Figure 1. Global and local coordinate systems for $I_z = 1$.

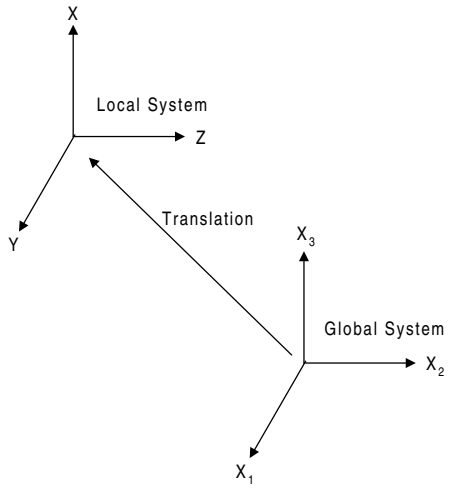

Figure 2. Global and local coordinate systems for $I_z = 2$.

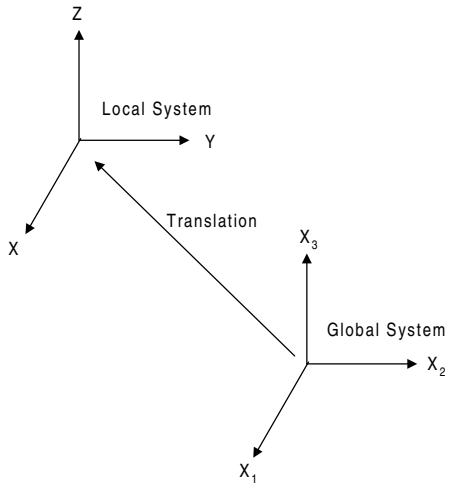

Figure 3. Global and local coordinate systems for $I_z = 3$.

Thus, each surface region is defined by relations of the form:

$$x = x(u, v) \qquad y = y(u, v) \qquad z = z(u, v),$$

and the specification of I_z and $\vec{t}$.

In addition to specifying a relation between the local and global coordinate systems, the input quantity I_z is also used to help specify the outward normal vector to the surface region. From the defining equations for a surface region, CHIEF can compute a normal vector at each point of the region, but it cannot determine whether this normal vector is inward or outward. The user specifies this by the sign of I_z. A plus sign accepts the choice made by the CHIEF program and a minus sign reverses the direction of the surface normal on this surface region. For example, in the case of a circular planar surface region, the program computes the normal pointing in the positive z-direction. This direction may or may not be the correct choice for the surface region under consideration. If the radiating surface was like a tin can with axis of rotation along the z-axis, then the outward normal would be in the positive z-direction on the top, but in the negative z-direction on the bottom. Thus, the user would specify a negative I_z for the bottom surface. The user can also specify a breakup of each surface region into subareas (on which the pressure and normal velocity will be assumed constant) by specifying a uniform breakup in the uv rectangular region. Conventional wisdom says that all subdivisions should be less than 1/4 wavelength with something on the order of 1/8 wavelength desirable. As far as integration over the subdivisions, four Gaussian quadrature points in 1/8 wavelength seems to give sufficient accuracy.

The first statement in the geometry section must be one of the CHIEF symmetry options (`ROTATIONAL`, `REFLECTIVE`, or `NONE`) along with any associated parameters. Following the symmetry specification, the surface regions may be defined in any order. However, the ordering will influence the numbering of the subdivisions. The numbering begins with the first region specified and continues consecutively with the other regions in the order specified. Within a region, the numbering starts with the subarea having the smallest u, v values. The numbering continues through the region in the direction of increasing u and v, with the variation in v occurring before variation in u. Figure 4 shows an example of the numbering for three adjacent planar surface regions.

The numbering proceeds sequentially through the various symmetry blocks and the numbering within each symmetry follows the same pattern as in the first block. By default, the symmetry plane for one plane of symmetry is $x_1 = 0$, and the symmetry planes for two planes of symmetry are $x_1 = 0$ and $x_2 = 0$. Other choices for symmetry planes can be specified. The numbering of symmetry blocks for reflective symmetry (default planes) is shown in tables 2 through 4. The first block is always taken to be in the half-plane, quadrant or octant containing the octant with all positive coordinates. For one plane of symmetry, the second block is in the half-space not containing the all-positive octant. For two planes of symmetry, the quadrants are numbered counterclockwise around the axis formed by the intersection of the two planes. For three planes of symmetry, the default numbering of blocks is always used.

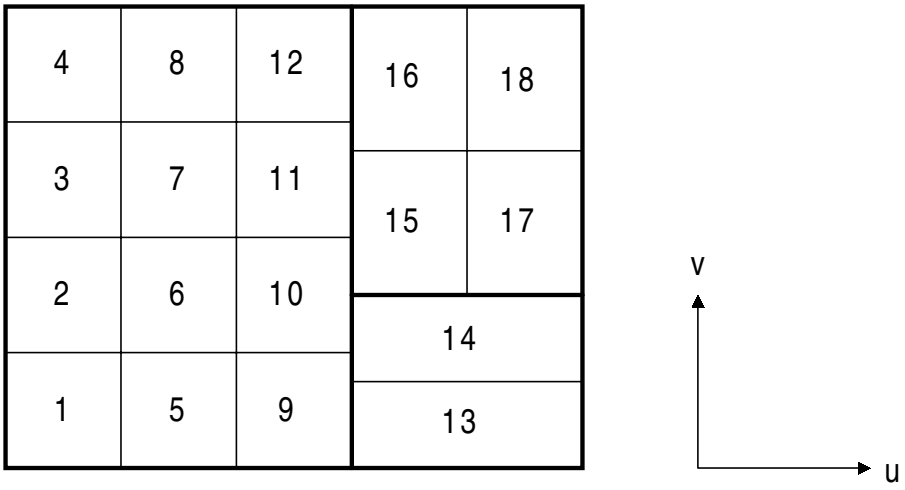

Figure 4. An example of the subdivision numbering scheme.

Table 2. One plane of symmetry ($x_1 = 0$).

Block	**Octants**		
	x_1	x_2	x_3
1	+	+	+
	+	-	+
	+	-	-
	+	+	-
2	-	+	+
	-	-	+
	-	-	-
	-	+	-

Table 3. Two planes of symmetry ($x_1 = 0$ and $x_2 = 0$).

Block	**Octants**		
	x_1	x_2	x_3
1	+	+	+
	+	+	-
2	-	+	+
	-	+	-
3	-	-	+
	-	-	-
4	+	-	+
	+	-	-

Table 4. Three planes of symmetry ($x_1 = 0$, $x_2 = 0$, and $x_3 = 0$).

Block	**Octants**		
	x_1	x_2	x_3
1	+	+	+
2	-	+	+
3	-	-	+
4	+	-	+
5	+	-	-
6	-	-	-
7	-	+	-
8	+	+	-

The geometry section ends with a statement like `end*geometry`, where only the first part, `end*g`, is significant to the program. Commands specifying the surface geometry are described on the following pages. The syntax for each command is given along with a description of the parameters and data items that accompany the command. For commands that do not contain an underline character, the program only looks at the first three letters. For commands that contain an underline character, the program looks at the first character, the underline character, and the first character after the underline character. Alternate forms of the commands are allowed as long as the three significant characters are in the right place. Optional parameters are enclosed in brackets. For those commands that define a surface region, the mapping from the u, v rectangular region to the surface is described. The user needs to understand what u and v are in order to understand the numbering of the surface subdivisions.

reflective symmetry

specifies planes of reflective symmetry

Command: **reflective symmetry**, *ip*

ip Integer specifying planes of symmetry. It consists of three digits each of which is 0 or 1. A one in the first position means that $x_1 = 0$ is a plane of symmetry; a one in the second position means that $x_2 = 0$ is a plane of symmetry; a one in the third position means that $x_3 = 0$ is a plane of symmetry. For example, $ip = 011$ specifies $x_2 = 0$ and $x_3 = 0$ as planes of symmetry. Three planes of symmetry is specified by $ip = 111$.

rotational symmetry

specifies finite order rotational symmetry

Command: **rotational symmetry**, N [, I_{ref}]

N	Specifies N-fold rotational symmetry
I_{ref}	Has the value 1 if $z = 0$ is a plane of symmetry. The default is $I_{\text{ref}} = 0$

none

specifies no symmetry

Command: none

rectangular_planar

rectangular region on surface parallel to the x, y plane

Command: **rectangular_planar**, z_0
u_L, u_U, v_L, v_U
N_u, N_v, Q_u, Q_v
I_z, $[t_1, t_2, t_3]$

z_0	Constant value of z in local coordinate system
u_L, u_U	Lower and upper limits for the surface coordinate u
v_L, v_U	Lower and upper limits for the surface coordinate v
N_u, N_v	Number of subdivisions in the u and v directions in the planar rectangular region
Q_u, Q_v	Number of Gaussian quadrature points per subarea in the u and v directions
I_z	Specifies rotation from local to global coordinate system and orientation of normal
t_1, t_2, t_3	Components of the translation vector from the global origin to the local origin (default is $t_1 = t_2 = t_3 = 0$)

Surface definition:

The local coordinate system must be set up so that the z-axis is normal to the plane of the surface. In this case u is equal to x and v is equal to y. The local coordinates and normal vector are related to the u, v coordinates by

$$\begin{aligned} x &= u & n_x &= 0 \\ y &= v & n_y &= 0 \\ z &= z_0 & n_z &= 1 \end{aligned}$$

circular_planar

circular region on surface parallel to the x, y plane

Command: **circular_planar**, z_0
u_L, u_U, v_L, v_U
N_u, N_v, Q_u, Q_v
I_z, $[t_1, t_2, t_3]$

z_0	Constant value of z in local coordinate system
u_L, u_U	Lower and upper limits for the surface coordinate u
v_L, v_U	Lower and upper limits for the surface coordinate v
N_u, N_v	Number of subdivisions in the u and v directions in the planar rectangular region
Q_u, Q_v	Number of Gaussian quadrature points per subarea in the u and v directions
I_z	Specifies rotation from local to global coordinate system and orientation of normal
t_1, t_2, t_3	Components of the translation vector from the global origin to the local origin (default is $t_1 = t_2 = t_3 = 0$)

Surface definition:

The local coordinate system must be set up so that the z-axis is normal to the plane of the surface. In this case u is the radial distance from the z-axis and v is the angle from the x-axis in the xy plane. The local coordinates and normal vector are related to the u, v coordinates by

$$\begin{aligned} x &= u\cos v & n_x &= 0 \\ y &= u\sin v & n_y &= 0 \\ z &= z_0 & n_z &= 1 \end{aligned}$$

elliptical_planar

elliptical region on surface parallel to the x, y plane

Command: **elliptical_planar**, a, b, z_0
u_L, u_U, v_L, v_U
N_u, N_v, Q_u, Q_v
I_z, $[t_1, t_2, t_3]$

a	Half the length of the major axis of the ellipse
b	Half the length of the minor axis of the ellipse
z_0	Constant value of z in local coordinate system
u_L, u_U	Lower and upper limits for the surface coordinate u
v_L, v_U	Lower and upper limits for the surface coordinate v
N_u, N_v	Number of subdivisions in the u and v directions in the planar rectangular region
Q_u, Q_v	Number of Gaussian quadrature points per subarea in the u and v directions
I_z	Specifies rotation from local to global coordinate system and orientation of normal
t_1, t_2, t_3	Components of the translation vector from the global origin to the local origin (default is $t_1 = t_2 = t_3 = 0$)

Surface definition:

The local coordinate system must be set up so that the z-axis is normal to the plane of the surface. In this case $u = r/f$ where r is the average of the radial distances of a point from the two foci and f is half the interfocal distance. The range of u is $1 \leq u \leq a/\sqrt{a^2 - b^2}$. The family of curves for which u is a constant consists of ellipses having the same foci. The ellipse corresponding to $u = 1$ is the degenerate ellipse consisting of the interval [-f,f] on the x-axis. The ellipse corresponding to $u = a/\sqrt{a^2 - b^2}$ is the boundary ellipse with major diameter of $2a$ and minor diameter of $2b$. Thus, the upper limit u_U of the variable u is almost always $a/\sqrt{a^2 - b^2}$. On an ellipse of constant u the variable v is related to the angle θ from the x-axis in the xy plane by

$$v = \tan^{-1}\left(\frac{u}{\sqrt{u^2 - 1}}\tan\theta\right).$$

The local coordinates and normal vector are related to the u, v coordinates by

$$\begin{aligned} x &= fu\cos v & n_x &= 0, \\ y &= f\sqrt{u^2-1}\sin v & n_y &= 0, \\ z &= z_0 & n_z &= 1, \end{aligned}$$

where $f = \sqrt{a^2 - b^2}$.

Equal increments in v do not produce equal increments in θ and vise versa. However, quadrants in the xy-plane do come out the same, as can be seen from the following table:

v	x	y	θ
0	fu	0	0
$\pi/2$	0	$f\sqrt{u^2-1}$	$\pi/2$
π	$-fu$	0	π
$3\pi/2$	0	$-f\sqrt{u^2-1}$	$3\pi/2$

circular_cylinder

cylindrical surface with axis along z-axis

Command: **circular_cylinder**, a
u_L, u_U, v_L, v_U
N_u, N_v, Q_u, Q_v
I_z, $[t_1, t_2, t_3]$

a	radius of cylinder
u_L, u_U	Lower and upper limits for the surface coordinate u
v_L, v_U	Lower and upper limits for the surface coordinate v
N_u, N_v	Number of subdivisions in the u and v directions in the planar rectangular region
Q_u, Q_v	Number of Gaussian quadrature points per subarea in the u and v directions
I_z	Specifies rotation from local to global coordinate system and orientation of normal
t_1, t_2, t_3	Components of the translation vector from the global origin to the local origin (default is $t_1 = t_2 = t_3 = 0$)

Surface definition:

The local coordinate system must be set up so that the z-axis coincides with the axis of the cylinder. In this case, $u = z$ and v is the angle from the x-axis in the xy plane. The local coordinates and normal vector are related to the u, v coordinates by

$$x = a\cos v \qquad n_x = \cos v$$
$$y = a\sin v \qquad n_y = \sin v$$
$$z = u \qquad n_z = 0$$

elliptic_cylinder

elliptic cylindrical surface with axis along z-axis

Command: **elliptic_cylinder**, a, b
u_L, u_U, v_L, v_U
N_u, N_v, Q_u, Q_v
I_z, $[t_1, t_2, t_3]$

a	Half the major diameter
b	Half the minor diameter
u_L, u_U	Lower and upper limits for the surface coordinate u
v_L, v_U	Lower and upper limits for the surface coordinate v
N_u, N_v	Number of subdivisions in the u and v directions in the planar rectangular region
Q_u, Q_v	Number of Gaussian quadrature points per subarea in the u and v directions
I_z	Specifies rotation from local to global coordinate system and orientation of normal
t_1, t_2, t_3	Components of the translation vector from the global origin to the local origin (default is $t_1 = t_2 = t_3 = 0$)

Surface definition:

The local coordinate system must be set up so that the axis of the cylinder coincides with the z-axis. In this case $u = z$ and v is related to the angle θ from the x-axis in the xy plane by

$$v = \tan^{-1}\left(\frac{a}{b}\tan\theta\right).$$

The local coordinates and normal vector are related to the u, v coordinates by

$$x = a\cos v \qquad n_x = \frac{b\cos v}{\sqrt{a^2 - (a^2 - b^2)\cos^2 v}}$$

$$y = b\sin v \qquad n_y = \frac{a\sin v}{\sqrt{a^2 - (a^2 - b^2)\cos^2 v}}$$

$$z = u \qquad n_z = 0$$

Equal increments in v are not equal increments in θ, and vise versa. However, quadrants in the xy-plane do come out the same as shown in the following table:

v	x	y	θ
0	a	0	0
$\pi/2$	0	b	$\pi/2$
π	$-a$	0	π
$3\pi/2$	0	$-b$	$3\pi/2$

sphere

spherical surface

Command: **sphere**, a
u_L, u_U, v_L, v_U
N_u, N_v, Q_u, Q_v
I_z, $[t_1, t_2, t_3]$

a	Radius of sphere
u_L, u_U	Lower and upper limits for the surface coordinate u
v_L, v_U	Lower and upper limits for the surface coordinate v
N_u, N_v	Number of subdivisions in the u and v directions in the planar rectangular region
Q_u, Q_v	Number of Gaussian quadrature points per subarea in the u and v directions
I_z	Specifies rotation from local to global coordinate system and orientation of normal
t_1, t_2, t_3	Components of the translation vector from the global origin to the local origin (default is $t_1 = t_2 = t_3 = 0$)

Surface definition:

The local coordinate system must be set up so that the polar axis coincides with the z-axis. In this case u is the angle from the z-axis and v is the angle from the x-axis in the xy plane. The local coordinates and normal vector are related to the u, v coordinates by

$$x = a \sin u \cos v \qquad n_x = \sin u \cos v$$
$$y = a \sin u \sin v \qquad n_y = \sin u \sin v$$
$$z = a \cos u \qquad n_z = \cos u$$

oblate_spheroid

oblate spheroidal surface obtained by rotating ellipse around minor axis

---

Command: **oblate_spheroid**, a, b
u_L, u_U, v_L, v_U
N_u, N_v, Q_u, Q_v
I_z, $[t_1, t_2, t_3]$

a	Half the major diameter of elliptical cross section
b	Half the minor diameter of elliptical cross section
u_L, u_U	Lower and upper limits for the surface coordinate u
v_L, v_U	Lower and upper limits for the surface coordinate v
N_u, N_v	Number of subdivisions in the u and v directions in the planar rectangular region
Q_u, Q_v	Number of Gaussian quadrature points per subarea in the u and v directions
I_z	Specifies rotation from local to global coordinate system and orientation of normal
t_1, t_2, t_3	Components of the translation vector from the global origin to the local origin (default is $t_1 = t_2 = t_3 = 0$)

Surface definition:

The local coordinate system must be set up so that the axis of rotation coincides with the z-axis. In this case v is the angle from the x-axis in the xy plane. u is related to the angle θ from the z-axis by

$$u = \tan^{-1}\left(\frac{b}{a}\tan\theta\right).$$

The local coordinates and normal vector are related to the u, v coordinates by

$$\begin{aligned} x &= a\sin u\cos v & n_x &= \frac{b\sin u\cos v}{\sqrt{a^2-(a^2-b^2)\cos^2 v}} \\ y &= a\sin u\sin v & n_y &= \frac{b\sin u\sin v}{\sqrt{b^2-(a^2-b^2)\cos^2 u}} \\ z &= b\cos u & n_z &= \frac{a\cos u}{\sqrt{b^2-(a^2-b^2)\cos^2 u}} \end{aligned}$$

Equal increments in u do not produce equal increments in θ, and vise versa. However, quadrants do come out the same as shown in the following table:

u	x	y	z	θ
0	0	0	b	0
$\pi/2$	$a\cos v$	$a\sin v$	0	$\pi/2$
π	0	0	$-b$	π

prolate_spheroid

prolate spheroidal surface obtained by rotating ellipse around major axis

---

Command: **prolate_spheroid**, a, b
u_L, u_U, v_L, v_U
N_u, N_v, Q_u, Q_v
I_z, $[t_1, t_2, t_3]$

a	Half the major diameter of elliptical cross section
b	Half the minor diameter of elliptical cross section
u_L, u_U	Lower and upper limits for the surface coordinate u
v_L, v_U	Lower and upper limits for the surface coordinate v
N_u, N_v	Number of subdivisions in the u and v directions in the planar rectangular region
Q_u, Q_v	Number of Gaussian quadrature points per subarea in the u and v directions
I_z	Specifies rotation from local to global coordinate system and orientation of normal
t_1, t_2, t_3	Components of the translation vector from the global origin to the local origin (default is $t_1 = t_2 = t_3 = 0$)

Surface definition:

The local coordinate system must be set up so that the axis of rotation coincides with the z-axis. In this case v is the angle from the x-axis in the xy plane and u is related to the angle θ from the z-axis by

$$u = \tan^{-1}\left(\frac{a}{b}\tan\theta\right).$$

The local coordinates and normal vector are related to the u, v coordinates by

$$\begin{aligned} x &= b\sin u\cos v & n_x &= \frac{a\sin u\cos v}{\sqrt{a^2-(a^2-b^2)\cos^2 v}} \\ y &= b\sin u\sin v & n_y &= \frac{a\sin u\sin v}{\sqrt{b^2-(a^2-b^2)\cos^2 u}} \\ z &= a\cos u & n_z &= \frac{b\cos u}{\sqrt{b^2-(a^2-b^2)\cos^2 u}} \end{aligned}$$

Equal increments in u are not equal increments in θ, and vise versa. However, quadrants do come out the same as shown in the following table:

u	x	y	z	θ
0	0	0	a	0
$\pi/2$	$b\cos v$	$b\sin v$	0	$\pi/2$
π	0	0	$-a$	π

fe_axisymmetric

curve generated by finite element interpolation functions is rotated around z-axis

---

Command: **fe_axisymmetric**, file.crd, file.elt, Q_u, Q_v
I_z, [t_1, t_2, t_3]

file.crd	File containing nodal coordinates. This file is an ASCII file containing three columns of data with one or more spaces between columns. The first column contains nodal numbers, the second column contains the radial r coordinates of the nodes, and the third column contains the axial z coordinates of the nodes. The nodes lie along the generating curve for the surface. This curve is rotated around the z-axis to generate the surface.
file.elt	File containing element node numbers. This file is an ASCII file that contains two columns if linear interpolation is used and three columns if quadratic interpolation is used. The columns are separated by one or more spaces. Each line of data contains the node numbers for an element. If there are two columns, the node numbers are for the first and last node in the element. If there are three columns, then the columns correspond to first node, middle node, and last node respectively.
Q_u, Q_v	Number of Gaussian quadrature points in the u and v directions
I_z	Specifies rotation from local to global coordinate system and orientation of normal
t_1, t_2, t_3	Components of the translation vector from the global origin to the local origin (default is $t_1 = t_2 = t_3 = 0$)

Surface definition:

The local coordinate system is set up with the z-axis being the axis of rotation. Each finite element corresponds to one CHIEF subdivision of the surface. The surface coordinate v is the angle from the x-axis in the xy-plane. The surface coordinate u is in the interval [-1,1]. The local coordinates and normal vector are related to the u, v coordinates by

$$\begin{aligned} x &= r(u)\cos v & \qquad n_x &= n_r(u)\cos v \\ y &= r(u)\sin v & \qquad n_y &= n_r(u)\sin v \\ z &= z(u) & \qquad n_z &= n_z(u) \end{aligned}$$

where

$$\begin{pmatrix} r(u) \\ z(u) \end{pmatrix} = \sum_{n=1}^{2,3} \phi_n(u) \begin{pmatrix} r_n \\ z_n \end{pmatrix}$$

and

$$\begin{pmatrix} t_r(u) \\ t_z(u) \end{pmatrix} = \sum_{n=1}^{2,3} \phi_n'(u) \begin{pmatrix} r_n \\ z_n \end{pmatrix} \qquad \text{tangent vector}$$

$$\begin{pmatrix} n_r(u) \\ n_z(u) \end{pmatrix} = \begin{pmatrix} -t_z(u)/\sqrt{t_r^2(u)+t_z^2(u)} \\ t_r(u)/\sqrt{t_r^2(u)+t_z^2(u)} \end{pmatrix}.$$

r_n, z_n	given coordinates of nodes belonging to element
$\phi_n(u)$	finite element interpolation function (linear or quadratic)

For linear interpolation, the interpolation functions ϕ_n are given by

$$\begin{aligned} \phi_1(u) &= \tfrac{1}{2}(1-u) \\ \phi_2(u) &= \tfrac{1}{2}(1+u). \end{aligned}$$

For quadratic interpolation,

$$\begin{aligned} \phi_1(u) &= \tfrac{1}{2}u(u-1) \\ \phi_2(u) &= 1-u^2 \\ \phi_3(u) &= \tfrac{1}{2}u(u+1). \end{aligned}$$

The normal has been chosen so that it points to the left as you travel along the cross-section curve.

fe_3d

surface generated by finite element interpolation functions

Command: **fe_3d**, file.crd, file.elt, Q_u, Q_v
I_z, [t_1, t_2, t_3]

file.crd — File containing nodal coordinates. This file is an ASCII file containing four columns of data with one or more spaces between columns. The first column contains nodal numbers and the second through fourth columns contain the x, y, z coordinates of the nodes.

file.elt — File containing element node numbers. This file is an ASCII file that contains four columns if linear interpolation on a rectangular patch is used and eight columns if quadratic interpolation on a rectangular patch is used. It contains three columns if linear interpolation on a triangular patch is used and six columns if quadratic interpolation on a triangular patch is used. The columns are separated by one or more spaces. Each line of data contains the node numbers for an element. If there are three or four columns, the node numbers are for the corner nodes of the element taken in a counterclockwise direction around the element. If there are eight (six) columns, then the first four (three)columns are the same as for the linear case and the last four (three) columns are for the mid nodes taken in a counterclockwise direction beginning with the node between the first and second nodes.

Q_u, Q_v — Number of Gaussian quadrature points in the u and v directions

I_z — Specifies rotation from local to global coordinate system and orientation of normal

t_1, t_2, t_3 — Components of the translation vector from the global origin to the local origin (default is $t_1 = t_2 = t_3 = 0$)

Surface definition:

Each finite element corresponds to one CHIEF subdivision of the surface. Figure 5 shows the base rectangle for a rectangular surface patch.

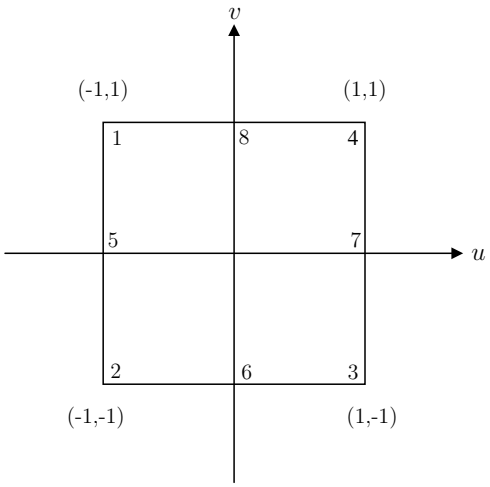

Figure 5. The base rectangle in the uv coordinate system.

The local coordinates and normal vector are related to the u, v coordinates by

$$\begin{aligned} x &= x(u,v) & n_x &= \nu_x(u,v)/\sqrt{\nu_x(u,v)^2+\nu_y(u,v)^2+\nu_z(u,v)^2} \\ y &= y(u,v) & n_y &= \nu_y(u,v)/\sqrt{\nu_x(u,v)^2+\nu_y(u,v)^2+\nu_z(u,v)^2} \\ z &= z(u,v) & n_z &= \nu_z(u,v)/\sqrt{\nu_x(u,v)^2+\nu_y(u,v)^2+\nu_z(u,v)^2}, \end{aligned}$$

where

$$\begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix} = \sum_{n=1}^{4,8} \phi_n(u,v) \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix}$$

and

$$\begin{pmatrix} t_x^u(u,v) \\ t_y^u(u,v) \\ t_z^u(u,v) \end{pmatrix} = \sum_{n=1}^{4,8} \frac{\partial \phi_n(u,v)}{\partial u} \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} \qquad u \text{ tangent vector}$$

$$\begin{pmatrix} t_x^v(u,v) \\ t_y^v(u,v) \\ t_z^v(u,v) \end{pmatrix} = \sum_{n=1}^{4,8} \frac{\partial \phi_n(u,v)}{\partial v} \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} \qquad v \text{ tangent vector}$$

$$\begin{pmatrix} \nu_x(u,v) \\ \nu_y(u,v) \\ \nu_z(u,v) \end{pmatrix} = \begin{pmatrix} t_y^u(u,v)t_z^v(u,v) - t_z^u(u,v)t_y^v(u,v) \\ t_z^u(u,v)t_x^v(u,v) - t_x^u(u,v)t_z^v(u,v) \\ t_x^u(u,v)t_y^v(u,v) - t_y^u(u,v)t_x^v(u,v) \end{pmatrix}$$

x_n, y_n, z_n	given coordinates of nodes belonging to element
$\phi_n(u,v)$	finite element interpolation function (linear or quadratic)

For linear interpolation on a rectangular element, the interpolation functions ϕ_n are given by

$$\begin{aligned} \phi_1(u,v) &= \tfrac{1}{4}(1-u)(1+v) \\ \phi_2(u,v) &= \tfrac{1}{4}(1-u)(1-v) \\ \phi_3(u,v) &= \tfrac{1}{4}(1+u)(1-v) \\ \phi_4(u,v) &= \tfrac{1}{4}(1+u)(1+v). \end{aligned}$$

For quadratic interpolation on a rectangular element,

$$\begin{aligned} \phi_1(u,v) &= \tfrac{1}{4}(1-u)(1+v)(v-u-1) \\ \phi_2(u,v) &= \tfrac{1}{4}(1-u)(1-v)(-v-u-1) \\ \phi_3(u,v) &= \tfrac{1}{4}(1+u)(1-v)(-v+u-1) \\ \phi_4(u,v) &= \tfrac{1}{4}(1+u)(1+v)(v+u-1) \\ \phi_5(u,v) &= \tfrac{1}{2}(1-u)(1+v)(1-v) \\ \phi_6(u,v) &= \tfrac{1}{2}(1-u)(1-v)(1+u) \\ \phi_7(u,v) &= \tfrac{1}{2}(1+u)(1-v)(1+v) \\ \phi_8(u,v) &= \tfrac{1}{2}(1+u)(1+v)(1-u). \end{aligned}$$

The normal was chosen so that the nodes appear in counterclockwise order when viewed from the positive normal side of the surface.

For triangular patches, the base triangle is shown in figure 6.

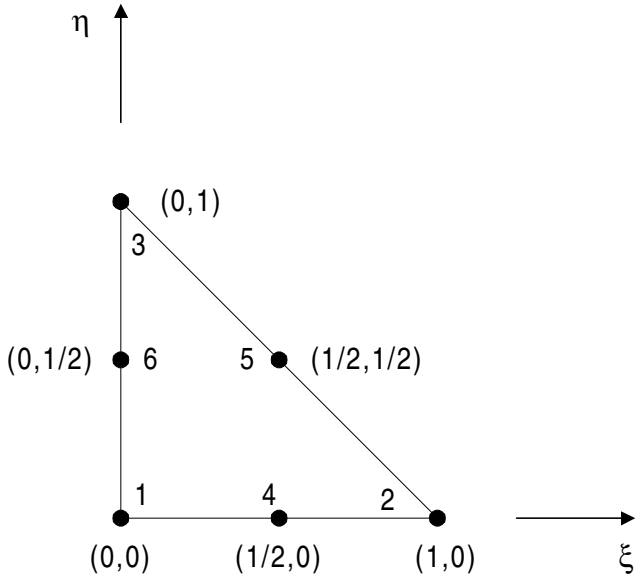

Figure 6. Base triangle for CHIEF element.

The local coordinates and normal vector are related to the ξ, η coordinates by

$$\begin{aligned} x &= x(\xi,\eta) & n_x &= \nu_x(\xi,\eta)/\sqrt{\nu_x(\xi,\eta)^2+\nu_y(\xi,\eta)^2+\nu_z(\xi,\eta)^2} \\ y &= y(\xi,\eta) & n_y &= \nu_y(\xi,\eta)/\sqrt{\nu_x(\xi,\eta)^2+\nu_y(\xi,\eta)^2+\nu_z(\xi,\eta)^2} \\ z &= z(\xi,\eta) & n_z &= \nu_z(\xi,\eta)/\sqrt{\nu_x(\xi,\eta)^2+\nu_y(\xi,\eta)^2+\nu_z(\xi,\eta)^2}, \end{aligned}$$

where

$$\begin{pmatrix} x(\xi,\eta) \\ y(\xi,\eta) \\ z(\xi,\eta) \end{pmatrix} = \sum_{n=1}^{3,6} \phi_n(\xi,\eta) \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix}$$

and

$$\begin{pmatrix} t_x^\xi(\xi,\eta) \\ t_y^\xi(\xi,\eta) \\ t_z^\xi(\xi,\eta) \end{pmatrix} = \sum_{n=1}^{3,6} \frac{\partial \phi_n(\xi,\eta)}{\partial \xi} \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} \qquad \xi \text{ tangent vector}$$

$$\begin{pmatrix} t_x^\eta(\xi,\eta) \\ t_y^\eta(\xi,\eta) \\ t_z^\eta(\xi,\eta) \end{pmatrix} = \sum_{n=1}^{3,6} \frac{\partial \phi_n(\xi,\eta)}{\partial \eta} \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} \qquad \eta \text{ tangent vector}$$

$$\begin{pmatrix} \nu_x(\xi,\eta) \\ \nu_y(\xi,\eta) \\ \nu_z(\xi,\eta) \end{pmatrix} = \begin{pmatrix} t_y^\xi(\xi,\eta) t_z^\eta(\xi,\eta) - t_z^\xi(\xi,\eta) t_y^\eta(\xi,\eta) \\ t_z^\xi(\xi,\eta) t_x^\eta(\xi,\eta) - t_x^\xi(\xi,\eta) t_z^\eta(\xi,\eta) \\ t_x^\xi(\xi,\eta) t_y^\eta(\xi,\eta) - t_y^\xi(\xi,\eta) t_x^\eta(\xi,\eta) \end{pmatrix}$$

x_n, y_n, z_n	given coordinates of nodes belonging to element
$\phi_n(\xi,\eta)$	finite element interpolation function (linear or quadratic)

For linear interpolation on a triangular element, the interpolation functions ϕ_n are given by

$$\begin{aligned} \phi_1(\xi,\eta) &= 1 - \xi - \eta \\ \phi_2(\xi,\eta) &= \xi \\ \phi_3(\xi,\eta) &= \eta. \end{aligned}$$

For quadratic interpolation on a triangular element,

$$\begin{aligned} \phi_1(\xi,\eta) &= 2(1-\xi-\eta)(\tfrac{1}{2}-\xi-\eta) \\ \phi_2(\xi,\eta) &= 2\xi(\xi-\tfrac{1}{2}) \\ \phi_3(\xi,\eta) &= 2\eta(\eta-\tfrac{1}{2}) \\ \phi_4(\xi,\eta) &= 4\xi(1-\xi-\eta) \\ \phi_5(\xi,\eta) &= 4\xi\eta \\ \phi_6(\xi,\eta) &= 4\eta(1-\xi-\eta). \end{aligned}$$

The normal is chosen so that the nodes appear in counterclockwise order when viewed from the positive normal side of the surface.

The base triangle is mapped onto a planar rectangular region by the polar-like transformation

$$\begin{aligned} \xi &= u(1-v) \\ \eta &= uv, \end{aligned}$$

where $0 \le u, v \le 1$.

This page intentionally left blank

6. CONTROL SECTION

In this section, commands are given to specify the type of job CHIEF is asked to perform. The commands in this section may be given in any order. Only the first three characters of the command word itself are used by the program. Other forms of the command words are perfectly valid as long as the same first three characters are used.

Each command line consists of a command or a command followed by one or more arguments. Some command lines must be followed by additional lines of data. Commands, arguments, and data items are separated by commas. Parameters that have been defined in the parameter section and expressions involving these parameters may be used for any numerical input in this section. As always, blank lines, comments, and spaces are ignored.

The following types of problems may be run in CHIEF (with only one type allowed per run). The first two types are the most commonly used. Following each type are a list of commands that must be included in the input file. Optional commands are given in braces.

1. Calculate surface and/or field pressures (see example problems 1, 2, 5, 6, 8):

   `velocity`, `acceleration`, or `planewave`

   `surface pressures`

   [`far-field`]

   [`near-field`]

2. Calculate impedance and/or field matrices (see example problem 4):

   `impedance`

   [`far-field`]

   [`near-field`]

3. Calculate A and B matrices (see example problem 9):

   `a_b matrices`

   One or more of the following are required:

   [`surface pressures`]

   [`far-field`]

   [`near-field`]

4. Point source check on numerical accuracy (see example problem 3):

   `point source`

5. Compute coupling matrix

```
couple
```

CHIEF solves exterior radiation or scattering problems by default. If the `inside` command is given, CHIEF solves an interior problem. For interior problems, commands such as `far-field` and `infinite baffle` make no sense. We now provide the format for each command. Optional parts of the command line are in brackets.

a_b matrices

outputs the surface or field A and B matrices

Command: **a_b matrices**

When the `surface pressures` command is given CHIEF produces a system of equations of the form $AP = BV + P_{inc}$ where P is the vector of surface pressures, V is the vector of surface normal velocities, and P_{inc} is the vector of incident surface pressures resulting from an incident plane wave specified by the user. When the `far-field pressures` command is given CHIEF produces a set of equations of the form $P_{ff} = AP + BV$ where P_{ff} is the vector of far-field radiated pressures at prescribed angles. When the `near-field pressures` command is given CHIEF produces a set of equations of the form $P_{nf} = AP + BV$ where P_{nf} is the vector of near-field pressures at prescribed locations.

A and B matrices may be output for any or all of the above three case. These matrices are output to binary files with extensions `.abs`, `.abf`, and/or `.abn` if the commands `surface pressures`, `far-field pressures`, and/or `near-field pressures`, respectively, have been given in the CHIEF input file.

acceleration

specifies surface normal acceleration input

Command: **acceleration**, <filename>

<filename> Name of file containing surface normal acceleration input. This file is an ASCII file containing five columns separated by one or more spaces. The first column is the acceleration distribution number (i.e., the first acceleration distribution will have a 1 in this column, the second a 2, etc.). The second column contains the symmetry block number. The third column contains the surface subdivision number. The fourth and fifth columns contain the real and imaginary parts of the normal acceleration.

If the acceleration has the same symmetry as the radiating surface, then only one symmetry block of the accelerations needs to be input.

acceleration (alternate form)

specifies surface normal acceleration input by means of a loop

---

Command: **acc**$(i_1 : i_2 : i_3, j_1 : j_2 : j_3, k_1 : k_2 : k_3) = (x, y) \quad I = I_1, I_2, I_3$

I	loop index (does not have to be I—the name can be selected by the user). If no loop is desired the portion of the command beginning with "$I =$" can be omitted.
I_1, I_2, I_3	I_1 is the starting index for the loop, I_2 is the ending index for the loop, and I_3 is the step size. These limits can be replaced by any expression involving previously defined parameters.
$i_1 : i_2 : i_3$	specifies a range for the surface subdivision number, where i_1 is the starting index, i_2 is the ending index, and i_3 is the step size. If i_3 is not present, it is set to 1. If i_2 and i_3 are not present, i_2 is set equal to i_1. The colon is not necessary if the following index is not present (e.g., the range could be given by $i_1 : i_2$ or just i_1). The range limits can be replaced by any expression involving previously defined parameters and/or the index I.
$j_1 : j_2 : j_3$	specifies a range for the symmetry block number, where j_1 is the starting index, j_2 is the ending index, and j_3 is the step size. If j_3 is not present, it is set to 1. If j_2 and j_3 are not present, j_2 is set equal to j_1.The colon is not necessary if the following index is not present (e.g., the range could be given by $j_1 : j_2$ or just j_1). The range limits can be replaced by any expression involving previously defined parameters and/or the index I.
$k_1 : k_2 : k_3$	specifies a range for the acceleration distribution number, where k_1 is the starting index, k_2 is the ending index, and k_3 is the step size. If k_3 is not present, it is set to 1. If k_2 and k_3 are not present, k_2 is set equal to k_1.The colon is not necessary if the following index is not present (e.g., the range could be given by $k_1 : k_2$ or just k_1). The range limits can be replaced by any expression involving previously defined parameters and/or the index I.
x, y	x is the real part of the acceleration and y is the imaginary part of the acceleration. If the imaginary part is zero, the user can use x in place of (x, y). x and y can be replaced by any expression involving previously defined parameters and/or the index I.

If the acceleration has the same symmetry as the radiating surface, then only one symmetry block of the accelerations needs to be input.

couple

calculates coupling matrix relating finite element forces to CHIEF pressures

---

Command: **couple**, Iprt, N_{dof}

Iprt — If Iprt = 1, then the coupling matrix will be output to the ASCII .out file in addition to being output to the binary file `<input filename>.cpl`. If Iprt = 0 (default), then the coupling matrix will only be output to the binary file.

N_{dof} — Number of finite element degrees of freedom per node (default 3).

The forcing vector F in the finite element program is given by

$$F_{m\alpha} = \int_S \vec{t} \cdot \vec{\phi}_{m\alpha}\, dS,$$

where m is the node number, α is the dof number ($1 \leq \alpha \leq 3$), $\vec{t}$ is the surface traction, and $\vec{\phi}_{m\alpha}$ is the finite element interpolation function corresponding to the node m and the dof α. The traction vector corresponding to acoustic loading is given by

$$\vec{t} = -p\vec{n},$$

where p is the acoustic pressure and $\vec{n}$ is the outward surface normal. Thus, the corresponding finite element force is given by

$$F_{m\alpha} = -\int_S p\vec{\phi}_{m\alpha} \cdot \vec{n}\, dS.$$

Because p is assumed constant over each surface subdivision S_n, we have

$$F_{m\alpha} = -\sum_n p_n \int_{S_n} \vec{\phi}_{m\alpha} \cdot \vec{n}\, dS,$$

where the summation index n extends over all surface subdivisions. The vector interpolation functions $\vec{\phi}_{m\alpha}$ are assumed to have the form

$$\vec{\phi}_{m\alpha} = \phi_m \vec{e}_\alpha,$$

where ϕ_m is a scalar interpolation function and $\vec{e}_\alpha$ is a unit vector in the α coordinate direction. Thus,

$$F_{m\alpha} = -\sum_n p_n \int_{S_n} \phi_m \vec{n} \cdot \vec{e}_\alpha\, dS.$$

This equation can be written in the alternate form

$$F_{m\alpha} = -\sum_n p_n A_n C_{m\alpha n},$$

where A_n is the area of the n-th subdivision and

$$C_{m\alpha n} = \frac{1}{A_n} \int_{S_n} \phi_m \vec{n} \cdot \vec{e}_\alpha \, dS$$

are the components of the coupling matrix. The coupling matrix is output as a two-dimensional matrix with the indices m and α combined into a single index in which the degrees of freedom for each node occur together, i.e.,

$$C_{m'n} = C_{m\alpha n} \qquad m' = (m-1)N_{\text{dof}} + \alpha.$$

far-field

calculates far-field pressures or matrix

Command: **far-field**, N, Iprt

θ_L^1, θ_U^1, θ_{inc}^1, ϕ_L^1, ϕ_U^1, ϕ_{inc}^1

$\vdots$

θ_L^N, θ_U^N, θ_{inc}^N, ϕ_L^N, ϕ_U^N, ϕ_{inc}^N

N	Number of far-field data lines to follow command
Iprt	Print flag. If Iprt=1, then far-field pressures are written to the standard output file (`<input filename>.out`). If Iprt=2, then the far-field pressures are written to an ASCII file (`<input filename>.ffp`) containing only far-field output. If Iprt=3, then the far-field pressures are written to both the standard output file and the far-field ASCII output file. If far-field matrices are computed, they are written to the binary file (`<input filename>.pmf`) regardless of the value of the print flag.
θ_L^k, θ_U^k, θ_{inc}^k	Lower limit, upper limit, and increment in θ for k-th θ, ϕ range
ϕ_L^k, ϕ_U^k, ϕ_{inc}^k	Lower limit, upper limit, and increment in ϕ for k-th θ, ϕ range

Each data line specifies a set of equally spaced ϕ's and a set of equally spaced θ's. The program performs far-field computations for all θ, ϕ pairs, where θ is taken from the set of equally spaced θ's and ϕ is taken from the set of equally spaced $\phi's$. Each successive data line specifies another collection of θ, ϕ pairs for which far-field calculations are to be performed.

If the `Impedance` command is contained in the control section, then far-field matrices are computed. Each matrix row corresponds to a direction specified by a θ, ϕ pair, and each column corresponds to a surface subdivision. If G is the far-field matrix, then the vector of far-field pressures P_{ff} at the specified directions is related to the vector of surface normal velocities V by

$$P_{ff} = GV.$$

If a surface velocity distribution is specified by a `Velocity` command in the control section, then the far-field radiated pressures corresponding to the velocity distribution are computed instead of the far-field matrix. If an incident plane wave is specified through the `Plane wave` command, then the scattered far-field pressures are computed.

fluid

user specified fluid density and sound speed

Command: **fluid**, ρ, c

ρ	Density of fluid
c	Sound speed in fluid

If the `fluid` command is omitted, then the default values of $\rho = 1000$ and $c = 1500$ will be used. These values are fairly typical for water in mks units, but if a different fluid or a different set of units is used, then the appropriate values can be set with this command.

frequency

specifies frequency range for calculations

Command: **frequency**, f_L [, f_U, f_{inc}]

f_L	Lower frequency limit for range of equally spaced frequencies
f_U	Upper frequency limit for range of equally spaced frequencies
f_{inc}	Frequency increment

If f_u and f_{inc} are missing, then only one frequency f_L will be used.

impedance

calculates impedance matrix

Command: **impedance**, Isym, Iprt

Isym	If Isym = 1, then the velocity distribution has the same symmetry as the radiating surface. If Isym = 0, then CHIEF will not take advantage of velocity symmetry
Iprt	Print flag. If Iprt = 1, then the impedance matrix will be output to the binary file `<input filename>.zmx`.

The impedance matrix Z is defined by

$$F = ZV,$$

where V is the vector of surface normal velocities on the CHIEF subdivisions and F is the vector of surface forces on the CHIEF subdivisions.

infinite baffle

specifies infinite rigid baffle

Command: infinite baffle

CHIEF treats the radiating surface as a planar region in an infinite rigid baffle. The surface definition must only consist of planar regions.

inside

specifies that an interior problem is to be solved

Command: inside

CHIEF solves an interior problem instead of an external problem. The positive normal direction is still taken as external to the surface.

interior points

adds equations corresponding to prescribed interior points to improve numerical stability of CHIEF equations at certain critical frequencies

---

Command: **interior points**, N

x_1, y_1, z_1

$\vdots$

x_N, y_N, z_N

N	Number of interior points per symmetry block
x_k, y_k, z_k	Global Cartesian coordinates of k-th interior point where $k = 1, \ldots, N$. If N sets of coordinates are not provided, the program will automatically generate the missing sets.

The CHIEF surface A matrix becomes ill-conditioned near certain critical frequencies corresponding to resonances of the interior problem with a zero surface pressure boundary condition. The addition of equations corresponding to interior points relieves this ill-conditioning if at least one of the points does not lie on a nodal surface for the interior mode corresponding to the critical frequency. To automatically generate an interior point CHIEF randomly chooses a surface field point (the center of some subdivision) and goes along the interior normal at that point a distance equal to the square root of the area of the subdivision. CHIEF then checks this point using an integral expression (see Appendix A) to determine that the point is inside the body. If not, the program tries another surface field point. The program gives a warning stating the number of unsuccessful attempts. Unsuccessful attempts often occur when the body is very thin and the subdivisions are too large.

memory

specifies amount of RAM memory available for storing large matrices.

Command: **memory**, *M*

M	Bytes of RAM memory available for storing large matrices such as the CHIEF A,B-matrices. The default is 20 MB.

The amount of RAM available is used to determine how many rows of each symmetry block can be stored in memory at any one time. Two complex matrix areas are dimensioned to hold the matrices in memory. The number of columns is dimensioned to the number of surface subdivisions and the number of rows is dimensioned to the number of rows per symmetry block times the number of symmetry blocks for the largest matrix used (surface, far-field, near-field).

near-field

calculates near-field pressures or matrix

Command: **near-field**, N, Iprt

x_1, y_1, z_1

$\vdots$

x_N, y_N, z_N

N	Number of near-field points
Iprt	Print flag. If Iprt = 1, then near-field pressures are written to the standard output file (`<input filename>.out`). If Iprt = 2, then the near-field pressures are written to an ASCII file (`<input filename>.nfp`) containing only near-field output. If Iprt = 3, then the near-field pressures are written to both the standard output file and the near-field ASCII output file. If near-field matrices are computed, they are output to the binary file (`<input filename>.pmn`) regardless of the value of the print flag.
x_k, y_k, z_k	Global Cartesian coordinates of k-th near-field point at which near-field pressures are desired ($k = 1, \ldots, N$).

If the `Impedance` command is contained in the control section , then near-field matrices are computed. Each row of this matrix corresponds to a near-field point and each column corresponds to a surface subdivision. If G is the near-field matrix, then the vector of near-field pressures P_{nf} at the specified points is related to the vector of surface normal velocities V by

$$P_{nf} = GV.$$

If a surface velocity distribution is specified by a `Velocity` command in the control section, then the near-field radiated pressures corresponding to the velocity distribution are computed instead of the far-field matrix. If an incident plane wave is specified through the `Plane wave` command, then the scattered near-field pressures are computed.

plane wave

specifies plane incident wave for rigid scattering problem

Command: **plane wave**, Isym, a_{inc}, θ_{inc}, ϕ_{inc}

Isym	If Isym = 1, then the incident pressure field has the same symmetry as the radiating surface. If Isym = 0, then CHIEF will not take advantage of velocity symmetry
a_{inc}	Amplitude of the incident plane wave
θ_{inc}, ϕ_{inc}	Angles (in degrees) specifying the direction from which the plane wave is coming relative to the global coordinate axes. θ_{inc} is the angle from the x_3-axis and ϕ_{inc} is the angle from the x_1-axis in the x_1x_2-plane.

Since CHIEF does not contain any structural capabilities, it can only compute rigid scattering.

pscheck

uses pressures from specified point sources to check the accuracy of CHIEF calculations

---

Command: **pscheck**, Isym, N

x_1, y_1, z_1, W_1^{re}, W_1^{im}

$\vdots$

x_N, y_N, z_N, W_N^{re}, W_N^{im}

Isym	Isym = 1 specifies that the point source distribution has the same symmetry as the radiating surface. In this case, the point sources only need to be input for the first symmetry block
N	Number of point sources
x_k, y_k, z_k	Global Cartesian coordinates of k-th point source
W_k^{re}, W_k^{im}	Real and imaginary parts of the weight for the k-th point source

The point source check uses analytic expressions to calculate surface velocities and incident surface pressures (for exterior point sources) due to the prescribed point sources assuming that the surface is transparent. The surface pressures are then calculated using the CHIEF equations

$$AP = BV + P^{inc}.$$

If all the point sources are interior to the surface, $P^{inc} = 0$. The surface pressures calculated using CHIEF can be compared to the pressures on the surface calculated using analytical expressions for point sources. This comparison can be used to determine the adequacy of the surface break-up into subareas, adequacy of interior points to remove ill-conditioning at critical frequencies, and errors due to improper input such as wrong directions for normal vectors. The location of the point sources should be selected to produce a velocity distribution that is as smooth as the expected velocity distribution. For thin bodies, the velocity due to an interior point source will vary very rapidly near the point source. Therefore, it may be necessary to use exterior point sources to obtain a velocity distribution with the desired smoothness. The program automatically determines if a point source is located interior to or exterior to the boundary surface.

pwcheck

uses pressures from incident plane wave to check the accuracy of CHIEF calculations

---

Command: **pwcheck**, a_{inc}, θ_{inc}, ϕ_{inc}, [Isym]

a_{inc}	magnitude of incident wave
θ_{inc}, ϕ_{inc}	Angles (in degrees) specifying the direction from which the plane wave is coming relative to the global coordinate axes. θ_{inc} is the angle from the x_3-axis and ϕ_{inc} is the angle from the x_1-axis in the x_1x_2-plane.
Isym	Isym = 1 specifies that the plane wave pressure distribution has the same symmetry as the radiating surface. The default is Isym=0.

The plane wave check uses an analytic expression to calculate surface velocities and incident surface pressures due to the plane wave assuming that the surface is transparent. The surface pressures are then calculated using the CHIEF equations

$$AP = BV + P^{inc}.$$

The surface pressures calculated using CHIEF can be compared to the pressures on the surface calculated using analytical expressions for point sources. This comparison can be used to determine the adequacy of the surface break-up into subareas, adequacy of interior points to remove ill-conditioning at critical frequencies, and errors due to improper input such as wrong directions for normal vectors.

rhs symmetry

specifies that the velocity (appearing on the right-hand side of the equation $Ap = Bv$) has the same symmetry as the surface

---

Command: **rhs**

point source

specifies spherical incident wave for rigid scattering problem

---

Command: **point source**, x_1, x_2, x_3, w_{real}, w_{imag}, [Isym]

x_1, x_2, x_3	global coordinates of point source producing spherical incident wave.
w_{real}, w_{imag}	real and imaginary parts of the weight applied to point source.
Isym	If Isym = 1, then the incident pressure field has the same symmetry as the radiating surface. The default is Isym = 0

Since CHIEF does not contain any structural capabilities, it can only compute rigid scattering.

surface pressures

specifies calculation of surface pressures

Command: **surface pressures**, [Iprt]

Iprt — Print flag. If Iprt = 1, then surface pressures are written to the standard output file (`<input filename>.out`). If Iprt = 2, then the surface pressures are written to an ASCII file (`<input filename>.sps`) that only contains surface pressure output. If Iprt = 3, then the surface pressures are written to both the standard output file and the surface pressure ASCII output file.

velocity

specifies surface normal velocity input

Command: **velocity**, <filename>

<filename> Name of file containing surface normal velocity input. This file is an ASCII file containing five columns separated by one or more spaces. The first column is the velocity distribution number (i.e., the first velocity distribution will have a 1 in this column, the second a 2, etc.). The second column contains the symmetry block number. The third column contains the surface subdivision number. The fourth and fifth columns contain the real and imaginary parts of the normal velocity.

If the velocity has the same symmetry as the radiating surface, then only one symmetry block of the velocities needs to be input.

velocity (alternate form)

specifies surface normal velocity input by means of a loop

Command: **vel**$(i_1 : i_2 : i_3, j_1 : j_2 : j_3, k_1 : k_2 : k_3) = (x, y) \quad I = I_1, I_2, I_3$

I	loop index (does not have to be I—the name can be selected by the user). If no loop is desired the portion of the command beginning with "$I =$" can be omitted.
I_1, I_2, I_3	I_1 is the starting index for the loop, I_2 is the ending index for the loop, and I_3 is the step size. These limits can be replaced by any expression involving previously defined parameters.
$i_1 : i_2 : i_3$	specifies a range for the surface subdivision number, where i_1 is the starting index, i_2 is the ending index, and i_3 is the step size. If i_3 is not present, it is set to 1. If i_2 and i_3 are not present, i_2 is set equal to i_1. The colon is not necessary if the following index is not present (e.g., the range could be given by $i_1 : i_2$ or just i_1). The range limits can be replaced by any expression involving previously defined parameters and/or the index I.
$j_1 : j_2 : j_3$	specifies a range for the symmetry block number, where j_1 is the starting index, j_2 is the ending index, and j_3 is the step size. If j_3 is not present, it is set to 1. If j_2 and j_3 are not present, j_2 is set equal to j_1.The colon is not necessary if the following index is not present (e.g., the range could be given by $j_1 : j_2$ or just j_1). The range limits can be replaced by any expression involving previously defined parameters and/or the index I.
$k_1 : k_2 : k_3$	specifies a range for the velocity distribution number, where k_1 is the starting index, k_2 is the ending index, and k_3 is the step size. If k_3 is not present, it is set to 1. If k_2 and k_3 are not present, k_2 is set equal to k_1.The colon is not necessary if the following index is not present (e.g., the range could be given by $k_1 : k_2$ or just k_1). The range limits can be replaced by any expression involving previously defined parameters and/or the index I.
x, y	x is the real part of the velocity and y is the imaginary part of the velocity. If the imaginary part is zero, the user can use x in place of (x, y). x and y can be replaced by any expression involving previously defined parameters and/or the index I.

If the velocity has the same symmetry as the radiating surface, then only one symmetry block of the velocities needs to be input.

7. OUTPUT FILE FORMATS

In this section, we describe the formats of the various output files. The files that are likely to be small are written as formatted ASCII files. The files that are likely to be large are written as unformatted binary files. The version of this program for Windows 95/98® and Windows NT® machines was compiled using the Lahey Fortran 95 compiler. The version for the DEC Alpha OSF1 was compiled using the Digital Fortran 90 compiler. In both cases, the unformatted sequential binary files are divided into records that have a 4-byte integer prefix and suffix containing the number of bytes in the record (excluding prefix and suffix). Both also use IEEE little endian format for floating point reals and integers. The first record in every binary file contains the following 14 items:

1. Starting frequency (32-bit real)
2. Frequency increment (32-bit real)
3. Fluid density (32-bit real)
4. Sound speed (32-bit real)
5. Symmetry type (4 characters, 32-bits)
6. Number of frequencies (32-bit integer)
7. Number of symmetry blocks (32-bit integer)
8. Number of symmetry blocks output (32-bit integer)
9. Right-hand-side symmetry flag (32-bit integer)
10. Number of surface field points (32-bit integer)
11. Number of surface subdivisions (32-bit integer)
12. Number of far field theta-phi angle pairs (32-bit integer)
13. Number of near-field points (32-bit integer)
14. Number of right-hand-sides (32-bit integer)

All or part of a binary output file can be converted to ASCII using the program 'convert'. The format for this command is

```
convert <filename> <frequency range> <block number range>,
```

where `<filename>` is the filename including extension; `<frequency range>` contains a starting frequency, an ending frequency, and a frequency increment separated by colons (e.g., 100:1000:10); `<block number range>` contains a beginning symmetry block number, an ending symmetry block number, and a block number increment separated by colons (e.g., 1:6:2). If the increment is omitted, then the increment used in creating the file is used. If the upper value and increment are omitted, then only the lower value is used. Thus, 100:1000 and 100 are valid ranges. If no ranges are specified, the program prints some header information from the file and asks if you wish to continue. If you answer 'N', the program quits. If you answer 'Y', the program converts all the results in the file. The converted results are put in a file with the same file name and "_ascii" appended to the extension.

The formats of quantities in the various output files are described using Fortran-type specifiers. For example, (F12.3) specifies a fixed-point-style real number with a total width of 12 and 3 digits to the right of the decimal point, (I5) specifies a 5-digit integer, and (E14.2) specifies an exponential-style real number with a total width of 14 and 3 digits to the right of the decimal place. The total width includes any preliminary plus or minus sign, the decimal point, and the exponent (E±xx). We will now look at each file type individually.

Surface pressure file (.sps)

This file is formatted ASCII. For each frequency, the first line contains the density, the sound speed, the number of velocity distributions, the number of symmetry blocks, the number of symmetry blocks output, the right-hand-side symmetry flag, and the number of areas with the format (2f10.3,5i5). The second line contains the frequency with format (f12.3). The following lines have the form,

$$i_{rhs} \quad i_{blk} \quad i \quad P_{real}(i) \quad P_{imag}(i) \quad area(i),$$

where i_{rhs} is an index running over the number of velocity distributions, i_{blk} is an index running over the number of symmetry blocks, i is an index running over the number of CHIEF subdivisions, $P_{real}(i)$ and P_{imag} are the real and imaginary parts of the surface pressure on the i-th subdivision, and $area(i)$ is the area of the i-th subdivision. The format for each of these lines is (3I5, 3E15.6). The index i varies the fastest, next is i_{blk}, and slowest is i_{rhs}. The above lines are repeated for each frequency.

Far-field pressure file (.ffp)

This file is formatted ASCII. For each frequency, the first line contains the density, the sound speed, the number of velocity distributions, and the number of far-field points with the format (2f10.3,2i5). The second line contains the frequency with the format (F12.3). The following lines have the form,

$$i_{rhs} \quad \phi \quad \theta \quad P_{ff}^{real}(\phi,\theta) \quad P_{ff}^{imag}(\phi,\theta),$$

where i_{rhs} is an index running over the number of velocity distributions, ϕ and θ run over all θ-ϕ pairs, and $P_{ff}^{real}(\phi,\theta)$, $P_{ff}^{imag}(\phi,\theta)$ are the real and imaginary parts of the far-field pressure. The format for each of these lines is (I5, 2F10.2, 2E15.6). The above lines are repeated for each frequency.

Near-field pressure file (.nfp)

This file is formatted ASCII. For each frequency, the first line contains the density, the sound speed, the number of velocity distributions, and the number of near-field points with the format (2f10.3,2i5). The second line contains the frequency with the format (F12.3). The following lines have the form,

$$i_{rhs} \quad x \quad y \quad z \quad P_{nf}^{real}(x,y,z) \quad P_{nf}^{imag}(x,y,z),$$

where i_{rhs} is an index running over the number of velocity distributions, x, y, and z run over all prescribed near-field points, and $P_{nf}^{real}(x,y,z)$, $P_{nf}^{imag}(x,y,z)$ are the real and imaginary parts of the near-field pressure. The format for each of these lines is (I5, 3E12.4, 2E15.6). The above lines are repeated for each frequency.

Impedance Matrix file (.zmx)

This file is binary. The first record is the standard header. The second record contains the surface areas of the CHIEF subdivisions. Each of the following records contains a frequency, a symmetry block number, and a block of the impedance matrix. The matrices are written out column by column. All the blocks for a frequency are output before the next frequency is output.

Far-field Matrix file (.pmf)

This file is binary. The first record is the standard header. The second record contains the theta-phi pairs for the far-field directions. Each of the following records contains a frequency, a symmetry block number, and a block of the far-field pattern matrix. The matrices are written out column by column. All the blocks for a frequency are output before the next frequency is output.

Near-field Matrix file (.pmn)

This file is binary. The first record is the standard header. The second record contains the coordinates for the near-field points. Each of the following records contains a frequency, a symmetry block number, and a block of the near field pattern matrix. The matrices are written out column by column. All the blocks for a frequency are output before the next frequency is output.

Surface A,B Matrices file (.abs)

This file is binary. The first record is the standard header. The second record contains the surface areas of the CHIEF subdivisions. Each of the following records contains a frequency, a symmetry block number, a block of the surface A matrix, and a block of the surface B matrix. The matrices are written out column by column. All the blocks for a frequency are output before the next frequency is output.

Far-field A,B Matrices file (.abf)

This file is binary. The first record is the standard header. The second record contains the theta-phi pairs for the far-field directions. Each of the following records contains a frequency, a symmetry block number, a block of the far-field A matrix, and a block of the far-field B matrix. The matrices are written out column by column. All the blocks for a frequency are output before the next frequency is output.

Near-field A,B Matrices file (.abn)

This file is binary. The first record is the standard header. The second record contains the coordinates for the near-field points. Each of the following records contains a frequency, a symmetry block number, a block of the near-field A matrix, and a block of the near-field B matrix. The matrices are written out column by column. All the blocks for a frequency are output before the next frequency is output.

Coupling Matrix file (.cpl)

This file is binary. The first record contains the number of finite element nodes, the number of degrees of freedom per node, and the number of surface elements. The second record contains the finite element node numbers. The third record contains the surface areas of the elements. The fourth record contains the coupling matrix for the first symmetry block. The coupling matrices for the other symmetry blocks are not output because they are identical to the coupling matrix for the first block.

8. EXAMPLE PROBLEMS

In this section, we will show how to set up the ASCII input files for ten example problems. Most of these examples are the same as those used in the original CHIEF manual. To gain maximum benefit from these example problems, the user should try to set up each of the input files without looking at the answers provided.

Example Problem 1

Figure 7 shows the geometry for the first problem. It consists of a free-flooded cylinder having an inner radius of 1.68 m, an outer radius of 2.04 m, and a height of 1.2 m. The normal velocity on the inner cylindrical surface is -1.029 m/sec, the normal velocity on the outer cylindrical surface is 0.971 m/sec, and the normal velocity on the top and bottom surfaces is 0.097 m/sec.

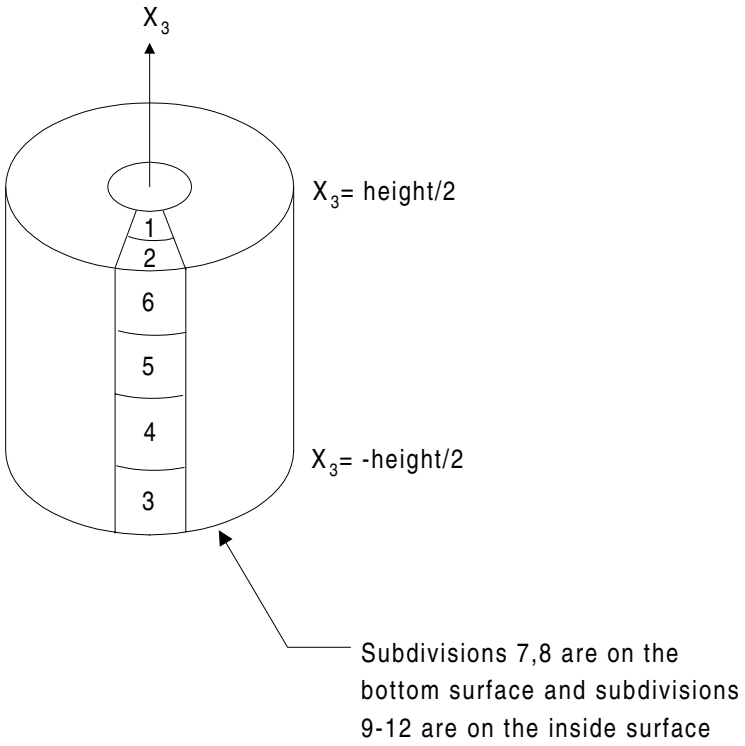

Figure 7. Geometry of example problem 1.

The problem is to compute the surface pressures and the far-field pattern in the plane $\phi = 0$ with $\Delta\theta = 5°$ and $0 \leq \theta \leq 90°$ for a frequency of $1500/(2\pi)$. Use 12-fold rotational symmetry and no velocity symmetry (even though there really is) to solve the problem. The surface and far-field pressures should be output to a file. The following is an input file for this problem:

```
freq1=1500/(2*pi)
nblks=12
ai=1.68
ao=2.04
ht=1.2
htd2=.5*ht
rotlim=pi/nblks
rho=999.89
```

```
end*params

rotational,nblks

circ_planar, htd2
ai,ao,-rotlim,rotlim
2,1,4,16
3,0,0,0

circ_cylinder, ao
-htd2,htd2,-rotlim,rotlim
4,1,4,16
3

circ_planar, -htd2
ai,ao,-rotlim,rotlim
2,1,4,16
-3

circ_cylinder, ai
-htd2,htd2,-rotlim,rotlim
4,1,4,16
-3

end*geom

fluid,rho,1500.05

freq, freq1, freq1, 0.

vel,tc1.vel

surface,3

farfield,1,3
0,90,5,0,0,0
```

The user also needs to construct a file containing the velocity distribution. Since we are not using the symmetry of the velocity, the velocity distribution must be input for all 12 of the rotational symmetry blocks. If we had chosen to use the velocity symmetry, then the input file would only need to contain the velocity distribution for the first symmetry block. The velocity input file for example problem 1 is shown below:

```
1 1 1   0.097  0
```

```
1 1 2   0.097  0
1 1 3   0.971  0
1 1 4   0.971  0
1 1 5   0.971  0
1 1 6   0.971  0
1 1 7   0.097  0
1 1 8   0.097  0
1 1 9  -1.029  0
1 1 10 -1.029  0
1 1 11 -1.029  0
1 1 12 -1.029  0
1 2 1   0.097  0
1 2 2   0.097  0
1 2 3   0.971  0
1 2 4   0.971  0
1 2 5   0.971  0
1 2 6   0.971  0
1 2 7   0.097  0
1 2 8   0.097  0
1 2 9  -1.029  0
1 2 10 -1.029  0
1 2 11 -1.029  0
1 2 12 -1.029  0
1 3 1   0.097  0
1 3 2   0.097  0
1 3 3   0.971  0
1 3 4   0.971  0
1 3 5   0.971  0
1 3 6   0.971  0
1 3 7   0.097  0
1 3 8   0.097  0
1 3 9  -1.029  0
1 3 10 -1.029  0
1 3 11 -1.029  0
1 3 12 -1.029  0
1 4 1   0.097  0
1 4 2   0.097  0
1 4 3   0.971  0
1 4 4   0.971  0
1 4 5   0.971  0
1 4 6   0.971  0
1 4 7   0.097  0
1 4 8   0.097  0
1 4 9  -1.029  0
1 4 10 -1.029  0
```

```
1 4 11 -1.029  0
1 4 12 -1.029  0
1 5 1   0.097  0
1 5 2   0.097  0
1 5 3   0.971  0
1 5 4   0.971  0
1 5 5   0.971  0
1 5 6   0.971  0
1 5 7   0.097  0
1 5 8   0.097  0
1 5 9  -1.029  0
1 5 10 -1.029  0
1 5 11 -1.029  0
1 5 12 -1.029  0
1 6 1   0.097  0
1 6 2   0.097  0
1 6 3   0.971  0
1 6 4   0.971  0
1 6 5   0.971  0
1 6 6   0.971  0
1 6 7   0.097  0
1 6 8   0.097  0
1 6 9  -1.029  0
1 6 10 -1.029  0
1 6 11 -1.029  0
1 6 12 -1.029  0
1 7 1   0.097  0
1 7 2   0.097  0
1 7 3   0.971  0
1 7 4   0.971  0
1 7 5   0.971  0
1 7 6   0.971  0
1 7 7   0.097  0
1 7 8   0.097  0
1 7 9  -1.029  0
1 7 10 -1.029  0
1 7 11 -1.029  0
1 7 12 -1.029  0
1 8 1   0.097  0
1 8 2   0.097  0
1 8 3   0.971  0
1 8 4   0.971  0
1 8 5   0.971  0
1 8 6   0.971  0
1 8 7   0.097  0
```

```
1 8 8   0.097  0
1 8 9  -1.029  0
1 8 10 -1.029  0
1 8 11 -1.029  0
1 8 12 -1.029  0
1 9 1   0.097  0
1 9 2   0.097  0
1 9 3   0.971  0
1 9 4   0.971  0
1 9 5   0.971  0
1 9 6   0.971  0
1 9 7   0.097  0
1 9 8   0.097  0
1 9 9  -1.029  0
1 9 10 -1.029  0
1 9 11 -1.029  0
1 9 12 -1.029  0
1 10 1   0.097  0
1 10 2   0.097  0
1 10 3   0.971  0
1 10 4   0.971  0
1 10 5   0.971  0
1 10 6   0.971  0
1 10 7   0.097  0
1 10 8   0.097  0
1 10 9  -1.029  0
1 10 10 -1.029  0
1 10 11 -1.029  0
1 10 12 -1.029  0
1 11 1   0.097  0
1 11 2   0.097  0
1 11 3   0.971  0
1 11 4   0.971  0
1 11 5   0.971  0
1 11 6   0.971  0
1 11 7   0.097  0
1 11 8   0.097  0
1 11 9  -1.029  0
1 11 10 -1.029  0
1 11 11 -1.029  0
1 11 12 -1.029  0
1 12 1   0.097  0
1 12 2   0.097  0
1 12 3   0.971  0
1 12 4   0.971  0
```

```
1 12 5   0.971  0
1 12 6   0.971  0
1 12 7   0.097  0
1 12 8   0.097  0
1 12 9  -1.029  0
1 12 10 -1.029  0
1 12 11 -1.029  0
1 12 12 -1.029  0
```

For this example, the velocity could be specified directly in the input file instead of reading the velocities from a data file. This is accomplished by replacing the line

```
vel, tc1.vel
```

by the lines,

```
vel(1:2,1:12,1)=(.097,0.)
vel(3:6,1:12,1)=(.971,0.)
vel(7:8,1:12,1)=(.097,0.)
vel(9:12,1:12,1)=(-1.029,0.)
```

The following pages show the output that would be obtained in the .out file for this problem.

```
SURFACE GEOMETRY INPUT

region  1

neqn  nu  nv  izaxis  iquadu  iquadv
  2    2   1     3       4       16

    ul             uu             vl             vu
0.168000E+01   0.204000E+01   -.261799E+00   0.261799E+00

cc array
0.600000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  2

neqn  nu  nv  izaxis  iquadu  iquadv
  4    4   1     3       4       16

    ul             uu             vl             vu
-.600000E+00   0.600000E+00   -.261799E+00   0.261799E+00

cc array
0.204000E+01   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  3

neqn  nu  nv  izaxis  iquadu  iquadv
  2    2   1    -3       4       16

    ul             uu             vl             vu
0.168000E+01   0.204000E+01   -.261799E+00   0.261799E+00

cc array
-.600000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  4

neqn  nu  nv  izaxis  iquadu  iquadv
  4    4   1    -3       4       16
```

```
    ul             uu             vl             vu
-.600000E+00   0.600000E+00   -.261799E+00   0.261799E+00

cc array
0.168000E+01   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


#areas =  12     # interior points = 0




 Memory Usage (bytes)

 quadrature points        24576
 surface pressures        1152
 far-field pressures      152
 near-field pressures     0
 point source pressures   0
 external pressures       0
 dpdn                     1152
 surface field points     1728
 far-field points         2736
 near-field points        0
 A,B matrices             43776
 total                    75272

 number of surface row blocks    = 1
 number of far-field row blocks  = 1
 number of near-field row blocks = 0



 Frequency = 238.732407

 SURFACE PRESSURES

 Block 1   Right-hand-side 1

          real           imag           area
  1    -0.724289E+06   0.223635E+06   0.166819E+00
  2    -0.371080E+06   0.523689E+06   0.183783E+00
  3     0.213456E+06   0.131391E+07   0.320443E+00
  4     0.345704E+06   0.154564E+07   0.320443E+00
  5     0.345704E+06   0.154564E+07   0.320443E+00
  6     0.213456E+06   0.131391E+07   0.320443E+00
  7    -0.724290E+06   0.223635E+06   0.166819E+00
  8    -0.371080E+06   0.523689E+06   0.183783E+00
```

```
 9    -0.162378E+07  -0.103561E+07   0.263894E+00
10    -0.189947E+07  -0.144824E+07   0.263894E+00
11    -0.189947E+07  -0.144824E+07   0.263894E+00
12    -0.162378E+07  -0.103561E+07   0.263894E+00


Block 2   Right-hand-side 1

          real           imag           area
 1    -0.724289E+06   0.223635E+06   0.166819E+00
 2    -0.371080E+06   0.523689E+06   0.183783E+00
 3     0.213456E+06   0.131391E+07   0.320443E+00
 4     0.345704E+06   0.154564E+07   0.320443E+00
 5     0.345704E+06   0.154564E+07   0.320443E+00
 6     0.213456E+06   0.131391E+07   0.320443E+00
 7    -0.724290E+06   0.223635E+06   0.166819E+00
 8    -0.371080E+06   0.523689E+06   0.183783E+00
 9    -0.162378E+07  -0.103561E+07   0.263894E+00
10    -0.189947E+07  -0.144824E+07   0.263894E+00
11    -0.189947E+07  -0.144824E+07   0.263894E+00
12    -0.162378E+07  -0.103561E+07   0.263894E+00


Block 3   Right-hand-side 1

          real           imag           area
 1    -0.724289E+06   0.223635E+06   0.166819E+00
 2    -0.371080E+06   0.523689E+06   0.183783E+00
 3     0.213456E+06   0.131391E+07   0.320443E+00
 4     0.345704E+06   0.154564E+07   0.320443E+00
 5     0.345704E+06   0.154564E+07   0.320443E+00
 6     0.213456E+06   0.131391E+07   0.320443E+00
 7    -0.724290E+06   0.223635E+06   0.166819E+00
 8    -0.371080E+06   0.523689E+06   0.183783E+00
 9    -0.162378E+07  -0.103561E+07   0.263894E+00
10    -0.189947E+07  -0.144824E+07   0.263894E+00
11    -0.189947E+07  -0.144824E+07   0.263894E+00
12    -0.162378E+07  -0.103561E+07   0.263894E+00


Block 4   Right-hand-side 1

          real           imag           area
 1    -0.724289E+06   0.223635E+06   0.166819E+00
 2    -0.371080E+06   0.523689E+06   0.183783E+00
 3     0.213456E+06   0.131391E+07   0.320443E+00
 4     0.345704E+06   0.154564E+07   0.320443E+00
 5     0.345704E+06   0.154564E+07   0.320443E+00
 6     0.213456E+06   0.131391E+07   0.320443E+00
 7    -0.724290E+06   0.223635E+06   0.166819E+00
 8    -0.371080E+06   0.523689E+06   0.183783E+00
 9    -0.162378E+07  -0.103561E+07   0.263894E+00
10    -0.189947E+07  -0.144824E+07   0.263894E+00
11    -0.189947E+07  -0.144824E+07   0.263894E+00
```

```
12    -0.162378E+07  -0.103561E+07   0.263894E+00


Block 5   Right-hand-side 1

          real           imag           area
 1    -0.724289E+06   0.223635E+06   0.166819E+00
 2    -0.371080E+06   0.523689E+06   0.183783E+00
 3     0.213456E+06   0.131391E+07   0.320443E+00
 4     0.345704E+06   0.154564E+07   0.320443E+00
 5     0.345704E+06   0.154564E+07   0.320443E+00
 6     0.213456E+06   0.131391E+07   0.320443E+00
 7    -0.724290E+06   0.223635E+06   0.166819E+00
 8    -0.371080E+06   0.523689E+06   0.183783E+00
 9    -0.162378E+07  -0.103561E+07   0.263894E+00
10    -0.189947E+07  -0.144824E+07   0.263894E+00
11    -0.189947E+07  -0.144824E+07   0.263894E+00
12    -0.162378E+07  -0.103561E+07   0.263894E+00


Block 6   Right-hand-side 1

          real           imag           area
 1    -0.724289E+06   0.223635E+06   0.166819E+00
 2    -0.371080E+06   0.523689E+06   0.183783E+00
 3     0.213456E+06   0.131391E+07   0.320443E+00
 4     0.345704E+06   0.154564E+07   0.320443E+00
 5     0.345704E+06   0.154564E+07   0.320443E+00
 6     0.213456E+06   0.131391E+07   0.320443E+00
 7    -0.724290E+06   0.223635E+06   0.166819E+00
 8    -0.371080E+06   0.523689E+06   0.183783E+00
 9    -0.162378E+07  -0.103561E+07   0.263894E+00
10    -0.189947E+07  -0.144824E+07   0.263894E+00
11    -0.189947E+07  -0.144824E+07   0.263894E+00
12    -0.162378E+07  -0.103561E+07   0.263894E+00


Block 7   Right-hand-side 1

          real           imag           area
 1    -0.724289E+06   0.223635E+06   0.166819E+00
 2    -0.371080E+06   0.523689E+06   0.183783E+00
 3     0.213456E+06   0.131391E+07   0.320443E+00
 4     0.345704E+06   0.154564E+07   0.320443E+00
 5     0.345704E+06   0.154564E+07   0.320443E+00
 6     0.213456E+06   0.131391E+07   0.320443E+00
 7    -0.724290E+06   0.223635E+06   0.166819E+00
 8    -0.371080E+06   0.523689E+06   0.183783E+00
 9    -0.162378E+07  -0.103561E+07   0.263894E+00
10    -0.189947E+07  -0.144824E+07   0.263894E+00
11    -0.189947E+07  -0.144824E+07   0.263894E+00
12    -0.162378E+07  -0.103561E+07   0.263894E+00
```

Block 8 Right-hand-side 1

	real	imag	area
1	-0.724289E+06	0.223635E+06	0.166819E+00
2	-0.371080E+06	0.523689E+06	0.183783E+00
3	0.213456E+06	0.131391E+07	0.320443E+00
4	0.345704E+06	0.154564E+07	0.320443E+00
5	0.345704E+06	0.154564E+07	0.320443E+00
6	0.213456E+06	0.131391E+07	0.320443E+00
7	-0.724290E+06	0.223635E+06	0.166819E+00
8	-0.371080E+06	0.523689E+06	0.183783E+00
9	-0.162378E+07	-0.103561E+07	0.263894E+00
10	-0.189947E+07	-0.144824E+07	0.263894E+00
11	-0.189947E+07	-0.144824E+07	0.263894E+00
12	-0.162378E+07	-0.103561E+07	0.263894E+00

Block 9 Right-hand-side 1

	real	imag	area
1	-0.724289E+06	0.223635E+06	0.166819E+00
2	-0.371080E+06	0.523689E+06	0.183783E+00
3	0.213456E+06	0.131391E+07	0.320443E+00
4	0.345704E+06	0.154564E+07	0.320443E+00
5	0.345704E+06	0.154564E+07	0.320443E+00
6	0.213456E+06	0.131391E+07	0.320443E+00
7	-0.724290E+06	0.223635E+06	0.166819E+00
8	-0.371080E+06	0.523689E+06	0.183783E+00
9	-0.162378E+07	-0.103561E+07	0.263894E+00
10	-0.189947E+07	-0.144824E+07	0.263894E+00
11	-0.189947E+07	-0.144824E+07	0.263894E+00
12	-0.162378E+07	-0.103561E+07	0.263894E+00

Block 10 Right-hand-side 1

	real	imag	area
1	-0.724289E+06	0.223635E+06	0.166819E+00
2	-0.371080E+06	0.523689E+06	0.183783E+00
3	0.213456E+06	0.131391E+07	0.320443E+00
4	0.345704E+06	0.154564E+07	0.320443E+00
5	0.345704E+06	0.154564E+07	0.320443E+00
6	0.213456E+06	0.131391E+07	0.320443E+00
7	-0.724290E+06	0.223635E+06	0.166819E+00
8	-0.371080E+06	0.523689E+06	0.183783E+00
9	-0.162378E+07	-0.103561E+07	0.263894E+00
10	-0.189947E+07	-0.144824E+07	0.263894E+00
11	-0.189947E+07	-0.144824E+07	0.263894E+00
12	-0.162378E+07	-0.103561E+07	0.263894E+00

Block 11 Right-hand-side 1

	real	imag	area

1	-0.724289E+06	0.223635E+06	0.166819E+00
2	-0.371080E+06	0.523689E+06	0.183783E+00
3	0.213456E+06	0.131391E+07	0.320443E+00
4	0.345704E+06	0.154564E+07	0.320443E+00
5	0.345704E+06	0.154564E+07	0.320443E+00
6	0.213456E+06	0.131391E+07	0.320443E+00
7	-0.724290E+06	0.223635E+06	0.166819E+00
8	-0.371080E+06	0.523689E+06	0.183783E+00
9	-0.162378E+07	-0.103561E+07	0.263894E+00
10	-0.189947E+07	-0.144824E+07	0.263894E+00
11	-0.189947E+07	-0.144824E+07	0.263894E+00
12	-0.162378E+07	-0.103561E+07	0.263894E+00

Block 12 Right-hand-side 1

	real	imag	area
1	-0.724289E+06	0.223635E+06	0.166819E+00
2	-0.371080E+06	0.523689E+06	0.183783E+00
3	0.213456E+06	0.131391E+07	0.320443E+00
4	0.345704E+06	0.154564E+07	0.320443E+00
5	0.345704E+06	0.154564E+07	0.320443E+00
6	0.213456E+06	0.131391E+07	0.320443E+00
7	-0.724290E+06	0.223635E+06	0.166819E+00
8	-0.371080E+06	0.523689E+06	0.183783E+00
9	-0.162378E+07	-0.103561E+07	0.263894E+00
10	-0.189947E+07	-0.144824E+07	0.263894E+00
11	-0.189947E+07	-0.144824E+07	0.263894E+00
12	-0.162378E+07	-0.103561E+07	0.263894E+00

Complex Power = (26697000.0,37852400.0) Power(kW) = 26697.0020

FAR-FIELD PRESSURES

Right-hand-side 1

	theta	phi	real	imag	dB
1	0	0	0.203828E+06	0.149905E+06	-19.25
2	5	0	0.187885E+06	0.125888E+06	-20.23
3	10	0	0.141056E+06	0.553571E+05	-23.71
4	15	0	0.662309E+05	-0.572913E+05	-28.47
5	20	0	-0.321061E+05	-0.205238E+06	-20.97
6	25	0	-0.148321E+06	-0.379923E+06	-15.11
7	30	0	-0.276163E+06	-0.571862E+06	-11.26
8	35	0	-0.409313E+06	-0.771487E+06	-8.49
9	40	0	-0.541879E+06	-0.969896E+06	-6.40
10	45	0	-0.668782E+06	-0.115945E+07	-4.78
11	50	0	-0.786007E+06	-0.133415E+07	-3.52
12	55	0	-0.890704E+06	-0.148977E+07	-2.53
13	60	0	-0.981144E+06	-0.162382E+07	-1.75

```
 14    65    0  -0.105657E+07  -0.173530E+07   -1.16
 15    70    0  -0.111699E+07  -0.182432E+07   -0.71
 16    75    0  -0.116287E+07  -0.189174E+07   -0.39
 17    80    0  -0.119492E+07  -0.193871E+07   -0.17
 18    85    0  -0.121382E+07  -0.196637E+07   -0.04
 19    90    0  -0.122006E+07  -0.197549E+07    0.00

 Source Level =247.3    DI = 2.3

condition number = 1.7E+01



TIME IN SUBROUTINES (secs)

QPTGEN              0.002
SURFACEGEN          0.000
GENSURFLDPTS        0.000
SURMATS             0.283
SURPRS              0.025
GENFARFLDPTS        0.000
FFMATS              0.325
FFPRS               0.001

Total Time =    3.097
```

Example Problem 2

Figure 8 shows the geometry for the second problem. The geometry of this problem is the same as example problem 1. In this example, three planes of reflective symmetry are used in place of rotational symmetry and velocity symmetry is used. The surface and far-field pressures should be output to a file.

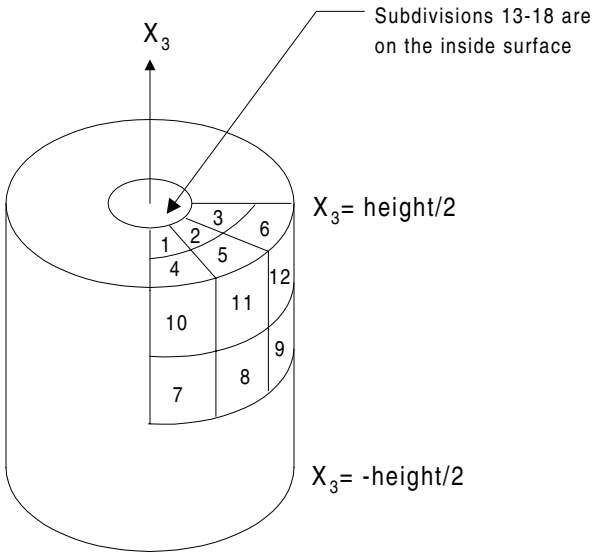

Figure 8. Geometry of example problem 2.

The following is an input file for this problem:

```
freq1=1500/(2*pi)
nblks=12
ai=1.68
ao=2.04
ht=1.2
htd2=.5*ht
reflim=pi/2
end*params

reflective,111

circ_planar, htd2
ai,ao,0,reflim
2,3,4,16
3,0,0,0

circ_cylinder, ao
0,htd2,0,reflim
2,3,4,16
3,0,0,0

circ_cylinder, ai
0,htd2,0,reflim
2,3,4,16
-3,0,0,0

end*geom

freq, freq1, freq1, 0.

rhs symmetry
vel, tc2.vel

surface,3

farfield,1,3
0,90,5,0,0,0
```

The user also needs to construct a file containing the velocity distribution. Since we chose to use the velocity symmetry, the input file only needs to contain the velocity distribution for the first octant. The velocity input file for example problem 2 is shown below:

```
1 1 1   0.097  0
1 1 2   0.097  0
1 1 3   0.097  0
1 1 4   0.097  0
1 1 5   0.097  0
1 1 6   0.097  0
1 1 7   0.971  0
1 1 8   0.971  0
1 1 9   0.971  0
1 1 10  0.971  0
```

```
1 1 11  0.971  0
1 1 12  0.971  0
1 1 13 -1.029  0
1 1 14 -1.029  0
1 1 15 -1.029  0
1 1 16 -1.029  0
1 1 17 -1.029  0
1 1 18 -1.029  0
```

As in example problem 1, the command

```
vel, tc2.vel
```

and the velocity input file can be replaced by the lines,

```
vel(1:6,1,1)=(.097,0.)
vel(7:12,1,1)=(.971,0.)
vel(13:18,1,1)=(-1.029,0.)
```

The following pages show the output that would be obtained in the .out file for this problem.

```
SURFACE GEOMETRY INPUT

region  1

neqn  nu  nv  izaxis  iquadu  iquadv
  2    2   3     3      4       16

    ul            uu            vl            vu
0.168000E+01   0.204000E+01   0.000000E+00   0.157080E+01

cc array
0.600000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  2

neqn  nu  nv  izaxis  iquadu  iquadv
  4    2   3     3      4       16

    ul            uu            vl            vu
0.000000E+00   0.600000E+00   0.000000E+00   0.157080E+01

cc array
0.204000E+01   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  3

neqn  nu  nv  izaxis  iquadu  iquadv
  4    2   3    -3      4       16

    ul            uu            vl            vu
0.000000E+00   0.600000E+00   0.000000E+00   0.157080E+01

cc array
0.168000E+01   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


#areas =  18     # interior points = 0
```

```
Memory Usage (bytes)

quadrature points        36864
surface pressures        144
far-field pressures      152
near-field pressures     0
point source pressures   0
external pressures       0
dpdn                     144
surface field points     1728
far-field points         1824
near-field points        0
A,B matrices             43776
total                    84632

number of surface row blocks    = 1
number of far-field row blocks  = 1
number of near-field row blocks = 0



Frequency = 238.732407

SURFACE PRESSURES

Block 1   Right-hand-side 1

          real           imag           area
 1    -0.724311E+06   0.223842E+06   0.166819E+00
 2    -0.724324E+06   0.223837E+06   0.166819E+00
 3    -0.724317E+06   0.223837E+06   0.166819E+00
 4    -0.371021E+06   0.523867E+06   0.183783E+00
 5    -0.371028E+06   0.523871E+06   0.183783E+00
 6    -0.371025E+06   0.523864E+06   0.183783E+00
 7     0.345971E+06   0.154586E+07   0.320443E+00
 8     0.345948E+06   0.154583E+07   0.320443E+00
 9     0.345965E+06   0.154587E+07   0.320443E+00
10     0.213674E+06   0.131412E+07   0.320443E+00
11     0.213658E+06   0.131409E+07   0.320443E+00
12     0.213669E+06   0.131412E+07   0.320443E+00
13    -0.189997E+07  -0.144817E+07   0.263894E+00
14    -0.189992E+07  -0.144813E+07   0.263894E+00
15    -0.189995E+07  -0.144816E+07   0.263894E+00
16    -0.162418E+07  -0.103550E+07   0.263894E+00
17    -0.162414E+07  -0.103547E+07   0.263894E+00
18    -0.162416E+07  -0.103550E+07   0.263894E+00


Complex Power = (26706058.0,37854312.0)    Power(kW) = 26706.0586
```

```
 FAR-FIELD PRESSURES

 Right-hand-side 1

     theta  phi     real           imag          dB
  1      0    0   0.203834E+06   0.149854E+06  -19.26
  2      5    0   0.187886E+06   0.125835E+06  -20.23
  3     10    0   0.141041E+06   0.552972E+05  -23.71
  4     15    0   0.661887E+05  -0.573618E+05  -28.47
  5     20    0  -0.321830E+05  -0.205322E+06  -20.96
  6     25    0  -0.148438E+06  -0.380022E+06  -15.11
  7     30    0  -0.276324E+06  -0.571978E+06  -11.26
  8     35    0  -0.409520E+06  -0.771618E+06   -8.49
  9     40    0  -0.542130E+06  -0.970041E+06   -6.40
 10     45    0  -0.669075E+06  -0.115961E+07   -4.78
 11     50    0  -0.786339E+06  -0.133431E+07   -3.52
 12     55    0  -0.891068E+06  -0.148994E+07   -2.53
 13     60    0  -0.981535E+06  -0.162400E+07   -1.75
 14     65    0  -0.105699E+07  -0.173548E+07   -1.16
 15     70    0  -0.111742E+07  -0.182451E+07   -0.71
 16     75    0  -0.116332E+07  -0.189193E+07   -0.39
 17     80    0  -0.119537E+07  -0.193890E+07   -0.17
 18     85    0  -0.121428E+07  -0.196655E+07   -0.04
 19     90    0  -0.122052E+07  -0.197568E+07    0.00

 Source Level =247.3    DI = 2.3

condition number = 1.9E+01



TIME IN SUBROUTINES (secs)

QPTGEN              0.018
SURFACEGEN          0.000
GENSURFLDPTS        0.000
SURMATS             0.361
SURPRS              0.004
GENFARFLDPTS        0.000
FFMATS              0.367
FFPRS               0.001

Total Time =    2.782
```

Example Problem 3

The geometry for this problem is the same as for example problem 1. This example illustrates the use of a point source check. A point source check can be used to determine the adequacy of the surface discretization as well as the need for and the placement of interior points. The basic idea of a point source check is to consider the boundary surface as an imaginary surface in an infinite fluid medium containing a finite number of point sources. The surface pressures are calculated using the CHIEF formulation with a surface velocity distribution produced by the point sources and compared with the exact pressures due to

the point sources. If the point sources are inside the body, then the CHIEF computation is like a radiation problem. If the point sources are outside the body, then the CHIEF computation is like a scattering problem. For this example, we will use one point source located at the origin, having a complex weight of 1+i0. (This point is outside the body.) For thin bodies like this free-flooded cylinder, it is usually better to place the point source outside the body, since point sources inside the body are so close to the surface that they will produce a surface velocity distribution that varies more rapidly than the ones likely to be used in practice. The input file for this example is shown below:

```
freq1=1500/(2*pi)
nblks=12
ai=1.68
ao=2.04
ht=1.2
htd2=.5*ht
rotlim=pi/nblks
end*params

rotational,nblks

circ_planar, htd2
ai,ao,-rotlim,rotlim
2,1,4,16
3,0,0,0

circ_cylinder, ao
-htd2,htd2,-rotlim,rotlim
4,1,4,16
3,0,0,0

circ_planar, -htd2
ai,ao,-rotlim,rotlim
2,1,4,16
-3,0,0,0

circ_cylinder, ai
-htd2,htd2,-rotlim,rotlim
4,1,4,16
-3,0,0,0

end*geom

freq, freq1, freq1, 0.
pscheck, 1,1
0,0,0,1,0,1
```

The following pages show the output that would be obtained in the .out file for this problem.

```
SURFACE GEOMETRY INPUT

region  1

neqn  nu  nv  izaxis  iquadu  iquadv
  2    2   1     3       4       16

    ul            uu            vl            vu
0.168000E+01   0.204000E+01   -.261799E+00   0.261799E+00

cc array
0.600000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  2

neqn  nu  nv  izaxis  iquadu  iquadv
  4    4   1     3       4       16

    ul            uu            vl            vu
-.600000E+00   0.600000E+00   -.261799E+00   0.261799E+00

cc array
0.204000E+01   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  3

neqn  nu  nv  izaxis  iquadu  iquadv
  2    2   1    -3       4       16

    ul            uu            vl            vu
0.168000E+01   0.204000E+01   -.261799E+00   0.261799E+00

cc array
-.600000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  4

neqn  nu  nv  izaxis  iquadu  iquadv
  4    4   1    -3       4       16
```

```
    ul             uu             vl             vu
-.600000E+00   0.600000E+00   -.261799E+00   0.261799E+00

cc array
0.168000E+01   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


#areas =  12     # interior points = 0




 Memory Usage (bytes)

 quadrature points        24576
 surface pressures        96
 far-field pressures      0
 near-field pressures     0
 point source pressures   96
 external pressures       0
 dpdn                     96
 surface field points     1728
 far-field points         0
 near-field points        0
 A,B matrices             27648
 total                    54240

 number of surface row blocks    = 1
 number of far-field row blocks  = 0
 number of near-field row blocks = 0



 Frequency = 238.732407

 POINT SOURCE SURFACE PRESSURES

 Block 1
                                                                    Error
          preal          pimag          psreal         psimag        mag%   ang(deg)
  1    -0.159219E+00  -0.485809E+00  -0.150085E+00  -0.488410E+00      0.06    -1.06
  2    -0.217887E+00  -0.415344E+00  -0.211734E+00  -0.417422E+00      0.21    -0.79
  3    -0.228535E+00  -0.392456E+00  -0.226435E+00  -0.397090E+00     -0.65    -0.52
  4    -0.214423E+00  -0.411792E+00  -0.213385E+00  -0.415221E+00     -0.55    -0.31
  5    -0.214423E+00  -0.411792E+00  -0.213385E+00  -0.415221E+00     -0.55    -0.31
  6    -0.228535E+00  -0.392456E+00  -0.226435E+00  -0.397090E+00     -0.65    -0.52
  7    -0.159219E+00  -0.485809E+00  -0.150085E+00  -0.488410E+00      0.06    -1.06
  8    -0.217888E+00  -0.415344E+00  -0.211734E+00  -0.417422E+00      0.21    -0.79
```

```
  9    -0.909805E-01  -0.550150E+00  -0.920394E-01  -0.541286E+00      1.56     0.26
 10    -0.661868E-01  -0.570716E+00  -0.654635E-01  -0.562361E+00      1.48     0.02
 11    -0.661868E-01  -0.570716E+00  -0.654635E-01  -0.562361E+00      1.48     0.02
 12    -0.909805E-01  -0.550150E+00  -0.920394E-01  -0.541286E+00      1.56     0.26


condition number = 1.7E+01



TIME IN SUBROUTINES (secs)

QPTGEN             0.002
SURFACEGEN         0.000
GENSURFLDPTS       0.000
SURMATS            0.321
SURPRS             0.004

Total Time =    1.733
```

Example Problem 4

The geometry for this example problem is the same as for example problem 1. In this example, the radiation impedance matrix relating surface forces and velocities and the field matrix relating far-field pressures and surface velocities are computed. It is assumed that the velocities have the same symmetry as the geometry. The results are to be written to an output file. The input file for this example problem is given below:

```
freq1=1500/(2*pi)
nblks=12
ai=1.68
ao=2.04
ht=1.2
htd2=.5*ht
rotlim=pi/nblks
end*params

rotational,nblks

circ_planar, htd2
ai,ao,-rotlim,rotlim
2,1,4,16
3,0,0,0

circ_cylinder, ao
-htd2,htd2,-rotlim,rotlim
4,1,4,16
3,0,0,0

circ_planar, -htd2
ai,ao,-rotlim,rotlim
2,1,4,16
```

```
-3,0,0,0

circ_cylinder, ai
-htd2,htd2,-rotlim,rotlim
4,1,4,16
-3,0,0,0

end*geom

freq, freq1, freq1, 0.

impedance matrix,1,1

farfield pressures,1,1,19
0,90,5,0,0,0
```

Example Problem 5

In this example problem, we consider a uniformly pulsating sphere of radius 1 m with a radial velocity of 1+i0 m/sec. Figure 9 shows the geometry and breakup for this example problem. This example problem shows the need and use of interior points. We consider two frequencies corresponding to $ka = \pi/3$ and $ka = \pi$. The value $ka = \pi$ corresponds to an interior resonance where interior points are needed. The value $ka = \pi/3$ is well below the first interior resonance and, thus, does not require interior points. Let us first calculate the far-field pressures for these two frequencies with no interior points.

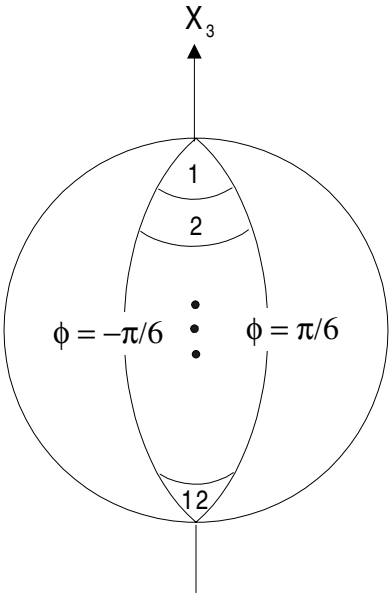

Figure 9. Geometry for example problem 5.

The input file for this calculation is shown below:

```
ka1=pi/3
ka2=pi
a=1.0
c=1500
fu=ka2*c/(2*pi)
fl=fu/3
finc=fu-fl
uu=pi/6
end*params

rot,6
sphere,1.0
0,pi,-uu,uu
12,1,4,16
3,0,0,0
end*geom

frequency,fl,fu,finc
```

```
!interior points,3
!0,0,0

rhs
vel(1:12,1,1)=(1.0,0.)
! or read velocities from a file with:  velocity,tc5.vel

surface pressures,3

far-field pressures,1,1,4
0,180,30,0,0,0
```

The following pages show the output that would be obtained in the .out file for this problem.

```
SURFACE GEOMETRY INPUT

region  1

neqn  nu  nv  izaxis  iquadu  iquadv
  6   12   1     3       4       16

    ul             uu             vl             vu
0.000000E+00   0.314159E+01   -.523599E+00   0.523599E+00

cc array
0.100000E+01   0.100000E+01   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


#areas =  12     # interior points = 0




 Memory Usage (bytes)

 quadrature points        24576
 surface pressures        96
 far-field pressures      56
 near-field pressures     0
 point source pressures   0
 external pressures       0
 dpdn                     96
 surface field points     864
 far-field points         504
 near-field points        0
 A,B matrices             13824
 total                    40016

 number of surface row blocks    = 1
 number of far-field row blocks  = 1
 number of near-field row blocks = 0



 Frequency = 250.000000

 SURFACE PRESSURES

 Block 1   Right-hand-side 1

           real           imag           area
  1     0.784838E+06   0.748495E+06   0.356824E-01
  2     0.784546E+06   0.749223E+06   0.104615E+00
```

```
 3     0.784558E+06   0.749222E+06   0.166419E+00
 4     0.784563E+06   0.749218E+06   0.216882E+00
 5     0.784561E+06   0.749202E+06   0.252564E+00
 6     0.784564E+06   0.749192E+06   0.271035E+00
 7     0.784579E+06   0.749205E+06   0.271035E+00
 8     0.784564E+06   0.749206E+06   0.252564E+00
 9     0.784554E+06   0.749210E+06   0.216882E+00
10     0.784562E+06   0.749225E+06   0.166419E+00
11     0.784537E+06   0.749215E+06   0.104615E+00
12     0.784839E+06   0.748497E+06   0.356824E-01

 Complex Power = (9859209.00,9414540.00)    Power(kW) = 9859.20898



 FAR-FIELD PRESSURES

 Right-hand-side 1

    theta  phi     real           imag          dB
 1      0    0  -0.256553E+06   0.105407E+07    0.00
 2     30    0  -0.256552E+06   0.105407E+07    0.00
 3     60    0  -0.256549E+06   0.105406E+07    0.00
 4     90    0  -0.256547E+06   0.105405E+07    0.00
 5    120    0  -0.256549E+06   0.105406E+07    0.00
 6    150    0  -0.256552E+06   0.105407E+07    0.00
 7    180    0  -0.256553E+06   0.105407E+07    0.00

 Source Level =240.7    DI = 0.0

condition number = 3.2E+00



 Frequency = 750.000000

 SURFACE PRESSURES

 Block 1   Right-hand-side 1

          real           imag           area
 1    -0.189111E+06   0.184947E+08   0.356824E-01
 2    -0.189098E+06   0.184909E+08   0.104615E+00
 3    -0.189162E+06   0.184911E+08   0.166419E+00
 4    -0.189192E+06   0.184912E+08   0.216882E+00
 5    -0.189173E+06   0.184912E+08   0.252564E+00
 6    -0.189140E+06   0.184913E+08   0.271035E+00
 7    -0.189130E+06   0.184916E+08   0.271035E+00
 8    -0.189151E+06   0.184913E+08   0.252564E+00
 9    -0.189168E+06   0.184910E+08   0.216882E+00
10    -0.189151E+06   0.184912E+08   0.166419E+00
11    -0.189100E+06   0.184907E+08   0.104615E+00
```

```
 12    -0.189128E+06   0.184948E+08   0.356824E-01


 Complex Power = (-2376929.25,232369088.)    Power(kW) = -2376.92944



 FAR-FIELD PRESSURES

 Right-hand-side 1

     theta  phi     real           imag          dB
  1      0    0   0.189201E+06  -0.184912E+08    0.00
  2     30    0   0.189158E+06  -0.184913E+08    0.00
  3     60    0   0.189144E+06  -0.184914E+08    0.00
  4     90    0   0.189155E+06  -0.184913E+08    0.00
  5    120    0   0.189151E+06  -0.184914E+08    0.00
  6    150    0   0.189140E+06  -0.184913E+08    0.00
  7    180    0   0.189121E+06  -0.184912E+08    0.00

 Source Level =265.3    DI =-NaN

condition number = 6.5E+05



TIME IN SUBROUTINES (secs)

QPTGEN             0.002
SURFACEGEN         0.000
GENSURFLDPTS       0.000
SURMATS            0.309
SURPRS             0.004
GENFARFLDPTS       0.000
FFMATS             0.127
FFPRS              0.000

Total Time =    2.003
```

Looking at the directivity index (DI) corresponding to $ka = \pi$ shows this answer is incorrect. The condition number for this case is very high, showing that the A matrix is very ill-conditioned. Let us now rerun the same problem with three interior points—one located at (0,0,0) and the other two generated automatically. The input file for this calculation is the same as the previous one, with the interior point control lines uncommented. The following pages show the output that would be obtained in the .out file for this example problem. It can be easily verified that the answers for this case are correct.

```
SURFACE GEOMETRY INPUT

region  1

neqn  nu  nv  izaxis  iquadu  iquadv
  6   12   1     3      4       16

    ul             uu             vl             vu
0.000000E+00   0.314159E+01   -.523599E+00   0.523599E+00

cc array
0.100000E+01   0.100000E+01   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


#areas =  12     # interior points = 3


 INTERIOR POINTS

          x1       x2       x3
  1      0.00     0.00     0.00
  2      0.42     0.00    -0.33
  3      0.26     0.00     0.63



 Memory Usage (bytes)

 quadrature points        24576
 surface pressures        120
 far-field pressures      56
 near-field pressures     0
 point source pressures   0
 external pressures       0
 dpdn                     96
 surface field points     864
 far-field points         504
 near-field points        0
 A,B matrices             17280
 total                    43496

 number of surface row blocks    = 1
 number of far-field row blocks  = 1
 number of near-field row blocks = 0



 Frequency = 250.000000
```

```
 FAR-FIELD PRESSURES

 Right-hand-side 1

     theta  phi     real           imag          dB
  1      0    0  -0.256553E+06   0.105407E+07    0.00
  2     30    0  -0.256552E+06   0.105407E+07    0.00
  3     60    0  -0.256549E+06   0.105406E+07    0.00
  4     90    0  -0.256547E+06   0.105405E+07    0.00
  5    120    0  -0.256547E+06   0.105406E+07    0.00
  6    150    0  -0.256550E+06   0.105407E+07    0.00
  7    180    0  -0.256551E+06   0.105407E+07    0.00

 Source Level =240.7    DI = 0.0

condition number = 1.4E+00



 Frequency = 750.000000


 FAR-FIELD PRESSURES

 Right-hand-side 1

     theta  phi     real           imag          dB
  1      0    0  -0.136197E+07  -0.433644E+06    0.00
  2     30    0  -0.136198E+07  -0.433520E+06    0.00
  3     60    0  -0.136199E+07  -0.433473E+06    0.00
  4     90    0  -0.136198E+07  -0.433616E+06    0.00
  5    120    0  -0.136200E+07  -0.433487E+06    0.00
  6    150    0  -0.136200E+07  -0.433543E+06    0.00
  7    180    0  -0.136199E+07  -0.433667E+06    0.00

 Source Level =243.1    DI = 0.0

condition number = 1.1E+00



TIME IN SUBROUTINES (secs)

QPTGEN             0.002
SURFACEGEN         0.000
GENSURFLDPTS       0.000
SURMATS            0.274
SURPRS             0.004
GENFARFLDPTS       0.000
FFMATS             0.113
FFPRS              0.000

Total Time =    1.994
```

Example Problem 6

In this example problem, we consider the plane-wave scattering from a rigid sphere of radius 5 m and $ka = 5$, using two methods of geometry generation. Both methods use 6-th order rotational symmetry with 20 subdivisions per symmetry block. The first method generates the geometry using the sphere equations. The second method generates the geometry using axisymmetric finite element inputs. Figure 10 shows the target strength versus angle for both methods of geometry generation along with the results of an analytic series solution.

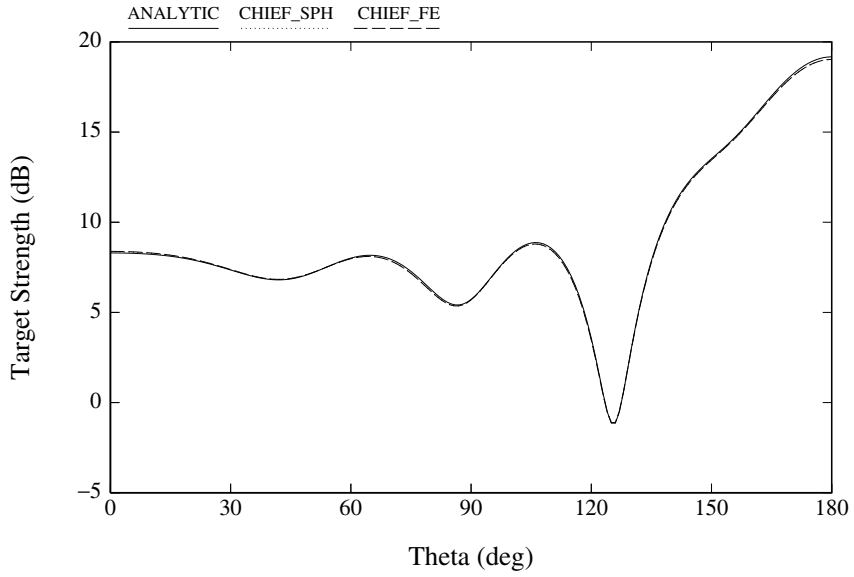

Figure 10. Target strength versus angle for a sphere ($ka = 5$).

The contents of the input file for the spherical geometry input is shown below:

```
nblks=6
c=1500
a=5
rka=5
freq=rka*c/(2*pi*a)
rotlim=pi/nblks

end*params

rot,nblks
sphere,a
0,pi,-rotlim,rotlim
20,1,4,16
3,0,0,0

end*geom

frequency,freq,freq,0
surface pressures,1
plane wave,1,1,0,0
```

```
far-field,1,3
0,180,2,0,0,0
```

The input file for the axisymmetric finite element input is exactly the same as above except that the geometry section is replaced by

```
rot,nblks
fE_axisymmetric, tc6.crd, tc6.elt, 4, 16
3
end*geom
```

The contents of the nodal coordinate file tc6.crd are shown below.

```
 1     0.0000E+00 0.5000E+01
 2     0.3923E+00 0.4985E+01
 3     0.7822E+00 0.4938E+01
 4     0.1167E+01 0.4862E+01
 5     0.1545E+01 0.4755E+01
 6     0.1913E+01 0.4619E+01
 7     0.2270E+01 0.4455E+01
 8     0.2612E+01 0.4263E+01
 9     0.2939E+01 0.4045E+01
10     0.3247E+01 0.3802E+01
11     0.3536E+01 0.3536E+01
12     0.3802E+01 0.3247E+01
13     0.4045E+01 0.2939E+01
14     0.4263E+01 0.2612E+01
15     0.4455E+01 0.2270E+01
16     0.4619E+01 0.1913E+01
17     0.4755E+01 0.1545E+01
18     0.4862E+01 0.1167E+01
19     0.4938E+01 0.7822E+00
20     0.4985E+01 0.3923E+00
21     0.5000E+01 -.2186E-06
22     0.4985E+01 -.3923E+00
23     0.4938E+01 -.7822E+00
24     0.4862E+01 -.1167E+01
25     0.4755E+01 -.1545E+01
26     0.4619E+01 -.1913E+01
27     0.4455E+01 -.2270E+01
28     0.4263E+01 -.2612E+01
29     0.4045E+01 -.2939E+01
30     0.3802E+01 -.3247E+01
31     0.3536E+01 -.3536E+01
32     0.3247E+01 -.3802E+01
33     0.2939E+01 -.4045E+01
34     0.2612E+01 -.4263E+01
35     0.2270E+01 -.4455E+01
36     0.1913E+01 -.4619E+01
37     0.1545E+01 -.4755E+01
38     0.1167E+01 -.4862E+01
```

```
 39     0.7822E+00 -.4938E+01
 40     0.3923E+00 -.4985E+01
 41     -.4371E-06 -.5000E+01
```

The contents of the element file tc6.elt is shown below.

```
 1    2    3
 3    4    5
 5    6    7
 7    8    9
 9   10   11
11   12   13
13   14   15
15   16   17
17   18   19
19   20   21
21   22   23
23   24   25
25   26   27
27   28   29
29   30   31
31   32   33
33   34   35
35   36   37
37   38   39
39   40   41
```

Example Problem 7

In this example problem, we illustrate the plane infinite rigid baffle option. The problem is to compute the radiation impedance of a half-wavelength diameter circular piston in a plane infinite rigid baffle. The impedance is to be calculated in CHIEF using four blocks of rotational symmetry with four radial subdivisions per block. The analytic solution for this problem gave the following result:

$$z = .952057 \times 10^7 + i.777030 \times 10^7.$$

To calculate the impedance with CHIEF we use a uniform velocity distribution of 1+i0. The impedance is then equal to the complex power given in the .out file. The value computed with this breakup is

$$z = .957883 \times 10^7 + i.780259 \times 10^7,$$

which agrees with the analytic solution to within about 0.5%. The input file for this run is shown below:

```
freq=1500/(2*pi)
nblks=4
a=pi/2
rotlim=pi/nblks
end*params

rot,nblks
circ_planar,0
0,a,-rotlim,rotlim
4,1,2,6
3
end*geom

frequency,freq,freq,0

inf baffle

rhs symmetry

velocity,tc7.vel
! or vel(1:4,1,1)=1.0

surface,3
```

The velocity file tc7.vel is as follows

```
1  1  1  1.  0.
1  1  2  1.  0.
1  1  3  1.  0.
1  1  4  1.  0.
```

Example Problem 8

This example problem shows how to compute near-field pressures. The radiating surface consists of a circular cylinder with flat circular end pieces. The cylinder radius is 1 m and the height is 2 m. We choose the origin at the center of the cylinder and let the z-axis coincide with the cylinder axis. The end surface at $z = 1$ is given a velocity of 1+i0 and the rest of the surface has zero velocity. We want to compute the pressure at the two points (0,0,1.25) and (0,0,1.5) on the axis near the piston. In this example, we also set the available RAM memory to be only 400 KB to force the program to use intermediate disk storage for the large matrices. The input file for this problem is shown below:

```
freql=100
frequ=1500
freqinc=1400
nblks=12
ao=1.0
ht=2.0
htd2=.5*ht
rotlim=pi/nblks
end*params

rotational,nblks

circ_planar, htd2
0,ao,-rotlim,rotlim
16,1,4,16
3

circ_cylinder, ao
-htd2,htd2,-rotlim,rotlim
32,1,4,16
3

circ_planar, -htd2
0,ao,-rotlim,rotlim
16,1,4,16
-3,0,0,0

end*geom

memory,400000  ! intentionally set low to force use of intermediate disk storage

freq, freql, frequ, freqinc
rhs symmetry
vel(1:16,1,1)=1.0

surface pressures

nearfield pressures,2,3
0, 0, 1.25
0, 0, 1.5
```

Note that you only have to input the nonzero velocities. The results from the file tc8.out are shown on the following page.

```
SURFACE GEOMETRY INPUT

region  1

neqn  nu  nv  izaxis  iquadu  iquadv
  2   16   1     3       4       16

    ul            uu            vl            vu
0.000000E+00   0.100000E+01   -.261799E+00   0.261799E+00

cc array
0.100000E+01   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  2

neqn  nu  nv  izaxis  iquadu  iquadv
  4   32   1     3       4       16

    ul            uu            vl            vu
-.100000E+01   0.100000E+01   -.261799E+00   0.261799E+00

cc array
0.100000E+01   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  3

neqn  nu  nv  izaxis  iquadu  iquadv
  2   16   1    -3       4       16

    ul            uu            vl            vu
0.000000E+00   0.100000E+01   -.261799E+00   0.261799E+00

cc array
-.100000E+01   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


#areas =  64     # interior points = 0
```

```
 Memory Usage (bytes)

 quadrature points        131072
 surface pressures        512
 far-field pressures      0
 near-field pressures     16
 point source pressures   0
 external pressures       0
 dpdn                     512
 surface field points     9216
 far-field points         0
 near-field points        288
 A,B matrices             258048
 total                    399664

 number of surface row blocks    = 4
 number of far-field row blocks  = 0
 number of near-field row blocks = 1



 Frequency = 100.000000


 NEAR-FIELD PRESSURES

 Right-hand-side 1

         x1       x2       x3       real           imag
  1      0.00     0.00     1.25     0.643337E+05   0.389493E+06
  2      0.00     0.00     1.50     0.643251E+05   0.290717E+06

condition number = 1.5E+01



 Frequency = 1500.00000


 NEAR-FIELD PRESSURES

 Right-hand-side 1

         x1       x2       x3       real           imag
  1      0.00     0.00     1.25    -0.163006E+07  -0.110086E+07
  2      0.00     0.00     1.50    -0.267511E+07   0.117691E+07

condition number = 1.5E+02



TIME IN SUBROUTINES (secs)
```

```
QPTGEN              0.007
SURFACEGEN          0.001
GENSURFLDPTS        0.000
SURMATS            10.496
SURPRS              0.168
GENNEARFLDPTS       0.000
NFMATS              0.312
NFPRS               0.001

Total Time =   12.420
```

Example Problem 9

This example problem is like example problem 2 except that both surface and far-field A and B matrices are output. In this case, it is not necessary to input a velocity distribution. The control section of the input file for this example is shown below:

```
freq1=1500/(2*pi)
nblks=12
ai=1.68
ao=2.04
ht=1.2
htd2=.5*ht
reflim=pi/2
end*params

reflective,111

circ_planar, htd2
ai,ao,0,reflim
2,3,4,16
3,0,0,0

circ_cylinder, ao
0,htd2,0,reflim
2,3,4,16
3,0,0,0

circ_cylinder, ai
0,htd2,0,reflim
2,3,4,16
-3,0,0,0

end*geom

freq, freq1, freq1, 0.
a_b matrices
surface matrices
farfield matrices,1,0
0,90,5,0,0,0
```

Example Problem 10

The geometry for this problem consists of a cube having sides of length 2. The corners of the cube are at (1,1,1), (-1,1,1), (-1,-1,1), (1,-1,1), (1,1,-1),(-1,1,-1), (-1,-1,-1), and (1,-1,-1). The geometry is input using rectangular finite elements. The purpose of this example is to generate a coupling matrix relating finite element forces and chief pressures. The input file for this example is shown below:

```
end*param

none

fe_3d,tc10.nde,tc10.elt,4,4
3

end*geom

couple,1
```

The node file tc10.nde contains

```
56   1.0   1.0   1.0
22  -1.0   1.0   1.0
 3  -1.0  -1.0   1.0
41   1.0  -1.0   1.0
57   1.0   1.0  -1.0
16  -1.0   1.0  -1.0
27  -1.0  -1.0  -1.0
48   1.0  -1.0  -1.0
```

and the element file tc10.elt contains

```
56   22    3  41
56   57   16  22
16   27    3  22
27   48   41   3
48   57   56  41
57   48   27  16
```

Note that CHIEF renumbers the nodes from 56, 22, 3, 41, 57, 16, 27, 48 to 1, 2, 3, 4, 5, 6, 7, 8. The output for this example is shown below:

```
SURFACE GEOMETRY INPUT

region  1

neqn  nu  nv  izaxis  iquadu  iquadv
 10    1   1     3      4        4

    ul             uu             vl             vu
-.100000E+01   0.100000E+01   -.100000E+01   0.100000E+01
```

```
cc array
0.100000E+01   0.200000E+01   0.300000E+01   0.400000E+01   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  2

neqn  nu  nv  izaxis  iquadu  iquadv
 10    1   1     3      4        4

    ul            uu            vl            vu
-.100000E+01   0.100000E+01   -.100000E+01   0.100000E+01

cc array
0.100000E+01   0.500000E+01   0.600000E+01   0.200000E+01   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  3

neqn  nu  nv  izaxis  iquadu  iquadv
 10    1   1     3      4        4

    ul            uu            vl            vu
-.100000E+01   0.100000E+01   -.100000E+01   0.100000E+01

cc array
0.600000E+01   0.700000E+01   0.300000E+01   0.200000E+01   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  4

neqn  nu  nv  izaxis  iquadu  iquadv
 10    1   1     3      4        4

    ul            uu            vl            vu
-.100000E+01   0.100000E+01   -.100000E+01   0.100000E+01

cc array
0.700000E+01   0.800000E+01   0.400000E+01   0.300000E+01   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
```

```
0.000000E+00   0.000000E+00   0.000000E+00


region  5

neqn  nu  nv  izaxis  iquadu  iquadv
 10    1   1     3      4        4

    ul             uu             vl             vu
-.100000E+01   0.100000E+01   -.100000E+01   0.100000E+01

cc array
0.800000E+01   0.500000E+01   0.100000E+01   0.400000E+01   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


region  6

neqn  nu  nv  izaxis  iquadu  iquadv
 10    1   1     3      4        4

    ul             uu             vl             vu
-.100000E+01   0.100000E+01   -.100000E+01   0.100000E+01

cc array
0.500000E+01   0.800000E+01   0.700000E+01   0.600000E+01   0.000000E+00
0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00

trns
0.000000E+00   0.000000E+00   0.000000E+00


#areas =   6     # interior points = 0




FE Node Numbers

 56 22 3 41 57 16 27 48


Coupling Matrix

   0.000000   0.000000   0.000000   0.000000   0.250000   0.000000
   0.000000   0.250000   0.000000   0.000000   0.000000   0.000000
   0.250000   0.000000   0.000000   0.000000   0.000000   0.000000
   0.000000   0.000000  -0.250000   0.000000   0.000000   0.000000
   0.000000   0.250000   0.000000   0.000000   0.000000   0.000000
   0.250000   0.000000   0.000000   0.000000   0.000000   0.000000
```

```
0.000000   0.000000  -0.250000   0.000000   0.000000   0.000000
0.000000   0.000000   0.000000  -0.250000   0.000000   0.000000
0.250000   0.000000   0.000000   0.000000   0.000000   0.000000
0.000000   0.000000   0.000000   0.000000   0.250000   0.000000
0.000000   0.000000   0.000000  -0.250000   0.000000   0.000000
0.250000   0.000000   0.000000   0.000000   0.000000   0.000000
0.000000   0.000000   0.000000   0.000000   0.250000   0.000000
0.000000   0.250000   0.000000   0.000000   0.000000   0.000000
0.000000   0.000000   0.000000   0.000000   0.000000  -0.250000
0.000000   0.000000  -0.250000   0.000000   0.000000   0.000000
0.000000   0.250000   0.000000   0.000000   0.000000   0.000000
0.000000   0.000000   0.000000   0.000000   0.000000  -0.250000
0.000000   0.000000  -0.250000   0.000000   0.000000   0.000000
0.000000   0.000000   0.000000  -0.250000   0.000000   0.000000
0.000000   0.000000   0.000000   0.000000   0.000000  -0.250000
0.000000   0.000000   0.000000   0.000000   0.250000   0.000000
0.000000   0.000000   0.000000  -0.250000   0.000000   0.000000
0.000000   0.000000   0.000000   0.000000   0.000000  -0.250000
```

In this case, an analytical evaluation of the integrals verifies that the resulting coupling matrix is correct.

This page intentionally left blank

REFERENCES

Baker, B.B. and E.T. Copson. 1950. *Mathematical Theory of Huygen's Principle*. Oxford University Press, Oxford, U.K.

Benthien, G., D. Barach, and D. Gillette. 1988. "CHIEF Users Manual." NOSC* Technical Document 970, Revision 1 (Sep). SSC San Diego, CA.

Benthien, G.W. and H.A. Schenck. 1997. "Nonexistence and Nonuniqueness Problems Associated with Integral Equation Methods in Acoustics," *Computers & Structures*, vol. 65, pp. 295–305.

Benthien, G. and D. Barach. 1998. "Handbook of Acoustic Projector Technology." SSC San Diego Technical Document 2980, Revision 1 (Sep). SSC San Diego, CA.

Benthien, G., D. Barach, S. Hobbs. 2000. "CHIEF 2000 Users Manual." SSC San Diego Technical Document 3104 (Mar). SSC San Diego, CA.

*Now SSC San Diego

This page intentionally left blank

APPENDIX: BASIC CHIEF THEORY

The principal quantity of interest in classical (linear) acoustics is the acoustic pressure $p(x,t)$ that is a function of position and time. The acoustic pressure is the excess pressure over the hydrostatic pressure due to compressional waves in the medium. The acoustic pressure $p(x,t)$ satisfies the wave equation,

$$\triangle p = \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2}. \tag{1}$$

Here, $\triangle$ is the spatial Laplacian operator and c is the speed of sound in the fluid medium. In acoustics, it is often more convenient to work in the frequency domain than in the time domain. By frequency domain we mean the Fourier transform domain. The Fourier transform P of p is defined by the pair of relations,

$$p(x,t) = \int_{-\infty}^{\infty} P(x,\omega)e^{i\omega t}\,df \qquad \text{(inverse Fourier transform)}, \tag{2a}$$

where

$$P(x,\omega) = \int_{-\infty}^{\infty} p(x,t)e^{-i\omega t}\,dt \qquad \text{(Fourier transform)}. \tag{2b}$$

The transform variable ω is called the angular frequency, and the related quantity $f = \omega/(2\pi)$ is called the frequency. Note that whereas the pressure p is a real quantity, the transformed pressure P is, in general, complex. The Fourier-transformed pressure $P(x,\omega)$ satisfies the reduced wave equation,

$$\triangle P + k^2 P = 0, \tag{3}$$

where $k = \omega/c$ is the acoustic wavenumber.

In scattering problems, the incident pressure p^{inc} is the pressure that would be present if the scattering object were removed. If the source of the incident pressure is far from the scattering object, p^{inc} is usually assumed to be a plane wave because the wave is approximately planar in the vicinity of the scatterer. If p is the total acoustic pressure present in a scattering problem, then the scattered pressure p^{s} is defined by

$$p^{\text{s}} = p - p^{\text{inc}}. \tag{4}$$

Clearly, the incident and scattered pressures also satisfy the acoustic wave equation.

To uniquely determine the pressure, it is necessary to specify some initial and boundary conditions in addition to the requirement that the wave equation be satisfied. In the time domain, we usually specify initial conditions throughout the region of interest for p and $\dot{p}$ as well as boundary conditions on either p or its normal derivative q. In the time domain, these boundary and initial conditions are sufficient to uniquely determine p for exterior (unbounded region) problems. In the frequency domain, we usually specify either the pressure transform P or the transform Q of the normal derivative of p on the boundary. For exterior problems in the frequency domain, it is necessary to supplement the boundary conditions with a radiation condition at infinity to eliminate the possibility of waves coming in from infinity. This condition usually takes the form,

$$\lim_{r\to\infty} r\left(\frac{\partial P}{\partial r} + ikP\right) = 0, \tag{5}$$

due to Sommerfeld. In effect, this condition says that the pressure field looks locally like an outgoing spherical wave at large distances from the sound source. This condition can be eliminated if the fluid is assumed slightly lossy (complex sound velocity). If the incident field in a scattering problem is considered a plane wave, then neither the incident pressure nor the total pressure satisfies the outgoing radiation condition. Therefore, in scattering problems, the outgoing radiation condition is applied to the scattered pressure. The normal derivative of pressure Q is related to the normal surface acceleration a by $Q = -\rho a$ and to the normal surface velocity v by $Q = -i\omega\rho v$.

The CHIEF method for numerically solving acoustic problems is based on the reduction of a boundary-value problem for the wave equation in the frequency domain to an equivalent integral equation over the boundary surface. This integral equation is obtained from the Helmholtz integral relations:

$$\int_S \left[G(x,\xi)Q(\xi) - P(\xi)\frac{\partial G}{\partial n}(x,\xi)\right] dS(\xi) = \begin{cases} -P(x) & x \text{ outside } S \\ -\frac{1}{2}P(x) & x \text{ on } S \\ 0 & x \text{ inside } S, \end{cases} \tag{6}$$

where $G(x,\xi)$ is the free-space Green's function defined by

$$G(x,\xi) = \frac{e^{-ik|x-\xi|}}{4\pi|x-\xi|}, \tag{7}$$

n is the outward unit normal to S, and $|x-\xi|$ is the distance between x and ξ. These relations come from the divergence theorem, the fact that P and G satisfy the wave equation, and the nature of the singularity of G. A good discussion of these integral relations is contained in [Baker and Copson, 1950]. If the normal derivative Q is specified on the boundary S, then the middle relation in equation (6) provides an integral equation for the surface pressure P, i.e.,

$$\tfrac{1}{2}P(\zeta) - \int_S P(\xi)\frac{\partial G}{\partial n}(\zeta,\xi)\, dS(\xi) = -\int_S G(\zeta,\xi)Q(\xi)\, dS(\xi) \qquad \zeta \text{ on } S. \tag{8}$$

Once this integral equation is solved for the surface pressure, the pressure at any point in the exterior region can be obtained using the first relation in equation (6).

For a scattering problem, the scattered pressure satisfies the integral equation (8). It can be shown that the total pressure satisfies the integral equation

$$\tfrac{1}{2}P(\zeta) - \int_S P(\xi)\frac{\partial G}{\partial n}(\zeta,\xi)\, dS(\xi) = -\int_S G(\zeta,\xi)Q(\xi)\, dS(\xi) + P_{\text{inc}}(\zeta) \qquad \zeta \text{ on } S. \tag{9}$$

In the remainder of this discussion, we will only consider the integral equation (9) because equation (8) can be obtained by setting $P_{\text{inc}} = 0$.

A problem with this reduction to an integral equation on the surface is that there is an infinite discrete set of frequencies $f_1, f_2, \ldots$ for which the integral equation does not have a unique solution. It turns out that these frequencies are the resonance frequencies of the interior problem for the boundary condition $P = 0$ on S. As one might expect, numerical methods based on these integral equations have a problem when the frequency is close to one of these interior resonance frequencies. Various methods for overcoming this difficulty can be found in the literature [Benthien and Schenck, 1997]. The resonance frequencies $f_1, f_2, \ldots$ are not resonances of the problem of interest. There are no resonances of the exterior problem. The exterior problem has a unique solution at all frequencies. The interior resonance problem is a mathematical difficulty, not a physical one. What happens at these frequencies is that the integral equation is no longer equivalent to the original exterior acoustic boundary value problem.

CHIEF uses a piecewise constant approximation to the integral equation (9). In this approximation, the boundary surface S is subdivided into a finite number of subareas $S_1, S_2, \ldots, S_N$. On each subarea S_n the pressure is approximated by a single value P_n and the normal derivative of pressure is approximated by a single value Q_n. With this approximation, the integral equation (9) becomes

$$\tfrac{1}{2}P(\zeta) - \sum_{n=1}^{N} P_n \int_{S_n} \frac{\partial G}{\partial n}(\zeta,\xi)\, dS(\xi) = -\sum_{n=1}^{N} Q_n \int_{S_n} G(\zeta,\xi)\, dS(\xi) + P_{\text{inc}}(\zeta). \tag{10}$$

Evaluating equation (10) at a set of reference points, $\zeta_1, \zeta_2, \ldots, \zeta_N$ (one on each subarea), we obtain the system of equations

$$\sum_{n=1}^{N} A_{mn}P_n = \sum_{n=1}^{N} B_{mn}Q_n + P_{\text{inc}}(\zeta_m) \qquad m = 1, 2, \ldots, N, \tag{11}$$

where

$$A_{mn} = \tfrac{1}{2}\delta_{mn} - \int_{S_n} \frac{\partial G}{\partial n}(\zeta_m, \xi)\, dS(\xi), \tag{12}$$

$$B_{mn} = -\int_{S_n} G(\zeta_m, \xi)\, dS(\xi), \tag{13}$$

and δ_{mn} is the Kronecker delta defined by

$$\delta_{mn} = \begin{cases} 0 & m \neq n \\ 1 & m = n. \end{cases} \tag{14}$$

ζ_m is taken at the center of the m-th subarea. The system of equations (11) can be written in the matrix form,

$$AP = BQ + P_{\text{inc}}, \tag{15}$$

where P is an N-vector whose n-th component is P_n, Q is an N-vector whose n-th component is Q_n, and P_{inc} is an N-vector whose n-th component is $P_{\text{inc}}(\zeta_n)$. If the normal derivatives $Q_1, Q_2, \ldots, Q_N$ are prescribed, then the system of equations (11) can be solved for the surface pressures $P_1, P_2, \ldots, P_N$.

The expression in equation (6) for the pressure at an arbitrary field point can be approximated using the same discrete approximations, yielding

$$P(x) = \sum_{n=1}^{N} [A_n(x) P_n + B_n(x) Q_n], \tag{16}$$

where

$$A_n(x) = \int_{S_n} \frac{\partial G}{\partial n}(x, \xi)\, dS(\xi) \quad \text{and} \quad B_n(x) = -\int_{S_n} G(x, \xi)\, dS(\xi). \tag{17}$$

In the scattering problem $P(x)$ must be replaced by the scattered pressure at x. The surface values P_n and Q_n can be either the total or the scattered values. Equation (16) can be used to calculate the pressure at an arbitrary field point x once the surface pressures and surface normal derivatives have been determined. Evaluating equation (16) at a set of field points $x_1, \ldots, x_M$, we obtain

$$P(x_m) = \sum_{n=1}^{N} [A_n(x_m) P_n + B_n(x_m) Q_n] \qquad m = 1, \ldots, M. \tag{18}$$

Equation (18) can be written in the matrix form,

$$P^{\text{fld}} = A^{\text{fld}} P + B^{\text{fld}} Q, \tag{19}$$

where P^{fld} is the vector of field pressures, P is the vector of surface pressures, Q is the vector of pressure normal derivatives, and the field matrices A^{fld}, B^{fld} are given by

$$A^{\text{fld}}_{mn} = A_n(x_m) \tag{20}$$

$$B^{\text{fld}}_{mn} = B_n(x_m). \tag{21}$$

CHIEF also contains far-field approximations for these field equations when the field points are far from the radiating surface. When $P_{\text{inc}} = 0$ equation (15) can be solved for P to obtain the impedance relation,

$$F = DP = ZV, \tag{22}$$

where D is the diagonal matrix of surface subdivision areas and Z is the impedance matrix given by

$$Z = -i\omega\rho D A^{-1} B. \tag{23}$$

Substituting equation (22) into equation (19), we obtain

$$P^{\text{fld}} = [A^{\text{fld}} D^{-1} Z - i\omega\rho B^{\text{fld}}]V. \tag{24}$$

The matrix on the right-hand side of equation (24) is sometimes referred to as the field matrix. In a scattering problem, P^{fld} must be replaced by the scattered field pressure and V must be replaced by the scattered velocity since the total pressure and velocity are not related by the impedance relation in equation (22).

In the vicinity of one of the frequencies for which the underlying integral equation does not have a unique solution, the system of equations (11) becomes numerically ill-conditioned. One method of overcoming this difficulty is to add some additional equations obtained by evaluating a discretized version of the last relation in equation (6) at a selected number of interior points. The overdetermined system of equations is then solved in a least squares sense. A more complete discussion of this approach to solving acoustic problems is contained in [Benthien, Barach, and Gillette, 1988].

While it is important to maintain the capability of handling quite general surfaces, it is also important to realize that significant reductions in computation time can be achieved by taking advantage of any symmetries that might be present in the problem. In the CHIEF approach, symmetry manifests itself in the structure of the A and B matrices. The previously referenced CHIEF manual shows that each type of symmetry results in the A and B matrices commuting with the appropriate symmetry operator Σ, for example,

$$A\Sigma = \Sigma A \quad \text{and} \quad B\Sigma = \Sigma B. \tag{25}$$

It follows from these commutation relations that the eigenspaces of Σ are invariant under both A and B. Thus, A and B become block diagonal when expressed in terms of a basis consisting of eigenvectors of Σ. The sizes of the diagonal blocks are the dimensions of the eigenspaces of Σ.

As an example of how symmetry is treated in CHIEF, consider the special case of one plane of symmetry. The symmetry operator for one plane of symmetry is

$$\Sigma = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix}, \tag{26}$$

assuming that symmetric elements are numbered in the same order. The fact that A commutes with Σ implies that A has the block form,

$$A = \begin{pmatrix} A_1 & A_2 \\ A_2 & A_1 \end{pmatrix}. \tag{27}$$

The matrix X of eigenvectors of Σ is given by

$$X = \begin{pmatrix} I & I \\ I & -I \end{pmatrix}. \tag{28}$$

The inverse X^{-1} of X turns out to be $\frac{1}{2}X$. It is easily verified that $X^{-1}AX$ and $X^{-1}BX$ have the block diagonal form,

$$X^{-1}AX = \begin{pmatrix} A_1 + A_2 & 0 \\ 0 & A_1 - A_2 \end{pmatrix}, \quad X^{-1}BX = \begin{pmatrix} B_1 + B_2 & 0 \\ 0 & B_1 - B_2 \end{pmatrix}. \tag{29}$$

Thus, the equation $AP = BQ$ can be written as

$$(X^{-1}AX)\hat{P} = (X^{-1}BX)\hat{Q}, \tag{30}$$

where

$$\hat{P} = X^{-1}P \qquad \text{and} \qquad \hat{Q} = X^{-1}Q. \tag{31}$$

Equation (30) can be written in the partitioned form,

$$\begin{pmatrix} A_1 + A_2 & 0 \\ 0 & A_1 - A_2 \end{pmatrix} \begin{pmatrix} \hat{P}_1 \\ \hat{P}_2 \end{pmatrix} = \begin{pmatrix} B_1 + B_2 & 0 \\ 0 & B_1 - B_2 \end{pmatrix} \begin{pmatrix} \hat{Q}_1 \\ \hat{Q}_2 \end{pmatrix}. \tag{32}$$

Thus, the original system of equations reduces to the pair of smaller systems of equations:

$$(A_1 + A_2)\hat{P}_1 = (B_1 + B_2)\hat{Q}_1 \tag{33}$$

$$(A_1 - A_2)\hat{P}_2 = (B_1 - B_2)\hat{Q}_2. \tag{34}$$

Once these equations are solved, the solution of the original system of equations is obtained by

$$P_1 = \hat{P}_1 + \hat{P}_2 \tag{35}$$

$$P_2 = \hat{P}_1 - \hat{P}_2. \tag{36}$$

Since solution time goes roughly as the cube of the number of equations, it is faster by a factor of four to solve the two smaller systems rather than the one larger system. The symmetry can also be used to reduce the time needed to generate the matrices. The CHIEF program can take advantage of one, two, or three planes of symmetry as well as any finite order of rotational symmetry in the surface. CHIEF can also take advantage of symmetry in the Q distribution when it exists. Table A-1 shows the time-reduction factors for various types of symmetry.

Table A-1. Time reduction factors in CHIEF for various types of symmetry.

Symmetry Type	Solution	Using Q symmetry	Matrix generation
One plane	4	8	2
Two planes	16	64	4
Three planes	64	512	8
N-fold rotational	N^2	N^3	N

In the point source check and in the automatic generation of interior points, CHIEF needs to check points to see if they are interior or exterior to the boundary surface. The check is based on a result for potential problems that is similar to the Helmholtz integral relations. If ϕ satisfies Laplace's equation, i.e.,

$$\triangle\phi = 0$$

interior to the boundary surface S, then

$$\int_S \left[G_0(x,\xi)\frac{\partial\phi}{\partial n}(\xi) - \phi(\xi)\frac{\partial G_0}{\partial n}(x,\xi)\right] dS(\xi) = \begin{cases} 0 & x \text{ outside } S \\ \frac{1}{2}\phi(x) & x \text{ on } S \\ \phi(x) & x \text{ inside } S, \end{cases} \tag{37}$$

where $G_0(x,\xi)$ is the free-space Green's function for the Laplace equation defined by

$$G_0(x,\xi) = \frac{1}{4\pi|x-\xi|}, \tag{38}$$

and n is the outward unit normal to S. Since the constant function $\phi(x) \equiv 1$ satisfies Laplace's equation interior to the boundary S and its normal derivative is zero on S, the integral relation (37) yields

$$-\int_S \frac{\partial G_0}{\partial n}(x,\xi)\, dS(\xi) = \begin{cases} 0 & x \text{ outside } S \\ \frac{1}{2} & x \text{ on } S \\ 1 & x \text{ inside } S \end{cases} \tag{39}$$

or

$$-\int_S \frac{\partial}{\partial n}\frac{1}{|x-\xi|}\, dS(\xi) = \begin{cases} 0 & x \text{ outside } S \\ 2\pi & x \text{ on } S \\ 4\pi & x \text{ inside } S \end{cases} . \tag{40}$$

The numerical evaluation of equation (40) is used to determine if a point x is inside or outside of S.

The reader should see the appendix to the original CHIEF manual for more details on the theory behind the CHIEF program.