NASA Technical Memorandum 110215

Fully-Coupled Fluid/Structure Vibration Analysis Using MSC/NASTRAN

Christian M. Fernholz
The George Washington University,
Joint Institute for Advancement of Flight Sciences, Hampton, Virginia

Jay H. Robinson
Langley Research Center, Hampton, Virginia

January 1996

National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia 23681-0001

FULLY-COUPLED FLUID/STRUCTURE VIBRATION ANALYSIS USING MSC/NASTRAN

Christian M. Fernholz
The George Washington University
Joint Institute for Advancement of Flight Sciences
Hampton, VA 23681-0001

Jay H. Robinson
NASA Langley Research Center
Structural Acoustics Branch
Hampton, VA 23681-0001

MSC/NASTRAN's performance in the solution of fully-coupled fluid/structure problems is evaluated. NASTRAN is used to perform normal modes (SOL 103) and forced-response analyses (SOL 108, 111) on cylindrical and cubic fluid/structure models. Each model is discretized using finite element methods. Bulk data file cards unique to the specification of a fluid/structure model are discussed and analytic partially-coupled solutions are derived for each type of problem. These solutions are used to evaluate NASTRAN's solutions for accuracy. Appendices to this work include NASTRAN data presented in fringe plot form, FORTRAN source code listings written in support of this work, and NASTRAN data file usage requirements for each analysis.

Nomenclature

A	Length of a side for cube, plate
F_o	Force amplitude
E	Young's Modulus
P	Acoustic pressure
a	Radius for cylinder
c_o	Acoustic speed of sound in an ideal gas
h	Thickness of plate, shell
i	$\sqrt{-1}$
l	Length of cylinder
t	Time
u, v	In-plane displacements
w	Out-of-plane displacements
x, y, z	Rectangular coordinate system
r, θ, z	Cylindrical coordinate system
γ	Damping coefficient

δ	Dirac delta
η	Structural damping coefficient
ν	Poission's ratio
ρ_f	Density of fluid
ρ_s	Density of structure
ω	Frequency

Section 1: Introduction

With the advent of more powerful computers, it is becoming possible to solve vibrations problems of ever increasing complexity. A greater dependence upon computers necessitates that researchers know the limits and capabilities of the computer systems and the software packages they are using. If these limits are unknown, the researcher may be unable to discern nonsensical data from accurate data provided by the computer.

It is the aim of this paper to evaluate the capabilities of a commercially available finite element analysis package called MSC/NASTRAN (versions 67 and 68) in the analysis of fully coupled fluid/structure problems. Normal modes analysis and forced response analysis is used on two simple, well-understood geometric models. The models are different from each other both in shape and complexity in an effort to determine the accuracy of NASTRAN. The accuracy of a NASTRAN-generated solution is determined through comparison with classical analytic solutions.

This paper is divided into two sections, one each for a cubic and a cylindrical geometry. Within each of these sections are analytic and numeric solutions for three types of problems. The first problem is a normal modes analysis for a model containing fluid elements only. The second is a normal modes analysis of a model containing both fluid and structure elements. Normal modes of vibration are determined for the structural and fluid portions of the system. The third problem is a forced response analysis of the same model as that used in the second problem. For the normal modes problems, both quadratic and linear element meshes are used. The forced response analysis makes use of a linear finite element mesh only.

MSC/PATRAN was used as a graphical pre- and post-processor. FEM models and bulk data files were constructed within PATRAN. However, at the time of this work, PATRAN did not have the capability to work with fluid elements. Thus, modifications were made to the bulk data file outside of PATRAN before it was submitted to NASTRAN for analysis. These modifications are discussed in this paper.

Appendices to this work provide additional information about the numeric solutions to these problems. Appendix A tabulates NASTRAN's numeric results for each problem and compares it to the appropriate analytic result. MSC/PATRAN was used to produce the fringe plots shown. It should be noted that PATRAN uses a linear interpolation between nodes in a finite element mesh, no mater what type of element was used. Thus, linear and quadratic element meshes having an equal number of nodes will appear identical in a PATRAN-generated fringe plot. Appendix B lists the FORTRAN codes written in support of this work. Several codes make calls to Bessel function algorithms. These algorithms can be found in the text *Numerical Recipes: The Art of Scientific Computing (FORTRAN Version)*[1]. Appendix C provides computer file usage data for the numerical solutions.

Section 2: The Cubic Problem

The fluid/structure interaction problem was first considered for a cubic domain. This domain was chosen because it represents a geometrical configuration wherein the wave equation is readily solvable in Cartesian coordinates. To further facilitate solution of the equation using analytic methods, boundary conditions of P=0 were enforced on all surfaces of the fluid not in contact with the structure. For structural portions of the model, thin elastic plates were used with boundary conditions of simple-support on each edge. For the forced response analysis, forces of equal magnitude and phase but opposite direction were applied to the model. They were applied such that translational vibration modes did not need to be considered for this analysis. The combination of these conditions makes an analytic solution of the problem straightforward. A skematic representation of the forced response fluid/structure model for the cubic geometry is shown in Figure 1.

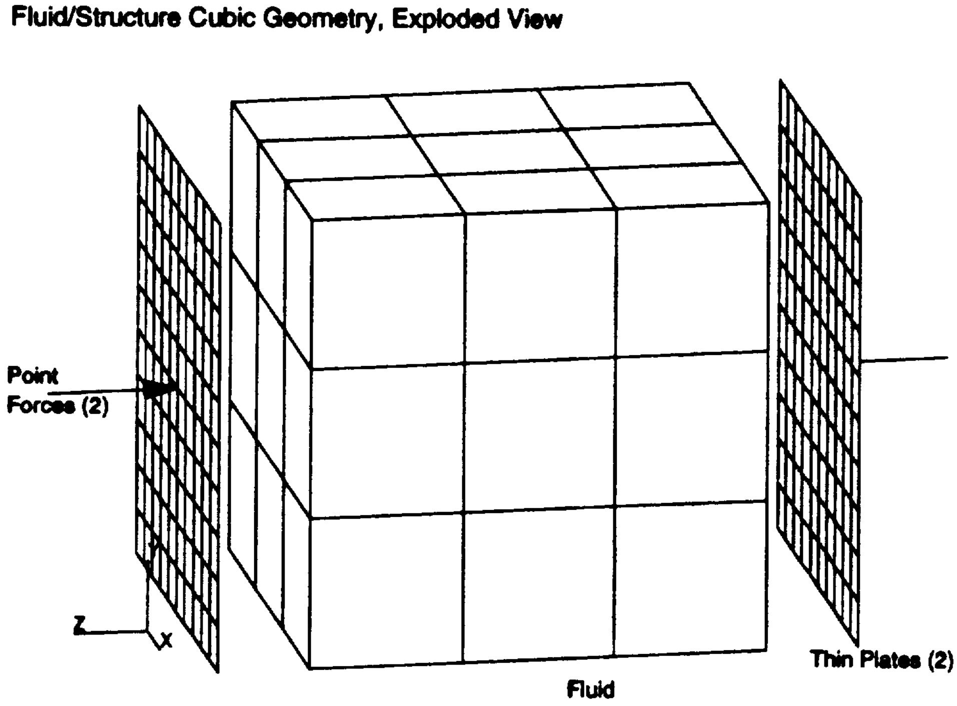
Fluid/Structure Cubic Geometry, Exploded View

Figure 1: Exploded view of the fluid/structure model for the cubic geometry.

The Free Fluid Problem

For an ideal, stationary fluid, the acoustic pressure field is described by the wave equation

$$\nabla^2 P - \frac{1}{c_o^2}\frac{\partial^2 P}{\partial t^2} = 0 \tag{1}$$

where, in Cartesian coordinates,

$$\nabla^2 P = \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} + \frac{\partial^2 P}{\partial z^2} \tag{2}$$

Using the separation of variables technique[2] in conjunction with the homogeneous boundary conditions

$$\begin{gathered} P(0,y,z,t) = P(A,y,z,t) = 0 \\ P(x,0,z,t) = P(x,A,z,t) = 0 \\ P(x,y,0,t) = P(x,y,A,t) = 0 \end{gathered} \tag{3}$$

and assuming a harmonic time dependance, the solution to Equation 1 is

$$P(x,y,z,t) = e^{i\omega t} \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} \sum_{k=1}^{\infty} D_{nmk} sin \frac{n\pi x}{A} sin \frac{m\pi y}{A} sin \frac{k\pi z}{A} \tag{4}$$

where D_{nmk} is a constant to be determined from initial conditions. The natural frequencies ω_{nmk} are given by

$$\omega_{nmk} = \frac{c_o \pi}{A} \sqrt{n^2 + m^2 + k^2} \tag{5}$$

A cubic fluid volume was discretized with three different meshes for analysis by MSC/NASTRAN. One model was constructed using 1000 linear HEX8 elements and 1331 nodes. The second model used 216 quadratic elements and 1225 nodes. The third model was discretized with 1000 quadratic HEX20 elements and 4961 nodes. The fluid in each of the models was given the material properties for density and speed of sound shown in Table 1. Boundary conditions in agreement with Equation 3 were also applied to the each of the models.

Symbol	Property Name	Material Property Value
E	Young's Modulus	10.3×10^6 psi
h	Thickness of structure	0.0625 in
A	Length of side	5 in
c_o	Acoustic speed of sound	13.620×10^3 in/sec
ν	Poission's Ratio	0.334
ρ_s	Density of structure	2.5383×10^{-4} slugs/in^3
ρ_f	Density of fluid	1.170×10^{-7} slugs/in^3

Table 1 Properties for the cubic model.

To specify a fluid element within NASTRAN, several bulk data cards must be specified or changed. The brief outline of these cards which follows is as shown in the *MSC/NASTRAN Quick Reference Guide, Version 68.* The reader is referred to this source for more detail regarding these cards. The first card to be modified is the GRID card. The general format of this card is as shown in Figure 2[3]. A "-1" in the CD field of the GRID card is used to signify a grid belonging to a fluid element. It should be noted that such a grid point can only be defined for volume elements.

GRID	ID	CP	X1	X2	X3	CD	PS	SEID	

ID	Grid point identification number.
CP	Identification number of coordinate system in which the location of the grid point is defined.
X1,X2,X3	Location of the grid point in coordinate system CP.
CD	Identification number of coordinate system in which the displacements, degrees of freedom, constraints, and solution vectors are defined at the grid point.
PS	Permanent single-point constraints associated with the grid point.
SEID	Superelement identification number.

Figure 2: NASTRAN GRID bulk data card format.

The second card that must be specified is the material properties card. The format of the material properties card is shown in Figure 3[3]. Note that the MAT10 card specifies material properties for fluid elements only.

MAT10	MID	BULK	RHO	C					

MID	Material identification number.
BULK	Bulk modulus.
RHO	Mass density.
C	Speed of sound

Figure 3: NASTRAN MAT10 bulk data card format.

The third bulk data card that must be specified is the PSOLID card, shown in Figure 4[3]. For fluid elements in the model, the FCTN field of the PSOLID card must be specified as "PFLUID".

PSOLID	PID	MID	CORDM	IN	STRESS	ISOP	FCTN		

PID — Property identification number.

MID — Identification number of a MAT1, MAT4, MAT5, MAT9, or MAT10 entry.

CORDM — Identification number of the material coordinate system.

IN — Integration network.

STRESS — Location selection for stress output.

ISOP — Integration scheme.

FCTN — Fluid element flag. (Character: "PFLUID" indicates a fluid element, "SMECH" indicates a structural element; defalut = "SMECH"

Figure 4: NASTRAN PSOLID bulk data card format.

MSC/NASTRAN performed a normal modes analysis (SOL 103) on the linear and quadratic models. The NASTRAN-calculated frequency for the linear HEX8 model mode 1,1,1 was 2368.77 Hz. For the 1225–node quadratic HEX20 model, the eigenfrequency for this mode was 2359.18 Hz. The 4961–node quadratic HEX20 model yielded a frequency of 2359.32 Hz. For comparison with NASTRAN's results, the natural frequencies for these models can be determined analytically using Equation 5. From this relation, the first natural frequency for each model is 2359.05 Hz, mode 1,1,1. A fringe plot of this mode shape for each of the models is shown in Figures A1, A2, and A3. It should be noted that in these (and subsequent) modal fringe plots, the displacements have been normalized to one.

The NASTRAN solution was compared for accuracy against the analytic solution, Equation 4. A fringe plot displaying the difference between analytic and numeric solutions at each node in the model was created for each case. The error plot for the linear HEX8 cube, mode 1,1,1 is shown in Figure A4. The error plot for the 1225–node quadratic HEX20 cube, mode

1,1,1 is shown in Figure A5. Figure A6 shows the error plot for the 4961-node quadratic model.

This analysis was repeated for mode 1,3,1 of the linear model and mode 1,1,3 of the quadratic models. These mode shapes are shown in Figures A7, A8, and A9. The associated error plots for each of these mode shapes are shown in Figures A10, A11, andA12. A summary of the normal modes analysis for the fluid cube is shown in Table 2. Here, the maximum model error is the largest difference anywhere in the model between the normalized analytic and numeric solutions.

Model Type	Elements	Nodes	Mode Shape	Eigenvalue Error	Maximum Error in Model
Fluid only	1000 Linear HEX8	1331	1,1,1	0.4117%	6.83×10^{-6}
	215 Quadratic HEX20	1225		0.0055%	0.0015
	1000 Quadratic HEX20	4961		0.0007%	0.0002
	1000 Linear HEX8	1331	1,3,1	3.134%	0.510
	215 Quadratic HEX20	1225		0.3137%	0.4593
	1000 Quadratic HEX20	4961		0.0432%	0.386

Table 2 Results for cubic geometry, normal modes analysis.

The Free Plate Problem for the Fluid/Structure Model

Next, the model was modified by bounding two opposing faces of the fluid cube with thin, elastic plates. A partially coupled solution to the free vibration problem for this system is considered first. For the unforced case, the thin elastic plate obeys the following equation of motion[4]:

$$D\nabla^4 w + \rho_s h \frac{\partial^2 w}{\partial t^2} = 0 \tag{6}$$

where w is the displacement of the plate in the direction normal to its surface and D, the flexural rigidity, is given by

$$D = \frac{Eh^3}{12(1-\nu^2)} \tag{7}$$

Assuming a harmonic time dependence and simply-supported boundary conditions at each edge, the out-of plane displacement w can be written

$$w(x,y,t) = e^{i\omega t}\sum_{n=1}^{\infty}\sum_{m=1}^{\infty} C_{mn} sin\Big(\frac{m\pi x}{A}\Big) sin\Big(\frac{n\pi y}{A}\Big) \tag{8}$$

with its natural frequencies given by

$$\omega_{mn} = \left\{\Big(\frac{m\pi}{A}\Big)^2 + \Big(\frac{n\pi}{A}\Big)^2\right\}\sqrt{\frac{D}{\rho_s h}} \tag{9}$$

The Free Fluid Problem for the Fluid/Structure Model

For the ideal fluid between the plates, the wave equation (Equation 1) still applies, but with the following boundary conditions:

$$\begin{gathered} P(0,y,z,t) = P(A,y,z,t) = 0 \\ P(x,0,z,t) = P(x,A,z,t) = 0 \\ \frac{\partial P(x,y,\frac{-A}{2},t)}{\partial z} = \rho_f \frac{\partial^2 w(x,y,t)}{\partial t^2} \\ \frac{\partial P(x,y,\frac{A}{2},t)}{\partial z} = -\rho_f \frac{\partial^2 w(x,y,t)}{\partial t^2} \end{gathered} \tag{10}$$

That is, the fluid is constrained to match the behavior of the plates at those points where they are in contact. Applying these boundary conditions yields the following expression for P:

$$P(x,y,z,t) = -e^{i\omega t}\sum_{n=1}^{\infty}\sum_{m=1}^{\infty} D_{mn} sin\Big(\frac{m\pi x}{A}\Big) sin\Big(\frac{n\pi y}{A}\Big) cos(\alpha_{mn} z) \tag{11}$$

where

$$\alpha_{mn}^2 = \frac{\pi^2}{A^2}(m^2+n^2) - \frac{\omega^2}{c_o^2} \tag{12}$$

and

$$D_{mn} = \frac{\rho_f \omega^2 C_{mn}}{\alpha_{mn} sin(\frac{A}{2}\alpha_{mn})(\omega_{mn}^2 - \omega^2)} \tag{13}$$

Three cubic fluid/structure models were constructed for a NASTRAN normal modes analysis. The fluid and structure elements within each of the models were given the material properties shown in Table 1. The number and type of elements and nodes were as shown in

Table 3. The boundary conditions of Equation 10 were applied to the fluid surfaces in the model and the edges of the plates were simply supported.

Model	Material	Element Type	Number of Nodes	Total Nodes in Model
1	Structure	200 Linear QUAD4	242	1573
	Fluid	1000 Linear HEX8	1331	
2	Structure	72 Quadratic QUAD8	266	1491
	Fluid	216 Quadratic HEX20	1225	
3	Structure	200 Quadratic QUAD8	682	5643
	Fluid	1000 Quadratic HEX20	4961	

Table 3 NASTRAN models for the cubic geometry fluid/structure problem

For models containing both fluid and structure elements, it is necessary in the NASTRAN bulk data file to explicitly specify which nodes lie on the interface between the fluid and structure portions of the model. This is accomplished by using the NASTRAN ACMODL card. The general format of ACMODL card is shown in Figure 5[3]. For this work, each structure grid in the SSET card had a corresponding fluid grid in the FSET card.

ACMODL	INTER	INFOR	FSET	SSET	FSTOL				

INTER Type of interface between the fluid and the structure.

INFOR Indicates wether FSET and SSET are to be used to define the fluid-structure interface

FSET Identification number of a SET1 entry that contains a list of fluid grid points on the interface.

SSET Identification number of a SET1 entry that contains a list of structural grid points on the interface.

FSTOL Tolerance, in units of length, used in determining the fluid-structure interface.

Figure 5: MSC/NASTRAN ACMODL bulk data card format.

Entry "by hand" of the nodes required for the ACMODL card would be an extraordinarily tedious task, particularly for more complicated or larger models. Several FORTRAN codes were created to write this data for the ACMODL card from data already in the bulk data file. These codes are listed in Appendix B of this work.

From Equation 9, the first natural frequency of the plate is 484.54 Hz, mode 1,1. For the linear QUAD4 model, the NASTRAN-calculated eigenfrequency for mode 1,1 of the plate was 486.86 Hz. For the 266–node quadratic QUAD8 model, it was 481.02 Hz. The 682–node QUAD8 model yielded a result of 482.37 Hz. This mode shape is shown for each of the QUAD4 and QUAD8 models in Figures A13, A14, and A15. From these figures, it is apparent that a strong cross-coupling between the front ($z=\frac{A}{2}$) and rear ($z=-\frac{A}{2}$) plates is occurring. Unlike the partially-coupled analytic solution derived above, NASTRAN assumes a fully coupled solution. Thus, the first mode shape of the front plate will be coupled through the fluid to the rear plate in the NASTRAN solution. In the uncoupled analytic solution, the first mode shape of the system is a 1,1 mode shape for one plate and a stationary (0,0) shape for the other.

The uncoupled analytic solution was used to analyze the mode shapes present on the front plate of the model. The difference between the analytic and numeric solutions at each node for all three models is shown in Figures A16, A17, and A18.

This analysis was repeated for plate mode 2,2. This mode shape is shown in Figures A19, A20, and A21. The associated error plots are shown in Figures A22, A23, and A24. A summary of the normal modes of vibration for the structure portions of the cubic models is listed in Table 4.

Model Type	**Elements**	**Nodes**	**Mode Shape**	**Eigenvalue Error**	**Maximum Error in Model**

Table 4 Normal modes analysis for structure portion only of the fluid/structure model, cubic geometry. (Continued) . . .

Fluid/struct	200 Linear QUAD4	243	1,1	0.4778%	-1.29×10^{-3}
	72 Quadratic QUAD8	266		0.7265%	-1.82×10^{-3}
	200 Quadratic QUAD8	682		0.4481%	-1.17×10^{-3}
	200 Linear QUAD4	243	2,2	2.898%	95.5×10^{-3}
	72 Quadratic QUAD8	266		0.7275%	133.4×10^{-3}
	200 Quadratic QUAD8	682		0.5628%	48.9×10^{-3}

Table 4 Normal modes analysis for structure portion
only of the fluid/structure model, cubic geometry.

For the fluid portions of the system, fringe plots of NASTRAN calculated mode 1,1,0 are shown in Figures A25 (linear HEX8 elements), A26 (1225-node quadratic HEX20 model), and A27 (4961-node quadratic HEX20 model). For the linear model, NASTRAN calculated a frequency for this mode of 1934.09 Hz. For the 1225-node quadratic model, it was 1926.26 Hz. The 4961-node HEX20 model yielded 1926.17 Hz. A fringe plot showing the difference between the analytic and numeric solutions of this mode for each model is shown in Figures A28, A29, and A30

This analysis was repeated for mode 1,1,1 of the fluid for both models. This mode shape is shown in Figures A31, A32, and A33. The associated error plots are shown in Figures A34, A35, and A36. A summary of the normal modes of vibration for the fluid is listed in Table 5.

Model Type	**Elements**	**Nodes**	**Mode Shape**	**Eigenvalue Error**	**Maximum Error in Model**
Fluid/struct	1000 Linear HEX8	1331	1,1,0	0.4637%	5.6×10^{-6}
	216 Quadratic HEX20	1225		0.0052%	728.0×10^{-6}
	1000 Quadratic HEX20	4961		0.0007%	96.1×10^{-6}
	1000 Linear HEX8	1331	1,1,1	0.4117%	6.6×10^{-6}
	216 Quadratic HEX20	1225		0.0054%	1.532×10^{-3}
	1000 Quadratic HEX20	4961		0.0007%	198.6×10^{-6}

Table 5 Normal modes analysis for fluid portion
only of fluid/structure model, cubic geometry.

The Forced Fluid/Structure Problem

The third and final analysis performed for the cubic geometry model was that of forced vibration. A finite element model identical to the linear model used for the free vibration analysis was used, the only difference being the application of a harmonic point force at the center of each plate in the system. The directional sense of these forces was such that they pointed in toward the fluid volume between the plates. Again, the analytic solution of the uncoupled problem is considered first. For a forced vibration analysis of a thin, elastic plate including damping effects, the governing equation is now written

$$D\nabla^4 w + \gamma\frac{\partial w}{\partial t} + \rho_s h\frac{\partial^2 w}{\partial t^2} = F(x,y,t) \tag{14}$$

where γ is the viscous damping coefficient and *F(x,y,t)* is a harmonic point force applied to the plate. For the case where this force is applied to the center of the plate, it can be written as:

$$F(x,y,t) = F_o e^{i\omega t}\delta\left(x-\frac{A}{2}\right)\delta\left(y-\frac{A}{2}\right) \tag{15}$$

Applying simply-supported boundary conditions to the plate, the solution to Equation 14 is given by

$$w(x,y,t) = e^{i\omega t}\sum_{m=1}^{\infty}\sum_{n=1}^{\infty} C_{mn} sin\left(\frac{m\pi x}{A}\right) sin\left(\frac{n\pi y}{A}\right) \tag{16}$$

Expanding *F(x,y,t)* in terms of the eigenfunctions of the plate, we have

$$F(x,y,t) = e^{i\omega t}\sum_{m=0}^{\infty}\sum_{n=0}^{\infty} F_{mn} sin\left(\frac{m\pi}{2}\right) sin\left(\frac{n\pi}{2}\right) \tag{17}$$

where

$$F_{mn} = \frac{4F_o}{A^2} \tag{18}$$

and thus

$$C_{mn} = \frac{4F_o}{A^2\rho_s h}\frac{sin(\frac{m\pi}{2})sin(\frac{n\pi}{2})}{(\omega_{mn}^2 - \omega^2 + 2i\eta\omega\omega_{mn})} \tag{19}$$

Note that in Equation 16, the viscous damping coefficient γ has been replaced by a frequency-dependent damping coefficient η. The relation between these two coefficients is:

$$\frac{\gamma}{\rho_s h} = 2\eta\omega_{mn} \tag{20}$$

For the fluid in the region between the plates, Equation 1 and boundary conditions 10 still apply. Using the above results for the plates, the equation for acoustic pressure at a point in the fluid is now written

$$P(x,y,z,t) = -e^{i\omega t}\sum_{n=1}^{\infty}\sum_{m=1}^{\infty} D_{mn} sin\left(\frac{m\pi x}{A}\right) sin\left(\frac{n\pi y}{A}\right) cos(\alpha_{mn} z) \tag{21}$$

where again

$$\alpha_{mn}^2 = \frac{\pi^2}{A^2}(m^2+n^2) - \frac{\omega^2}{c_o^2} \tag{22}$$

but

$$D_{mn} = \frac{4F_o}{A^2\rho_s h}\frac{\rho_f \omega^2 sin(\frac{n\pi}{2}) sin(\frac{m\pi}{2})}{\alpha_{mn} sin(\frac{\alpha A}{2})(\omega_{mn}^2 - \omega^2 + 2i\eta\omega\omega_{mn})} \tag{23}$$

For the numerical analysis by NASTRAN of this forced-response problem point forces of $-5\sin(\omega t)\hat{n}$ lbs. were applied to the center of each plate in the linear model. Boundary conditions for the fluid and structure portions of the model were otherwise identical to those used for the free vibration problem.

A direct frequency response analysis (SOL 108) was performed by NASTRAN over a frequency range from zero to 5000 Hz. Structural damping in the model was specified by using the PARAM G card in the NASTRAN bulk data file. Figure A37 shows the displacement of a node located at the center of the front plate for the linear QUAD4 model. Figure A38 shows the acoustic pressure disturbance at a node halfway between the center of the front plate and the center of the fluid region. Also shown are the partialy-coupled analytic solutions for the displacement and pressure respectively at these two locations.

A modal frequency response analysis (SOL 111) was next performed by NASTRAN on the same cubic-geometry model. The first twenty NASTRAN-calculated fluid and structural modes were used by NASTRAN to calculate the overall response of the model. It should be noted that the twentieth natural frequency of the structure, as calculated by NASTRAN, occurs at approximately 3000 HZ, while the twentieth natural frequency of the fluid occurs above 5000 Hz. The results of this analysis for a frequency range of zero to 5000 Hz are shown in Figures A39 for a point at the center of the front plate, and A40 for a point midway between the front plate and the center of the cube, along with analytic solutions at each of these points.

Section 3: The Cylindrical Problem

A cylindrical domain was next defined for analysis. As was the case for the cubic domain, a cylindrical geometry was chosen because an analytic solution of the wave equation in cylindrical coordinates becomes straightforward and uncomplicated. To facilitate solution of the wave equation using the separation of variables technique, boundary conditions of $P=0$ were again enforced on all fluid surfaces not in contact with any structures in the model. For models in which a structure was present, a thin cylindrical shell was used. Boundary conditions for the shell were chosen which simplified the analytic solution. For the forced response analysis, forces applied to the model were equal in magnitude and phase, but opposite in direction. They were applied to the cylindrical shell such that translational vibration modes would not have to be considered in this analysis. A skematic representation of the cylindrical fluid/structure geometry for the forced response analysis is shown in Figure 6.

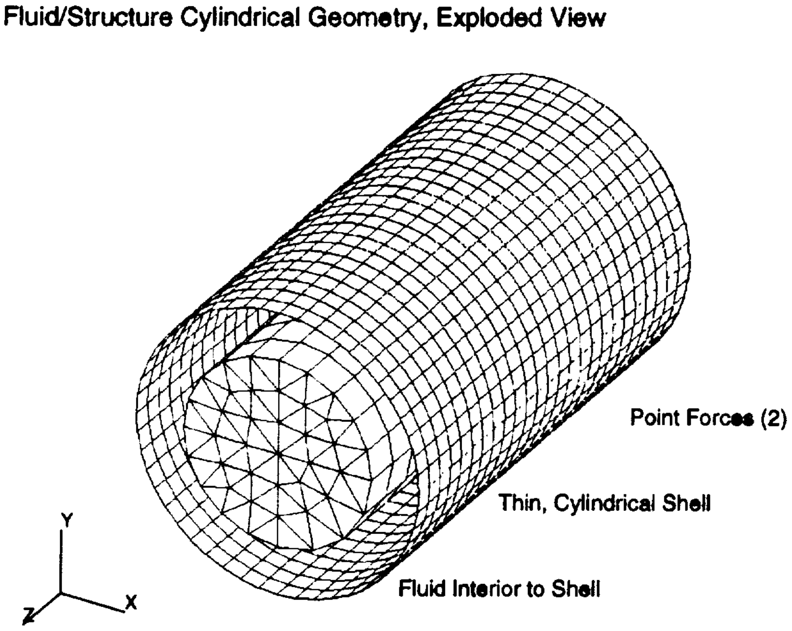

Figure 6: Exploded view of fluid/structure model for the cylindrical geometry.

The Free Fluid Problem

For an ideal, stationary fluid, the wave equation in cylindrical coordinates now becomes

$$\nabla^2 P - \frac{1}{c_o^2}\frac{\partial^2 P}{\partial t^2} = 0 \tag{24}$$

where $\nabla^2 P$ is given by

$$\nabla^2 P = \frac{\partial^2 P}{\partial r^2} + \frac{1}{r}\frac{\partial P}{\partial r} + \frac{1}{r^2}\frac{\partial^2 P}{\partial \theta^2} + \frac{\partial^2 P}{\partial z^2} \tag{25}$$

Assuming a harmonic time dependance and homogenous boundary conditions:

$$\begin{gathered} P(r,\theta,0) = P(r,\theta,l) = 0 \\ P(a,\theta,z) = 0 \\ |P(0,\theta,z)| < \infty \text{ (boundedness)} \\ P(r,\theta,z) = P(r,\theta+2\pi n,z) \text{ (periodicity)} \end{gathered} \tag{26}$$

separation of variables yields the solution

$$P(r,\theta,z,t) = e^{i\omega t}\sum_{j=1}^{\infty}\sum_{m=1}^{\infty}\sum_{n=0}^{\infty} J_n\left(\frac{r_j r}{a}\right) sin\left(\frac{m\pi z}{l}\right)[B_{mnj}cos(n\theta) + C_{mnj}sin(n\theta)] \tag{27}$$

where J_n is the nth order Bessel function, r_j is its jth root, and a and l are the radius and length of the cylinder, respectively. The natural frequencies ω_{jmn} are given by

$$\omega_{jnm} = c_o\sqrt{\left(\frac{r_j}{a}\right)^2 + \left(\frac{m\pi}{l}\right)^2} \tag{28}$$

Three cylindrical fluid volumes were discretized for analysis by MSC/NASTRAN. One model was constructed using 2240 linear WEDGE6 elements and 1449 nodes, one used 348 quadratic WEDGE15 elements and 1200 nodes, and the third used 2240 quadratic WEDGE15 elements and 6609 nodes. The fluid in each of the models was given the material properties for density and speed of sound shown in Table 6. Geometric dimensions are also shown in this table. Boundary conditions in agreement with Equation 26 were applied to each model.

Symbol	Property Name	Material Property Value
E	Young's Modulus	10.3×10^6 psi
a	Radius of cylinder	1 in
c_o	Acoustic speed of sound	13.620×10^3 in/sec
h	Thickness of structure	0.0625 in
l	Length of the cylinder	5 in
ν	Poission's Ratio	0.334
ρ_s	Density of structure	2.5383×10^{-4} slugs/in^3
ρ_f	Density of fluid	1.170×10^{-7} slugs/in^3

Table 6 Properties for cylindrical model.

For each model, NASTRAN performed a normal modes analysis (SOL 103). The natural frequencies for this geometry can be determined analytically from Equation 28 for comparison to NASTRAN's results. Analytically, the first natural frequency for this model is 5387.91 Hz, mode 1,0,1. The NASTRAN linear WEDGE6 model calculated an eigenfrequency of 5445.67 Hz for this mode. The quadratic 348-element model yielded a frequency 5392.38 Hz. The mode 1,0,1 eigen frequency for the 2240-element WEDGE15 model was 5388.06 Hz. A normalized fringe plot of this mode shape is shown for each model in Figures A41, A42, and

A43. The NASTRAN numeric solution was compared to the analytic solution, Equation 27. A fringe plot for each model showing the difference between the analytic and numeric solutions at each node is shown in Figures A44, A45, and A46.

This analysis was repeated for mode 1,1,3 for each model. This mode shape is shown in Figures A47, A48, and A49. However, at the time of this writing, it was not possible to create an error fringe plot for this mode. In calculating the eigenvectors for a model in which a rotational symmetry is present, NASTRAN introduces an arbitrary phase angle with respect to the analytic solution into its numeric solution. That is, the numeric solution is accurate, but its mode shape is rotated by some angle about its axis of symmetry. It was not possible to adequately determine what this angle was. Thus, a node-by node comparison to the analytic solution was not possible. A summary of the cylindrical normal modes analysis is shown in Table 7. The maximum model error is again the largest difference anywhere in the model between the normalized analytic and numeric solutions.

Model Type	**Elements**	**Nodes**	**Mode Shape**	**Eigenvalue Error**	**Maximum Error in Model**
Fluid only	2240 Linear WEDGE6	1449	1,0,1	1.07%	0.017
	348 Quadratic WEDGE15	1200		82.87×10^{-3}%	0.005
	2240 Quadratic WEDGE15	6609		6.50×10^{-3}%	774.2×10^{-6}
	2240 Linear WEDGE6	1149	1,1,3	3.04%	N/A
	348 Quadratic WEDGE15	1200		0.356%	N/A
	2240 Quadratic WEDGE15	6609		0.015%	N/A

Table 7 Results for cylindrical geometry, normal modes analysis.

The Free Shell Problem

The next problem considered for the cylindrical geometry was that of a fluid-structure interaction. We begin by defining a thin, finitely-long cylindrical elastic shell filled with a stationary, ideal fluid. As we did for the cubic fluid/structure geometry, we shall solve the free vibration problem for this system using an uncoupled solution. For a thin, cylindrical shell, the Donnell-Mushtari equations of motion are:[5]

$$\begin{gathered}
\left(\frac{1}{C_L^2}\right)\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial z^2} - \left(\frac{1+\nu}{2a^2}\right)\frac{\partial^2 u}{\partial \theta^2} - \left(\frac{1+\nu}{2a^2}\right)\frac{\partial^2 v}{\partial z \partial \theta} - \frac{\nu}{a}\frac{\partial w}{\partial z} = 0 \\
\left(\frac{1}{C_L^2}\right)\frac{\partial^2 v}{\partial t^2} - \left(\frac{1+\nu}{2a}\right)\frac{\partial^2 u}{\partial z \partial \theta} - \left(\frac{1-\nu}{2}\right)\frac{\partial^2 v}{\partial z^2} - \left(\frac{1}{a^2}\right)\frac{\partial^2 v}{\partial \theta^2} - \left(\frac{1}{a^2}\right)\frac{\partial w}{\partial \theta} = 0 \\
\left(\frac{1}{C_L^2}\right)\frac{\partial^2 w}{\partial t^2} + \frac{\nu}{a}\left(\frac{\partial u}{\partial z}\right) + \left(\frac{1}{a^2}\right)\frac{\partial v}{\partial \theta} + \frac{1}{a^2}w + \frac{h^2}{12}\nabla^4 w = 0
\end{gathered} \tag{29}$$

where

$$C_L = \left[\frac{E}{\rho_s(1-\nu^2)}\right]^{\frac{1}{2}} \tag{30}$$

On the LHS of Equation 29, $u(z,\theta)$, $v(z,\theta)$, $w(z,\theta)$ represent displacements in the axial, circumferential, and radial directions respectively. Assuming a harmonic time dependence with the following boundary conditions[5]:

$$\begin{gathered}
v(0,\theta) = v(l,\theta) = 0 \\
w(0,\theta) = w(l,\theta) = 0
\end{gathered} \tag{31}$$

the complete solution to Equation 29 can be written as:

$$\begin{aligned}
u(\theta,z) &= e^{i\omega t}\sum_{n=0}^{\infty}\sum_{m=0}^{\infty} cos\left(\frac{m\pi z}{l}\right)[A_{mn}cos(n\theta) + A^\star_{mn}sin(n\theta)] \\
v(\theta,z) &= e^{i\omega t}\sum_{n=0}^{\infty}\sum_{n=0}^{\infty} sin\left(\frac{m\pi z}{l}\right)[B_{mn}sin(n\theta) + B^\star_{mn}cos(n\theta)] \\
w(\theta,z) &= e^{i\omega t}\sum_{n=0}^{\infty}\sum_{m=0}^{\infty} sin\left(\frac{m\pi z}{l}\right)[C_{mn}cos(n\theta) + C^\star_{mn}sin(n\theta)]
\end{aligned} \tag{32}$$

Because of the boundary conditions imposed for this analysis, either the starred or un-starred terms can be considered as a complete solution[6]. This work will use the un-starred solution. That is, we will take

$$\begin{aligned}
u(z,\theta) &= e^{i\omega t}\sum_{n=0}^{\infty}\sum_{m=0}^{\infty} A_{mn}cos\left(\frac{m\pi z}{l}\right)cos(n\theta) \\
v(z,\theta) &= e^{i\omega t}\sum_{n=0}^{\infty}\sum_{m=0}^{\infty} B_{mn}sin\left(\frac{m\pi z}{l}\right)sin(n\theta) \\
w(z,\theta) &= e^{i\omega t}\sum_{n=0}^{\infty}\sum_{m=0}^{\infty} C_{mn}sin\left(\frac{m\pi z}{l}\right)cos(n\theta)
\end{aligned} \tag{33}$$

The end conditions shown in Equation 31 represent "shear diaphragms"[7]. Non—axial displacements at the ends of the cylindrical shell (at $z = 0$ and $z = l$) are zero. Substituting Equation 33 into Equation 29 yields:

$$\begin{bmatrix} (-\lambda^2 - \frac{1-\nu}{2}\lambda n^2 + \Omega_{mn}^2) & \frac{1+\nu}{2}\lambda n & \nu\lambda \\ \frac{1+\nu}{2}\lambda n & (-\frac{1-\nu}{2}\lambda^2 - n^2 + \Omega_{mn}^2) & -n \\ -\nu\lambda & n & \left(1 + \frac{h^2}{12a^2}(\lambda^2 + n^2)^2 - \Omega_{mn}^2\right) \end{bmatrix} \begin{bmatrix} A_{mn} \\ B_{mn} \\ C_{mn} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{34}$$

where

$$\Omega_{mn}^2 = \frac{\rho_s(1-\nu^2)a^2\omega_{mn}^2}{E} \tag{35}$$

and

$$\lambda = \frac{m\pi a}{l} \tag{36}$$

For non-trivial solutions to Equation 34, the determinant of the coefficient matrix is set equal to zero. Doing so produces the following characteristic equation in Ω_{mn}^2[7]:

$$\Omega_{mn}^6 - K_2\Omega_{mn}^4 + K_1\Omega_{mn}^2 - K_0 = 0 \tag{37}$$

where

$$\begin{aligned} K_2 &= 1 + \frac{3-\nu}{2}(n^2+\lambda^2) + \frac{h^2}{12a^2}(n^2+\lambda^2)^2 \\ K_1 &= \frac{1-\nu}{2}\left[(3+2\nu)\lambda^2 + n^2 + (n^2+\lambda^2)^2 + \left(\frac{3-\nu}{1-\nu}\right)\frac{h^2}{12a^2}(n^2+\lambda^2)^3\right] \\ K_0 &= \frac{1-\nu}{2}\left[(1-\nu^2)\lambda^4 + \frac{h^2}{12a^2}(n^2+\lambda^2)^4\right] \end{aligned} \tag{38}$$

The roots of this equation in combination with Equation 35 can be used to calculate the natural frequencies ω_{mn} of the cylindrical shell.

A more accurate description of the motion of thin cylindrical shell is provided by the Epstein-Kennard equations of motion. Unlike the Donnell-Mushtari formulation, the Epstein-Kennard relation does not neglect changes in shear stresses acting through the thickness of the shell. For a more detailed analysis of this particular shell theory, the reader is referred to the literature [7]. Here, it is noted that, with respect to the natural frequencies of vibration, Equation 37 becomes[7]

$$\Omega_{mn}^6 - (K_2 + k\Delta K_2)\Omega_{mn}^4 + (K_1 + k\Delta K_1)\Omega_{mn}^2 - (K_0 + k\Delta K_0) = 0 \tag{39}$$

where, for the Epstein-Kennard theory,

$$\begin{aligned}
\Delta K_2 &= \frac{1+3\nu}{1-\nu} - \frac{(2-8\nu^2+3\nu^3)\lambda^2}{2(1-\nu)^2} - \frac{(19-37\nu+19\nu^2+\nu^3)}{2(1-\nu)^2} - \frac{\nu^2(n^2+\lambda^2)}{(1-\nu)^2} \\
\Delta K_1 &= \frac{(3+8\nu-5\nu^2-\nu^3)\lambda^2}{2(1-\nu)} + \frac{(2+\nu)n^2}{2} - \frac{(6+4\nu-8\nu^2+3\nu^3-8\nu^4)\lambda^4}{4(1-\nu)} - \frac{\nu^2(n^2+\lambda^2)^3}{2(1-\nu)} \\
&\quad - \frac{(26-60\nu+40\nu^2-3\nu^3-8\nu^4)\lambda^2 n^2}{2(1-\nu)} - \frac{(13-22\nu+10\nu^2)n^4}{2(1-\nu)} \\
\Delta K_0 &= \frac{1}{2}(1-\nu)\left[\frac{(2+6\nu-2\nu^2-3\nu^3)\lambda^4}{2(1-\nu)} + 4\lambda^2 n^2 + n^4 - \frac{1+\nu}{1-\nu}\lambda^6 - \frac{7-5\nu}{1-\nu}\lambda^4 n^2 - 8\lambda^2 n^4 - 2n^6\right]
\end{aligned} \tag{40}$$

and

$$k = \frac{h^2}{12a^2} \tag{41}$$

and the natural frequencies can be calculated as before.

The Free Fluid Problem

For the fluid-filled region inside the cylindrical shell, the wave equation in cylindrical coordinates (Equation 24) can be used. The boundary conditions now become

$$\begin{gathered}
P(r,\theta,0) = P(r,\theta,l) = 0 \\
\frac{\partial P(a,\theta,z)}{\partial r} = \rho_f \frac{\partial^2 w(\theta,z)}{\partial t^2} \\
|P(0,\theta,z)| < \infty \text{ (boundedness)} \\
P(r,\theta,z) = P(r,\theta+2\pi n,z) \text{ (periodicity)}
\end{gathered} \tag{42}$$

where $w(\theta,z)$ represents the radial displacement of the cylindrical shell. That is, just as in the case for the cubic geometry, the fluid is constrained to match the behavior of the structure at those points where they are in contact. Applying these boundary conditions and using the results for the cylindrical shell yields the following expression for P^5:

$$P(r,\theta,z) = e^{i\omega t}\sum_{n=0}^{\infty}\sum_{m=1}^{\infty} D_{mn} J_n(\alpha_{mn} r) sin\left(\frac{m\pi z}{l}\right) cos(n\theta) \tag{43}$$

where

$$D_{mn} = \frac{C_{mn}\rho_f\omega^2}{\alpha_{mn}\left[\frac{n}{a}J_n(\alpha_{mn}a) - \alpha_{mn}J_{n+1}(\alpha_{mn}a)\right]} \tag{44}$$

and

$$\alpha_{mn}^2 = \left(\frac{\omega_{mn}}{c_o}\right)^2 - \left(\frac{m\pi}{l}\right)^2 \tag{45}$$

Three cylindrical fluid/structure models were constructed for NASTRAN normal modes analysis. The three models were discretized as shown in Table 8. The fluid and structure elements within each of the models were given the material properties shown in Table 6. Boundary conditions as shown in Equations 31 and 42 were applied on each model.

Model	**Material**	**Element Type**	**Number of Nodes**	**Total Nodes in Model**
1	Structure	480 Linear QUAD4	504	1953
	Fluid	2240 Linear WEDGE6	1449	
2	Structural	192 Quadratic QUAD8	608	2729
	Fluid	672 Quadratic WEDGE15	2121	
3	Structure	480 Quadratic QUAD8	1488	8097
	Fluid	2240 Quadratic WEDGE15	6609	

Table 8 NASTRAN models for fluid/structure problem, cylindrical geometry.

From Equation 35, using Epstein-Kennard theory, the natural frequency of the cylindrical shell for mode 1,1 is 6248.99 Hz. Using the linear QUAD4 model, NASTRAN calculated an eigenfrequency of 6251.14 Hz. The 192-element QUAD8 model yielded a frequency for this mode of 6256.34 Hz. A frequency of 6245.55 Hz was calculated using the 480-element QUAD8 model. This mode shape is shown for the u, v, and w displacements for each model in Figures A50, A51, A52, A53, A54, A55, A56, A57, and A58. The "breathing modes" of the shell, i.e. modes where m=1 and n=0 in Equation 33 occur (using Epstein-Kennard theory) for this model at 12,318.19 Hz, 19,585.91 Hz and 35,037.39 Hz. The second frequency, 19,585.91 Hz, is shown for each model in Figures A59, A60, A61, A62, A63, A64, A65, A66, and A67, with error plots for the radial displacements in Figures A68, A69, and A70.

For the fluid inside the cylindrical shell, fringe plots of NASTRAN-calculated mode 1,0,1 is shown in Figures A71 (linear WEDGE6 elements), A72 (672-element WEDGE15 model) and A73 (2240-element WEDGE15 model). A fringe plot showing the difference between the analytic and numeric solutions of this mode is shown in Figures A74, A75, and A76. A summary of the normal modes analysis for the structure portion of the fluid/structure cylinder is shown in Table 9.

Model Type	Elements	Nodes	Mode Shape	Eigenvalue Error	Maximum Error in Model
Fluid/struct.	2240 Linear WEDGE6	1449	1,0,1	0.102%	98.4×10^{-6}
	672 Quadratic WEDGE15	2121		2.937×10^{-4}%	0.0824
	2240 Quadratic WEDGE15	6609		7.34×10^{-5}%	101.3×10^{-6}
	480 Linear QUAD4	504	1,1	0.143%	N/A
	192 Quadratic QUAD8	608		0.118%	N/A
	480 Quadratic QUAD8	1488		0.055%	N/A
	480 Linear QUAD4	504	1,0	0.397%	0.087×10^{-6}
	192 Quadratic QUAD8	608		0.294%	-0.0270
	480 Quadratic QUAD8	1488		0.040%	13.279×10^{-3}

Table 9 Results for cylindrical fluid/structure geometry, normal modes analysis.

The Forced Fluid/Structure Problem

The third problem considered for the cylindrical fluid/structure model was that of a forced response. As we did for the cubic model, we first consider the analytic partially-coupled solution of this problem. The cylindrical shell equations of motion with viscous damping are given by[5]:

$$\begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{bmatrix}_{DM} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \frac{1-\nu^2}{Eh} \begin{bmatrix} f_z \\ f_\theta \\ -f_r \end{bmatrix} \tag{46}$$

where

$$\begin{aligned}
L_{11} &= -\omega^2 + 2i \\
L_{12} &= -\frac{1+\nu}{2a}\frac{\partial^2}{\partial z \partial \theta} \\
L_{13} &= -\frac{\nu}{a}\frac{\partial}{\partial z} \\
L_{21} &= -\frac{1+\nu}{2a}\frac{\partial^2}{\partial z \partial \theta} \\
L_{22} &= \frac{1}{C_L^2}\frac{\partial^2}{\partial t^2} + \gamma\frac{(1-\nu^2)}{Eh}\frac{\partial}{\partial t} - \frac{1-\nu}{2}\frac{\partial^2}{\partial z^2} - \frac{1}{a^2}\frac{\partial^2}{\partial \theta^2} \\
L_{23} &= -\frac{1}{a^2}\frac{\partial}{\partial \theta} \\
L_{31} &= \frac{\nu}{a}\frac{\partial}{\partial z} \\
L_{32} &= \frac{1}{a^2}\frac{\partial}{\partial \theta} \\
L_{33} &= \frac{1}{C_L^2}\frac{\partial^2}{\partial t^2} + \gamma\frac{(1-\nu^2)}{Eh}\frac{\partial}{\partial t} + \frac{1}{a^2} + \frac{h^2}{12}\nabla^4
\end{aligned} \tag{47}$$

and f_z, f_θ, and f_r represent forces in the axial, circumferential, and radial directions. Here the convention of a positive inward-directed radial force has been assumed[5]. Applying forces to the cylinder of

$$\begin{aligned}
f_z &= 0 \\
f_\theta &= 0 \\
f_r &= F_o sin(\omega t)\delta\left(z - \frac{l}{2}\right)[\delta(\theta) + \delta(\theta - \pi)],
\end{aligned} \tag{48}$$

expanding these forces in terms of the eigenfunctions of the shell and once again using Equation 33 with the boundary conditions of Equation 31, Equation 46 yields

$$\begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} A_{mn} \\ B_{mn} \\ C_{mn} \end{bmatrix} = -\frac{1}{\rho_s h}\begin{bmatrix} 0 \\ 0 \\ F_{mn} \end{bmatrix} \tag{49}$$

where now

$$\begin{aligned}
L_{11} &= -\omega^2 + 2i\eta\omega\omega_{mn} + \lambda^2 C_L^2 + \frac{1-\nu}{2a^2} n^2 C_L^2 \\
L_{12} = L_{21} &= -\frac{1+\nu}{2a} n\lambda C_L^2 \\
L_{13} = L_{31} &= -\frac{\nu}{a}\lambda C_L^2 \\
L_{22} &= -\omega^2 + 2i\eta\omega\omega_{mn} + \frac{1-\nu}{2}\lambda^2 C_L^2 + \frac{n^2}{a^2} C_L^2 \\
L_{23} = L_{32} &= \frac{n}{a^2} C_L^2 \\
L_{33} &= -\omega^2 + 2i\eta\omega\omega_{mn} + \frac{1}{a^2} C_L^2 + \frac{h^2 C_L^2}{12}(\lambda^2 a^2 + n^2)^2
\end{aligned} \tag{50}$$

$$\lambda = \frac{m\pi}{l} \tag{51}$$

and F_{mn} is given by

$$F_{mn} = \frac{4}{\pi l} F_o sin\left(\frac{m\pi}{2}\right)[cos(n\pi) + 1] \tag{52}$$

Note that in these equations, the viscous damping term γ has once again been replaced by a frequency-dependent damping term η.

For the fluid inside the shell, we proceed as we did for the undamped, free vibration case. That is, applying the boundary conditions of Equation 42 to the cylindrical wave equation (Equation 24), the solution for the acoustic pressure field becomes:[5]

$$P(r,\theta,z) = e^{i\omega t} \sum_{n=0}^{\infty} \sum_{m=1}^{\infty} D_{mn} J_n(\alpha_{mn} r) sin\left(\frac{m\pi z}{l}\right) cos(n\theta) \tag{53}$$

with

$$D_{mn} = \frac{C_{mn}\rho_f \omega^2}{\alpha_{mn}\left[\frac{n}{a} J_n(\alpha_{mn} a) - \alpha_{mn} J_{n+1}(\alpha_{mn} a)\right]} \tag{54}$$

and

$$\alpha_{mn}^2 = \left(\frac{\omega_{mn}}{c_o}\right)^2 - \left(\frac{m\pi}{l}\right)^2 \tag{55}$$

For the NASTRAN analysis of this problem, a cylindrical finite element model was constructed using linear elements. The finite element mesh was identical to the linear mesh used for the cylindrical free vibrations problems. The geometric characteristic of this problem were modified as shown in Table 10. The structural and fluid elements were given the material properties shown in Table 6.

Radius (a)	10.0 in
Length (l)	50.0 in
Shell Thickness (h)	0.0625 in

Table 10 Geometric dimensions for the cylindrical geometry forced-response analysis model.

Boundary conditions in agreement with Equations 31 and 42 were applied to the appropriate structural and fluid elements respectively. Two point forces of $50 \sin(\omega t)\hat{k}$ lbs., in phase and both pointing radially inward were located at $\theta = 0,\ z = \frac{l}{2}$ and $\theta = \pi,\ z = \frac{l}{2}$. NASTRAN performed a direct frequency response (SOL 108) over the frequency range zero to 5000 Hz.

Analytic and NASTRAN-generated numeric solutions for the frequency range zero to 1000 Hz are shown in Figures A78 and A77. Figure A78 shows the displacement vs. frequency of a structural node located at $r = a,\ \theta = 0,\ z = \frac{l}{2}$. Figure A77 shows the acoustic pressure of a fluid node at location $r = \frac{a}{2},\ \theta = 0,\ z = \frac{l}{2}$.

Section 4: Conclusions

MSC/NASTRAN has been used to numerically calculate a number of different mode shapes for a cubic and a cylindrical acoustic cavity. The accuracy of the NASTRAN fully-coupled fluid/structure solution improves with the use of elements having a greater number of nodes (i.e. linear vs. quadratic elements). The difference in performance between these two elements becomes particularly significant in models involving a curved geometry. This result is not unexpected, as, it is difficult to model accurately a curved shape using only straight linear elements. No substantial change in performance was noted in going from a fluid-only model to a model containing both fluid and structure elements. NASTRAN was able to compute the normal modes for each model quickly and accurately. In sum, NASTRAN's calculation of the normal modes (SOL 103) for a model tended to be simple, direct, quick, and accurate.

Hard disk space limitations were factor that became an important consideration during the forced response analysises, particularly when the direct method (SOL 108) was used. In general, rather than use system memory, NASTRAN writes data to files during the solution of a finite element problem. Although most of these files are deleted when the NASTRAN solution is completed, they can become quite large during the process of solving the problem. If enough free space is not available for use by NASTRAN, the problem cannot be solved. To work around this problem, the forced response analysis for the cylindrical geometry was submitted to NASTRAN as four separate problems, each covering a range of 250 frequencies. After the solutions for each problem were calculated, they were combined to form a complete solution for the entire frequency range of interest. Appendix C of this work contains a listing of the significant files produced by NASTRAN during each analysis and the size of each file. While this method was workable for the problem at hand, it could become very unwieldy with the addition of more nodes and/or elements to the model.

In short, NASTRAN's normal modes analysis was very robust. For a reasonable model discretization, its results can be considered accurate. However, for the cases where a forced response analysis of a fluid/structure model is required, careful consideration should be given to ways in which the model can be simplified and which frequencies are of prime interest.

Acknowledgments

The authors gratefully acknowledge the assistance of Travis Turner and Richard S. Matthews, without whom this work would not have been possible

References

[1] William H. Press. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, first edition, 1990.

[2] Tyn Myint-U with Lokenath Debnath. *Partial Differential Equations for Scientists and Engineers.* PTR Prentice-Hall, third edition, 1987.

[3] Michael Reymond and Mark Miller Editors. *MSC/NASTRAN Quick Reference Guide, Version 68.* The MacNeal-Schwindler Corporation, 1994.

[4] Arthur W. Leissa. *Vibration of Plates.* USGPO, NASA SP-160, 1968.

[5] H. C. Lester and S. Lefebvre. *Piezoelectric Actuator Models for Active Sound and Vibration Control of Cylinders.* Proceedings of the Converence on Recent Advances in Active Control of Sound and Vibration, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, April 15–17, 1991, Tecnomic Publishing Company, 1991.

[6] Richard H. MacNeal. *NASTRAN Theoretical Manual.* The MacNeal-Schwindler Corporation, 1972.

[7] Arthur W. Leissa. *Vibration of Shells.* USGPO, NASA SP-288, 1973.

Appendix A Figures

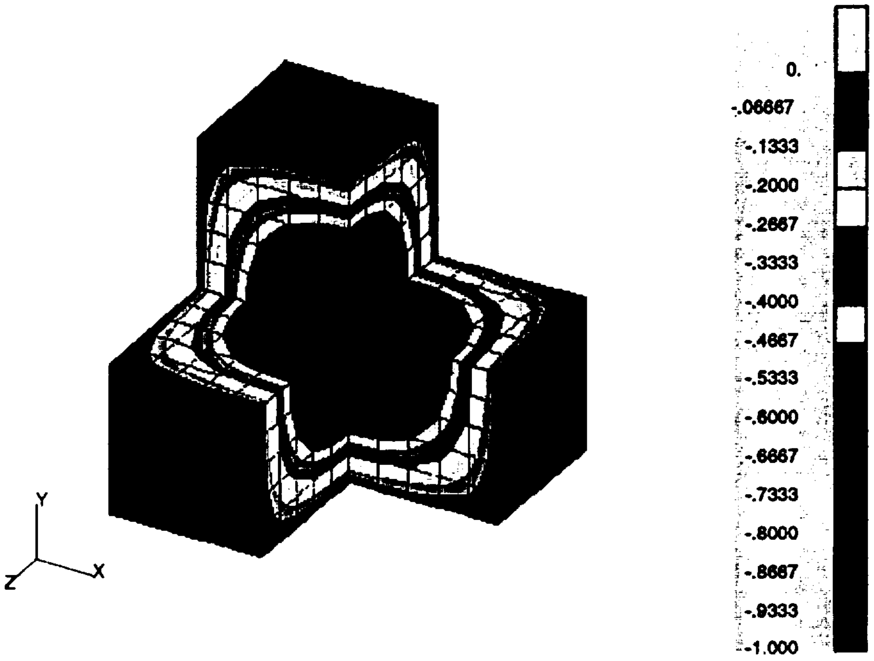

Figure A1: Mode 1,1,1 for cubic geometry, 1000 linear HEX8 elements, 1331 nodes.

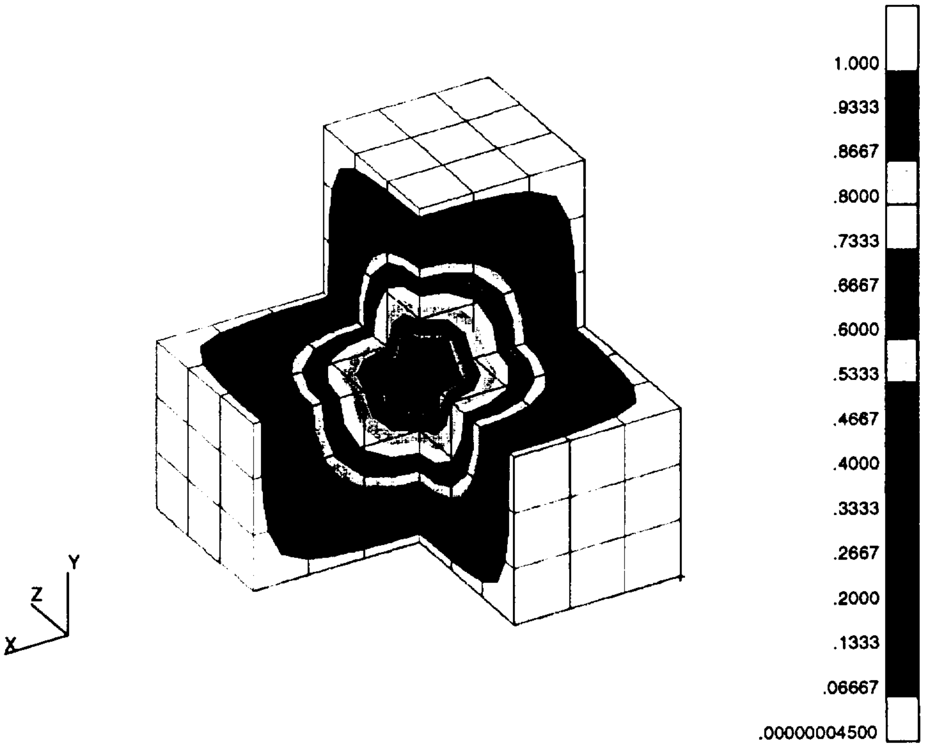

Figure A2: Mode 1,1,1 for cubic geometry,
215 quadratic HEX20 elements, 1225 nodes.

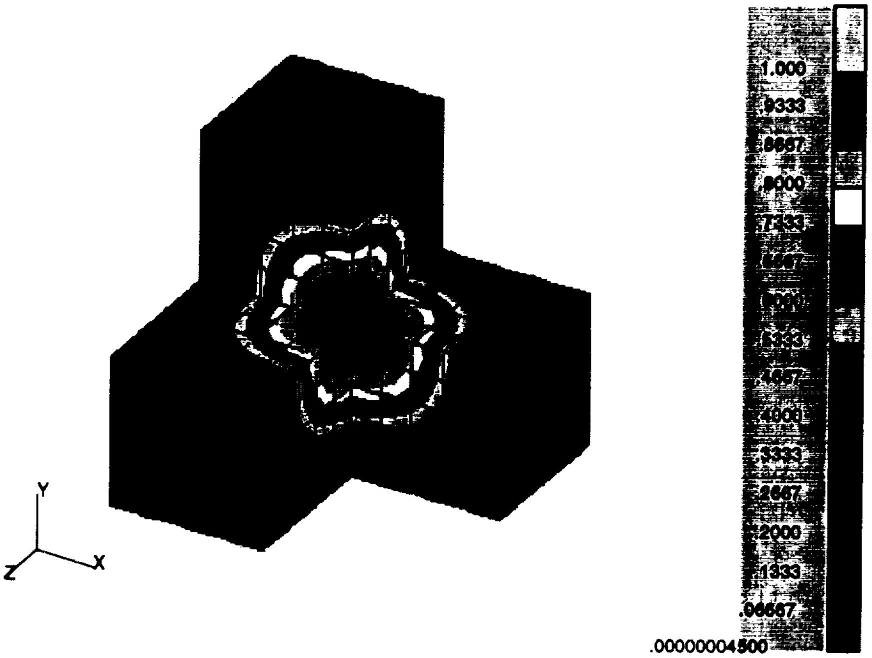

Figure A3: Mode 1,1,1 for cubic geometry, 1000 quadratic HEX20 elements, 4961 nodes.

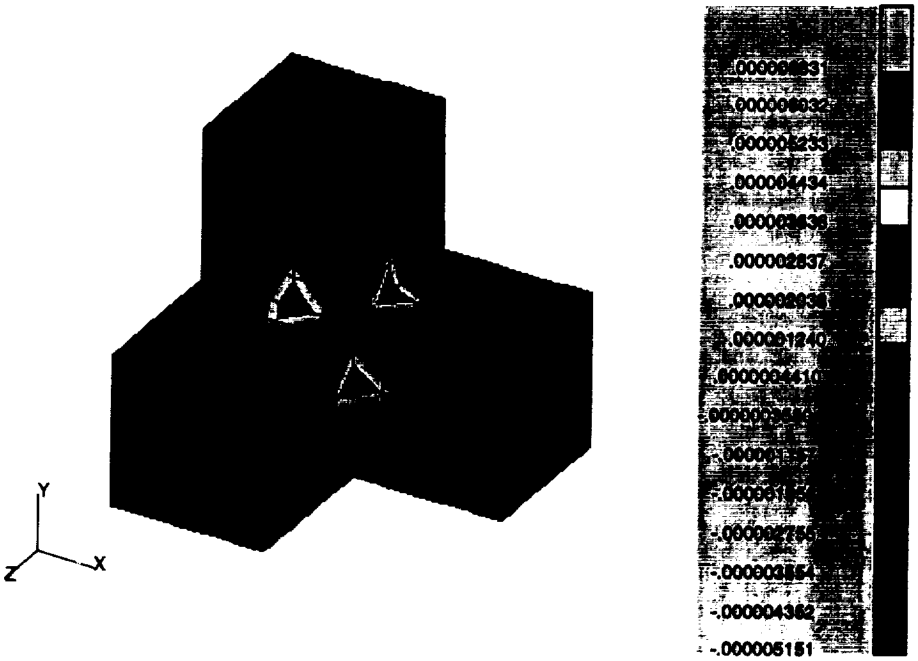

Figure A4: Mode 1,1,1 error for cubic geometry, 1000 linear HEX8 elements, 1331 nodes.

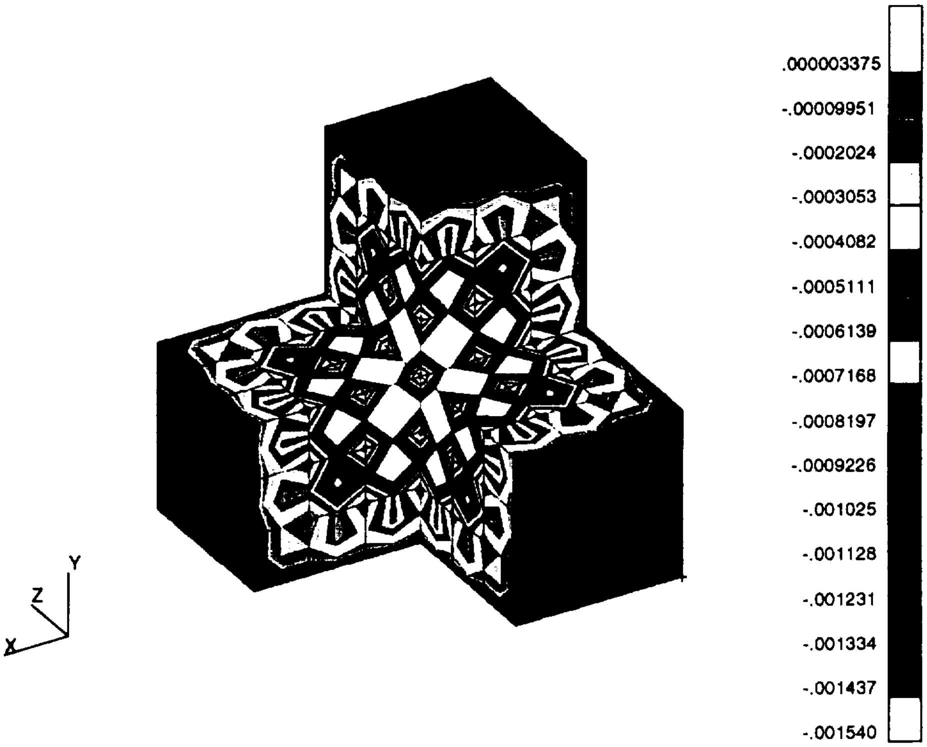

Figure A5: Mode 1,1,1 error for cubic geometry, 215 quadratic HEX20 elements, 1225 nodes.

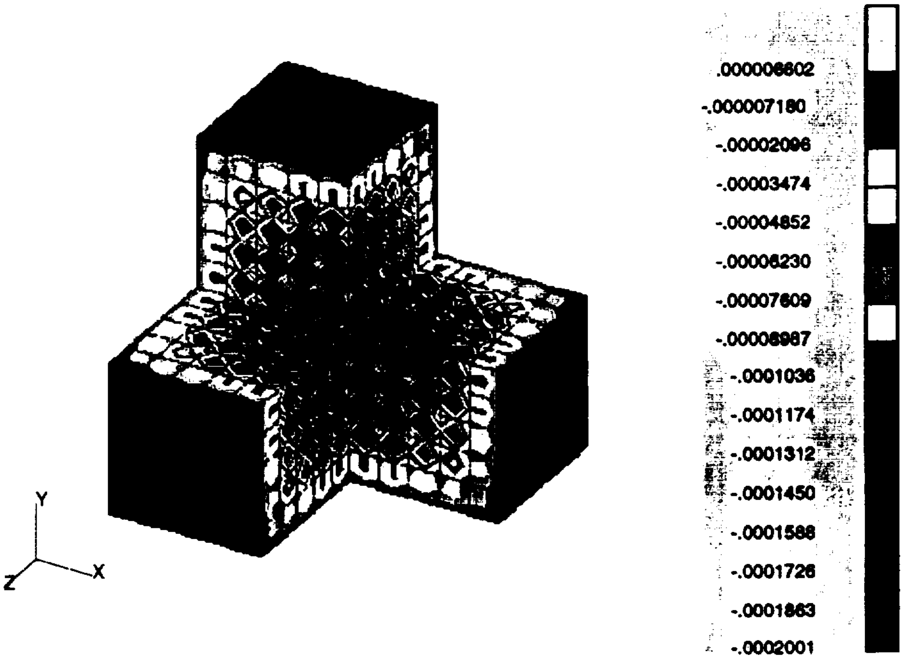

Figure A6: Mode 1,1,1 error for cubic geometry, 1000 quadratic HEX20 elements, 4961 nodes.

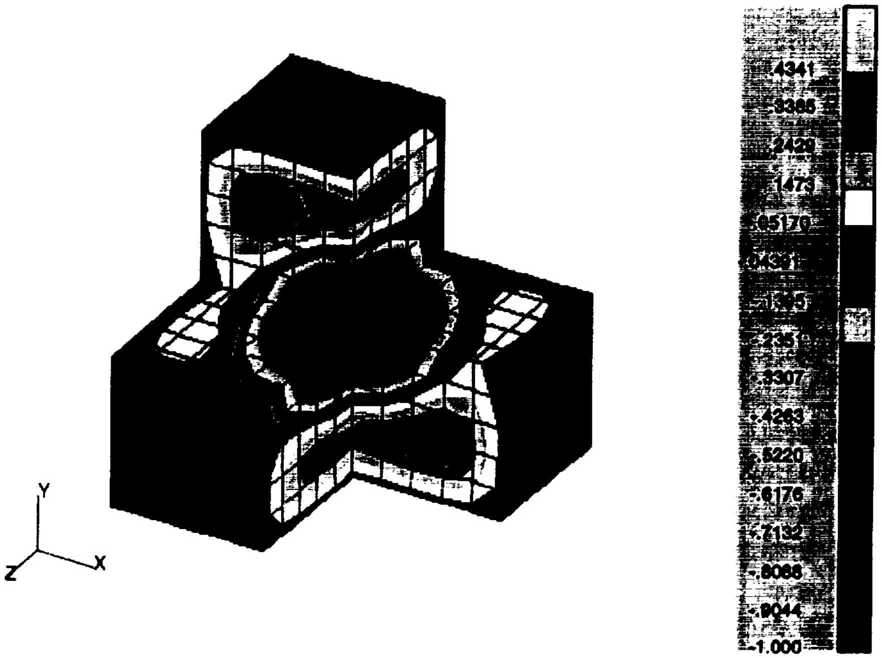

Figure A7: Mode 1,3,1 for cubic geometry, 1000 linear HEX8 elements, 1331 nodes.

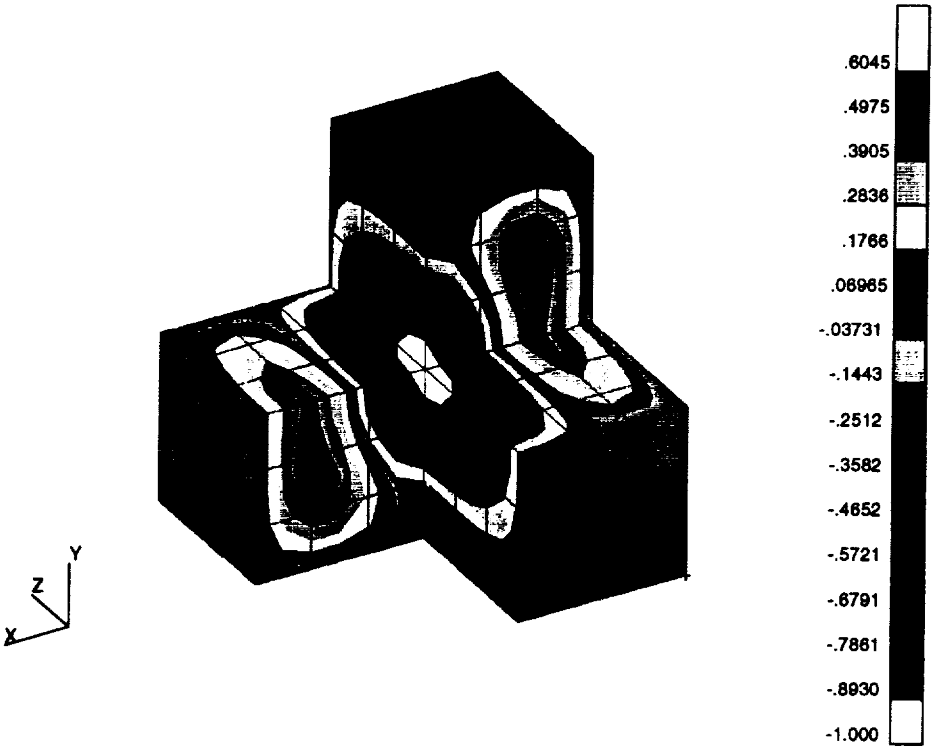

Figure A8: Mode 1,1,3 error for cubic geometry, 215 quadratic HEX20 elements, 1225 nodes.

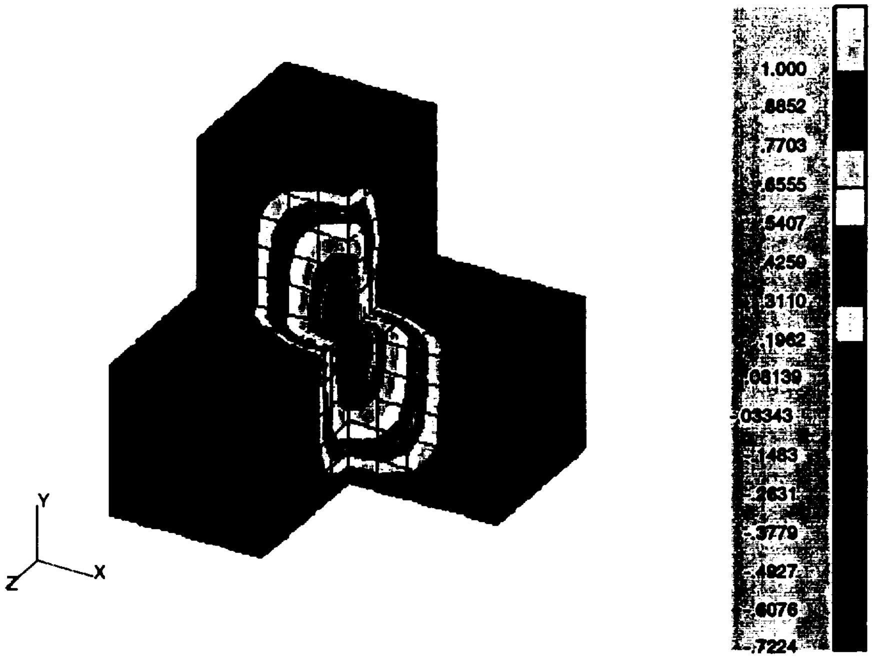

Figure A9: Mode 1,1,3 for cubic geometry, 1000 quadratic HEX20 elements, 4961 nodes.

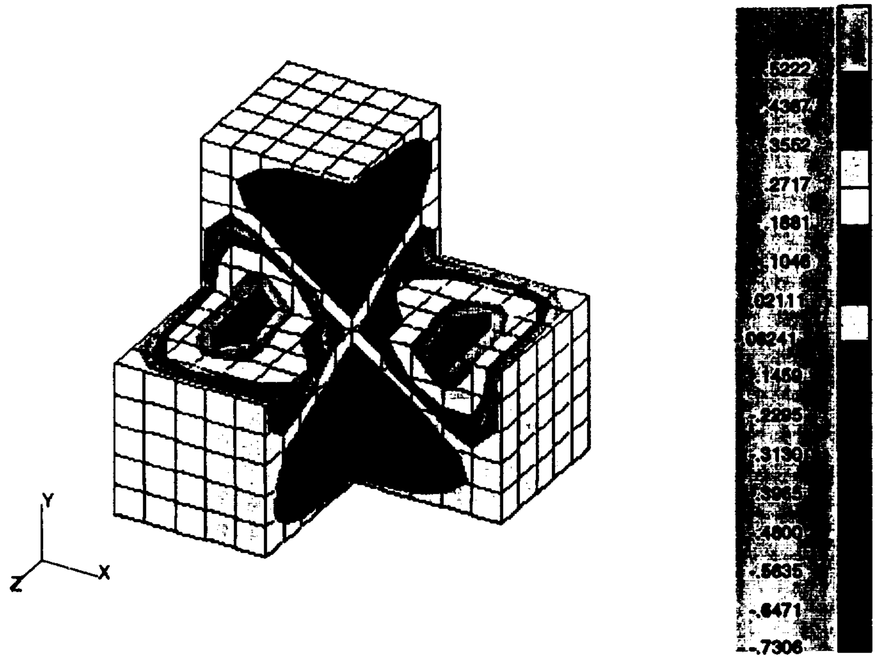

Figure A10: Mode 1,3,1 error for cubic geometry, 1000 linear HEX8 elements, 1331 nodes.

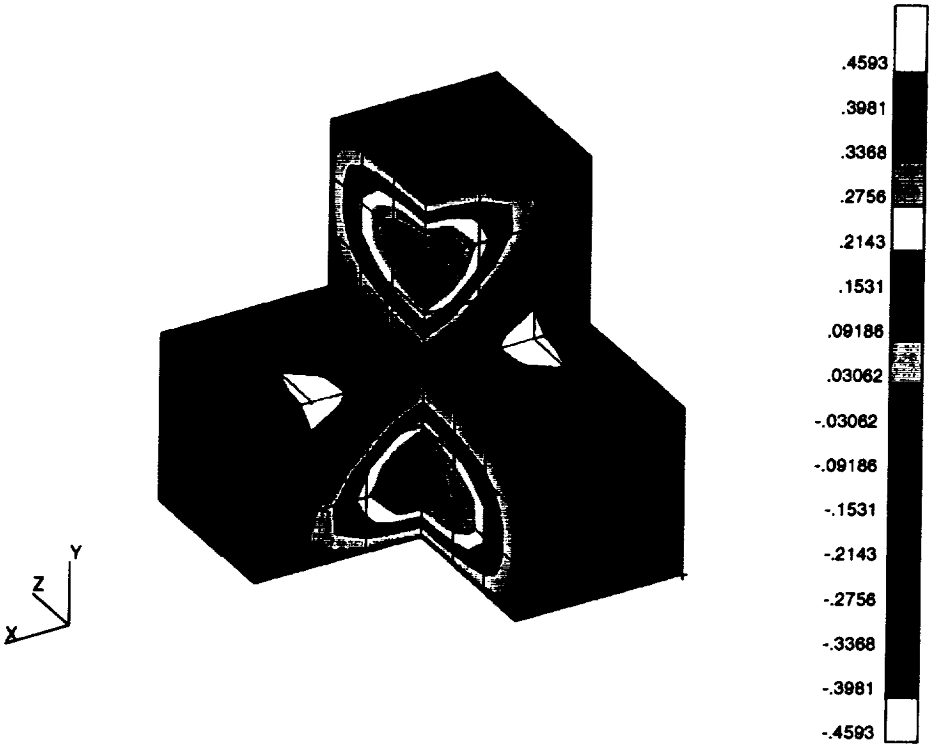

Figure A11: Mode 1,1,3 error for cubic geometry, 215 quadratic HEX20 elements, 1225 nodes.

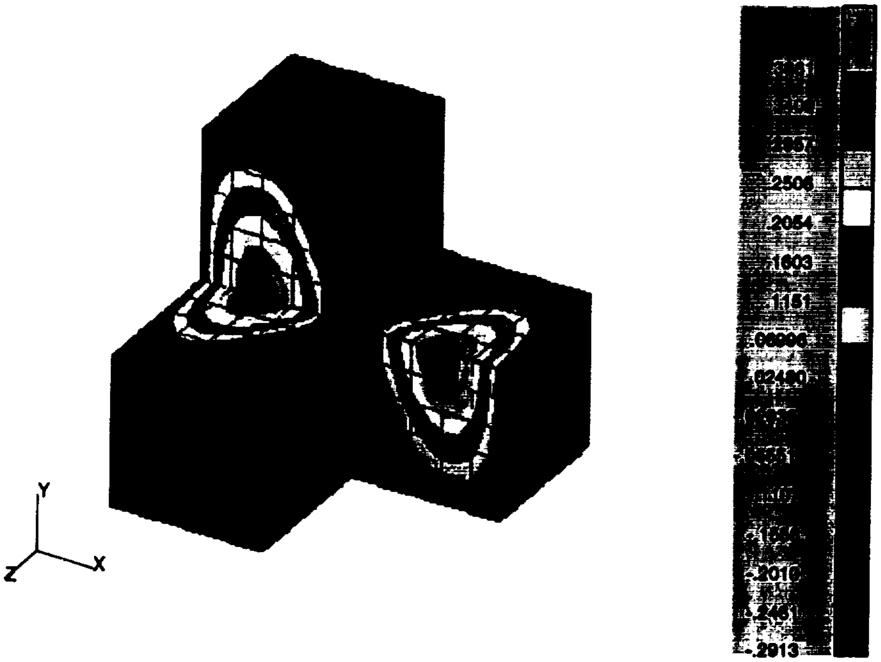

Figure A12: Mode 1,1,3 error for cubic geometry, 1000 quadratic HEX20 elements, 4961 nodes.

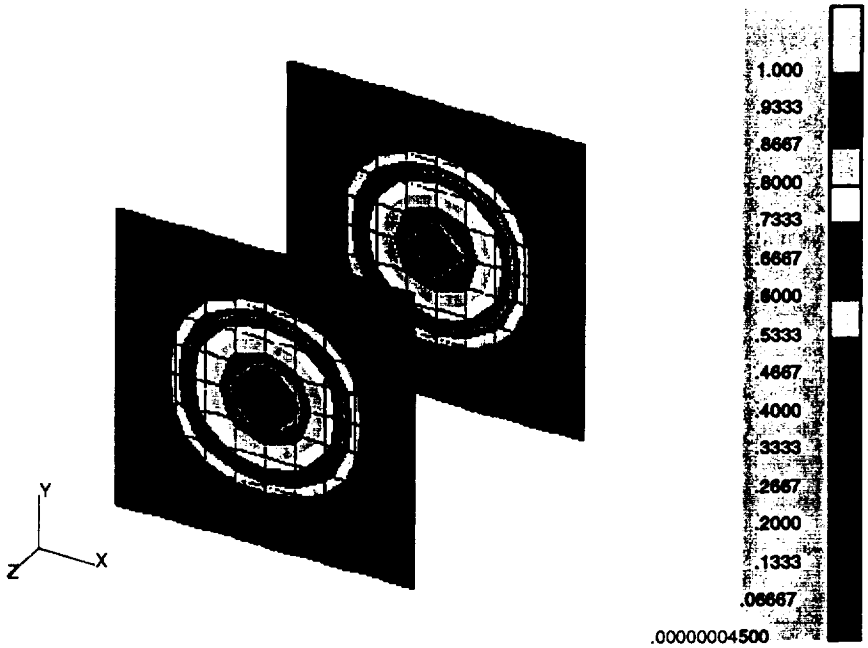

Figure A13: Mode 1,1 for structure portion of fluid/structure cube. 200 linear QUAD4 elements, 242 nodes.

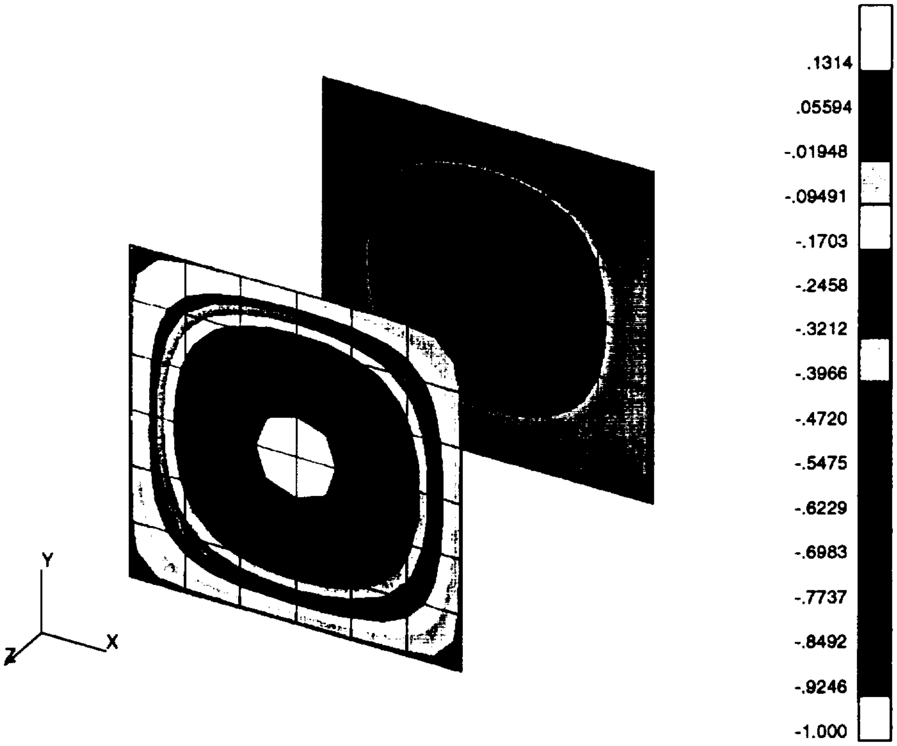

Figure A14: Mode 1,1 for structure portion of fluid/structure cube. 72 quadratic QUAD8 elements, 266 nodes.

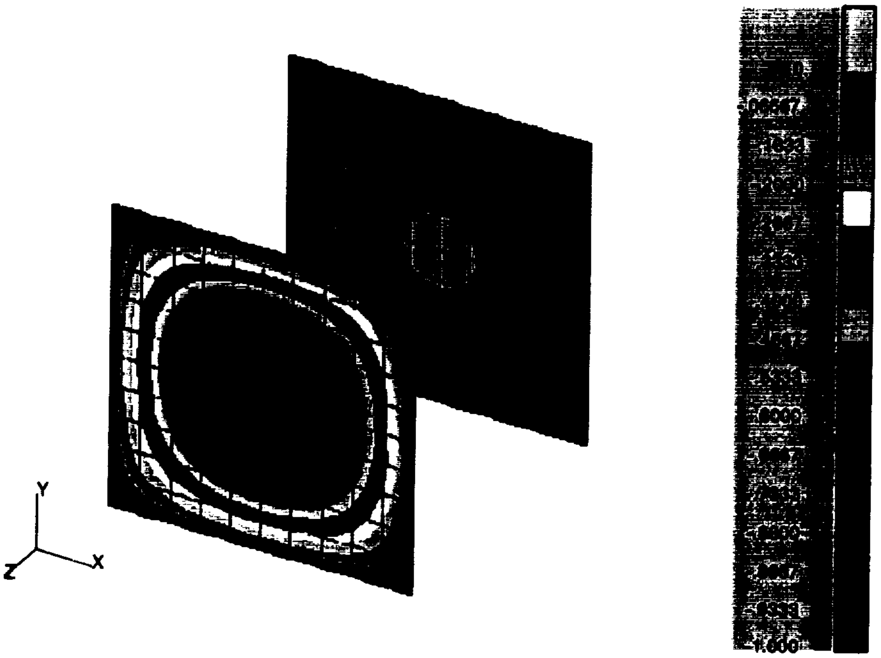

Figure A15: Mode 1,1 for structure portion of fluid/structure cube. 200 quadratic QUAD8 elements, 682 nodes.

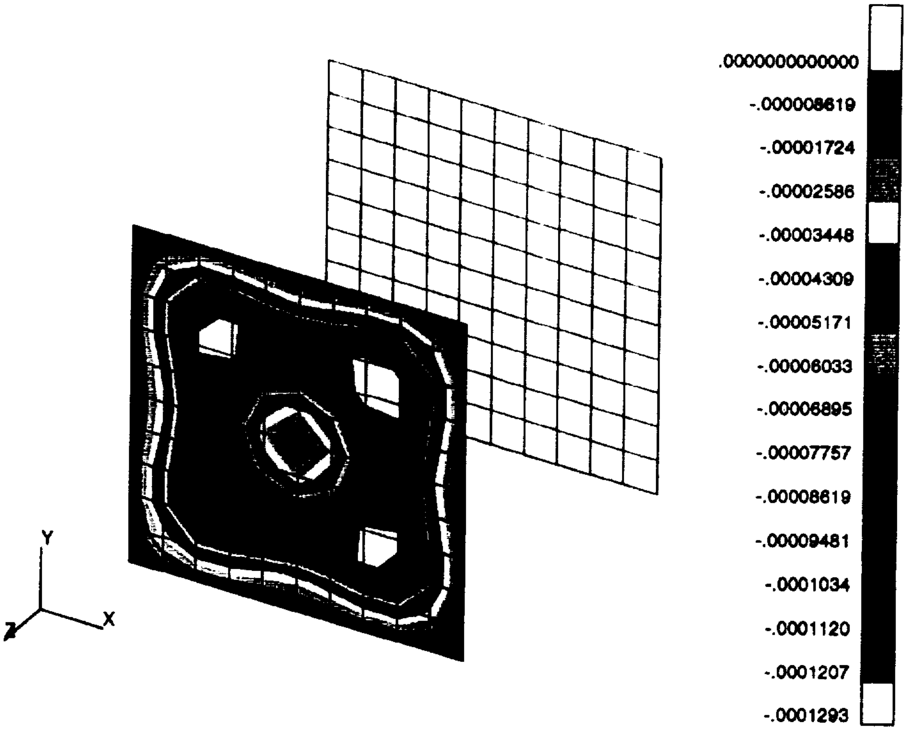

Figure A16: Mode 1,1 error for cubic fluid/structure geometry (front plate only). 200 quadratic QUAD4 elements, 242 nodes.

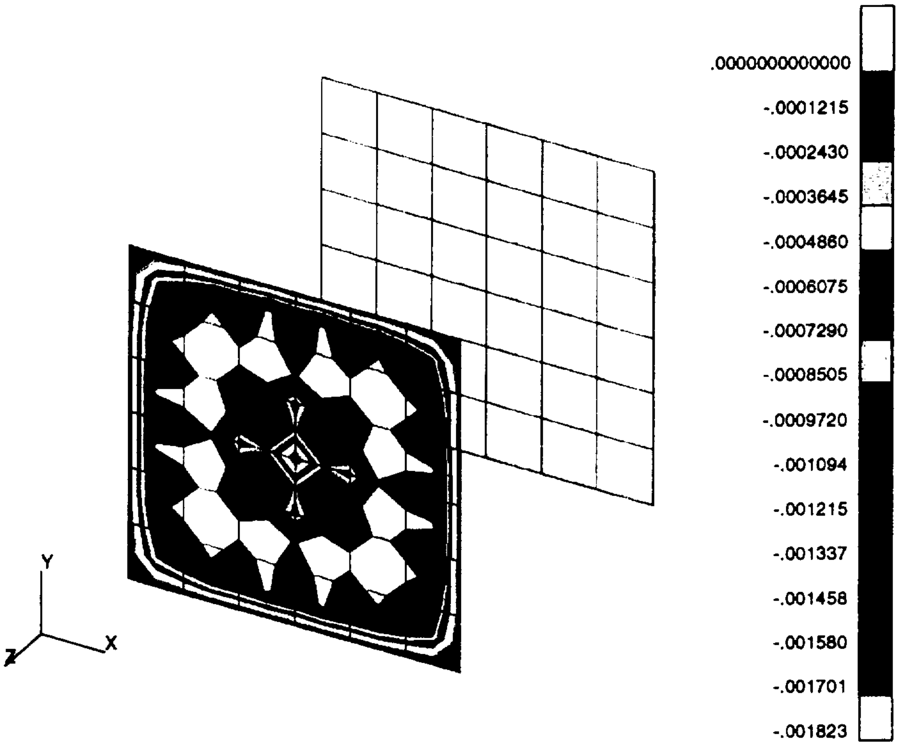

Figure A17: Mode 1,1 error for cubic fluid/structure geometry (front plate only). 72 quadratic QUAD8 elements, 266 nodes.

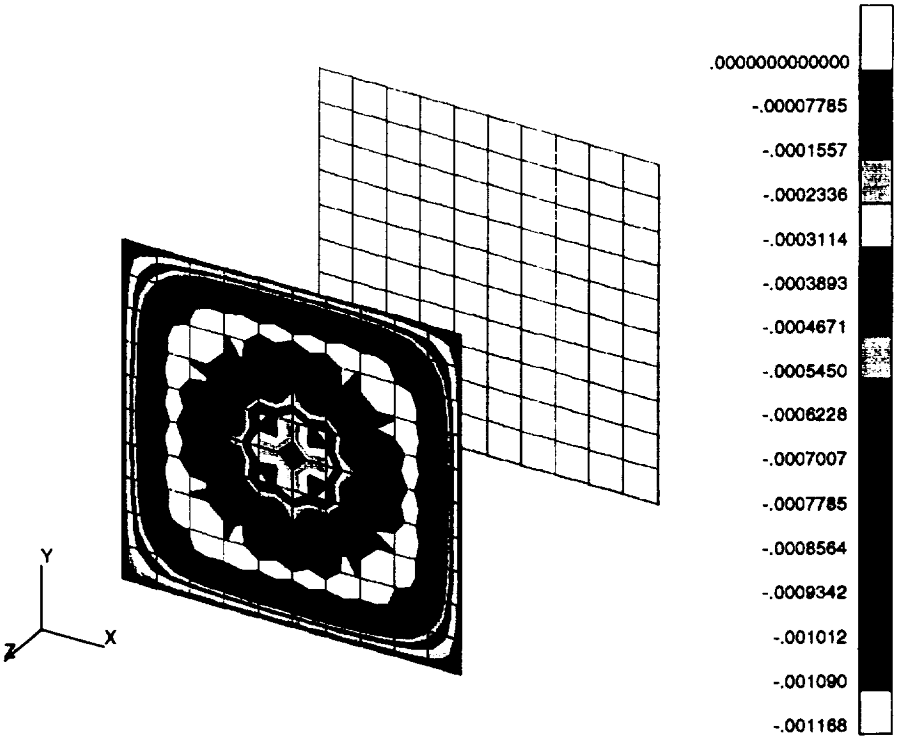

Figure A18: Mode 1,1 error for cubic fluid/structure geometry (structure only). 200 quadratic QUAD8 elements, 682 nodes.

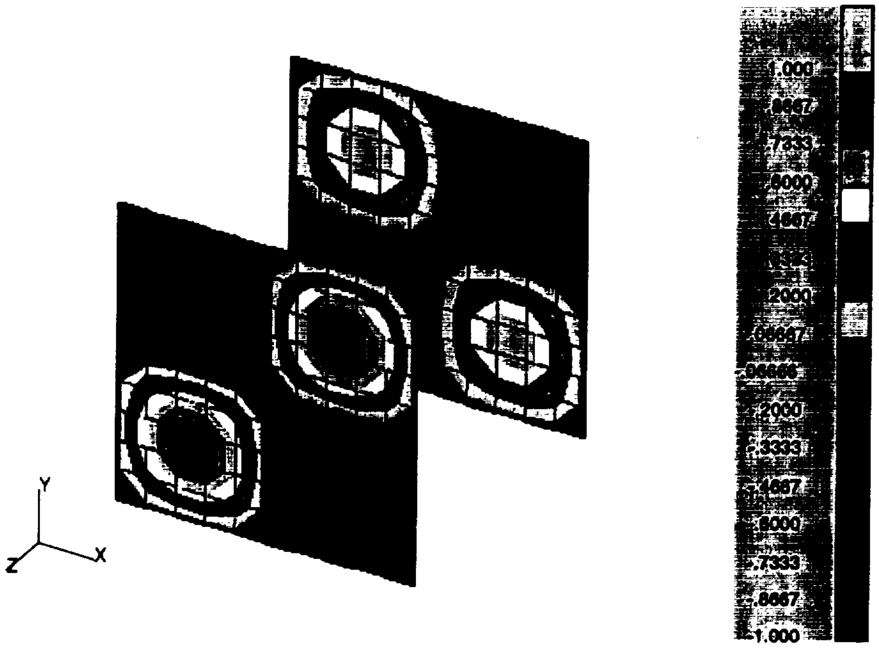

Figure A19: Mode 2,2 for structure portion of fluid/structure cube. 200 linear QUAD4 elements, 242 nodes.

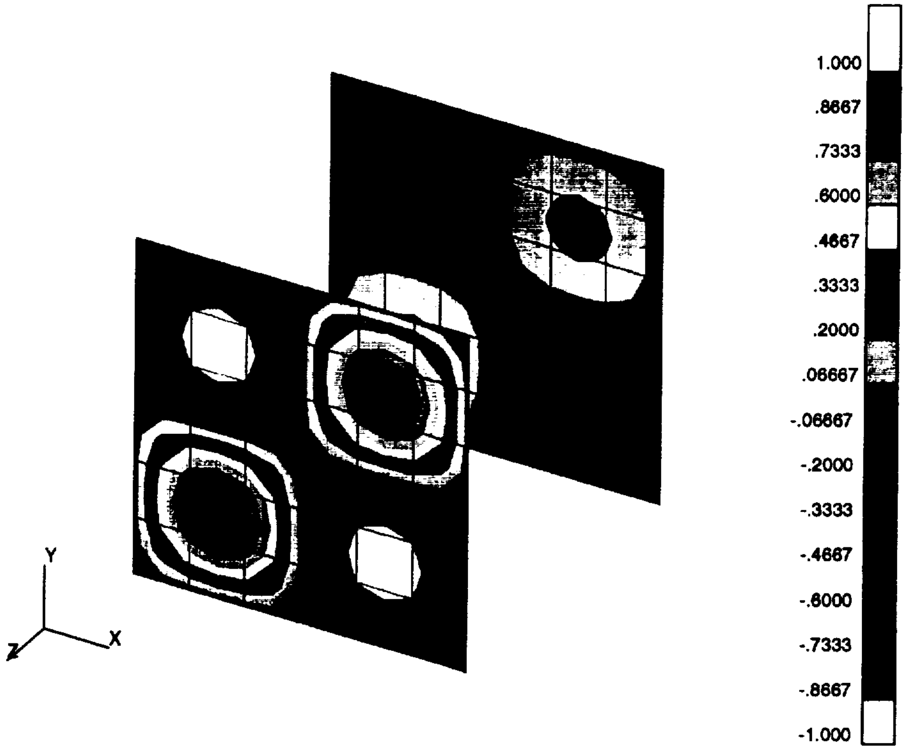

Figure A20: Mode 2,2 for structure portion of fluid/structure cube. 72 quadratic QUAD8 elements, 266 nodes.

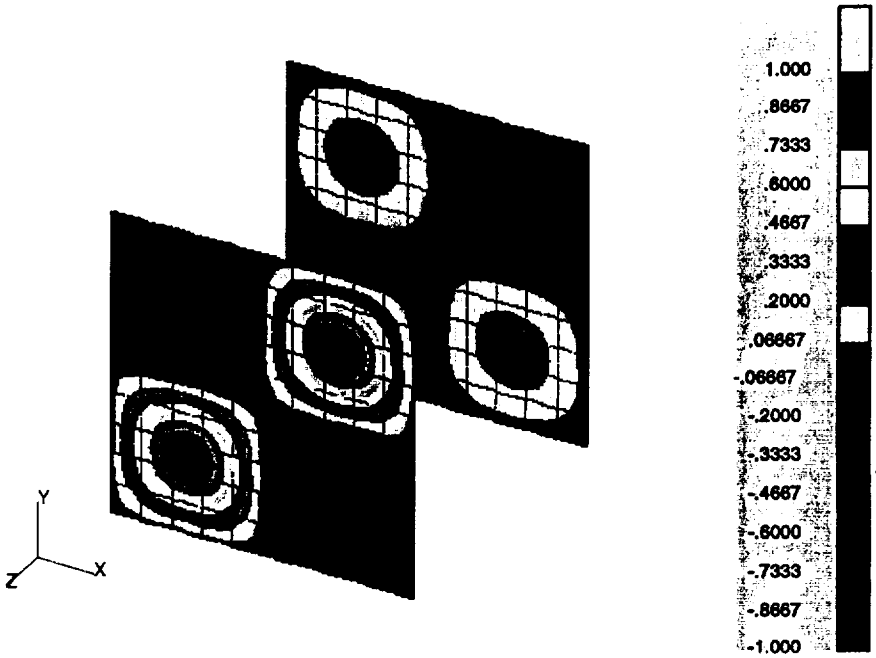

Figure A21: Mode 2,2 for structure portion of fluid/structure cube. 200 QUAD8 elements, 682 nodes.

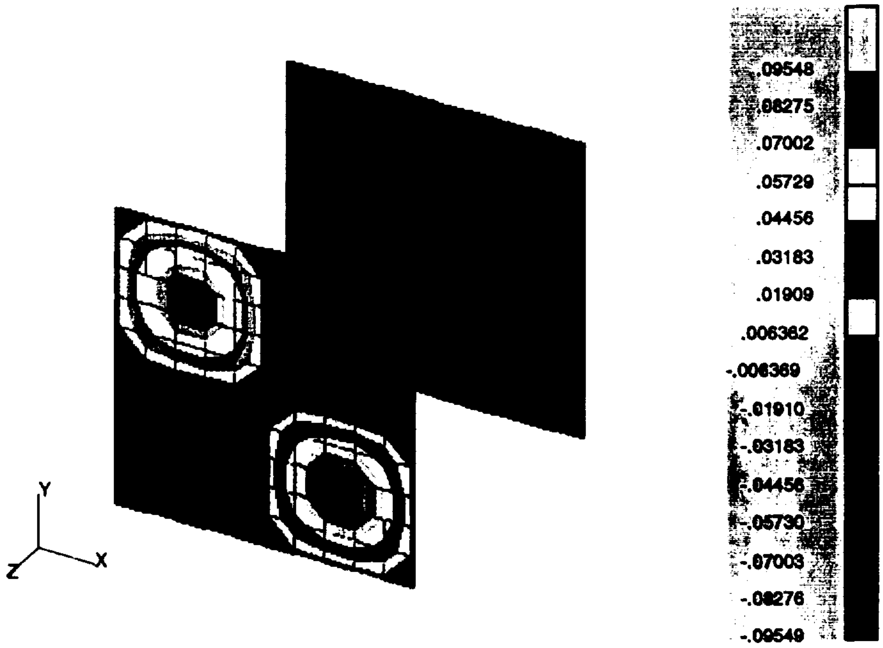

Figure A22: Mode 2,2 error for cubic fluid/structure geometry (front plate only). 200 linear QUAD4 elements, 242 nodes.

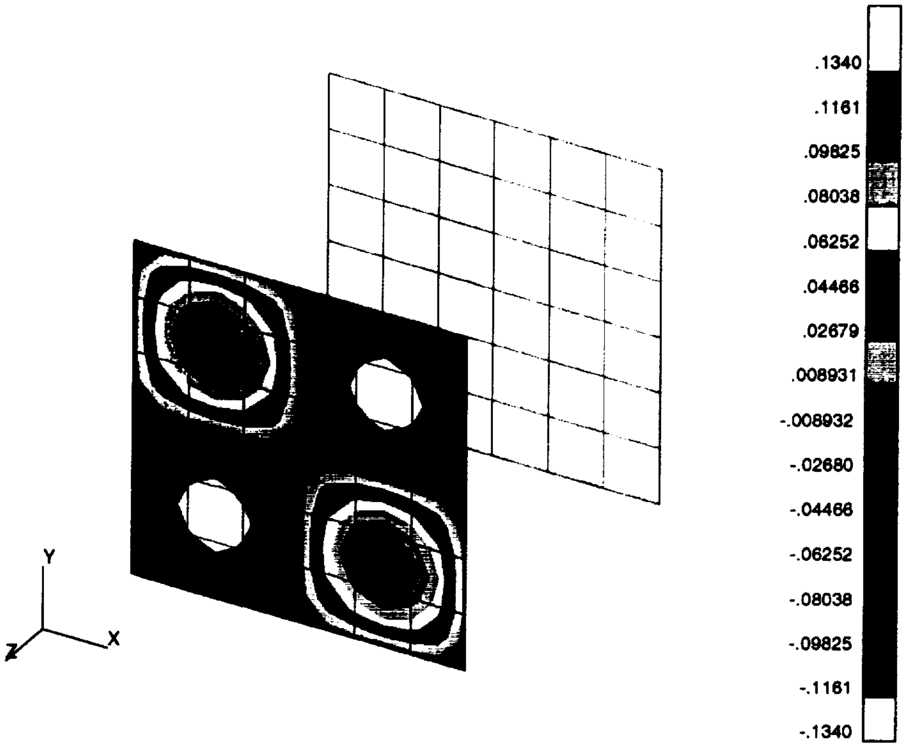

Figure A23: Mode 2,2 error for cubic fluid/structure geometry (front plate only). 72 quadratic QUAD8 elements, 266 nodes.

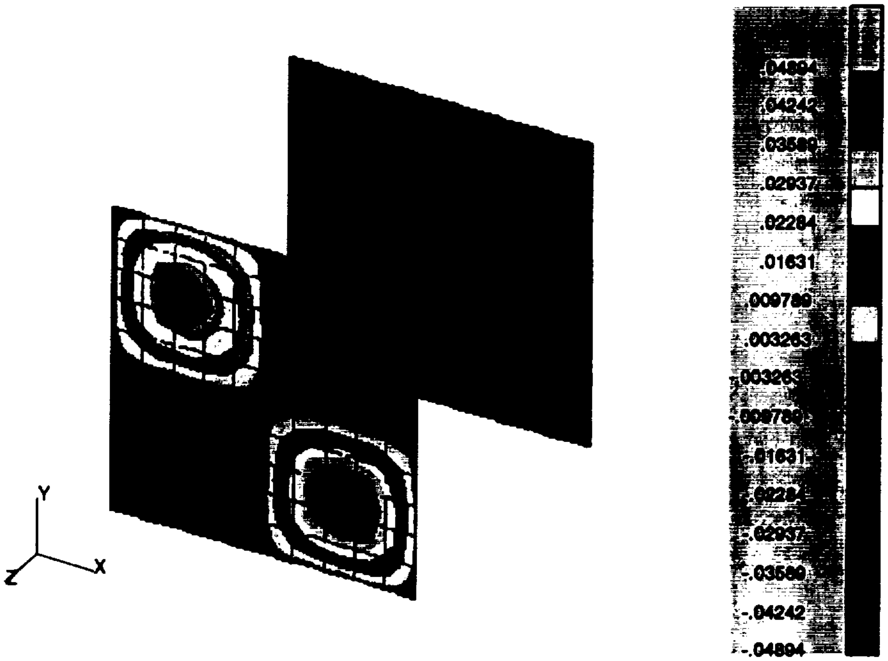

Figure A24: Mode 2,2 error for cubic fluid/structure geometry (front plate only). 200 quadratic QUAD8 elements, 682 nodes.

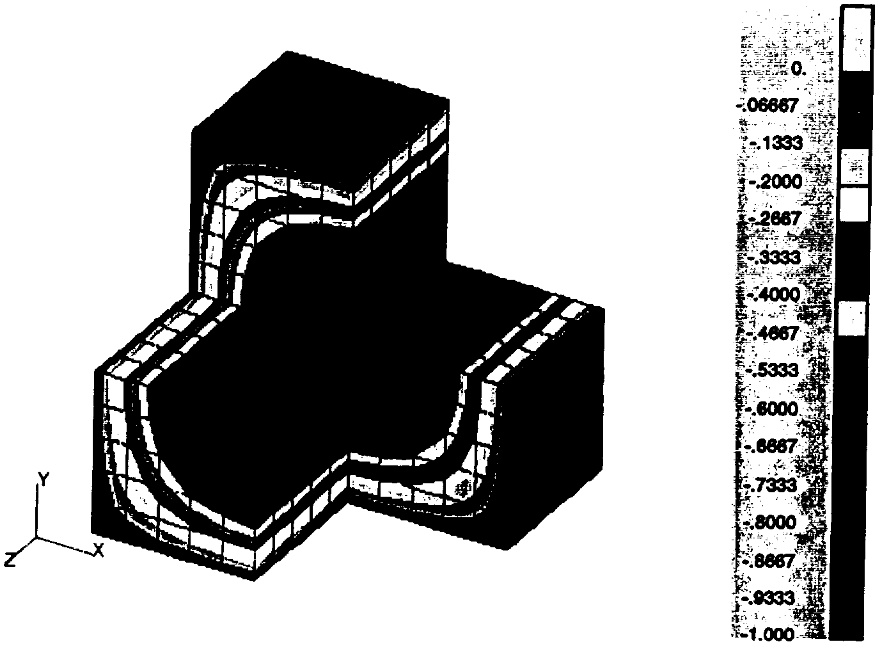

Figure A25: Mode 1,1,0 for fluid portion of fluid/structure cube. 1000 linear HEX8 elements, 1331 nodes.

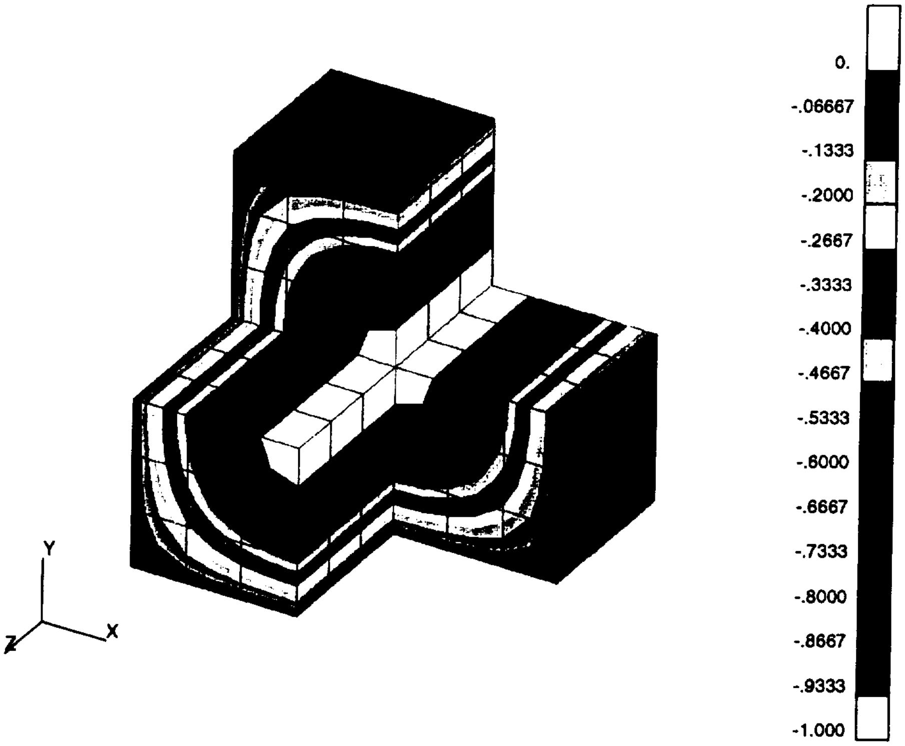

Figure A26: Mode 1,1,0 for fluid portion of fluid/structure cube. 216 quadratic HEX20 elements, 1225 nodes.

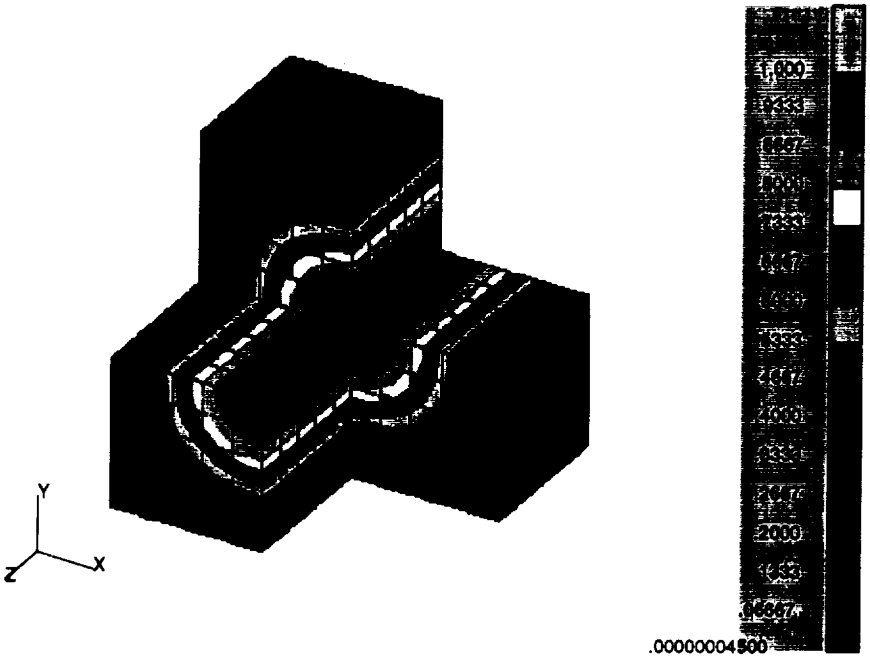

Figure A27: Mode 1,1,0 for fluid portion of fluid/structure cube. 1000 quadratic HEX20 elements, 4962 nodes.

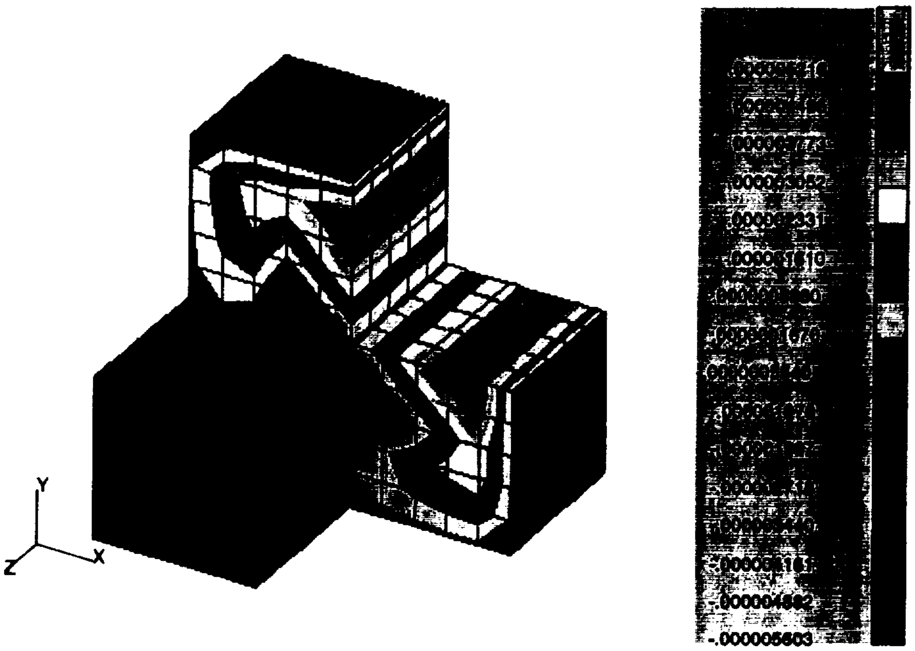

Figure A28: Mode 1,1,0 error for fluid/structure geometry (fluid only). 1000 linear HEX8 elements, 1331 nodes.

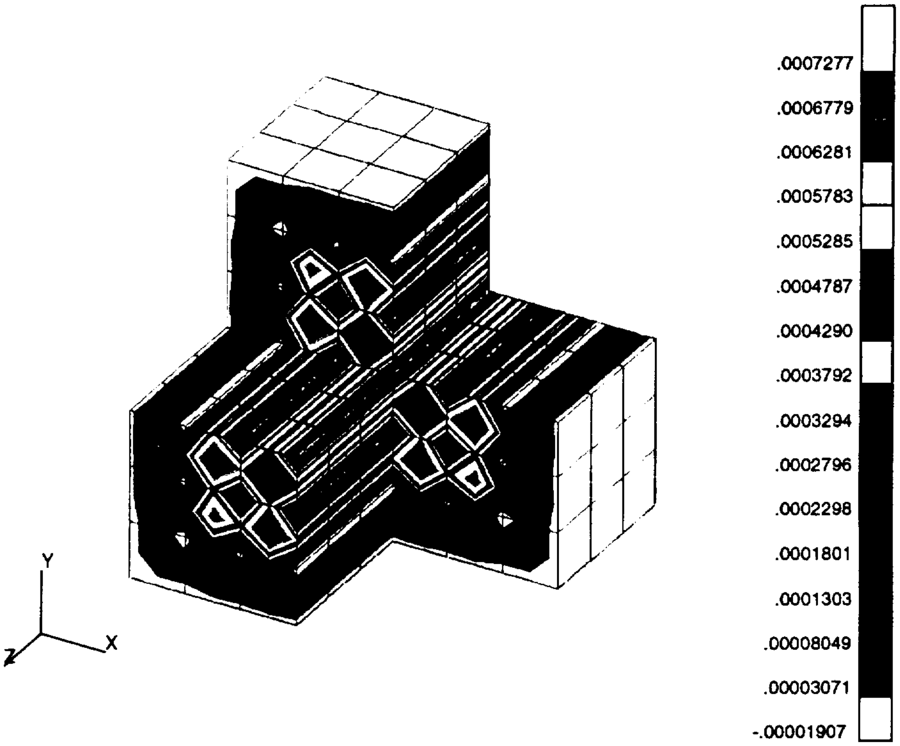

Figure A29: Mode 1,1,0 error for fluid/structure geometry (fluid only). 216 quadratic HEX20 elements, 1225 nodes.

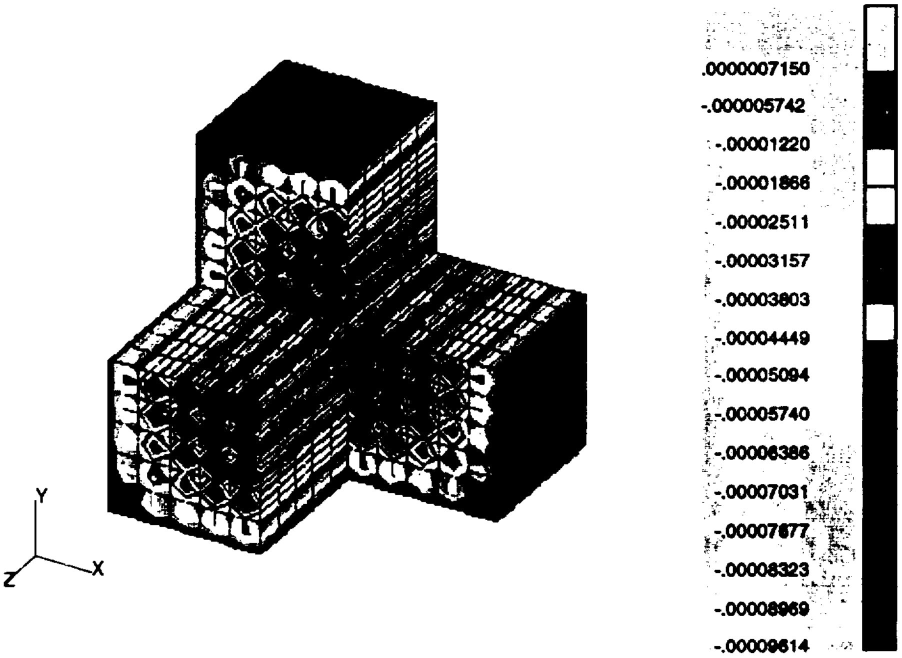

Figure A30: Mode 1,1,0 error for cubic fluid/structure geometry (fluid only). 1000 quadratic HEX20 elements, 4962 nodes.

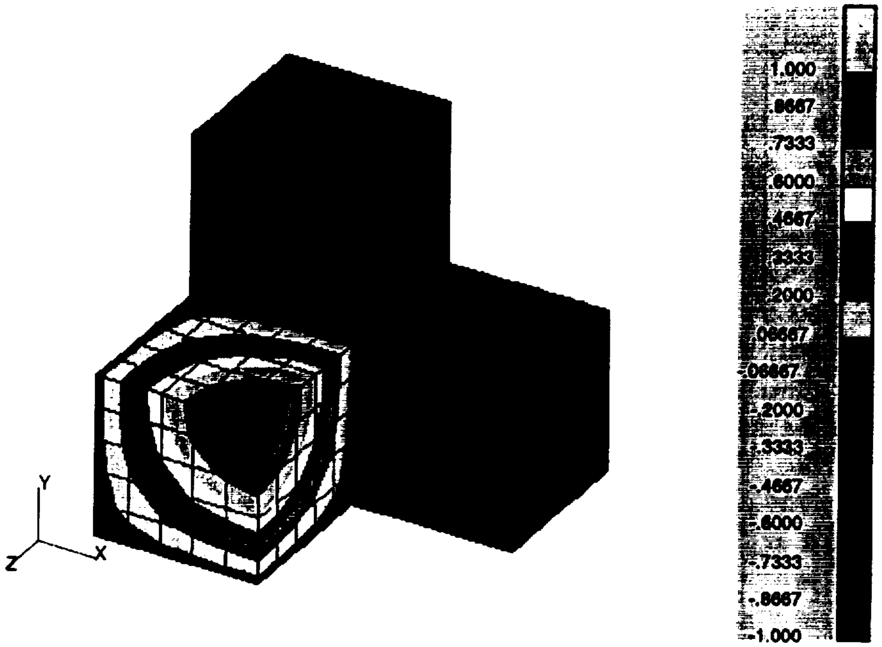

Figure A31: Mode 1,1,1 for fluid portion of fluid/structure cube. 1000 linear HEX8 elements, 1331 nodes.

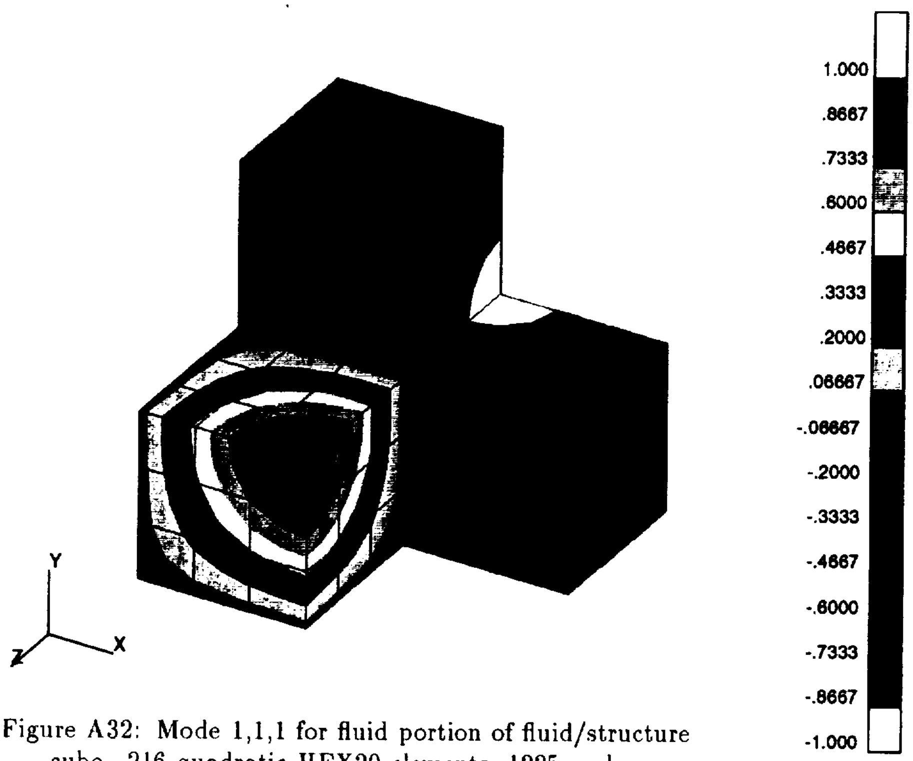

Figure A32: Mode 1,1,1 for fluid portion of fluid/structure cube. 216 quadratic HEX20 elements, 1225 nodes.

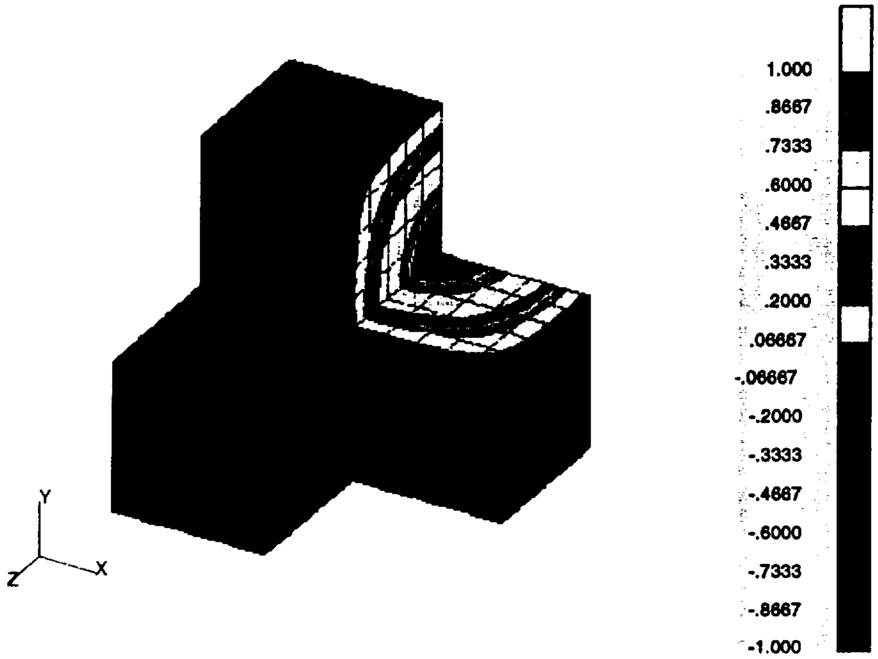

Figure A33: Mode 1,1,1 for fluid portion of fluid/structure cube. 1000 quadratic HEX20 elements, 4962 nodes.

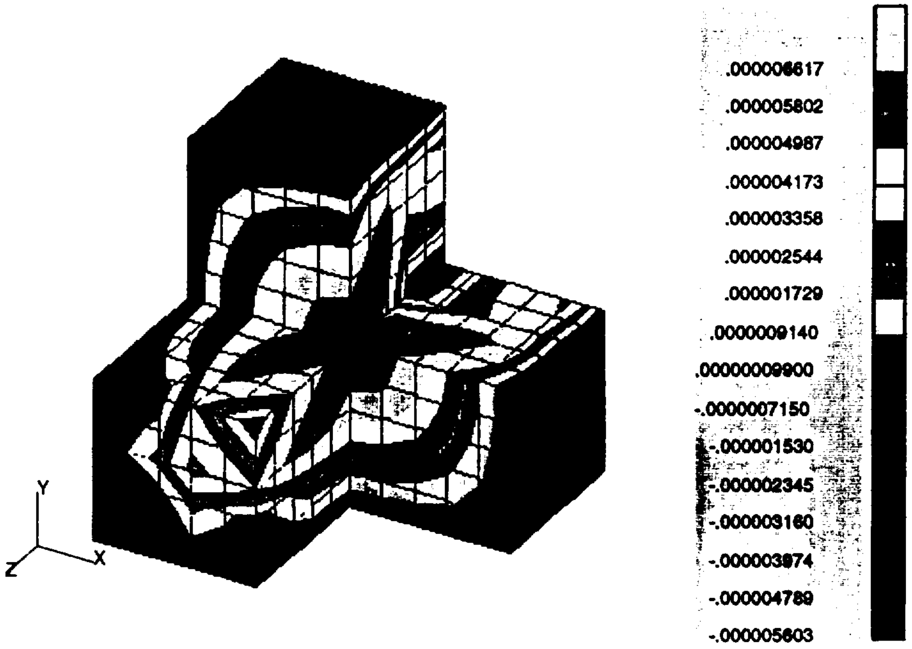

Figure A34: Mode 1,1,1 error for cubic fluid/structure geometry (fluid only). 1000 linear HEX8 elements, 1331 nodes.

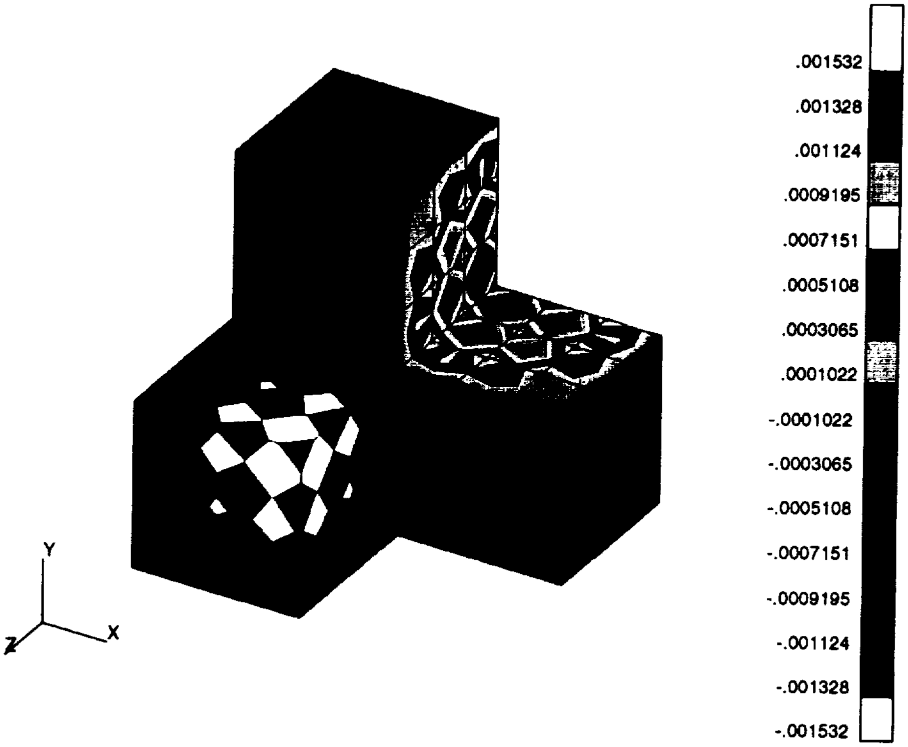

Figure A35: Mode 1,1,1 error for cubic fluid/structure geometry (fluid only). 216 quadratic HEX20 elements, 1225 nodes.

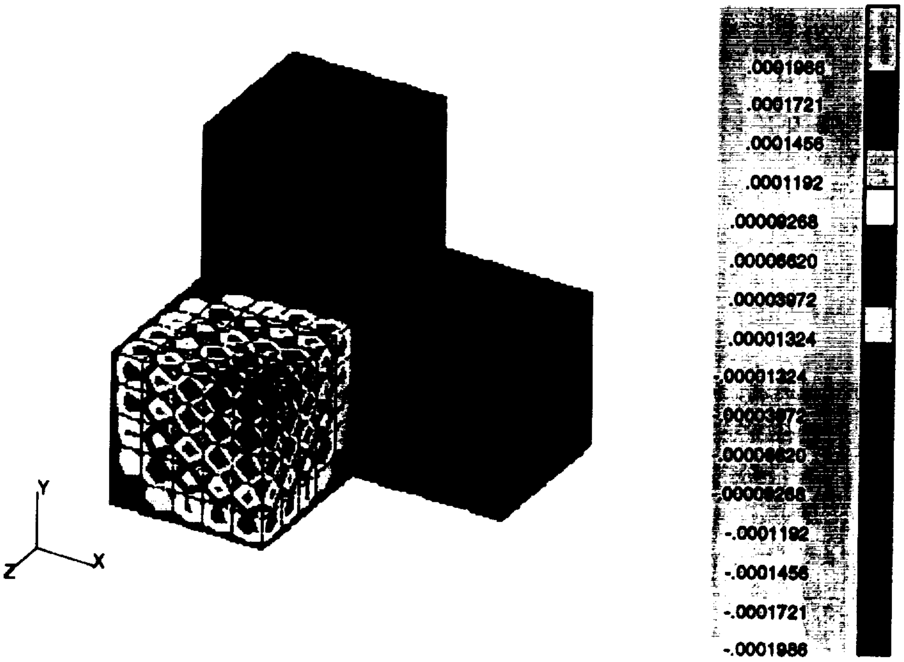

Figure A36: Mode 1,1,1 error for cubic fluid/structure geometry (fluid only). 1000 quadratic HEX20 elements, 4962 nodes.

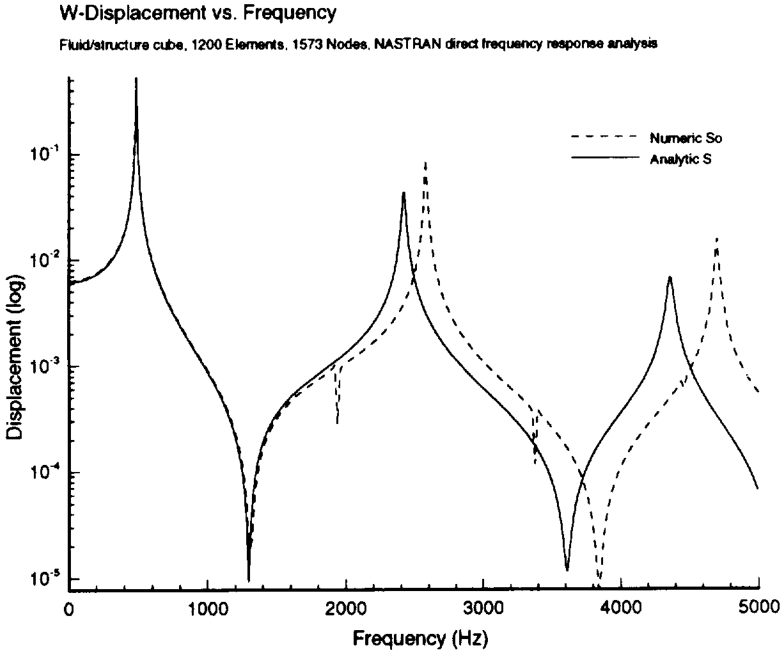

Figure A37: Displacement at the center of the z=5 plate on the fluid/structure cube. NASTRAN direct frequency response analysis. Analytic and numeric solutions shown.

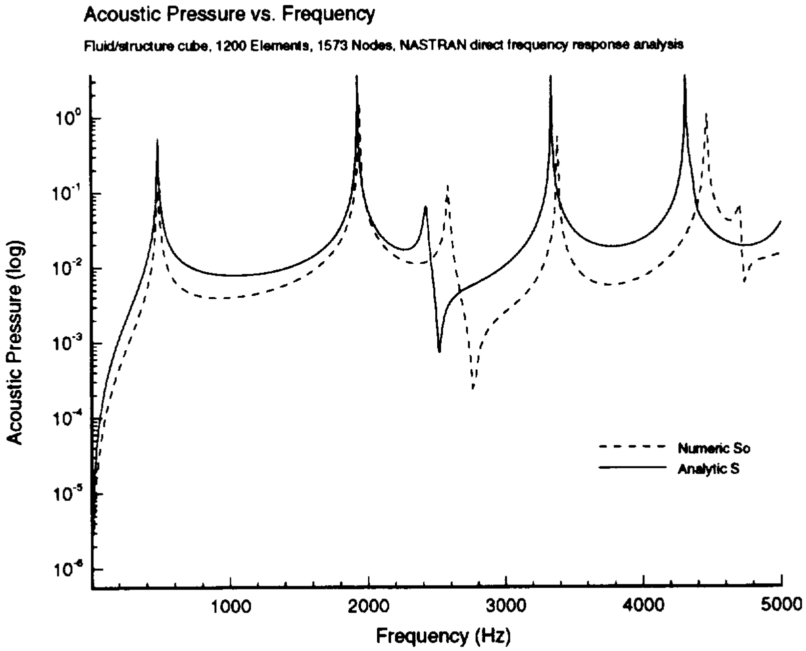

Figure A38: Acoustic pressure at the point (2.5,2.5,3.5) in the fluid/structure cube. NASTRAN direct frequency response analysis. Analytic and numeric solutions shown.

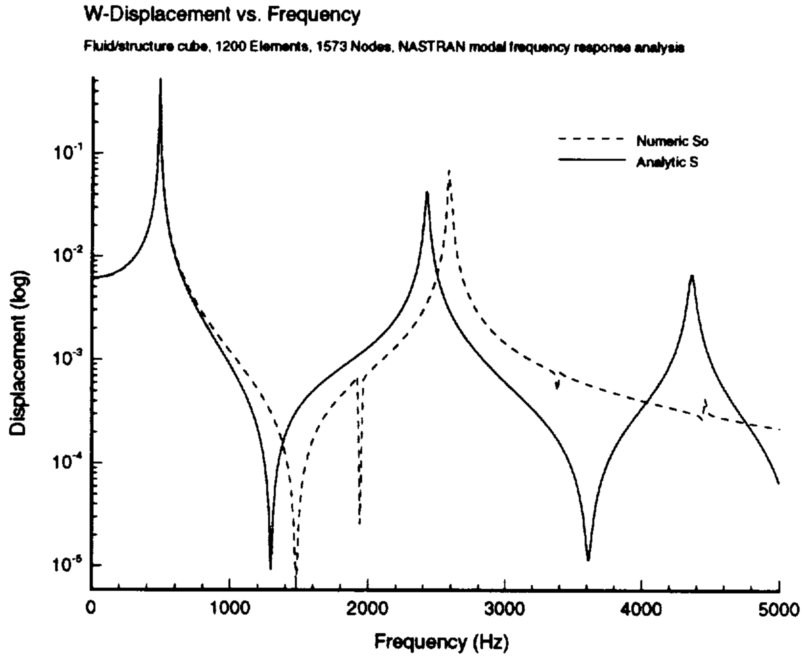

Figure A39: Displacement at the center of the z=5 plate on the fluid/structure cube. NASTRAN modal frequency response analysis. Analytic and numeric solutions shown.

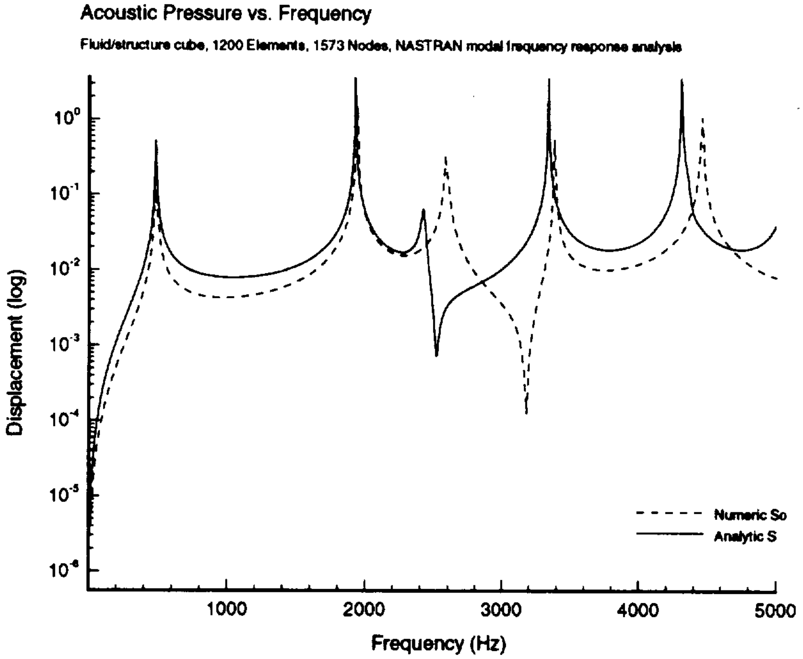

Figure A40: Acoustic pressure at the point (2.5,2.5,3.5) in the fluid/structure cube. NASTRAN modal frequency response analysis. Analytic and numeric solutions shown.

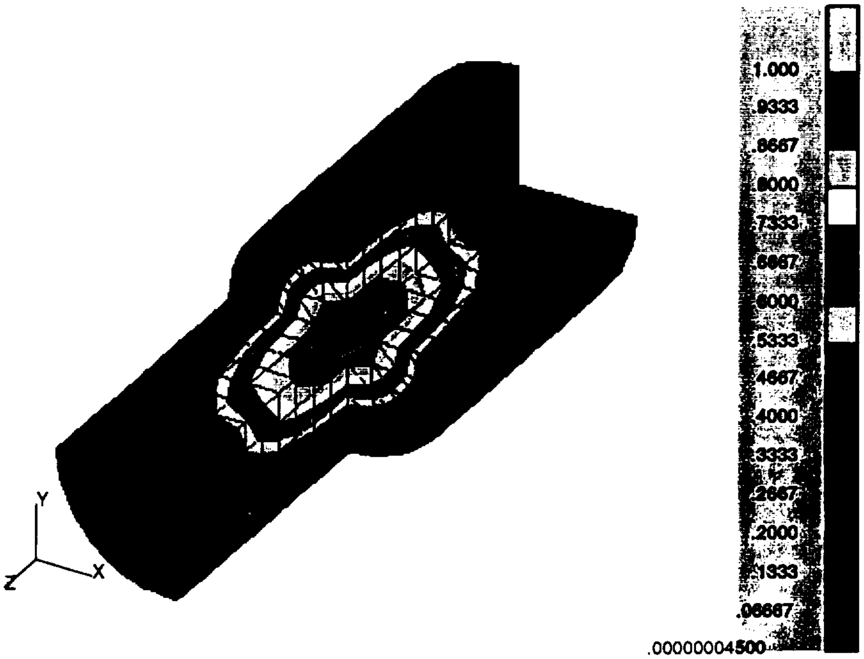

Figure A41: Fluid mode 1,0,1 for cylindrical geometry, 2240 linear WEDGE6 elements, 1449 nodes.

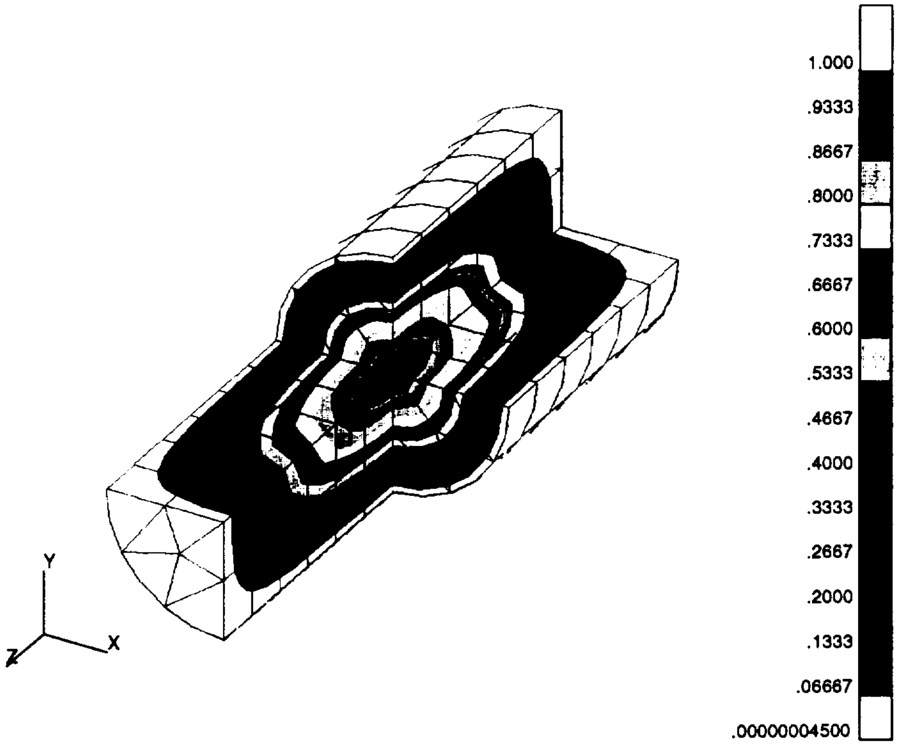

Figure A42: Fluid mode 1,0,1 for cylindrical geometry, 348 quadratic WEDGE15 elements, 1200 nodes.

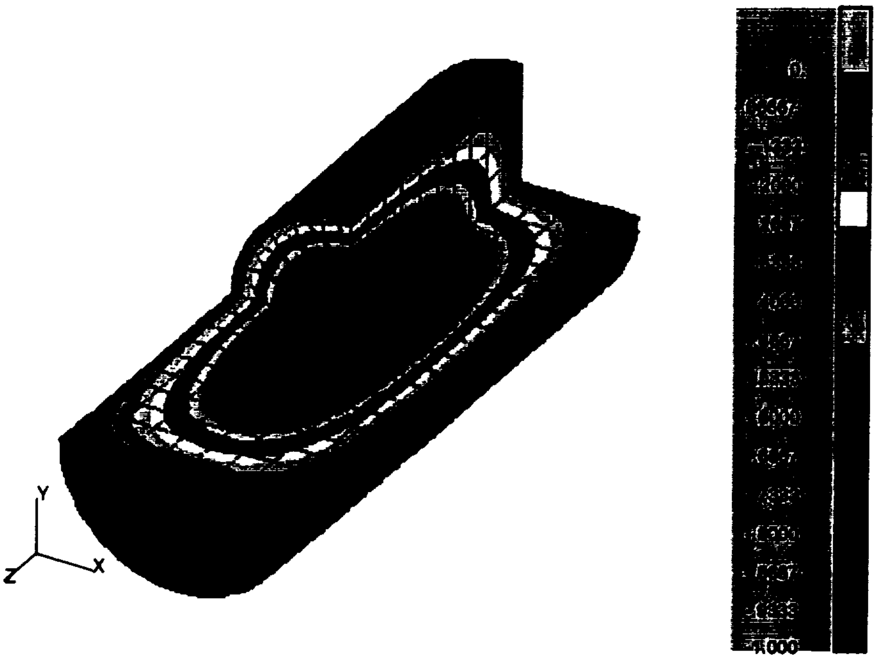

Figure A43: Fluid mode 1,0,1 for cylindrical geometry, 2240 quadratic WEDGE15 elements, 6609 nodes.

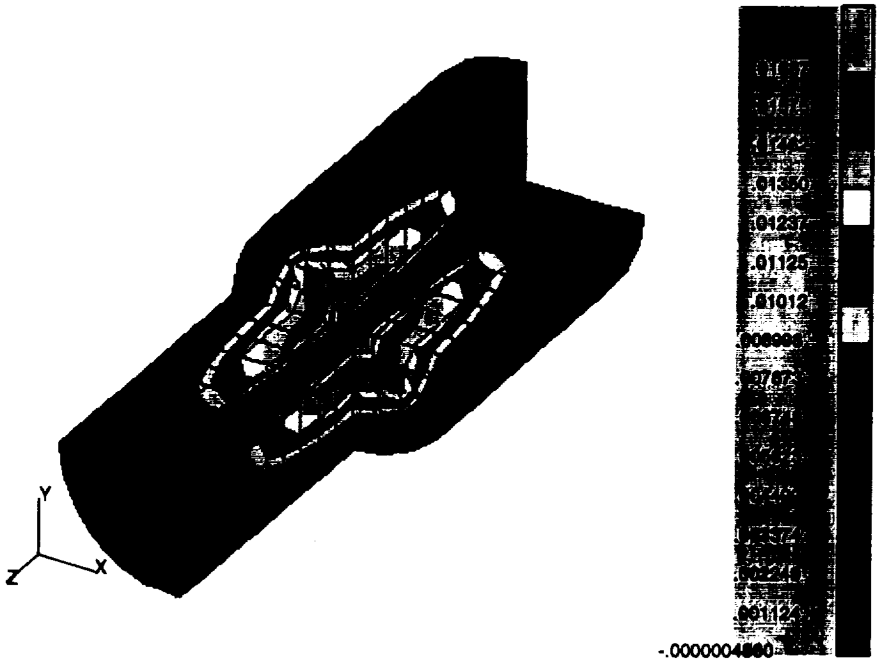

Figure A44: Fluid mode 1,0,1 error for cylindrical geometry, 2240 linear WEDGE6 elements, 1449 nodes.

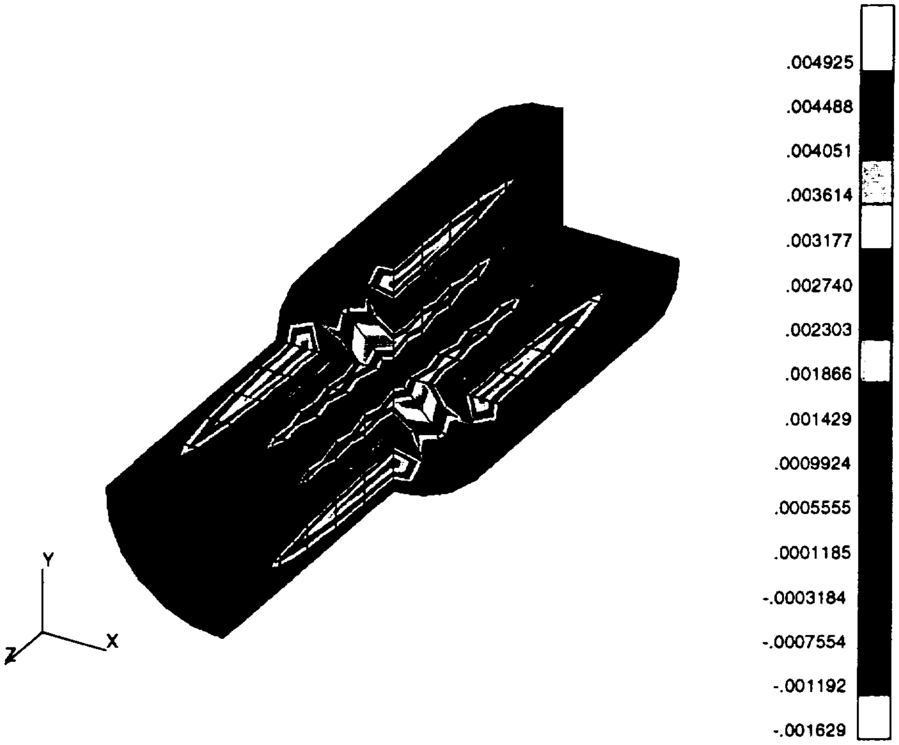

Figure A45: Fluid mode 1,0,1 error for cylindrical geometry, 348 quadratic WEDGE15 elements, 1200 nodes.

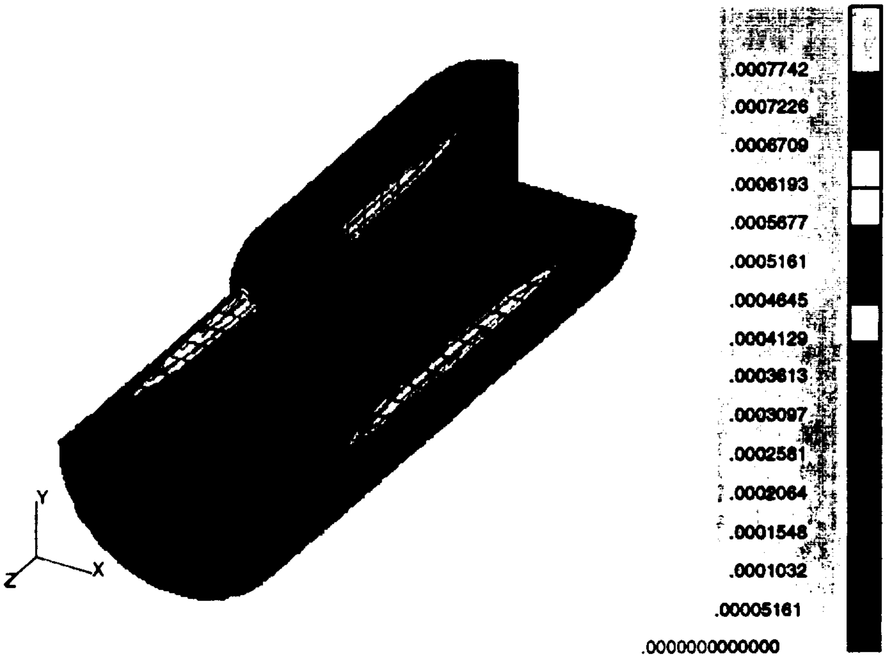

Figure A46: Fluid mode 1,0,1 error for cylindrical geometry, 2440 quadratic WEDGE15 elements, 6609 nodes.

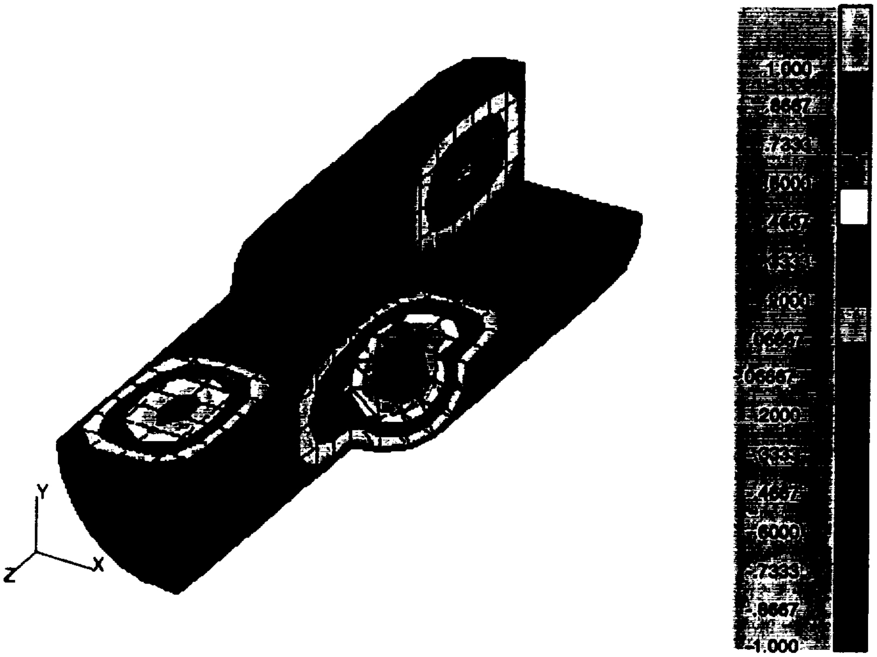

Figure A47: Fluid mode 1,1,3 for cylindrical geometry, 2240 linear WEDGE6 elements, 1449 nodes.

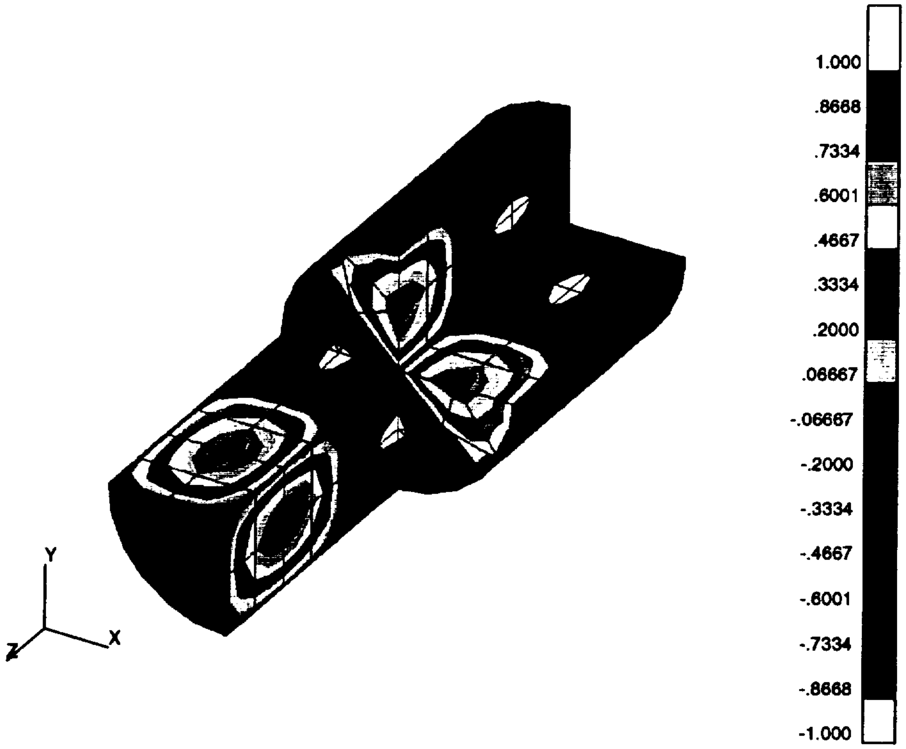

Figure A48: Fluid mode 1,1,3 for cylindrical geometry, 348 quadratic WEDGE15 elements, 1200 nodes.

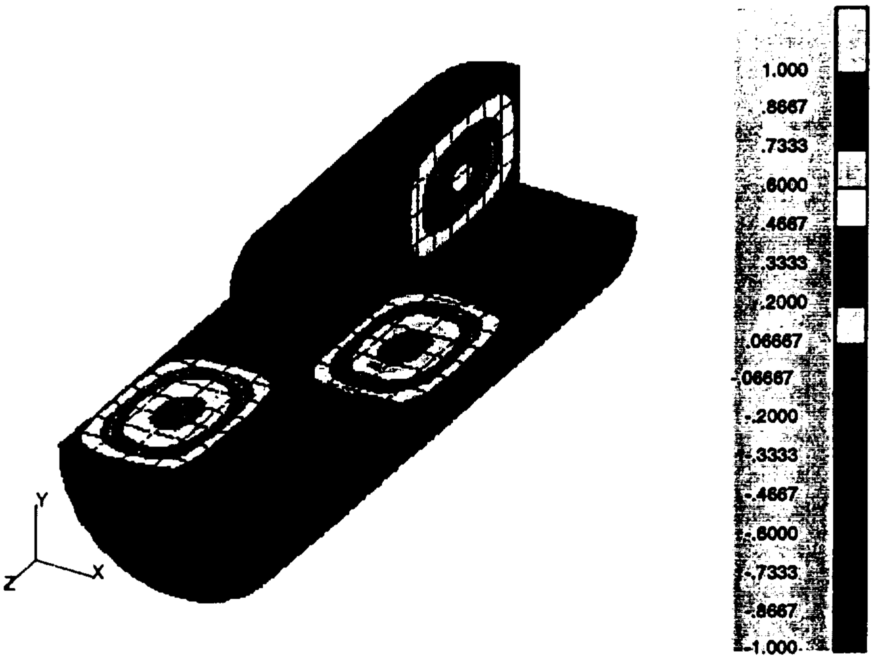

Figure A49: Fluid mode 1,1,3 for cylindrical geometry, 2240 quadratic WEDGE15 elements, 6609 nodes.

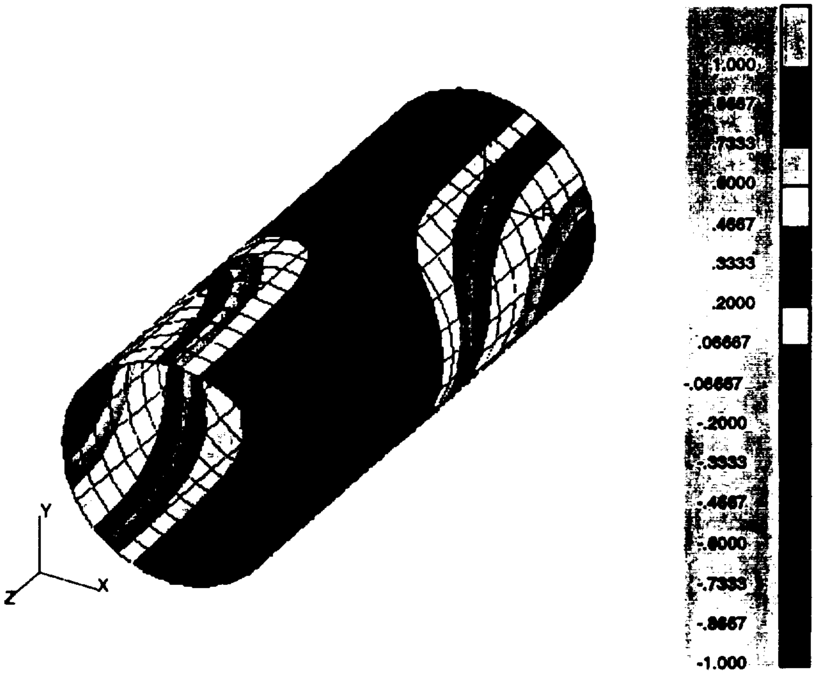

Figure A50: U-displacement (axial) of cylindrical shell, mode 1,1. 480 linear QUAD4 elements, 504 nodes.

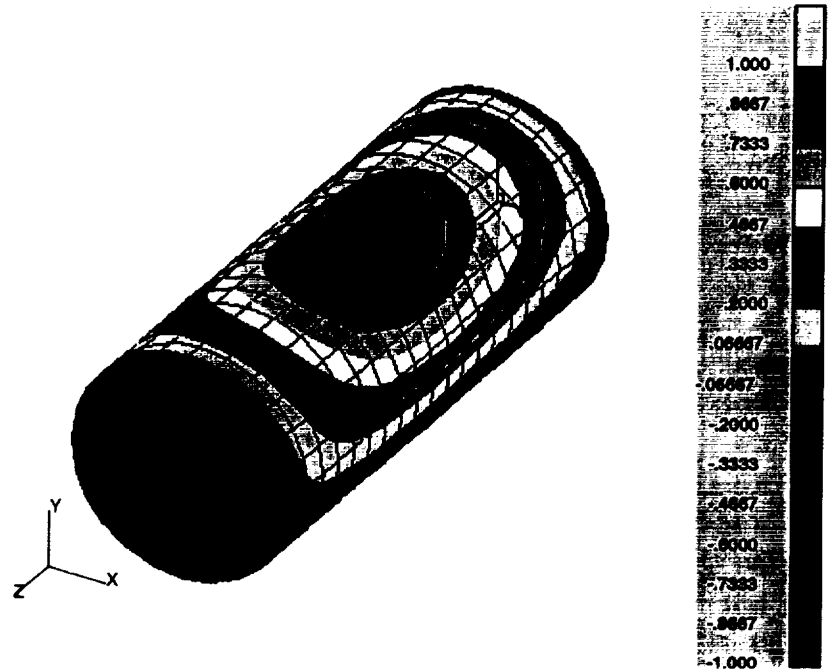

Figure A51: V-displacement (circumferential) of cylindrical shell, mode 1,1. 480 linear QUAD4 elements, 504 nodes.

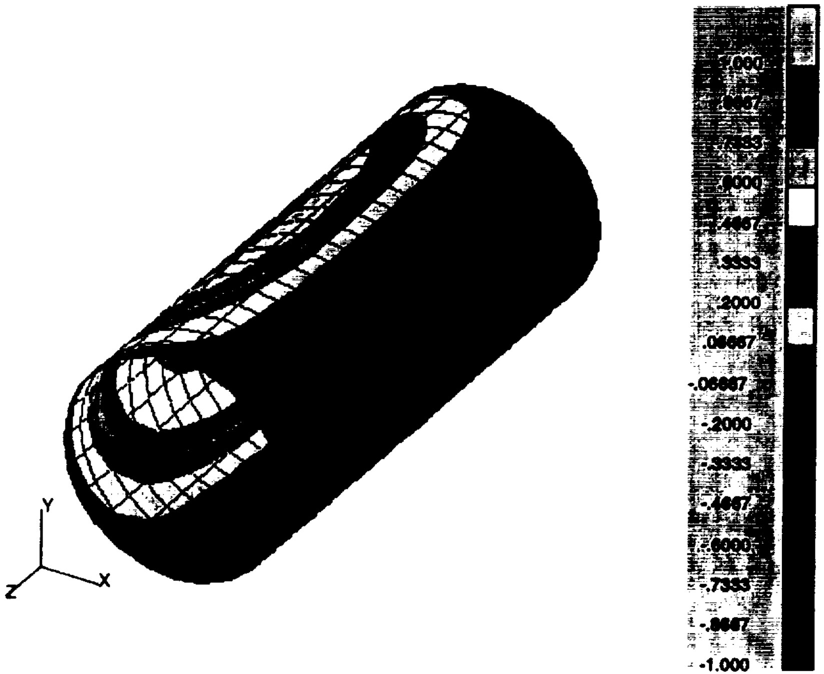

Figure A52: W-displacement (radial) of cylindrical shell, mode 1,1. 480 linear QUAD4 elements, 504 nodes.

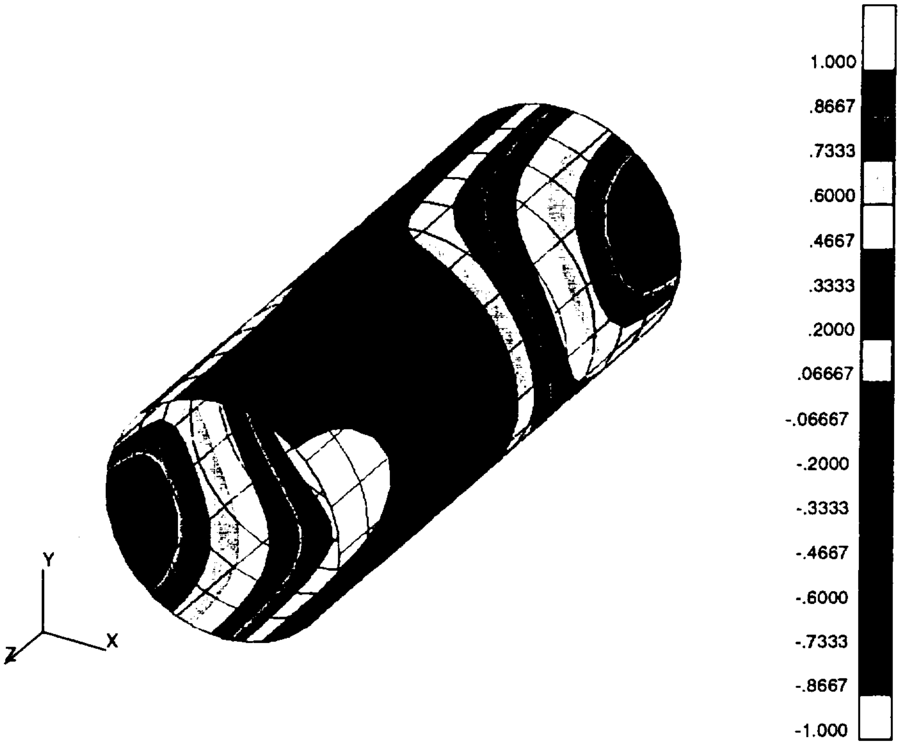

Figure A53: U-displacement (axial) of cylindrical shell, mode 1,1. 192 quadratic QUAD8 elements, 608 nodes.

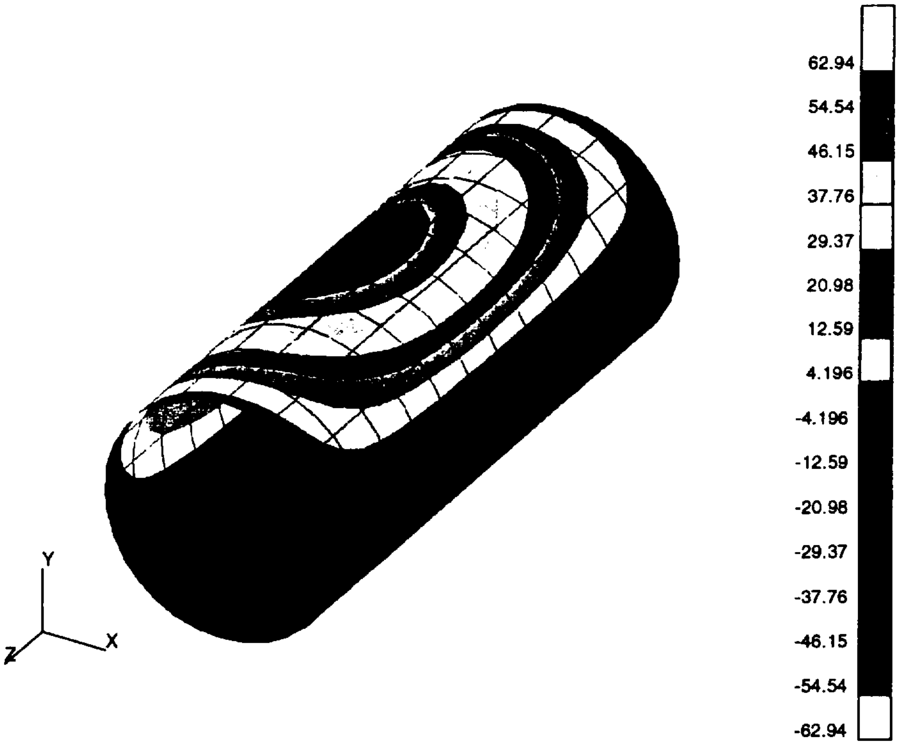

Figure A54: V-displacement (circumferential) of cylindrical shell, mode 1,1. 192 quadratic QUAD8 elements, 608 nodes.

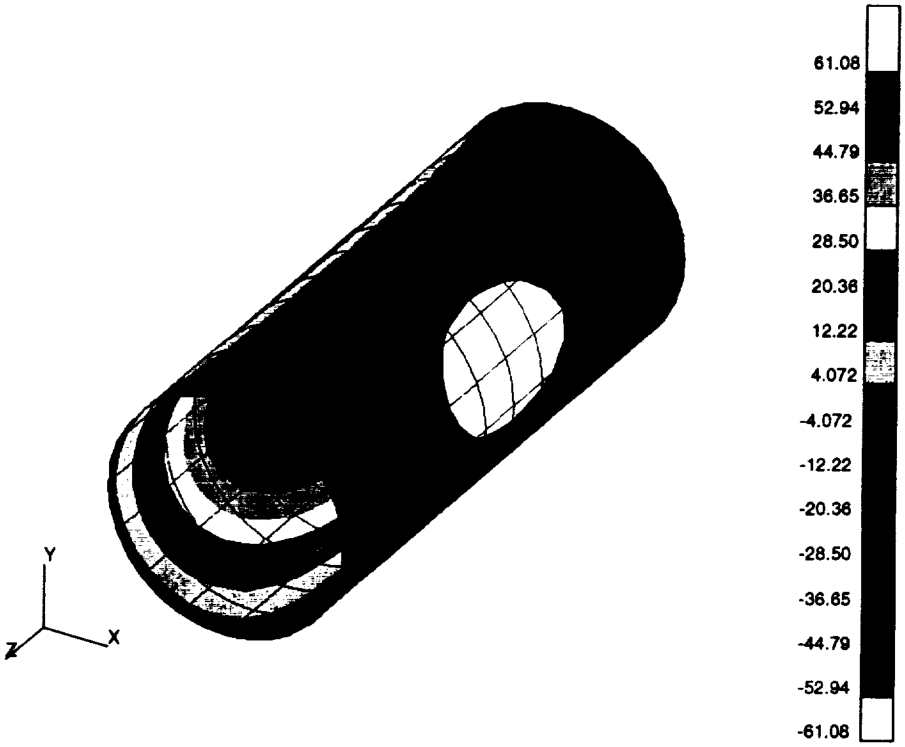

Figure A55: W-displacement (radial) of cylindrical shell, mode 1,1. 192 quadratic QUAD8 elements, 608 nodes.

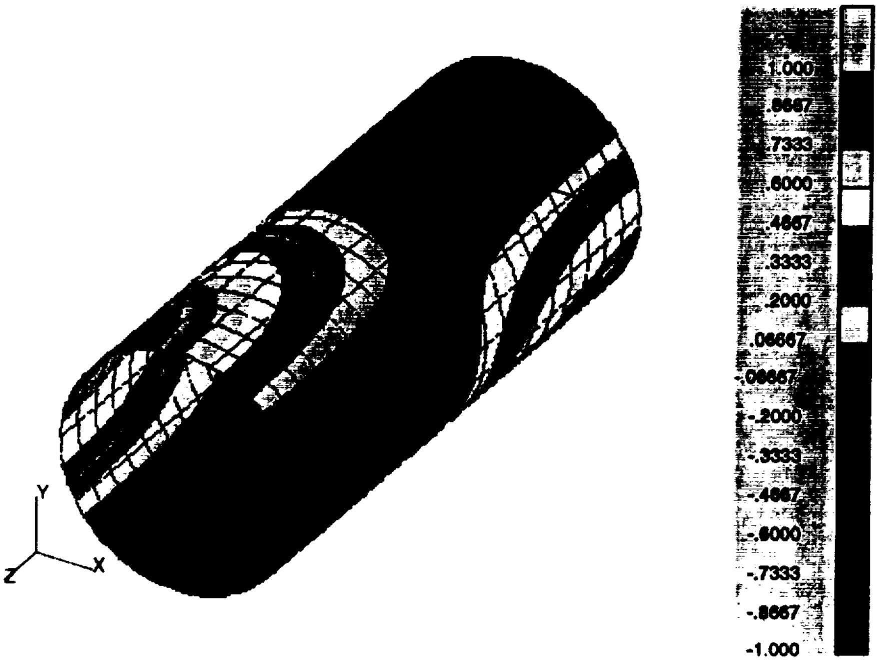

Figure A56: U-displacement (axial) of cylindrical shell, mode 1,1. 480 quadratic QUAD8 elements, 1488 nodes.

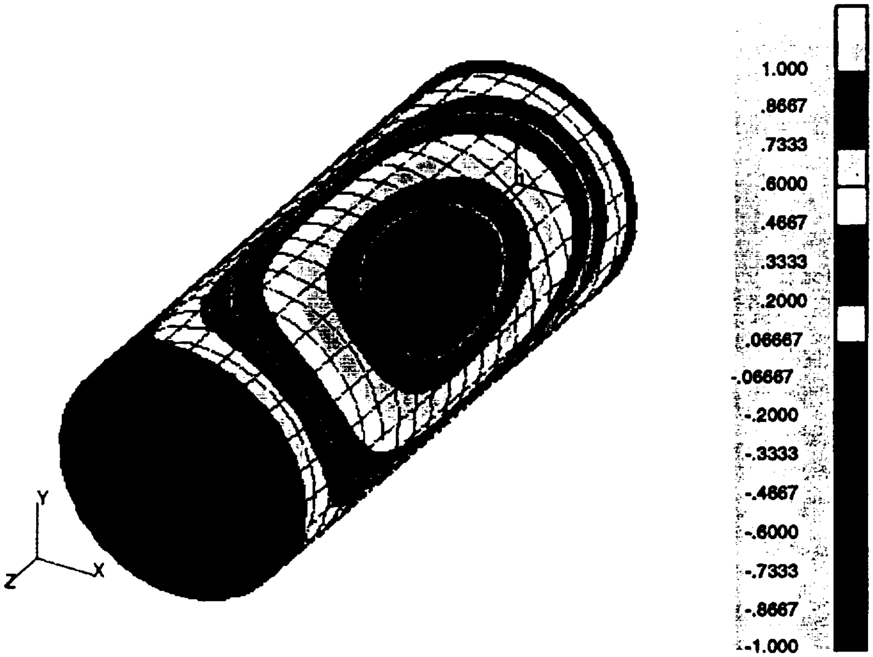

Figure A57: V-displacement (circumferential) of cylindrical shell, mode 1,1. 480 quadratic QUAD8 elements, 1488 nodes.

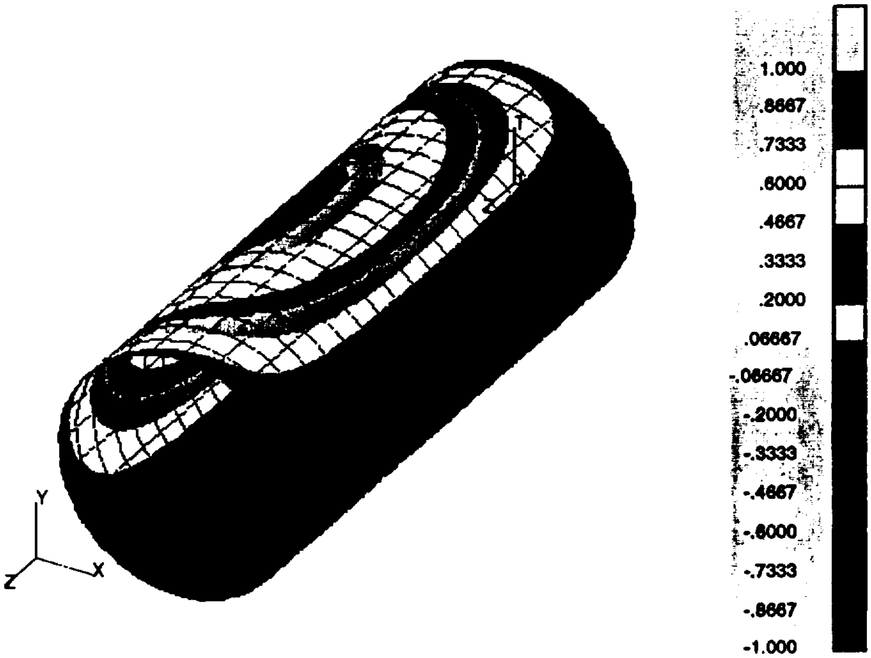

Figure A58: W-displacement (radial) of cylindrical shell, mode 1,1. 480 quadratic QUAD8 elements, 1488 nodes.

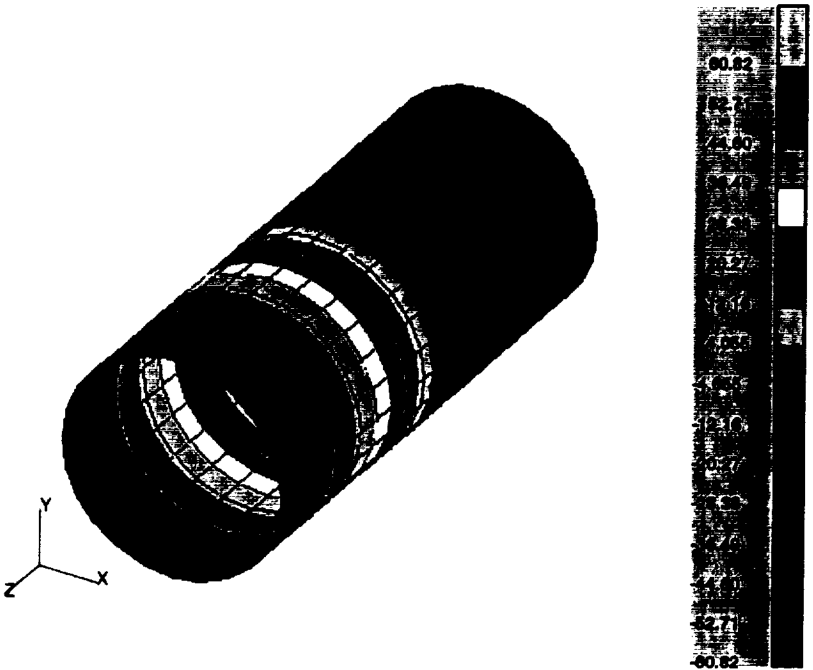

Figure A59: U-displacement (axial) of cylindrical shell, mode 1,0 (breathing mode). 480 linear QUAD4 elements, 504 nodes

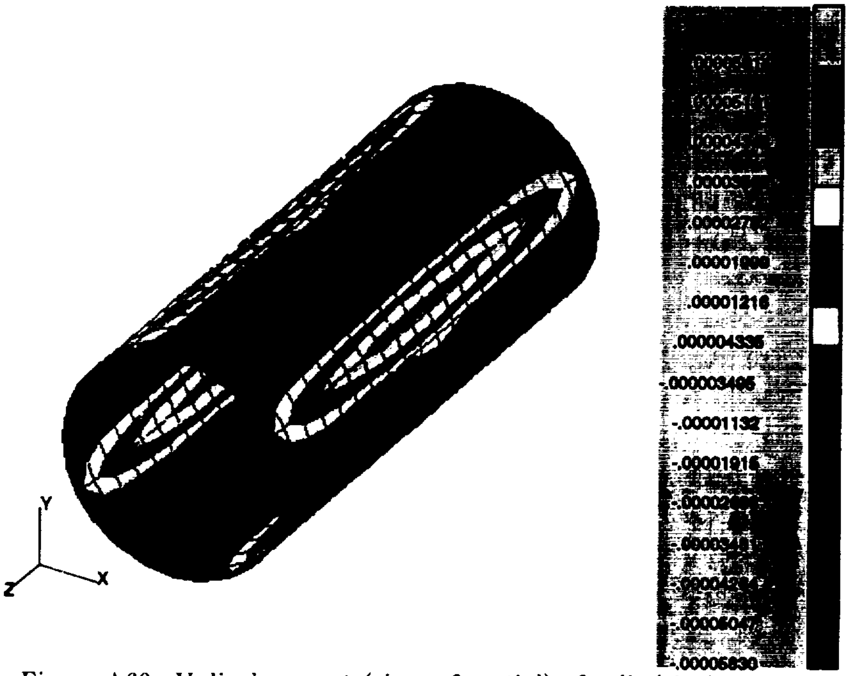

Figure A60: V-displacement (circumferential) of cylindrical shell, mode 1,0 (breathing mode). 480 linear QUAD4 elements, 504 nodes. Eigenvector not normalized to one.

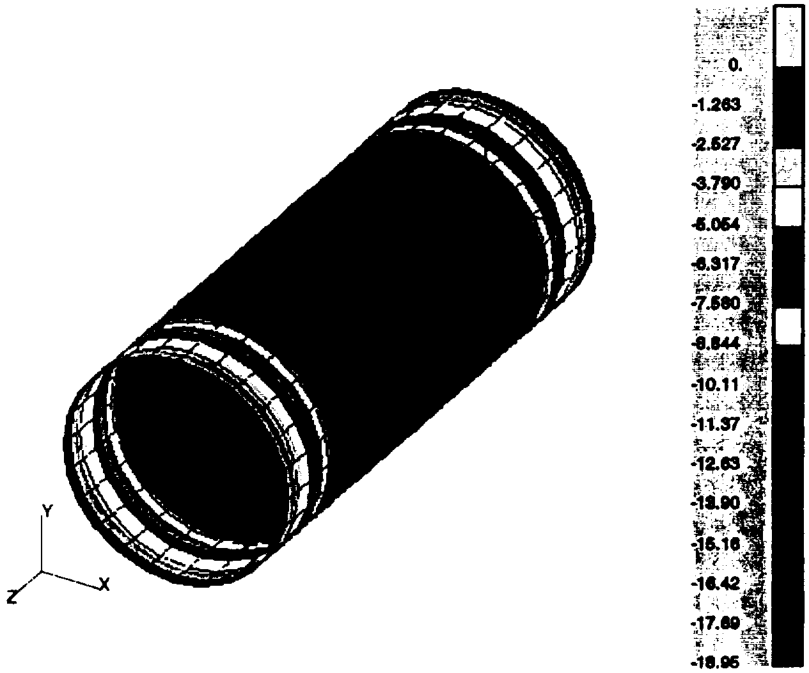

Figure A61: W-displacement (radial) of cylindrical shell, mode 1,0 (breathing mode). 480 linear QUAD4 elements, 504 nodes.

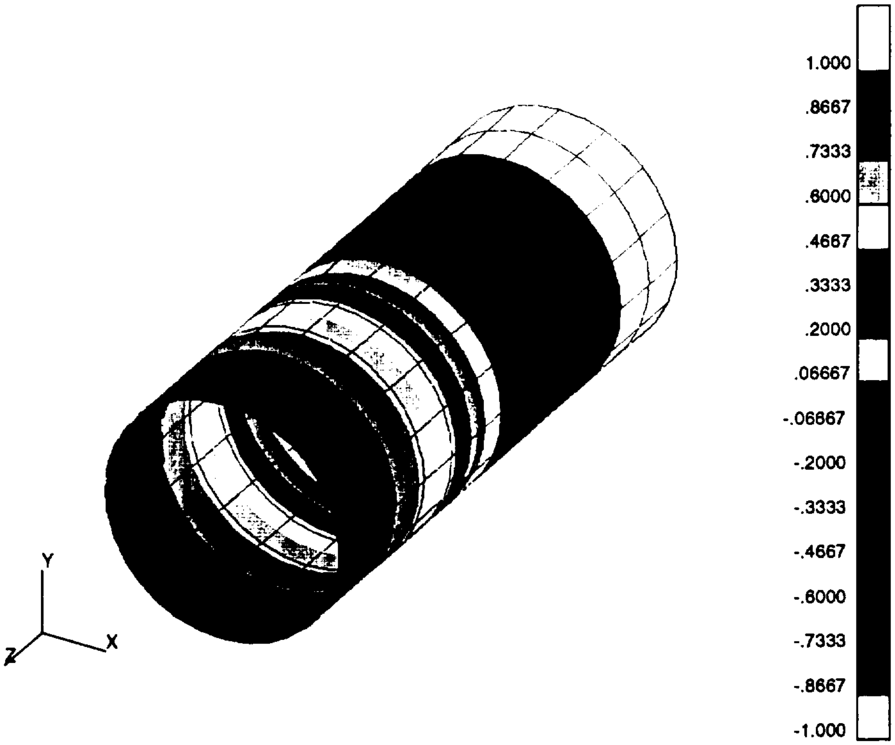

Figure A62: U-displacement (axial) of cylindrical shell, mode 1,0 (breathing mode). 192 quadratic QUAD8 elements, 608 nodes.

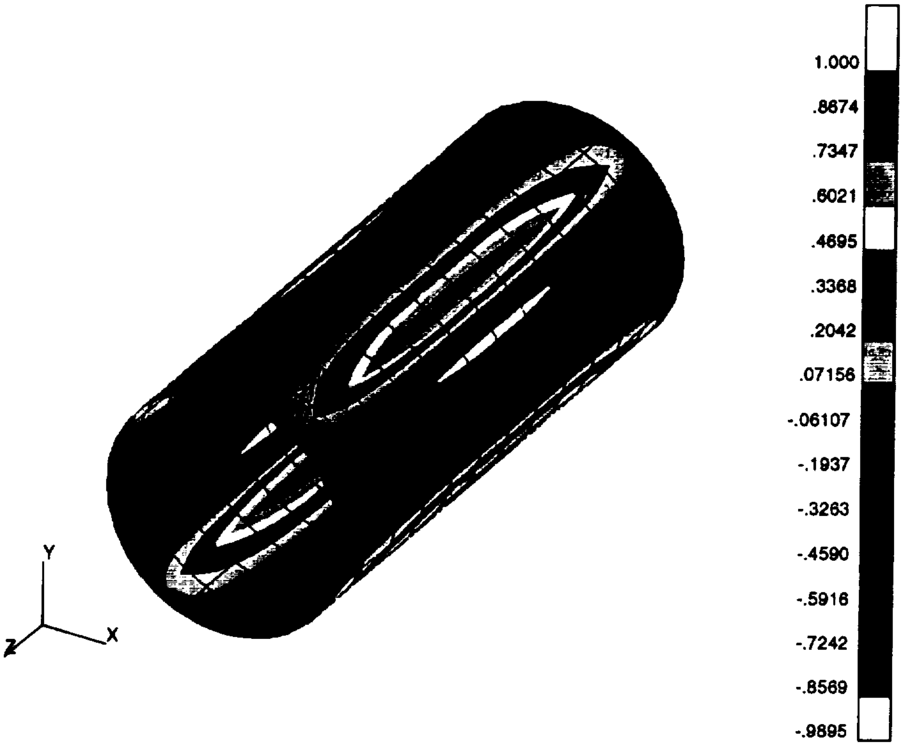

Figure A63: V-displacement (circumferential) of cylindrical shell, mode 1,0 (breathing mode). 192 quadratic QUAD8 elements, 608 nodes.

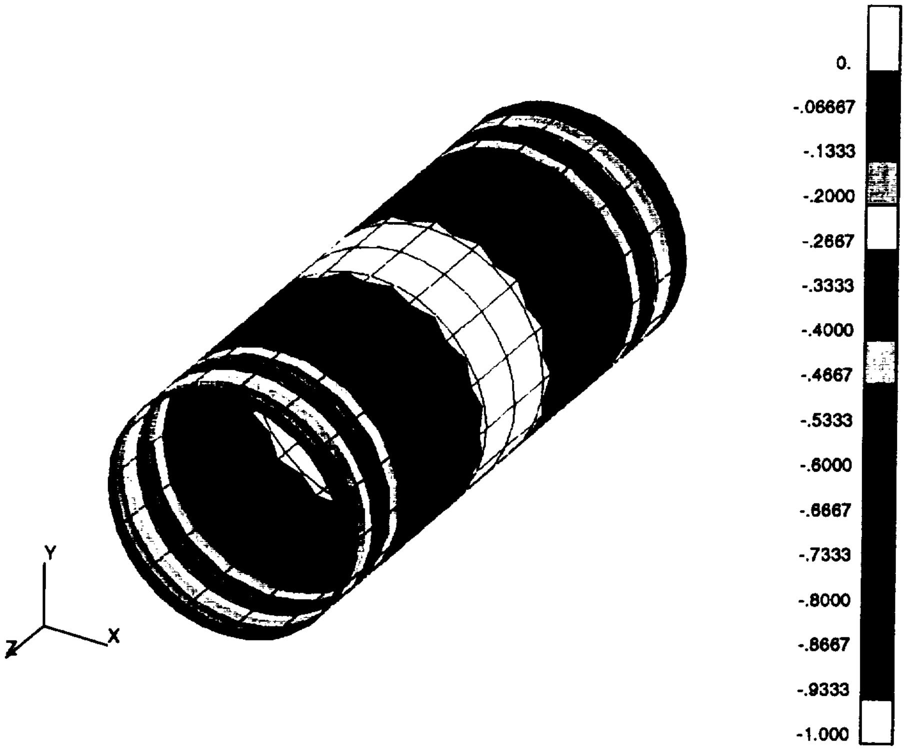

Figure A64: W-displacement (axial) of cylindrical shell, mode 1,0 (breathing mode). 192 quadratic QUAD8 elements, 608 nodes.

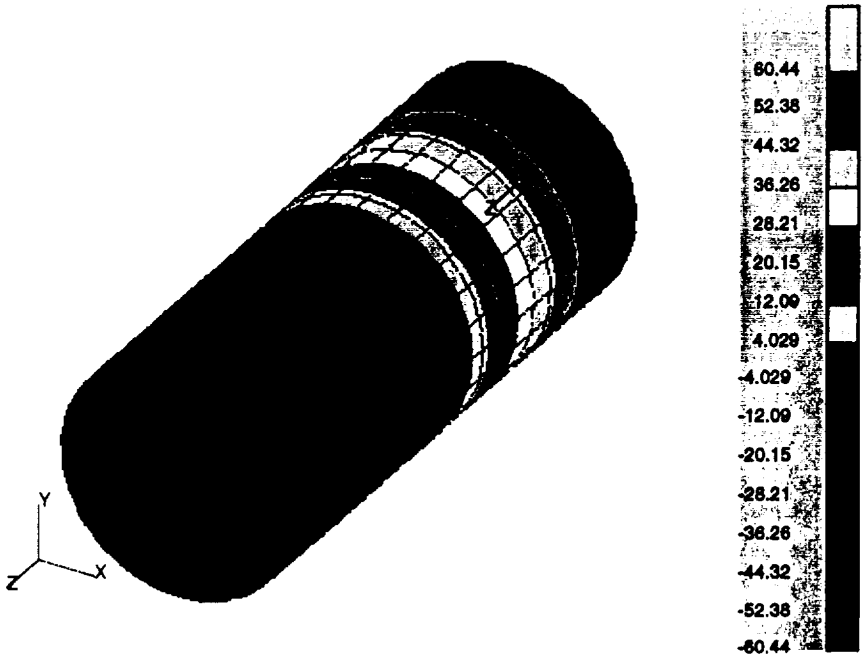

Figure A65: U-displacement (axial) of cylindrical shell, mode 1,0 (breathing mode). 480 quadratic QUAD8 elements, 1488 nodes.

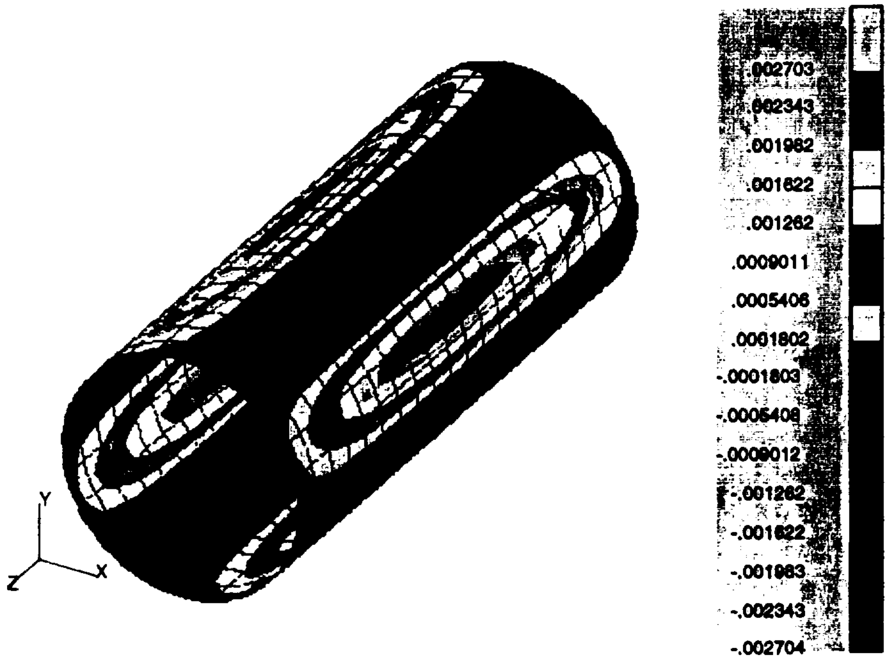

Figure A66: V-displacement (circumferential) of cylindrical shell, mode 1,0 (breathing mode). 480 quadratic QUAD8 elements, 1488 nodes. Eigenvector not normalized to one.

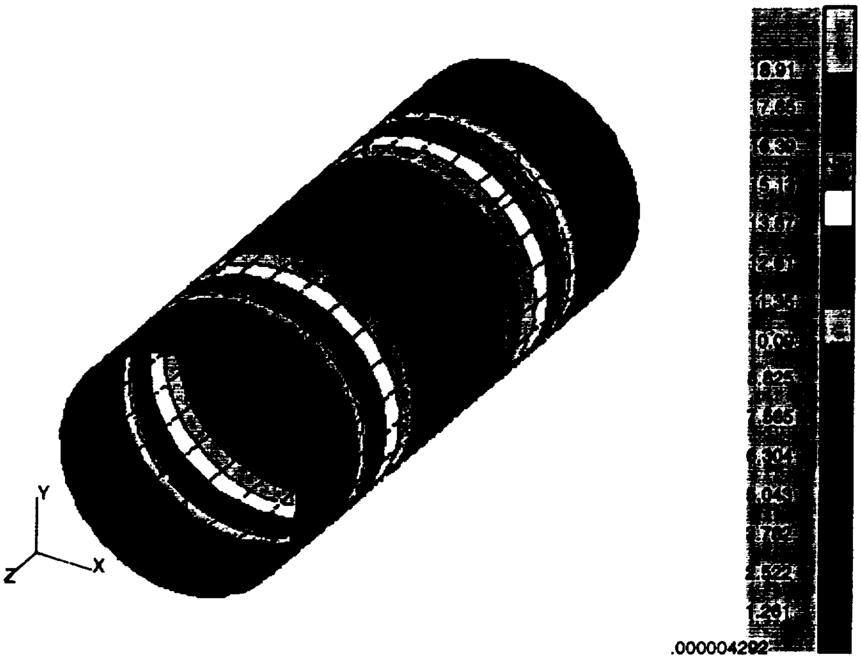

Figure A67: W-displacement (radial) of cylindrical shell, mode 1,0 (breathing mode). 480 quadratic QUAD8 elements, 1488 nodes.

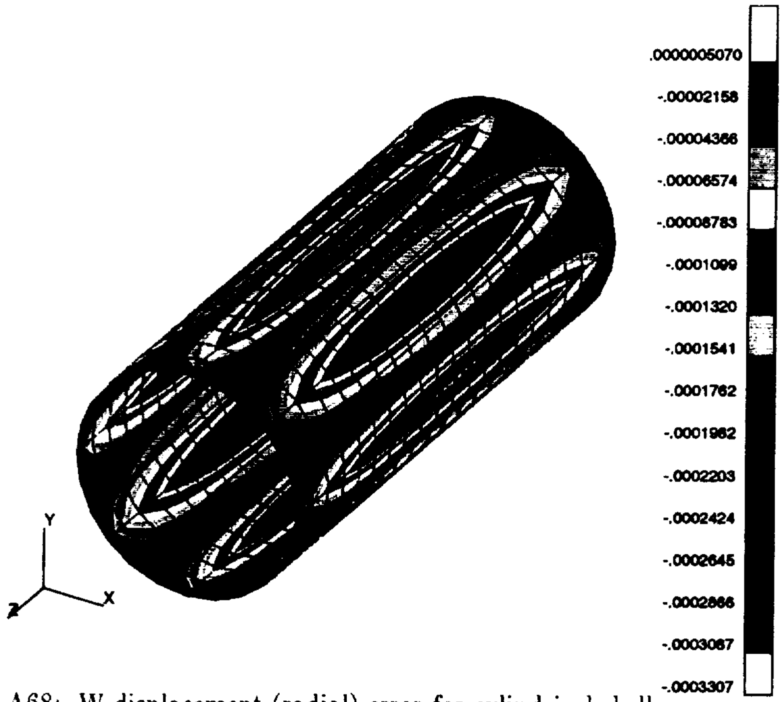

Figure A68: W-displacement (radial) error for cylindrical shell, mode 1,0 (breathing mode). 480 linear QUAD4 elements, 504 nodes.

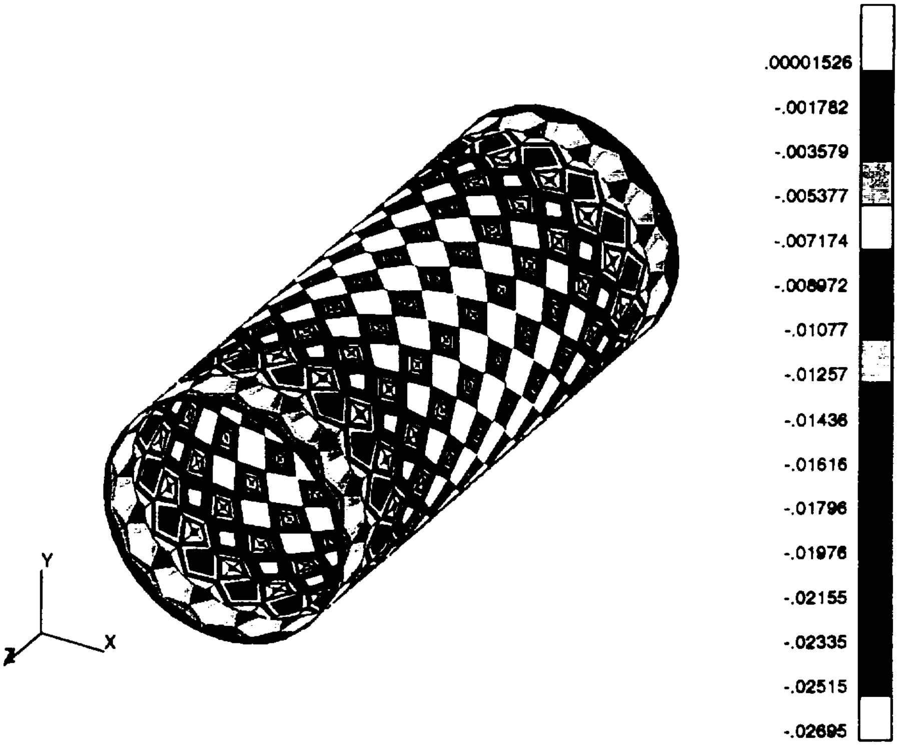

Figure A69: W-displacement (radial) error for cylindrical shell, mode 1,0 (breathing mode). 192 quadratic QUAD8 elements, 608 nodes.

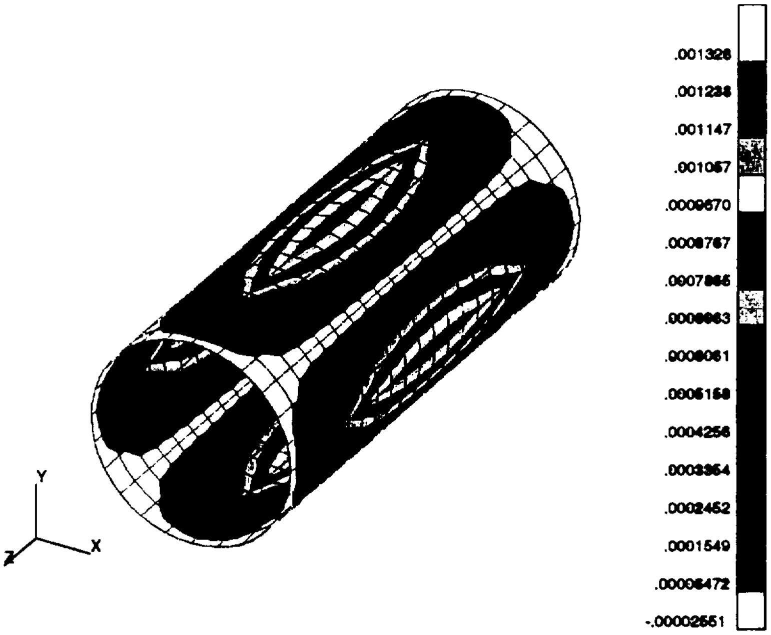

Figure A70: W-displacement (radial) error for cylindrical shell, mode 1,0 (breathing mode). 480 quadratic QUAD8 elements, 1488 nodes.

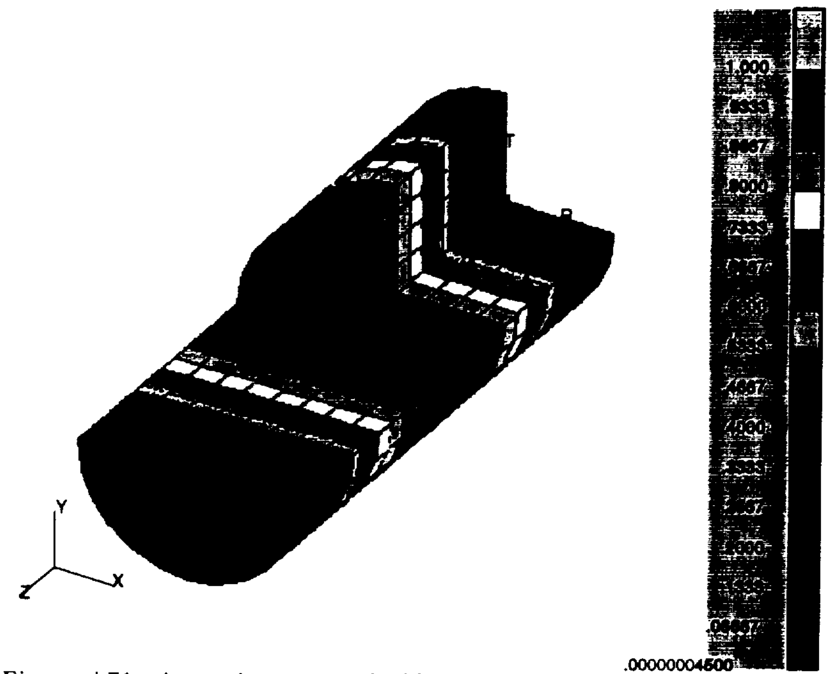

Figure A71: Acoustic pressure inside cylindrical shell, fluid mode 1,0,1. 2241 linear WEDGE6 elements, 1449 nodes.

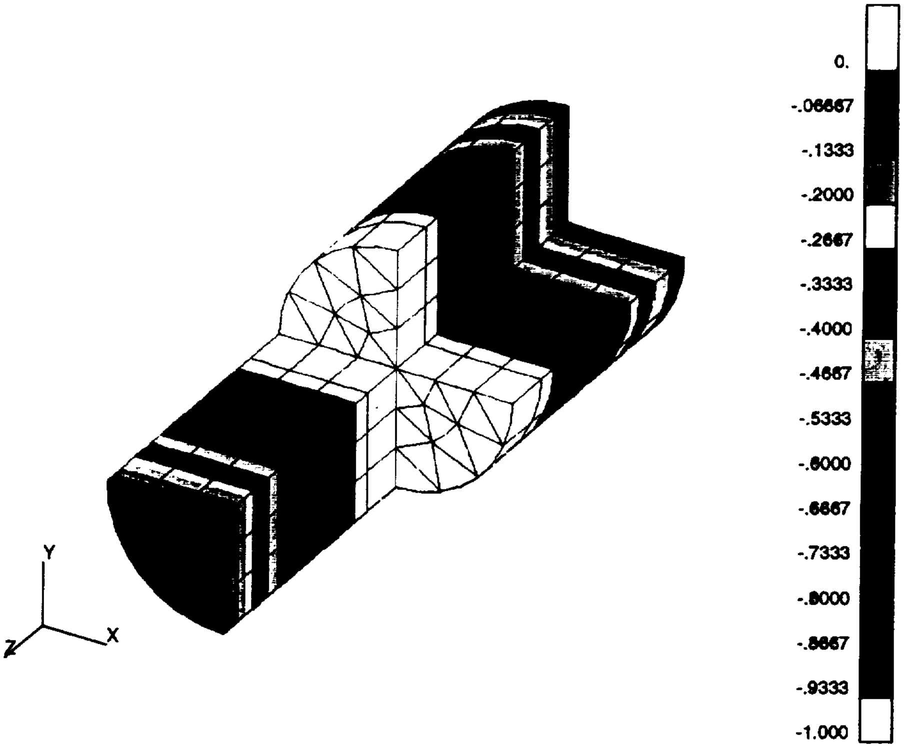

Figure A72: Acoustic pressure inside cylindrical shell, fluid mode 1,0,1. 672 quadratic WEDGE15 elements, 2121 nodes.

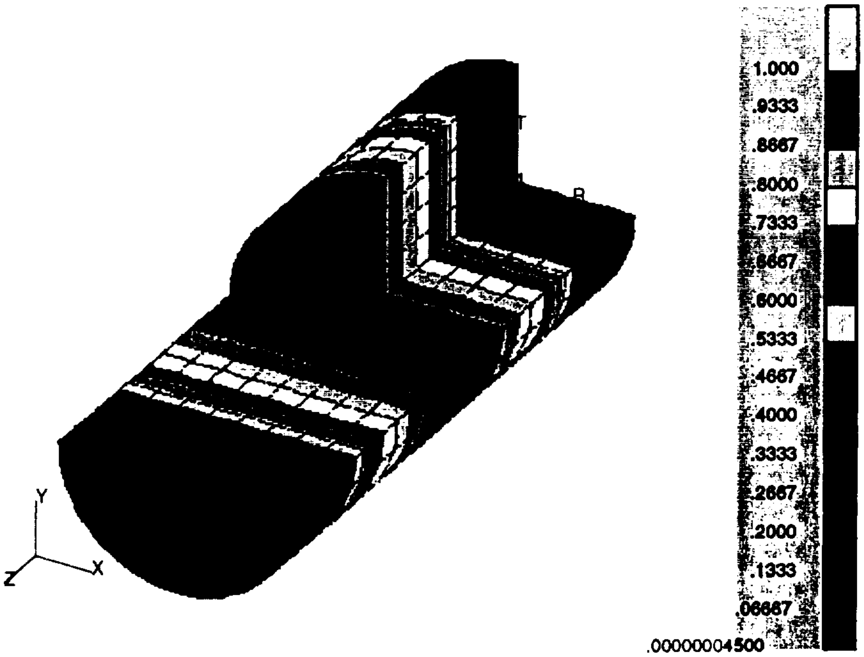

Figure A73: Acoustic pressure inside cylindrical shell, mode 1,0,1. 2241 quadratic WEDGE15 elements, 6609 nodes.

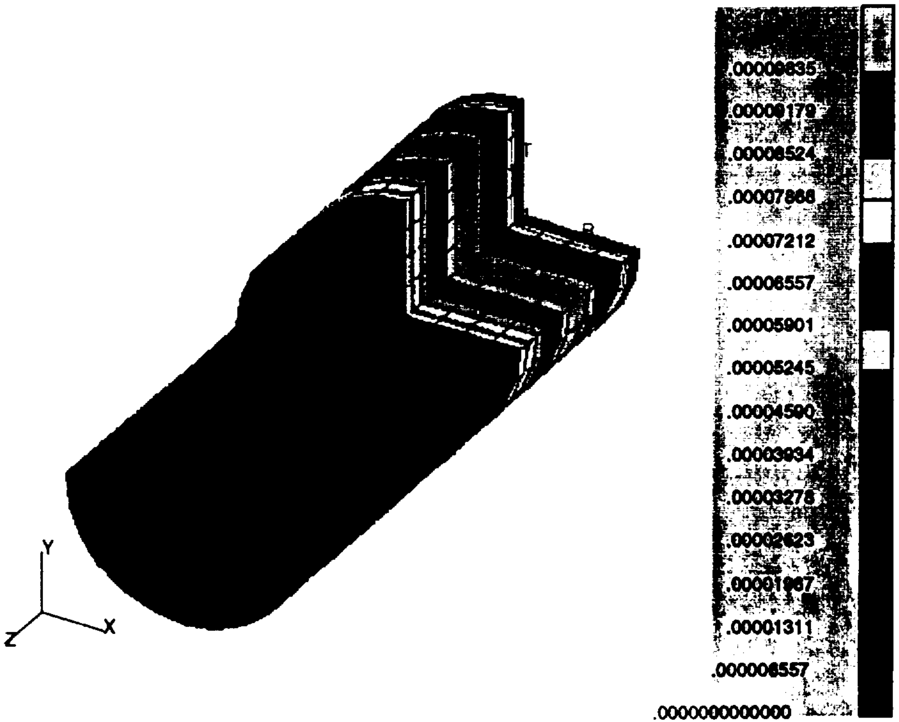

Figure A74: Mode 1,0,1 error for cylindrical fluid/structure geometry (fluid only). 2241 linear WEDGE6 elements 1449 nodes.

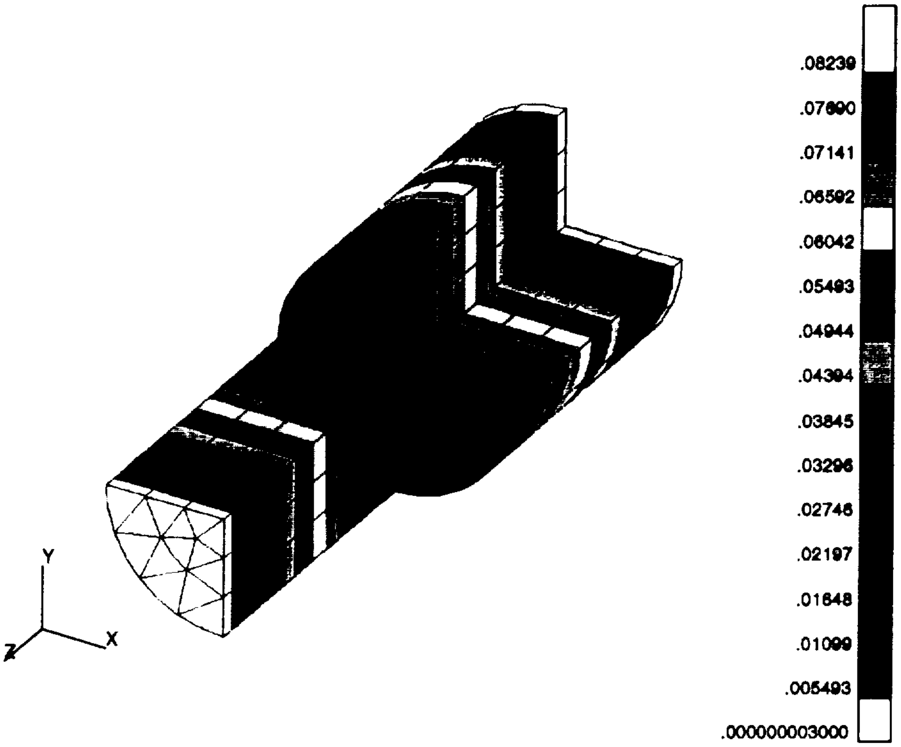

Figure A75: Mode 1,0,1 error for cylindrical fluid/structure geometry (fluid only). 672 quadratic WEDGE15 elements, 2121 nodes.

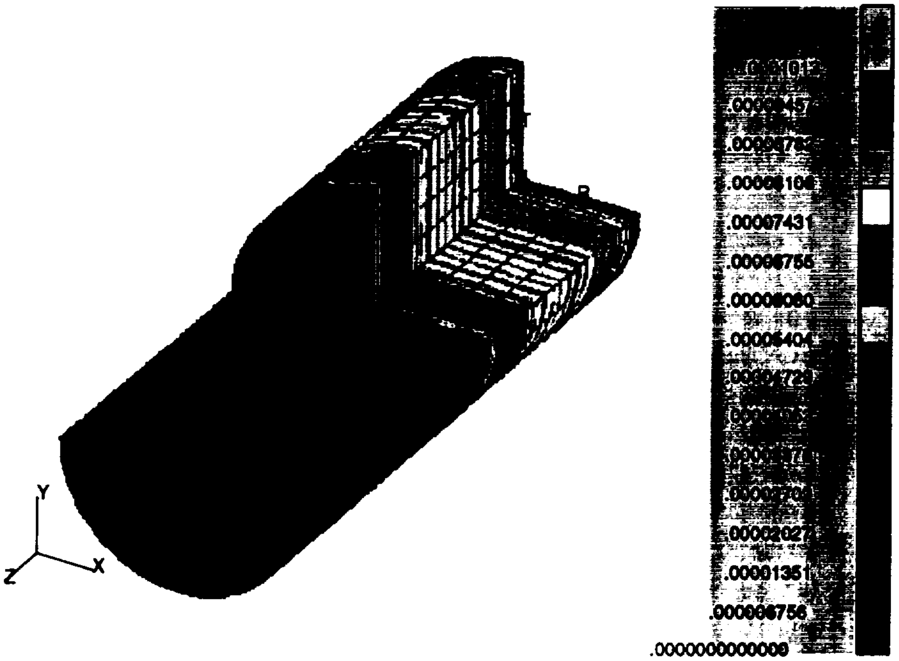

Figure A76: Mode 1,0,1 error for cylindrical fluid/structure geometry (fluid portion only). 2241 quadratic WEDGE15 elements, 6609 nodes.

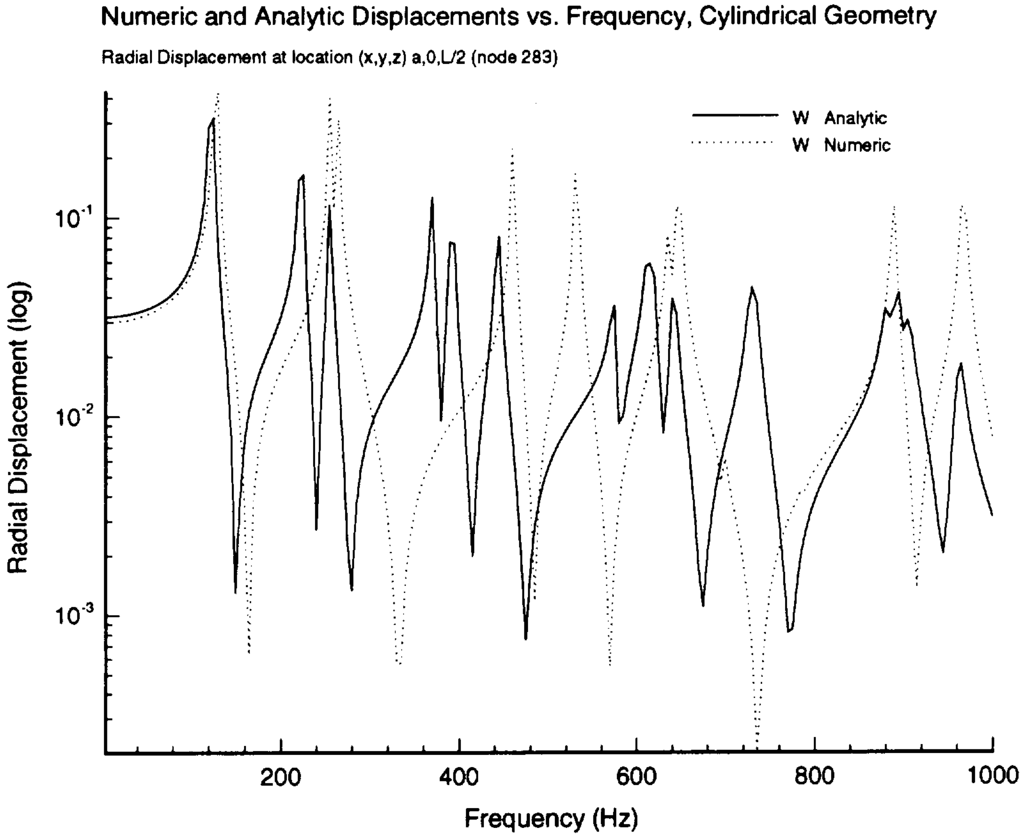

Figure A77: Radial displacement at the coordinates $r = a$, $\theta = 0$, $z = \frac{l}{2}$ for the fluid/structure cylinder. NASTRAN direct frequency response analysis. Analytic and numeric solutions shown.

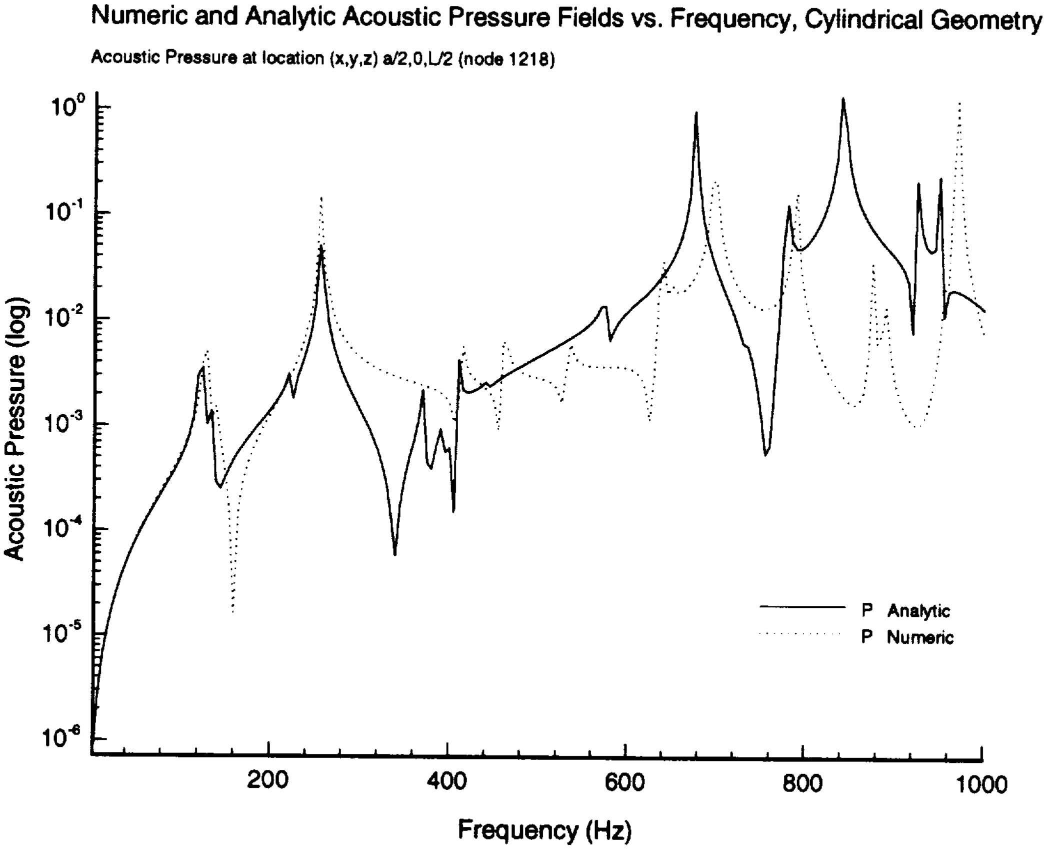

Figure A78: Acoustic pressure at the coordinates $r = \frac{a}{2}$, $\theta = 0$, $z = \frac{l}{2}$ for the fluid/structure cylinder. NASTRAN direct frequency response. Analytic and numeric solutions shown

Appendix B: FORTRAN Codes

This Appendix contains a source code listing of the FORTRAN programs written in support of this work. A brief description of each of these codes is shown in Table 11.

General Data Manipulation	
prefluid.f	NASTRAN bulk data file is modified such that it contains fluid elements only.
pch2res.f	Converts NASTRAN .pch file to a PATRAN-readable .res file. Fluid pressures at each node are written as "displacements" in the x-direction.
convert.f	Converts NASTRAN .pch file to a TECPLOT data file. For use with forced-response analysis
Written for Cubic Geometry Models	
reader.f	Compares analytic and numeric normal modes solutions for fluid-only cube at each node.
setmaker.f	Creates ACMODL card for a cubic geometry.
plate.f	Compares analytic and numeric normal modes solutions for the fluid/structure cube at each node.
forres.f	Computes analytic forced-response at a given location for the fluid/structure cubic geometry.
Written for Cylindrical Geometry Models	
compare.f	Compares analytic and numeric normal modes solutions at each node for fluid-only cylinder.
card.f	Creates ACMODL card for a cylindrical geometry.
modesm.f	Calculates the natural frequencies of a thin, elastic, cylindrical shell using Epstien-Kennard theory.
forced2.f	Computes analytic forced-response at a given location for the fluid/structure cylindrical geometry.

Table 11 Description of FORTRAN codes used.

The codes listed in this Appendix have been written specifically for this study and should be considered to be research code only. They are provided for completeness. Their successful operation cannot be guaranteed outside the scope of this work.

Program prefluid.f

```
c1234&123456789012
      program pre_fluid
      character*8  minus1,fst8,snd8,trd8,frt8,fth8,six8,
     & sth8,eth8,nth8,tth8
      character*20 zfile,ofile
      minus1='-1      '
      write(*,*)'ENTER BDF FILE NAME'
      read(*,1)zfile
 1    format(a)
      ofile='P'//zfile
      open(unit=10,file=zfile,status='old')
      open(unit=11,file=ofile,status='unknown')
 100  read(10,1121,end=200)fst8,snd8,trd8,frt8,fth8,six8
     &         ,sth8,eth8,nth8,tth8
      if(fst8.eq.'GRID    ')then
      write(11,1121)fst8,snd8,trd8,frt8,fth8,six8,minus1,
     &              eth8,nth8,tth8
      elseif(fst8.eq.'MAT1    ')then
      fst8='MAT10   '
      trd8='        '
      frt8='1.170E-7'
      fth8='13620.0 '
      write(11,1121)fst8,snd8,trd8,frt8,fth8,six8
      elseif(fst8.eq.'PSOLID  ')then
      frt8='        '
      fth8='        '
      six8='        '
      sth8='        '
      eth8='PFLUID  '
      nth8='        '
      tth8='        '
      write(11,1121)fst8,snd8,trd8,frt8,fth8,six8,sth8,
     &              eth8,nth8,tth8
      elseif(fst8(1:6).eq.'ASSIGN')then
      elseif(fst8(1:4).eq.'TIME')then
         write(11,*)'TIME  600'
         do 110 ii=1,600
         read(10,1121,end=200)fst8,snd8,trd8,frt8,fth8,six8
     &         ,sth8,eth8,nth8,tth8
         if(fst8(1:4).eq.'CEND')goto 120
 110     continue
 120     write(11,*)'CEND'
      elseif(fst8.eq.'   METHO')then
      write(11,*)fst8,'D(STRUCT)',snd8(2:8)
      write(11,*)fst8,'D(FLUID)',snd8(2:8)
      elseif(fst8.eq.'   VECTO')then
      write(11,*)'   DISPLACEMENT(SORT1,PUNCH) = ALL'
      else
      write(11,1121)fst8,snd8,trd8,frt8,fth8,six8,sth8,
     &              eth8,nth8,tth8
      endif
      goto 100
 200  close(10)
      close(11)
 1121 format(10a8)
      stop
      end
```

Program pch2res.f

```
C1234&123456789012
      program pch2res
      character*72 title,subtitle,label,analysistype,datatype
      character*30 ifile,ofile
      character*15 subcase,eigen
      character*8 icase,inumber,sn,mn
      character*6  mode,cont
      write(*,*)'ENTER PUNCH FILE'
      read(*,1)ifile
      write(*,*)'ENTER NUMBER OF DATA POINTS'
      read(*,*)numdata
      open(unit=11,file=ifile,status='old')
 1101 read(11,2,end=205)title,iline
      read(11,2)subtitle,iline
      read(11,2)label,iline
      read(11,2)analysistype,iline
      read(11,2)datatype,iline
 1    format(a)
 2    format(a72,i8)
c
      read(11,3)subcase,icase,iline
      read(11,4)eigen,freq,mode,inumber,iline
 3    format(a15,5x,a8,44x,i8)
 4    format(a15,E13.4,2x,a6,a8,28x,i8)
      if(analysistype(2:11).eq.'EIGENVECTO')then
c extract job name
      i=0
 105  i=i+1
      if(ifile(i:i).ne.'.'.and.ilen.lt.20)then
         ofile(1:i)=ifile(1:i)
         ilen=i
         goto 105
       else
       endif
c extract mode number
      i=0
      im=0
 106  i=i+1
      if(inumber(i:i).ne.' '.and.im.le.8)then
         im=im+1
         mn(im:im)=inumber(i:i)
         goto 106
       elseif(i.lt.8)then
         goto 106
       else
       endif
c extract subcase number
      i=0
      is=0
 107  i=i+1
      if(icase(i:i).ne.' '.and.is.le.8)then
         is=is+1
         sn(is:is)=icase(i:i)
         goto 107
       elseif(i.lt.8)then
         goto 107
       else
       endif
c create output file name
      write(*,*)inumber,mn
      write(*,*)icase,sn
      ofile=ifile(1:ilen)//'_mode'//mn(1:im)//'.dis.'//sn(1:is)
      else
        write(*,*)analysistype(1:11)
        ofile='error.dis.1'
      endif

c
      defmax=1
      nwidth=6
      ndmax=int(numdata/2)
      open(unit=12,file=ofile,status='unknown')
      write(12,1003)eigen,freq
      write(12,1111)numdata,numdata,defmax,ndmax,nwidth
      write(12,5)title
      write(12,5)subtitle
 5    format(a72)
 1003 format(a15,E13.4)
      do 100 i=1,numdata
      read(11,1001)cont,node,typevar,a1,a2,a3,iline
      read(11,1002)cont,a4,a5,a6,iline
      write(12,1112)node,a1,a2,a3,a4,a5,a6
 1111 format(2i9,e15.9,2i9)
 1112 format(i8,(5e13.7))
 1001 format(a6,i8,a4,3(5x,e13.4),i8)
 1002 format(a6,12x,3(5x,e13.4),i8)
 100  continue
      close(12)
      goto 1101
 205  close(11)
      stop
      end
```

Program convert.f

```
      program convert
c
c  Program to convert xxx.pch file from NASTRAN forced-response
c  analysis to PATRAN-readable XY-Plot files or TECPLOT
c  formatted data files.
c
c  Input consists of punch file from NASTRAN forced-response
c  analysis. The total number of nodes (maxn) in the model, the
c  number of nodes calculated (iwant), the number of frequencies
c  (maxfreq) and the output format must be programed prior to
c  compiling.
c
c  Output consists of XY-Plot or TECPLOT data for displacement
c  magnitudes at a point vs. frequency.  XYPlot files contain
c  magnitude or phase information for all nodes.  Tecplot files
c   contain infromation for phase and magnitude for only one node.
c
c  Written by C.M.Fernholz 48221
c
c  Declarations
c
      implicit real*8(a-h,o-y)
      implicit complex(z)
c
      parameter (maxn=5643,maxfreq=126,iwant=3,pi=3.141592654)
c
      character*72 title,subtitle,label,analysistype,datatype
      character*30 ifile,ofileRe,ofileIm,ofileu,ofilev,ofilew,
     &             ofilex,ofiley,ofilez
      character*15 subcase, point,displace
      character*8  icase,inumber,sn,mn
      character*6  mode,cont
      character*4  typevar
c
      dimension disr(6),disi(6),frequency(maxfreq),node(maxn),
     &          freqRe(maxfreq,maxn,6),freqIm(maxfreq,maxn,6),
     &          udata(maxn,maxfreq),utheta(maxn,maxfreq),
     &          vdata(maxn,maxfreq),vtheta(maxn,maxfreq),
     &          wdata(maxn,maxfreq),wtheta(maxn,maxfreq),
     &          uphase(maxn,maxfreq),vphase(maxn,maxfreq),
     &          wphase(maxn,maxfreq),size(3)
c
      logical patran,tecplot
c
c  Begin program
c
c  Choose output type
c
      patran=.FALSE.
      tecplot=.TRUE.
c
      write(*,*)'Enter punch file name'
      read(*,1)ifile
 1    format(a)
c
c  Open punch file
      open(unit=10,file=ifile,status='old')
c
      do 50 i=1,iwant
c
c  Read headers from punch file
c
         read(10,1000)title,iline
         read(10,1000)subtitle,iline
         read(10,1000)label,iline
         read(10,1000)analysistype,iline
         read(10,1000)datatype,iline
         read(10,1005)subcase,icase,iline
         read(10,1010)point,node(i)
c
c  Read in frequencies, displacements
c
         do 100 j=1,maxfreq
c
            read(10,1100)frequency(j),typevar,
     &                   (disr(k),k=1,3),iline
            read(10,1105)cont,(disr(k),k=4,6),iline
            read(10,1105)cont,(disi(k),k=1,3),iline
            read(10,1105)cont,(disi(k),k=4,6),iline
c
            do 9 k=1,3
        zdisp=cmplx(disr(k),disi(k))
        partone=real(zdisp)
        parttwo=imag(zdisp)
               size(k)=SQRT(partone*partone+parttwo*parttwo)
 9          continue
c
c  Store real and imaginary displacements for each frequency in an
c  array. Note:  used if PATRAN fringe plots of displacements for
c  a given frequency are desired.
c
            do 160 k=1,6
               freqRe(j,i,k)=disr(k)
               freqIm(j,i,k)=disi(k)
 160        continue
c
c  Store displacement magnitudes for each node
c
            if (patran) then
               udata(i,j)=size(1)
               vdata(i,j)=size(2)
               wdata(i,j)=size(3)
            endif
c
            if (tecplot) then
               udata(node(i),j)=size(1)
               vdata(node(i),j)=size(2)
               wdata(node(i),j)=size(3)
            endif
c
c  Store displacement phase angles for each node
c
            if (disr(1).EQ.0.0) then
               if (disi(1).GT.0.0) then
                  utheta(i,j)=90.0
               elseif (disi(1).LT.0.0) then
                  utheta(i,j)=-90.0
               elseif (disi(1).EQ.0.0) then
                  utheta(i,j)=0.0
               endif
            else
               utheta(i,j)=(180.0/pi)*ATAN(disi(1)/disr(1))
            endif
c
            if (disr(2).EQ.0.0) then
               if (disi(2).GT.0.0) then
                  vtheta(i,j)=90.0
               elseif (disi(2).LT.0.0) then
                  vtheta(i,j)=-90.0
               else
                  vtheta(i,j)=0.0
               endif
            else
               vtheta(i,j)=(180.0/pi)*ATAN(disi(2)/disr(2))
            endif
c
            if (disr(3).EQ.0.0) then
               if (disi(3).GT.0.0) then
                  wtheta(i,j)=90.0
               elseif (disi(3).LT.0.0) then
                  wtheta(i,j)=-90.0
               else
                  wtheta(i,j)=0.0
               endif
            else
               wtheta(i,j)=(180.0/pi)*ATAN(disi(3)/disr(3))
            endif
c
            if (tecplot) then
               uphase(node(i),j)=utheta(i,j)
               vphase(node(i),j)=vtheta(i,j)
               wphase(node(i),j)=wtheta(i,j)
            endif
c
 100     continue
 50   continue
c
      close(10)
c
c  Dummy print
      write(*,*)title
c
c  Magnitude output files for u,v, and w displacements
c
      if (patran) then
         ofileu='magu.xyd'
         ofilev='magv.xyd'
         ofilew='magw.xyd'
      elseif (tecplot) then
         ofileu='magu.plt'
         ofilev='magv.plt'
         ofilew='magw.plt'
      endif
c
c  Phase angle ouput files for u,v, and w displacements
c
      if (patran) then
         ofilex='phau.xyd'
         ofiley='phav.xyd'
         ofilez='phaw.xyd'
      elseif (tecplot) then
         ofilex='phau.plt'
         ofiley='phav.plt'
         ofilez='phaw.plt'
      endif
c
      open(unit=40,file=ofileu,status='unknown')
      open(unit=50,file=ofilev,status='unknown')
      open(unit=60,file=ofilew,status='unknown')
      open(unit=70,file=ofilex,status='unknown')
      open(unit=80,file=ofiley,status='unknown')
      open(unit=90,file=ofilez,status='unknown')
c
      if (patran) then
         do 300 l=1,iwant
c
            write(40,1300)'XYDATA,U-DISP, NODE',node(l)
            write(50,1300)'XYDATA,V-DISP, NODE',node(l)
            write(60,1300)'XYDATA,W-DISP, NODE',node(l)
            write(70,1300)'XYDATA,U-PHASE NODE',node(l)
            write(80,1300)'XYDATA,V-PHASE NODE',node(l)
            write(90,1300)'XYDATA,W-PHASE NODE',node(l)
c
            do 350 m=1,maxfreq
               write(40,1305)frequency(m),udata(l,m)
               write(50,1305)frequency(m),vdata(l,m)
               write(60,1305)frequency(m),wdata(1,m)
               write(70,1305)frequency(m),utheta(1,m)
               write(80,1305)frequency(m),vtheta(1,m)
               write(90,1305)frequency(m),wtheta(1,m)
 350        continue
c
 300     continue
c
         do 14 n=40,90,10
            write(n,1310)'END'
 14      continue
c
      elseif (tecplot) then
c
         write(40,1400)'TITLE = "Magnitude for u-displacements"'
         write(50,1400)'TITLE = "Magnitude for v-displacements"'
         write(60,1400)'TITLE = "Magnitude for w-displacements"'
         write(70,1401)'TITLE = "Phase Angle for u-displacements"'
         write(80,1401)'TITLE = "Phase Angle for v-displacements"'
         write(90,1401)'TITLE = "Phase Angle for w-displacements"'
c
         do 11 n=40,60,10
            write(n,1405)'VARIABLES = "Frequency","Displacement"'
            write(n,1410)'ZONE T="Numeric Solution", I =',maxfreq
 11      continue
c
         do 12 n=70,90,10
            write(n,1406)'VARIABLES = "Frequency","Phase"'
            write(n,1411)'ZONE T="Phases", I =',maxfreq
 12      continue
c
         write(*,*)'Enter desired node number for analysis'
         read(*,2)l
 2       format(i)
c
         do 450 m=1,maxfreq
            write(40,1415)frequency(m),udata(1,m)
            write(50,1415)frequency(m),vdata(1,m)
            write(60,1415)frequency(m),wdata(1,m)
            write(70,1415)frequency(m),uphase(1,m)
            write(80,1415)frequency(m),vphase(1,m)
            write(90,1415)frequency(m),wphase(1,m)
 450     continue
c
      endif
c
      do 13 n=40,90,10
         close(n)
 13   continue
c
 1000 format(a72,i8)
 1005 format(a15,5x,a8,44x,i8)
 1010 format(a13,5x,i8,46x,i8)
 1100 format(4x,e13.4,a1,3(5x,e13.4),i8)
 1105 format(a6,12x,3(5x,e13.4),i8)
 1200 format(a15,e13.4)
 1205 format(2i9,e15.9,2i9)
 1210 format(a72)
 1215 format(i8,(5e13.7)/e13.7)
 1300 format(a19,i4)
 1305 format(f10.3,2x,f10.6)
 1310 format(a3)
 1400 format(a39)
 1401 format(a41)
 1405 format(a38)
 1406 format(a31)
 1410 format(a30,i5)
 1411 format(a20,i5)
 1415 format(f10.3,2x,e13.5)
 1420 format(a25,i5)
c
 9999 stop
      end
```

Program reader.f

```
      program reader
c
c  Program to compare analytical and numerical displacement
c  calculations for cubic geometry, fluid only.
c
c  Input files include:
c     NASTRAN-generated neutral file:  used to determine
c     geometric location of each node in the model.  Neutral
c     file should contain only node location information.
c     NASTRAN-generated punch file:  used to determine the
c     displacement of each node, as calculated by NASTRAN.
c
c  Output files:
c     read.out:  includes information about the maximum
c     displacement error for each direction and the maximum
c     displacement error magnitude for the system.
c     error.dis.1:  PATRAN-readable displacement file that
c     can be used to display the displacement errors for the
c     model in fringe plot form.
c
c  It is necessary to specify the number of nodes (maxn) and the
c  number of elements (maxe) in the model prior to compiling.
c
c  Declarations
c
      implicit real*8(a-h,o-z)
c
      parameter (maxn=4961, maxe=1000)
c
      character*15 zfile,pchfile,eigen
      character*8 count,typevar
      character za(maxn)*1
c
      dimension x(maxn),y(maxn),z(maxn),
     &          aprod(maxn),
     &          px(maxn),py(maxn),pz(maxn),
     &          node(maxn),error(maxn),
     &          ndf(maxn),junk(6)
c
c  Begin program
c
      write(*,*)'Enter neutral file name'
      read(*,1)zfile
      write(*,*)'Enter punch file name'
      read(*,1)pchfile
 1    format(a)
c
c  Define geometry (cube)
c
      side=5.0
      pi=3.141592654
      idn=1.0
c
c  Mode shape of interest (nnn=x-dir,mmm=y-dir,kkk=z-dir)
c
      nnn=1
      mmm=1
      kkk=-3
c
c  Open neutral file
      open(unit=10,file=zfile,status='old')
c  Open punch file
      open(unit=20,file=pchfile,status='old')
c  Open output file
      open(unit=30,file='read.out',status='unknown')
c
c  Sort loop to determine maximum displacement in punch file
c  Punch file should contain data only for mode of interest
c
      bigx=0
      bigy=0
      bigz=0
c
      do 50 j=1,(maxn-1)
c
         read(20,1004) cont,node(j),typevar,
     &                 px(j),py(j),pz(j),iline
         read(20,1005) cont,a4,a5,a6,iline
c
         if (ABS(px(j)).GT.ABS(bigx)) bigx=px(j)
         if (ABS(py(j)).GT.ABS(bigy)) bigy=py(j)
         if (ABS(pz(j)).GT.ABS(bigz)) bigz=pz(j)
c
 50   continue
c
      if (bigx.EQ.0.0) bigx=1.0
      if (bigy.EQ.0.0) bigy=1.0
      if (bigz.EQ.0.0) bigz=1.0
c
      write(30,1)'Error:  Analytic vs. Numeric Solutions'
      write(30,1015)'Cubic geometry, mode ',nnn,mmm,kkk
      write(30,1010)'bigx =    ',bigx
      write(30,1010)'bigy =    ',bigy
      write(30,1010)'bigz =    ',bigz
c
      rewind(unit=20)
c
 100  read(10,1000)idpacket,idn,iv,kc,n1,n2,n3,n4,n5
c
      if (idpacket.eq.99) then
         close(10)
         close(20)
         goto 500
c
      elseif (idpacket.eq.1) then
c
c  Determine location of grid point
```

```
         read(10,1001) x(idn),y(idn),z(idn)
         read(10,1002) icf,za(idn),ndf(idn),ncnfig,ncid,
     &                 (junk(i),i=1,6)
c
c  Determine analytic displacement of grid point, mode (1,1,1)
         ax=sin((nnn*pi*x(idn))/side)
         ay=sin((mmm*pi*y(idn))/side)
         az=sin((kkk*pi*z(idn))/side)
         aprod(idn)=ax*ay*az
c
c  Determne grid point displacement from punch file
         read(20,1004) cont,node(idn),typevar,
     &                 px(idn),py(idn),pz(idn),iline
         read(20,1005) cont,a4,a5,a6,iline
         px(idn)=px(idn)/bigx
         py(idn)=py(idn)/bigy
         pz(idn)=pz(idn)/bigz
c
      else
         write(*,*)'Invalid packet ID'
         stop
      endif
c
      goto 100
c
c  Dummy print
 500  write(*,*)'write to read.out completed'
c
c  Calculate displacement error at each node, max error for system
c
      bigerror=0
c
      do 250 m=1,maxn
c
         error(m)=aprod(m)-px(m)
c
         if (ABS(error(m)).GT.bigerror) then
            bigerror=error(m)
         endif
c
 250  continue
c
      write(30,1010)'Max error ',bigerror
c
      close(30)
c
c  Write PATRAN input file
c
      eigen='$EIGENVALUE ='
      freq=bigerror
      defmax=1
      nwidth=6
      ndmax=int(maxn/2)
c
      open(unit=40,file='error.dis.1',status='unknown')
      write(40,1100)eigen,freq
      write(40,1101)maxn,maxn,defmax,ndmax,nwidth
      write(40,1102)'$TITLE GOES HERE'
      write(40,1102)'$SUBTITLE=LOAD_CASE_ONE'
c
      do 200 l=1,maxn
         write(40,1103)node(l),error(l),0.0,0.0,0.0,0.0,0.0
 200  continue
c
      close(40)
c
 1000 format(i2,8i8)
 1001 format(3e16.9)
 1002 format(i1,1a1,3i8,2x,6i1)
 1003 format(4e16.9)
 1004 format(a6,i8,a4,3(5x,e13.4),i8)
 1005 format(a6,12x,3(5x,e13.4),i8)
 1010 format(a10,e18.9)
 1015 format(a21,i2,ix,i2,1x,i2)
 1100 format(a15,e13.4)
 1101 format(2i9,e15.9,2i9)
 1102 format(a72)
 1103 format(i8,(5e13.7))
c
      stop
      end
```

Program setmaker.f

```
      program setmaker
c
c  Program to produce NASTRAN ACMODL set cards for a cubic geometry
c  as well as fluid/solid grid set.  If grid point is determined to
c  be a fluid point, grid co-ordinate remains "-1".  If grid point
c  is a solid, co-ordinate ID is changed to "0." (solid).
c
c  Input consists of original GRID data only from the NASTRAN
c  bulk data file.  It is necessary to specifiy the total number
c  of grids in the model (maxn) prior to compiling.
c
c  Output consists of two files:
c     grids.out:  contains fluid/solid grid points for .bdf file
c     set555:     contains ACMODL set cards.
c  In the ACMODL output file, SET1=555 contains the solid grid
c  points and SET1=666 contains the fluid grid points.  Line
c  continuation markers start at "AAAAAAA".
c
c  Assumptions:  Structure nodes are assumed to lie in two planes
c  only; one at z=0 and the other at z=5.  If a fluid node as a z
c  coordinate of either 0 or 5, it is assumed to lie on the fluid/
c  structure interface and will be included in the ACMODL card
c  set.  It is assumed that every structure node is on the fluid/
c  structure interface.  The structure nodes are assumed to be
c  sequential (ie xxx THRU xxx).
c
c  Program written by C.M.Fernholz (48221)
c
c  Declarations
c
      implicit real*8(a-h,o-z)
c
      parameter (maxn=5643)
c
      character*8 grid,set,thru,cont,con2
      character*15 gridfile,yfile,zfile
c
      logical test,check
c
      dimension inode(maxn),m(8)
c
c  Begin program
c
      write(*,*)'Enter GRID file name'
      read(*,2)gridfile
      write(*,*)'Enter minimum structure grid point ID'
      read(*,1)nodemin
      write(*,*)'Enter maximum structure grid point ID'
      read(*,1)nodemax
 1    format(i)
 2    format(a)
c
      yfile='grids.out'
      zfile='set555'
c
c  Open file containing grid information
      open(unit=10,file=gridfile,status='old')
c  Open file for ACMODL gridset
      open(unit=20,file=zfile,status='unknown')
c  Open file for grids output
      open(unit=30,file=yfile,status='unknown')
c
c  Read grid file, determine fluid and solid grids, write new gridset
c
      do 20 i=1,maxn
         read(10,1000)grid,node,xx,yy,zz,fs
         if (node.LT.nodemin.OR.node.GT.nodemax) then
            fs=-1
            write(30,1000)grid,node,xx,yy,zz,fs
         elseif (node.GE.nodemin.OR.node.LE.nodemax) then
            fs=0
            write(30,1000)grid,node,xx,yy,zz,fs
         else
            write(*,*)'WARNING:  Grid type indeterminate'
         endif
 20   continue
c
      close(10)
      rewind(unit=30)
c
c  Write structure SET card
c
      isset=555
      set='SET1'
      thru='    THRU'
      write(20,1005)set,isset,nodemin,thru,nodemax
c
c  Read grid file, determine which fluid points lie on interface
c
      jj=0
      do 50 i=1,maxn
c
         read(30,1000)grid,node,xx,yy,zz,fs
c
         if (node.LT.nodemin.OR.node.GT.nodemax) then
c
            if (zz.EQ.(0.0).OR.zz.EQ.(5.0)) then
               jj=jj+1
               inode(jj)=node
            endif
c
         endif
c
 50   continue
c
      close(30)
c
c  Write fluid SET card
c
 55   ifset=666
c
c  Logic for line continuation markers
c  Note:  'A' = char(65), '+' = char(43)
c
      m(1)=43
      do 60 i=2,8
         m(i)=65
 60   continue
c
      cont=char(m(1))//char(m(2))//char(m(3))//char(m(4))//
     &     char(m(5))//char(m(6))//char(m(7))//char(m(8))
      con2=cont
c
c  Write first seven fluid points to SET card (first line)
      write(20,1010)set,ifset,(inode(k),k=1,7),cont
c
c  Write groups of remaining fluid grid nodes eight at a time
c
      imax=INT((jj-7)/8)
      ithing=8
      icount=1
      check=.TRUE.
c
 100  if (check) then
         cont=con2
         test=.TRUE.
 80      if (test) then
            i=8
            if (m(i).GE.90) then
               m(i)=65
               m(i-1)=m(i-1)+1
               if (m(i-1).LT.90) then
                  test=.FALSE.
               else
                   i=i-1
                   if (i.LT.2) then
                      write(*,*)'Too many grid points'
                      goto 999
                   endif
               endif
            else
               m(i)=m(i)+1
               test=.FALSE.
            endif
 85         goto 80
         endif
c
         con2=char(m(1))//char(m(2))//char(m(3))//char(m(4))//
     &        char(m(5))//char(m(6))//char(m(7))//char(m(8))
c
         write(20,1015)cont,(inode(k),k=ithing,ithing+7),con2
c
         ithing=ithing+8
         icount=icount+1
         if (icount.GT.imax) then
            check=.FALSE.
         endif
         goto 100
      endif
c
c  Write last line of file
c
      cont=con2
      write(20,1020)cont,(inode(k),k=ithing,jj)
c
      close(20)
c
 1000 format(a8,i8,8x,3F8.4,i8)
 1005 format(a8,2i8,a8,i8)
 1010 format(a8,i8,7i8,a8)
 1015 format(a8,8i8,a8)
 1020 format(a8,8i8)
c
 999  stop
      end
```

Program plate.f

```
      program plate
c
c  Program to compare analytic and numeric displacement
c  calculations for the cubic fluid/structure geometry.
c  Errors for w-displacement and pressure are determined.
c
c  Input files include:
c  PATRAN-generated punch (.pch) file.  Numeric dis-
c  placements for each node are read from this file.
c  PATRAN-generated neutral (.out) file containing
c  node information only.  Geometric node locations
c  are read from this file.
c
c  Output files include:
c  error.out:  contains information about the maximum
c  error occuring anywhere in the model.
c  error.dis.1:  PATRAN-readable displacement file that
c  can be used to display the error results for the
c  model in fringe plot form.
c
c  It is necessary to specify the number of nodes (maxn),
c  the number of elements (maxe), and the mode shape of
c  interest prior to compiling.
c
c  Assumptions:
c  Analytic model uses an uncoupled solution.  NASTRAN
c  uses a coupled one.  For the first structural mode,
c  coupling will occur between the two plates of the
c  model.  The mode shape of the plate that is moving
c  will be superimposed (with some attenuation of the
c  amplitude) upon the other plate.  To account for this,
c  displacement amplitudes in the two plates are handled
c  seperately in this code.  The rear plate (at z=0) is
c  assumed to be stationary.  The error.dis file can be
c  modified to delete the nodes corresponding to the
c  rear plate.
c
c  Written by C.M.Fernholz (48221)
c
c  Declarations
c
      implicit real (a-h,o-z)
c
      parameter (maxn=1573,maxe=1200,pi=3.141592654)
c
      character*72 title,subtitle,label,analysistype,
     &             datatype,header
      character*30 pchfile,neufile,header2
      character*15 subcase,eigen
      character*8  icase,an,mn
      character*6  mode,cont
      character*5  solidtype,fluidtype
      character*1  flustr
c
      dimension px(maxn),py(maxn),pz(maxn),node(maxn),
     &          w(maxn),p(maxn),serror(maxn),ferror(maxn),
     &          nodefluid(maxn),nodesolid(maxn),
     &          x(maxn),y(maxn),z(maxn),
     &          junk(6),itrash(9)
c
      logical fluid
c
c  Begin program
c
c  Choose mode shape of interest (m=x-dir,n=y-dir,k=z-dir)
c
      m=-2
      n=2
      k=0
c
c  Define cubic geometry
c
      side=5.0
      thick=0.0625
c
c  FEM geometry
c
      ii=0
      jj=0
      fluidtype='HEX8'
      solidtype='QUAD4'
c
      write(*,*)'Fluid or structure mode? (f/s)'
      read (*,1)flustr
      write(*,*)'Enter desired mode (eigenvalue) number'
      read (*,2)modenumber
      write(*,*)'Enter punch (.pch) file name'
      read (*,1)pchfile
      write(*,*)'Enter neutral (.out) file name'
      read (*,1)neufile
 1    format(a)
c
      if (flustr.EQ.'f') then
         fluid=.TRUE.
      elseif (flustr.EQ.'s') then
         fluid=.FALSE.
      endif
c
c  Open punch file
      open (unit=10,file=pchfile,status='old')
c  Open neutral file
      open (unit=20,file=neufile,status='old')
c  Open error output data file
      open (unit=30,file='error.out',status='unknown')
c
      write(*,*)'Enter min structure grid ID for front plate'
      read (*,2)nodeminf
      write(*,*)'Enter max structure grid ID for front plate'
      read (*,2)nodemaxf
      write(*,*)'Enter min structure grid ID for back plate'
      read (*,2)nodeminb
      write(*,*)'Enter max structure grid ID for back plate'
      read (*,2)nodemaxb
 2    format(i)
c
c  Read displacements from punch file
c
 100  bigx=0.0
      bigy=0.0
      bigzf=0.0
      bigzb=0.0
c
      read(10,1000)title,iline
      read(10,1000)subtitle,iline
      read(10,1000)label,iline
      read(10,1000)analysistype,iline
      read(10,1000)datatype,iline
      read(10,1005)subcase,icase,iline
      read(10,1010)eigen,freq,mode,inumber,iline
c
      do 150 j=1,maxn
c
         read(10,1015)cont,node(j),typevar,px(j),py(j),pz(j),
     &                 iline
         read(10,1020)cont,a4,a5,a6,iline
c
         if (node(j).LT.nodeminb.OR.node(j).LT.nodeminf.OR.
     &       node(j).GT.nodemaxb.OR.node(j).GT.nodemaxf) then
            if (ABS(px(j)).GT.ABS(bigx)) bigx=px(j)
         endif
c
         if (ABS(py(j)).GT.ABS(bigy)) bigy=py(j)
c
         if (node(j).LE.nodemaxb.AND.node(j).GE.nodeminb) then
            if (ABS(pz(j)).GT.ABS(bigzb)) bigzb=pz(j)
         elseif (node(j).LE.nodemaxf.AND.node(j).GE.nodeminf) then
            if (ABS(pz(j)).GT.ABS(bigzf)) bigzf=pz(j)
         endif
c
 150  continue
c
      if (inumber.NE.modenumber) goto 100
c
      close(10)
c
      if (bigx.EQ.0.0)  bigx=1.0
      if (bigy.EQ.0.0)  bigy=1.0
      if (bigzf.EQ.0.0) bigzf=1.0
      if (bigzb.EQ.0.0) bigzb=1.0
c
c  Write to error.out file
c
      write(30,1100)'Analytic vs. Numeric Solutions'
      write(30,1100)'Cubic Fluid/Structure Geometry'
      write(30,1105)maxe,solidtype,',',fluidtype,'elements'
      write(30,1110)maxn,'nodes'
c
      if (fluid) then
         write(30,1130)'Fluid analysis'
      elseif (.NOT.fluid) then
         write(30,1130)'Solid analysis'
      endif
c
      write(30,1115)'bigx = ',bigx
      write(30,1115)'bigy = ',bigy
      write(30,1115)'bizgf= ',bigzf
      write(30,1115)'bigzb= ',bigzb
c
c  Normalize displacements
c
      do 200 j=1,maxn
         px(j)=px(j)/ABS(bigx)
         py(j)=py(j)/ABS(bigy)
         if (node(j).LE.nodemaxb.AND.node(j).GE.nodeminb) then
            pz(j)=pz(j)/bigzb
         elseif(node(j).LE.nodemaxf.AND.node(j).GE.nodeminf)then
            pz(j)=pz(j)/bigzf
         endif
 200  continue
c
c  Read grid point locations from neutral file
c
      read(20,1200)(itrash(i),i=1,9)
      read(20,1205)header
      read(20,1200)(itrash(i),i=1,9)
      read(20,1210)header2
c
 300  read(20,1200)idpacket,idn,iv,kc,n1,n2,n3,n4,n5
c
      if (idpacket.EQ.99) then
         close(20)
         goto 350
c
      elseif (idpacket.EQ.1) then
c
         read(20,1215)x(idn),y(idn),z(idn)
         read(20,1220)icf,za,ndf,ncnfig,ncid,(junk(i),i=1,6)
c
         if (idn.LE.nodemaxb.AND.idn.GE.nodeminb) then
c
            ii=ii+1
            nodesolid(ii)=idn
            w(ii)=0.0
c
            serror(ii)=w(ii)-pz(idn)
c
         elseif (idn.LE.nodemaxf.AND.idn.GE.nodeminf) then
c
            ii=ii+1
            nodesolid(ii)=idn
            ax=sin(m*pi*x(idn)/side)
            ay=sin(n*pi*y(idn)/side)
            w(ii)=ax*ay
c
            serror(ii)=w(ii)-pz(idn)
c
         else
c
            jj=jj+1
            nodefluid(jj)=idn
            ax=sin(m*pi*x(idn)/side)
            ay=sin(n*pi*y(idn)/side)
            az=cos(k*pi*z(idn)/side)
            p(jj)=ax*ay*az
c
            ferror(jj)=p(jj)-px(idn)
c
         endif
c
      else
         write(*,*)'WARNING: Invalid packet ID'
c
      endif
c
      goto 300
c
c  Determine maximum errors in model
c
 350  bigerrorb=0.0
      bigerrorf=0.0
c
      do 400 i=1,ii
         if (nodesolid(i).LE.nodemaxb.AND.
     &       nodesolid(i).GE.nodeminb) then
            if (ABS(serror(i)).GT.ABS(bigerrorb)) then
               bigerrorb=serror(i)
            endif
         elseif (nodesolid(i).LE.nodemaxf.AND.
     &           nodesolid(i).GE.nodeminf) then
            if (ABS(serror(i)).GT.ABS(bigerrorf)) then
               bigerrorf=serror(i)
            endif
         endif
 400  continue
c
      write(30,1120)'Maximum error in front plate',bigerrorf
      write(30,1120)'Maximum error in rear plate ',bigerrorb
c
      bigerror=0.0
c
      do 450 i=1,jj
         if (ABS(ferror(i)).GT.ABS(bigerror)) then
            bigerror=ferror(i)
         endif
 450  continue
c
      write(30,1125)'Maximum error in fluid',bigerror
c
```

```
      close(30)
c
c Open error.dis file
      open (unit=40,file='error.dis.1',status='unknown')
c
      eigen='$EIGENVALUE ='
      defmax=1
      nwidth=6
      ndmax=INT(maxn/2)
c
      if (.NOT.fluid) then
         subtitle='$SUBTITLE = STRUCTURE ERROR'
      elseif (fluid) then
         subtitle='$SUBTITLE = FLUID ERROR'
      endif
c
      write(40,1300)eigen,freq
      write(40,1305)maxn,maxn,defmax,ndmax,nwidth
      write(40,1310)title
      write(40,1310)subtitle
c
      if (.NOT.fluid) then
c
         do 500 i=1,ii
            write(40,1315)nodesolid(i),0.0,0.0,
     &                    serror(i),0.0,0.0,0.0
 500     continue
c
      elseif (fluid) then
c
         do 550 i=1,jj
            write(40,1315)nodefluid(i),ferror(i),
     &                    0.0,0.0,0.0,0.0,0.0
 550     continue
c
      endif
c
      close(40)
c
 1000 format(a72,i8)
 1005 format(a15,5x,a8,44x,i8)
 1010 format(a15,e13.4,2x,a6,i8,28x,i8)
 1015 format(a6,i8,a4,3(5x,e13.4),i8)
 1020 format(a6,12x,3(5x,e13.4),i8)
 1100 format(a30)
 1105 format(i4,1x,a5,a1,1x,a5,1x,a8)
 1110 format(i4,1x,a5)
 1115 format(a7,f)
 1120 format(a28,1x,f)
 1125 format(a22,1x,f)
 1130 format(a14)
 1200 format(i2,8i8)
 1205 format(a72)
 1210 format(a30)
 1215 format(3e16.9)
 1220 format(i1,1a1,3i8,2x,6i1)
 1300 format(a15,e13.4)
 1305 format(2i9,e15.9,2i9)
 1310 format(a72)
 1315 format(i8,(5e13.7))
c
      stop
      end
```

Program forres.f

```
      program forres
c
c Program to determine the analytic solution for the forced
c frequency response of a cubic fluid/structure geometry.
c Model consists of fluid cube with sides of length A having
c structrual plates located at z = -A/2,A/2.
c
c No input files required.  Fluid and structure material
c properties must be programmed prior to compiling.
c
c Output consists of five files:
c     struc.eig:  Eigen values of the structure
c     sfreq.plt:  Disp. vs. freq. of the structure at a point
c     sphas.plt:  Phase angle vs. freq. of the struct at a point
c     pfreq.plt:  Pressure vs. freq. of a point in the fluid
c     pphas.plt:  Phase angle vs. freq. of the fluid at a point
c Output files are written in TECPLOT format.
c
c A non-coupled model is assumed. Structure obeys non-homogeneous
c plate equation of motion. Fluid obeys homogeneous wave equation.
c
c Written by C.M.Fernholz (48221)
c
c Declarations
c
      implicit real (a-h,o-y)
      implicit complex (z)
c
      parameter (nfreq=1000,step=5.0,pi=3.141592654)
c
      dimension zqmn(30,30), zdisp(nfreq),
     &          zqpress(30,30),zpress(nfreq)
c
      logical once,print
c
c Begin program
c
      once=.TRUE.
      print=.TRUE.
c
c Fluid and structure material properties
c
c Two pi
      twopi=2.0*pi
c Length of a side (inches)
      A=5.0
c Thickness of plates (inches)
      h=0.0625
c Young's Modulus for the plates (psi)
      E=10.3e6
c Density of the plate (slugs/in**3)
      rhos=2.5383e-4
c Damping coefficient
      eta=0.005
c Amplitude of input force (lbs)
      F=5.0
c Poission ratio for plate
      uu=0.334
c Speed of sound in fluid (in/sec)
      co=13620.0
c Density of the fluid (slugs/in**3)
      rhof=1.17e-7
c
c Structure stiffness
      d=(E*h**3)/(12*(1-uu*uu))
c
c Point of interest
c
      write(*,2)'Enter x co-ordinate'
      read(*,1)x
      write(*,*)'Enter y co-ordinate'
      read(*,1)y
      write(*,2)'Enter z co-ordinate'
      read(*,1)ez
      write(*,*)'Entered: ',x,y,ez
 1    format(f4.2)
 2    format(a)
c
      freq=0.0
      skip=step*twopi
      zo=(0.0,0.0)
      zi=(0.0,1.0)
c
      open(unit=10,file='struct.eig',status='unknown')
      open(unit=20,file='sfreq.plt',status='unknown')
      open(unit=25,file='sphas.plt',status='unknown')
      open(unit=30,file='pfreq.plt',status='unknown')
      open(unit=35,file='pphas.plt',status='unknown')
c
      do 100 i=1,nfreq
c
         freq=i*skip
         zdisp(i)=zo
         zpress(i)=zo
c
         do 200 m=1,20
            do 250 n=1,20
c
               wo=((pi*pi*(n*n+m*m))/(A*A))*SQRT(d/(rhos*h))
c
               if (print) then
                  write(10,1000)'Mode',m,',',n,'and
     &            Frequency',wo/twopi
               endif
c
               zdenom=cmplx(wo*wo-freq*freq,2.0*eta*wo*freq)
               zqmn(m,n)=4.0*F*sin(m*pi/2.0)*sin(n*pi/2.0)/
     &                   (A*A*rhos*h*zdenom)
               zdisp(i)=zdisp(i)+zqmn(m,n)*
     &                        sin(m*pi*x/A)*sin(n*pi*y/A)
c
               alf=-(freq/co)*(freq/co)+
     &              (n*pi/A)*(n*pi/A)+
     &              (m*pi/A)*(m*pi/A)

               if (alf.LT.0.0) then
                  alf=SQRT(ABS(alf))
                  zalpha=cmplx(0.0,alf)
               else
                  alf=SQRT(alf)
                  zalpha=cmplx(alf,0.0)
               endif
c
               zqpress(m,n)=-(rhof*freq*freq*zqmn(m,n))/zalpha
               zpress(i)=zpress(i)+zqpress(m,n)*
     &                    sin(m*pi*x/A)*
     &                    sin(n*pi*y/A)*
     &                   (csin(zalpha*ez)+
     &                   (1+ccos(zalpha*A))*ccos(zalpha*ez)/
     &                     csin(zalpha*A))
c
 250        continue
 200     continue
c
         call tecplot(freq,zpress(i),zdisp(i),
     &                 frequency,pmag,ppha,dmag,dpha)
c
         if (once) then
            write(20,2000)'TITLE = 'Analytic W-Displacements''
            write(25,2000)'TITLE = 'Analytic W-Phase Angles ''
            write(30,2000)'TITLE = 'Analytic U-Displacements''
            write(35,2000)'TITLE = 'Analytic U-Phase Angles ''
c
            write(20,2005)'VARIABLES = 'Frequency','Magnitude''
            write(25,2010)'VARIABLES = 'Frequency','Phase''
            write(30,2005)'VARIABLES = 'Frequency','Magnitude''
            write(35,2010)'VARIABLES = 'Frequency','Phase''
c
            do 300 j=20,35,5
               write(j,2015)'ZONE T='Analytic Solution''
 300        continue
c
            once=.FALSE.
         endif
c
         write(20,2020)frequency,dmag
         write(25,2020)frequency,dpha
         write(30,2020)frequency,pmag
         write(35,2020)frequency,ppha
c
         print=.FALSE.
 100  continue
c
      close(10)
      do 305 j=20,35,5
         close(j)
 305  continue
c
 1000 format(a4,1x,i2,a1,i2,1x,a13,1x,f10.2)
 2000 format(a34)
 2005 format(a35)
 2010 format(a31)
 2015 format(a26)
 2020 format(f10.2,1x,e13.5)
c
      stop
      end
c
c  ***************************************************************
      subroutine tecplot(freq,zpres,zdisp,fre,pmag,ppha,dmag,dpha)
c
c  Subroutine to format data for use by TECPLOT
c
c  Declarations
c
      implicit real (a-h,o-y)
      implicit complex (z)
c
      parameter (pi=3.141592654)
c
c  Begin subroutine
c
c  Frequency output
c
      fre=freq/(2*pi)
c
c  Pressure output
c
      preal=real(zpres)
      pimag=imag(zpres)
      pmag=SQRT(preal*preal+pimag*pimag)
c
      if (preal.EQ.0.0) then
         if (pimag.GT.0.0) then
            ppha=(pi/2.0)
         elseif (pimag.LT.0.0) then
            ppha=-(pi/2.0)
         else
            ppha=0.0
```

```
         endif
      else
         ppha=ATAN(pimag/preal)
      endif
c
c  Displacement output
c
      dreal=real(zdisp)
      dimag=imag(zdisp)
      dmag=SQRT(dreal*dreal+dimag*dimag)
c
      if (dreal.EQ.0.0) then
         if (dimag.GT.0.0) then
            dpha=(pi/2.0)
         elseif (dimag.LT.0.0) then
            dpha=-(pi/2.0)
         else
            dpha=0.0
         endif
      else
         dpha=ATAN(dimag/dreal)
      endif
c
      return
      end
c
c  ***************************************************************
      complex function ccos(z)
c
c  Function to return the cosine of a complex argument
c
c  Declarations
c
      implicit real (a-h,o-y)
      implicit complex (z)
c
c  Begin function
c
      argr=real(z)
      argi=imag(z)
c
      partone=cos(argr)*cosh(argi)
      parttwo=-1*sin(argr)*sinh(argi)
c
      ccos=cmplx(partone,parttwo)
c
      return
      end
c
c  ***************************************************************
      complex function csin(z)
c
c  Function to return the sine of a complex argument
c
c  Declarations
c
      real argr,argi,partone,parttwo
      complex z
c
c  Begin function
c
      argr=real(z)
      argi=imag(z)
c
      partone=sin(argr)*cosh(argi)
      parttwo=cos(argr)*sinh(argi)
c
      csin=cmplx(partone,parttwo)
c
      return
      end
```

Program compare.f

```
      program compare
c
c  Program to compare analytical and numerical displacement
c  calculations of first normal mode for a cylinder-shaped
c  geometry.
c
c  Input files:
c     PATRAN-generated neutral file: used to determine geometric
c     location of each node in the model.  Neutral file should
c     contain only node location information.
c     NASTRAN-generated punch file: used to determine the
c     displacement of each node, as calculated by NASTRAN.
c  It is necessary to specify the total number of nodes (maxn)
c  and the number of elements (maxe) in the model prior to
c  compiling.
c  Output files:
c     error.out:  includes information about the maximum
```

```
c     displacement
c     error for the system.
c     error.dis.1:  PATRAN-readable displacement file that can be
c     used to display the displacement errors for the model in
c     fringe plot form.
c
c  Written by C.M.Fernholz (48221)
c
c  Declarations
      implicit real*8(a-h,o-z)
c
      parameter (maxn=6609, maxe=2440, pi=3.141592654)
c
      character*15 zfile,pchfile,eigen
      character*8 count,typevar
      character za(maxn)*1
c
      dimension x(maxn),y(maxn),z(maxn),
     &          ax(maxn),ay(maxn),az(maxn),
     &          px(maxn),py(maxn),pz(maxn),
     &          node(maxn),error(maxn),aprod(maxn),
     &          ndf(maxn), junk(6)
c
c  Begin Program
c
      write(*,*)'Enter Neutral File Name'
      read(*,1)zfile
      write(*,*)'Enter Punch File Name'
      read(*,1)pchfile
 1    format(a)
c
c  Define Geometry
      axis=5.0
      radius=1.0
c
c  Zero-order Bessel function, first root
      root=2.40482556
c
c  Open neutral file
      open(unit=10,file=zfile,status='old')
c  Open punch file
      open(unit=20,file=pchfile,status='old')
c  Open error data output file
      open(unit=30,file='error.out',status='unknown')
c
c  Sort loop to determine maximum displacement in punch file
c
      bigx=0.0
      bigy=0.0
      bigz=0.0
c
      do 50 j=1,maxn
c
         read(20,1004) cont,node(j),typevar,
     &                 px(j),py(j),pz(j),iline
         read(20,1005) cont,a4,a5,a6,iline
 1004    format(a6,i8,a4,3(5x,e13.4),i8)
 1005    format(a6,12x,3(5x,e13.4),i8)
c
         if (ABS(px(j)).GT.ABS(bigx)) bigx=px(j)
         if (ABS(py(j)).GT.ABS(bigy)) bigy=py(j)
         if (ABS(pz(j)).GT.ABS(bigz)) bigz=pz(j)
c
 50   continue
c
      if (bigx.EQ.0.0) bigx=1.0
      if (bigy.EQ.0.0) bigy=1.0
      if (bigz.EQ.0.0) bigz=1.0
c
c  Dummy print
      write(*,*)'Displacements read'
c
c  Read in data from neutral file
c
 100  read(10,1000)idpacket,idn,iv,kc,n1,n2,n3,n4,n5
 1000 format(i2,8i8)
c
      if (idpacket.EQ.99) then
         close(10)
         close(20)
         goto 500
c
      elseif (idpacket.EQ.1) then
c
c  Determine location of grid point
         read(10,1001)x(idn),y(idn),z(idn)
         read(10,1002)icf,za(idn),ndf(idn),ncnfig,ncid,
     &                (junk(i),i=1,6)
 1001    format(3e16.9)
 1002    format(i1,1a1,3i8,2x,6i1)
c
c  Determine analytical displacement of grid point
         rad=SQRT(x(idn)*x(idn)+y(idn)*y(idn))
         axial=sin((pi*z(idn))/axis)
         radial=BESSJ0((rad*root)/radius)
         aprod(idn)=axial*radial
c
c  Normalize grid point displacements from punch file
         px(idn)=px(idn)/bigx
         py(idn)=py(idn)/bigy
         pz(idn)=pz(idn)/bigz
c
      else
         write(*,*)'Invalid packet ID'
         stop
      endif
c
      goto 100
c
c  Calculate displacement error at each node, max error for system
c
 500  bigerror=0.0
c
      do 250 m=1,maxn
         error(m)=aprod(m)-px(m)
         if (ABS(error(m)).GT.ABS(bigerror)) bigerror=error(m)
 250  continue
c
      write(30,1)'Error: Analytic vs. Numeric Solutions'
      write(30,1)'Cylinder, fluid only, mode 1,0,1'
      write(30,*)'bigx = ',bigx
      write(30,*)'bigy = ',bigy
      write(30,*)'bigz = ',bigz
      write(30,*)'bigerror= ',bigerror
c
c  Dummy print
      write(*,*)'Errors determined'
c
c  Write PATRAN input file
c
      eigen='$EIGENVALUE ='
      freq=bigerror
      defmax=1
      nwidth=6
      ndmax=INT(maxn/2)
c
      open(unit=40,file='error.dis.1',status='unknown')
      write(40,1100)eigen,freq
      write(40,1101)maxn,maxn,defmax,ndmax,nwidth
      write(40,1102)'$TITLE GOES HERE'
      write(40,1102)'$SUBTITLE=LOAD_CASE_ONE'
c
      do 200 l=1,maxn
         write(40,1103)node(l),error(l),0.0,0.0,0.0,0.0,0.0
 200  continue
c
      close(40)
      close(30)
c
 1100 format(a15,e13.4)
 1101 format(2i9,e15.9,2i9)
 1102 format(a72)
 1103 format(i8,(5e13.7))
c
 999  stop
      end
```

Program card.f

```
      program card
c
c  Program to producte NASTRAN ACMODL set cards for a cylindrical
c  geometry as well as fluid/solid grid set.  If grid point is
c  determined to be a fluid point, grid co-ordinate ID remains
c  "-1". If grid point is a solid, co-ordinate ID is changed to
c  "0" (solid).
c
c  Input data consists of original GRID data only from the
c  NASTRAN bulk data file.
c
c  Output files:
c      'grids.out' contains fluid/solid grid points
c      'set555' contains ACMODL set cards. SET1 555 = solid grids,
c       SET1 666 = fluid grids on fluid/solid interface.  Line
c       continuation markers start with 'AAAAAAA'
c
c  Assumptions: program matches each structure grid point with its
c  closest fluid grid point.  Model can only represent a fluid
c  volume surrounded by a structural shell.  The shell only touches
c  the fluid on one side.  2D elements must have nodal configura-
c  tions compatible with 3D elements (eg QUAD4 2D elements with HEX8
c  3D elements).
c  Program verifies that no fluid grid appears twice in the ACMODL
```

```
c  fluid set card.  Structure grid points are assumed to be
c  sequential (ie xxx THRU xxx).
c
c  Written by:  Christian M. Fernholz 4-8221
c
c  Declarations
c  Note:  necessary to assign maxn=number of grids in model,
c
      implicit real*8(a-h,o-z)
c
      parameter (maxn=868)
c
      character*8 grid,set,thru,cont,con2
      character*15 gridfile,yfile,zfile
      character*1 type(maxn)
c
      logical repeat,test,check
c
      dimension node(maxn),m(8),
     &          yf(maxn),xf(maxn),zf(maxn),nodf(maxn),
     &          ys(maxn),xs(maxn),zs(maxn),nods(maxn),
     &          ytemp1(maxn),xtemp1(maxn),ntemp1(maxn),
     &          xtemp2(maxn),ntemp2(maxn)
c
c  Begin program
c
      write(*,*)'Enter grid file name'
      read(*,2)gridfile
      write(*,*)'Enter minimum structure grid point ID'
      read(*,1)nodemin
      write(*,*)'Enter maximum structure grid point ID'
      read(*,1)nodemax
 1    format(i)
 2    format(a)
c
      zfile='set555'
      yfile='grids.out'
c
      pi=3.141592654
      rad=1.0
c
c  Error tolerances
      toly=0.001
      tolz=0.001
      tolr=0.001
c
c  Initialize node array
      do 10 i=1,maxn
         node(i)=9999
 10   continue
c
c  Open file containing grid information
      open (unit=10,file=gridfile,status='old')
c  Open file for ACMODL set cards
      open (unit=20,file=zfile,status='unknown')
c  Open file for solid/fluid grid output
      open (unit=30,file=yfile,status='unknown')
c
c  Read grid file, determine fluid/solid grids, write new gridset
c
      do 20 i=1,maxn
c
         read(10,1000)grid,nod,xx,yy,zz,fs
c
         if (nod.LT.nodemin.OR.nod.GT.nodemax) then
            fs=-1
            write(30,1000)grid,nod,xx,yy,zz,fs
         elseif (nod.GE.nodemin.OR.nod.LE.nodemax) then
            fs=0
            write(30,1000)grid,nod,xx,yy,zz,fs
         else
            write(*,*)'WARNING:  Grid type indeterminate'
         endif
 20   continue
c
      close(10)
      rewind(unit=30)
c
c  Write structure grid point ACMODL set card
c
      isset=555
      set='SET1'
      thru='    THRU'
      write(20,1005)set,isset,nodemin,thru,nodemax
c
c  Read grid co-ordinates (x,y,z) from new grid file
c
      j=0
      k=0
      irad=0
c
      do 50 i=1,maxn
c
         read(30,1000)grid,nod,xx,yy,zz,fs
c
         if (nod.LT.nodemin.OR.nod.GT.nodemax) then
            j=j+1
            xf(j)=xx
            yf(j)=yy
            zf(j)=zz
            nodf(j)=nod
            radius=SQRT(xx*xx+yy*yy)
            if (radius.LT.(rad+tolr).AND.radius.GT.(rad-tolr)) then
               irad=irad+1
            endif
         elseif (nod.GE.nodemin.OR.nod.LE.nodemax) then
            k=k+1
            xs(k)=xx
            ys(k)=yy
            zs(k)=zz
            nods(k)=nod
         else
            write(*,*)'ERROR:  Grid type indeterminate'
            goto 999
         endif
 50   continue
c
c  Dummy print
      write(*,*)'Grids successfully read'
      write(*,*)'Fluid grids total: ',j
      write(*,*)'Structure grids total: ',k
c  Dummy print
      write(*,*)'Number of surface structure nodes: ',k
      write(*,*)'Number of surface fluid nodes:     ',irad
c
c  Verify that two fluid grid points do not occupy same location
c
      repeat=.FALSE.
      n=0
c
      do 700 i=1,(j-1)
         do 710 ii=(i+1),j
c
            if (xf(i).EQ.xf(ii).AND.
     &                 yf(i).EQ.yf(ii).AND.
     &                 zf(i).EQ.zf(ii)) then
               repeat=.TRUE.
               n=n+1
            endif
c
 710     continue
 700  continue
c
      if (repeat) then
         write(*,*)'WARNING: ',n,' Fluid grids identical'
      endif
c
c Match structure grid points to closest fluid grid point
c
      do 500 i=1,k
c
c  Find all fluid grids with same z coordinate
c
         ll=1
         do 505 l=1,j
            if (zf(l).GT.(zs(i)-tolz).AND.
     &           zf(l).LT.(zs(i)+tolz)) then
               ytemp1(ll)=yf(l)
               xtemp1(ll)=xf(l)
               ntemp1(ll)=nodf(l)
               ll=ll+1
            endif
 505     continue
c
c  Find fluid grids with same y coordinate
c
         mm=1
         do 510 mn=1,ll
            if (ytemp1(mn).LT.(ys(i)+toly).AND.
     &           ytemp1(mn).GT.(ys(i)-toly)) then
               xtemp2(mm)=xtemp1(mn)
               ntemp2(mm)=ntemp1(mn)
               mm=mm+1
            endif
 510     continue
c
c  Find fluid grid with closest x coordinate
c
         err=1000000.0
         do 515 n=1,mm
            error=ABS(xtemp2(n)-xs(i))
            if (xtemp2(n).EQ.xs(i)) then
               node(i)=ntemp2(n)
               goto 500
            elseif (error.LT.err) then
               node(i)=ntemp2(n)
               err=error
            endif
 515     continue
c
 500  continue
c
c  Dummy print
      write(*,*)'Structure/fluid grids matched'
c
c  Check to see if any fluid points are repeated
c
      repeat=.FALSE.
      n=0
c
      do 600 i=1,(k-1)
         do 610 ii=(i+1),k
            if (node(i).EQ.node(ii)) then
               repeat=.TRUE.
               n=n+1
            endif
 610     continue
 600  continue
c
      if (repeat) then
         write(*,*)'WARNING:',n,' Nodes repeated in ACMODL card'
      endif
c
c  Write fluid grid point ACMODL set card
c
 55   ifset=666
      m(1)=43
      do 60 i=2,8
         m(i)=65
 60   continue
c
      cont=char(m(1))//char(m(2))//char(m(3))//char(m(4))//
     &     char(m(5))//char(m(6))//char(m(7))//char(m(8))
      con2=cont
c
c  Write first line of fluid grid points
      write(20,1010)set,ifset,(node(n),n=1,7),cont
c
c  Write remaining fluid grid points, eight at a time
c
      imax=INT((k-7)/8)
c
      ithing=8
      icount=1
      check=.TRUE.
c
 100  if (check) then
         cont=con2
         test=.TRUE.
 80      if (test) then
            i=8
            if (m(i).GE 90) then
               m(i)=65
               m(i-1)=m(i-1)+1
               if (m(i-1).LT.90) then
                  test=.FALSE.
               else
                  i=i-1
                  if (i.LT.2) then
                     write(*,*)'Too many grid points'
                     goto 999
                  endif
               endif
            else
               m(i)=m(i)+1
               test=.FALSE.
            endif
 85         goto 80
         endif
c
         con2=char(m(1))//char(m(2))//char(m(3))//char(m(4))//
     &        char(m(5))//char(m(6))//char(m(7))//char(m(8))
c
         write(20,1015)cont,(node(n),n=ithing,ithing+7),con2
c
         ithing=ithing+8
         icount=icount+1
         if (icount.GT.imax) then
            check=.FALSE.
         endif
         goto 100
      endif
c
c  Write last line of file
      cont=con2
      write(20,1020)cont,(node(n),n=ithing,k)
c
      close(30)
      close(20)
c
 1000 format(a8,i8,8x,3f8.4,i8)
 1001 format(a8,i8,8x,3f8.4)
 1005 format(a8,2i8,a8,i8)
 1010 format(a8,i8,7i8,a8)
 1015 format(a8,8i8,a8)
 1020 format(a8,8i8)
c
 999  stop
      end
```

Program modesm.f

```
      program modesm
c
c  Program to calculate natural frequencies of vibration for a
c  cylindrical shell.
c
c  No input files required.  Shell material properties and
c  geometric dimensions must be programmed prior to compiling.
c
c  Output file freq.out contains mode numbers and natural
c  frequency.
c
c  Uses Epstein-Kennard theory and solution outlined in Leissa:
c  Vibration of Shells, NASA SP-288
c
c  Declarations
c
      implicit real (a-h,o-y)
      implicit complex (z)
      real l,nu,lam,k1,k2,k0
c
      parameter (pi=3.141592654)
c
c  Begin program
c
c  Cylindrical shell, dimensions, material properties (Al)
c
c  Length (in)
      l=5.0
c  Radius (in)
      a=1.0
c  Thickness (in)
      h=0.0625
c  Poission's ratio
      nu=0.334
c  Young's Modulus (psi)
      E=10.3E6
c  Density (slugs/in**3)
      rho=2.5383E-4
c  Square root of -1
      zi=cmplx(0.0,1.0)
c
      open(unit=10,file='freq.out',status='unknown')
      write(10,1005)'Cylindrical Shell Natural Frequencies'
      write(10,1010)'Epstein-Kennard Theory'
      write(10,1015)'Mode','m','n','Roots (Re,Im):',
     &              'Root 1 (Hz)','Root 2 (Hz)','Root 3 (Hz)'
c
      do 200 m=0,10
      do 300 n=0,10
c
         if (m.EQ.0.AND.n.EQ.0) goto 300
c
         lam=(m*pi*a/l)
c
c  Define Donnel-Mustari constants
c
         k2=1.+.5*(3.-nu)*(n*n+lam*lam)+(h*h/(12.*a*a))*
     &      (n*n+lam*lam)*(n*n+lam*lam)
c
         k1=.5*(1-nu)*((3.+2.*nu)*lam*lam+n*n+(n*n+lam*lam)*
     &        (n*n+lam*lam)+((3.0-nu)/(1.0-nu))*(h*h/(12.*a*a))*
     &        (n*n+lam*lam)*(n*n+lam*lam)*(n*n+lam*lam))
c
         k0=.5*(1.-nu)*((1.-nu*nu)*(lam*lam*lam*lam)+
     &        (h*h/(12.*a*a))*(n*n+lam*lam)*(n*n+lam*lam)*
     &        (n*n+lam*lam)*(n*n+lam*lam))
c
c  Modifying constants for Epstein-Kennard theory
c
         delk2=(1+3*nu)/(1-nu)-(2-8*nu*nu+3*nu**3)*lam*lam/
     &           (2*(1-nu)**2)-(19-37*nu+19*nu*nu+nu**3)/
     &           (2*(1-nu)**2)-nu*nu*(n*n+lam*lam)/((1-nu)**2)
c
         delk1=(3+8*nu-5*nu*nu-nu**3)*lam*lam/(2*(1-nu))+
     &           (2+nu)*n*n/2-(6+4*nu-8*nu*nu+3*nu**3)*lam**4/
     &          (4*(1-nu))-nu*nu*(n*n+lam*lam)**3/(2*(1-nu))-
     &          (26-60*nu+40*nu*nu-3*nu**3-8*nu**4)*lam*lam*n*n/
     &          (2*(1-nu))-(13-22*nu+10*nu*nu)*n**4/(2*(1-nu))
c
        delk0=0.5*(1-nu)*((2+6*nu-2*nu*nu-3*nu**3)*lam**4/
     &         (2*(1-nu))+4*lam*lam*n*n+n**4-(1+nu)*lam**6/(1-nu)-
     &         (7-5*nu)*lam**4*n*n/(1-nu)-8*lam*lam*n**4-
     &          2*n**6)
c
c  Define cubic equation constants
c
        c2=k2+(h*h/(12*a*a))*delk2
        c1=k1+(h*h/(12*a*a))*delk1
        c0=k0+(h*h/(12*a*a))*delk0
c
c  Solve cubic equation for omega squared. Use method outlined
c  in CRC Standard Mathematical Tables, 23rd ed. page 105
c
        c=(1.0/3.0)*(3.0*c1-c2*c2)
        d=(1.0/27.0)*(-2.0*c2*c2*c2+9.0*c2*c1-27.0*c0)
c
        delta=(d*d/4.0)+(c*c*c/27.0)
c
        if (delta.LT.0.0) then
           partone=-d/2.0
           parttwo=SQRT(-1.0*delta)
           zP=cmplx(partone,parttwo)
           zQ=cmplx(partone,-parttwo)
        else
           partone=-d/2.0
           parttwo=SQRT(delta)
           zP=cmplx((partone+parttwo),0.0)
           zQ=cmplx((partone-parttwo),0.0)
        endif
c
        zP=zP**(1.0/3.0)
        zQ=zQ**(1.0/3.0)
c
        zfirst=-.5*(zP+zQ)
        zsecnd=-.5*(zP-zQ)*SQRT(3.0)*zi
c
        partone=real(zfirst)+real(zsecnd)
        parttwo=imag(zfirst)+imag(zsecnd)
        parttre=real(zfirst)-real(zsecnd)
        partfor=imag(zfirst)-imag(zsecnd)
c
        zroot1=zP+zQ+c2/3.0
        zroot2=cmplx((partone+c2/3.0),parttwo)
        zroot3=cmplx((parttre+c2/3.0),partfor)
c
        zfreq1=SQRT(zroot1*E/(rho*(1-nu*nu)*a*a))/(2.0*pi)
        zfreq2=SQRT(zroot2*E/(rho*(1-nu*nu)*a*a))/(2.0*pi)
        zfreq3=SQRT(zroot3*E/(rho*(1-nu*nu)*a*a))/(2.0*pi)
c
        write(10,1000)'Mode',m,',',n,'and frequencies',
     &                     zfreq1,zfreq2,zfreq3
c
 300  continue
 200  continue
c
      close(10)
c
 1000 format(a4,1x,i2,a1,i2,1x,a15,1x,3(f12.4,1x,f3.1,1x))
 1005 format(a37)
 1010 format(a22)
 1015 format(a4,2x,a1,2x,a1,1x,a14,3(5x,a11))
c
      stop
      end
```

Program forced2.f

```
      program forced2
c
c  Program to calculate natural frequencies of vibration for a
c  cylindrical shell and the forced response over a range of
c  frequencies.
c
c  No input files required. NASTRAN-generated punch file is optional
c  if comparison to a numceric theory is desired. Shell material
c  properties and geometric dimensions must be programmed prior to
c  compiling.
c
c  Output files:  freq.out contains mode numbers and natural
c                 frequencies.
c                 displace.out contains displacement information
c                 written in TECPLOT form.
c
c  Uses Donnell-Mustari theory and solution outlined in Leissa:
c  Vibration of Shells, NASA SP-288
c
c  Declarations
c
      implicit real (a-h,o-y)
      implicit complex (z)
      real l,nu,lam
c
      parameter (pi=3.141592654,maxf=1000,modes=10)
c
      dimension u(maxf),v(maxf),w(maxf),p(maxf),freq(maxf)
c
      logical print
c
c  Begin program
c
c  Cylindrical shell, dimensions, material properties (Al)
c
c  Length (in)
      l=50.0
c  Radius (in)
      a=10.0
c  Thickness (in)
      h=0.0625
c  Poission's ratio
      nu=0.334
c  Young's Modulus (psi)
      E=10.3E6
c  Density of the structure (slugs/in**3)
      rho=2.5383E-4
c  Density of the fluid (slugs/in**3)
      rhof=1.170e-7
c  Speed of sound in fluid (in/sec)
      co=13620.0
c  Damping coefficient
      eta=0.005
c  Applied force (lbs)
      Fo=-50.0
c  Square root of -1
      zi=cmplx(0.0,1.0)
c  Common coefficient (wave speed in structure)
      cl2=E/(a*a*rho*(1-nu*nu))
c
c  Frequency increment (Hz)
      step=5.0
c  Starting frequency
      frequency=5.0*(2.0*pi)
c  Print natural frequencies once
      print=.FALSE.
c
c  Determine location of interest
c
      rr=a/2
      tt=0.0
      yz=l/2
c
 1    format(a)
 2    format(f)
 3    format(a13,f4.1,1x,f4.1,1x,f4.1)
c
      tt=tt*(pi/180.0)
c
      if (print) then
         open(unit=10,file='freq.out',status='unknown')
         write(10,1005)'Cylindrical Shell Natural Frequencies'
         write(10,1010)'Donnell-Mustari Theory'
         write(10,1015)'Mode','m','n','Roots (Re,Im):',
     &                 'Root 1 (Hz)','Root 2 (Hz)','Root 3 (Hz)'
      endif
c
c  'Prep' tecplot output file
c
      open(unit=20,file='displace.out',status='unknown')
      write(20,1)'TITLE = "FORCED FLUID/STRUCTURE CYLINDER RESP."'
      write(20,1)'VARIABLES = "FREQ","U","V","W","P"'
      write(20,1)'ZONE T="Analytic"'
c
      do 100 k=1,maxf
         zu=cmplx(0.0,0.0)
         zv=cmplx(0.0,0.0)
         zw=cmplx(0.0,0.0)
         zp=cmplx(0.0,0.0)
      do 200 m=1,modes
      do 300 n=0,modes
c
         if (m.EQ.0.AND.n.EQ.0) goto 300
c
c  Calculate natural frequency for mode
c
         lam=(m*pi*a/l)
c
c  Define cubic equation constants
c
         c2=1.+.5*(3.-nu)*(n*n+lam*lam)+(h*h/(12.*a*a))*
     &     (n*n+lam*lam)*(n*n+lam*lam)
c
      c1=.5*(1-nu)*((3.+2.*nu)*lam*lam+n*n+(n*n+lam*lam)*
     &      (n*n+lam*lam)+((3.0-nu)/(1.0-nu))*(h*h/(12.*a*a))*
     &      (n*n+lam*lam)*(n*n+lam*lam)*(n*n+lam*lam))
c
      c0=.5*(1.-nu)*((1.-nu*nu)*(lam*lam*lam*lam)+
     &      (h*h/(12.*a*a))*(n*n+lam*lam)*(n*n+lam*lam)*
     &      (n*n+lam*lam)*(n*n+lam*lam))
c
c  Solve cubic equation for omega squared. Use method outlined
c  in CRC Standard Mathematical Tables, 23rd ed. page 105
c
      c=(1.0/3.0)*(3.0*c1-c2*c2)
      d=(1.0/27.0)*(-2.0*c2*c2*c2+9.0*c2*c1-27.0*c0)
c
      delta=(d*d/4.0)+(c*c*c/27.0)
c
      if (delta.LT.0.0) then
         partone=-d/2.0
         parttwo=SQRT(-1.0*delta)
         zPP=cmplx(partone,parttwo)
         zQQ=cmplx(partone,-parttwo)
      else
         partone=-d/2.0
         parttwo=SQRT(delta)
         zPP=cmplx((partone+parttwo),0.0)
         zQQ=cmplx((partone-parttwo),0.0)
      endif
c
      zPP=zPP**(1.0/3.0)
      zQQ=zQQ**(1.0/3.0)
c
      zfirst=-.5*(zPP+zQQ)
      zsecnd=-.5*(zPP-zQQ)*SQRT(3.0)*zi
c
      partone=real(zfirst)+real(zsecnd)
      parttwo=imag(zfirst)+imag(zsecnd)
      parttre=real(zfirst)-real(zsecnd)
      partfor=imag(zfirst)-imag(zsecnd)
c
      zroot1=zPP+zQQ+c2/3.0
      zroot2=cmplx((partone+c2/3.0),parttwo)
      zroot3=cmplx((parttre+c2/3.0),partfor)
c
      zfreq1=SQRT(zroot1*E/(rho*(1-nu*nu)*a*a))
      zfreq2=SQRT(zroot2*E/(rho*(1-nu*nu)*a*a))
      zfreq3=SQRT(zroot3*E/(rho*(1-nu*nu)*a*a))
c
      if (print) then
         write(10,1000)'Mode',m,',',n,'and frequencies',
     &      zfreq1/(2.*pi),zfreq2/(2.*pi),zfreq3/(2.*pi)
         if (m.EQ.modes) close(10)
      endif
c
      if (imag(zfreq1).NE.0.0.OR.
     &     imag(zfreq2).NE.0.0.OR.
     &     imag(zfreq3).NE.0.0) then
          write(*,*)'Fatal error: natural frequency is ',
     &               'imaginary'
          goto 9999
      endif
c
      call magnitude(zfreq1,freq1)
      call magnitude(zfreq2,freq2)
      call magnitude(zfreq3,freq3)
c
c  Calculate force coefficient
c
      lam=(m*pi/l)
c
      Fmn=a*(2.0/(pi*l))*Po*sin(m*pi/2.0)*(1.0+cos(n*pi))
c
      freqA=-SQRT(-(C12/(2.0*a*a))*(nu-1.0)*(nu*lam*lam*
     &               a*a-n*n)/nu)
      freqB=-SQRT(-(C12/(2.0*a*a))*(nu-1.0)*(nu*lam*lam*
     &               a*a+2.0*lam*lam*a*a+n*n))
      freqC1=SQRT((C12/a*a)*(lam*lam*a*a+n*n))
      freqC2=SQRT(-(C12/(2.0*a*a))*(lam*lam*a*a+n*n)*(nu-1.0))
c
c  Calculate cofactors for zAmn, zBmn, zCmn
c
      zdet=cmplx((frequency*frequency-freq3*freq3),
     &       2.0*eta*frequency*freq3)*
     &      cmplx((frequency*frequency-freq2*freq2),
     &       2.0*eta*frequency*freq2)*
     &      cmplx((frequency*frequency-freq1*freq1),
     &       2.0*eta*frequency*freq1)
c
      zcofA=cmplx((frequency*frequency-freqA*freqA),
     &              2.0*eta*frequency*freqA)
      zcofB=cmplx((frequency*frequency-freqB*freqB),
     &              2.0*eta*frequency*freqB)
      zcofC=cmplx((frequency*frequency-freqC1*freqC1),
     &              2.0*eta*frequency*freqC1)*
     &         cmplx((frequency*frequency-freqC2*freqC2),
     &              2.0*eta*frequency*freqC2)
c
      zAmn=-(Fmn/(rho*h))*zcofA/(zdet)
      zBmn= (Fmn/(rho*h))*zcofB/(zdet)
      zCmn=-(Fmn/(rho*h))*zcofC/(zdet)
c
c  Sum displacements
c
      zu=zu+zAmn*cos(m*pi*yz/l)*cos(n*tt)
      zv=zv+zBmn*sin(m*pi*yz/l)*sin(n*tt)
      zw=zw+zCmn*sin(m*pi*yz/l)*cos(n*tt)
c
c  Solve for pressure coefficient
c
      alphsqr=(frequency*frequency)/(co*co)-(m*pi/l)*(m*pi/l)
c
      if (alphsqr.LT.0.0) then
         alph=SQRT(-1.0*alphsqr)
         call cofimag(a,alph,n,bottom)
         zDmn=(zCmn*rhof*frequency*frequency)/bottom
         call bessimag(rr,alph,n,press)
c
      elseif (alphsqr.GT.0.0) then
         alph=SQRT(alphsqr)
         call realcof(a,alph,n,bottom)
         zDmn=(zCmn*rhof*frequency*frequency)/bottom
         call bessreal(rr,alph,n,press)
c
      elseif (alphsqr.EQ.0.0) then
         write(*,*)'Alpha = 0.0'
c
      endif
c
c  Sum pressure
c
      zp=zp+zDmn*press*sin(m*pi*yz/l)*cos(n*tt)
c
 300  continue
 200  continue
c
c  Dummy print
      write(*,4)'Complex data calculated for ',k
 4    format(a28,i4)
c
c  Convert complex displacements, pressure to magnitudes
c
      call magnitude(zu,u(k))
      call magnitude(zv,v(k))
      call magnitude(zw,w(k))
      call magnitude(zp,p(k))
c
c  Dummy print
      write(*,1105)zp
c
c  Write to TECPLOT file
c
      freq(k)=frequency/(2.0*pi)
      write(20,1100)freq(k),u(k),v(k),w(k),p(k)
c
c  Increment frequency
      frequency=frequency+step*(2.0*pi)
c
 100  continue
c
      close(20)
c
 1000 format(a4,1x,i2,a1,i2,1x,a15,1x,3(f12.4,1x,f3.1,1x))
 1005 format(a37)
 1010 format(a22)
 1015 format(a4,2x,a1,2x,a1,1x,a14,3(5x,a11))
 1100 format(f7.2,4(1x,e16.9))
 1105 format(2(e16.9,1x))
c
 9999 stop
      end
c
c  **************************************************************
      subroutine magnitude(z,x)
c
c  Subroutine to return the magnitude x of a complex argument z
c
      implicit real (a-h,o-y)
      implicit complex (z)
c
      partone=real(z)
      parttwo=imag(z)
c
      x=SQRT(partone*partone+parttwo*parttwo)
```

```
c
      return
      end
c
c ***************************************************************
      subroutine realcof(a,alph,n,out)
c
c Subroutine to return the denominator of the pressure coefficient
c term. Used when the argument of the n-th order bessel function
c is a real.
c
      implicit real (a-h,o-y)
      implicit complex (z)
c
c Square root of -1
      zi=cmplx(0.0,1.0)
c
      x=alph*a
c
      if (n.EQ.0) then
         term1=BESSJ0(x)
         term2=BESSJ1(x)
      elseif (n.eq.1) then
         term1=BESSJ1(x)
         term2=BESSJ(2,x)
      else
         term1=BESSJ(n,x)
         term2=BESSJ((n+1),x)
      endif
c
c Construct denominator
c
      out=(n/a)*term1-alph*term2
c
      return
      end
c
c ***************************************************************
      subroutine cofimag(a,alpha,n,out)
c
c Subroutine to return the denominator of the pressure coefficient
c term. Used when the argument of the n-th order bessel function is
c complex.
c
      implicit real (a-h,o-y)
      implicit complex (z)
c
      x=alpha*a
c
      if (n.EQ.0) then
         term1=BESSI0(x)
         term2=BESSI1(x)
       elseif (n.EQ.1) then
          term1=BESSI0(x)
          term2=BESSI(2,x)
       else
          term1=BESSI(n,x)
         term2=BESSI((n+1),x)
      endif
c
c Construct denominator
c
      out=alpha*term2+(n/a)*term1
c
      return
      end
c
c ***************************************************************
      subroutine bessreal(r,alpha,n,out)
c
c Subroutine to return the n-th order bessel function of a real
c argument.
c
      implicit real (a-h,o-y)
      implicit complex (z)
c
      zi=cmplx(0.0,1.0)
      x=r*alpha
c
      if (n.EQ.0) then
         out=BESSJ0(x)
      elseif (n.EQ.1) then
         out=BESSJ1(x)
      else
         out=BESSJ(n,x)
      endif
c
      return
      end
c
c ***************************************************************
      subroutine bessimag(r,alpha,n,out)
c
c Subroutine to return the n-th order modified bessel function of
c a real argument
c
      implicit real (a-h,o-y)
      implicit complex (z)
c
      x=r*alpha
c
      if (n.EQ.0) then
         out=BESSI0(x)
      elseif (n.EQ.1) then
         out=BESSI1(x)
      else
         out=BESSI(n,x)
      endif
c
      return
      end
```

Appendix C: NASTRAN File Useage Statistics

File Size (MB)		Analysis Type	Calculated Eigenvalues	Requested Frequencies	Nodes
DBALL	SCRATCH				
4.686	0.459	SOL 103	10	-	1331
19.759	5.947		10	-	4961
6.660	0.975		14	-	1573
27.394	8.004		14	-	5644
5.890	0.500		10	-	1449
24.478	6.554		10	-	6609
14.025	3.228		47	-	1953
61.538	17.564		51	-	8097
23.400	6.554	SOL 108	-	250	5644
24.396	7.406	SOL 111	20	250	5644
48.931	16.687	SOL 108	-	250	1953

Figure 31: MSC/NASTRAN file useage data for DBALL and SCRATCH files.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
January 1996

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
Fully-Coupled Fluid/Structure Vibration Analysis Using MSC/NASTRAN

5. FUNDING NUMBERS
WU 505-63-50-10

6. AUTHOR(S)
Christian M. Fernholz and Jay H. Robinson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA TM 110215

11. SUPPLEMENTARY NOTES
Fernholz: The George Washington University, Hampton, VA; Robinson: Langley Research Center, Hampton, VA. Research done by the first author, completed under Cooperative Agreement No. NCC1-14, is a portion of what will be included in a Master Of Science Thesis with The George Washington University.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 71

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

MSC/NASTRAN's performance in the solution of fully-coupled fluid/structure problems is evaluated. NASTRAN is used to perform normal modes (SOL 103) and forced-response analyses (SOL 108, 111) on cylindrical and cubic fluid/structure models. Bulk data file cards unique to the specification of a fluid element are discussed and analytic partially-coupled solutions are derived for each type of problem. These solutions are used to evaluate NASTRAN's solutions for accuracy. Appendices to this work include NASTRAN data presented in fringe plot form, FORTRAN source code listings written in support of this work, and NASTRAN data file usage requirements for each analysis.

14. SUBJECT TERMS
NASTRAN, Cylindrical Shells, Fluid/Structure Modeling, Finite Elements, Frequency Response, Structural Acoustics

15. NUMBER OF PAGES
85

16. PRICE CODE
A05

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT