Computation of acoustic scattering from elastic conical shells with endcaps using the hybrid finite element/ virtual source approach

by

Hwee Min Charles Low

B.Eng (1998)
National University of Singapore

Submitted to the Department of Ocean Engineering
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Ocean Engineering
At the
Massachusetts Institute of Technology
February 2004

Author...
Department of Ocean Engineering
May 7, 2004

Certified by...
Professor Henrik Schmidt
Department of Ocean Engineering
Thesis Supervisor

Accepted by...
Professor Michael S. Triantafyllou
Department of Ocean Engineering
Chair, Department Committee on Graduate Studies

ABSTRACT

Studying and understanding acoustic scattering pattern from underwater targets has been of interest to various communities such as the archeologists and the navy for several reasons and applications. The present state-of-the-art technique in this area involves such methods as analytical approach and FEM/BEM numerical technique. This thesis aims to study and demonstrate the power of using the hybrid virtual source/FE approach where the physical presence of a target is replaced by virtual sources placed in the vicinity of the target and in a manner where the pressure/displacement relationship on the target surface is satisfied by the virtual sources when the target is being insonified. Accurate results for the far-field radiation of the target can be obtained by superposition of the point source Green's function of each virtual source. The hybrid virtual source/FE approach shows potential to be a computationally efficient method for computing acoustic scattering.

The derivation of the dynamic flexibility matrix for an elastic conical shell with endcaps will be illustrated in this thesis. It will be shown that the dynamic flexibility matrix corresponds to the acoustic admittance matrix in the virtual source approach where the scattering functions are computed in the MIT's program OASES/SCATT. Moreover, the benchmarking and validation of the approach will be conducted with the hybrid analytical/ virtual source approach. Firstly, the approach predicts natural frequencies close to the theoretical analysis for higher order modes with more than 2 circumferential transverse vibration lobes. Secondly, it produces displacement profile that conforms to analytical results. The scattering functions are also in agreement those computed by the hybrid analytical/ virtual source approach, with discrepancies observed at lower frequencies. In exact terms, discrepancies start to appear for frequency in the range of 1000 to 2000 Hz for a 0.01m thick, 2 m long, 0.3m radius steel cylinder without endcaps.

The scattering functions will be compared with the SCATT/OASES virtual source approach for pressure release and rigid cylinders and cones. For the hybrid FE/virtual source approach, the structural sound speed and density approach zero and infinity for pressure-release and rigid target respectively. On the other hand, in the SCATT/OASES virtual source approach, the pressure and displacement are required to vanish on the target surface respectively. It will be shown that the two approaches agree with each other.

Moreover, scattering functions over steel cones and cylinders for various frequencies have also been derived in this research. The results will be interpreted physically and theoretically in this thesis. The importance of including structural damping in the finite element formulation of the target so as to reflect the effect of resonance on scattering will be illustrated. Other issues, such as effect of target orientations on scattering, will also be investigated in this thesis.

The code has shown good potential for adaptation to compute scattering over other axi-symmteric shapes using conical shells and circular plates as building blocks and the hybrid FE/ virtual source approach.

ACKNOWLEDGEMENT

One of my most valuable rewards of coming to MIT is to stretch my modest intellect to a limit far beyond what I could imagine myself doing before coming.

One reason this has been possible is that I have the precious opportunity of working with my advisor, Henrik Schmidt, who is one of the most eminent scholars in the field of ocean acoustics. I would thus like to thank him for his guidance and patience in my research.

My second home in Boston (thankfully not my first, though) is my office in 5-007. Life here in MIT would not be as valuable and academically rewarding if without the company of the people in 5-007: Bertrand, Wenyu Luo, Mark Rapo, Hun Joe Kim, Tai Wei Wang, who are all both my colleagues and friends.

I would also like to thank my other friends in MIT and Boston. I have the rare opportunity of staying in graduate apartment, Tang hall 19D for my entire two years here and would like to thank those who have stayed with me for their friendship: Erico Guizzo, Fabio Rabbani and many others. My special thanks to Lisa Kang and other close friends from Boston Chinese Evangelical Church (BCEC).

My program here would not have been possible without the support of DSO National Laboratories. My sincere thanks to Tong Boon Quek and Frank Teo for their support of my postgraduate studies and Joo Thiam Goh who have sparked my interest in ocean acoustics and for guidance in deciding what I want to pursue in MIT.

Last but not least, I would like to thank my dear family, though physically absence during my two years away from home, have given so much support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

English Alphabets:

a	Outer radius of annular plate element
b	Inner radius of annular plate element
c	Sound speed
c_p	Compressional sound speed
D	Directivity function
d	Plate rigidity
d_1	Spacing of virtual sources
d_2	Distance of virtual sources from target surface
E	Young's Modulus
h	Plate/ Shell thickness
k	Wavenumber ($\frac{\omega}{c}$)
L,l	Length of cylinder, cone/ conical shell elements
n	Number of circumferential lobes
M	Moment
p	Number of nodes in 2D FE forumulation
P_i, P_s, P_t	Incident, scattered and total pressure respectively
q	Number of circumferential steps in FE discretization
Q	Shear
r	Radius/ radial coordinate/ radial distance from origin
s	Source strength
t	Time

u	Deflection in axial direction of conical shell
V	Volume
v	Deflection in circumferential direction of conical shell
w	Deflection in radial direction of conical shell
W	Work done
U_i, U_s, U_t	Incident, scattered and total displacement respectively
x, y	Coordinate axis on the horizontal plane
z	Vertical axis/ Conical shell axial coordinate

Greek symbols:

β	Normalized thickness
ψ_i, ψ_s, ψ_t	Incident, scattered and total displacement potential respectively
λ	Wavelength
ρ_s	Density of target structure
Ω	Normalized frequency
ω	Frequency
Φ	Shape function
φ	Scattering angle / Structural nodal rotation
θ	Circumferential/ azimuthal angle
ξ	Non-parameterized length ($\frac{z}{l}$ in axial direction of conical shell)
σ	Stress
ε	Strain
υ	Poisson's ratio

Column vectors:

{s}	Virtual source strength vector

Matrices:

[A]	Admittance
[B]	Strain
[D]	Rigidity
$[G_\psi]$, $[G_u]$	Displacement potential/ displacement Green's function
[K]	Static stiffness
[N]	Shape function
[Q]	Nodal forces to elemental pressure relation

1 INTRODUCTION

1.1 Background

The ability to detect and classify underwater targets has been of interest to various communities. For example, undersea archaeologists apply this technique to recover the remains of material evidence and artifacts. The military also has an interest in this for purpose of undersea warfare involving submarines, torpedoes and mines.

For most of these applications, it is important to be able to interpret and analyze sonar signals reflected from the targets. For example, most recent technology for real-time detection of underwater targets involve bistatic synthetic aperture sonar (SAS) where autonomous underwater vehicles (AUVs) are deployed for maneuvering and collecting scattered signals from targets. Such targets could be static such as sea mines or moving such as torpedoes and submarines. It is crucial to interpret the signals accurately to classify the targets for such purposes as real-time decision making in cases of undersea warfare or post-processing for undersea archaeology. Other examples where signal interpretation is involved include undersea observatories where a real-time network involving several multi-array sonars and hydrophones is deployed for multi-static detection and classification.

The physics behind the scattering from targets is itself a complicated issue. In order to interpret and analyze the scattered signals from the targets, it is important to first understand the scattering function and strength of basic target shapes with common material properties, including spheres, cylinders and cones. Firstly, this will aid in the preliminary classification of targets under this broad category. Moreover, such understanding shows potential for evolving into prediction for more complicated shapes and other material properties, especially when robust scattering patterns have been determined. Extensive research has been conducted to understand the response of underwater objects such as cylindrical and spherical shells when subjected to acoustic wave excitation. The methods adopted can be broadly classified under

(i) analytical and

(ii) numerical techniques.

Closed form analytical solutions are available for the response of elastic spheres and cylinders under dynamic loading. For more complicated shapes, numerical techniques have to be employed to determine the response. The solutions are then coupled with computational ocean acoustic methods to derive the incident pressure on these objects from source and the re-radiation to receivers in the waveguide. For more simplified waveguides, such as the Pekeris waveguide and non-stratified pressure-release surface – rigid bottom, closed form solutions for computing the acoustic propagation are available.

Current research in this area evolves primarily from the need to optimize solution techniques and accuracy. The general trend at present is to use numerical techniques to compute radiation from complicated shapes. Since numerical methods are more computationally expensive and time-consuming than analytical techniques, it is necessary to find optimum methods for the wide spectrum of problems without compromising accuracy.

1.2 Literature Review

1.2.1 Analytical Methods

Junger and Feit [1] formulated the analytical solution for dynamic response of spherical and cylindrical shells under acoustic pressure loading using elastic wave theory. Using Hamilton's principle and thin shell theories, the equations of motion for spherical and cylindrical shells are first derived in the shells' local coordinates. The response of the shells is then determined analytically from the partial differential equations (PDFs), with reference to the incident acoustic pressure coming from the far-field. The response of the shells can then be combined with the free-space Green's function to compute the far-field radiation of the shells, e.g. by applying large-argument asymptotic expansions. In cases of rigid and pressure release spheres and cylinders, which approximate the real world of elastic shells, the pressure and displacement potential relation is obtained by applying the relevant Dirichlet or Neumann boundary conditions on the target surface. The response is then used to derive the far-field radiation. Junger and Feit applied the above methods mainly for simply supported or infinite cylindrical shells. They have also treated the problem of fluid-filled shells by using the method of effective spring and mass and included considerations such as damping.

Williams [2] presented another analytical concept for computing scattering on the basis of Fourier acoustics. In this approach, the surface deformation and normal pressure of the target is represented in K-space using Fourier transformation and propagated to the farfield using Green's function. The actual pressure is then determined using inverse Fourier transform. Other important tools were also introduced, such as Ewald sphere construction, Rayleigh's integral, plate radiation and supersonic intensity.

1.2.2 Numerical methods

Gordon et al [3] presented numerical techniques for 3-dimensional steady-state fluid-structure interactions. In particular, the coupled finite element/ boundary element approach was discussed. The surface fluid pressures and normal velocities were calculated by coupling the Finite Element (FE) model of the underwater structure with a discretized form of the Helmholtz surface integral equation of the exterior fluid. Farfield radiation pressures are then calculated from the surface solution using the Helmholtz exterior integral equation. The approach was successfully validated using known analytical solutions for submerged spheres subjected to both incident pressure and uniform and non-uniform applied mechanical loads.

Burnett and Zampolli [4, 5] also developed the Finite Element for Structural Acoustic (FESTA) tool for modeling the response of the arbitrarily shaped targets under acoustic pressure loading. The multi-static scattering behavior of single and multiple fluid-loaded elastic targets can thus be derived. The target and its surrounding media are discretized using finite elements to study the overall response. The h-p adaptive finite-element technology allows for optimization of convergence and computing time. This is done by adjusting to an optimum the element size (h-refinement) and polynomial order of selected elements (p-enrichment).

1.2.3 Computational Acoustic Method

Jensen et al [6] presented various computational acoustic tools for studying the acoustic propagation in the ocean waveguide. Among the tools is the wavenumber integration method, an upgraded version of SAFARI [7], by Schmidt. The wavenumber integration method is based on solving for the depth-dependent Green's function for horizontally stratified media using a Direct Global Matrix (DGM) solution or Propagator Matrix approach. By implementing the Fast-Fourier Transform (FFT) technique in the method, the computational efficiency is greatly enhanced as the far-field radiation can be computed taking advantage of the asymptotic behavior of the inverse Hankel transform. Other propagation models introduced include the normal mode and the parabolic equation approaches.

With the methods and tools established, Schmidt developed OASES [8] wavenumber integration code and C-SNAP, which implements the range-dependant normal mode approach. Porter [9] also implemented the range-independent normal mode approach in the KRAKEN code. With these tools in place, the acoustic propagation can be predicted accurately in complex environments, including range-dependant waveguide, e.g. around sea-mounts. This allows sonar signals to be predicted accurately for purpose of target detection and classification.

Zampolli et al [10] coupled the FESTA solution with OASES to allow the modeling of scattering in the ocean waveguide. Due to the computational cost, FESTA is normally implemented in a domain containing the target and the surrounding medium not larger than a few, at most a few tens of wavelengths. By coupling with OASES, the scattered pressure in the near-field is treated as the boundary conditions to OASES for the rest of the computation to be carried out by wavenumber integration. The proportion of computational time is normally a few hours for the FESTA step and a few minutes only for OASES.

1.2.4 Virtual Source Approach

The utilization of the virtual source approach to solve radiation and scattering problems is well-known. One recent example is the solution to scattering of 3-dimensional sources by rigid barriers in the vicinity of tall buildings. Godinho et al [11] solved for the response of the virtual sources equally spaced on the vertical axis of the tall building at different spatial wavenumbers. The summation of the responses from individual sources and subsequent spatial Fourier transformation then yields the pressure radiation in the spatial domain.

Of relevance to the underwater acoustics problems, Schmidt [12] implemented the virtual source approach to determine the target scattering in an insonified ocean waveguide. With this approach, the finite element domain is reduced to the target structure itself. The target is then replaced by point sources that virtually represent the target by having a consistent displacement-displacement potential relation in the surrounding fluid at locations on the target surface as dynamic stiffness/ flexibility representation. The computational time by this method is typically in the order of a few minutes once the dynamic stiffness/ flexibility matrix of the target is available. This method has proven to have great potential to derive the scattering in a most computationally efficient fashion without compromising accuracy.

1.2.5 FE Modeling of axi-symmetrical shells and plates

The FE modeling method is widely used to solve dynamic problems involving complex shapes and geometry. The computational efficiency of axi-symmetric objects can be greatly improved by simplifying the model and using 2-dimensional elements to represent the object. Ross [13, 14] conducted analysis of this form on FE modeling of cylinders and conical shells, primarily using the Principle of Virtual Work. By using Fourier expansion to represent the excitation force, moment, displacement and rotation at the nodes of the FE model, the shells are replaced by 2-dimensional elements, much like those that represent beams, except for the additional degrees-of-freedom (DOFs) due to circumferential displacement and differences in the dynamic stiffness/ flexibility representation. Numerical solutions using FE are superior for more complicated problems, such as vessels with varying thickness and complex boundary conditions.

Pardoen [15] adopted a similar approach to analyze the static behavior, vibration and buckling of axi-symmetric circular plates. The 2-dimensional plate elements have 2 DOFs, transverse displacement and rotation per node. The dynamic stiffness formulation was presented in the paper. Each element represents an annular plate by itself. The FE formulation was implemented on a slightly different basis than what Ross performed. Pardeon used Hermitian interpolation polynomials as shape functions for the elemental displacement in terms of nodal displacement. Moreover, the static stiffness matrix was derived using plate theories by Timoshenko [13] to predict the static nodal force vector associated with the degrees of freedom on the plate. In comparison with the exact analytical solution, it was demonstrated that the FE model was able to predict natural frequencies for a clamped-clamped annular plate at a discrepancy not more than 0.1%.

1.3 Motivation

Currently, the various techniques for computing acoustic scattering from underwater targets can be categorized and summarized using Fig. 1.1. The boxes in green indicate focus of the research discussed in this thesis.

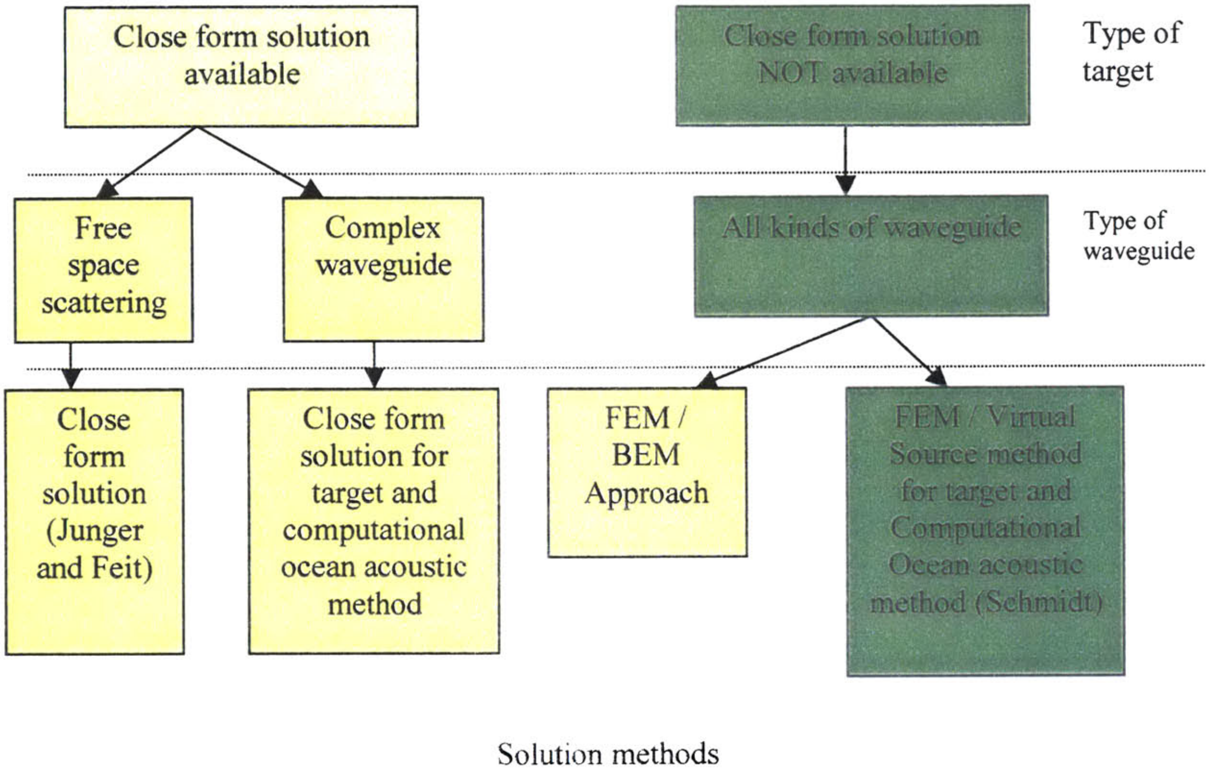

Close form solution available
Close form solution NOT available
Type of target
Free space scattering
Complex waveguide
All kinds of waveguide
Type of waveguide
Close form solution (Junger and Feit)
Close form solution for target and computational ocean acoustic method
FEM / BEM Approach
FEM / Virtual Source method for target and Computational Ocean acoustic method (Schmidt)
Solution methods

Figure 1-1 A summary of various techniques for computing acoustic scattering

In the real world, there are many objects that exhibit axial symmetry. Such objects can be simplified by a 2-dimensional representation by taking a cross-sectional view of the objects on the plane where the axis lies on. The dynamic response of the target can then be computed by FE analysis of the 2-dimensional model by discretizing the object using appropriate elements. With the advent of computational acoustic tools over the last 2 decades, acoustic propagation in the ocean waveguide can be accurately modeled. In particular, with the virtual source approach being proven in various areas, it has also shown potential to be applied in underwater acoustics. With these advancement, it is thus of increasing interest to study the acoustic scattering from underwater targets using these tools. The research conducted and described in this thesis is motivated by this.

The research aims to combine the tools available i.e. axi-symmetric FE modeling, virtual source approach and computational acoustics to predict and prove the method's accuracy in the prediction of sound radiation from underwater targets. In particular, this research is geared with MCM (Mine-counter measures) applications in mind. Thus, elastic conical shells with endcaps are the objects of interest. Moreover, most axi-symmetric objects can be approximated using these conical shells and endcaps as building blocks. The research aims to study and derive the method as a useful tool with accuracy and computational efficiency.

1.4 Research Objectives

The main objective of this research is to implement the FE method for elastic axi-symmetric conical shells with endcaps to derive the dynamic flexibility matrix for subsequent computation of acoustic scattering by the virtual source approach using the SCATT/OASES program by MIT. In the virtual source approach, the dynamic flexibility matrix relates the displacement to the pressure on the target surface when the target is replaced by the virtual sources. The dynamic flexibility matrix thus forms the acoustic boundary conditions for deriving the target scattering in the waveguide.

The FE code was written based on papers by Ross and Pardeon [13, 14, 15] to compute acoustic boundary conditions for a conical shell with flat circular plates at both ends, using 2-dimensional elements to discretize the target. The thickness and material properties could be independently defined for the endcaps and the conical shell. The thickness is assumed to be uniform for individual sections of the target. Verification of the hybrid FE/ virtual source approach was then conducted using the following approaches:

a. Derivation of the acoustic boundary conditions for an elastic cylinder with pressure release endcaps using an analytical method as outlined by Junger and Feit [1]. Similar boundary conditions were being computed using the FE code developed here. The scattering predicted using the two sets of boundary conditions were then computed and compared.

b. Computation of the acoustic scattering from rigid and pressure release cones of various geometry with endcaps using the hybrid FE/ virtual source approach. Subsequently, the scattering functions were computed by requiring the normal displacement and pressure on the surface to vanish for the rigid and pressure release cases, respectively. The results were then compared and checked for agreement.

c. Computation of acoustic scattering from elastic cylinders with various orientations using the hybrid FE/ virtual source approach and analysis of results with physical interpretation.

d. Computation of acoustic scattering from elastic cylinders at resonant frequencies and off-resonant frequencies and comparison of results with theoretical prediction.

With the above validation and comparison completed, the code was then used to compute the boundary conditions for elastic conical shells and the acoustic scattering was determined.

1.5 Thesis Outline

This thesis consists of 5 chapters. The present chapter included background of the research, a review of current relevant literature, description of motivation and research objectives.

In chapter 2, the theory behind the hybrid FE/ virtual source approach will be described. The method to replace the target in the waveguide by virtual internal sources and derivation of the acoustic boundary conditions will first be described. The computation of scattering by method of superposition of the Green's function of the virtual sources will then be discussed, followed by the FE approach to derive the dynamic flexibility matrices of conical shells with endcaps. The efficiency of 2-dimensional representation of the target and subsequent implementation of Fourier series expansion to compute the vibration response of the shells and plates in 3-dimensional space will be demonstrated. Another numerical approach to be described in this chapter is the computational acoustic method to compute the incident wavefield on the target and the radiation of the target to the far-field. The various established tools available include the wavenumber integration and normal mode approach. In particular, the MIT code, OASES and SCATT, which implements the wave number integration approach was the primary tool in this research.

In Chapter 3, the scattering computational results for an elastic cylinder with or without endcaps will be used for benchmarking of the FE code. The analytical method of computing the acoustic response of the shell will first be discussed. The analytical prediction of resonant frequencies of shells will then be presented and compared with those estimated by the FE approach. The acoustic scattering from an elastic cylinder with simply supported end conditions will also be computed and compared for the two approaches using the analytical approach and the FE method respectively to derive the acoustic boundary conditions. The scattering from rigid and pressure release cylinder with endcaps can also be computed by requiring the normal displacement and pressure on the surface to vanish respectively. The results will be compared with the scattering computed by the hybrid FE/ virtual source approach for very stiff and flexible target respectively. Other issues, such as effect of target orientations and shell resonance on scattering, will also be discussed in this chapter.

In Chapter 4, the results for acoustic scattering from conical shells with endcaps as computed by the hybrid FE/ virtual source approach will be presented. Scattering from pressure-release and rigid conical shells will be computed for benchmarking and validation.

In the final chapter, the summary and conclusion of the thesis will be presented. Future work will be suggested such as extending the current model to compute the dynamic flexibility matrix for other arbitrarily shaped axi-symmetric elastic shells using conical shell elements of varying angles and plates as building blocks. The acoustic scattering problem for these shapes can thus be subsequently treated using the virtual source approach.

2 THEORY

2.1 Virtual Source Approach

The virtual source approach is illustrated in Fig. 2.1. The physical presence of the target in the waveguide is replaced by an interior distribution of point virtual sources.

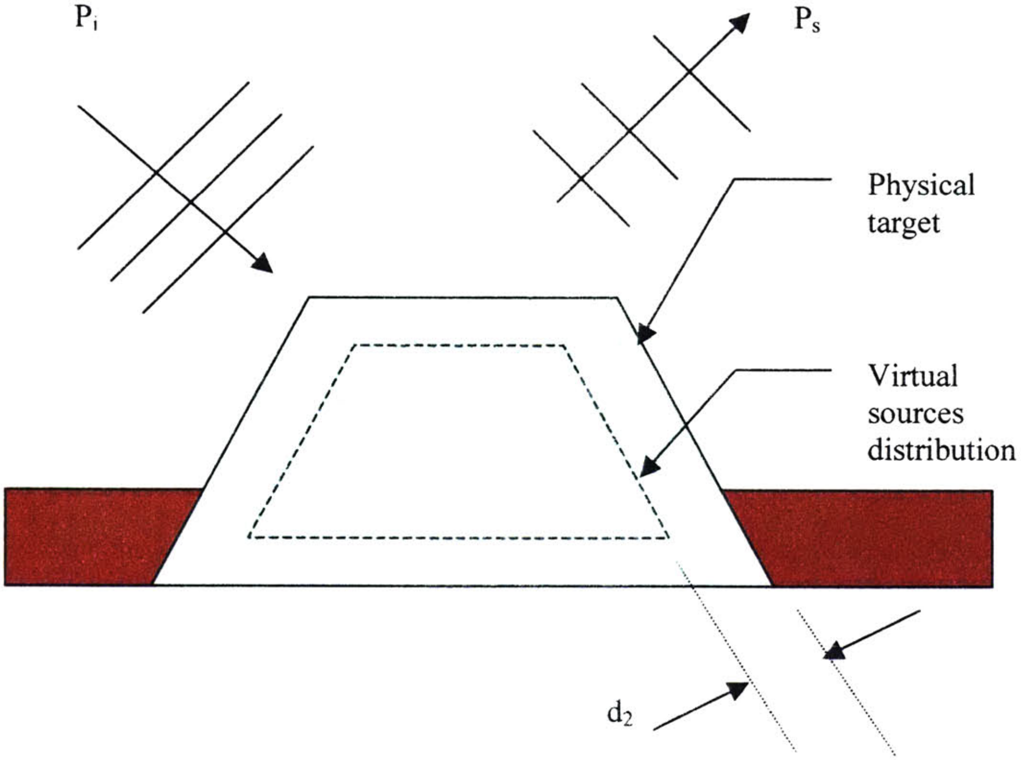


Figure 2-1 Virtual source representation of target

The criteria for placing the virtual sources in this research are adopted from Schmidt [12]

$$\begin{aligned} d_1 &\leq \frac{\lambda}{4} \\ d_2 &= 0.6 d_1 \end{aligned} \quad , \qquad (2.1)$$

where d_1 : Spacing of virtual sources distributed on the target surface, λ : Wavelength of the impinging acoustic waves in surrounding fluid,

d_2 : Distance of virtual sources from target surface along surface normal.

The incident pressure from the source is denoted $P_i(x, y, z)$ with the associated displacement $U_i(x,y,z)$ where x, y, z are the coordinates in the 3-dimensional space domain. The scattered pressure is $P_s(x, y, z)$ with displacement $U_s(x,y,z)$. The total pressure and displacement on the target surface is thus $P_t(x, y, z)$ and $U_t(x,y,z)$ respectively which are the superposition of the incident and scattered fields

$$P_t = P_i + P_s \, , \tag{2.2a}$$

$$U_t = U_i + U_s \, . \tag{2.2b}$$

The pressure on the target surface is expressed in terms of the displacement potential, ψ,

$$P_t = -\rho_s \frac{d^2\psi}{dt^2} = \rho_s \omega^2 \psi_t \, , \tag{2.3}$$

where ρ_s: Density of target structure,

t : Time,

ω : Radian frequency.

The relation between the total displacement potential and total pressure at the surface of the target is determined from the boundary conditions, and may be expressed in terms of the admittance matrix, [A],

$$U_t = [A] \bullet \psi_t \, . \tag{2.4}$$

By treating each individual virtual source as a point source, the source strength of the sources can be represented by a column vector {s}. Moreover, by writing the displacement potential and displacement at discrete surface points, or nodes, as vectors, the relations between source strength, scattered field and displacement are

$$\{\psi_s\} = [G_\psi] \bullet \{s\}, \tag{2.5a}$$

$$\{U_s\} = [G_u] \bullet \{s\}, \tag{2.5b}$$

where $[G_\psi]$, $[G_u]$ are the matrices of Green's function relating the displacement potential and displacement field at the nodal locations on target surface due to each individual virtual source.

Combination of equations (2.2) to (2.5) leads to an expression for the source strength vector {s},

$$\{s\} = [[G_\psi] - [A] \bullet [G_u]]^{-1} \bullet [[A] \bullet \{U_i\} - \{\psi_i\}]. \tag{2.6}$$

The virtual sources {s} ensure that the total displacement and displacement potential field on the target surface is consistent with the actual field generated by the physical presence of the target. The scattered field due to the virtual source representation is thus identical to that of actual target.

2.1.1 Green's Function Matrix

Assuming spherical coordinates in an unbounded medium, the Helmholtz equation governing the acoustic propagation in an infinite medium is

$$\left[\frac{1}{r^2}\frac{\partial}{\partial r}r^2\frac{\partial}{\partial r} + k^2\right]\psi(r) = 0. \tag{2.7}$$

By applying the radiation condition of no incoming waves from infinity, the displacement potential and the associated displacement for a point source can then be expressed as

$$\psi(r) = -S\frac{e^{ikr}}{4\pi r}, \tag{2.8a}$$

$$U(r) = -S\frac{e^{ikr}}{4\pi}(\frac{ik}{r} - \frac{1}{r^2}), \tag{2.8b}$$

where S : Source Strength of the virtual point source,

k : Wavenumber (Division of frequency (rad/s) by sound speed (m/s) of surrounding fluid,

r : Radial distance from source.

Thus, the ith-row, jth-column component of the $[G_\psi]$, $[G_u]$ can be expressed as functions of the distance of the virtual source to the nodal location on the target surface by

$$[G_\psi]_{ij} = -\frac{e^{ik|r_i - r_j|}}{4\pi\,|r_i - r_j|} \quad , \tag{2.9a}$$

$$[G_u]_{ij} = -\frac{e^{ik|r_i - r_j|}}{4\pi}(\frac{ik}{|r_i - r_j|} - \frac{1}{|r_i - r_j|^2}), \tag{2.9b}$$

where r_i and r_j represent the radial distance from a fixed origin of the ith-node on target surface and jth-virtual point source respectively. The above expressions are valid for all cases where the target is positioned within a localized homogenous medium, such as those within an ocean waveguide or those which are fully buried in a lower halfspace

For the case of targets half-buried in the seafloor, the treatment is slightly more complicated in the sense that the stratified Green's function has to be applied. Thus, the Green's function has to derived based on integral transform solution by application to point source in a fluid halfspace and ensuring satisfaction of continuity across interface [12].

2.1.2 Wavefield Superposition

Once the source strength vector *{S}* has been determined, the acoustic propagation can be computed by linear superposition. In the case of unbounded medium, the wavefield is simply the summation of the source strengths multiplied by the free-field Green's functions. For stratified waveguide, computational ocean acoustics techniques have to applied as described in section 2.3.

2.2 Acoustic Boundary Conditions for axi-symmetric elastic target

2.2.1 Dynamic Stiffness Matrix of Conical Shells in the 2D domain

Figure 2.2 illustrates the parameters of the conical shell and the discretizing elements.

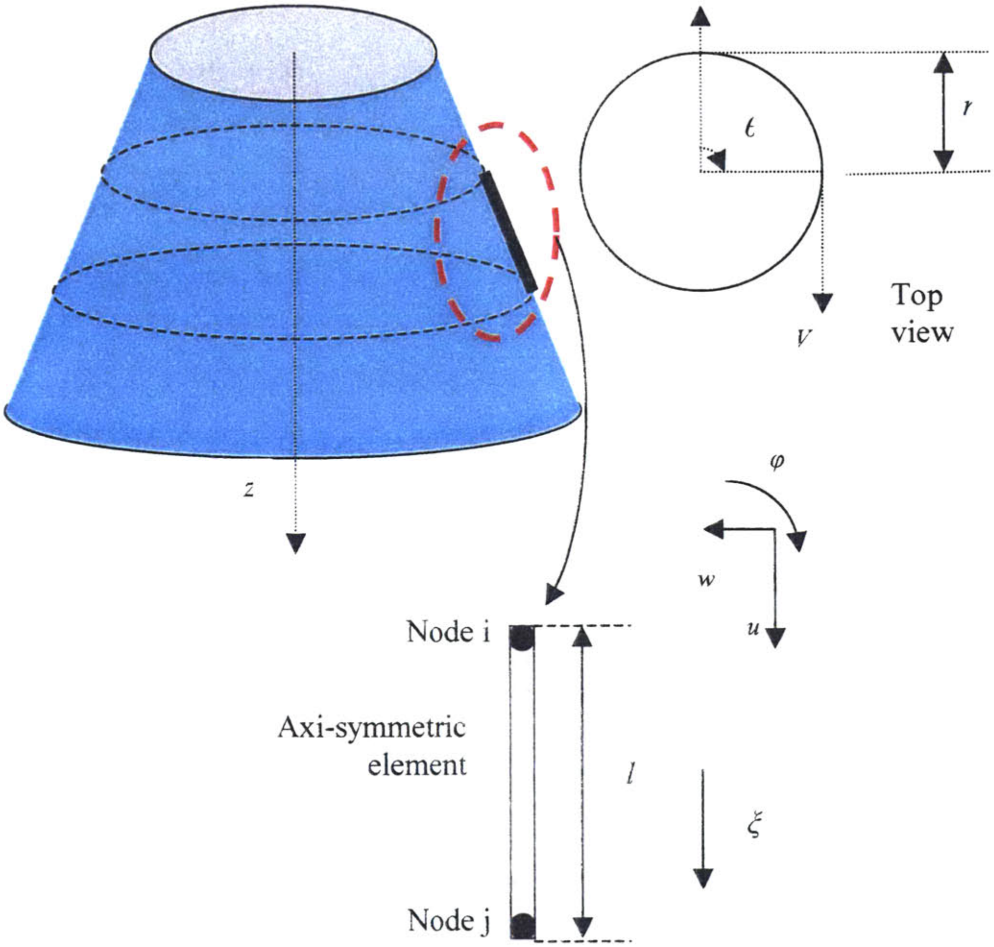


Figure 2-2 Conical shell element

The elements have 2 nodes with 4 degree-of-freedom (DOF) at each node, 3 translational and 1 rotational. The stiffness matrix for each element was derived by Ross [13],

$$[K(n)]_{cone} = \int\int [B(n)]^T [D][B(n)] r d\theta dx . \qquad (2.10)$$

The interpolation matrix [B(n)] relates the strain on the conical shell to the nodal displacement and rotation while [D] relates the conical shell stresses to its strain components. The formulation is a result of the application of the principle of virtual work on the potential strain energy of the conical shell element. n is the order of a Fourier expansion in azimuth.

Ross also derived the mass matrix in [14]

$$[M] = \rho_s h \int\int [N]^T [N] r d\theta dx , \qquad (2.11)$$

where h: Thickness of conical shell,

[N]: Shape functions relating the elemental displacement to nodal displacement. [N] was derived on the basis of shape functions of Fourier order n,

$$u = [u_i(1-\xi) + u_j \xi] \cos n\theta e^{i\omega t} , \qquad (2.12a)$$

$$v = [v_i(1-\xi) + v_j \xi] \sin n\theta e^{i\omega t} , \qquad (2.12b)$$

$$w = [w_i(1-3\xi^2+2\zeta^3) + \varphi_i l(\xi - 2\zeta^2 + \xi^{3)} + w_j(3\xi^2 - 2\zeta^3) + \varphi_j l(-\xi^2 + \xi^{3)}] \cos n\theta e^{i\omega t} .$$

$$(2.12c)$$

In formulating the equation of motion of cylindrical shells, Junger and Feit [1] in addition included contribution of rotary inertia in the kinetic energy (K.E.) integral formulation,

$$K.E. = \frac{1}{2}\rho_s \left(\frac{h^3}{12}\right) \iint \left[\left(\frac{\partial \dot{w}}{\partial z}\right)^2 + \frac{1}{r^2}\left(\frac{\partial \dot{w}}{\partial \theta} - \dot{v}\right)^2 \right] r d\theta dz . \qquad (2.13)$$

In matrix form, the equation can be expressed in terms of the shape functions to yield

$$[M(n)]_{rotary} = \frac{\rho_s h^3}{12} \pi l \int_0^1 \left[\frac{\partial \dot{w}}{\partial z} \quad \frac{1}{r}\left(\frac{\partial \dot{w}}{\partial \theta} - \dot{v} \right) \right] \left[\frac{\partial \dot{w}}{\partial z} \quad \frac{1}{r}\left(\frac{\partial \dot{w}}{\partial \theta} - \dot{v} \right) \right]^T r d\zeta \,. \tag{2.14}$$

Thus, the mass matrix $[\mathrm{M(n)}]_{\mathrm{cone}}$ for conical shell can be represented as the summation of the contribution due to mass moment of inertia and rotary inertia,

$$[M(n)]_{cone} = [M] + [M(n)]_{rotary} \,. \tag{2.15}$$

By assuming that the displacements of the conical shell are to be represented by above equations and applying the principle of virtual work (which states that if the elastic body under a system of external forces (including D'Alembert's forces) is given a small virtual displacement, the net increase in work done by the forces is equal to the increase in strain energy), the dynamic stiffness matrix of the conical shell can be derived,

$$[K(n)]_{cone} - \omega^2 [M(n)]_{cone} = 0 \,. \tag{2.16}$$

2.2.2 Dynamic Stiffness Matrix of Circular Plates in 2D domain

The element needed to discretize a circular plate is as shown in Figure 2-3. The axi-symmetric element contains 2 DOF at each node, 1 translational and 1 rotational. The shape functions that describe the relation between elemental displacement and nodal displacement without circumferential lobes were described by Pardoen in [15]. The expressions can be extended to include the effect of the number of circumferential lobes when the shell vibrates in an acoustic field. The relation between elemental displacement, shape functions, nodal displacement and number of circumferential lobes to be used in this present study is

$$w(r,\theta) = [\Phi_1(r) w_1 + \Phi_2(r)\varphi_1 + \Phi_3(r) w_2 + \Phi_4(r)\varphi_2]\cos(n\theta) \,, \tag{2.17}$$

where Φs are the shape functions and subscripts indicate the following:

1 : Displacement at node i,

2 : Rotation at node i,

3 : Displacement at node j,

4 : Rotation at node j.

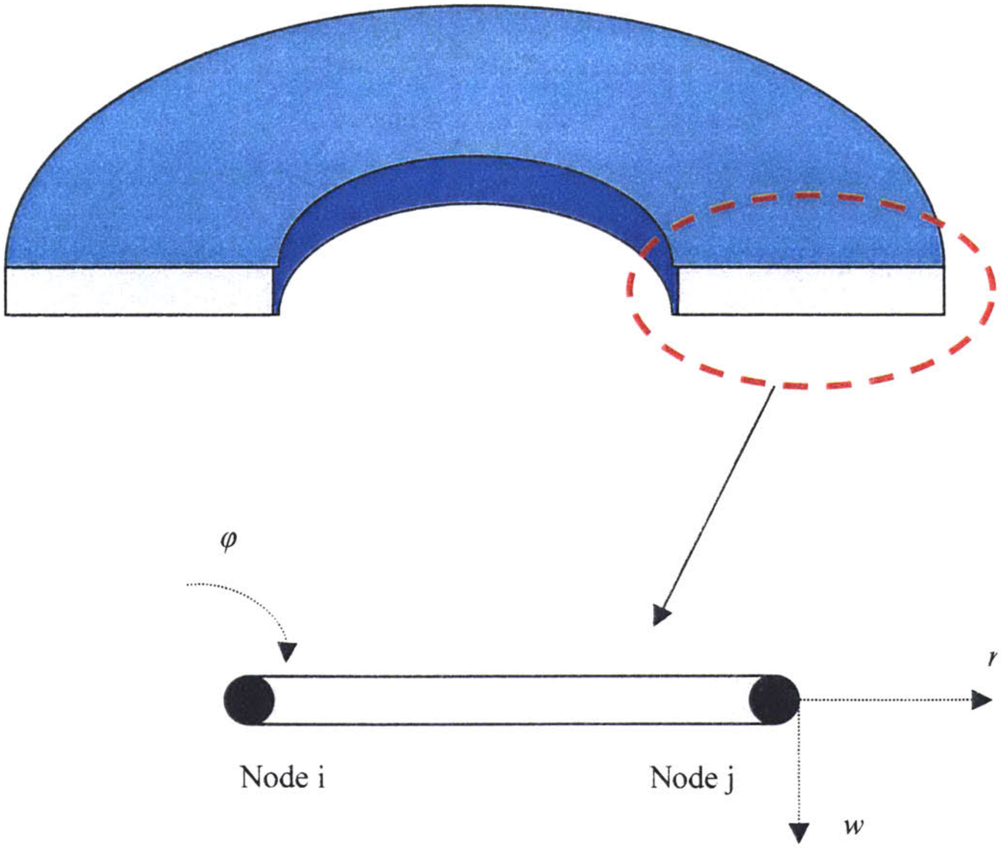


Figure 2-3 Elements to discretize circular plates and related DOFs and coordinate axis

Based on plate theory [16], the equations that relate the moments and shear forces on an element to the elemental lateral displacement are

$$M = d\left[\frac{\partial^2 w}{\partial r^2} + \mu(\frac{1}{r}\frac{\partial w}{\partial r} + \frac{1}{r^2}\frac{\partial^2 w}{\partial \theta^2}\right], \tag{2.18}$$

$$Q = d\left\{\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial w}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 w}{\partial \theta^2}\right]\right\}, \tag{2.19}$$

where d : plate rigidity.

To derive the elemental stiffness matrix, the procedures as outlined by Pardoen [15], together with equations (2.17) to (2.19), were adopted. In principle, K_{ij}th term in $[K(n)]_{pl}$ was derived by calculating the load required at DOF *i* to result in a unit displacement at DOF *j*. The elemental stiffness matrix is thus a 4 x 4 matrix.

Pardoen [15] derived the mass matrix, by direct substitution of the shape functions into the element energy integral,

$$[M]_{pl} = \rho_s h \int\int [N]^T [N] r dr d\theta, \tag{2.20}$$

Where $N = [\Phi_1 \ \ \Phi_2 \ \ \Phi_3 \ \ \Phi_4]$.

2.2.3 Global Flexibility Matrix for conical shells with endcaps

The global stiffness and mass matrices were assembled using a standard FE method. The nodes were numbered top-down, starting from the centre of the top circular plate for a standing cylinder. The assembly is illustrated in Figure 2.4.

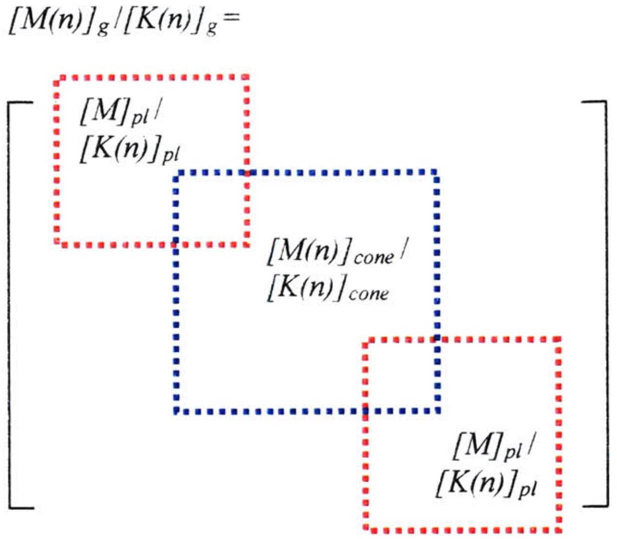


Figure 2-4 Assembly of global stiffness and mass matrices from individual plate and conical matrices

The overlapping of the corners of $[K]_{cy}$ and $[K]_{pl}$, $[M]_{cy}$ and $[M]_{pl}$ matrices indicates the sharing of nodes by the two types of elements at the edge of the endcaps. The equation of motion for the object with *n* circumferential lobes can be represented in the familiar undamped matrix form by equation (2.21).

$$\left[[K(n)]_G - \omega^2 [M(n)]_G\right]\{\delta\}\cos(n\theta) = \{F\}\cos(n\theta)\,, \tag{2.21}$$

where $\{\delta\}$ is the vector of nodal displacements on plates and shell

$\{F\}$ is the vector of nodal forces on plates and shell

Ambiguity in the surface normal direction of the corner nodes that connect the endcap elements to conical shell elements arises since the adjacent surfaces are perpendicular on a 2-D view. This is dealt with by "shifting" of nodal locations by procedures as described in section 2.2.4. As a result, no corner node will be represented in the admittance matrix for virtual source computation. Moreover, the circular plate elements that represent the endcaps do not have torsional degree of freedom in the circumferential direction, θ, and are not able to take membrane forces. Thus, the corresponding 2 degree of freedoms at the corner nodes are removed at this stage by eliminating the associated rows and columns in the dynamic stiffness matrix.

Only the displacement normal to the object is assumed to be coupled to the external fluid medium. Under this assumption, the dynamic stiffness matrix of shell normal displacement was further condensed by extracting only the rows and columns of the inverse of the dynamic stiffness matrix of all nodal displacements that correspond to the normal displacement, w and rotation, φ of the object, both on the circular plates and cylinder. Thus, equation (2.21) is rearranged to yield

$$\{w,\varphi\}\cos(n\theta) = \left[[K(n)]_G - \omega^2[M]\right]^{-1}\{F\}\cos(n\theta)\,. \tag{2.22}$$

2.2.4 Admittance matrix in the 3D domain

By principle of virtual work [17], the nodal force vector can be derived from the elemental pressure by the volume integral,

$$W = \int PwdV\,. \tag{2.23}$$

By expressing w in terms of the nodal displacement, i.e. $w = \Phi_{i1}w_i + \Phi_{i2}\varphi_i + \Phi_{j1}w_j + \Phi_{j2}\varphi_j$ where Φs are the respective shape functions, the nodal force vector {F} for one single plate element becomes

$$\{F\} = P\pi\int_b^a [\Phi_{i1}\ \ \Phi_{i2}\ \ \Phi_{j1}\ \ \Phi_{j2}]^T rdr\,, \tag{2.24}$$

where a: Outer radius of annular plate element,

b: Inner radius of annular plate element.

Similarly, for one single conical shell element, {F} is expressed as

$$\{F\} = P\pi l\int_0^1 [\Phi_{i1} \quad \Phi_{i2} \quad \Phi_{j1} \quad \Phi_{j2}]^T [\alpha_i(r) \quad \alpha_j(r)] d\xi [r_i \quad r_j]^T \ , \tag{2.25}$$

where r is expressed in terms of shape functions and radius of the cone at the ith and jth node i.e.

$$r = \alpha_i(r) r_i + \alpha_j(r) r_j \, . \tag{2.26}$$

In assembling the nodal force vector to form the global force vector in the 2D domain, a matrix, {Q}, can be formed from equations (2.24) to (2.26) to relate the nodal forces to elemental pressure, such that

$$\{F\}_n = \{Q\}\{p\}_n \, . \tag{2.27}$$

By using shape functions, elemental displacement at the mid-point of elements can be expressed in terms of nodal displacement and rotation, forming a matrix at the global assembly level,

$$\{U\}_n = [N]\{w,\theta\}_n \, . \tag{2.28}$$

By combining equations (2.3), (2.22), (2.27) and (2.28), the following expression can be derived:

$$\{U\}_n = \rho\omega^2 [N]\left[[K(n)]_G - \omega^2 [M(n)]_G\right]^{-1} [Q]\{\psi\}_n \, , \tag{2.29}$$

leading to the 2D admittance matrix

$$[A]_n = \rho\omega^2 [N]\left[[K(n)]_G - \omega^2 [M(n)]_G\right]^{-1} [Q] \, . \tag{2.30}$$

For *p* number of elements in the 2-Dimensional finite element system of the structure and with the structure further discretised into *q* orders in the circumferential coordinate, *θ*, the total number of elements is thus *pq*. To derive the 3-D admittance matrix [*A*] defined as

$$\{U\}=[A]\{\psi\}, \tag{2.31}$$

where $\{\psi\}$ is the vector containing all the elemental displacement potential and $\{U\}$ is the vector of elemental normal displacement at midpoint of element. Figure 2-5 illustrates the elemental label in greater details.

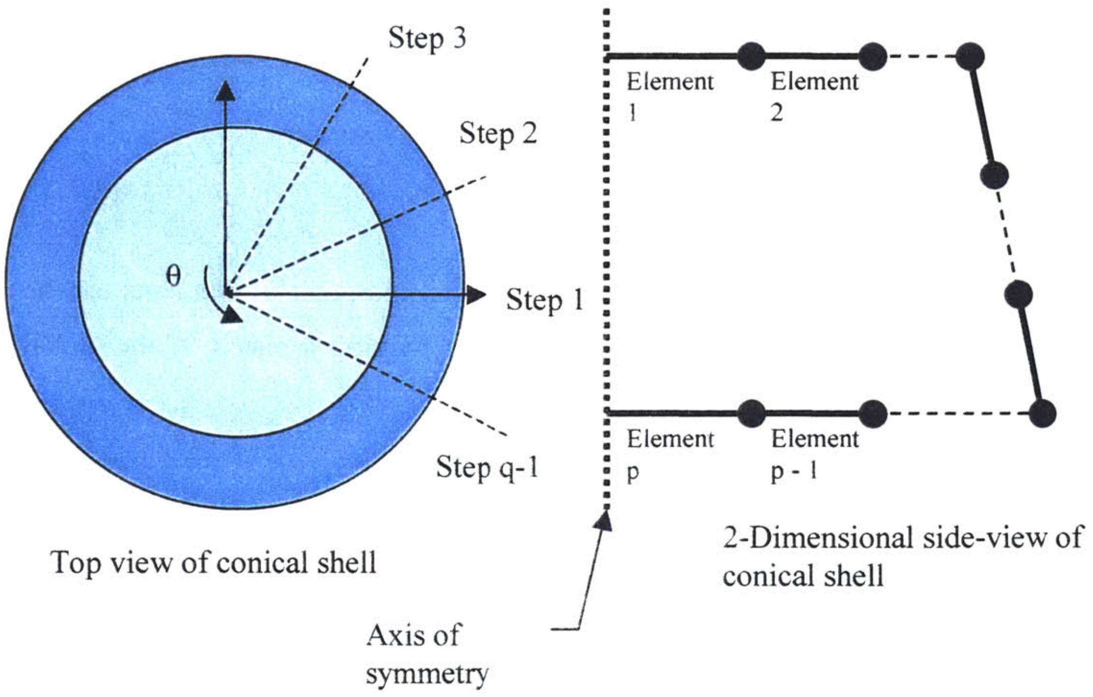


Figure 2-5 Illustration of nodal labeling of finite element structure.

For pressure $(\rho\omega^2\psi_n cos(n\theta))$ applied on element j, the displacement on the entire cone can be represented in discrete form,

$$\{U(\theta)\}=\{A_j\}_n\psi_n\cos(n\theta), \tag{2.32}$$

where $\{A_j\}_n$ is a column vector extracted from jth column of the $[A]_n$ matrix.

Thus, for a unit pressure applied on element j, the displacement on the entire cone is expressed by the Fourier series expansion

$$\{U(\theta)\} = \sum_{n=o}^{\infty} \{A_j\}_n \psi_n \cos(n\theta)\,, \tag{2.33}$$

where the terms in the Fourier expansion are

$$A_o = \frac{\Delta\theta}{2\pi}$$
$$A_n = \frac{2}{n\pi}\sin(\frac{n\Delta\theta}{2})\,.$$

For computational reasons, the summation in equation (2.33) has to be truncated to finite n. The structural stiffness of the conical shell results in a limit on the bending radius of the shell, which in turn causes a decrease in magnitude of transverse shell displacement with circumferential lobes. Thus, higher order modes do not have significant effect on the radiated field. It has been investigated that orders more than 10 do not have contribution to the radiation. Thus, the summation truncates at n = 10.

By repeating the above for element 1 to p, the displacement potential – displacement relation is thus fully defined, due to the axisymmetric geometry of the target. The 3-D admittance matrix [A] can then be computed and assembled.

2.3 Farfield Radiation of Target

The incident wavefield on the target from the source and the subsequent waveguide scattering are computed using MIT's OASES propagation model. The approach used in the OASES program is the wavenumber integration approach [6,8].

In this approach, the waveguide is first stratified into layers. Within each layer, the wavefield is represented by an upward propagating and a downward propagating wave representation in the wavenumber domain. Analogous to the inverse Fourier transform that allows the signal in the time domain to be computed from the magnitude in the frequency spectrum, the wavefield in the layer as a function of depth and range can be thus represented by inverse Hankel transform of the wavenumber representation as

$$\psi(r,z) = \int_{0}^{\infty} \psi(k_r,z) J_o(k_r r) k_r dk_r . \qquad (2.34)$$

For a layer without a source, the potential representation in terms of wavenumber in the kernel is comprised of 2 components - $A^{-}e^{-iK_z z} + A^{+}e^{+iK_z z}$. For a layer with a source, the displacement potential field includes an additional term in the integration kernel that represents the Green's function due to a point source. The coefficients in the kernel (i.e. A^+ and A^-) are determined from boundary conditions at the interfaces between layers, such as continuity of displacement and stresses or requirement for some stresses to vanish, depending on the type of waveguide and the sea bottom characteristics.

The evaluation of the wavenumber integral requires truncation. Thus, the integration is normally truncated when the above kernel no longer has any significant contribution to the integration value beyond a certain wavenumber.

There are various ways to solve for the kernel coefficients. In the Direct Global Matrix Approach, (DGM), equations are set up to solve for the kernel coefficients using the boundary conditions in the form of matrix representation. Numerical stability and other computational-related issues such as mapping of matrix elements and wavenumber discretization are being discussed in [6]. In the absence of coupling issues and numerical instability due to evanescent layer, another approach, the Propagator Matrix approach also proves the be an effective tool for solving the coefficients. Other tools available include the invariant embedding method. OASES uses the DGM approach with high numerical efficiency. Moreover, numerical stability has been ensured due to the global matrix mapping implemented.

The wavenumber integration method takes advantage of the FFT algorithm to evaluate the integration which makes the computation very efficient and time-consuming. This is also due to the asymptotic nature of the Bessel function that allows the representation in a format suitable for FFT possible.

The above allows the field to be computed from and to the farfield. The local scattering of the target is computed using equation (2.6) to solve for the source strength vector {s} by MIT's SCATT. Using wavefield superposition, the directivity function D(θ,Φ) can be derived. The directivity function can be understood as a 3D function that represents the strength of the radiation as function of the azimuthal and elevation angle such that in a free-space, it can be simply combined with the Green's function to give the field,

$$\psi(r,\theta,\Phi) = D(\theta,\Phi)\frac{e^{ik_r r}}{r} \tag{2.35}$$

The directivity function changes with the environment that the target is being placed. For example, in the case of a half-buried target, the directivity function changes due to interaction of the field with 2 different medium with different velocities and densities.

In SCATT, the scattered field (in terms of dB scale) is plotted in the form of scattering functions. The scattering functions are expressed in both the imaginary part and real part of the directivity function which represents the field in the radiating and evanescent regime respectively. To compute total pressure, the OASES program combines the scattered pressure with the original incident pressure in the waveguide.

3 COMPUTATIONAL RESULTS FOR ELASTIC CYLINDER

In this chapter, the benchmarking and evaluation of the FE code written based on the equations and theories described in Chapter 2 will be discussed.

For this purpose, an analytical model of the vibration of the shell was set up. In particular, close form solution for a cylinder, which is a special case of a cone, with simply supported boundary conditions is available and will be adopted as a model for this purpose of evaluation. The analytical model will be described in this chapter. The resonant frequencies of the cylinder will be computed and compared with the FE model as a first check. This is followed by the computation of the response of the shell when subjected to various kinds of loading and comparison using analytical and FE model. Subsequently, computation of scattering by virtual source approach using the admittance matrix generated by both methods will be compared. Moreover, scattering functions will also be compared with pressure-release and rigid targets computed internally in SCATT using the virtual source approach. Other issues, such as changes in scattering pattern with target orientation and resonant effect of shell on scattering, will be discussed. An analysis and discussion of results will conclude this chapter.

3.1 Analytical modeling of in vacuo vibration of cylinder

Junger and Feit [1] outlined the procedures for studying the in vacuo vibration of cylindrical shells starting with the equations of motion. The geometry of the cylindrical shell is as shown in Fig. 3.1.

In the formulation of equations of motion for the shell, the 3 terms considered are

i. Kinetic energy (K.E.),

$$K.E.=\frac{1}{2}\rho_s h\iint(\dot{u}^2+\dot{v}^2+\dot{w}^2)dS\,, \tag{3.1}$$

ii. Potential energy (P.E.),

$$P.E. = \frac{1}{2}\iiint\left(\sigma_{zz}\varepsilon_{zz} + \sigma_{\theta\theta}\varepsilon_{\theta\theta} + \sigma_{z\theta}\varepsilon_{z\theta}\right)dV\ , \tag{3.2}$$

where V: Volume,

σ, ε: Stresses and strains.

iii. Work done by external displacement potential,

$$W = \iint \rho\omega^2\psi w dS\ . \tag{3.3}$$

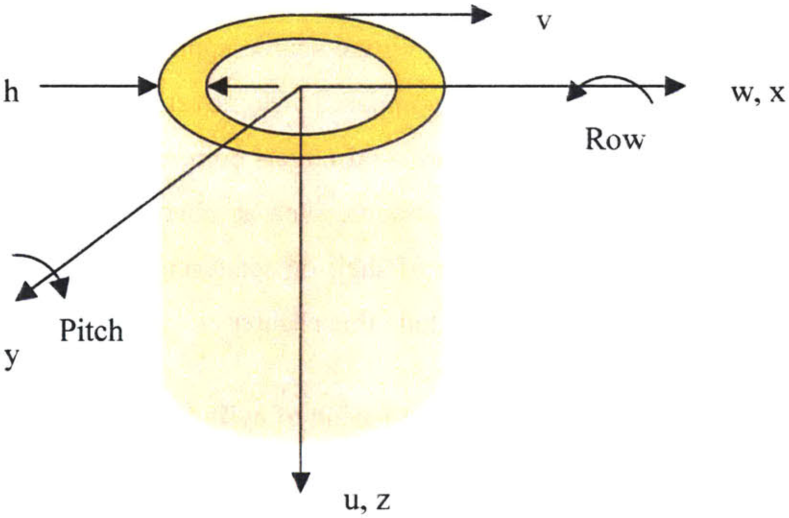


Figure 3-1 Analytical cylindrical shell model

Hamilton's principle was applied to all 3 directions of shell to derive the 3 equationss of motion. Furthermore, simply supported boundary conditions was applied to the shell,

$\frac{\partial u}{\partial z} = 0$ at z = 0 and L,

$v = 0$ at $z = 0$ and L,

$w = 0$ at $z = 0$ and L.

Thus, to allow computation of deformation of shell under acoustic excitation, the displacements and displacement potential are expanded in a Fourier series. The 4th order partial differential equations of motion can be reduced to a linear system of equations. The Fourier expansion of the displacements and displacement potential are

$$u = \Sigma U_{mn} \cos n\theta \cos k_m z e^{-i\omega t}, \tag{3.5a}$$

$$v = \Sigma V_{mn} \sin n\theta \sin k_m z e^{-i\omega t}, \tag{3.5b}$$

$$w = \Sigma W_{mn} \cos n\theta \sin k_m z e^{-i\omega t}, \tag{3.5c}$$

$$\psi = \Sigma \Psi_{mn} \cos n\theta \sin k_m z e^{-i\omega t}, \tag{3.5d}$$

where $k_m = \frac{m\pi}{L}$.

By using a 2D Fourier series expansion of rectangular function of unit magnitude, the displacement due to a rectangular patch of pressure applied along $z = l_1$ to $z = l_2$ at $\theta = 0^\circ$ can be computed at all locations on the shell, where the summation truncates at m = 10 and n =10 due to reasons as described earlier in section 2.2.4. The admittance matrix can thus be formulated by calculating the displacement at discrete patches on the cylindrical surface due to pressure on all discrete patches at $\theta = 0^\circ$ along the z-axis.

$$\psi = \Sigma\Sigma \Psi_{mn} \cos n\theta \sin k_m z, \tag{3.6}$$

where $$\Psi_{mn} = \frac{2\Delta\theta}{m\pi^2}[\cos(\frac{m\pi l_1}{L}) - \cos(\frac{m\pi l_2}{L})] \qquad \text{for } n = 0,$$

$$\Psi_{mn} = \frac{4}{mn\pi^2}\sin(\frac{n\Delta\theta}{2})[\cos(\frac{m\pi d_1}{L}) - \cos(\frac{m\pi d_2}{L})] \qquad \text{for n} \neq 0,$$

and $\Delta\theta$ is the width of rectangular patch in the θ direction.

3.2 Resonant frequencies of FE model

Using equations (3.1) and (3.2), the equations of motion of the cylinder without pressure loading are derived. The 4th order partial differential equations are reduced to 3 linear equations in terms of U_{mn}, V_{mn} and W_{mn} using equations (3.5a) to (3.5c). In terms of matrix representation, the equations are re-written to yield

$$\begin{bmatrix} \Omega^2-(k_m a)^2-\frac{n^2(1-\upsilon)}{2} & \frac{1+\upsilon}{2}nk_m a & \upsilon k_m a \\ \frac{1+\upsilon}{2}nk_m a & \Omega^2-\frac{1-\upsilon}{2}(k_m a)^2-n^2 & -n \\ \upsilon k_m a & n & -\Omega^2+1+\beta^2[(k_m a)^2+n^2]^2 \end{bmatrix} \begin{bmatrix} U_{mn} \\ V_{mn} \\ W_{mn} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad (3.7)$$

where Ω : Normalized frequency ($\frac{\omega a}{c_p}$),

β^2 : Square of normalized thickness ($\frac{h^2}{12a^2}$),

c_p : Sound speed ($\sqrt{\frac{E}{\rho_s(1-\upsilon^2)}}$).

By setting m = 1, the resonant frequencies for various n values from n = 0 to 10 was computed by solving for the eigenvalue Ω that appears only in the diagonal terms in the 3 x 3 matrix.

Using the FE approach described in chapter 2, corresponding discrete model for the cylinder without endcaps was derived. The circumferential degree of freedom, v and transverse degree of freedom, w at the ends of the cylinder were constrained so that the boundary conditions were also identical.

To compare the FE code with the analytical modeling, a simply supported cylinder with material properties and geometry as described in Table 3.1 was used for simulation by both approaches. 2 elements of length 1m each were used to represent the cylinder, thus resulting in 8 degrees of freedom after 2 had been removed from each ends due to constraints of boundary conditions.

Table 3-1 Material properties and geometry of a simply supported cylinder for comparing natural frequencies

Thickness (m)	0.01
Radius (m)	0.3
Height (m)	2
Young's Modulus (N/m^2)	206
Density(kg/m^3)	7500
Poisson's ratio	0.3

The natural frequencies for the transverse vibration were plotted and compared as shown in Fig. 3.2 for both the analytical and FE model. The natural frequencies were not plotted for n values less than 3 for the FE model as the FE model was not able to produce eigenvalues corresponding to distinct mode shapes that represent resonance of transverse vibration. Moreover, it was also observed that the FE model agrees very well with the analytical model with slight discrepancies at lower order modes n = 3 and 4. This shows that the FE model generated from 2-D elements may not be able to represent the cylindrical shell for lower order modes as well as higher order modes.

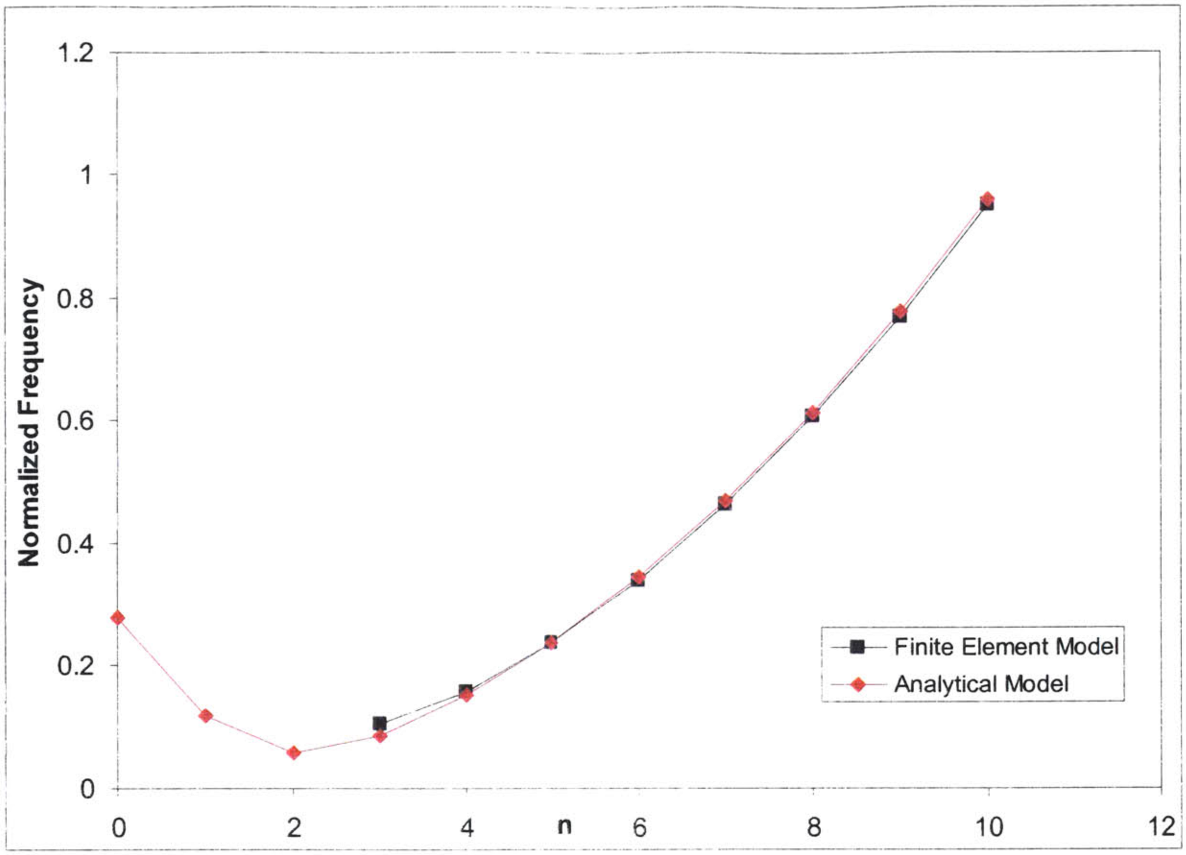


Figure 3-2 Normalized natural frequencies, Ω of transverse vibration mode for various number of circumferential lobes, n as predicted by both the FE and analytical models

The reasons for the differences and absence of eigenvalues in FE model corresponding to lower order modes can be deduced by studying the main difference between the analytical and FE models i.e. stiffness. The finite degrees-of-freedom in FE models results in greater stiffness. This effect is more prominent in the prediction of lower natural frequencies where the ratio of added stiffness due to discretization to the actual stiffness of physical model is large. Moreover, due to the simplification of a 3-D to 2-D model, 3-D mode shapes corresponding to coupling of vibrations in various directions may not be very well represented in a 2-D axi-symmetric conical shell model.

For such cases, 3-D thin shell elements might be necessary for the FE implementation to generate the respective mode shapes with lesser circumferential lobes.

3.3 Response of cylinder under acoustic pressure loading

In this section, response of the FE model under acoustic excitation will be further compared with the analytical model.

3.3.1 Displacement potential of the form $\psi = \Psi_n \cos n\theta \sin\frac{\pi}{L}ze^{-i\omega t}$

For the analytical model, equation (3.7) was used to solve for the response. The right-hand side of the equation can be replaced by column vector $\begin{Bmatrix} 0 \\ 0 \\ \Psi_{mn} \end{Bmatrix}$ where Ψ_{mn} is the coefficient as used in equation (3.6). For purpose of evaluation, the pressure is assumed to be

$$P = \rho\omega^2\psi = \rho\omega^2\Psi_n \cos n\theta \sin\frac{\pi}{L}ze^{-i\omega t} \tag{3.8}$$

Where Ψ_n is the coefficient of the Fourier series expansion as defined in equation (2.33) and $\Delta\theta$ had been arbitrarily assumed to be 10°.

By solving equation (3.7) with the above inputs for U_{mn}, V_{mn} and W_{mn}, the deformation magnitude on the entire cylinder is given by equations (3.5a) to (3.5c).

For the FE model, equation (2.29) was used. The column vector $\{\psi\}_{2D}$ takes values of the function $\Psi_n \sin(\frac{\pi}{L}z)$ where z is the z-coordinate on the shell of the mid-point of the element corresponding to the particular row of the column $\{\psi\}_{2D}$. The vector $\{U\}_{2D}\cos(n\theta)$ thus defines the deflection of the entire cylinder under the loading.

For an excitation frequency corresponding to kr = 3 for the model illustrated on Table 3-1 and surrounding fluid sound speed of 1500 m/s,

➔ K(0.3) = 3

➔ $\frac{\omega}{c} = 10$

➔ ω = 15000

➔ f = 2387 Hz.

Also, a pressure loading of magnitude $\rho\omega^2\Psi_n \cos n\theta \sin\frac{\pi}{L}$ was applied. The difference in loading between the 2 approaches is as illustrated in Fig. 3.3, for a discretization length of $\frac{\lambda}{4}$. Due to the discrete nature of the FE model, the loading in this case is step-wise sinusoidal as opposed to the continuous analytical model that produced a smooth sinusoidal loading profile.

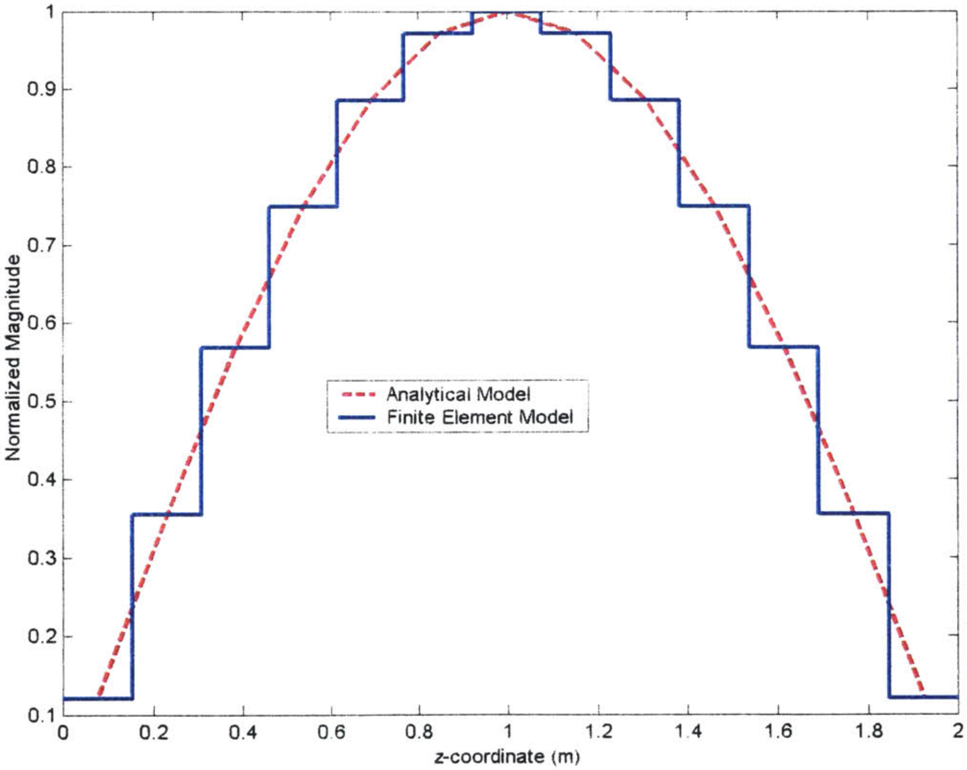
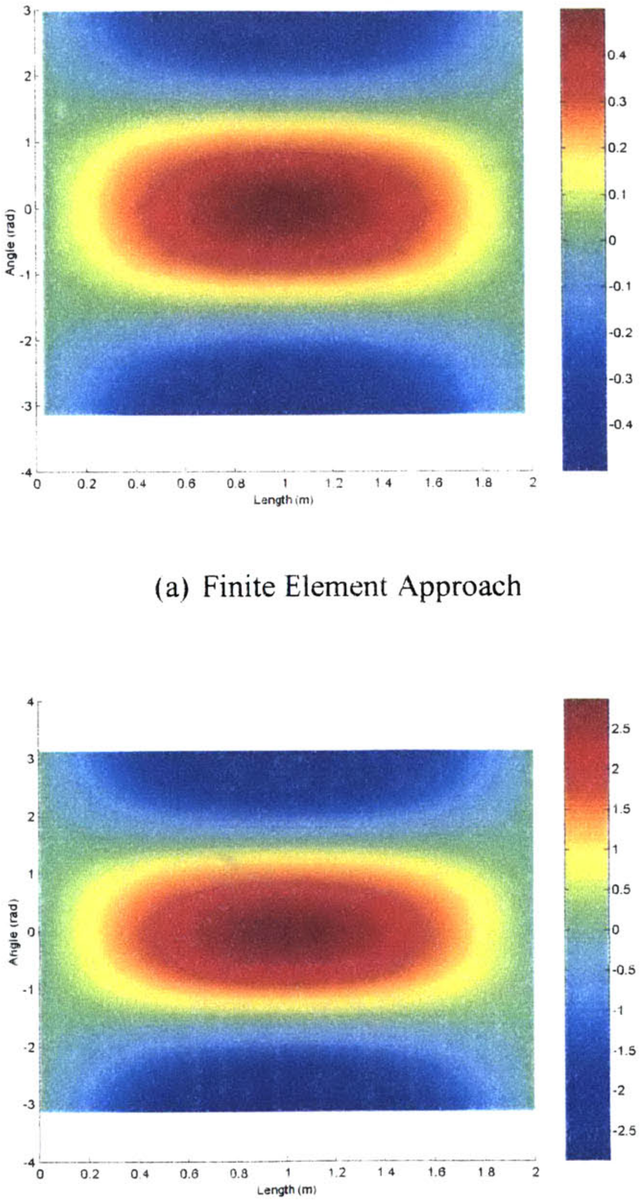


Figure 3-3 Plot of normalized displacement potential $\frac{\psi}{\Psi_n}$ along the z-coordinate for θ = 0° for both analytical and FE model

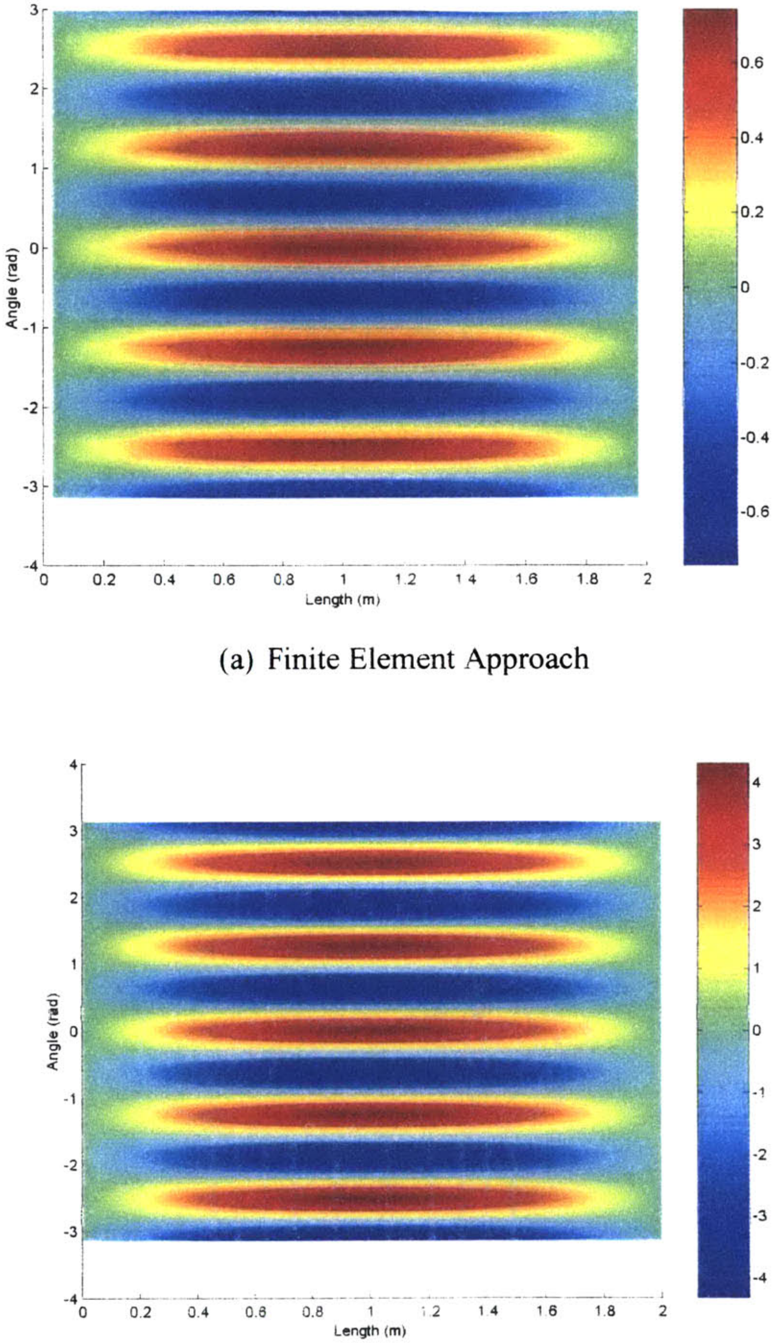
The deformation obtained by the FE and analytical models are as shown in Fig. 3-4 for n = 1 and in Fig 3.4 for n = 5.



(a) Finite Element Approach

(b) Analytical Approach

Figure 3-4 Cylindrical shell deformation (m) for pressure loading of $\rho\omega^2\Psi_n \cos n\theta \sin\frac{\pi}{L}$ for n = 1



(a) Finite Element Approach

(b) Analytical Approach

Figure 3-5 Cylindrical shell deformation (m) for displacement potential loading of $\rho\omega^2\Psi_n \cos n\theta \sin\frac{\pi}{L}$ for n = 5

From Fig 3.4 and Fig. 3.5, it is observed that the patterns of deformation are very similar for the two approaches. The peak deformation is also observed to be consistently higher for n = 5 than n = 1 for both methods. The discrete nature of the FE approach resulted in lower resolution in the deformation profile.

Moreover, due to the limited degrees of freedom as opposed to the analytical approach that has an infinite degree of freedom, the FE model is stiffer. There is also a difference in loading profile between the 2 approaches as described in Fig. 3.3. As a result of these factors, the magnitude of deformation is lower for the FE model

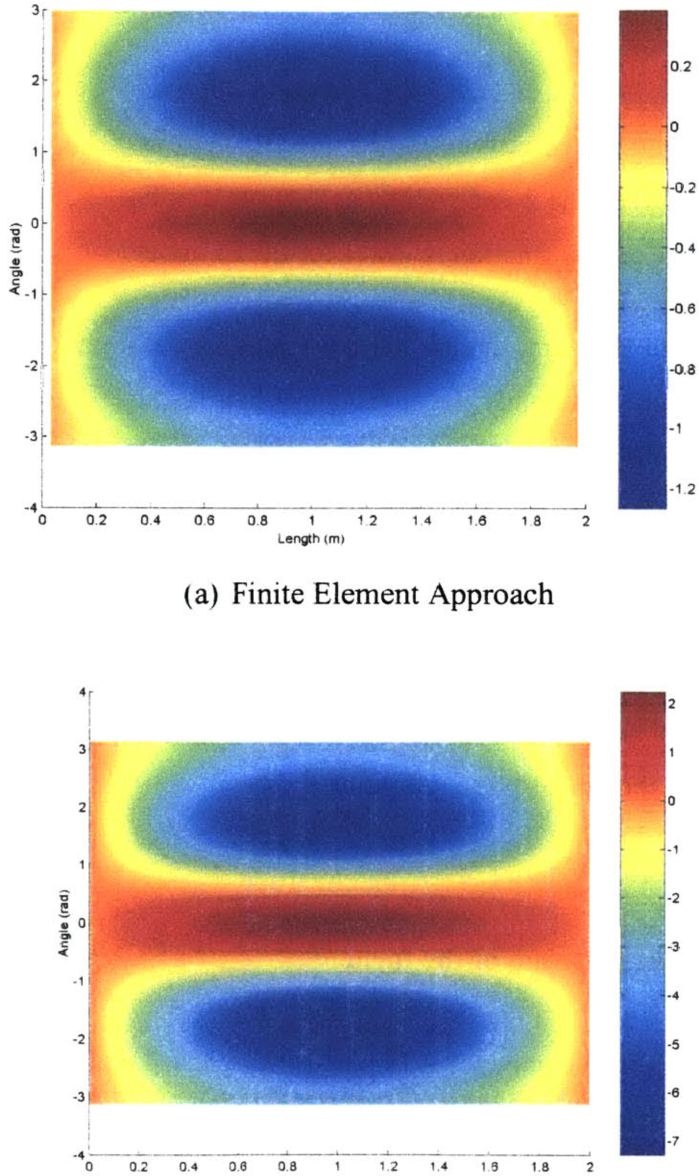
It is also interesting to note that though the magnitude in deformation is positive for $\theta = 0^{\circ}$, the applied pressure is actually negative. This is a result of the requirement for the sign convention for the displacement potential and displacement to be consistent with SCATT program. The applied pressure and displacement are thus in the opposite direction. Thus, the plots in Fig. 3.4 and Fig. 3.5 illustrated that the displacement is actually out of phase with the pressure loading. This is consistent with theoretical prediction. As shown in Fig. 3.2, the natural frequencies for both modes (i.e. 1 and 5 circumferential lobe) have natural frequencies lower than the excitation frequency of 2387 Hz, resulting in the out-of-phase vibration.

3.3.2 Displacement potential of the form $\psi = \sum_{n=0}^{N} \Psi_n \cos n\theta \sin\frac{\pi}{L} z e^{-i\omega t}$

A similar analysis as the one conducted in part I. was carried out. The only difference is in the loading, which takes the form of equation (3.9)

$$P = \rho\omega^2\psi = \rho\omega^2 \sum_{n=0}^{N} \Psi_n \cos n\theta \sin\frac{\pi}{L} z e^{-i\omega t} \tag{3.9}$$

Where Ψ_n is the coefficient of the Fourier series expansion as defined in equation (2.33) and $\Delta\theta$ had been arbitrarily assumed to be 10°. For purpose of this analysis, n was arbitrarily set to n = 2 and n = 7 for verifying consistency between the 2 approaches. The deformation as a result of analysis by both FE and analytical models are as shown in Fig. 3-6 and Fig. 3-7.

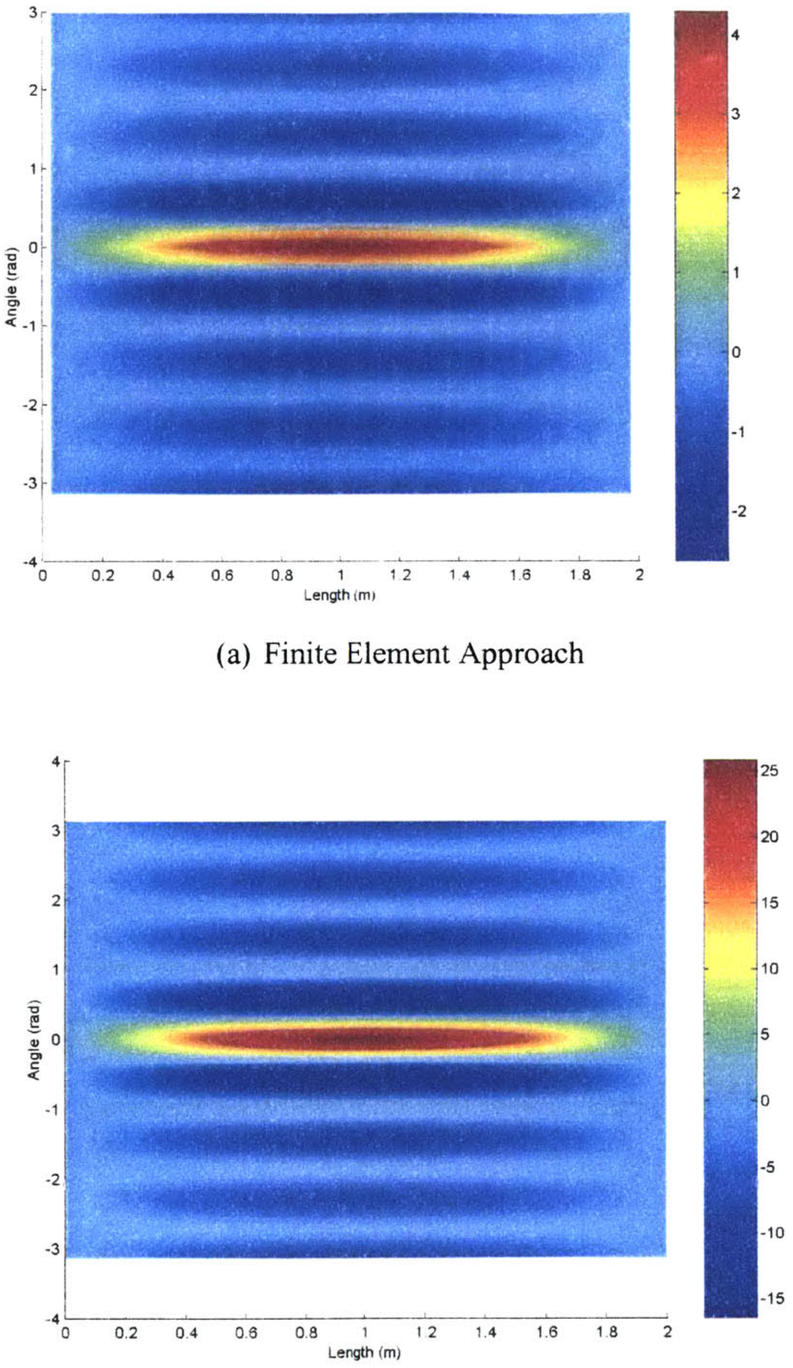


(a) Finite Element Approach

(b) Analytical Approach

Figure 3-6 Cylindrical shell deformation (m) for pressure loading of

$$P = \rho\omega^2 \sum_{n=0}^{N} \Psi_n \cos n\theta \sin\frac{\pi}{L}z \text{ for n = 2}$$



(a) Finite Element Approach

(b) Analytical Approach

Figure 3-7 Cylindrical shell deformation (m) for displacement potential loading of

$$\psi = \rho\omega^2 \sum_{n=0}^{N} \Psi_n \cos n\theta \sin\frac{\pi}{L}z \text{ for n = 7}$$

Close agreement in the deformation profile is observed in this part of evaluation between the 2 approaches. Similar scale of difference in magnitude between the 2 approaches as observed in section 3.3.1 is also observed here due to reasons accounted for.

To provide an alternate perspective in comparison of the 2 approaches, Fig 3.8 illustrates the deformation predicted by FE modeling, analytical modeling and displacement potential due to the the applied loading of $P = \rho\omega^2 \sum_{n=0}^{N} \Psi_n \cos n\theta \sin\frac{\pi}{L}z$ for n = 2 at z = 1m. It can be observed that the analytical model predicts greater deformation than the FE model due to its lower stiffness as discussed earlier in section 3.3.1. Moreover, the deformation along the circumferential angle by both methods follows closely the lobes of the applied loading.

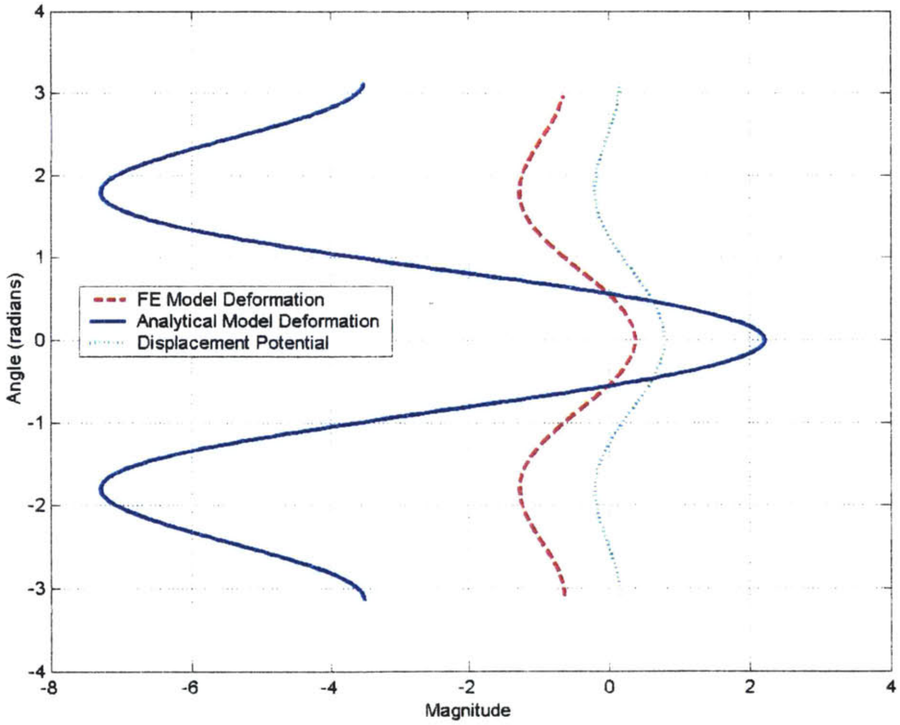


Figure 3-8 Comparison of FE model, analytical model deformation and displacement potential $\psi = \sum_{n=0}^{N} \Psi_n \cos n\theta \sin\frac{\pi}{L}z$ for n = 2 at z = 1m

This section demonstrates the accuracy level of the FE model in prediction of the deformation profile and thus ability in generating the admittance matrix for representing the acoustic boundary condition in the virtual source approach of target scattering prediction. The accuracy in the scattered field computed using these admittance matrices will be further investigated in the following sections.

3.4 Acoustic Scattering from steel cylinder without endcap

The admittance matrices for representation of boundary conditions on the target surface were generated for various frequencies and material properties using the FE approach as outlined in section 2.2. Moreover, the corresponding matrices were generated using the analytical approach as outlined in section 3.1.

The material properties are those for a steel cylinder used in chapter 2 of [2]. In summary, the material properties and the dimension of the simply supported cylinder without endcap is as shown in Table 3-2. The coordinate system and pitch/roll axes of the cylinder are consistent with that shown in Fig. 3-1. In summary, the target orientation and position with reference to the incident plane waves is as illustrated in the benchmark scenario for target scattering in free-space model (Appendix A).

Table 3-2 Material properties and geometry of a simply supported steel cylinder for studying acoustic scattering

Thickness (m)	0.01
Radius (m)	0.3
Height (m)	2
Young's Modulus (N/m^2)	195 x 10^9
Density(kg/m^3)	7700
Poisson's ratio	0.28

The admittance matrices generated were then input into OASES/SCATT for computation of scattering function for both the horizontal (x-y) plane and vertical (x-z) plane in a free-space. The .dat file for use with the oast command in OASES [8] is as shown in Appendix B.

For SCATT, the parameters used are as shown in Table 3-3.

Table 3-3 SCATT parameters used in studies

Number of sources in axial direction	30
Pitch of target	0°
Roll of target	0°

The scattering is computed for a hypothetical target due to the fact that the interior of the simply supported cylinder is vacuum. In reality, this is not possible since the endcaps are not present. But, for the purpose of benchmarking the FE approach against the hybrid analytical/virtual source approach, it is an effective model. The scattering function (SF) is is defined as

$$SF = 20\log|D(\theta,\Phi)| \tag{3.9}$$

where D is the directivity function [2]. Moreover, in the virtual source approach representation, the incident field is of unity pressure plane waves.

Fig. 3-9 to Fig. 3-11 illustrate the vertical scattering function (x-z plane) and horizontal scattering function (x-y plane) for frequencies of 5000 Hz, 2387 Hz and 1000 Hz respectively, for plane waves in the direction perpendicular to the cylinder's z-axis. Due to the symmetry of scattering functions about the x-axis as shown in Fig. 3-1, the vertical scattering function has been plotted only for angle, φ_V between -90° to $+90^{\circ}$ where 0° corresponds to the vertical z-axis (i.e. broadside axis). The scattering angle φ_V, is defined as

$$\varphi_V = \sin^{-1}(\frac{k_r}{k}) \tag{3.10}$$

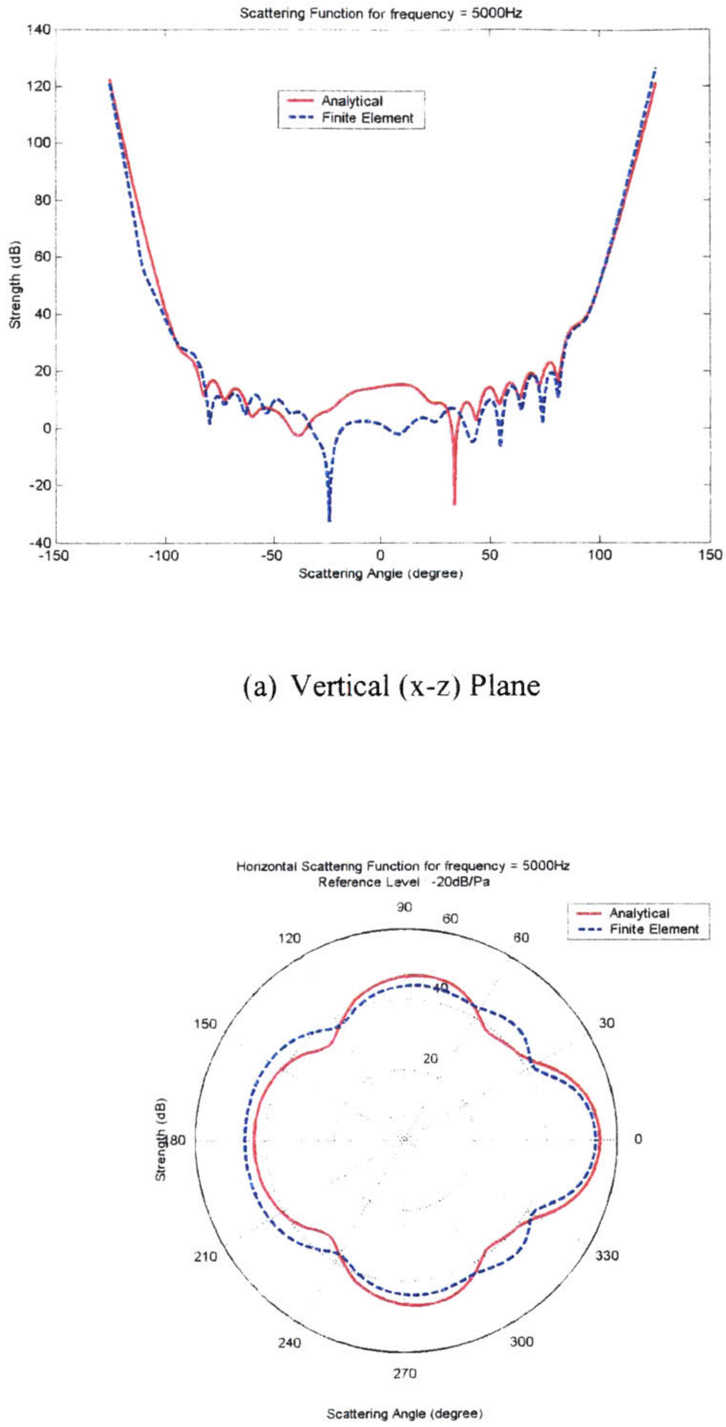
Moreover, to demonstrate the evanescent field, the scattering function plots have also been extended for $|\varphi_V| > 90^\circ$ where the scattering angle and the radial wavenumber are arbitrarily related by $\varphi_V = 90 + \frac{360}{2\pi}\log[\frac{k_r}{k} + \sqrt{(\frac{k_r}{k}) - 1}]$. In the plots, positive angle implies scattering in the forward direction while negative angles implies backscatter.

For horizontal scattering functions, polar plots are shown where 0° corresponds to the x-axis in the forward scatter direction. The horizontal scattering function plots were with reference to -20dB/Pa.

From the Fig 3.9 to 3.11, it is observed that there is close agreement between the hybrid finite element/virtual source approach and the analytical/virtual source approach in computing the scattering from the simply supported cylinder for higer frequencies. This is evident from the Figs. 3-9 and 3-10. However, at lower frequency of 1000 Hz (ka = 1.26), there was distinct difference between the 2 approaches, especially at broadside angle.

The earlier observation as described in section 3.2 that the FE and analytical models show discrepancies in prediction of natural frequencies for modes of lesser circumferential order indicates that at low frequencies, the 2 models do not agree well. Thus, at the lower frequency of 1000 Hz, the effect of difference has not dampened off as compared to higher frequencies.

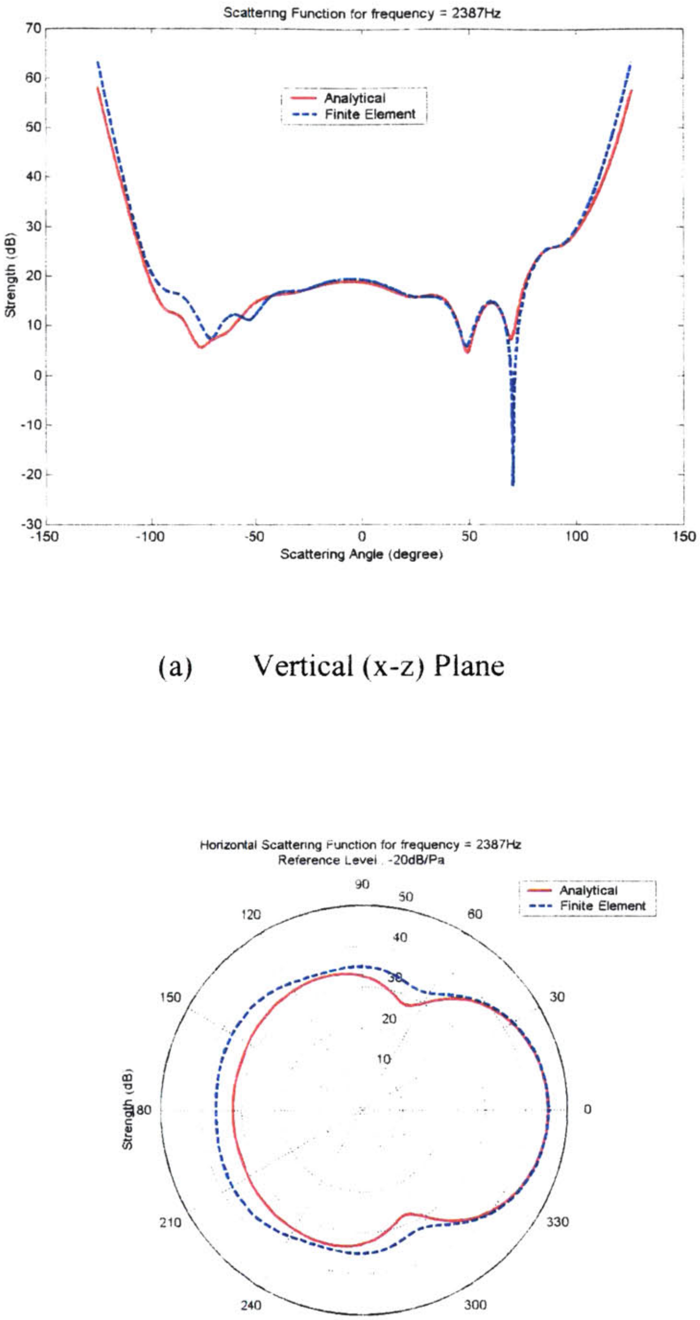
A similar simulation was conducted for a cylinder of thinner shell of 0.005m. This cylinder has lower natural frequencies for lower order modes due to the more flexible shell. Thus, at 1000 Hz, the effect of discrepancies for those modes is expected to have dampened off more as compared to that for 0.01m shell. The scattering functions are plotted as shown in Fig. 3-12 supporting the above conclusion at low frequencies since the 2 approaches agreed well in this case.



(a) Vertical (x-z) Plane

(b) Horizontal Plane (x-y) Plane

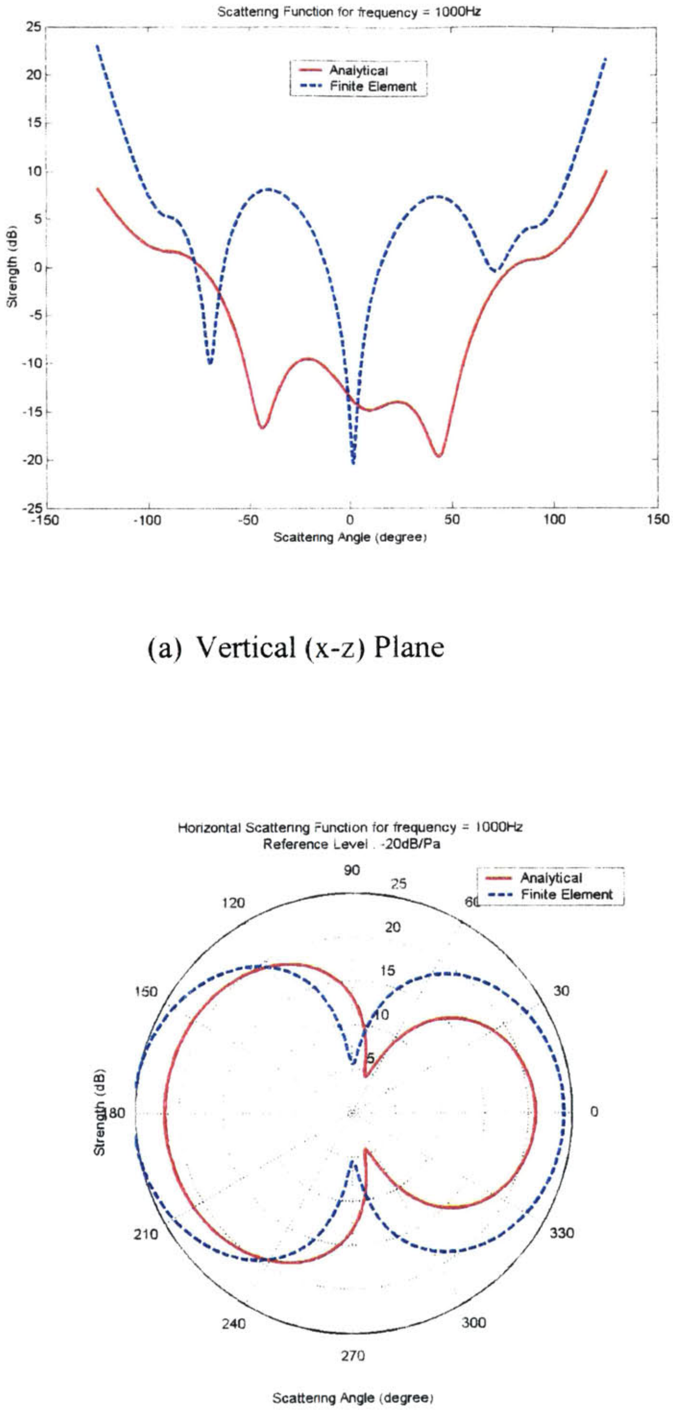
Figure 3-9 Scattering functions for frequency 5000 Hz and shell thickness 0.01m



(a) Vertical (x-z) Plane

(b) Horizontal Plane (x-y) Plane

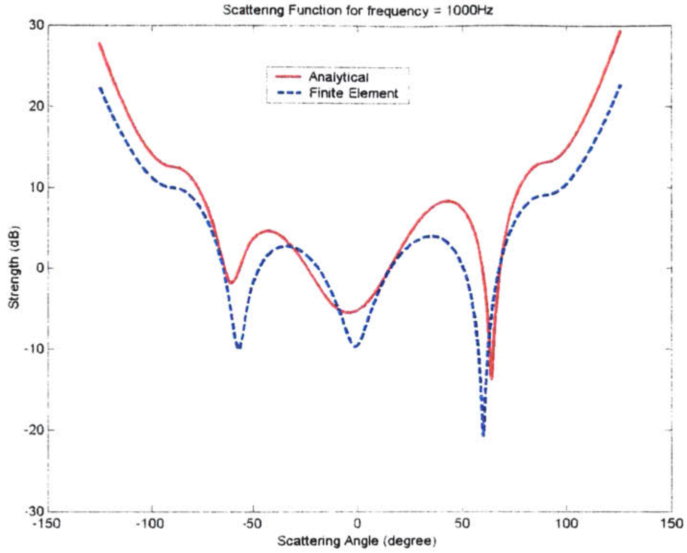
Figure 3-10 Scattering functions for frequency 2387 Hz and shell thickness 0.01m



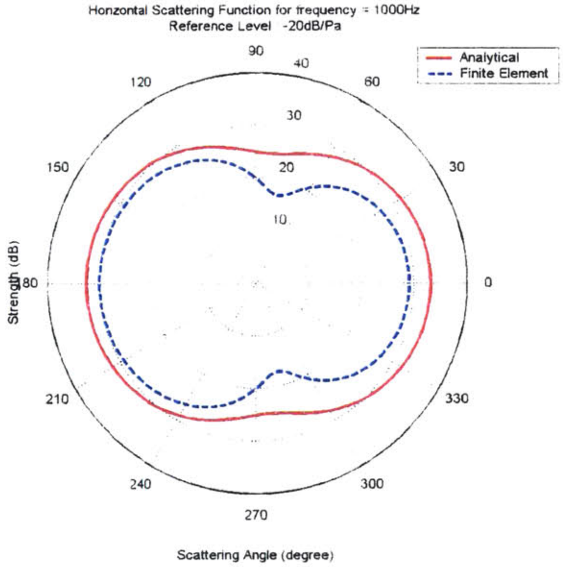
(a) Vertical (x-z) Plane

(b) Horizontal Plane (x-y) Plane

Figure 3-11 Scattering functions for frequency 1000 Hz and shell thickness 0.01m



(a) Vertical (x-z) Plane



(b) Horizontal Plane (x-y) Plane

Figure 3-12 Scattering functions for frequency 1000 Hz and shell thickness 0.005

3.5 Acoustic Scattering from steel cylinder with endcap

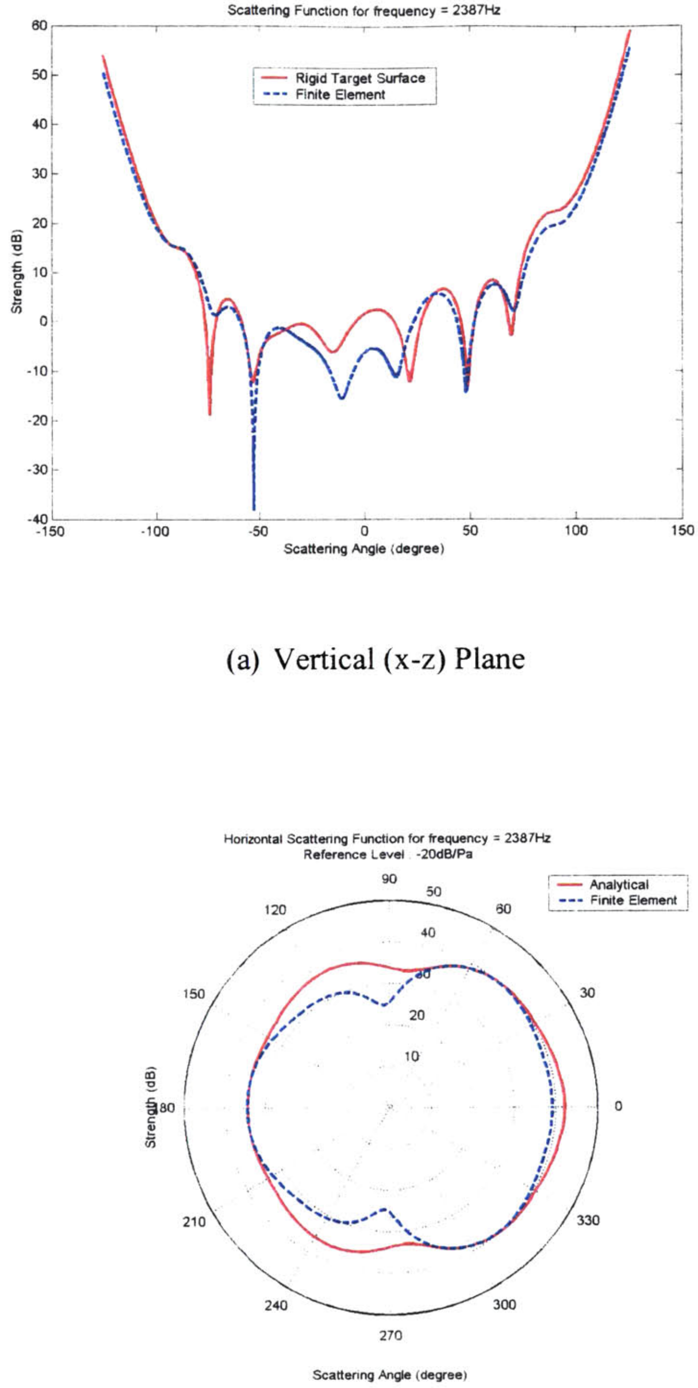
In this section, the evaluation of the computation of acoustic scattering from steel cylinders with endcaps will be discussed. The entire steel cylinder capped with flat plates on both ends was discretized with the elements as described in Chapter 2 to represent various target surface boundary conditions.

3.5.1 Rigid and Pressure Release Target Surface

By allowing the structural sound speed and density to approach infinity, the target material represents a rigid boundary condition on the surface. To achieve this, the density of the material was set to an order of 10^{23} times that of the steel material as described in Table 3-2. Since the structural sound speed is proportional to $\sqrt{\frac{E}{\rho_s}}$, the Young's Modulus was set to 10^{69} times that of the same steel material. Other geometry as illustrated in Table 3-2 were preserved.

In the SCATT virtual source approach, the scattering function of an equivalent target with rigid surface boundary condition can be computed analytically by requiring the total displacement on the target surface to vanish. The source strength vector for the virtual sources was then computed to meet this requirement and the scattering function subsequently computed.

Fig 3-13 shows the comparison of scattering functions as computed by both the 2 approaches. The frequency is 2387 Hz corresponding a ka (product of wavenumber and cylinder outer radius) value of 3.



(a) Vertical (x-z) Plane

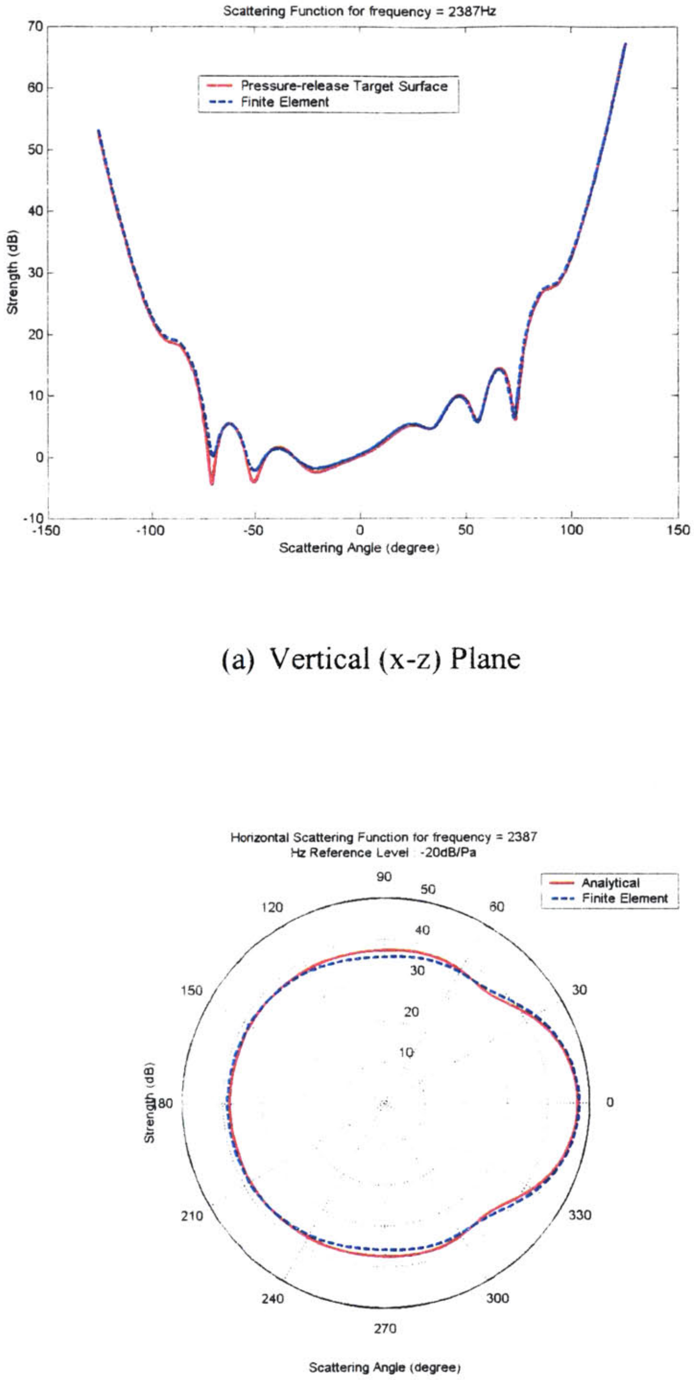
(b) Horizontal (x-y) Plane

Figure 3-13 Scattering function at ka =3 for rigid steel cylinder with endcaps

The Figure shows the close agreement in scattering functions computed between the 2 approaches, illustrating the validity of FE generation of admittance matrix for virtual source computation in this aspect.

A similar approach was conducted for pressure-release target surface. For the FE approach, the density and Young's modulus were respectively set to 10^{-23} and 10^{-69} times that of the material in Table 3-2. This corresponds to near-zero structural sound speed and density and thus represents a pressure-release target. In the SCATT virtual source approach, the scattering function of an equivalent target with pressure-release surface boundary condition can be computed analytically by requiring the total pressure on the target surface to vanish. The source strength vector for the virtual sources was then computed to meet this requirement and the scattering function subsequently computed.

Fig. 3-14 shows the comparison of scattering functions by the two approaches and the close agreement between them.



(a) Vertical (x-z) Plane

(b) Horizontal (x-y) Plane

Figure 3-14 Scattering function at ka =3 for pressure-release steel cylinder with endcaps

3.5.2 Elastic Target Surface

For the material and geometry as described in Table 3-2, the acoustic scattering for the steel cylinder with endcaps was computed using the admittance matrices generated from the FE code. The coordinate system and pitch/roll axes of the cylinder are consistent with that shown in Fig. 3-1.

Figs. 3-15 and 3-16 show the comparison between the scattering functions in the horizontal and vertical directions for the cylinder at a frequency of 2387 Hz. The scattering functions are compared to earlier results for the cylinder without endcaps. 3 plots are shown in the 2 figures:

a. Fig 3.15 – Horizontal scattering function for $0^{\circ}<\varphi_H<360^{\circ}$ where $\varphi_H = 0^{\circ}$ is the forward scatter direction.

b. Fig 3.16a – Vertical scattering function corresponding to the propagating wavefield for $0^{\circ}<\varphi_v<360^{\circ}$ where $\varphi_v = 0^{\circ}$ is the forward scatter direction and

$$\varphi_V \qquad = \sin^{-1}(\frac{k_r}{k})$$

c. Fig 3.16b – Vertical scattering function corresponding to the evanescent wavefield and

$$\varphi_V \qquad = \frac{360}{2\pi}\log[\frac{k_r}{k}+\sqrt{(\frac{k_r}{k})-1}]$$

For the frequency at 2387 Hz, it can be observed that there's a significant difference in scattering function between the models with and without endcaps. In particular, there is an increase in scattering strength in the range $60^{\circ}<\varphi_H<300^{\circ}$ in the horizontal plane when the endcaps are being removed. The increase even extends to a range of $30^{\circ}<\varphi_V<330^{\circ}$ for propagating scattering function in the vertical plane. For the evanescent field, the cylinder without endcaps also produce stronger field in the backscatter direction.

The above observation is interesting as it is different from a first intuition that the presence of endcaps should reflect sound better and thus cause stronger scattering. This can be accounted for by looking at the virtual source representation in the perspective of the requirement for these sources to satisfy the acoustic boundary conditions for an exterior problem [2]. Due to the presence of encaps, conditions have to be specified for a broader boundary that spans the endcaps' surface. Such conditions do not necessarily imply a stronger radiation, depending on a myriad of factors such as frequency, geometry of target and surrounding medium. This is analogous to an interior problem, such as plate vibration problems, where specifying the loading-displacement relation actually results in lesser degrees-of-freedom and may result in lower dynamic deformation.

The main physical difference between the two cases is that the cylinder without endcaps approximates an open tube while that with endcaps represents exactly a closed cylinder. Such factors as difference in diffraction pattern and the interaction of the elastic plates with the medium, resulting in possibility of partial cancellation of scattered waves from the cylinder by those from the endcaps, can also account for the lower scattering.

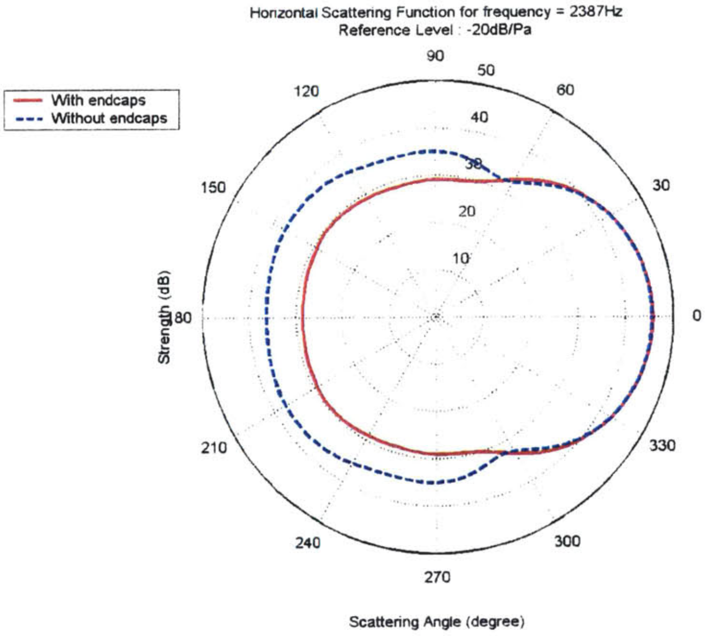
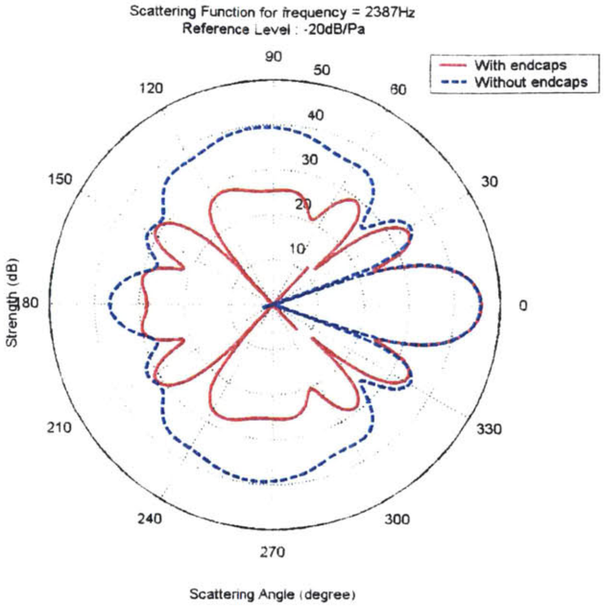
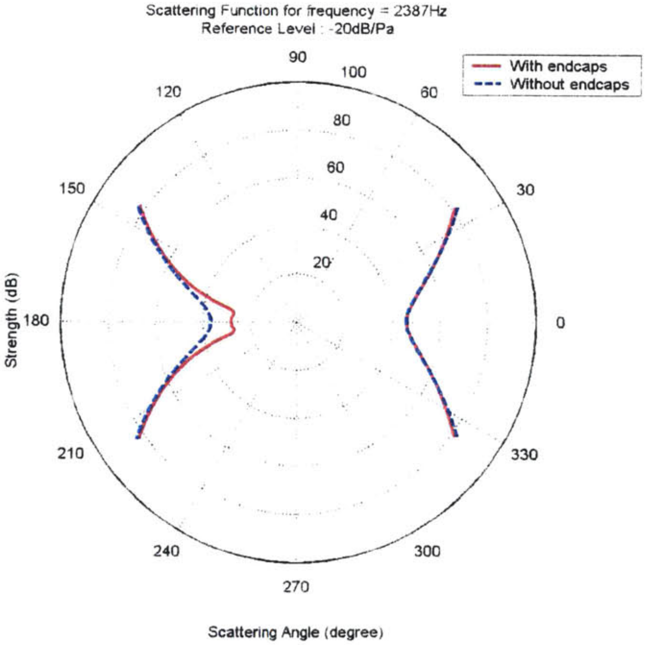


Figure 3-15 Horizontal Scattering function at 2387 Hz for steel cylinder



(a) Propagating



(b) Evanascent

Figure 3-16 Vertical scattering function at 2387 Hz for steel cylinder

Figs. 3-17 and 3-18 show the comparison between the scattering functions in the horizontal and vertical direction for the cylinders at a frequency of 5000 Hz. The scattering functions are at the same time compared to the solution obtained if endcaps were not included.

At the higher frequency of 5000Hz, scattering from the cylinders are more specular than that at lower frequencies where interaction between the sound waves and structure contributes significantly to scattering. Thus, sidelobes of the vertical scattering at 5000Hz from the cylinder with endcaps are stronger than that for the cylinder without endcaps since direct specular reflection from the endcaps results in stronger scattering in the vertical z-direction. This is not observed at the lower frequency of 2387Hz.

It is also observed that at higher frequencies, i.e. higher wavenumber, the scattering pattern has more sidelobes. This is also consistent with the pattern change with increasing frequency for rectangular baffled piston [2].

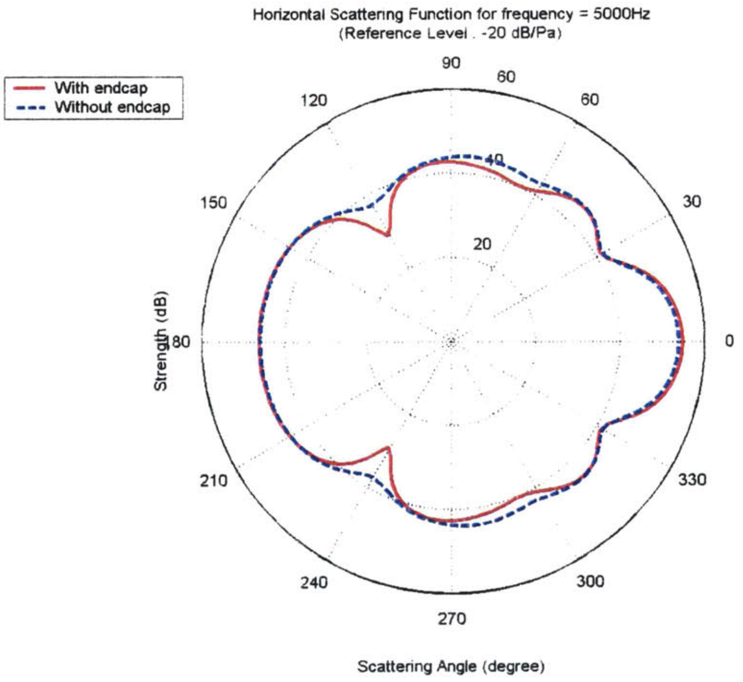
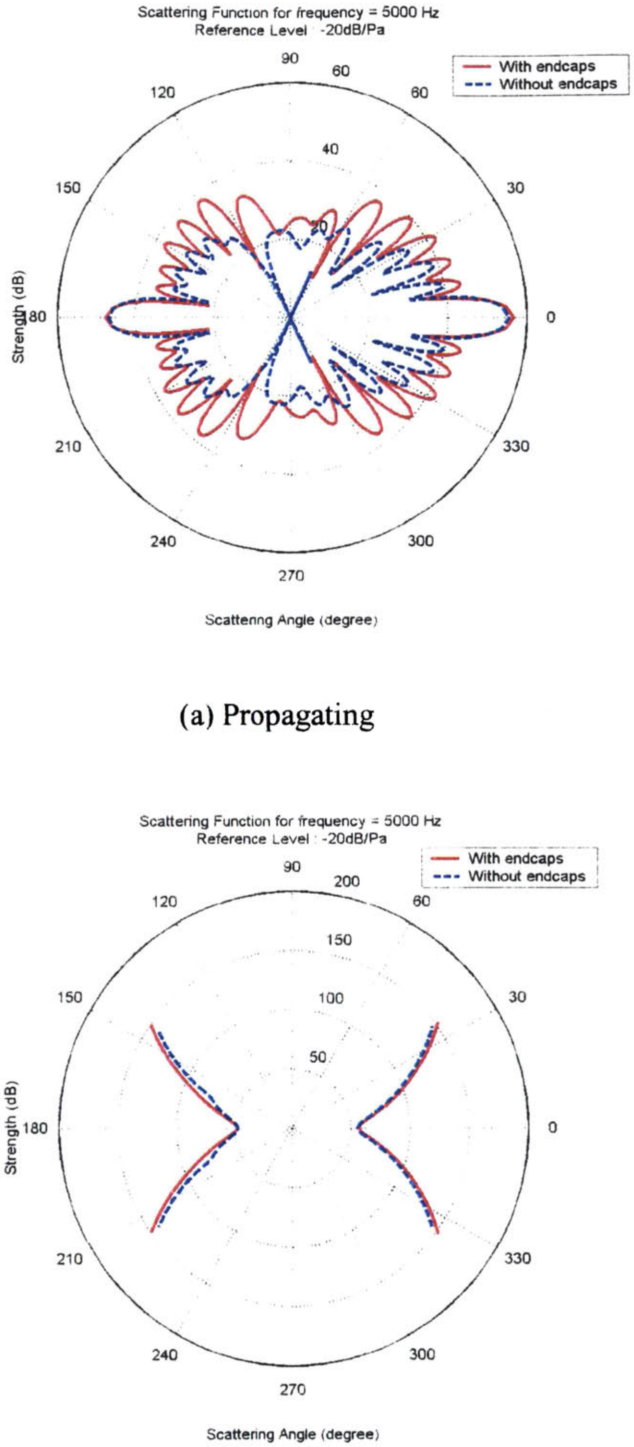


Figure 3-17 Horizontal Scattering function at 5000 Hz for steel cylinder



(a) Propagating

(b) Evanascent

Figure 3-18 Vertical Scattering function at 5000 Hz for steel cylinder

3.6 Scattering from a pitched cylinder

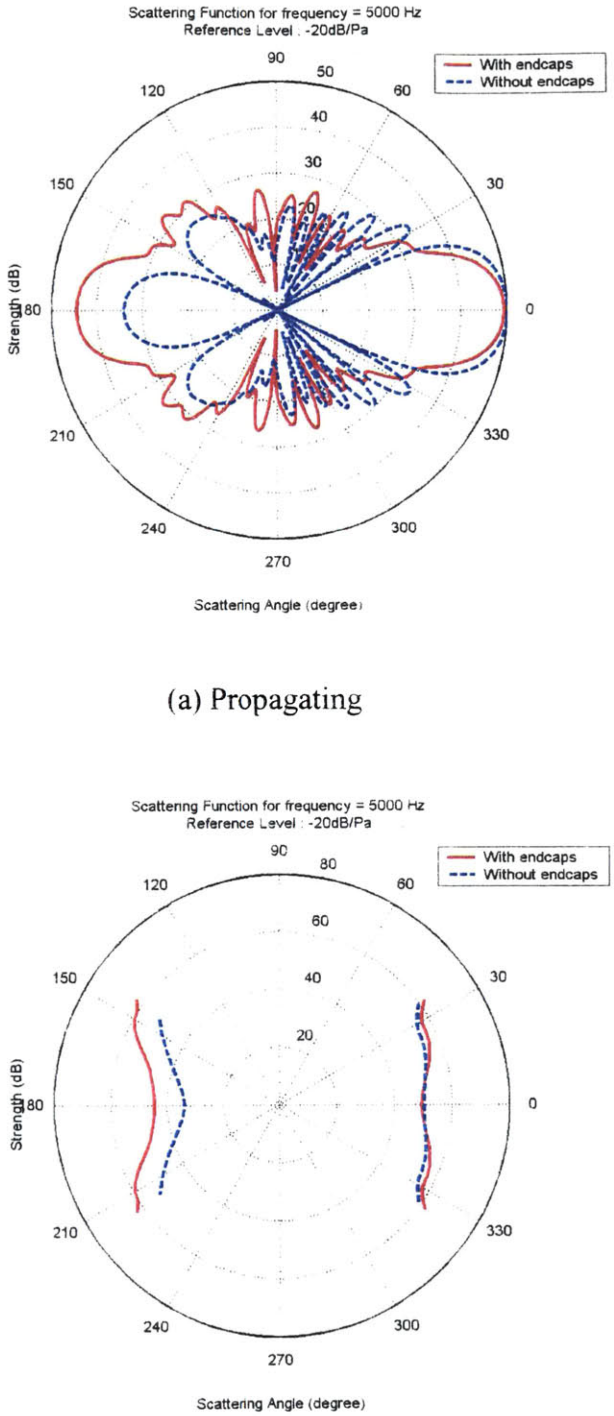
The steel cylinder with endcaps and material properties as described in Table 3-2 was analyzed with a different target orientation from the studies described so far in this thesis. The pitch and roll angle of the cylinder were consistent with that shown in Fig. 3-1. In summary, the target orientation and position with reference to the incident plane waves is as illustrated in the benchmark scenario for target scattering in free-space model (Appendix A).

The SCATT parameters defined in Table 3-3 were used, except the pitch was set to 90^{o}. Thus, the cylinder's axis (z-axis) is in the same direction as the wave propagation direction. The endcaps were thus normal to the wave direction. The scattering functions using the FE admittance matrix were computed for the cases with and without endcaps and plotted in Fig 3-19. As the vertical and horizontal scattering functions are identical in this case, due to symmetry, except for an additional evanescent field in the vertical scattering function, only the vertical scattering functions were plotted.

Contrary to the identical forward and backward scattering strength for the cylinders with and without endcaps when the target is not pitched (e.g. as observed in Fig. 3-18 of section 3.5.2), there is a significant difference in the strength in the backscatter when the target is pitched. It can be observed that the presence of endcaps result in a stronger backscatter due to the direct reflection from the endcaps. The difference is of order 10dB. In particular, the difference is very prominent in the range $120^{o}<\varphi_{V}<240^{o}$. This is consistent with the physics of sound reflection from a plate.

For the forward scatter direction, the scattering strength is comparable for both cases. Thus, it is apparent that the endcaps on the bottom of the cylinder (i.e. on the rear of the pitched cylinder) has relatively lesser effect on the scattering profile and strength.

The evanescent field is also stronger by about 5 dB for the cylinder with endcaps in the backscatter direction while the forward evanescent scatter is almost unaffected by the endcaps.



(a) Propagating

(b) Evanascent

Figure 3-19 Vertical Scattering function at 5000 Hz for steel cylinder

3.7 Scattering strength at resonance

The effect of resonance of cylinder on scattering is being studied and discussed in this section. The simply supported cylinder without endcaps and parameters as described in Table 3-2 was analyzed at 2 frequencies – 2160 Hz and 2387 Hz. 2160 Hz corresponds to resonance of the simply supported cylinder at n = 9 i.e. shell transverse vibration mode shape with 9 circumferential lobes. 2387 Hz corresponds to off-resonance frequency and ka value of 3. The scattering functions were computed for parameters according to Table 3-3. The horizontal scattering functions were plotted on Fig. 3-20 while vertical scattering functions, both propagating and evanescent, in Fig 3-21.

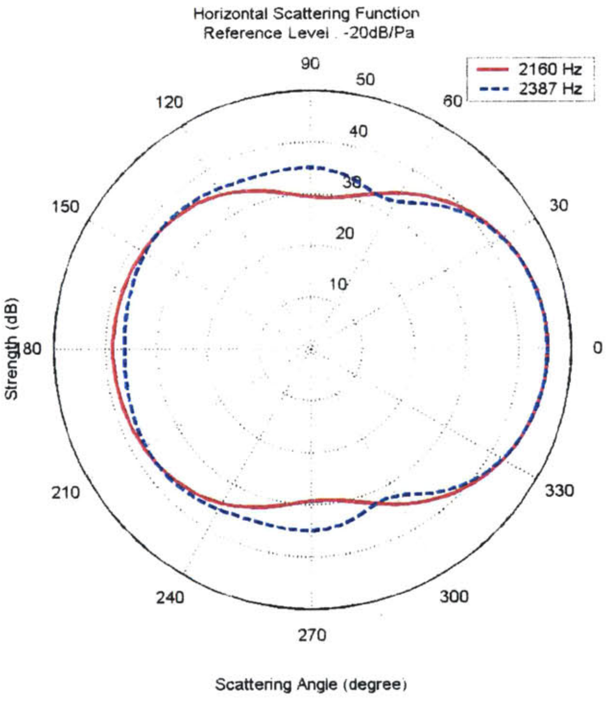
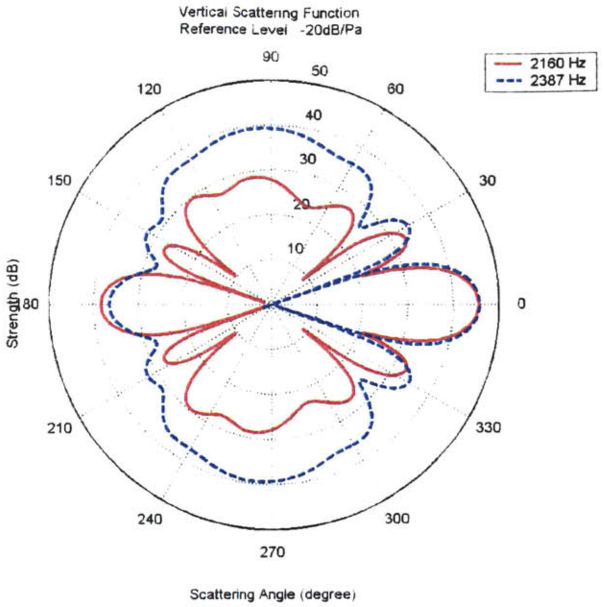
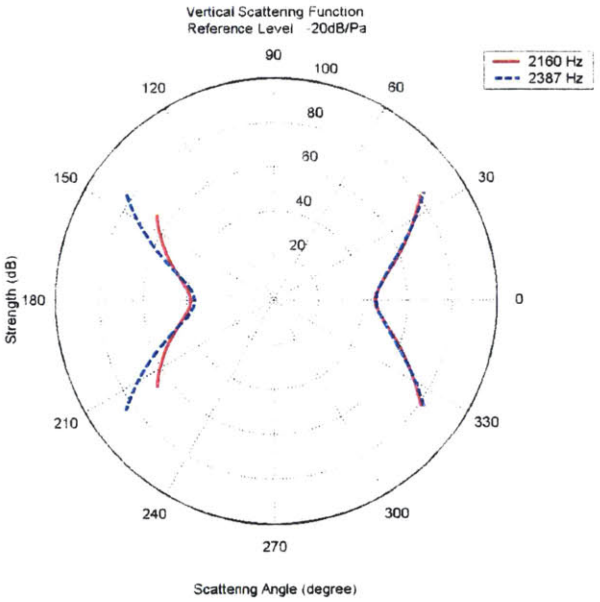


Figure 3-20 Comparison of horizontal scattering functions for simply supported steel cylinder at and off resonant frequency for n = 9



(a) Propagating



(b) Evanascent

Figure 3-21 Comparison of vertical scattering functions for simply supported steel cylinder at and off resonant frequency for n = 9

A sharp peak is expected in the plot of shell transverse displacement against frequency at frequencies corresponding to the eigenvalues to FE model, since damping is not considered in the FE formulation. Since the eigenvalues of the dynamic stiffness matrix do not agree exactly with the resonant frequencies of the continuous cylindrical model, 2160 Hz might not actually correspond to the eigenvalues. Thus, increase in scattering is not observed at 2160 Hz.

From these observations, implications could be drawn that it is important to include damping in the model to provide a better physical representation. It can be seen from the example above that unless the scattering is computed exactly at where the peak corresponding to a singularity in magnitude occurs, it is not possible to reflect contribution of shell resonance in scattering strength. Moreover, the eigenvlaues of the FE model do not correspond exactly to the resonance of a real damped structure, since damping shifts the resonant frequencies.

Thus, structural damping could be included in the FE model so that instead of a sharp peak at eigenvalues, a gentler slope leading to and falling off the peak of shell displacement at resonant frequencies would result. While sampling the scattering strength from the shell, increase in magnitude could be obtained since scattering would be higher for a band of frequencies containing the resonant frequencies, instead of a singularity. When coupled with the virtual source approach, fluid added mass and radiation damping would also be included that provides a complete model for the scattering computation.

To show the robustness of the above conclusions, 2 more cases were computed and the results are as shown in Fig. 3.22 to 3.25. Fig. 3.22 and 3.23 show the scattering from the cylinder at 1304 Hz (resonant frequency of the cylinder for n = 7) as compared to a slightly higher frequency at 1350 Hz. Fig. 3.22 and 3.23 show the scattering from the cylinder at 3230 Hz (resonant frequency of the cylinder for n = 11) as compared to a slightly higher frequency at 3300 Hz. No increase in scattering strength at resonant frequencies of the cylinder is observed for these cases, supporting the above argument and emphasizing the importance of considering damping in the structural model.

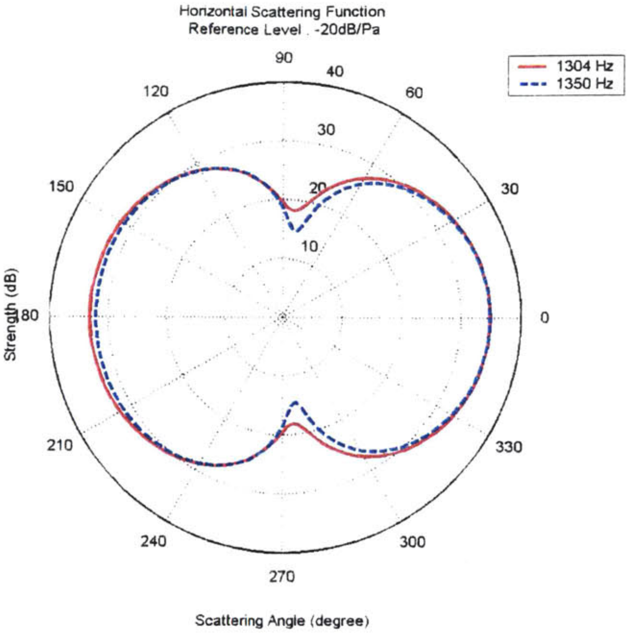
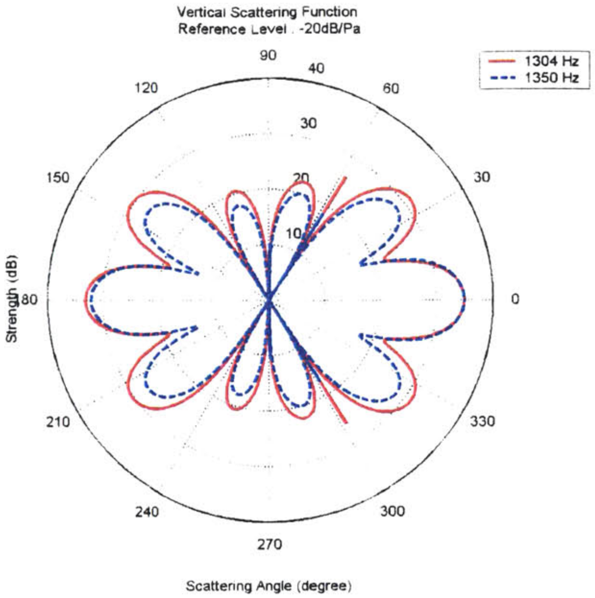
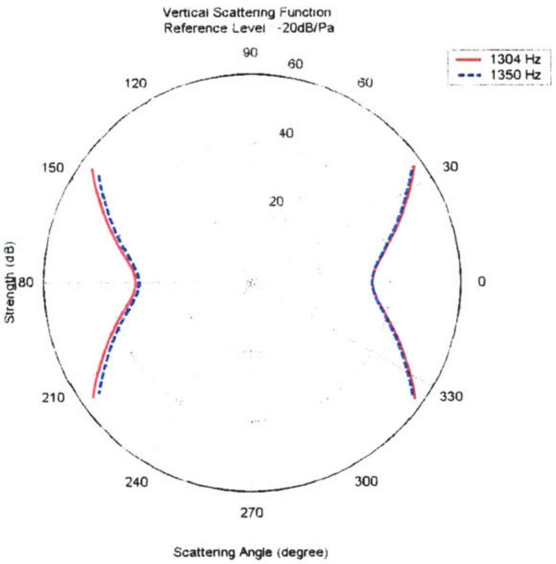


Figure 3-22 Comparison of horizontal scattering functions for simply supported steel cylinder at and off resonant frequency for n = 7



(a) Propagating



(b) Evanescent

Figure 3-23 Comparison of vertical scattering functions for simply supported steel cylinder at and off resonant frequency for n = 7

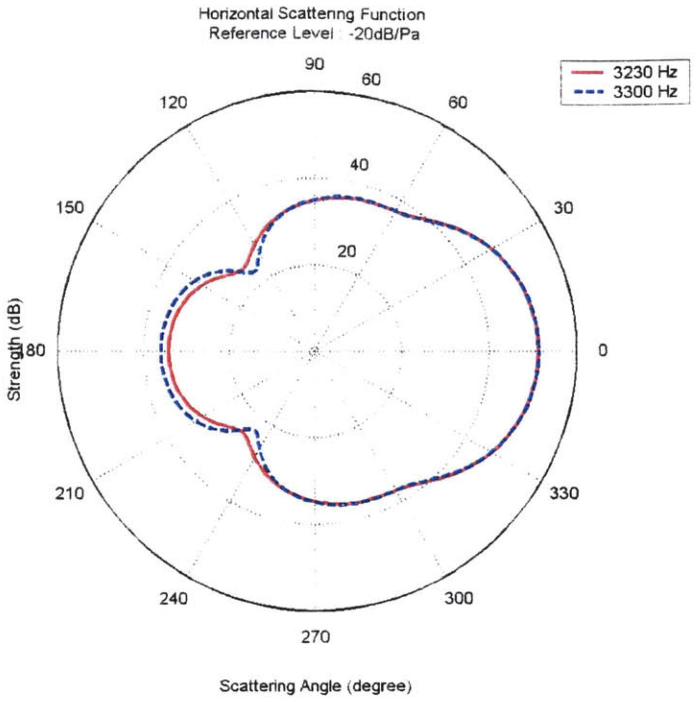
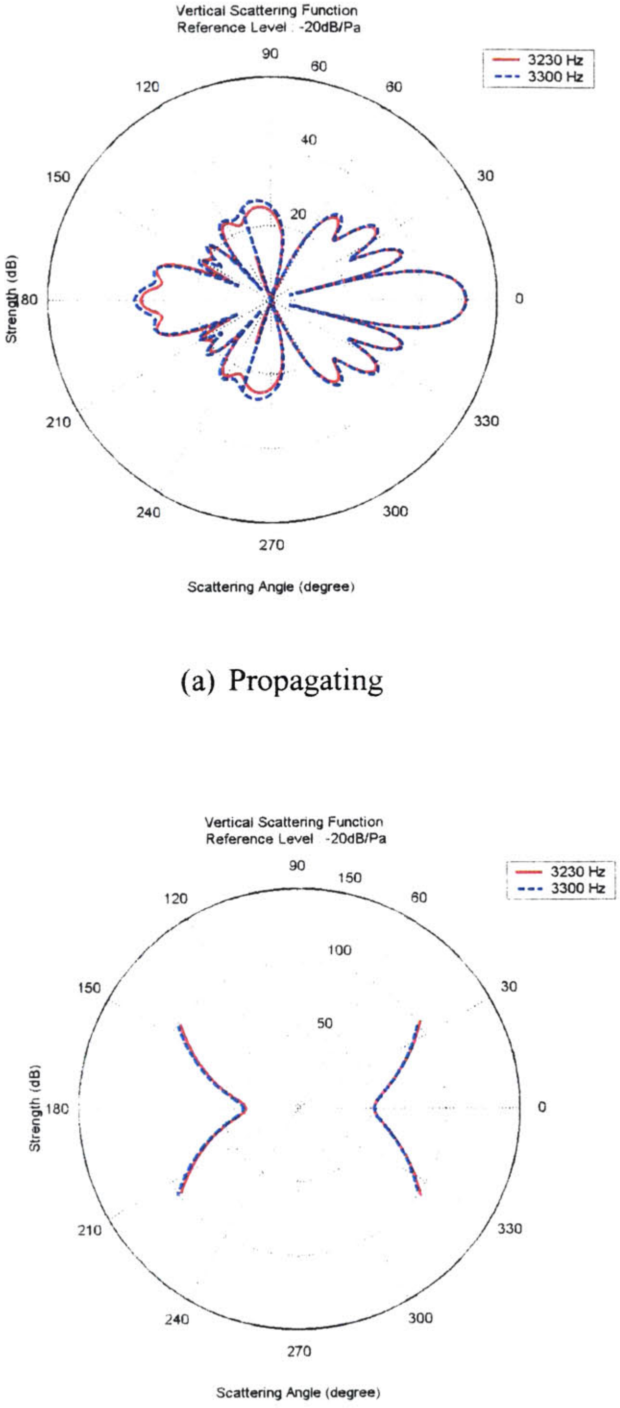


Figure 3-24 Comparison of horizontal scattering functions for simply supported steel cylinder at and off resonant frequency for n = 11



(a) Propagating

(b) Evanescent

Figure 3-25 Comparison of vertical scattering functions for simply supported steel cylinder at and off resonant frequency for n = 11

3.8 Discussion

This section provides a summary of the findings in this chapter and conclusions drawn from the observations.

Fig 3-2 shows the accurate prediction of resonant frequencies of the transverse cylindrical vibration modes with more than 2 circumferential lobes by the 2D axi-symmetric conical shell FE model. However, no eigenvalue was observed for distinct mode shapes that correspond to transverse vibration modes with 2 or less circumferential lobes in the FE dynamic stiffness matrix, and it thus does not yield frequency for these modes. This is due to the coupling of lower order modes that results in 3-D mode shapes that may not be very well represented by a 2-D axi-symmetric conical shell model.

Moreover, from Fig. 3-4 to 3-7, it can be gathered that the FE code predicts displacement profile of a simply supported cylinder very closely to analytical predictions, but yields lower magnitude due to the stiffness of the elements that constrain the target to a finite degree-of freedom system. When implemented with the virtual source approach, it computes the scattering functions of pressure-release and rigid cylinders with endcaps that agree with the computation by requiring the pressure and displacement to vanish on the target surface respectively (Fig. 3-13 and 3-14). It also predicts scattering accurately for higher frequencies but start to show some discrepancies from the hybrid analytical/ virtual source approach at low frequencies. As a reference, for a 0.01m thick, 2 m long, 0.3m radius steel cylinder, discrepancies start to appear for frequency in the range of 1000 to 2000 Hz, as observed in Fig.3-11. The discrepancies tie in closely with earlier observations about the estimation of natural frequencies for lower order circumferential modes.

For the cylinders, it was also observed that the absence of endcaps produce stronger vertical scattering in the endfire direction and backscatter for plane waves coming in a direction normal to the axial direction of the cylinder, as seen in Fig 3-15 and 3-16, for frequency of 2387 Hz. This phenomenon has been understood as a result of the requirement to satisfy the conditions for a broader boundary for the cylinder with endcaps in the form of an exterior problem. Physically, such factors as difference in diffraction pattern and the interaction of the elastic plates with the medium in the case of the cylinder with endcaps can also account for the lower scattering.

At higher frequencies, due to the dominance of specular reflection in the scattered wavefield, the scattering functions, relative to that at lower frequencies, are more identical as illustrated in Fig. 3-17 and 3-18 for the case of 5000 Hz.

Agreement with physical prediction of the scattered wavefield was observed for the hybrid FE/ virtual source approach as shown in Fig. 3-19 for the case of plane waves coming in a direction parallel to the axial direction of the cylinder. The cylinder with endcaps produce a stronger backscatter that that without endcaps as the incident wavefield was directly reflected in the normal direction backwards by the flat endcaps.

Finally, as opposed to the general expectation that the scattered wavefield should be stronger at cylinder's resonance, the hybrid FE/ virtual source approach developed so far in this thesis did not produce stronger scattering functions for the cylinder at resonance. This is due to the singularity in shell displacement at the eigenvalues of FE model and the difference between eigenvalues and the actual resonant frequencies of the continuous cylindrical model. Thus, strong scattering as anticipated can only be obtained at the peaks corresponding to the singularities, which is very hard to achieve due to precision of computation. A feasible solution is to include structural damping in the FE model, as discussed. Considering damping results in stronger scattering for a band of frequencies containing the resonant frequencies, instead of a singularity. When coupled with the virtual source approach, fluid added mass and radiation damping would also be included that provides a complete model for the scattering computation.

4 COMPUTATIONAL RESULTS FOR ELASTIC CONE

In this chapter, computation of scattering by FE/ virtual source approach will be implemented for conical shells with endcaps. Scattering functions will be compared with pressure-release and rigid targets computed internally in SCATT using the virtual source approach and by requiring the pressure and normal displcament to vanish respectively. Scattering functions for elastic conical shells will also be computed. A discussion and comparison of results will conclude this chapter.

In this section, unless otherwise stated, the material properties and geometry of the cones studies are stated in Table 4-1. Moreover, the target orientation and position with reference to the incident plane waves is as illustrated in the benchmark scenario for target scattering in free-space model (Appendix A).

Table 4-1 Material properties and geometry of steel cone with endcaps for studying acoustic scattering

Thickness (m)	0.01
Top Radius (m)	0.2
Bottom Radius (m)	0.4
Height (m)	2
Young's Modulus (N/m^2)	195
Density(kg/m^3)	7700
Poisson's ratio	0.28

4.1 Acoustic scattering from pressure release and rigid cone with endcaps

In this section, the evaluation of the computation of acoustic scattering from rigid and pressure release cones with endcaps will be discussed. The entire cone capped with flat plates on both ends was discretized with the elements as described in Chapter 2 to represent the target surface boundary conditions and the scattering function computed for evaluation.

For the rigid cone, the structural sound speed and density was set to approach infinity so that the target material represents a rigid boundary condition on the surface. All other parameters are unchanged. On the other hand, scattering functions of an equivalent target with rigid surface boundary condition can be computed analytically by requiring the total displacement on the target surface to vanish in the SCATT virtual source approach. Fig 4-1 and Fig. 4-2 show the comparison of horizontal and vertical scattering functions as computed by both the 2 approaches described. The frequency was 2387 Hz corresponding to a ka (product of wavenumber and cone mean outer radius) value of 3. The scattering angles, φ_H and φ_V are defined as before.

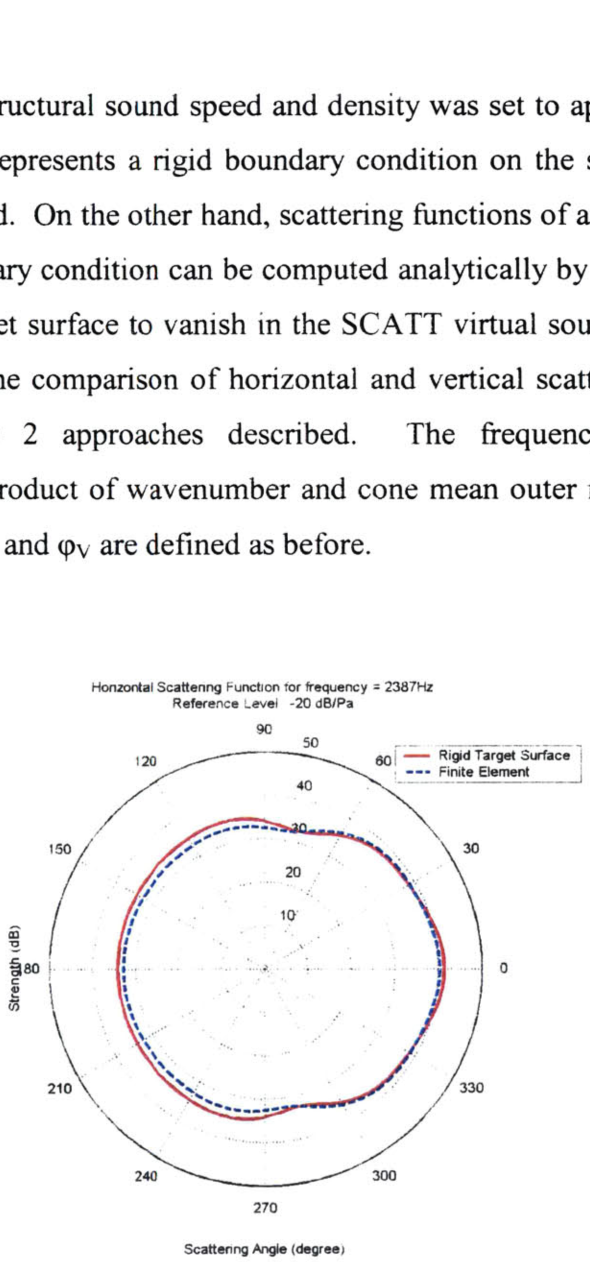
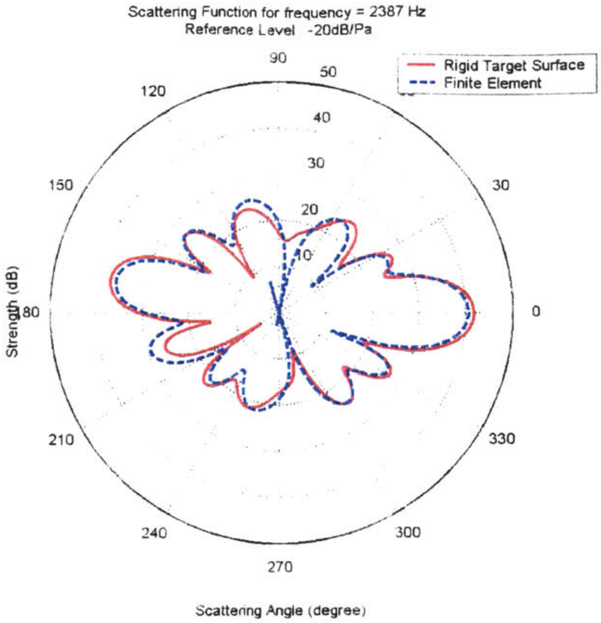
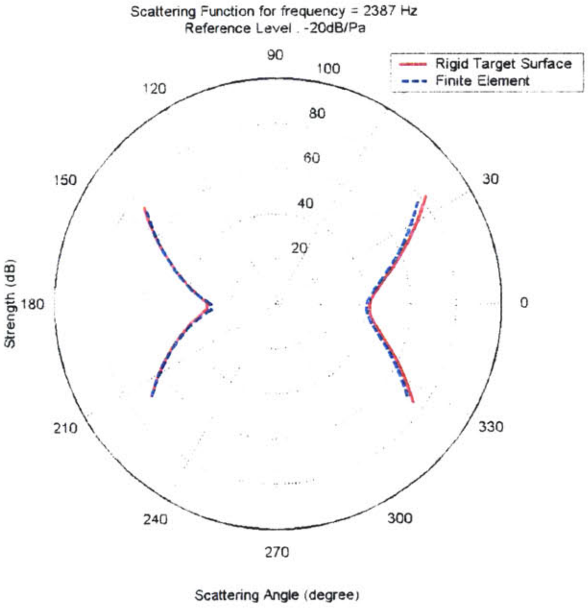


Figure 4-1 Horizontal Scattering function at ka =3 for rigid cone with endcaps



(a) Propagating



(b) Evanascent

Figure 4-2 Vertical Scattering function at ka =3 for rigid cone with endcaps

The Figure shows the close agreement in scattering functions computed between the 2 approaches, illustrating the validity of FE generation of admittance matrix for virtual source computation in this aspect.

A similar approach was applied for the pressure-release target surface. For the FE approach, the density and Young's modulus were respectively set to 10^{-23} and 10^{-69} times that of the material in Table 4-1. This corresponds to near-zero structural sound speed and density and thus represents a pressure-release target. On the other hand, scattering functions of an equivalent target with pressure-release surface boundary condition can be computed analytically by requiring the total pressure on the target surface to vanish in the SCATT virtual source approach.

Fig. 4-3 and 4-4 show the comparison of horizontal and vertical scattering functions by both approaches and the close agreement between them.

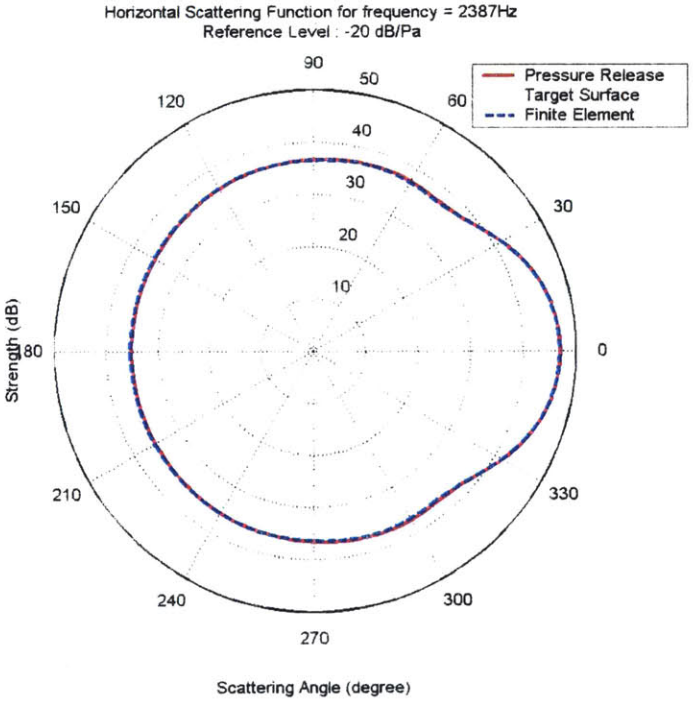
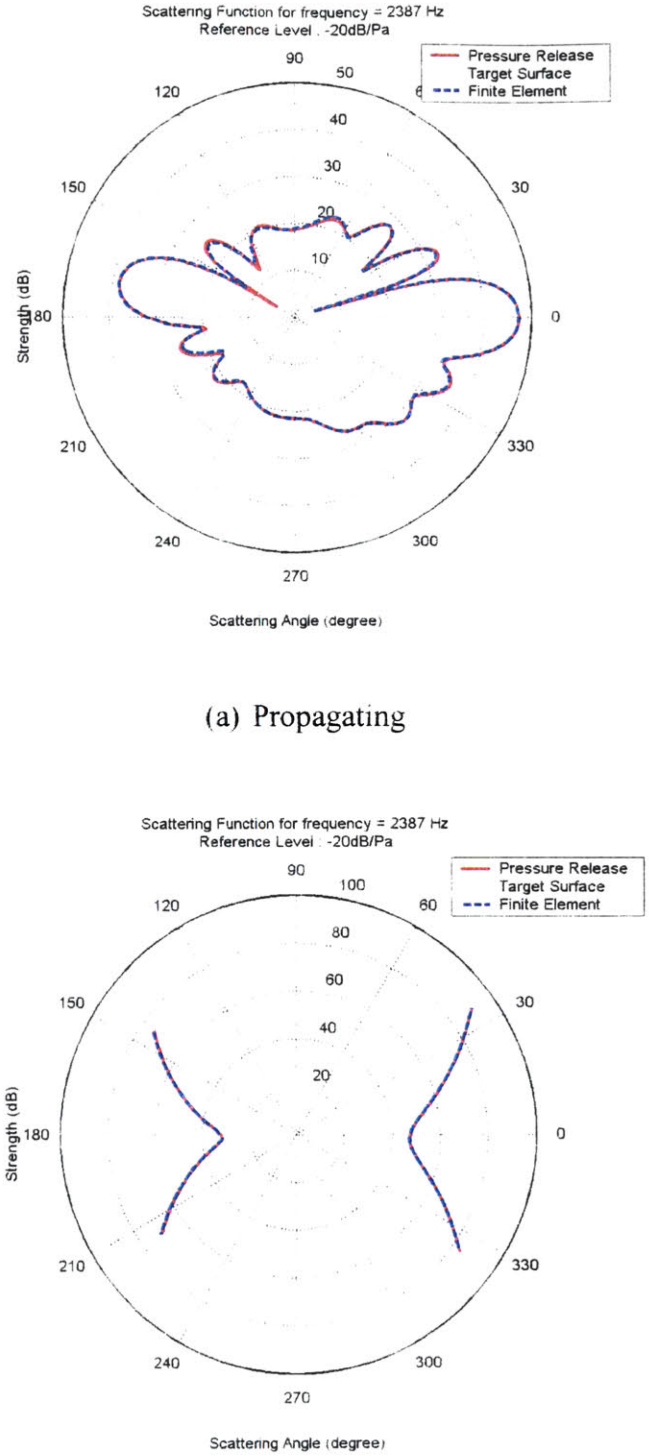


Figure 4-3 Horizontal Scattering function at ka =3 for pressure cone with endcaps



(a) Propagating

(b) Evanascent

Figure 4-4 Vertical Scattering function at ka =3 for pressure-release cone with endcaps

4.2 Acoustic scattering from elastic cone with endcaps

For the material and geometry as described in Table 4-1, the acoustic scattering for the steel cone with endcaps was computed using the admittance matrices generated from the FE code. Figs. 4-5 and 4-6 show the scattering functions in the horizontal and vertical direction respectively for the cone at a frequency of 2387 Hz. The scattering functions for the cylinder as described in section 3.5.2 and material properties and geometry in Table 3-2 was plotted as well for comparison. The scattering angles, φ_H and φ_V take on the earlier definitions in section 3.5.2.

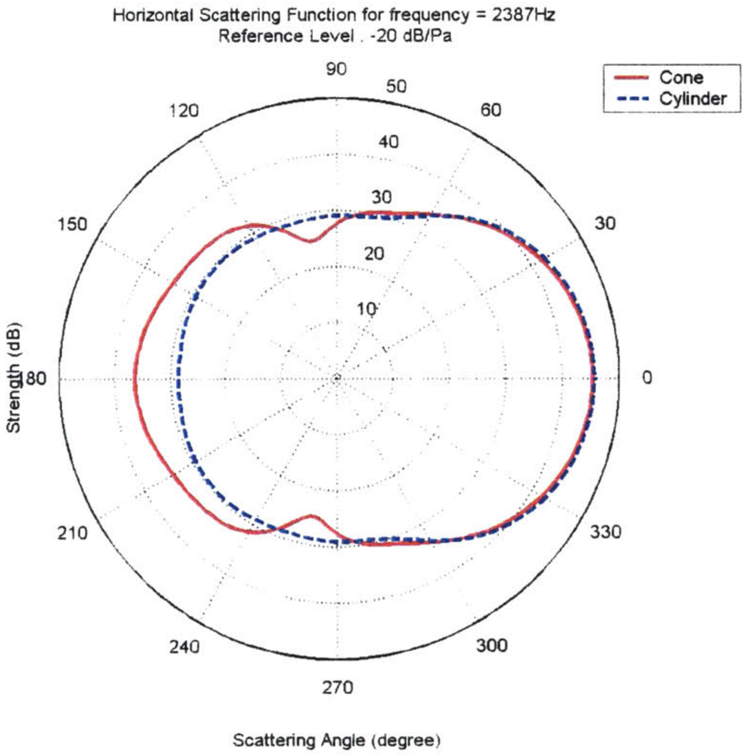
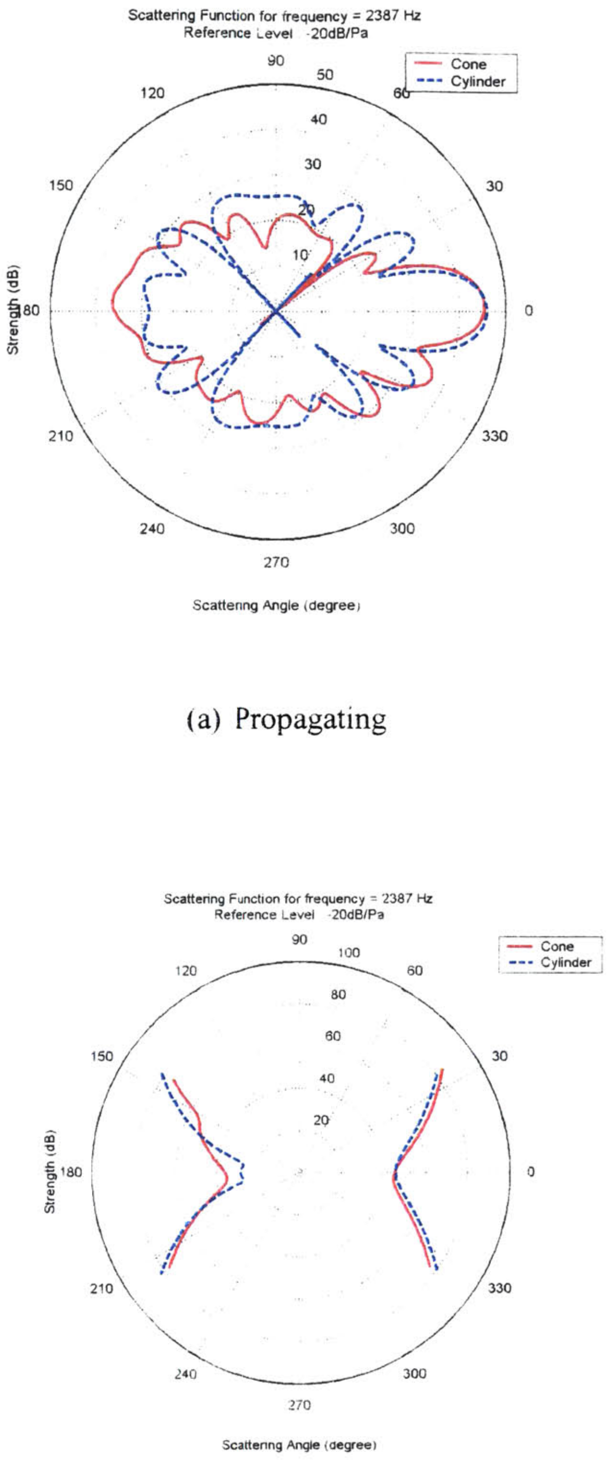


Figure 4-5 Horizontal scattering functions at ka =3 for steel cone with endcaps



(a) Propagating

(b) Evanescent

Figure 4-6 Vertical scattering functions at ka =3 for steel cone with endcaps

From the plots, it can be observed that the conical shell produces a very much stronger backscatter of close to 10dB than a cylinder of equivalent size. The forward scatter is, on the other hand, very similar.

In particular, the backscatter is strongest in the upward direction, which is in agreement with intuition that the oblique angle of the shell produces strong specular reflection in the upward backscatter direction. The forward scatter is also stronger in the upward direction for the same reason. In all, the shape of the target has resulted in more energy to be scattered in the upward than downward direction, relative to that by a cylinder.

4.3 Discussion

In this section, it has been shown that the hybrid FE/ virtual source approach computes the scattering functions of pressure-release and rigid cones with endcaps that agree with the computation by requiring the pressure and displacement to vanish on the target surface respectively (Fig. 4-1 to 4-4).

Moreover, for a steel cone of 0.01m shell thickness, 0.3m radius and 2 m height, it was also observed that more backscatter was generated when compared to that of a steel cylinder of equivalent size. This is particular so in the $150^o<\varphi_V<180^o$ and $120^o<\varphi_H<240^o$ range, as observed in the Fig. 4-5 and 4-6, due to the oblique angle of the conical profile. Moreover, the shape of the target has resulted in more energy to be scattered in the upward than downward direction, relative to that by a cylinder.

5 CONCLUSIONS

In this section, a summary of the research as described in this thesis will be presented. This is followed by a review of the research objectives set up in section 1.4 of this thesis. Finally, future work in this area will proposed.

5.1 Summary

The theory of behind the hybrid FE/ virtual source approach has been described in this thesis. The broad range of engineering techniques and theories involved in implementing the hybrid FE/ virtual source approach has been demonstrated. Computational and ocean acoustic tools are necessary to derive the incident and radiated field from the target. The scattering of the target has to be derived by computational structural dynamics techniques and applying the physics of sound radiation. These tools blend together to form the hybrid FE/virtual source approach that computes scattering efficiently.

The validation of the hybrid FE/ virtual source approach has also been illustrated. Detailed analyses on the deformation profile of the FE model and scattering computation have shown close agreement between this approach and the hybrid analytical/ virtual source approach. Furthermore, the approach also demonstrates consistency with the scattering predicted for extreme cases, i.e. rigid and pressure-release cylinders, by requiring the total displacement and pressure to vanish on the target surface. Such issues as effect of resonance on scattering and validity of computational results remain open and require further work.

The scattering from elastic cone with endcaps has been computed. Agreement of results by the hybrid FE/ virtual source approach with that by requiring the total pressure and displacement to vanish for pressure release and rigid targets has been demonstrated.

5.2 Research objectives

The research objectives as stated in section 1.4 have been achieved as follows:

a. The FE method for elastic axi-symmetric conical shells had been implemented. The method had been programmed in Matlab. The manual for using the Matlab code has also been written as shown in Appendix C. Moreover, the Matlab .m files are attached in Appendix D. The code has also been proven to be able to produce the necessary input files to OASES/SCATT to generate the scattering functions as illustrated throughout this thesis.

b. The benchmarking and validation of the code has been conducted with the hybrid analytical/ virtual source approach. Firstly, the code predicted natural frequencies close to the theoretical analysis for higher order modes with more than 2 circumferential transverse vibration lobes. Secondly, it produced displacement profile that conforms to analytical results. The scattering functions are also in agreement with the hybrid analytical/ virtual source approach, with discrepancies observed at lower frequencies. In exact terms, discrepancies start to appear for frequency in the range of 1000 to 2000 Hz for a 0.01m thick, 2 m long, 0.3m radius steel cylinder without endcaps.

c. The scattering functions have also been compared with the SCATT/OASES virtual source approach for pressure release and rigid cylinders and cones. The difference in the 2 approaches is that for the hybrid FE/virtual source approach, this is achieved by setting the structural sound speed and density to zero/ infinity while for the SCATT/OASES virtual source approach, this is achieved by requiring the pressure/ displacement to vanish. The two approaches have shown to agree with each other in this research.

d. Agreement with physical prediction of the scattered wavefield was observed for the hybrid FE/ virtual source approach for a different target orientation. In the case of plane waves coming in a direction parallel to the axial direction of the cylinder, stronger backscatter was observed from the cylinder with endcaps than that from one without endcaps. Wavefield was directly reflected in the normal direction backwards by the flat endcaps.

e. As illustrated in this thesis, the hybrid FE/ virtual source approach developed so far in this thesis did not produce stronger scattering functions for the cylinder at resonance. This is due to the singularity in shell displacement at the eigenvalues of FE model and the difference between eigenvalues and the actual resonant frequencies of the continuous cylindrical model. Structural damping has to be included in the FE model to improve the scattering estimation in this aspect.

5.3 Future work

The code has shown to have good potential for adaptation to compute scattering over other axi-symmteric shapes using conical shells and circular plates as building blocks and the hybrid FE/ virtual source approach.

Future work in this area includes:

a. Looking into the issue of computation of vibration response for lower order modes by the FE model. This will enhance the ability of the hybrid FE/ virtual source approach to yield accurate results for low frequencies.

b. Considering structural damping in the FE model. This results in stronger scattering for a band of frequencies containing the resonant frequencies, instead of a singularity. When coupled with the virtual source approach, fluid added mass and radiation damping would also be included that provides a complete model for the scattering computation.

c. Testing the robustness of the approach by running the simulation for a broad spectrum of frequencies, geometry and material properties. Subsequently, agreement with physical prediction and theory can be tested with analytical models.

d. Computation of scattering from other axi-symmetric targets using conical shells and endcaps as building blocks. Subsequently, experimental data and numerical techniques (such as FEM/BEM) can be used to verify the approach.

6 BIBLIOGRAPHY

1. M.C. Junger and D. Feit, *Sound, structures and their interaction*, The MIT Press, second edition, 1986.

2. Earl G. Williams, *Fourier Acoustics*, Academic Press, 1999.

3. Gordon C. Everstine and Francis M. Henderson, "Coupled finite element/boundary element approach for fluid-structure interaction", *Journal of Acoustical Society of America*, v. 87 (5), pgs 1938-1947, 1990.

4. Burnett, D.S. and Zampolli, M., "Development of a finite-element, steady-state, 3-D acoustics code for target scattering", Tech. Rep. SR-379, NATO Res. Ctr., La Spezia, Italy (2003).

5. Burnett, D.S., "Finite-element methods for structural acoustics: physics, mathematics and modeling", Proc. 10th Internat. Cong. On Sound and Vibration, Stockholm, Sweden, 2003.

6. Finn B. Jensen, William A. Kuperman, Michael B. Porter, Henrik Schmidt, *Computational Ocean Acoustics*, AIP Press, 2000.

7. Henrik Schmidt, SAFARI: "Seismic-acoustic fast-field algorithm for range independent environments. User's Guide", Rep. SR-113, SACLANT Undersea Res. Ctr., La Spezia, Italy (1988).

8. Henrik Schmidt, OASES User Guide and Reference Manual, Department of Ocean Engineering, Massachusetts Institute of Technology, 2003.

9. Michael B. Porter, KRAKEN Normal Mode Program, http://oalib.saic.com/Modes/index.html.

10. Mario Zampolli, David S. Burnett, Finn B. Jensen, Alessandra Tesei, Henrik Schmidt and John B. Blottman III, "A finite-element tool for scattering from localized inhomogeneities and submerged elastic structures", Proceedings of High-Frequency Ocean Acoustics Conference 2004.

11. L. Godinho, J. António and A. Tadeu, "The scattering of 3D sound sources by rigid barriers in the vicinity of tall buildings", *Engineering Analysis with Boundary Elements* 26(2002), pgs 781-787.

12. Henrik Schmidt, "Virtual source approach to scattering from partially buried elastic targets", Proceedings of High-Frequency Ocean Acoustics Conference 2004.

13. C.T.F. Ross, "Lobar buckling of thin-walled cylindrical and truncated conical shells under external pressure", *Journal of Ship Research*, v. 18, No. 4, pgs 272-277, Dec 1974.

14. C.T.F. Ross, "Finite elements for vibration of cones and cylinders", *International Journal for numerical methods in engineering*, v. 9, pgs 833-845, 1975.

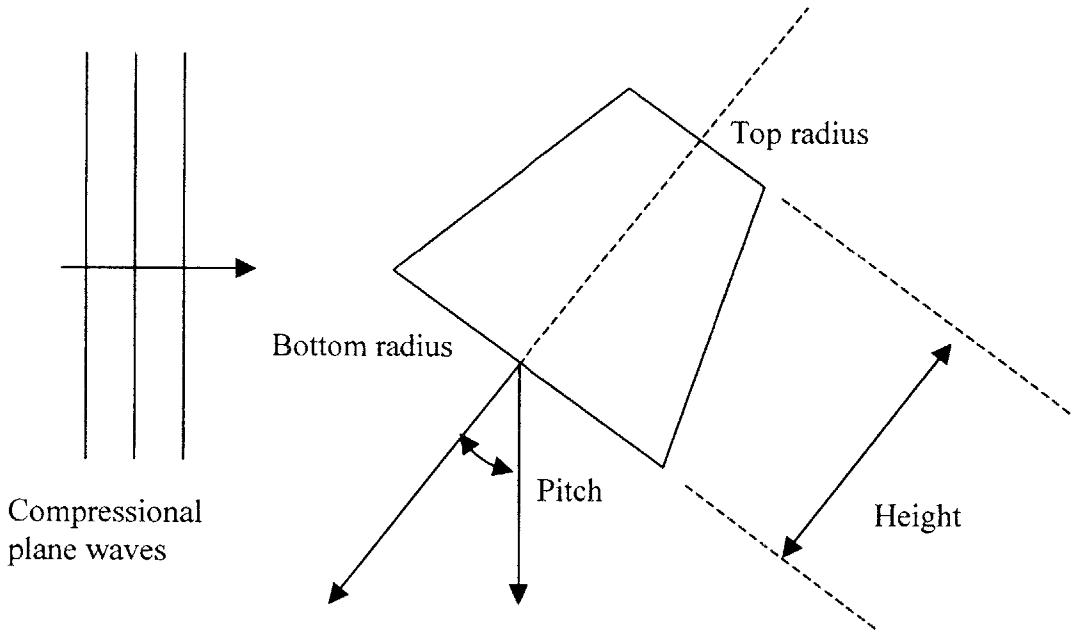
15. G. C. Pardoen, "Static vibration and buckling analysis of axisymmetric circular plates using finite elements", *Computers and structures*, v. 3, pgs 355-375, 1973.

16. R. D. Blevins, *Formulas for natural frequency and mode shape*, Robert E. Krieger Publishing, second edition, 1984.

17. C.T.F. Ross, *Advanced Applied Finite Element Methods*, Horwood 1998.

7 APPENDIX A

Appendix A – Benchmark scenario for target scattering model in free-space



Medium sound speed : 1500m/s
Medium density : 1000kg/m^3

For cylinder:
top radius = bottom radius

8 APPENDIX B

Appendix B - Explanation on .dat file used with oast command for free-space scattering function computation

The .dat file is a text file with lines containing data defining the ocean environment and other computational parameters.

Below is a table containing definitions of the parameters entered in each line. More details can be referred to the OASES manual [8].

Line	Description
1	Title printed on all graphics output generated by OAST
2	Output Options (Refer Pg 33 of manual [8] for more details
3	First frequency (Hz) Last frequency (Hz) Number of frequencies Integration contour offset (dB/Λ)
4	Number of layers, including halfspaces
5	Depth of upper interface of layer (m) Compressional speed of layer (m/s) Shear speed of layer (m/s) Compressional attenuation of layer (dB/Λ) Shear attenuation of layer (dB/Λ) Density (g/cm^3) RMS value of interface roughness (m) Correlation length of roughness
: (6) : : :	1 line per layer to define the environment in each layer
7	Source depth (m)
8	Depth of first receiver (m) Depth of last receiver (m) Number of receivers Plot output increment
9	Minimum phase velocity (m/s) Maximum phase velocity (m/s)
10	Number of wavenumber samples (-1 for automatic) First sampling point Last sampling
11	Minimum range on plots Maximum range on plots Length of x-axis for all plots

	Distance between tick marks
12	Horizontal range from source to center of scatterer in km Interface number where a scatter is located in the middle depth in the layer Number of patch grids in x-direction (meaningless in this case for target) Number of patch grids in y-direction (meaningless in this case for target) Diameter of spherical target (m) (meaningless in this case for cylinder and cone) Length of patch in y direction (m) (meaningless in this case for target)

A sample of .dat file is as shown below:

```
vsccvs 3d test 0 deg
N P E
1000 1000 1 0
4
0 1500 0 0 0 1.0 0 0
0 1500 0 0 0 1.0 0 0
20 1500 0 0 0 1.0 0 0
22 1500 0 0 0 1.0 0 0

10
20 20 1 1
1500 1500
-1 1 1
0 0.1 20 0.02
0.03077 3 0 0 1.06 0
```

9 APPENDIX C

Appendix C – 2-D Axi-symmetric Conical Shell Finite Element Code Manual

Setup

1. Create a directory where you want to run the code and copy all the m. files to that directory.

Running the code

1. Start Matlab and go to the directory where you store the .m files.
2. In the command window, type the following:

```
>> cone_main
```

3. Follow the prompt to enter the following parameters:
    a. Thickness of tapered tube material (m)
    b. Thickness of top endcap (m)
    c. Thickness of bottom endcap (m)
    d. Diameter of top endcap (m)
    e. Diameter of bottom endcap (m)
    f. Height of cone (m)
    g. Young's Modulus of tapered tube material (GPa)
    h. Density of tapered tube material (kg/m^3)
    i. Young's Modulus of endcaps' material (GPa)
    j. Density of endcaps' material (kg/m^3)
    k. Poisson's ratio of cone and endcaps'material
    l. ka (product of wavenumber and cone's mean outer radius)

Post-processing

1. 2 files with default names as follow will be generated at the end of the simulation:
    a. impet.adm – Admittance matrix data file
    b. impet.srf – Nodal coordinates and normal vectors data file
2. Rename the files to the same name as the .dat and .sck files used in the SCATT vsccvs_simp command. (See SCATT manual for more details)
3. The 2 files generated comply with all definitions and requirements set up in SCATT.

10 APPENDIX D

Appendix D – FE 2-D axi-symmetric conical shell with endcaps admittance matrix generation code in the form of Matlab .m files.

This appendix contains all the Matlab .m files for the FE code. The primary file "cone_main.m" file is shown first. This file controls all the computation and calling of functions. The rest of the files are in the form of function files and are sorted in alphabetical order. Each file starts on a fresh page.

```
%Script file : cone_main.m
%
%Main file for computing acoustics response for a cone
%
%*Define variables*
%tube_thic -- Thickness of tapered tube (assume uniform) (m)
%topl_thic -- Thickness of top plate (assume uniform) (m)
%bopl_thic -- Thickness of bottom plate (assume uniform) (m)
%top_diam -- Top diameter of cone (m)
%bot_diam -- Bottom diameter of cone (m)
%cone_ht -- Height of cone (m)
%
%*Material properties*
%youn_cy -- Young's modulus of tapered cylinder material (x10e9 N/m2)
%dens_cy -- Density of tapered cylinder material (kg/m3)
%youn_pl -- Young's modulus of top and bottom plate material (x10e9 N/m2)
%dens_pl -- Density of top and bottom plate material (kg/m3)
%pois -- Poisson's ratio of material
%nwav -- Wave number in consideration
%
%*Computed variables*
%tube_leng -- Length of each element on the cylindrical tube (m)
%topl_leng -- Length of each element on top end plate (m)
%bopl_leng -- Length of each element on bottom end plate (m)
%tradi -- Top radius of cone (m)
%bradi -- Bottom radius of cone (m)
%taper - Taper angle of cone
%total_dof -- Total no. of dofs
%top_n -- Total no. of dofs on top plate
%tub_n -- Total no. of dofs on tapered tube
%bot_n -- Total no. of dofs on bottom plate
%
%*Others*
%topl_num -- No. of elements at top end circular plate
%bopl_num -- No. of elements at bottom end circular plate
%tube_num -- No. of taper cylindrical elements
%ka -- ka of the excitation frequency
%frad -- Excitation frequency (rad/s)
%
%*Matrices*
%kmat_tu -- Stiffness Matrix (tube)
%mmat_tu -- Mass Matrix (tube)
%kmat_to -- Stiffness Matrix (top plate)
%mmat_to -- Mass Matrix (top plate)
%kmat_bo -- Stiffness Matrix (bottom plate)
%mmat_bo -- Mass Matrix (bottom plate)
```

```
%
%*Global Matrices*
%k_globe -- Global Stiffness Matrix
%m_globe -- Global Mass Matrix
%
%Invoke the file to request info from user and define variables
%
clear;
[tube_thic, topl_thic, bopl_thic, top_diam, bot_diam, cone_ht, youn_cy, dens_cy,
youn_pl, dens_pl, pois, ka] = cy_parameters_define;
%
%Compute variables
%
[tube_leng, topl_leng, bopl_leng, tradi, bradi, frad, topl_num, bopl_num, tube_num,
taper] = para_compute_cy (cone_ht, top_diam, bot_diam, ka);
%
for nwav = 0:10
%
%Get the mass matrix for tapered tube
%
mmat_tu(:,:,nwav+1) = tube_mass(dens_cy, tube_thic, tradi, bradi, tube_leng, tube_num,
taper, nwav);
%
%Get the mass matrix for top plate
%
mmat_to(:,:,nwav+1) = toplate_mass(topl_thic, tradi, dens_pl, topl_num, nwav);
%
%Get the mass matrix for bottom plate
%
mmat_bo(:,:,nwav+1) = boplate_mass(bopl_thic, bradi, dens_pl, bopl_num, nwav);
%Get the stiffness matrix for tapered tube
%
kmat_tu(:,:,nwav+1) = tube_stiff(tube_thic, youn_cy, pois, tradi, bradi, tube_num,
tube_leng, taper, nwav);
%
%Get the stiffness matrix for top plate
%
kmat_to(:,:,nwav+1) = toplate_stiff(topl_thic, tradi, youn_pl, pois, topl_num, nwav);
%
%Get the stiffness matrix for bottom plate
%
kmat_bo(:,:,nwav+1) = boplate_stiff(bopl_thic, bradi, youn_pl, pois, bopl_num, nwav);
%
%Combine all matrices to form global mass and stiffness matrices
%
```

```
[k_globe(:,:,nwav+1), m_globe(:,:,nwav+1), total_dof, top_n, tub_n, bot_n] =
globecomp(kmat_tu(:,:,nwav+1), mmat_tu(:,:,nwav+1), kmat_to(:,:,nwav+1),
mmat_to(:,:,nwav+1), kmat_bo(:,:,nwav+1), mmat_bo(:,:,nwav+1));
%
%To derive the dynamic flexibility matrix from global mass and stiffness matrices
%
flex(:,:,nwav+1) = flexglobe (k_globe(:,:,nwav+1), m_globe(:,:,nwav+1), frad, top_n,
tub_n, bot_n, total_dof, taper);
%
%Orientate the flex matrix to agree with coordinate system in SCATT
%
orflex(:,:,nwav+1) = orient(flex(:,:,nwav+1), topl_num, tube_num, bopl_num);
%
%Allocate orflex matrix to appropriate layer in flex3d matrix
%
flex3d(:, :, nwav+1) = orflex(:,:,nwav+1);
end
%
%Compute 3-D impedance matrix
%
[immata, fluidense] = stiffglobe(flex3d, topl_num, topl_leng, tube_num, tube_leng,
bopl_num, bopl_leng, taper, frad);
%
%Copy data to file
%
filegen(immata, fluidense, topl_leng, bopl_leng, tube_leng, tradi, bradi, topl_num,
bopl_num, tube_num, cone_ht, taper);
%
%End
```

```
function mmat_bo = boplate_mass(bopl_thic, bradi, dens_pl, bopl_num, nwav)
%
%To assemble the mass matrix for bottom plate
%
%*Define variables*
%
%*Input arguments*
%bopl_thic -- Thickness of bottom plate (assume uniform) (m)
%bradi -- Bottom radius of cone (m)
%dens_pl -- Density of top and bottom plate material (kg/m3)
%bopl_num -- No. of elements at bottom end circular plate
%
%*Other arguments*
%total_node -- Total number of nodes
%tdof -- Total number of DOFs
%jjs -- Number of iteration of "jj"th element
%a_val -- outer radius of annulur element (m)
%b_val -- inner radius of annulur element (m)
%
%*Matrices*
%m_plele4 -- Elemental mass matrix (4 x 4)
%m_pla -- Elemental mass matrix (tdof by tdof)
%mmat_bo -- Mass Matrix (bottom plate)
%mass_cent -- Mass matrix of centre plate (2 x 2)
%
%Prepare variables to input to elemental mass matrix computation
%
total_node = bopl_num;
tdof = 2*total_node;
mmat_bo = zeros(tdof, tdof);
%
%Add mass contribution by centre plate element
mass_cent = m_centplate(bopl_thic, dens_pl, bradi, bopl_num);
mmat_bo(tdof-1:tdof, tdof-1:tdof) = mass_cent;
%
%Assemble global mass matrix for bottom plate
for jjs = 1:bopl_num-1
   b_val = jjs*bradi/bopl_num;
   a_val = (jjs+1)*bradi/bopl_num;
   m_pla = zeros(tdof, tdof);
   %Formation of 4x4 elemental mass matrix
   m_plele4 = m_boplate (a_val, b_val, bopl_thic, dens_pl);
   %Formation of elemental mass matrix in entire plate matrix dimensions
   m_pla = mplglo_bo (m_plele4, jjs, tdof);
   %Increment to entire plate mass matrix
   mmat_bo = mmat_bo + m_pla;
```

```
end
%
return
```

```
function kmat_bo = boplate_stiff(bopl_thic, bradi, youn_pl, pois, bopl_num, nwav)
%
%To assemble the stiffness matrix for bottom plate
%
%*Define variables*
%
%*Input arguments*
%bopl_thic -- Thickness of bottom plate (assume uniform) (m)
%bradi -- Bottom radius of cone (m)
%youn_pl -- Young's modulus of top and bottom plate material (x10e9 N/m2)
%pois -- Poisson's ratio of material
%bopl_num -- No. of elements at bottom end circular plate
%nwav -- Wave number in consideration
%
%*Other arguments*
%total_node -- Total number of nodes
%tdof -- Total number of DOFs
%jjs -- Number of iteration of "jj"th element
%a_val -- Outer radius of annular (m)
%b_val -- Inner radius of annular (m)
%d_val -- Plate rigidity (Nm)
%
%*Matrices*
%k_plele4 -- Elemental stiffness matrix (4 x 4)
%k_pla -- Elemental stiffness matrix (tdof by tdof)
%kmat_bo -- Stiffness Matrix (bottom plate)
%
%Prepare variables to input to elemental stiffness matrix computation
total_node = bopl_num;
tdof = 2*total_node;
kmat_bo = zeros(tdof, tdof);
%
%Add stiffness of centre plate (only K22 is affected by the centre plate)
d_val = (youn_pl*1000000000*bopl_thic^3)/(12*(1-pois^2));
kmat_bo(tdof, tdof) = 2*d_val*pi*(1+pois);
%
%Assemble global stiffness matrix for plate only
for jjs = 1:bopl_num-1
   k_pla = zeros(tdof, tdof);
   b_val = jjs*bradi/(bopl_num);
   a_val = (jjs+1)*bradi/(bopl_num);
   %Formation of 4x4 elemental stiffness matrix
   k_plele4 = k_boplate(a_val, b_val, pois, d_val, nwav);
   %Formation of tdof x tdof elemental stiffness matrix
   k_pla = kplglo_bo(k_plele4, jjs, tdof);
   %Increment to entire plate stiffness matrix
```

```
    kmat_bo = kmat_bo + k_pla;
end
%
return
```

```
function [tube_thic, topl_thic, bopl_thic, top_diam, bot_diam, cone_ht, youn_cy,
dens_cy, youn_pl, dens_pl, pois, ka] = cy_parameters_define
%
%To input parameters for computation and define variables
%
%Define variables
%tube_thic -- Thickness of tapered tube (assume uniform) (m)
%topl_thic -- Thickness of top plate (assume uniform) (m)
%bopl_thic -- Thickness of bottom plate (assume uniform) (m)
%top_diam -- Top diameter of cone (m)
%bot_diam -- Bottom diameter of cone (m)
%cone_ht -- Height of cone (m)
%
%*Material properties*
%youn_cy -- Young's modulus of tapered cylinder material (x10e9 N/m2)
%dens_cy -- Density of tapered cylinder material (kg/m3)
%youn_pl -- Young's modulus of top and bottom plate material (x10e9 N/m2)
%dens_pl -- Density of top and bottom plate material (kg/m3)
%pois -- Poisson's ratio of material
%ka -- ka of the excitation frequency
%
%Getting information
tube_thic = input('Enter the thickness of tapered tube in m:');
topl_thic = input('Enter the thickness of top plate in m:');
bopl_thic = input('Enter the thickness of bottom plate in m:');
top_diam = input('Enter the diameter of top plate in m:');
bot_diam = input('Enter the diameter of bottom plate in m:');
cone_ht = input('Enter the height of cone in m:');
youn_cy = input('Enter the Young''s Modulus of tapered cylinder material in GPa:');
dens_cy = input('Enter the density of tapered cylinder material in kg/m3:');
youn_pl = input('Enter the Young''s Modulus of endcaps material in GPa:');
dens_pl = input('Enter the density of endcaps material in kg/m3:');
pois = input('Enter the Poisson ratio of material:');
ka = input('Enter the ka of the excitation frequency:');
return
```

```
function umat = defo_bopl2(kk, bopl_leng, bopl_num)
%
%This computes the conversion matrix from nodal force to elemental pressure
%for bottom annulur plate
%
a = (bopl_num - kk + 1)*bopl_leng;
b = (bopl_num - kk)*bopl_leng;
r = (bopl_num - kk + 0.5)*bopl_leng;
al = a/b;
%
%Definition of nu terms for deriving the displacement shape function
%
delta = (al^2 - 1)^2 - 4*al^2*(log(al))^2;
nu11 = (al^2) * (al^2 - 1 + 2*log(al) - 4*(log(al))^2);
nu12 = (1 - 2*al^2*log(al) - al^2);
nu13 = 4*al^2*log(al);
nu14 = 2*(al^2 - 1);
nu21 = 2*b*al^2*(log(al))^2;
nu22 = -nu21;
nu23 = b*al^2*(al^2 - 1 - 2*log(al));
nu24 = b*(2*al^2*log(al) - al^2 +1);
nu31 = nu12;
nu32 = -nu12;
nu33 = -nu13;
nu34 = -nu14;
nu41 = b*al*log(al)*(al^2 - 1);
nu42 = -nu41;
nu43 = b*al*(2*al^2*log(al) - al^2 + 1);
nu44 = b*al*(al^2 - 1 - 2*log(al));
%
%Definition of rho terms
%
ro1 = 1;
ro2 = (r/b)^2;
ro3 = log(r/b);
ro4 = (r/b)^2*log(r/b);
%
%ci1 is the shape function of w1 displacement
%ci2 is the shape function of w2 displacement
%ci3 is the shape function of w3 displacement
%ci4 is the shape function of w4 displacement
%
ci1 = (nu11*ro1 + nu12*ro2 + nu13*ro3 + nu14*ro4)/delta;
ci2 = (nu21*ro1 + nu22*ro2 + nu23*ro3 + nu24*ro4)/delta;
ci3 = (nu31*ro1 + nu32*ro2 + nu33*ro3 + nu34*ro4)/delta;
ci4 = (nu41*ro1 + nu42*ro2 + nu43*ro3 + nu44*ro4)/delta;
```

```
ci = [ci3 ci4 ci1 ci2];
umat = ci;
return
```

```
function umat = defo_pl(pl_leng)
%
%This computes the conversion matrix from nodal displacement to elemental
displacement
%for centre circular plate
%
r = pl_leng/2;
ph1 = 1;
ph2 = pl_leng/2*((r/pl_leng)^2 - 1);
umat = [ph1 ph2];
return
```

```
function umat = defo_topl2(ii, topl_leng)
%
%This computes the conversion matrix from nodal force to elemental pressure
%for top annulur plate
%
a = ii*topl_leng;
b = (ii-1)*topl_leng;
r = (ii-0.5)*topl_leng;
al = a/b;
%
%Definition of nu terms for deriving the displacement shape function
%
delta = (al^2 - 1)^2 - 4*al^2*(log(al))^2;
nu11 = (al^2) * (al^2 - 1 + 2*log(al) - 4*(log(al))^2);
nu12 = (1 - 2*al^2*log(al) - al^2);
nu13 = 4*al^2*log(al);
nu14 = 2*(al^2 - 1);
nu21 = 2*b*al^2*(log(al))^2;
nu22 = -nu21;
nu23 = b*al^2*(al^2 - 1 - 2*log(al));
nu24 = b*(2*al^2*log(al) - al^2 +1);
nu31 = nu12;
nu32 = -nu12;
nu33 = -nu13;
nu34 = -nu14;
nu41 = b*al*log(al)*(al^2 - 1);
nu42 = -nu41;
nu43 = b*al*(2*al^2*log(al) - al^2 + 1);
nu44 = b*al*(al^2 - 1 - 2*log(al));
%
%Definition of rho terms
%
ro1 = 1;
ro2 = (r/b)^2;
ro3 = log(r/b);
ro4 = (r/b)^2*log(r/b);
%
%ci1 is the shape function of w1 displacement
%ci2 is the shape function of w2 displacement
%ci3 is the shape function of w3 displacement
%ci4 is the shape function of w4 displacement
%
ci1 = (nu11*ro1 + nu12*ro2 + nu13*ro3 + nu14*ro4)/delta;
ci2 = (nu21*ro1 + nu22*ro2 + nu23*ro3 + nu24*ro4)/delta;
ci3 = (nu31*ro1 + nu32*ro2 + nu33*ro3 + nu34*ro4)/delta;
ci4 = (nu41*ro1 + nu42*ro2 + nu43*ro3 + nu44*ro4)/delta;
```

```
ci = [ci1 ci2 ci3 ci4];
umat = ci;
return
```

```
function umat = defo_tu(jj, tube_leng)
%
%This computes the conversion matrix from nodal force to elemental pressure
%for conical shell
%
zi = 0.5;
%
ph1 = 1-3*zi^2+2*zi^3;
ph2 = tube_leng*(zi-2*zi^2+zi^3);
ph3 = 3*zi^2-2*zi^3;
ph4 = tube_leng*(-zi^2+zi^3);
%
umat = [ph1 ph2 ph3 ph4];
return
```

```
function umat = dipmat(topl_num, bopl_num, tube_num, topl_leng, tube_leng,
bopl_leng)
%
%This computes the conversion matrix from nodal displacement to elemental
displacement
%
dd1 = topl_num + bopl_num + tube_num;
dd2 = 2*(topl_num + bopl_num) + 2*(tube_num - 1);
umat = zeros(dd1, dd2);
%
for ii = 1 : topl_num
   if ii == 1
      umat(ii, ii : ii+1) = defo_pl(topl_leng);
   else
      umat(ii, 2*ii-3 : 2*ii) = defo_topl2(ii, topl_leng);
   end
end
%
for jj = topl_num + 1 : topl_num + tube_num
   umat(jj, 2*jj-3 : 2*jj) = defo_tu(jj, tube_leng);
end
%
for kk = topl_num + tube_num + 1 : dd1
   if kk == dd1
      umat(kk, 2*kk-3 : 2*kk-2) = defo_pl(bopl_leng);
   else
      umat(kk, 2*kk-3 : 2*kk) = defo_bopl2(kk - topl_num - tube_num, bopl_leng,
bopl_num);
   end
end
return
```

```
function filegen(immata, fluidense, topl_leng, bopl_leng, tube_leng, tradi, bradi,
topl_num, bopl_num, tube_num, cone_ht, taper)
%
%To generate the file with printed admittance matrix
%
%*Inputs*
%imped -- Admittance matrix for cone with endcaps (3D).
%tube_leng -- Length of each element on the cylindrical tube (m)
%topl_leng -- Length of each element on top end plate (m)
%bopl_leng -- Length of each element on bottom end plate (m)
%tradi -- Top radius of cone (m)
%bradi -- Bottom radius of cone (m)
%taper - Taper angle of cone
%topl_num -- No. of elements at top end circular plate
%bopl_num -- No. of elements at bottom end circular plate
%tube_num -- No. of taper cylindrical elements
%cone_ht -- Height of cone (m)
%
%*Matrices*
%stiff -- 3-D matrix with layer by layer inverse of 3-D admittance matrix
%glstiff -- Global 2-D dynamic flexibility matrix
%
%*Variables*
%xx -- x-cordinate of node
%yy -- y-cordinate of node
%zz -- z-cordinate of node
%xxnorm -- vector component in x-direction of surface normal vector of node
%yynorm -- vector component in x-direction of surface normal vector of node
%zznorm -- vector component in x-direction of surface normal vector of node
%
%Start
%
%Definitions
d=size(immata);
row = d(2);
column = d(2);
angle_step = d(1)/d(2);
glim = zeros(row*angle_step, column*angle_step);
%
%Start computing stiffness matrix from admittance matrix
%
%Unravel the 3-D admittance matrix into equivalent 2-D global
%admittance matrix
%
for jj = 1 : angle_step
   for kk = jj : angle_step
```

```
      glim((kk-1)*row + 1 : kk*row, (jj-1)*column + 1 : jj*column) = immata((kk-
jj)*d(2)+1 : (kk-jj+1)*d(2), :);
   end
   %
   for mm = 1 : jj - 1
      glim((mm - 1)*row + 1 : mm*row, (jj-1)*column + 1 : jj*column) =
immata((angle_step-jj+mm)*d(2)+1 : (angle_step-jj+mm+1)*d(2), :);
   end
end
end
%
%Below steps to start writing the glstiff matrix entries into
%.imp file base on format by SCATT
%
%Open file
%
fid = fopen('impet.adm', 'w');
%
%Start writing
%
for aa = 1 : angle_step*row
   for bb = 1: angle_step*row
      fprintf(fid, '%d %d\n', real(glim(bb, aa)), imag(glim(bb, aa)));
   end
end
status = fclose(fid);
%
%Start computing and writing nodal positions and surface normal vector
%into .srf file base on format by SCATT
%
%Open file
%
fid = fopen('impet.srf', 'w');
%
%Computation of parameters to generate the nodal distance
%
arc_to = topl_leng*2*pi/angle_step;
arc_bo = bopl_leng*2*pi/angle_step;
%
%Start computing and writing
%
fprintf(fid, '%d %d\n', row*angle_step, fluidense);
%
%Start with angular position 0 degree and progress in positive angular
%direction
%
for dd = 0:1:angle_step-1
```

```
   %
   %Start with top plate nodes
   %
   for ee = 1 : topl_num
      xx = ((ee-0.5)*topl_leng)*cos((dd+0.5)*2*pi/angle_step);
      yy = -((ee-0.5)*topl_leng)*sin((dd+0.5)*2*pi/angle_step);
      zz = -cone_ht/2;
      xxnorm = 0;
      yynorm = 0;
      zznorm = -1;
      dn = ((ee-0.5)*arc_to + topl_leng)/2;
      fprintf(fid, '%11.8e %11.8e %11.8e %11.8e %11.8e %11.8e %11.8e\n', xx, yy, zz,
xxnorm, yynorm, zznorm, dn);
   end
   %
   %Then with tube nodes
   %
   for ff = 1 : tube_num
      xx = (tradi + (ff-0.5)*tube_leng*sin(taper))*cos((dd+0.5)*2*pi/angle_step);
      yy = -(tradi + (ff-0.5)*tube_leng*sin(taper))*sin((dd+0.5)*2*pi/angle_step);
      zz = -0.5*cone_ht + (ff-0.5)*tube_leng*cos(taper);
      xxnorm = cos((dd+0.5)*2*pi/angle_step)*cos(taper);
      yynorm = -sin((dd+0.5)*2*pi/angle_step)*cos(taper);
      zznorm = -sin(taper);
      dn = (2*pi*(tradi + (ff-0.5)*tube_leng*sin(taper))/angle_step + tube_leng)/2;
      fprintf(fid, '%11.8e %11.8e %11.8e %11.8e %11.8e %11.8e %11.8e\n', xx, yy, zz,
xxnorm, yynorm, zznorm, dn);
   end
   %
   %Finally with with bottom plate nodes
   %
   for gg = 1 : bopl_num
      xx = ((bopl_num - (gg-0.5))*bopl_leng)*cos((dd+0.5)*2*pi/angle_step);
      yy = -((bopl_num - (gg-0.5))*bopl_leng)*sin((dd+0.5)*2*pi/angle_step);
      zz = cone_ht/2;
      xxnorm = 0;
      yynorm = 0;
      zznorm = 1;
      dn = ((bopl_num - (gg-0.5))*arc_bo + bopl_leng)/2;
      fprintf(fid, '%11.8e %11.8e %11.8e %11.8e %11.8e %11.8e %11.8e\n', xx, yy, zz,
xxnorm, yynorm, zznorm, dn);
   end
end
%
%Done!
%
```

```
status = fclose(fid);
return
```

```
function flex = flexglobe (k_globe, m_globe, frad, to, tu, bo, total_dof, taper)
%
%To compute the flexibility matrix for conical shell
%lateral displacement with no v, w vibration at corners.
%
%*Inputs*
%k_globe -- Stiffness Matrix (global)
%m_globe -- Mass Matrix (global)
%frad -- Excitation frequency (rad/s)
%total_dof -- Total no. of dofs
%to -- Number of dof for top plate
%tu -- Number of dof for tube
%bo -- Number of dof for bottom plate
%taper - Taper angle of cone
%
%Output*
%stiff -- Dynamic stiffness matrix for normal shell
%displacement of assembled plate/ tapered tube model.
%
%*Others*
%tmat -- Transformation matrix to resolve the longitudinal and lateral
%coordinates of tapered tube to align coordinates with normal displacement
%of tapered tube
%dystiff_mat -- Dynamic stiffness matrix
%dystiff -- Dynamic stiffness matrix with resolution of longitudinal
%and lateral coordinates of tapered tube to align coordinates
%with normal displacement
%
%Dynamic stiffness matrix is equal K-(omega^2).*M
%
dystiff_mat = k_globe-(frad^2).*m_globe;
%
%Resolve the longitudinal and lateral coordinates of tapered tube to align
%coordinates with normal displacement of tapered tube
%
tmat = zeros(total_dof , total_dof);
%
for ii = 1 : to+2
   tmat(ii, ii) = 1;
end
%
for jj = 1 : tu/4 - 2
   tmat(to+2 + 4*(jj-1) + 1 : to+2 + 4*jj, to+2 + 4*(jj-1) + 1 : to+2 + 4*jj) = [cos(taper) 0
-sin(taper) 0; 0 1 0 0; sin(taper) 0 cos(taper) 0; 0 0 0 1];
end
%
```

```
for kk = 1 : bo+2
   tmat(to + tu - 6 + kk, to + tu - 6 + kk) = 1;
end
%
dystiff = tmat'*dystiff_mat*tmat;
dystcomp = dystiff([1:(to-1) (to+2):(to+tu-5) (to+tu-2):(to+tu+bo-4)], [1:(to-1)
(to+2):(to+tu-5) (to+tu-2):(to+tu+bo-4)]);
%
dyinv = inv(dystcomp);
%
aa = [1:to];
for pp = 1:tu/4-2
   aa(1,to+2*(pp-1)+1 : to+2*pp) = [to+(pp-1)*4 + 3 to + 4*pp];
end
aa(1, to + 2*(tu/4 -2) + 1 : to + 2*(tu/4 -2) + bo) = to+tu-8 +1 : to+tu+bo -8;
%
invmat = dyinv(aa, aa);
flex = invmat;
%
return
```

```
function qmat = for_mat(topl_num, bopl_num, tube_num, topl_leng, tube_leng,
bopl_leng, taper, tradi, bradi)
%
%This computes the conversion matrix from nodal force to elemental pressure
%
dd1 = 2*(topl_num + bopl_num) + 2*(tube_num - 1);
dd2 = topl_num + bopl_num + tube_num;
qmat = zeros(dd1, dd2);
%
for ii = 1 : topl_num
   if ii == 1
      qmat(ii : ii+1, ii) = shape_pl(topl_leng);
   else
      qmat(2*ii-3 : 2*ii, ii) = shape_topl2(ii, topl_leng);
   end
end
%
for jj = topl_num + 1 : topl_num + tube_num
   temp = 0;
   temp = shape_tu(jj, tube_leng, tradi, bradi, tube_num, taper);
   qmat(2*jj-3 : 2*jj, jj) = tube_leng.*temp;
end
%
for kk = topl_num + tube_num + 1 : dd2
   if kk == dd2
      qmat(2*kk-3 : 2*kk-2, kk) = shape_pl(bopl_leng);
   else
      qmat(2*kk-3 : 2*kk, kk) = shape_bopl2(kk - topl_num - tube_num, bopl_leng,
bopl_num);
   end
end
return
```

```
function fourier_term = fourier_angle(angle_step)
%
%*Input*
%angle_step -- angle step as selected by user to analyse response
%
%*Output*
%fourier_term -- Vector containing all the 11 fourier terms
%from n =0 to 10.
%
%*Symbols*
%t -- angle
%
%Computation
angle = 2*pi/angle_step;
fourier_term = zeros(51, 1);
%First fourier term for n=0
fourier_term(1,1) = 1/angle_step;
%
%The rest of the fourier terms
for ii = 1:50
   fourier_term(ii+1,1) = 2/ii/pi*sin(ii*pi/angle_step);
end
return
```

```
function [k_globe, m_globe, total_dof, top_n, tub_n, bot_n] = globecomp(kmat_tu,
mmat_tu, kmat_to, mmat_to, kmat_bo, mmat_bo)
%
%To compute global stiffness and mass matrix for a conical shell
%with flat end caps
%
%*Input matrices*
%kmat_tu -- Stiffness Matrix (tube)
%mmat_tu -- Mass Matrix (tube)
%kmat_to -- Stiffness Matrix (top plate)
%mmat_to -- Mass Matrix (top plate)
%kmat_bo -- Stiffness Matrix (bottom plate)
%mmat_bo -- Mass Matrix (bottom plate)
%
%*Length of matrices*
%total_dof -- Total no. of dofs
%top_n -- Total no. of dofs on top plate
%tub_n -- Total no. of dofs on tapered tube
%bot_n -- Total no. of dofs on bottom plate
%
%Computing dofs in individual matrices
top_n = length(kmat_to);
tub_n = length(kmat_tu);
bot_n = length(kmat_bo);
total_dof = top_n-2 + tub_n + bot_n-2;
%
k_globe = zeros(total_dof, total_dof);
m_globe = zeros(total_dof, total_dof);
k_globe_top = zeros(total_dof, total_dof);
m_globe_top = zeros(total_dof, total_dof);
k_globe_tub = zeros(total_dof, total_dof);
m_globe_tub = zeros(total_dof, total_dof);
k_globe_bot = zeros(total_dof, total_dof);
m_globe_bot = zeros(total_dof, total_dof);
%
%Formulating k_globe
%
%Position top plate elements in global matrices
k_globe_top(1:top_n-1, 1:top_n-1) = kmat_to(1:top_n-1, 1:top_n-1);
k_globe_top(top_n+2, 1:top_n-1) = kmat_to(top_n, 1:top_n-1);
k_globe_top(1:top_n-1, top_n+2) = kmat_to(1:top_n-1, top_n);
k_globe_top(top_n+2, top_n+2) = kmat_to(top_n, top_n);
%
m_globe_top(1:top_n-1, 1:top_n-1) = mmat_to(1:top_n-1, 1:top_n-1);
m_globe_top(top_n+2, 1:top_n-1) = mmat_to(top_n, 1:top_n-1);
m_globe_top(1:top_n-1, top_n+2) = mmat_to(1:top_n-1, top_n);
m_globe_top(top_n+2, top_n+2) = mmat_to(top_n, top_n);
%
%Position tube elements in global matrices
k_globe_tub(top_n-1:(top_n-1)+(tub_n-1), top_n-1:(top_n-1)+(tub_n-1)) = kmat_tu;
%
m_globe_tub(top_n-1:(top_n-1)+(tub_n-1), top_n-1:(top_n-1)+(tub_n-1)) = mmat_tu;
%
%Position bottom plate elements in global matrices
k_globe_bot((top_n-2+tub_n)-3, (top_n-2+tub_n)-3) = kmat_bo(1,1);
k_globe_bot((top_n-2+tub_n)-3, (top_n-2+tub_n):(top_n-2+tub_n)+(bot_n-2)) =
kmat_bo(1,2:bot_n);
k_globe_bot((top_n-2+tub_n):(top_n-2+tub_n)+(bot_n-2), (top_n-2+tub_n)-3) =
kmat_bo(2:bot_n,1);
k_globe_bot((top_n-2+tub_n):(top_n-2+tub_n)+(bot_n-2), (top_n-2+tub_n):(top_n-
2+tub_n)+(bot_n-2)) = kmat_bo(2:bot_n,2:bot_n);
%
m_globe_bot((top_n-2+tub_n)-3, (top_n-2+tub_n)-3) = mmat_bo(1,1);
m_globe_bot((top_n-2+tub_n)-3, (top_n-2+tub_n):(top_n-2+tub_n)+(bot_n-2)) =
mmat_bo(1,2:bot_n);
m_globe_bot((top_n-2+tub_n):(top_n-2+tub_n)+(bot_n-2), (top_n-2+tub_n)-3) =
mmat_bo(2:bot_n,1);
m_globe_bot((top_n-2+tub_n):(top_n-2+tub_n)+(bot_n-2), (top_n-2+tub_n):(top_n-
2+tub_n)+(bot_n-2)) = mmat_bo(2:bot_n,2:bot_n);
%
k_globe = k_globe_top + k_globe_tub + k_globe_bot;
m_globe = m_globe_top + m_globe_tub + m_globe_bot;
%
return
```

```
function k_plele4 = k_boplate(a_val, b_val, pois, d_val, nwav)
%
%To compute the stiffness matrix for an
%axisymmetric annular element subjected to vibration
%
%Refer to paper "Static, vibration and buckling analysis of axisymmetric
%circular plates using finite elements", Gerard C. Pardoen, Computers and
%Structures, Vol.3, pp.355-375.
%
%Define variables
%
%*Inputs*
%a_val -- outer radius of annulur element (m)
%b_val -- inner radius of annulur element (m)
%pois -- Poisson's ratio of material
%d_val -- Plate rigidity (Nm)
%nwav -- Wave number in consideration
%
%*Output*
%k_plele4 -- Elemental stiffness matrix (4 x 4)
%
%*Symbols*
%al -- symbol of al_val
%r -- radius
%b -- symbol of b_val
%thet -- angle in circumferential direction
%n -- symbol of nwav
%
%*Others*
%al_val -- Ratio of a_val/b_val
%
k_plele4 = zeros(4,4);
syms al r b thet n;
al_val = a_val/b_val;
%
%Definition of nu terms for deriving the displacement shape function
%
nu11 = (al^2) * (al^2 - 1 + 2*log(al) - 4*(log(al))^2);
nu12 = (1 - 2*al^2*log(al) - al^2);
nu13 = 4*al^2*log(al);
nu14 = 2*(al^2 - 1);
nu21 = 2*b*al^2*(log(al))^2;
nu22 = -nu21;
nu23 = b*al^2*(al^2 - 1 - 2*log(al));
nu24 = b*(2*al^2*log(al) - al^2 +1);
nu31 = nu12;
```

```
nu32 = -nu12;
nu33 = -nu13;
nu34 = -nu14;
nu41 = b*al*log(al)*(al^2 - 1);
nu42 = -nu41;
nu43 = b*al*(2*al^2*log(al) - al^2 + 1);
nu44 = b*al*(al^2 - 1 - 2*log(al));
%
%Definition of rho terms
%
ro1 = 1;
ro2 = (r/b)^2;
ro3 = log(r/b);
ro4 = (r/b)^2*log(r/b);
%
%ci1 is the shape function of w1 displacement
%ci2 is the shape function of w2 displacement
%ci3 is the shape function of w3 displacement
%ci4 is the shape function of w4 displacement
%
ci1 = (nu11*ro1 + nu12*ro2 + nu13*ro3 + nu14*ro4)*cos(n*thet);
ci2 = (nu21*ro1 + nu22*ro2 + nu23*ro3 + nu24*ro4)*cos(n*thet);
ci3 = (nu31*ro1 + nu32*ro2 + nu33*ro3 + nu34*ro4)*cos(n*thet);
ci4 = (nu41*ro1 + nu42*ro2 + nu43*ro3 + nu44*ro4)*cos(n*thet);
%
%Define the shear forces and moments for each individual respective
%displacement shape function
%
q1 = 2*r*pi*d_val*(diff(r^-1*diff(r*diff(ci1, r), r), r) + diff(r^-2*diff(ci1, thet, 2), r));
m1 = 2*r*pi*d_val*((diff(ci1, r, 2) + pois*r^-1*diff(ci1, r)) + (pois*r^-2*diff(ci1, thet,
2)));
q2 = 2*r*pi*d_val*(diff(r^-1*diff(r*diff(ci2, r), r), r) + diff(r^-2*diff(ci2, thet, 2), r));
m2 = 2*r*pi*d_val*((diff(ci2, r, 2) + pois*r^-1*diff(ci2, r)) + (pois*r^-2*diff(ci2, thet,
2)));
q3 = 2*r*pi*d_val*(diff(r^-1*diff(r*diff(ci3, r), r), r) + diff(r^-2*diff(ci3, thet, 2), r));
m3 = 2*r*pi*d_val*((diff(ci3, r, 2) + pois*r^-1*diff(ci3, r)) + (pois*r^-2*diff(ci3, thet,
2)));
q4 = 2*r*pi*d_val*(diff(r^-1*diff(r*diff(ci4, r), r), r) + diff(r^-2*diff(ci4, thet, 2), r));
m4 = 2*r*pi*d_val*((diff(ci4, r, 2) + pois*r^-1*diff(ci4, r)) + (pois*r^-2*diff(ci4, thet,
2)));
%
%Get rid of cos(n*theta) term and add delta term
%
delta = (al_val^2 - 1)^2 - 4*al_val^2*(log(al_val))^2;
qq1 = q1/cos(n*thet)/delta;
mm1 = m1/cos(n*thet)/delta;
```

```
qq2 = q2/cos(n*thet)/delta;
mm2 = m2/cos(n*thet)/delta;
qq3 = q3/cos(n*thet)/delta;
mm3 = m3/cos(n*thet)/delta;
qq4 = q4/cos(n*thet)/delta;
mm4 = m4/cos(n*thet)/delta;
%
%Compute kplate terms
k_plele4(3,3) = double(subs(qq1, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
tempkplate(4,3) = -double(subs(mm1, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
k_plele4(1,3) = -double(subs(qq1, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
k_plele4(2,3) = double(subs(mm1, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
%
tempkplate(3,4) = double(subs(qq2, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
k_plele4(4,4) = -double(subs(mm2, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
k_plele4(1,4) = -double(subs(qq2, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
k_plele4(2,4) = double(subs(mm2, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
%
k_plele4(3,1) = double(subs(qq3, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
k_plele4(4,1) = -double(subs(mm3, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
k_plele4(1,1) = -double(subs(qq3, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
tempkplate(2,1) = double(subs(mm3, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
%
k_plele4(3,2) = double(subs(qq4, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
k_plele4(4,2) = -double(subs(mm4, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
tempkplate(1,2) = -double(subs(qq4, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
k_plele4(2,2) = double(subs(mm4, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
%
%Averaging for (1,2) and (2, 1) elements
%Averaging for (3,4) and (4, 3) elements
%
k_plele4(2,1) = (tempkplate(2,1)+tempkplate(1,2))/2;
k_plele4(1,2) = (tempkplate(2,1)+tempkplate(1,2))/2;
k_plele4(4,3) = (tempkplate(4,3)+tempkplate(3,4))/2;
k_plele4(3,4) = (tempkplate(4,3)+tempkplate(3,4))/2;
%
return
```

```
function k_plele4 = k_toplate(a_val, b_val, pois, d_val, nwav)
%
%To compute the stiffness matrix for an
%axisymmetric annular element subjected to vibration
%
%Refer to paper "Static, vibration and buckling analysis of axisymmetric
%circular plates using finite elements", Gerard C. Pardoen, Computers and
%Structures, Vol.3, pp.355-375.
%
%Define variables
%
%*Inputs*
%a_val -- outer radius of annulur element (m)
%b_val -- inner radius of annulur element (m)
%pois -- Poisson's ratio of material
%d_val -- Plate rigidity (Nm)
%nwav -- Wave number in consideration
%
%*Output*
%k_plele4 -- Elemental stiffness matrix (4 x 4)
%
%*Symbols*
%al -- symbol of al_val
%r -- radius
%b -- symbol of b_val
%thet -- angle in circumferential direction
%n -- symbol of nwav
%
%*Others*
%al_val -- Ratio of a_val/b_val
%
k_plele4 = zeros(4,4);
syms al r b thet n;
al_val = a_val/b_val;
%
%Definition of nu terms for deriving the displacement shape function
%
nu11 = (al^2) * (al^2 - 1 + 2*log(al) - 4*(log(al))^2);
nu12 = (1 - 2*al^2*log(al) - al^2);
nu13 = 4*al^2*log(al);
nu14 = 2*(al^2 - 1);
nu21 = 2*b*al^2*(log(al))^2;
nu22 = -nu21;
nu23 = b*al^2*(al^2 - 1 - 2*log(al));
nu24 = b*(2*al^2*log(al) - al^2 +1);
nu31 = nu12;
```

```
nu32 = -nu12;
nu33 = -nu13;
nu34 = -nu14;
nu41 = b*al*log(al)*(al^2 - 1);
nu42 = -nu41;
nu43 = b*al*(2*al^2*log(al) - al^2 + 1);
nu44 = b*al*(al^2 - 1 - 2*log(al));
%
%Definition of rho terms
%
ro1 = 1;
ro2 = (r/b)^2;
ro3 = log(r/b);
ro4 = (r/b)^2*log(r/b);
%
%ci1 is the shape function of w1 displacement
%ci2 is the shape function of w2 displacement
%ci3 is the shape function of w3 displacement
%ci4 is the shape function of w4 displacement
%
ci1 = (nu11*ro1 + nu12*ro2 + nu13*ro3 + nu14*ro4)*cos(n*thet);
ci2 = (nu21*ro1 + nu22*ro2 + nu23*ro3 + nu24*ro4)*cos(n*thet);
ci3 = (nu31*ro1 + nu32*ro2 + nu33*ro3 + nu34*ro4)*cos(n*thet);
ci4 = (nu41*ro1 + nu42*ro2 + nu43*ro3 + nu44*ro4)*cos(n*thet);
%
%Define the shear forces and moments for each individual respective
%displacement shape function
%
q1 = 2*r*pi*d_val*(diff(r^-1*diff(r*diff(ci1, r), r), r) + diff(r^-2*diff(ci1, thet, 2), r));
m1 = 2*r*pi*d_val*((diff(ci1, r, 2) + pois*r^-1*diff(ci1, r)) + (pois*r^-2*diff(ci1, thet,
2)));
q2 = 2*r*pi*d_val*(diff(r^-1*diff(r*diff(ci2, r), r), r) + diff(r^-2*diff(ci2, thet, 2), r));
m2 = 2*r*pi*d_val*((diff(ci2, r, 2) + pois*r^-1*diff(ci2, r)) + (pois*r^-2*diff(ci2, thet,
2)));
q3 = 2*r*pi*d_val*(diff(r^-1*diff(r*diff(ci3, r), r), r) + diff(r^-2*diff(ci3, thet, 2), r));
m3 = 2*r*pi*d_val*((diff(ci3, r, 2) + pois*r^-1*diff(ci3, r)) + (pois*r^-2*diff(ci3, thet,
2)));
q4 = 2*r*pi*d_val*(diff(r^-1*diff(r*diff(ci4, r), r), r) + diff(r^-2*diff(ci4, thet, 2), r));
m4 = 2*r*pi*d_val*((diff(ci4, r, 2) + pois*r^-1*diff(ci4, r)) + (pois*r^-2*diff(ci4, thet,
2)));
%
%Get rid of cos(n*theta) term and add delta term
%
delta = (al_val^2 - 1)^2 - 4*al_val^2*(log(al_val))^2;
qq1 = q1/cos(n*thet)/delta;
mm1 = m1/cos(n*thet)/delta;
```

```
qq2 = q2/cos(n*thet)/delta;
mm2 = m2/cos(n*thet)/delta;
qq3 = q3/cos(n*thet)/delta;
mm3 = m3/cos(n*thet)/delta;
qq4 = q4/cos(n*thet)/delta;
mm4 = m4/cos(n*thet)/delta;
%
%Compute kplate terms
k_plele4(1,1) = double(subs(qq1, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
tempkplate(2,1) = -double(subs(mm1, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
k_plele4(3,1) = -double(subs(qq1, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
k_plele4(4,1) = double(subs(mm1, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
%
tempkplate(1,2) = double(subs(qq2, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
k_plele4(2,2) = -double(subs(mm2, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
k_plele4(3,2) = -double(subs(qq2, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
k_plele4(4,2) = double(subs(mm2, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
%
k_plele4(1,3) = double(subs(qq3, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
k_plele4(2,3) = -double(subs(mm3, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
k_plele4(3,3) = -double(subs(qq3, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
tempkplate(4,3) = double(subs(mm3, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
%
k_plele4(1,4) = double(subs(qq4, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
k_plele4(2,4) = -double(subs(mm4, {r, b, al, n}, {b_val, b_val, al_val, nwav}));
tempkplate(3,4) = -double(subs(qq4, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
k_plele4(4,4) = double(subs(mm4, {r, b, al, n}, {a_val, b_val, al_val, nwav}));
%
%Averaging for (1,2) and (2, 1) elements
%Averaging for (3,4) and (4, 3) elements
%
k_plele4(2,1) = (tempkplate(2,1)+tempkplate(1,2))/2;
k_plele4(1,2) = (tempkplate(2,1)+tempkplate(1,2))/2;
k_plele4(4,3) = (tempkplate(4,3)+tempkplate(3,4))/2;
k_plele4(3,4) = (tempkplate(4,3)+tempkplate(3,4))/2;
%
return
```

```
function k_cydim = k_tube (tube_thic, tradi, bradi, youn_cy, pois, tube_leng, tube_num,
taper, jjs, nwav)
%
%To compute the elemental stiffness matrix for tapered cylindrical elements.
%
%Define variables
%
%*Input arguments*
%tube_thic -- Thickness of tapered tube (assume uniform) (m)
%youn_cy -- Young's modulus of tapered cylinder material (x10e9 N/m2)
%pois -- Poisson's ratio of material
%tradi -- Top radius of cone (m)
%bradi -- Bottom radius of cone (m)
%tube_num -- No. of taper cylindrical elements
%tube_leng -- Length of each element on the cylindrical tube (m)
%taper - Taper angle of cone
%nwav -- Wave number in consideration
%jjs -- Number of iteration of "jj"th element
%
%*Output argument*
%k_cydim -- Double of k_stf
%
%*Others*
%r1 -- Radius at node 1
%r2 -- Radius at node 2
%rvar - Shape function of radius
%ze -- shape function parameter and integrating variable
%d_matrix -- D-matrix (4 by 4)
%b_matrix -- B-matrix (6 by 4)
%btrans_matrix -- Transpose of B-matrix (4 by 6)
%inted -- B(transpose) x D x B (6 by 6)
%ingd -- Integrated matrix (6 by 6)
%k_stf -- Elemental stiffness matrix (6 by 6)
%
%Generate D-matrix
d_mat = zeros(6,6);
d_mat(1,1) = 1;
d_mat(1,2) = pois;
d_mat(2,1) = pois;
d_mat(2,2) = 1;
d_mat(3,3) = (1-pois)/2;
d_mat(4,4) = tube_thic^2/12;
d_mat(4,5) = pois*tube_thic^2/12;
d_mat(5,4) = pois*tube_thic^2/12;
d_mat(5,5) = tube_thic^2/12;
d_mat(6,6) = (1-pois)*tube_thic^2/24;
```

```
d_matrix = [(youn_cy*1000000000*tube_thic)/(1-pois^2)].*d_mat;
%
r1 = tradi + (jjs-1)/tube_num*(bradi - tradi);
r2 = tradi + jjs/tube_num*(bradi - tradi);
%
%Define B-matrix
syms ze rvar real;
rvar = r1*(1-ze) + r2*ze;
%
umat = [(1-ze) 0 0 0 ze 0 0 0];
vmat = [0 (1-ze) 0 0 0 ze 0 0];
wmat = [0 0 (1-3*ze^2+2*ze^3) (tube_leng*(ze-2*ze^2+ze^3)) 0 0 (3*ze^2-2*ze^3)
(tube_leng*(-ze^2+ze^3))];
%
b_matrix(1,:) = (1/tube_leng).*diff(umat, ze);
b_matrix(2,:) = (nwav/rvar).*vmat + (1/rvar).*(umat.*sin(taper) - wmat.*cos(taper));
b_matrix(3,:) = (1/tube_leng).*diff(vmat, ze) - (1/rvar).*vmat.*sin(taper) -
(nwav/rvar).*umat;
b_matrix(4,:) = (1/tube_leng^2).*diff(wmat, ze, 2);
b_matrix(5,:) = (-nwav^2/rvar^2).*wmat + (nwav/rvar^2).*vmat.*cos(taper) +
(sin(taper)/rvar).*(1/tube_leng).*diff(wmat, ze);
b_matrix(6,:) = (-2*nwav/rvar/tube_leng).*diff(wmat, ze) +
(nwav/rvar^2).*wmat.*sin(taper) + (1/rvar).*(1/tube_leng).*diff(vmat, ze).*cos(taper) -
(1/rvar^2).*vmat.*sin(taper).*cos(taper);
%
%Formation of transpose b matrix
btrans_matrix = b_matrix';
%Formation of Integrand Matrix
inted=btrans_matrix*d_matrix*b_matrix.*rvar;
%
%Integrating to form the elemental stiffness matrix
ingd=int(inted, 0, 1);
%
%Transformation of local to global coordinates
%
tmat = [cos(taper) 0 -sin(taper) 0
   0 1 0 0
   sin(taper) 0 cos(taper) 0
   0 0 0 1];
transmat = [tmat zeros(4, 4); zeros(4, 4) tmat];
%
k_stf=pi.*tube_leng.*ingd;
stiff_stf = transmat'*k_stf*transmat;
k_cydim=double(stiff_stf);
%
return
```

```
function k_cylin = k_tubeglo (k_cydim, jjs, tdof)
%
%To position the elemental stiffness matrix at proper
%position of global matrix for tapered cylindrical elements.
%
%Define variables
%
%*Input arguments*
%k_cylin -- Elemental stiffness matrix in global dimensions of tapered tube (tdof by tdof)
%jjs -- Number of iteration of "jj"th element
%tdof -- Total number of DOFs
%
%*Output argument*
%k_cydim -- Elemental stiffness matrix in local dimensions
%
k_cylin=zeros(tdof, tdof);
k_cylin(1+(jjs-1)*4 : (jjs+1)*4, 1+(jjs-1)*4 : (jjs+1)*4)=k_cydim;
%
return
```

```
function k_pla = kplglo_bo(k_plele4, jjs, tdof)
%
%Tto position the elemental stiffness matrix at proper
%position of entire plate matrix.
%
%Define variables
%
%*Input arguments*
%k_plele4 -- Elemental stiffness matrix (4 x 4)
%jjs -- Number of iteration of "jj"th element
%tdof -- Total number of DOFs
%
%*Output argument*
%k_pla -- Elemental stiffness matrix (tdof by tdof)
%
k_pla=zeros(tdof, tdof);
k_pla((tdof+1)-(jjs+1)*2 : tdof-(jjs-1)*2, (tdof+1)-(jjs+1)*2 : tdof-(jjs-1)*2)=k_plele4;
%
return
```

```
function k_pla = kplglo_to(k_plele4, jjs, tdof)
%
%To position the elemental stiffness matrix at proper
%position of entire plate matrix.
%
%Define variables
%
%*Input arguments*
%k_plele4 -- Elemental stiffness matrix (4 x 4)
%jjs -- Number of iteration of "jj"th element
%tdof -- Total number of DOFs
%
%*Output argument*
%k_pla -- Elemental stiffness matrix (tdof by tdof)
%
k_pla=zeros(tdof, tdof);
k_pla(1+(jjs-1)*2 : (jjs+1)*2, 1+(jjs-1)*2 : (jjs+1)*2)=k_plele4;
%
return
```

```
function m_plele4 = m_boplate (a, b, bopl_thic, dens_pl)
%
%To compute the mass matrix for an
%axisymmetric annular element subjected to vibration
%
%Refer to paper "Static, vibration and buckling analysis of axisymmetric
%circular plates using finite elements", Gerard C. Pardoen, Computers and
%Structures, Vol.3, pp.355-375.
%
%Define variables
%
%*Inputs*
%a -- outer radius of annulur element (m)
%b -- inner radius of annulur element (m)
%dens_pl -- Density of top and bottom plate material (kg/m3)
%bopl_thic -- Thickness of top plate (assume uniform) (m)
%
%*Output*
%m_plele4 -- Elemental mass matrix (4 x 4)
%
%*Symbols*
%r -- radius
%
%*Others*
%al -- a/b
%d_value -- Plate rigidity (Nm)
%mplate -- Elemental mass matrix in symbolic form
%
%*Matrices*
%ci -- Matrix of shape functions
%ci_trans - Transpose of ci
%inted -- ci(transpose) x ci (4 by 4)
%ingd -- Integrated matrix (4 by 4)
%
syms r;
al = a/b;
%
%Definition of nu terms for deriving the displacement shape function
%
delta = (al^2 - 1)^2 - 4*al^2*(log(al))^2;
nu11 = (al^2) * (al^2 - 1 + 2*log(al) - 4*(log(al))^2);
nu12 = (1 - 2*al^2*log(al) - al^2);
nu13 = 4*al^2*log(al);
nu14 = 2*(al^2 - 1);
nu21 = 2*b*al^2*(log(al))^2;
nu22 = -nu21;
```

```
nu23 = b*al^2*(al^2 - 1 - 2*log(al));
nu24 = b*(2*al^2*log(al) - al^2 +1);
nu31 = nu12;
nu32 = -nu12;
nu33 = -nu13;
nu34 = -nu14;
nu41 = b*al*log(al)*(al^2 - 1);
nu42 = -nu41;
nu43 = b*al*(2*al^2*log(al) - al^2 + 1);
nu44 = b*al*(al^2 - 1 - 2*log(al));
%
%Definition of rho terms
%
ro1 = 1;
ro2 = (r/b)^2;
ro3 = log(r/b);
ro4 = (r/b)^2*log(r/b);
%
%ci1 is the shape function of w1 displacement
%ci2 is the shape function of w2 displacement
%ci3 is the shape function of w3 displacement
%ci4 is the shape function of w4 displacement
%
ci1 = (nu11*ro1 + nu12*ro2 + nu13*ro3 + nu14*ro4)/delta;
ci2 = (nu21*ro1 + nu22*ro2 + nu23*ro3 + nu24*ro4)/delta;
ci3 = (nu31*ro1 + nu32*ro2 + nu33*ro3 + nu34*ro4)/delta;
ci4 = (nu41*ro1 + nu42*ro2 + nu43*ro3 + nu44*ro4)/delta;
ci = [ci1 ci2 ci3 ci4];
ci_trans = [ci1
   ci2
   ci3
   ci4];
%Formation of Integrand Matrix
inted = r.*ci_trans*ci;
%
%Integrating to form the elemental stiffness matrix
ingd=int(inted, b, a);
mplate=pi.*dens_pl.*bopl_thic.*ingd;
m_pldouble = double(mplate);
%
%Derive matrix for bottom plate
m_plele4(1,1) = m_pldouble(3,3);
m_plele4(2,1) = m_pldouble(4,3);
m_plele4(3,1) = m_pldouble(1,3);
m_plele4(4,1) = m_pldouble(2,3);
%
```

```
m_plele4(1,2) = m_pldouble(3,4);
m_plele4(2,2) = m_pldouble(4,4);
m_plele4(3,2) = m_pldouble(1,4);
m_plele4(4,2) = m_pldouble(2,4);
%
m_plele4(1,3) = m_pldouble(3,1);
m_plele4(2,3) = m_pldouble(4,1);
m_plele4(3,3) = m_pldouble(1,1);
m_plele4(4,3) = m_pldouble(2,1);
%
m_plele4(1,4) = m_pldouble(3,2);
m_plele4(2,4) = m_pldouble(4,2);
m_plele4(3,4) = m_pldouble(1,2);
m_plele4(4,4) = m_pldouble(2,2);
%
return
```

```
function mass_cent = m_centplate(plate_thic, dens_pl, pradi, eplno)
%
%To compute the mass matrix for the
%centre plate
%
%Refer to paper "Static, vibration and buckling analysis of axisymmetric
%circular plates using finite elements", Gerard C. Pardoen, Computers and
%Structures, Vol.3, pp.355-375.
%
%Define variables
%
%*Inputs*
%dens_pl -- Density of top and bottom plate material (kg/m3)
%eplno -- No. of plate elements (including the centre plate)
%pradi-- Radius of plate (m)
%plate_thic -- Thickness of plate (assume uniform) (m)
%
%*Output*
%mass_cent -- Mass matrix of centre plate (2 x 2)
%
%Assume the rigid plate to represent centre plate
mass_cent = zeros(2,2);
mass_cent(1,1) = dens_pl*plate_thic*pi*(pradi/eplno)^2;
mass_cent(1,2) = -(dens_pl*plate_thic*pi*(pradi/eplno)^3)/4;
mass_cent(2,1) = -(dens_pl*plate_thic*pi*(pradi/eplno)^3)/4;
mass_cent(2,2) = (dens_pl*plate_thic*pi*(pradi/eplno)^4)/12;
%
return
```

```
function m_plele4 = m_toplate (a, b, topl_thic, dens_pl)
%
%To compute the mass matrix for an
%axisymmetric annular element subjected to vibration
%
%Refer to paper "Static, vibration and buckling analysis of axisymmetric
%circular plates using finite elements", Gerard C. Pardoen, Computers and
%Structures, Vol.3, pp.355-375.
%
%Define variables
%
%*Inputs*
%a -- outer radius of annulur element (m)
%b -- inner radius of annulur element (m)
%dens_pl -- Density of top and bottom plate material (kg/m3)
%topl_thic -- Thickness of top plate (assume uniform) (m)
%
%*Output*
%m_plele4 -- Elemental mass matrix (4 x 4)
%
%*Symbols*
%r -- radius
%
%*Others*
%al -- a/b
%d_value -- Plate rigidity (Nm)
%mplate -- Elemental mass matrix in symbolic form
%
%*Matrices*
%ci -- Matrix of shape functions
%ci_trans - Transpose of ci
%inted -- ci(transpose) x ci (4 by 4)
%ingd -- Integrated matrix (4 by 4)
%
syms r;
al = a/b;
%
%Definition of nu terms for deriving the displacement shape function
%
delta = (al^2 - 1)^2 - 4*al^2*(log(al))^2;
nu11 = (al^2) * (al^2 - 1 + 2*log(al) - 4*(log(al))^2);
nu12 = (1 - 2*al^2*log(al) - al^2);
nu13 = 4*al^2*log(al);
nu14 = 2*(al^2 - 1);
nu21 = 2*b*al^2*(log(al))^2;
nu22 = -nu21;
```

```
nu23 = b*al^2*(al^2 - 1 - 2*log(al));
nu24 = b*(2*al^2*log(al) - al^2 +1);
nu31 = nu12;
nu32 = -nu12;
nu33 = -nu13;
nu34 = -nu14;
nu41 = b*al*log(al)*(al^2 - 1);
nu42 = -nu41;
nu43 = b*al*(2*al^2*log(al) - al^2 + 1);
nu44 = b*al*(al^2 - 1 - 2*log(al));
%
%Definition of rho terms
%
ro1 = 1;
ro2 = (r/b)^2;
ro3 = log(r/b);
ro4 = (r/b)^2*log(r/b);
%
%ci1 is the shape function of w1 displacement
%ci2 is the shape function of w2 displacement
%ci3 is the shape function of w3 displacement
%ci4 is the shape function of w4 displacement
%
ci1 = (nu11*ro1 + nu12*ro2 + nu13*ro3 + nu14*ro4)/delta;
ci2 = (nu21*ro1 + nu22*ro2 + nu23*ro3 + nu24*ro4)/delta;
ci3 = (nu31*ro1 + nu32*ro2 + nu33*ro3 + nu34*ro4)/delta;
ci4 = (nu41*ro1 + nu42*ro2 + nu43*ro3 + nu44*ro4)/delta;
ci = [ci1 ci2 ci3 ci4];
ci_trans = [ci1
   ci2
   ci3
   ci4];
%Formation of Integrand Matrix
inted = r.*ci_trans*ci;
%
%Integrating to form the elemental stiffness matrix
ingd=int(inted, b, a);
mplate=pi.*dens_pl.*topl_thic.*ingd;
m_plele4 = double(mplate);
%
return
```

```
function mass_cydim = mass_tube (dens_cy, tube_thic, tradi, bradi, tube_leng,
tube_num, taper, jjs, nwav)
%
%To compute the elemental mass matrix for tapered cylindrical elements.
%
%Define variables
%
%*Input arguments*
%tdof -- Total number of DOFs
%dens_cy -- Density of tapered cylinder material (kg/m3)
%tube_thic -- Thickness of tapered tube (assume uniform) (m)
%tradi -- Top radius of cone (m)
%bradi -- Bottom radius of cone (m)
%tube_num -- No. of taper cylindrical elements
%tube_leng -- Length of each element on the cylindrical tube (m)
%taper - Taper angle of cone
%jjs -- Number of iteration of "jj"th element
%
%*Output argument*
%mass_cydim -- Double of mass_stf
%
%*Others*
%r1 -- Radius at node 1
%r2 -- Radius at node 2
%zi -- shape function parameter and integrating variable
%rvar - Shape function of radius
%n_matrix -- N-matrix (2 by 6)
%ntrans_matrix -- N-matrix transpose (6 by 2)
%inted -- N(transpose) x N (6 by 6)
%ingd -- Integrated matrix (6 by 6)
%mass_stf -- Elemental stiffness matrix (6 by 6)
%
%Generate N-matrix
syms rvar zi real;
r1 = tradi + (jjs-1)/tube_num*(bradi - tradi);
r2 = tradi + jjs/tube_num*(bradi - tradi);
rvar = r1*(1-zi) + r2*zi;
%
n_matrix(1,:) = [1-zi 0 0 0 zi 0 0 0];
%
n_matrix(2,:) = [0 1-zi 0 0 0 zi 0 0];
%
n_matrix(3,:) = [0 0 1-3*zi^2+2*zi^3 tube_leng*(zi-2*zi^2+zi^3) 0 0 3*zi^2-2*zi^3
tube_leng*(-zi^2+zi^3)];
%
```

```
n_mat(1,:) = [0 0 (-6*zi+6*zi^2)/tube_leng (1-4*zi+3*zi^2) 0 0 (6*zi-6*zi^2)/tube_leng
(-2*zi+3*zi^2)];
%
n_mat(2,:) = [0 -(1-zi) (-nwav/rvar)*(1-3*zi^2+2*zi^3) (-nwav/rvar)*tube_leng*(zi-
2*zi^2+zi^3) 0 -zi (-nwav/rvar)*(3*zi^2-2*zi^3) (-nwav/rvar)*tube_leng*(-zi^2+zi^3)];
%Define transpose of n matrix
ntrans_matrix = n_matrix';
ntrans_mat = n_mat';
%
%Formation of Integrand Matrix
inted1=ntrans_matrix*n_matrix.*rvar;
inted2=ntrans_mat*n_mat.*rvar;
%
%Integrating to form the elemental stiffness matrix
ingd1=int(inted1, 0, 1);
ingd2=int(inted2, 0, 1);
%
%Transformation of local to global coordinates
%
tmat = [cos(taper) 0 -sin(taper) 0
   0 1 0 0
   sin(taper) 0 cos(taper) 0
   0 0 0 1];
transmat = [tmat zeros(4, 4); zeros(4, 4) tmat];
%
m_stf=pi.*tube_leng.*tube_thic.*dens_cy.*ingd1 +
(pi*tube_leng*tube_thic^3*dens_cy/12).*ingd2;
mass_stf = transmat'*m_stf*transmat;
mass_cydim=double(mass_stf);
%
return
```

```
function mass_cylin = mass_tubeglo (mass_cydim, jjs, tdof)
%
%To position the elemental mass matrix at proper
%position of global matrix for tapered cylindrical elements.
%
%Define variables
%
%*Input arguments*
%jjs -- Number of iteration of "jj"th element
%mass_cydim -- Elemental mass matrix in local dimensions
%tdof -- Total number of DOFs
%
%*Output argument*
%mass_cylin -- Elemental mass matrix in global dimensions (tdof by tdof)
%
mass_cylin=zeros(tdof, tdof);
mass_cylin(1+(jjs-1)*4 : (jjs+1)*4, 1+(jjs-1)*4 : (jjs+1)*4)=mass_cydim;
%
return
```

```
function m_pla = mplglo_bo(m_plele4, jjs, tdof)
%
%To position the elemental mass matrix at proper
%position of global matrix for bottom plate.
%
%Define variables
%
%*Input arguments*
%jjs -- Number of iteration of "jj"th element
%m_plele4 -- Elemental mass matrix (4 x 4)
%tdof -- Total number of DOFs
%
%*Output argument*
%m_pla -- Elemental mass matrix (tdof by tdof)
%
m_pla=zeros(tdof, tdof);
m_pla((tdof+1)-(jjs+1)*2 : tdof-(jjs-1)*2, (tdof+1)-(jjs+1)*2 : tdof-(jjs-1)*2)=m_plele4;
%
return
```

```
function m_pla = mplglo_to(m_plele4, jjs, tdof)
%
%PTo position the elemental mass matrix at proper
%position of global matrix for top plate.
%
%Define variables
%
%*Input arguments*
%jjs -- Number of iteration of "jj"th element
%m_plele4 -- Elemental mass matrix (4 x 4)
%tdof -- Total number of DOFs
%
%*Output argument*
%%m_pla -- Elemental mass matrix (tdof by tdof)
%
m_pla=zeros(tdof, tdof);
m_pla(1+(jjs-1)*2 : (jjs+1)*2, 1+(jjs-1)*2 : (jjs+1)*2)=m_plele4;
%
return
```

```
function orflex = orient(flex, topl_num, tube_num, bopl_num)
%
%To orient the flex matrix so that
%it is properly oriented to the coordinate system in SCATT (i.e.
%pressure vector inwards, displacement vector outwards positive)
%
%*Inputs*
%flex -- Flexibility matrix for shell lateral displacement
%topl_num -- No. of elements at top end circular plate
%bopl_num -- No. of elements at bottom end circular plate
%tube_num -- No. of taper cylindrical elements
%
%*Outputs*
%orflex - properply oriented flex matrix
%
%*Others*
%dof - Total number of dofs
%ormat - orientation matrix
%
dof = 2*(topl_num + tube_num + bopl_num - 1);
ormat(1:dof, 1:dof) = 1;
ormat(1:2*(topl_num + tube_num - 1), 1:2*(topl_num + tube_num - 1)) = -1;
ormat(2*(topl_num + tube_num - 1) + 1 : dof, 2*(topl_num + tube_num - 1) + 1 : dof) =
-1;
orflex = ormat.*flex;
return
```

```
function [tube_leng, topl_leng, bopl_leng, tradi, bradi, frad, topl_num, bopl_num,
tube_num, taper] = para_compute_cy (cone_ht, top_diam, bot_diam, ka)
%
%Purpose is to compute some essential parameters from those defined by user
%
%Define variables:
%
%Output arguments:
%tube_leng -- Length of each element on the cylindrical tube (m)
%topl_leng -- Length of each element on top end plate (m)
%bopl_leng -- Length of each element on bottom end plate (m)
%tradi -- Top radius of cone (m)
%bradi -- Bottom radius of cone (m)
%frad -- Excitation frequency (rad/s)
%topl_num -- No. of elements at top end circular plate
%bopl_num -- No. of elements at bottom end circular plate
%tube_num -- No. of taper cylindrical elements
%taper - Taper angle of cone
%
%Input arguments:
%top_diam -- Top diameter of cone (m)
%bot_diam -- Bottom diameter of cone (m)
%cone_ht -- Height of cone (m)
%ka -- ka of the excitation frequency
%
%Others:
%waveleng -- Wavelength of acoustics waves
%mradi -- Mean radius of cone (m)
%cone_leng -- Total length of cone (m)
%
%Calculations
%
tradi = 0.5*top_diam;
bradi = 0.5*bot_diam;
mradi = 0.5*(tradi + bradi);
%
frad = ka/mradi*1500;
%
waveleng = 2*pi*mradi/ka;
topl_num = round(tradi*8/waveleng);
bopl_num = round(bradi*8/waveleng);
%
cone_leng = sqrt(cone_ht^2 + (bradi - tradi)^2);
%
tube_num = round(cone_leng*4/waveleng);
%
```

```
tube_leng = cone_leng/tube_num;
topl_leng = tradi/topl_num;
bopl_leng = bradi/bopl_num;
%
taper = atan((bradi - tradi)/cone_ht);
return
```

```
function qmat = shape_bopl2(kk, bopl_leng, bopl_num)
%
%This computes the conversion matrix from nodal force to elemental pressure
%for bottom annulur plate
%
a = (bopl_num - kk + 1)*bopl_leng;
b = (bopl_num - kk)*bopl_leng;
syms r
al = a/b;
%
%Definition of nu terms for deriving the displacement shape function
%
delta = (al^2 - 1)^2 - 4*al^2*(log(al))^2;
nu11 = (al^2) * (al^2 - 1 + 2*log(al) - 4*(log(al))^2);
nu12 = (1 - 2*al^2*log(al) - al^2);
nu13 = 4*al^2*log(al);
nu14 = 2*(al^2 - 1);
nu21 = 2*b*al^2*(log(al))^2;
nu22 = -nu21;
nu23 = b*al^2*(al^2 - 1 - 2*log(al));
nu24 = b*(2*al^2*log(al) - al^2 +1);
nu31 = nu12;
nu32 = -nu12;
nu33 = -nu13;
nu34 = -nu14;
nu41 = b*al*log(al)*(al^2 - 1);
nu42 = -nu41;
nu43 = b*al*(2*al^2*log(al) - al^2 + 1);
nu44 = b*al*(al^2 - 1 - 2*log(al));
%
%Definition of rho terms
%
ro1 = 1;
ro2 = (r/b)^2;
ro3 = log(r/b);
ro4 = (r/b)^2*log(r/b);
%
%ci1 is the shape function of w1 displacement
%ci2 is the shape function of w2 displacement
%ci3 is the shape function of w3 displacement
%ci4 is the shape function of w4 displacement
%
ci1 = (nu11*ro1 + nu12*ro2 + nu13*ro3 + nu14*ro4)/delta;
ci2 = (nu21*ro1 + nu22*ro2 + nu23*ro3 + nu24*ro4)/delta;
ci3 = (nu31*ro1 + nu32*ro2 + nu33*ro3 + nu34*ro4)/delta;
ci4 = (nu41*ro1 + nu42*ro2 + nu43*ro3 + nu44*ro4)/delta;
```

```
ci = [ci3; ci4; ci1; ci2];
inted = ci.*r;
qsy = int(inted, b, a);
qmat = double(qsy);
return
```

```
function qmat = shape_pl(pl_leng)
%
%This computes the conversion matrix from nodal force to elemental pressure
%for centre circular plate
%
syms r;
ph1 = 1;
ph2 = pl_leng/2*((r/pl_leng)^2 - 1);
inted = [ph1; ph2].*r;
qsy = int(inted, 0, pl_leng);
qmat = double(qsy);
return
```

```
function qmat = shape_topl2(ii, topl_leng)
%
%This computes the conversion matrix from nodal force to elemental pressure
%for top annulur plate
%
a = ii*topl_leng;
b = (ii-1)*topl_leng;
syms r
al = a/b;
%
%Definition of nu terms for deriving the displacement shape function
%
delta = (al^2 - 1)^2 - 4*al^2*(log(al))^2;
nu11 = (al^2) * (al^2 - 1 + 2*log(al) - 4*(log(al))^2);
nu12 = (1 - 2*al^2*log(al) - al^2);
nu13 = 4*al^2*log(al);
nu14 = 2*(al^2 - 1);
nu21 = 2*b*al^2*(log(al))^2;
nu22 = -nu21;
nu23 = b*al^2*(al^2 - 1 - 2*log(al));
nu24 = b*(2*al^2*log(al) - al^2 +1);
nu31 = nu12;
nu32 = -nu12;
nu33 = -nu13;
nu34 = -nu14;
nu41 = b*al*log(al)*(al^2 - 1);
nu42 = -nu41;
nu43 = b*al*(2*al^2*log(al) - al^2 + 1);
nu44 = b*al*(al^2 - 1 - 2*log(al));
%
%Definition of rho terms
%
ro1 = 1;
ro2 = (r/b)^2;
ro3 = log(r/b);
ro4 = (r/b)^2*log(r/b);
%
%ci1 is the shape function of w1 displacement
%ci2 is the shape function of w2 displacement
%ci3 is the shape function of w3 displacement
%ci4 is the shape function of w4 displacement
%
ci1 = (nu11*ro1 + nu12*ro2 + nu13*ro3 + nu14*ro4)/delta;
ci2 = (nu21*ro1 + nu22*ro2 + nu23*ro3 + nu24*ro4)/delta;
ci3 = (nu31*ro1 + nu32*ro2 + nu33*ro3 + nu34*ro4)/delta;
ci4 = (nu41*ro1 + nu42*ro2 + nu43*ro3 + nu44*ro4)/delta;
```

```
ci = [ci1; ci2; ci3; ci4];
inted = ci.*r;
qsy = int(inted, b, a);
qmat = double(qsy);
return
```

```
function qmat = shape_tu(jj, tube_leng, tradi, bradi, tube_num, taper)
%
%This computes the conversion matrix from nodal force to elemental pressure
%for conical shell
%
syms rvar zi real;
r1 = tradi + (jj-1)/tube_num*(bradi - tradi);
r2 = tradi + jj/tube_num*(bradi - tradi);
rvar = r1*(1-zi) + r2*zi;
%
ph1 = 1-3*zi^2+2*zi^3;
ph2 = tube_leng*(zi-2*zi^2+zi^3);
ph3 = 3*zi^2-2*zi^3;
ph4 = tube_leng*(-zi^2+zi^3);
%
inted = [ph1; ph2; ph3; ph4].*rvar;
qsy = int(inted, 0, 1);
qmat = double(qsy);
return
```

```
function [immata, fluidense] = stiffglobe(flex3d, topl_num, topl_leng, tube_num,
tube_leng, bopl_num, bopl_leng, taper, frad)
%
%To compute the 3-D dynamic flexibility matrix for
%tapered cylindrical tube with flat endcaps.
%
%*Inputs*
%topl_num -- No. of elements at top end circular plate
%bopl_num -- No. of elements at bottom end circular plate
%tube_num -- No. of taper cylindrical elements
%tube_leng -- Length of each element on the cylindrical tube (m)
%topl_leng -- Length of each element on top end plate (m)
%bopl_leng -- Length of each element on bottom end plate (m)%diam -- Diameter of
cylinder (m)
%tube_leng -- Length of tube (m)
%taper - Taper angle of cone
%frad -- Excitation frequency (rad/s)
%
%*Others*
%angle_step -- User input on number of angular steps required for the
%dynamic stiffness matrix.
%fluidense -- Density of fluid medium (kg/m3)
%
tradi = topl_num*topl_leng;
bradi = bopl_num*bopl_leng;
d = size(flex3d);
angle_step = input('Enter the number of angular steps for impedance matrix:');
%
%Getting fourier terms to represent unit force at theta=0
fourier_term = fourier_angle(angle_step);
%
fluidense = input('Enter the fluid density (kg/m3):');
%
%F = CONV * PSI
%where CONV = {Area*Density*frequency^2)
%For CONV matrix
%
qmat = for_mat(topl_num, bopl_num, tube_num, topl_leng, tube_leng, bopl_leng, taper,
tradi, bradi);
qmatrix = (pi*fluidense*frad^2).*qmat;
umat = dipmat(topl_num, bopl_num, tube_num, topl_leng, tube_leng, bopl_leng);
%
umat(topl_num+1, 2*topl_num-1) = 0;
umat(topl_num+tube_num, 2*(topl_num+tube_num)-1) = 0;
umat(topl_num+tube_num, 2*(topl_num+tube_num)) = -umat(topl_num+tube_num,
2*(topl_num+tube_num));
```

```
qmatrix(2*topl_num-1, topl_num+1) = 0;
qmatrix(2*(topl_num+tube_num)-1, topl_num+tube_num) = 0;
qmatrix(2*(topl_num+tube_num), topl_num+tube_num) = -
qmatrix(2*(topl_num+tube_num), topl_num+tube_num);
%
for qq = 1 : d(3)
  immat(: , :, qq) = umat*flex3d(:, :, qq)*qmatrix;
end
%
%Compute the dynamic flexibility matrix
%
d11 = topl_num+bopl_num+tube_num;
for ii = 1:d11
   displ = zeros(d11*angle_step, 1);
   for kk = 1:angle_step
      for jj = 1:d(3)
         displ((kk-1)*d11 + 1 : kk*d11, 1) = displ((kk-1)*d11 + 1 : kk*d11, 1) + immat(:,
ii, jj).*fourier_term(jj,1).*cos((jj-1)*(kk-1)*2*pi/angle_step);
      end
   end
   immata(:, ii) = displ;
end
return
```

```
function mmat_to = toplate_mass(topl_thic, tradi, dens_pl, topl_num, nwav)
%
%To assemble the mass matrix for top plate
%
%*Define variables*
%
%*Input arguments*
%topl_thic -- Thickness of top plate (assume uniform) (m)
%tradi -- Top radius of cone (m)
%dens_pl -- Density of top and bottom plate material (kg/m3)
%topl_num -- No. of elements at top end circular plate
%
%*Other arguments*
%total_node -- Total number of nodes
%tdof -- Total number of DOFs
%jjs -- Number of iteration of "jj"th element
%a_val -- outer radius of annulur element (m)
%b_val -- inner radius of annulur element (m)
%
%*Matrices*
%m_plele4 -- Elemental mass matrix (4 x 4)
%m_pla -- Elemental mass matrix (tdof by tdof)
%mmat_to -- Entire plate mass matrix (tdof by tdof)
%mass_cent -- Mass matrix of centre plate (2 x 2)
%
%Prepare variables to input to elemental mass matrix computation
total_node = topl_num;
tdof = 2*total_node;
mmat_to = zeros(tdof, tdof);
%
%Add mass contribution by centre plate element
mass_cent = m_centplate(topl_thic, dens_pl, tradi, topl_num);
mmat_to(1:2, 1:2) = mass_cent;
%
%Assemble global mass matrix for top plate
for jjs = 1:topl_num-1
   b_val = jjs*tradi/topl_num;
   a_val = (jjs+1)*tradi/topl_num;
   m_pla = zeros(tdof, tdof);
   %Formation of 4x4 elemental mass matrix
   m_plele4 = m_toplate (a_val, b_val, topl_thic, dens_pl);
   %Formation of elemental mass matrix in entire plate matrix dimensions
   m_pla = mplglo_to (m_plele4, jjs, tdof);
   %Increment to entire plate mass matrix
   mmat_to = mmat_to + m_pla;
end
```

```
%
return
```

```
function kmat_to = toplate_stiff(topl_thic, tradi, youn_pl, pois, topl_num, nwav)
%
%To assemble the stiffness matrix for top plate
%
%*Define variables*
%
%*Input arguments*
%topl_thic -- Thickness of top plate (assume uniform) (m)
%tradi -- Top radius of cone (m)
%youn_pl -- Young's modulus of top and bottom plate material (x10e9 N/m2)
%pois -- Poisson's ratio of material
%topl_num -- No. of elements at top end circular plate
%nwav -- Wave number in consideration
%
%*Other arguments*
%total_node -- Total number of nodes
%tdof -- Total number of DOFs
%jjs -- Number of iteration of "jj"th element
%a_val -- Outer radius of annular (m)
%b_val -- Inner radius of annular (m)
%d_val -- Plate rigidity (Nm)
%
%*Matrices*
%k_plele4 -- Elemental stiffness matrix (4 x 4)
%k_pla -- Elemental stiffness matrix (tdof by tdof)
%kmat_to -- Entire plate stiffness matrix (tdof by tdof)
%
%Prepare variables to input to elemental stiffness matrix computation
total_node = topl_num;
tdof = 2*total_node;
kmat_to = zeros(tdof, tdof);
%
%Add stiffness of centre plate (only K22 is affected by the centre plate)
d_val = (youn_pl*1000000000*topl_thic^3)/(12*(1-pois^2));
kmat_to(2,2) = 2*d_val*pi*(1+pois);
%
%Assemble global stiffness matrix for plate only
for jjs = 1:topl_num-1
   k_pla = zeros(tdof, tdof);
   b_val = jjs*tradi/(topl_num);
   a_val = (jjs+1)*tradi/(topl_num);
   %Formation of 4x4 elemental stiffness matrix
   k_plele4 = k_toplate(a_val, b_val, pois, d_val, nwav);
   %Formation of tdof x tdof elemental stiffness matrix
   k_pla = kplglo_to(k_plele4, jjs, tdof);
   %Increment to entire plate stiffness matrix
```

```
  kmat_to = kmat_to + k_pla;
end
%
return
```

```
function mmat_tu = tube_mass(dens_cy, tube_thic, tradi, bradi, tube_leng, tube_num,
taper, nwav)
%
%Purpose is to assemble the mass matrix for tapered tube
%
%*Define variables*
%
%*Input arguments*
%dens_cy -- Density of tapered cylinder material (kg/m3)
%tube_thic -- Thickness of tapered tube (assume uniform) (m)
%tradi -- Top radius of cone (m)
%bradi -- Bottom radius of cone (m)
%tube_num -- No. of taper cylindrical elements
%tube_leng -- Length of each element on the cylindrical tube (m)
%taper - Taper angle of cone
%
%*Other arguments*
%total_node -- Total number of nodes
%tdof -- Total number of DOFs
%jjs -- Number of iteration of "jj"th element
%
%*Matrices*
%mass_cydim -- Elemental mass matrix in local dimensions
%mass_cylin -- Elemental mass matrix in global dimensions (tdof by tdof)
%mmat_tu -- Tapered tube mass matrix (tdof by tdof)
%
%Prepare variables to input to elemental mass matrix computation
total_node = tube_num + 1;
tdof = 4*total_node;
mmat_tu = zeros(tdof, tdof);
%
%Assemble global stiffness matrix for tube only
for jjs = 1:tube_num
   mass_cylin = zeros(tdof, tdof);
   %Formation of 6x6 elemental mass matrix
   mass_cydim = mass_tube (dens_cy, tube_thic, tradi, bradi, tube_leng, tube_num, taper,
jjs, nwav);
   %Formation of elemental mass matrix in global matrix dimensions
   mass_cylin = mass_tubeglo (mass_cydim, jjs, tdof);
   %Increment to global mass matrix
   mmat_tu = mmat_tu + mass_cylin;
end
return
```

```
function kmat_tu = tube_stiff(tube_thic, youn_cy, pois, tradi, bradi, tube_num,
tube_leng, taper, nwav)
%
%Purpose is to assemble the global stiffness matrix for tapered tube
%
%*Define variables*
%
%*Input arguments*
%tube_thic -- Thickness of tapered tube (assume uniform) (m)
%youn_cy -- Young's modulus of tapered cylinder material (x10e9 N/m2)
%pois -- Poisson's ratio of material
%tradi -- Top radius of cone (m)
%bradi -- Bottom radius of cone (m)
%tube_num -- No. of taper cylindrical elements
%tube_leng -- Length of each element on the cylindrical tube (m)
%taper - Taper angle of cone
%nwav -- Wave number in consideration
%
%*Other arguments*
%total_node -- Total number of nodes
%tdof -- Total number of DOFs
%jjs -- Number of iteration of "jj"th element
%
%*Matrices*
%k_cydim -- Elemental stiffness matrix in local dimensions
%k_cylin -- Elemental stiffness matrix in global dimensions of tapered tube (tdof by tdof)
%kmat_tu -- Stiffness Matrix (tube)
%
%Prepare variables to input to elemental stiffness matrix computation
total_node = tube_num + 1;
tdof = 4*total_node;
kmat_tu = zeros(tdof, tdof);
%
%Assemble global stiffness matrix for tapered tube only
for jjs = 1:tube_num
   k_cylin = zeros(tdof, tdof);
   %Formation of 6x6 elemental stiffness matrix
   k_cydim = k_tube (tube_thic, tradi, bradi, youn_cy, pois, tube_leng, tube_num, taper,
jjs, nwav);
   %Formation of elemental stiffness matrix in global matrix dimensions
   k_cylin = k_tubeglo (k_cydim, jjs, tdof);
   %Increment to global stiffness matrix
   kmat_tu = kmat_tu + k_cylin;
end
%
return
```