Coupling of Structural Dynamic and Acoustic Analyses Using the FEM and BEM

A thesis submitted to the Graduate School of the University of Cincinnati in partial fulfillment of the requirements for the degree of

Master of Science

in Mechanical Engineering, School of Dynamic Systems

of the College of Engineering & Applied Science

July 2012

By:

Xin Han

Bachelor of Engineering, Material Science and Engineering

Southeast University, Nanjing, P. R. China

Committee Chair:

Yijun Liu, Ph.D.

Committee Members:

Jay Kim, Ph.D.

Kumar Vemaganti, Ph.D.

Abstract

Noise from structural vibration is a major consideration in the design and manufacturing of a product, such as an automobile, airplane, and other consumer goods. Acoustic simulation is an important step to optimize the performance of many new products early in the design stage rather than correcting the mistakes afterwards. Effective and efficient numerical modeling and simulation methods will be important tools for noise prediction during the design stage of products.

This thesis work focuses on integrating the numerical methods in the prediction of noise from vibrating structures. To calculate the sound radiated from a vibrating structure, both a structural dynamic problem and an acoustic wave problem should be considered. In this study, the finite element method (FEM) is chosen for the structural dynamic analysis in order to calculate the natural frequencies and harmonic responses of the structure. The boundary element method (BEM) is used in the acoustic analysis of the structure to calculate the radiated sound field. In the BEM, only the surface of a sound radiating structure is discretized and the simulation of sound fields in unbounded domains is easy, yielding an efficient mesh generation and preprocessing. To couple the FEM analysis with the BEM analysis, a computer program, or translation code, is developed for mapping the model and velocity boundary condition (BC) for the acoustic analysis from the results of the dynamic analysis. Several element types in the FEM using the software ABAQUS® are implemented in the translation code and their performances are studied. Different BEM solvers in the software *FastBEM Acoustics®* developed earlier at the University of Cincinnati (UC) and based on the fast multipole method (FMM), the adaptive cross approximation (ACA) method and the fast conventional BEM are selected in this study.

The correctness and feasibility of the developed code are verified using a pulsating sphere model and a vibrating plate model. Numerical results and analytical data are found to be in good agreement using the developed translation code. As a large-scale and practical application, a wind turbine model is also used to study the noise propagation on the ground due to the vibration of the turbine structure and the rotation of turbine blades. It is found that the coupled analysis with the FEM and BEM can model the noise prediction of large-scale structures effectively and efficiently.

Table of Contents

1 Introduction

1.1 Background

Nowadays it is of high importance to produce low noise products for manufactures. There are policies that exist to minimize the noise level radiated by machinery, airports and high ways. Besides obeying these policies that set forth by the government, manufactures need to comply with the demands of the customer, who often prefer low noise designs. On the other hand, there are products that have to produce sound for functional reasons, like the speaker system. For these reasons, it is important to understand the behavior of mechanical structures that produce sound, especially to distinguish whether it is functional sound or noise [1].

There are different kinds of computer software that has been developed to answer the need of manufactures to be able to quantify the sound production of their product beforehand without the use of a prototype. With the ever increasing computing power of modern computers, more powerful software is introduced to perform simulations to calculate the radiated sound of complex structures.

Interior and exterior acoustic problems that are commonly found in engineering applications require a proper handling of the interaction between the vibrating structure and the acoustic fluid field. Analytical solutions to these acoustic fluid-structure interaction problems are only available when the structures have simple geometries with simple settings. For more realistic problems with complicated geometries, it is impossible to find analytical solutions, so hence effective numerical methods are required.

During the past decade, significant progresses have been made in computational methods for acoustic simulation, such as the finite element method (FEM), infinite finite element method (IFEM), and boundary element method (BEM).

1.2 Literature Review

The goal of this section is to review the current research of the structural vibration analysis, the acoustic wave analysis and the interaction between these two analyses.

1.2.1 Boundary Element Methods in Acoustics

A large amount of engineering experience is available for the FEM because of its wide applications in structural solid mechanics. However, the FEM is only applicable to bounded domains. For exterior acoustic problems, special schemes, e.g. artificial absorbing boundary conditions, perfectly matched layers or infinite elements, must be applied to quickly reduce the efficiency of the method. In contrast, when using the BEM, the Sommerfeld radiation condition is implicitly fulfilled [2]. A further advantage of the BEM is the reduction of the problem dimension: only the boundary of the sound radiating structure must be discretized. Therefore the cost of preprocessing and mesh generation is greatly reduced.

However the main disadvantage of the BEM is its full dense system matrices and the arising computing cost and memory requirements of order N^2, where N is the number of unknowns. Even for a moderate number of unknowns, the computational cost cannot be handled by standard implementations. Significant research efforts have been made in recent years to tackle this problem to develop fast BEM algorithms. The proposed algorithms can be divided into two major groups: matrix compression algorithms and techniques for the fast evaluation of the matrix-vector products. Bebendorf [3] have presented the adaptive cross approximation in 2000, which is the most prominent example of the first group. Another way to handle the BEM matrices is to use wavelets to compress the stiffness matrix [4]. In the second group, panel clustering method proposed by Hackbusch and Nowak [5] and fast multipole method (FMM) [6] are the dominate examples. In both of these, the dense matrix can be approximated by several sparse matrices.

It is important to understand FMM because it is regarded as one of the top ten algorithms of the last century and was first innovated by Rokhlin and Greengard [6-8] in the mid of 1980s. In conjunction with iterative equation solvers, FMM can reduce the matrix vector multiplication, as well as the memory requirement to $O(N^a \log N^b)$, where $1 \leq a < 2$ and $b > 0$ in solving BEM equation systems, with *N* being the number of unknowns. These advantages overcome the well-known drawbacks with regard to the computational efficiency and memory requirement of the conventional BEM and make the fast multipole BEM one of the most popular fast solution methods for the BEM. The application of the fast multipole algorithm to accelerate the BEM is triggered by the similarity between the potential in particle simulations and the fundamental solution in the kernels of the boundary integral operators. The key idea is to approximate the fundamental solution at some distance from the source point by a multipole series expansion. A multilevel scheme then allows the efficient evaluation of the matrix-vector product. More details about the fast multipole BEM can be found in Ref. [9] and [10].

1.2.2 Structure-Acoustic Interaction

Many researchers have investigated vibration-acoustic behavior from different points of views, by using different numerical techniques and different models.

Bernardi [11] proposed a mortar method that is adapted in the FEM-BEM coupling to yield a high flexibility in the choice of discretization. The mortar finite element discretization is a discontinuous Galerkin approximation. Because of the discontinuity on the interface, we classify the mortar finite element method as a nonconforming finite element method [12].

Gaul, Brunner and Junge [13] have discussed several different conforming fluid-structure coupling schemes. In their research, a fully coupled Burton-Miller approach [14] is presented. Secondly, a mortar coupling formulation is introduced. The coupled problems are solved with iterative solvers. Also, the different schemes are compared with respect to their accuracy, memory consumption and computation time using a simple spherical shell structure. With the mortar coupling formulations, different non-conforming meshes for the boundary element and finite element domain may be used on the coupling interface.

Gaul and Wenzel [15] use a hybrid boundary element formulation to describe the behavior of the compressible fluid. To introduce the new efficient computational procedure, the hybrid boundary element method (HBEM) in time domain is outlined. Emphasis is placed on the transformation of domain integrals into boundary integrals. It is shown that the HBEM formulation in time domain is similar to a FEM hyperelement formulation. A straight forward coupling formulation is obtained. In the frequency domain of the fluid formulation, pressure and flux are selected as field variables.

Frendi and Robinson [16] have studied the effect of acoustic coupling on plate vibrations using two numerical models of random and harmonic excitations. They concluded that the coupling is important at high excitation levels for both random and harmonic excitations, and therefore it should be taken into account for accurate prediction of the plate response. They also stated that the coupling is important for high frequencies, nonlinear structural response and acoustic radiation.

Lee [17] has studied the effect of structural-acoustic coupling on the nonlinear natural frequency of a rectangular box which consists of one flexible plate. The analysis has been performed by a finite element multi-mode approach and the effect of the cavity depth on the natural frequencies, and convergence studies have been discussed. It has been concluded that air cavity formed by a rectangular box has a negative stiffness effect on the flexible plate whereas a positive stiffness effect is present when the air pressure acting on plate is out of phase to the plate motion.

In recent years, the mathematical theories for domain decomposition methods have quickly advanced. Domain decomposition methods solve a boundary value problem by splitting it into smaller boundary value problems on subdomains and iterating to coordinate the solution between adjacent subdomains. The extended theoretical insight can give valuable motivation for the engineering application of coupling schemes. Braess and Dahmen [18] are concerned with the stability estimates of the mortar element method for three-dimensional problems. Ben Belgacem [19] proposed a mortar formulation with Lagrange multipliers to demonstrate the non-conforming domain decomposition technique.

1.3 Motivation

It is important to understand the acoustic sound prediction from the structure vibration analysis. In this thesis, a coupling method and program is proposed for coupling the FEM and BEM analyses. The main motivation for this coupled method is to develop the ability to predict the numerical solutions of sound pressure and sound power of a vibrating structure.

The commercial finite element program ABAQUS® [20] is used for the FEM analysis while the BEM program developed earlier at the UC and named *FastBEM Acoustic*® [21] is used for the BEM analysis. At the same time, LMS Virtual.Lab [22] is also used for some verification cases. The whole procedures for the coupled analysis are illustrated in Figure 1-1.

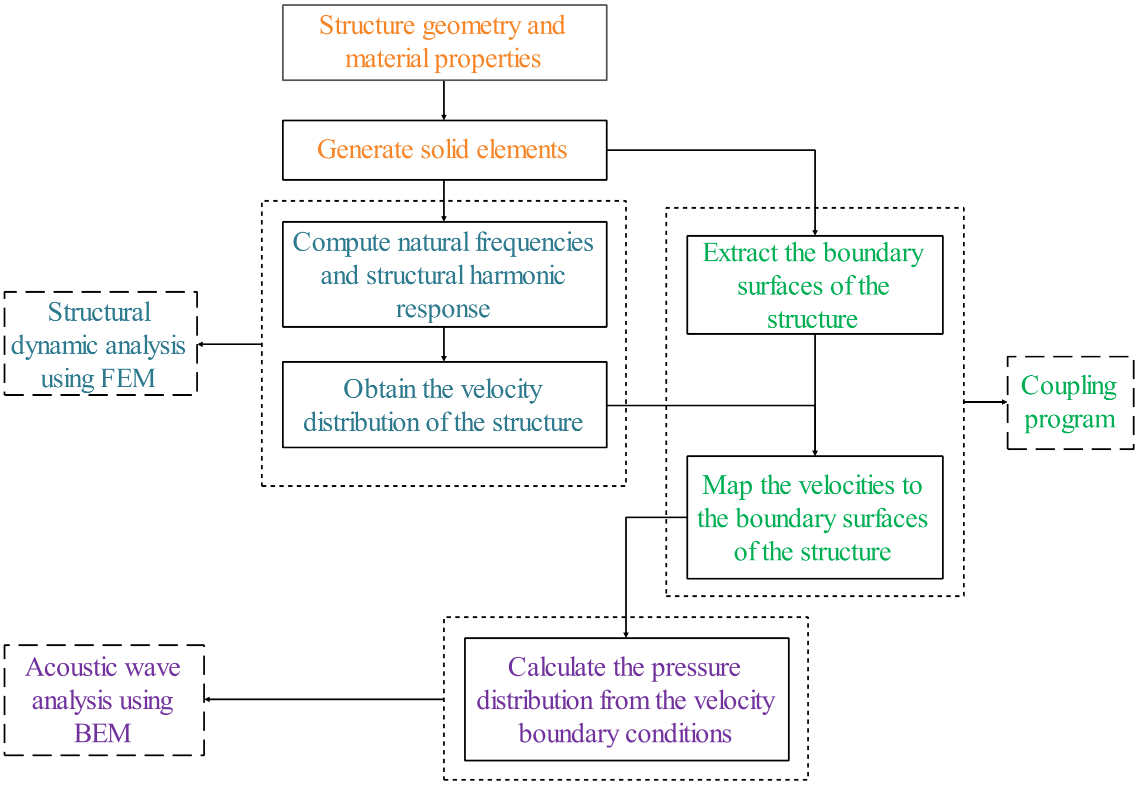

Structure geometry and material properties
Generate solid elements
Compute natural frequencies and structural harmonic response
Obtain the velocity distribution of the structure
Structural dynamic analysis using FEM
Extract the boundary surfaces of the structure
Map the velocities to the boundary surfaces of the structure
Coupling program
Calculate the pressure distribution from the velocity boundary conditions
Acoustic wave analysis using BEM

Figure 1-1 The whole procedures of the coupling of the structural dynamic analysis and acoustic wave analysis

1.4 Objective

This study develops and validates a translation program to couple the structure vibration analysis with acoustic wave analysis. The structure is discretized using the FEM, whereas for the acoustic field the BEM is used in the analysis. The whole translation code has the following capabilities:

1. It handles different types of solid elements, such as eight node brick elements, and six node wedge elements.
2. It processes the results in wide frequency spectrum of interest.
3. It generates the input files for BEM acoustic analysis.
4. And the parameters for BEM Solver could be specified from the program.

1.5 BEM Solver Used

In the FastBEM Acoustics program [21], there are three BEM solvers available: the fast multipole BEM, the ACA BEM and the conventional BEM. The fast multipole BEM with the iterative linear equation solver is the most efficient BEM solver for solving large-scale BEM models. However, its performance may slow down solving models at high frequencies (for example when the non-dimensional wavenumber *ka* is above 200, for which a large value of expansion order *p* is required). The ACA BEM also uses iterative equation solver and expansions of the coefficients. However, in small to medium sized models, it can be more efficient than the fast multipole BEM. Also, its efficiency is not very sensitive to the frequency. The drawback of the ACA BEM is that it requires a larger RAM size as compared with the fast multipole BEM. The conventional BEM uses a fast direct linear equation solver and delivers the most accurate result. However it is limited to solving smaller BEM model due to its requirement of large memory size and long CPU time.

For comparison, the results obtained from the software LMS Virtual.Lab are also provided shown in Chapter 5 and Chapter 6. The comparison is based on the same BEM mesh and type of elements.

1.6 Structure of the Thesis

The structure of this thesis is as follows: Chapter 1 gives an introduction of the background, motivation and objective of this research work. Chapter 2 summarizes the theory of structure vibration and how to solve this problem using the FEM. Chapter 3 discusses the theory of the acoustic waves and how to apply the BEM for the calculation. Chapter 4 discusses the coupling techniques between the FEM and BEM. Chapter 5 presents three verification cases. Chapter 6 details the practical application to a wind turbine problem to test the program. Finally in Chapter 7, some discussions and conclusions are given. The sample input files of ABAQUS and FastBEM Acoustics are listed in Appendix A and the translation programs for coupling the FEM and BEM analyses are listed in Appendix B.

2 Structural Dynamics and the Finite Element Method

The formulations for structural dynamics and the related FEM are reviewed in this chapter. Section 2.1 shows the general dynamic equations of the structural dynamic problem. Section 2.2 describes the procedure of natural frequency calculation and corresponding formulations. Section 2.3 discusses the frequency response analysis and formulations. Different finite element types are also presented in Section 2.4 to show different FEM modeling techniques. Section 2.5 shows the outputs from the harmonic response analysis to the BEM analysis.

2.1 Dynamic Equations

Equations that govern the dynamic response of a structure are derived by implementing the work of external forces to be balanced by the work of internal, inertia, and viscous forces for any small kinematical admissible motion. Equations of the structure can be written in the form [23]:

$$\mathbf{M}\frac{d^2\mathbf{D}}{dt^2}+\mathbf{C}\frac{d\mathbf{D}}{dt}+\mathbf{R}^{\text{int}}=\mathbf{R}^{ext}, \qquad (2.1)$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the damping matrix, and $\mathbf{R}^{ext}$ corresponds to applied load which is in general a function of time. For linear elastic material behavior, the internal force has:

$$\mathbf{R}^{\text{int}}=\mathbf{KD}, \qquad (2.2)$$

where **K** is the stiffness matrix, **D** is the displacement vector.

A popular damping scheme, called Rayleigh or proportional damping, forms damping matrix $\mathbf{C}$ as a linear combination of the stiffness and mass matrices, as demonstrated by the following formula:

$$\mathbf{C}=\alpha\mathbf{K}+\beta\mathbf{M}, \qquad (2.3)$$

where α and β are called, respectively, the stiffness and mass proportional damping constants.

2.2 Natural Frequency Analysis

An undamped structure, with no external loads applied to unstrained DOF (degree of freedom), undergoes harmonic motion in which each DOF moves in phase with all other DOF Thus,

$$\mathbf{D} = \bar{\mathbf{D}} \sin \omega t \text{ and } \frac{d^2\mathbf{D}}{dt^2} = -\omega^2 \bar{\mathbf{D}} \sin \omega t , \tag{2.4}$$

where $\bar{\mathbf{D}}$ is the amplitude of nodal DOF vibration and ω is circular frequency.

Combining Eq. (2.1) and Eq. (2.4), and with $\mathbf{C}$ and $\mathbf{R}^{ext}$ both zero, we obtain,

$$\left(\mathbf{K} - \lambda \mathbf{M}\right) \bar{\mathbf{D}} = \mathbf{0} , \tag{2.5}$$

where

$$\lambda = \omega^2 . \tag{2.6}$$

This is the basic statement of the vibration problem. Eq. (2.5) is called a generalized eigenproblem or simply an eigenproblem. To have non-zero solutions, the determinant of the matrix must vanish:

$$\det\left(\mathbf{K} - \lambda \mathbf{M}\right) = 0 , \tag{2.7}$$

which gives eigenvalues λ_i. Associated with each eigenvalue λ_i is the eigenvector $\bar{\mathbf{D}}_i$, which is sometimes called the normal (or natural, or characteristic, or principal) mode.

The natural frequency is a very important characteristic of the structure. When a structure is excited by a load with a frequency close to one of the structure's natural frequencies, the structure can undergo an extremely violent vibration, which often leads to catastrophic failure of the structure. This phenomenon is called resonance. Therefore, an eigenvalue analysis has to be performed when designing a structure that is to be subjected to dynamic loadings.

2.3 Harmonic Response

When the harmonic response of a dynamic system is solved by using the finite element method, the system of equations can be written in the following form [23]:

$$\left[-\omega^2\mathbf{M}+i\omega\mathbf{C}+\mathbf{K}\right]\mathbf{D}=\mathbf{F}, \tag{2.8}$$

where $\mathbf{D}$ is the nodal structural displacement vector and $\mathbf{F}$ is the harmonic external excitation force vector applied at the structure nodes.

2.4 Types of Finite Elements Considered

The following five finite element types are considered in this thesis:

a) Four node shell elements

b) Four node linear solid tetrahedron elements

c) Eight node linear solid brick elements

d) Eight node continuum shell brick elements

e) Six node continuum shell wedge elements

Shell elements are usually used to model structures in which one dimension, the thickness, is significantly smaller than the other dimensions. Shell elements use this condition to discretize a body by defining the geometry at a reference surface. In this case the thickness is defined through the section property definition. Shell elements have both displacement and rotational degrees of freedom.

In contrast, continuum shell elements discretize an entire three-dimensional body. The thickness is determined from the element nodal geometry. Continuum shell elements have only displacement degrees of freedom. From a modeling point of view continuum shell elements look like three-dimensional continuum solids, but their kinematic and constitutive behavior is similar to conventional shell elements.

The shapes of the above five element types and node ordering are shown in Figure 2-1 [20].

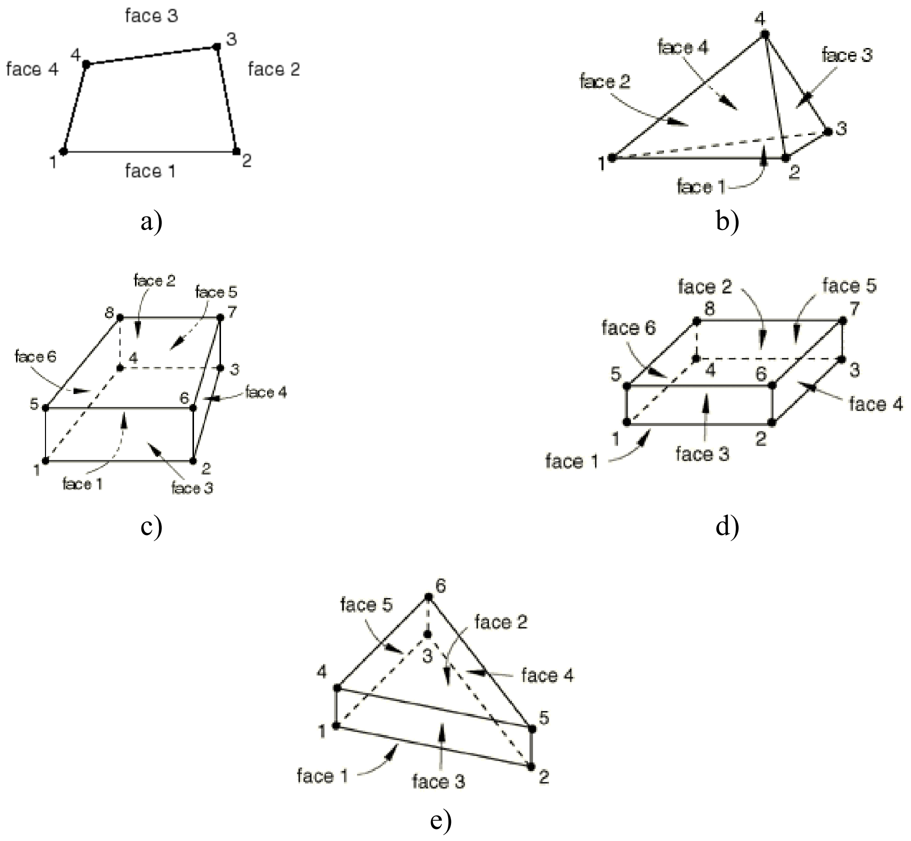


Figure 2-1 Shapes of different element types and node ordering – a) four node shell element b) four node linear solid tetrahedron element c) eight node linear solid brick element d) eight node continuum shell brick element e) six node continuum shell wedge element [20]

2.5 Output from the Harmonic Response Analyses

The ABAQUS procedure generates several files. Some of these files exist only while ABAQUS is executing and are deleted when a run completes. Other files contain analysis, post-processing, and translation results and are retained for use by other analysis options, restarting, or post-processing.

The ODB file, which is the output database, is required for the post-processing. The LOG file contains start and end times for modules run by the current ABAQUS execution procedure. The STA file, which is the state file, includes the increment summaries during the analysis. The PRT file, which is the part file, is used to store part and assembly information.

From the harmonic response analysis, the FEM results provide the displacements and velocities of all the nodes in the model. The boundary nodal velocity information is exported to the BEM code and is used as the boundary conditions (BCs) in the BEM analysis.

3 Acoustics and the Boundary Element Method

The formulations for acoustic analysis and the related BEM are reviewed in this chapter. Section 3.1 illustrates the general equations in acoustic wave problems. Section 3.2 describes how to solve the acoustic wave problem by the BEM. Section 3.3 shows the data structure of the BEM analysis.

3.1 Acoustic Wave Equations

Assuming an ideal fluid is at rest and homogeneous throughout the domain E, the governing equation for the acoustic wave propagation is a second-order partial differential equation, shown as the following [9]:

$$\nabla^2\phi - \frac{1}{c^2}\frac{\partial^2\phi}{\partial t^2} + Q\delta(\mathbf{x},\mathbf{x}_Q) = 0 \tag{3.1}$$

where $\phi = \phi(\mathbf{x},t)$ is the perturbation acoustic pressure at point $\mathbf{x}$ and time t, c is the speed of sound, $Q\delta(\mathbf{x},\mathbf{x}_Q)$ is a typical point source located at $\mathbf{x}_Q$ in acoustic domain E, and $\nabla^2(\ ) = \partial^2(\ )/\partial x_k \partial x_k = (\ )_{,kk}$ is the Laplace operator.

For the case of time harmonic wave problems, the point source intensity $Q = \hat{Q}e^{-i\omega t}$ and the solution to the governing equation (3.1) can be written as

$$\phi(\mathbf{x},t) = \hat{\phi}(\mathbf{x},\omega)e^{-i\omega t}\,, \tag{3.2}$$

where ω is the circular frequency, $i = \sqrt{-1}$, and $\hat{\phi}(\mathbf{x},\omega)$ represents the (complex) acoustic pressure in the frequency domain. Substituting Eq. (3.2) into the wave Eq. (3.1) yields

$$\nabla^2\hat{\phi}(\mathbf{x},\omega) + k^2\hat{\phi}(\mathbf{x},\omega) + \hat{Q}\delta(\mathbf{x},\mathbf{x}_Q) = 0\,. \tag{3.3}$$

Note that $k = \omega / c$ is called the wave number in the above equation.

For convenience, we drop the hats in Eq. (3.3), and then obtain the following governing equation for acoustic wave problems:

$$\nabla^2\phi(\mathbf{x},\omega)+k^2\phi(\mathbf{x},\omega)+Q\delta(\mathbf{x},\mathbf{x}_Q)=0\,, \tag{3.4}$$

which is known as the Helmholtz Equation.

The boundary conditions for the governing equation can be classified as follows [9]:

1. A given surface pressure (Dirichlet condition):

$$\phi=\overline{\phi}\,; \tag{3.5}$$

2. A given surface normal velocity (Neumann condition):

$$q\equiv\frac{\partial\phi}{\partial n}=\overline{q}=i\omega\rho v_n\,; \tag{3.6}$$

3. A given relation between the pressure and velocity on the surface (also known as impedance, mixed or Robin condition):

$$\phi=Zv_n\,, \tag{3.7}$$

where ρ is the mass density, v_n is the normal velocity, Z is the specific acoustic impedance, and the quantities with over-bars indicate given values.

3.2 BEM Formulations for Acoustics

Solving acoustic wave problems is one of the most important applications of the BEM, which can be applied to predict sound fields for noise control in automobiles, airplanes, and many other consumer products [9]. Acoustic waves often exist in an infinite medium outside a structure that is in vibration or impinged upon by an incident wave.

The governing equation for acoustic wave problems is the Helmholtz equation expressed by Eq. (3.4), which is replaced by an integral equation that only covers the boundary surface *S* in the BEM. This reduction in dimension is achieved through application of Green's second identity.

The resulting integral equation is known as the conventional BIE (CBIE). It relates the acoustic pressure and normal velocity on the closed boundary surface *S* of a vibrating object to the pressure field in the fluid domain [9]:

$$c(\mathbf{x})\phi(\mathbf{x}) = \int_S \left[G(\mathbf{x},\mathbf{y},\omega)q(\mathbf{y}) - F(\mathbf{x},\mathbf{y},\omega)\phi(\mathbf{y})\right]dS(\mathbf{y}) + \phi^I(\mathbf{x}) + QG(\mathbf{x},\mathbf{x}_Q,\omega)\,, \quad (3.8)$$

where $\mathbf{x}$ is the source point, $\mathbf{y}$ is the field point, $c(\mathbf{x})$ is a geometry related coefficient, $c(\mathbf{x}) = \frac{1}{2}$ if boundary surface *S* is smooth around $\mathbf{x}$, $q = \partial\phi / \partial n$. Green's function $G(\mathbf{x},\mathbf{y},\omega)$, also referred to the kernel of the integral equation, is given by:

$$G(\mathbf{x},\mathbf{y},\omega) = \frac{1}{4\pi r}e^{ikr}\,, \quad (3.9)$$

where *r* is the distance from the source point $\mathbf{x}$ to the field point $\mathbf{y}$, and its normal derivative is

$$F(\mathbf{x},\mathbf{y},\omega) \equiv \frac{\partial G(\mathbf{x},\mathbf{y},\omega)}{\partial n(\mathbf{y})} = \frac{1}{4\pi r^2}(ikr-1)r_{,j}n_j(\mathbf{y})e^{ikr}\,, \quad (3.10)$$

with $r_{,j} = \partial r / \partial y_j = (y_j - x_j)/r$.

When the boundary element method is implemented to discretize Eq. (3.8), a linear system of equations,

$$\mathbf{F}\boldsymbol{\phi} = \mathbf{G}\mathbf{q}\,, \quad (3.11)$$

is obtained [9]. Here $\boldsymbol{\phi}$ and $\mathbf{q}$ are $N \times 1$ vectors containing the acoustic pressure and velocity of N nodes in the boundary element meshes, respectively. $\mathbf{F}$ and $\mathbf{G}$ are $N \times N$ square coefficient matrices.

3.3 Data Structure of *FastBEM Acoustics*

FastBEM Acoustics [21] is a commercial code for acoustic analysis based on the fast multipole, ACA and conventional BEM. FastBEM Acoustics was used in this thesis work. A sample input file of FastBEM Acoustics is listed in Appendix A.2. The *input.dat* file contains data for the BEM model to be solved using the FastBEM Acoustics program. The file *input.fmm* contains parameters used for the fast multipole and ACA BEM and the iterative linear equation solver. The iterative equation solver is used in both the fast multipole BEM and the ACA BEM. If the boundary conditions need to be updated at each frequency in a frequency response analysis, such as the case of computing the sound fields due to the harmonic responses of a structure, the user needs first change the value of the last integer in the line specifying the frequencies in the *input.dat* file to 1 and then provide the BC values in a new input file named *input.jbc* [21].

4 Coupling of the FEM and BEM

4.1 General Principals

When the boundary condition, usually normal velocity, is known on the surfaces of the vibrating structure, the acoustic field can be calculated easily with the same procedures as the conventional acoustic analysis. The element normal velocities are mapped from the structure vibrating results to the acoustic radiation model as the boundary conditions.

From the structural vibration data at the nodes, the acoustic pressures at the surface can be calculated with the CBIE equation (3.8). This requires a discretization of this continuous equation, see Ref. [9]. This results in the following matrix calculations:

$$\mathbf{F}\boldsymbol{\phi} = \mathbf{G}\mathbf{q}, \tag{4.1}$$

where the matrices **F** and **G** are the influence matrices, which are dependent on the geometry of the structure and comply with the CBIE equation (3.8). The influence matrices are calculated inside the BEM solver packages. The vectors $\boldsymbol{\phi}$ and **q** contain the sound pressures and velocities at each node respectively. So with the velocities from the structural vibration data the sound pressure at each node can be calculated.

4.2 Mapping Procedures

In the FastBEM Acoustic code [21], constant triangular surface elements are used in the discretizations of the boundaries. Thus before the mapping procedure starts, quadrilateral elements using ABAQUS are split into two triangular elements.

A sample ABAQUS input file for structure analysis is listed in Appendix A.1, while a sample FastBEM Acoustics input file for acoustic analysis is listed in Appendix A.2.

The translation programs developed in this thesis contains five scripts, which are:

1) A script to build the models (*a-build-model.py*), which includes importing the object geometries and/or creating native geometries, meshing the geometries, and the boundary conditions. The details are shown in Appendix B.1.

2) A script to get the boundary elements from the solid models (*b-get-boundary.py*), which converts the boundary mesh to three-node triangular elements mesh and flip the normal directions of the boundary elements to assure that the directions face outward to the fluid domain and inward to the structural domain. The details are demonstrated in Appendix B.2.

3) A script to output the field points and field cells of interest (*c-get-field.py*). This script is optional since the field solutions are not necessary for all the BEM analyses. The details are listed in Appendix B.3.

4) A script to post-process the output ODB file (*d-output-odb.py*), which extracts the nodal velocities from the structural dynamic analysis and maps the normal velocities to the boundary elements of the surfaces. The core function of the script is listed in Appendix B.4.

5) A script to generate the parameters for the BEM equation solvers (*e-fmm.py*). This code is listed in Appendix B.5.

The whole structure of the translation program is demonstrated in Figure 4-1.

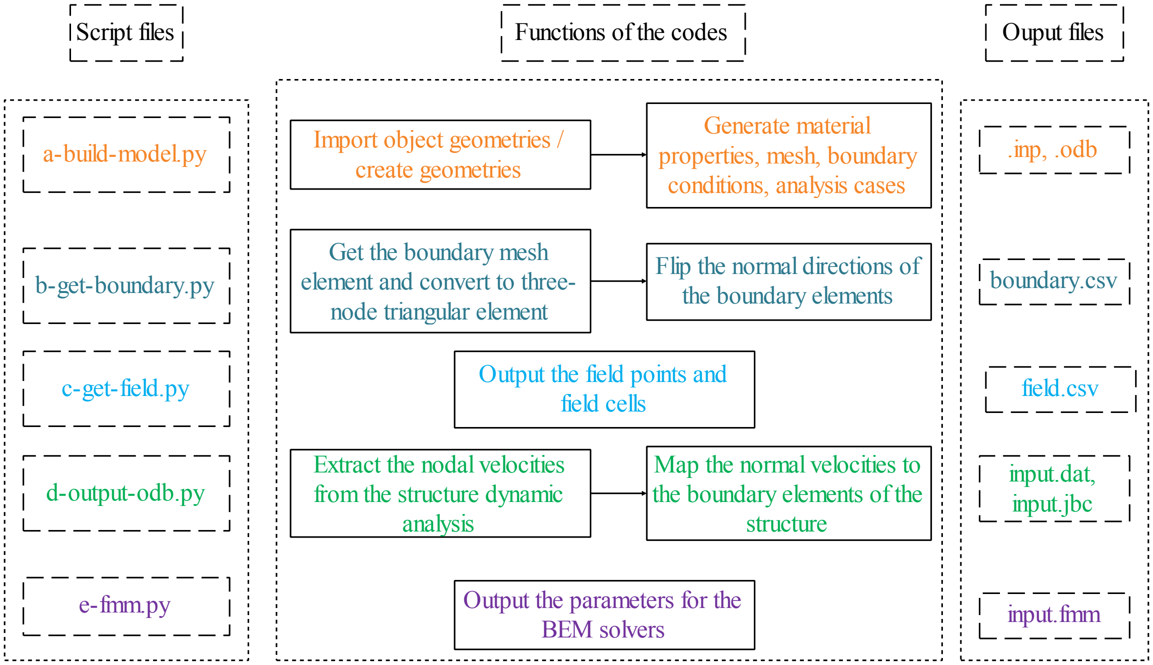

Script files
Functions of the codes
Ouput files
a-build-model.py
b-get-boundary.py
c-get-field.py
d-output-odb.py
e-fmm.py
Import object geometries / create geometries
Generate material properties, mesh, boundary conditions, analysis cases
Get the boundary mesh element and convert to three-node triangular element
Flip the normal directions of the boundary elements
Output the field points and field cells
Extract the nodal velocities from the structure dynamic analysis
Map the normal velocities to the boundary elements of the structure
Output the parameters for the BEM solvers
.inp, .odb
boundary.csv
field.csv
input.dat, input.jbc
input.fmm

Figure 4-1 Whole structure of the translation program

4.3 Velocity Projection at the Normal Direction

A triangular surface with three points A, B and C has the normal direction **n**. The velocity vector of this triangular surface is **v**, which has the arbitrary direction in space. If the velocity vector **v** projects to the surface normal direction **n** (a unit vector), we can get normal velocity $\mathbf{v}_n$ as shown in Figure 4-2.

Velocities from the results of structural vibration analysis are complex numbers with both real and imaginary parts. For example, the velocity at point A has six components (v_{Axr} , v_{Ayr} , v_{Azr} , v_{Axi} , v_{Ayi} , v_{Azi}), in which, the footnote *A* means point A; footnote *x*, *y*, *z* means in the directions of *x*, *y* and *z* coordinates; and footnote *r* and *i* means real part and imaginary part of a complex number, respectively. For instance, v_{Axr} means velocity at node point A, in the *x* direction and with the real part.

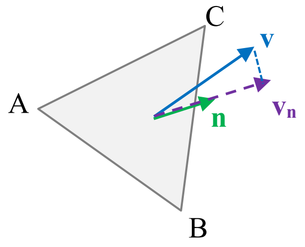


Figure 4-2 Velocity projection at the normal direction of a three point triangular

The velocity vector $\mathbf{v}$ at the centroid of the three points triangular has the real part and imaginary part. The real part of the velocity vector can be expressed as,

$$\begin{aligned}\mathbf{v}_{real} &= (v_{real-x}, v_{real-y}, v_{real-z}) \\ &= (\frac{v_{Axr}+v_{Bxr}+v_{Cxr}}{3}, \frac{v_{Ayr}+v_{Byr}+v_{Cyr}}{3}, \frac{v_{Azr}+v_{Bzr}+v_{Czr}}{3})\end{aligned}, \tag{4.2}$$

while the imaginary part of the velocity vector can be written as,

$$\begin{aligned}\mathbf{v}_{img} &= (v_{img-x}, v_{img-y}, v_{img-z}) \\ &= (\frac{v_{Axi}+v_{Bxi}+v_{Cxi}}{3}, \frac{v_{Ayi}+v_{Byi}+v_{Cyi}}{3}, \frac{v_{Azi}+v_{Bzi}+v_{Czi}}{3})\end{aligned}. \tag{4.3}$$

If the velocity vector $\mathbf{v}$ projects to the unit normal direction **n**, it will get projected velocity vector $\mathbf{v}_n$. The relation of these three vectors is expressed as follows:

$$\mathbf{v}_n = \left(\mathbf{v} \cdot \mathbf{n}\right)\mathbf{n} \tag{4.4}$$

The magnitudes of the real part velocity and imaginary part velocity in the *x*, *y* and *z* directions are calculated in Eq. (4.5) and Eq. (4.6).

$$v_{n-real-magnitude} = \sqrt{v_{real-x}^2 + v_{real-y}^2 + v_{real-z}^2} \tag{4.5}$$

$$v_{n-img-magnitude} = \sqrt{v_{img-x}^2 + v_{img-y}^2 + v_{img-z}^2} \tag{4.6}$$

4.4 Mesh Size of the Acoustic Mesh

With increasing frequency, the mesh that uses for the FEM or BEM simulations must be fine enough to resolve the variation of the acoustic field. In engineering practice, the average size of the elements in a mesh should obey the rule that at least six to ten elements per wavelength should be used for the high frequencies in dynamic analysis. The maximum of the element size can be found by:

$$ElementSize < \frac{\lambda}{6} = \frac{c}{6f_{Max}} \tag{4.7}$$

in which λ is the wave length, c is the sound velocity in the acoustic medium, f is the frequency of interest.

5 Verification Study of the Translation Program

To show the correct implementation and verify the developed translation program, three verification cases are selected to verify the process and the code. There are analytical solutions of the acoustic wave radiation of the pulsating sphere problem with constant velocity boundary conditions. Also for a simply supported plate problem, analytical solutions of natural frequencies are available. By discretizing the geometry model into finite elements, numerical methods in software ABAQUS® and *FastBEM Acoustics*® can be used to calculate the eigenfrequencies and sound radiation power. They are compared with the theoretical values.

5.1 A Pulsating Sphere Problem

5.1.1 Problem Description

As a verification case, let us consider a pulsating sphere in an infinite acoustic medium as shown in Figure 5-1. This problem has an analytical solution that can be applied to comparisons with the numerical solutions.

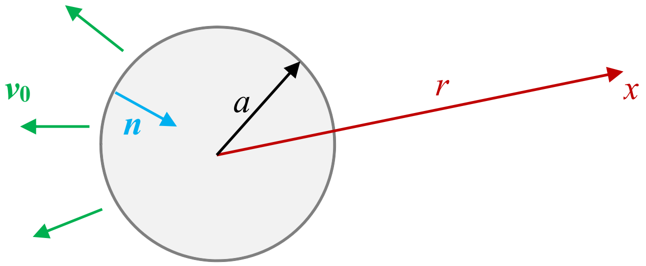


Figure 5-1 A pulsating sphere in an infinite acoustic domain

The dynamic structural analysis is performed first, and then the velocity results from its finite element analysis are used as the input boundary conditions of the boundary element analysis to calculate the acoustic pressure in the field domain. In addition, the analytical solution with approximate velocity boundary conditions is obtained.

The sphere has a radius *a* and there is a field point *x* with distance *r* from the center of the sphere. The normal directions of the boundary are inward to the structure.

5.1.2 Analytical Formulations

This is a radiation problem, and the boundary-value problem is given by

$$\nabla^2\phi + k^2\phi = 0\,, \tag{5.1}$$

$$\frac{\partial\phi}{\partial n} = -i\omega\rho v_0\,. \tag{5.2}$$

With this formula, the solution to the pulsating-sphere problem is [9]

$$\phi(r,\omega) = \frac{\rho c v_0 (ika)}{ika-1}\frac{a}{r}e^{ik(r-a)}\,, \tag{5.3}$$

in which, ρ is the acoustic density, *c* is the sound velocity in the acoustic medium, *a* is the radius of the sphere, *r* is the distance from the center of sphere to the field point *x*, *k* is the wave number, v_0 is the velocity boundary condition of the sphere, and ϕ is the pressure in the exterior domain.

5.1.3 Finite Element Model

The pulsating sphere geometry is meshed by four-node linear solid tetrahedron element as shown in Figure 5-2. The material of the pulsating sphere is similar to polystyrene with a density of 450 kg/m^3, elastic Young's modules of 3.4E+09 Pa, and Poisson's ratio of 0.3.

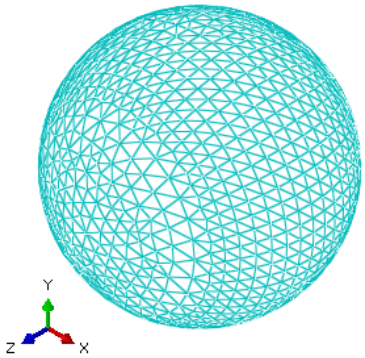


Figure 5-2 Finite element model of a pulsating sphere

The boundary condition of this model is determined by adding the excited pressure with the amplitude of -100,000 Pa on the outside boundary surface as shown in Figure 5-3. The structural dynamic harmonic analysis is conducted with frequency range from 1 Hz to 550 Hz subdivided with 10 frequencies.

There are a total of 64,657 four-node solid tetrahedron elements and 12,535 nodes in this finite element model.

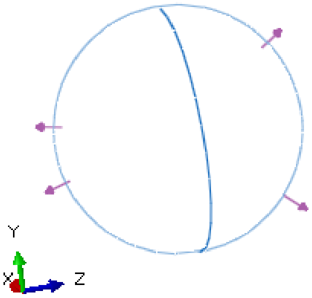


Figure 5-3 Boundary condition of the finite element model of a pulsating sphere

5.1.4 Boundary Element Model

In the boundary element model, the material parameters of the acoustic medium is air with a density of 1.225 kg/m^3, a sound velocity of 340 m/s, a reference pressure of 2E-05 Pa, and a reference intensity of 1E-12 W/m^2.

Three different solvers, including the *FastBEM Acoustics* fast multipole (FMM) solver, the *FastBEM Acoustics* adaptive cross approximation (ACA) solver and the *FastBEM Acoustics* conventional BEM with fast direct solver, are used in the BEM analysis.

There are a total of 7,120 three-node boundary triangular elements, 3,562 boundary nodes, 441 field points, and 400 four-node field cells in this BEM model.

5.1.5 Analytical Solution with Approximate Boundary Conditions

One can determine the analytical solution of Eq. (5.3) with the given parameters. The radius of the sphere *a* is 1 m, the distance from the center of the sphere to the field point *r* is 5 m, the acoustic density ρ is 1.225 kg/m^3, the sound velocity in the acoustic medium *c* is 340 m/s, and the frequency range of interest *f* is from 1 Hz to 550 Hz.

In regards to the problem of a dynamic sphere, the application of the translation program will get the projected and normalized boundary element velocities, which could be mapped into the boundary element analysis.

At the same time, to obtain the approximate solutions at each frequency in the frequency range, the average velocity from the projected and normalized velocity of all the boundary elements must be calculated. The velocity values have both the real part and the imaginary part.

The averaged velocities with real and imaginary parts, standard deviation and standard error are extracted by the translation program and summarized in Table 5-1.

Table 5-1 Averaged velocity from the projected and normalized velocity of the boundary elements

Frequency (Hz)	ka	Averaged velocity from the projected and normalized velocity of the boundary elements (m/s)		Standard deviation *s*	Standard error *SE*
		v_{real}	v_{img}		
1.00	0.02	0.00E+00	7.39E-05	1.21E-08	1.43E-10
62.00	1.15	0.00E+00	4.59E-03	7.49E-07	8.87E-09
123.00	2.27	0.00E+00	9.14E-03	1.49E-06	1.76E-08
184.00	3.40	0.00E+00	1.38E-02	2.24E-06	2.65E-08
245.00	4.53	0.00E+00	1.86E-02	3.01E-06	3.56E-08
306.00	5.65	0.00E+00	2.35E-02	3.83E-06	4.54E-08
367.00	6.78	0.00E+00	2.88E-02	4.75E-06	5.63E-08
428.00	7.91	0.00E+00	3.43E-02	5.85E-06	6.94E-08
489.00	9.04	0.00E+00	4.03E-02	7.27E-06	8.61E-08
550.00	10.16	0.00E+00	4.68E-02	9.25E-06	1.10E-07

Then, we can calculate the approximate analytical solutions of this pulsating sphere problem with boundary conditions from structural vibration by analytical formulation Eq. (5.3). The approximate analytical solutions of pressure at field point *x* with frequency range from 1.00 Hz to 550.00 Hz are listed in Table 5-2.

5.1.6 Numerical Solutions

The numerical solutions are calculated from the structural finite element analysis, coupling program and acoustic boundary element analysis. The error percentage can be obtained from the numerical solutions and analytical solutions with Eq. (5.4).

$$Error = \frac{\left|NumericalSolution - AnalticalSolution\right|}{\left|AnalyticalSolution\right|} \tag{5.4}$$

The analytical solution, numerical solutions with three different solvers, and errors are all summarized in Table 5-2.

Table 5-2 Comparisons of analytical solution with approximate boundary conditions and numerical solutions of a pulsating sphere

Frequency (Hz)	ka	|Pressure| at field point x (Pa)						
		FMM Solver	Error	ACA Solver	Error	Conventional BEM Solver	Error	Analytical solution with approximate boundary conditions
1.00	0.02	0.0001136	0.098%	0.0001136	0.098%	0.0001136	0.098%	0.0001137
62.00	1.15	0.2877	0.05%	0.2877	0.06%	0.2877	0.06%	0.2878
123.00	2.27	0.6964	0.10%	0.6963	0.11%	0.6964	0.10%	0.6971
184.00	3.40	1.1046	0.26%	1.1068	0.46%	1.1060	0.39%	1.1017
245.00	4.53	1.5128	0.18%	1.5108	0.05%	1.5107	0.04%	1.5101
306.00	5.65	1.9359	0.28%	1.9366	0.32%	1.9353	0.25%	1.9305
367.00	6.78	2.4053	1.50%	2.3904	0.87%	2.3761	0.27%	2.3697
428.00	7.91	2.8008	1.20%	2.8072	0.97%	2.8348	0.00%	2.8348
489.00	9.04	3.4449	3.33%	3.4424	3.26%	3.3523	0.55%	3.3339
550.00	10.16	3.8545	0.58%	3.6582	5.64%	3.8775	0.02%	3.8768

The pressure amplitudes at the field point x are plotted in Figure 5-4, which shows that all three FastBEM Acoustics solvers can predict the field pressure very well compared with the analytical solution with approximate boundary conditions.

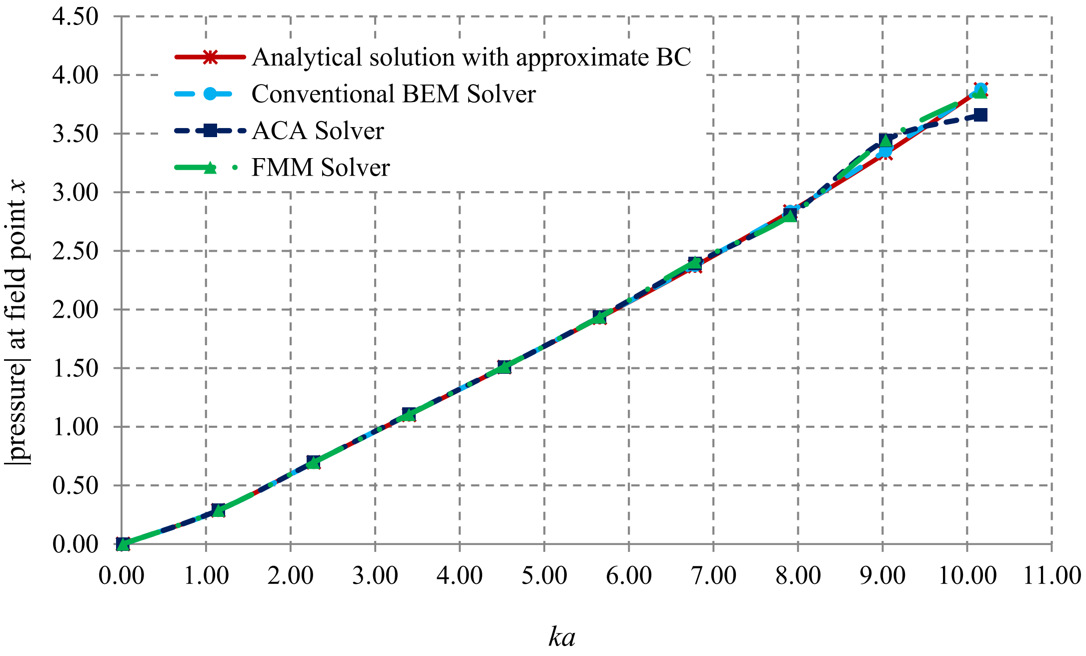


Figure 5-4 Plot of pressure at field point *x* from different numerical solvers and analytical solution with approximate boundary conditions

5.2 A Simply Supported Plate Problem

5.2.1 Problem Description

There is a thin rectangular plate with the vertices *A*, *B*, *C* and *D* in the space, where edges *AB*, *BC*, *CD* and *DA* are all fixed in three translational directions. The length of the plate is *a*, the width of the plate is *b*, and the thickness of the plate is *t*. The perpendicular force *F* is added to the central point of the plate. The geometry of the plate is shown in Figure 5-5.

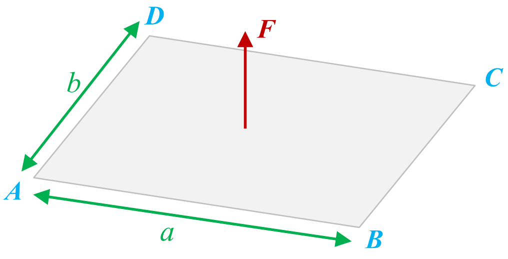


Figure 5-5 A thin simply supported plate with perpendicular force F

In this case, the length of the plate *a* is 0.414 m, the width of the plate *b* is 0.314 m, and the thickness *t* is 0.002 m. The perpendicular harmonic point force *F* has an amplitude of 1 N at the center point.

5.2.2 Finite Element Model

The thin plate geometry is meshed as shown in Figure 5-6 with three different types of elements, which are eight node linear solid brick element, eight node continuum shell brick element, and four node shell element.

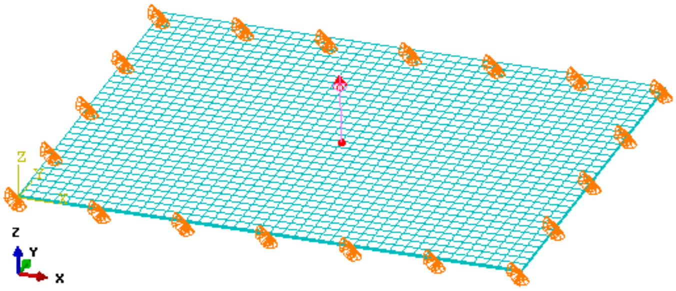


Figure 5-6 Finite element model of a thin simple supported plate

The material density of this thin plate is 2,700.00 kg / m^3, elastic Young's module is 7.10E+10 Pa and Poisson's ratio is 0.33.

The boundary condition of the finite element model includes fixing four edges of the plate in three translational directions. The loading condition is determined by adding harmonic force *F* with the amplitude of 1 N to the central point of the plate. The analysis type is determined by first extracting the first ten natural frequencies of the model, and then calculating the dynamic structural analysis with frequency range from 1 Hz to 1,000 Hz subdivided with a total of 50 frequencies.

There are a total of 2,688 four-node solid/continuum shell elements and 4,257 nodes in the finite element model. Ten eigenvalues are extracted from the natural frequency analysis. The lowest frequency is 1 Hz and the highest frequency is 1,000 Hz in the harmonic analysis with a total of 50 points in the frequency range.

5.2.3 Boundary Element Model

The material property of the acoustic medium of this case is air. There are a total of 5,968 three-node boundary triangular elements, and 2,986 boundary nodes in the boundary element model.

5.2.4 Natural Frequency Solutions

The natural frequencies and eigenmodes of this simply supported plate shown in Figure 5-5 have the analytical solution. From theory [1] the eigenfrequencies can be expressed in Eq. (5.5).

$$\omega_{mn} = \pi^2 \left(\left(\frac{m}{a} \right)^2 + \left(\frac{n}{b} \right)^2 \right) \sqrt{\frac{D}{\rho t}}, m, n = 1, 2, ... \qquad (5.5)$$

where,

$$D = \frac{Et^3}{12\left(1 - v^2\right)}, \qquad (5.6)$$

is the bending stiffness of the plate, *E* is elastic Young's modulus, *t* is the thickness of the plate, *v* is Poisson's ratio, *ρ* is the density, and *a* and *b* are geometry length and width. By substituting the geometry parameters and material properties to Eq. (5.6) and (5.5), we can calculate the analytical solution of natural frequencies of this thin simple supported plate.

Natural frequencies can also be obtained numerically by finite element method. Eight-node solid brick element, four-node shell element and eight-node continuum shell brick element are used in the finite element models. The error estimate Eq. (5.4) can be used to determine the error percentage of each numerical scheme. The whole set of the results including the analytical solution and numerical solutions are listed in Table 5-3.

Table 5-3 Comparisons of natural frequencies from analytical solution and different numerical solutions

Mode No.	Natural Frequency (Hz)						
	Eight node solid brick element		Four node shell element		Eight node continuum shell brick element		Analytical solution
1	78.73	0.03%	78.70	0.01%	78.66	0.06%	78.71
2	165.12	0.11%	165.12	0.11%	164.93	0.01%	164.94
3	229.01	0.17%	229.39	0.34%	229.08	0.20%	228.61
4	309.41	0.24%	309.98	0.43%	309.39	0.24%	308.66
5	315.61	0.24%	315.57	0.23%	314.85	0.00%	314.84
6	460.27	0.37%	460.12	0.34%	458.56	0.00%	458.56
7	480.30	0.39%	482.85	0.92%	481.58	0.65%	478.45
8	511.99	0.42%	514.41	0.89%	512.92	0.60%	509.87
9	567.30	0.46%	568.79	0.73%	566.66	0.35%	564.68
10	663.41	0.55%	664.2	0.67%	661.09	0.20%	659.77

By plotting the first ten eigenvalues of the thin simple supported plate model as shown in Figure 5-7, one can determine that the numerical solutions from all three element types are very close to the analytical solution.

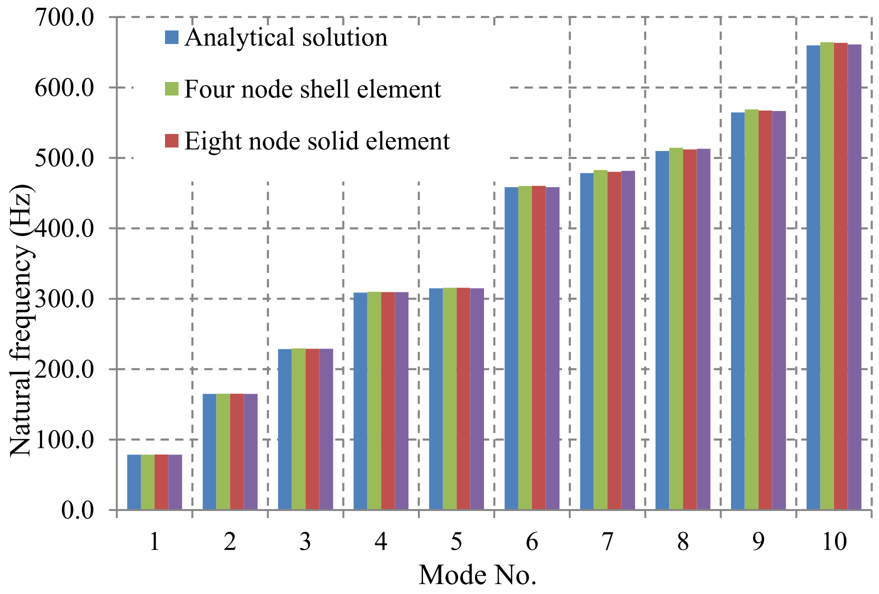
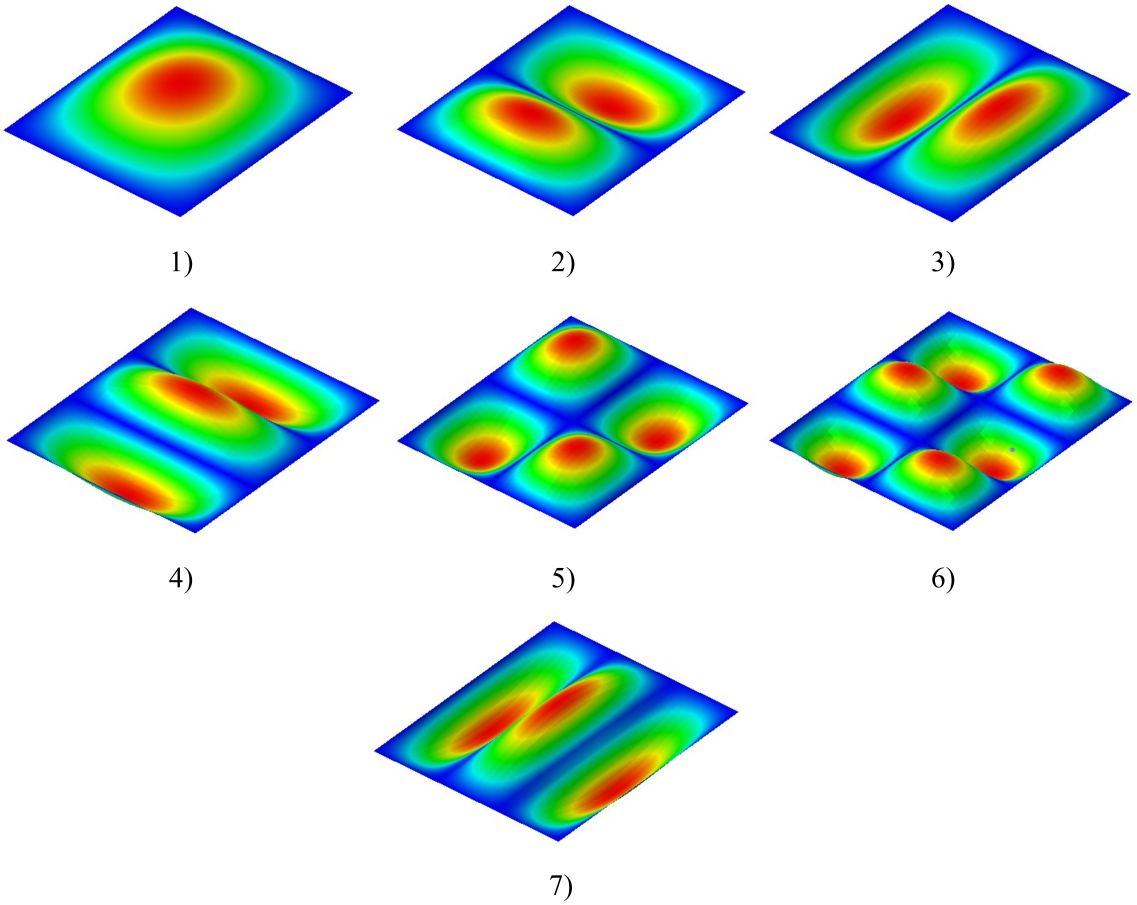


Figure 5-7 Plot of comparisons of first ten modes of natural frequency

First, seven mode shapes are plotted in Figure 5-8, from which we can see that mode 1, mode 4 and mode 7 are symmetric modes, while mode 2, mode 3, mode 5 and mode 6 are asymmetric modes.



1) 2) 3)

4) 5) 6)

7)

Figure 5-8 Mode shapes of a simply supported plate: 1) mode shape at mode 1, 2) mode shape at mode 2, 3) mode shape at mode 3, 4) mode shape at mode 4, 5) mode shape at mode 5, 6) mode shape at mode 6, 7) mode shape at mode 7

5.2.5 Acoustic Wave Analysis

For the flat plate problem, the sound pressure can be obtained from the CBIE (3.8) with the boundary normal velocities. The sound power radiation of the simply supported plate model can be calculated from the sound pressure and normal velocities. Different numerical methods are used for the calculations, including FastBEM Acoustic solvers (FMM solver, ACA solver and

Conventional BEM solver) and LMS Virtual.Lab solver to solve the acoustic wave problem. The frequency-sound power plots are shown in Figure 5-9.

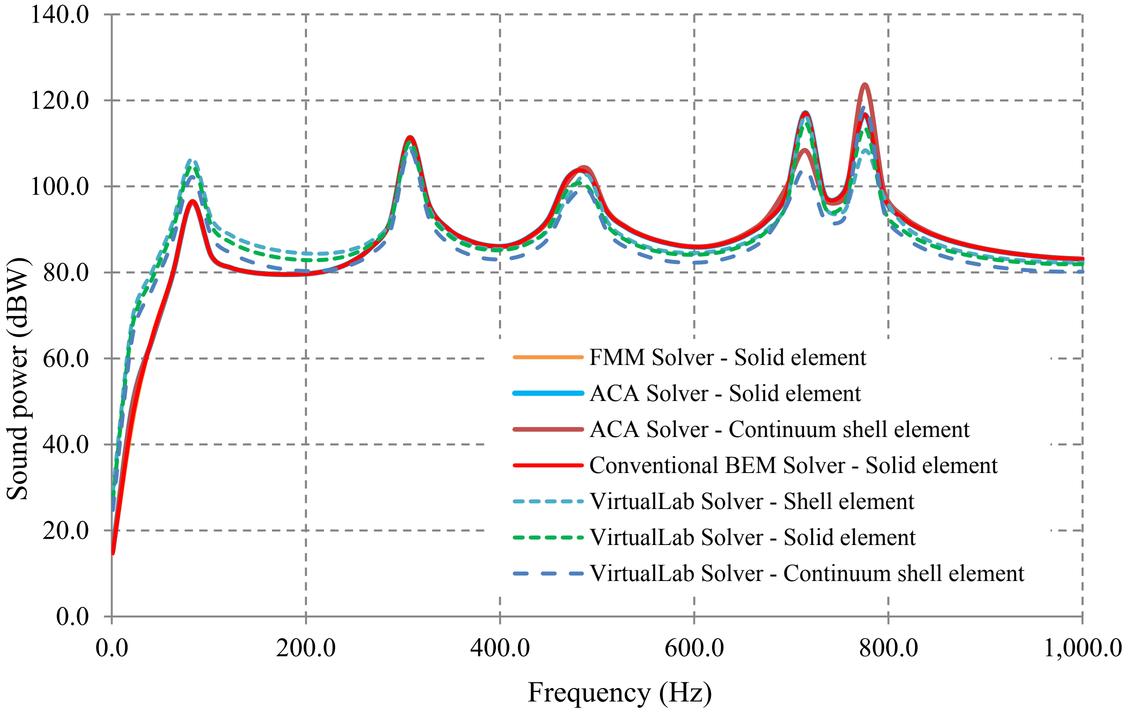


Figure 5-9 Plot of frequency - sound power of a simply supported plate

5.2.6 Solution Discussions

As is illustrated in Figure 5-9, the radiated sound power is characterized by a sequence of peaks corresponding to the eigenfrequencies of the plate. It is important to note that the peaks do not correspond with all structural eigenfrequencies. From the mode shape plots Figure 5-8, we can see that the first three peaks correspond to the mode 1, mode 4 and mode 7 respectively. These three modes are symmetric modes and radiate sound efficiently, while the asymmetric modes do not [24]. For example, the second mode shape is an asymmetric mode, which causes high pressure at one half of the plate and a low pressure at the other half at the same time. In this way the variation in pressure at the surface caused by the two parts of the plate cancel each contribution to the total radiated power.

5.3 A Corner-fixed Plate Problem

5.3.1 Problem Description

A thin rectangular plate has four vertices *A*, *B*, *C* and *D*, which are fixed in six translational and rotational directions in the space. The setup of the plate is shown in Figure 5-10.

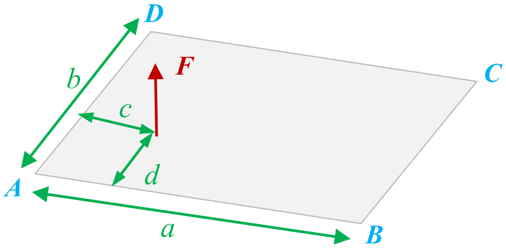


Figure 5-10 A thin corner-fixed rectangular plate with perpendicular force F

In this case, the length of the plate *a* is 0.1 m, the width of the plate *b* is 0.05 m, and the thickness *t* is 0.005 m. The distance from the load point to the edge *d* is 0.01667 m and the distance from the load point to the edge *c* is 0.03 m. The harmonic perpendicular force *F* with the amplitude 1 N is added to the plate.

5.3.2 Finite Element Model

The appropriate mesh size can be obtained to capture the dynamic effects of high frequencies of interest by using Eq. (4.7). In this case the highest frequency of interest is 4,000 Hz and the sound velocity at acoustic medium is 340 m/s. The maximum element size can then be calculated. In this example, it is 0.0085 m. The average mesh size of this finite element model is chosen to be 0.0065 m. The geometry is meshed with three different element types, which are eight-node linear solid brick element, four-node shell element and six-node continuum shell wedge element. The finite element model is shown in Figure 5-11.

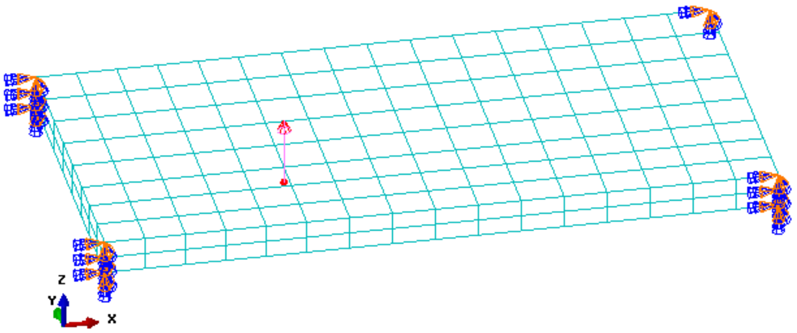


Figure 5-11 Finite element model of a corner-fixed plate

The material density of the plate is 2,700.00 kg/m^3, elastic Young's Modules is 7.0E+10 Pa, and Poisson's ratio is 0.33.

The boundary conditions of the finite element model are determined by fixing four corners of the plate in all six translational and rotational directions. The loading condition is determined by adding harmonic force *F* with the amplitude of 1 N to the point which has the distance of 0.01667 m to the length edge of the plate and the distance of 0.03 m to the width edge of the plate. The analysis first extracts the natural frequencies of the model between 0 Hz to 6,000 Hz, and then develops the dynamic structural analysis with frequency range from 8 Hz to 4,000 Hz subdivided by 8 Hz per step. Table 5-4 shows the summary of the finite element model.

There are a total of 256 four-node solid brick elements and 459 nodes in this finite element model.

Table 5-4 Summary of the finite element model of a corner-fixed plate

Parameter		Value	Unit
Element types		1) Eight-node solid brick element 2) Four-node shell element 3) Six-node continuum shell wedge element	
Natural frequency	Lower frequency	0	Hz
	Upper frequency	6,000	Hz
Dynamic structural analysis	Lower frequency	8	Hz
	Upper frequency	4,000	Hz
	Frequency per step	8	Hz
	Total points	500	

5.3.3 Boundary Element Model

The material property of the acoustic medium of this case is air. There are a total of 704 three-node boundary triangular elements and 354 boundary nodes in the boundary element model.

5.3.4 Natural Frequency Solutions

Natural frequencies of the corner-fixed plate model are calculated numerically by using the finite element method with different element types, which are eight node solid brick element, four node shell element and six node continuum shell wedge element.

All natural frequency modes between 0 Hz and 6,000 Hz of the corner-fixed plate model with different three element types are extracted, which are listed in Table 5-5 and plotted in Figure 5-12.

Table 5-5 Modes between 0 Hz and 6000 Hz of a corner-fixed plate

Mode No.	Natural Frequency (Hz)		
	Eight-node brick element	Four-node shell element	Six-node continuum shell element
1	1,649.6	1,663.3	1,639.3
2	3,646.4	3,684.8	3,655.6
3	4,941.7	5,037.8	5,025.3

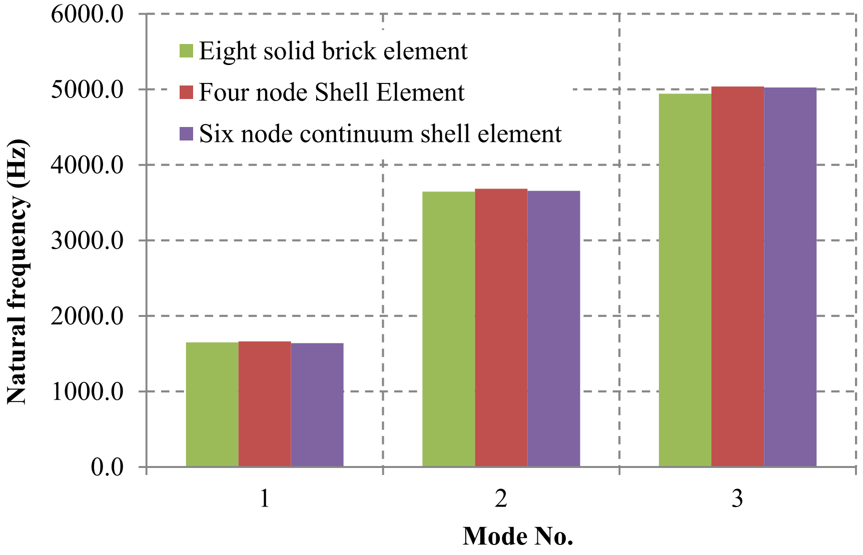


Figure 5-12 Plot the first three natural frequencies comparisons of a corner-fixed plate

First three mode shapes of this corner-fixed plate model are plotted in Figure 5-13 for later discussions.

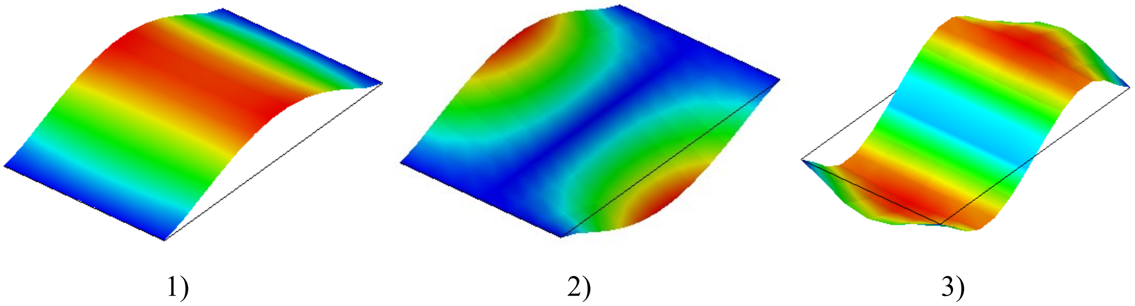


Figure 5-13 First three mode shapes of a corner-fixed plate: 1) mode shape of Mode 1, 2) mode shape of Mode 2, 3) mode shape of Mode 3

5.3.5 Acoustic Wave Analysis

The sound power radiation of the corner-fixed plate model is calculated by numerical methods. The acoustic wave model is solved by FastBEM Acoustics and LMS Virtual.Lab. Different FastBEM Acoustics solvers (including fast multipole solver, adaptive cross approximation solver and conventional BEM with fast direct solver) are chosen for the comparisons. The obtained frequency-sound power plots are shown in Figure 5-14.

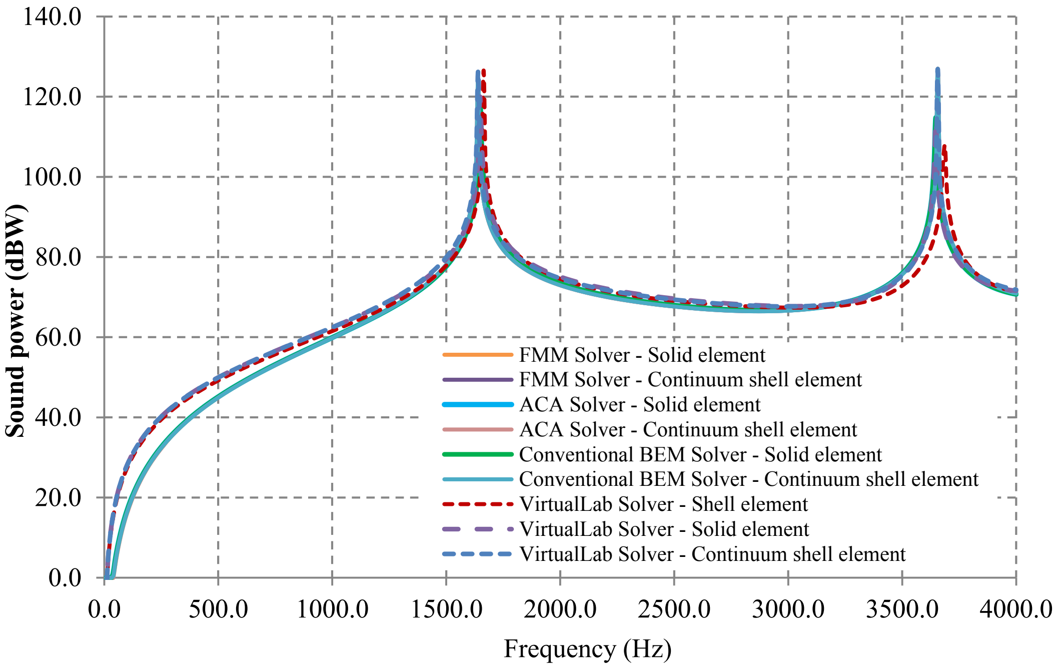


Figure 5-14 Plot of frequency-sound power of a corner-fixed plate

5.3.6 Solution Discussions

As can be seen in Figure 5-14 the radiated sound power is characterized by two peaks corresponding to the eigenfrequencies of the corner-fixed plate in the frequency range from 0 Hz to 4,000.0 Hz. From the first two mode shapes plotted in Figure 5-13, we can conclude that this corner-fixed plate radiates sound power efficiently at both of the two eigenmodes.

6 Application to Wind Turbine Acoustic Analysis

A wind turbine acoustic problem is chosen to further test the coupling method developed in this thesis work.

The U.S. Department of Energy (DOE) estimates that more than 900,000 MW^2 (close to the total current installed U.S. electrical capacity) of potential wind energy exists off the coasts of the United States, often near major population centers, where energy costs are high and land-based wind development opportunities are limited [25].

Wind turbines are used to harness the kinetic energy of the moving air and convert it to electricity. Offshore winds are less turbulent (because the ocean is flat relative to onshore topography), and they tend to flow at higher speeds than onshore winds, thus allowing turbines to produce more electricity. Because the potential energy produced from the wind is directly proportional to the cube of the wind speed, increased wind speeds of only a few miles per hour can produce a significantly larger amount of electricity. For instance, a turbine at a site with an average wind speed of 16 mph would produce about 50% more electricity than at a site with the same turbine and an average wind speed of 14 mph [25].

6.1 Literature Review of Wind Turbine Problem

6.1.1 Wind-Structure Interaction

By nature, air flow is generally unsteady causing induced forces on structures to fluctuate with time. In some cases, however, winds would be steady which causes across-wind harmonic vibrations due to the vortex shedding phenomenon. Along-wind response is also of concern due to wind gusts on the structure and possible vibration amplification or even resonance. Extreme wind conditions may cause significantly high stress and possible yielding in the structure. Another complication is the general transient nature of wind at low velocities, which could cause strain cycling and eventually high cycle fatigue.

6.1.2 Behavior under Steady Wind

Slender structures obstructing steady air flow experience aerodynamic forces in both directions along-wind and across-wind (or drag and lift forces, respectively) [26]. If the structure has an asymmetric cross-section, then the dominant force would be the drag force. When the vortex shedding phenomenon is exhibited, the lift force increases and becomes significant. To illustrate this, the Bernoulli's equation can be used to determine the pressure on a structure due to steady wind as shown below:

$$p = \frac{1}{2}\rho v^2 \tag{5.7}$$

where p is the pressure, v is the speed of the wind and ρ is the air density.

6.1.3 Types of Loading

The loading regimes to which a wind turbine is subject are extremely complex requiring special attention in their design, operation and maintenance. An understanding of the loading on wind turbines and their origins, as well the turbines response to them is crucial to avoid their catastrophic failure.

The types of loads a wind turbine is subjected to during service can be classified as the static loading, the cyclic loading and the stochastic loading.

6.1.4 Examples of Design

Two examples of design for wind turbines are listed in Table 6-1, which summarizes the final design for two of the tower locations at both of the Savoonga and Mekoryuk villages [26]. The system natural frequencies of the wind turbines are 1.128 Hz and 1.148 Hz in Savoonga and Mekoryuk respectively. These data can provide a reference for the FEA study of the harmonic response of a wind turbine model.

Table 6-1 Final design summary of the wind turbine for Savoonga and Mekoryuk villages

Village name	Savoonga	Mekoryuk	Unit
Tower height	29	29	m
Number of piles	6	6	
Point of fixity	18-11 ft below soil surface	30 ft below soil surface	
System natural frequency	1.128	1.148	Hz
Recommended maximum rpm	57	58	rpm

6.1.5 Material of the Blades

The blades of a wind turbine are normally made of composite materials containing more than one bonded material, each with different structural properties. One of the materials, which is called the reinforcing phase [27] is embedded in the other material of the matrix phase. If the composite is designed and fabricated correctly, it combines the strength of the reinforcement with the toughness of the matrix to achieve a combination of desirable properties not available in any single conventional material. The main advantage of composite materials is the potential for a high ratio of stiffness to weight. Composites used for typical engineering applications are advanced fiber or laminated composites, such as fiberglass, glass epoxy, graphite epoxy, and boron epoxy. Composites are somewhat more difficult to model than an isotropic material such as iron or steel. The special care must be taken in defining the properties and orientations of the various layers since each layer may have different orthotropic material properties.

The majority of wind turbine blades are made of fiberglass and reinforced with polyester or epoxy resin. Construction using wood–epoxy or other materials also can be found. Small turbine blades are made of steel or aluminum, but they are heavier.

6.2 Problem Description

A three-blade wind turbine model with the geometry shown in Figure 6-1 has been studied. The detailed geometry is created by Parasolid as described in Ref. [28]. The height of vertical supporting rod is 17.68 m, the length of each blade is 10.31 m, with a maximum width of each blade is 1.27 m, and the length of the horizontal shaft is 3.55 m.

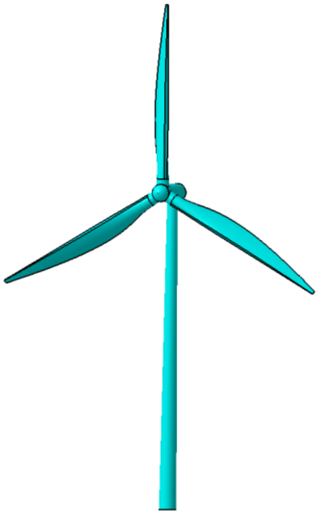


Figure 6-1 Geometry of a three-blade wind turbine

6.3 Finite Element Model

Considering the complexity of the wind turbine geometry, four node solid tetrahedron elements are chosen to mesh the whole model. The mesh topology of part of the geometry is shown in Figure 6-2.

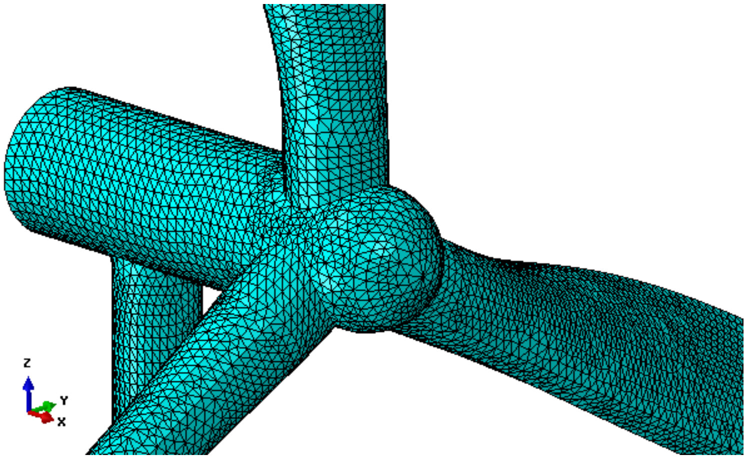


Figure 6-2 Mesh of the three-blade wind turbine

From the study of the material properties in Section 6.1.5, we are assuming that the material of the wind turbine is isotropic and aluminum like. The material density is 2,700.0 kg/m^3, elastic Young's modulus is 7.0E+10 Pa and Poisson's ratio is 0.33.

During the structural dynamic analyses, a first step of static analysis is added before the natural frequency analysis. In this static analysis, a rotational body force due to the rotation of the three blades at a speed of 58 rpm is applied to the blades. This rotational speed equals to 6.07 radians per second according to Eq. (6.1).

$$1rpm = \frac{2\pi}{60} radians/second \tag{6.1}$$

The second step is the natural frequency analysis, which extracts the first ten natural frequencies of the wind turbine models. The third step is the harmonic response analysis. In this analysis step, the amplitude of the applied dynamic pressure load is assumed to be 30.0 Pa.

There are a total of 98,470 four-node solid tetrahedron elements and 21,399 nodes in the finite element model with a mesh size of 0.15 m. First ten eigenvalues are calculated by using the FEM.

The lower bound of frequency in interest is 0.1 Hz, while the upper bound of frequency in interest is 10 Hz. Ten frequency values of interest are chosen within the frequency range.

6.4 Boundary Element Model

In *FastBEM Acoustics*, there are two parameters *nruleb* and *nruef* to determine the integration quadrature rule or the number of Gaussian points in the numerical integration for the boundary integrals and field points respectively [21]. Large numbers of Gaussian points can improve the accuracy, but can also reduce the efficiency of the code. For most problems, selection of *nruleb* = 3, which corresponds to 6 Gaussian points, has been found to be sufficient for the accuracy of the BEM results.

In LMS Virtual.Lab, there are several parameters of quadrature options to set the accuracy of the numerical integration scheme by specifying the order of the Gauss quadrature [29]. The quadrature, i.e. the number of Gauss points used in one direction for the numerical evaluation of integrals, can be selected between 1 and 5 for three different regions: the near field, the middle field and the far field.

Different quadrature options are studied on this wind turbine model. There are a total of 18,740 three-node boundary triangular elements, 9,372 boundary nodes, 2,916 field points and 2,809 field cells (four-node quadrilateral elements) in the boundary element half-space model. The field surface is the ground support of the wind turbine as it is in the physical situation.

6.5 Natural Frequency Solutions

The wind turbine model is meshed by four-node solid tetrahedron elements with average element size of 0.08 m, 0.15 m, 0.25 m and 0.5 m to have a convergence study. The first ten modes of natural frequencies are given in Table 6-2.

Table 6-2 First ten modes of natural frequency of a wind turbine

Mode No.	Natural frequency of wind turbine model meshed with four node solid tetrahedron element (Hz)			
	element size 0.08 m	element size 0.15 m	element size 0.25 m	element size 0.5 m
1	1.18	1.20	1.22	1.34
2	2.80	2.82	2.86	3.15
3	5.31	5.42	5.56	6.11
4	7.79	7.81	7.82	7.95
5	10.06	10.28	10.59	11.87
6	10.68	10.90	11.20	12.45
7	11.83	11.95	12.12	13.25
8	12.45	12.67	13.03	14.61
9	15.92	16.34	16.99	19.16
10	19.92	20.61	21.60	24.59

The natural frequencies for different element sizes of four-node solid tetrahedron elements are plotted in Figure 6-3. We can see that the natural frequencies of different element size models converge at very close values at the first four modes. Thus, the element size is chosen as 0.15 m for the subsequent dynamic structural analysis and acoustic wave analysis.

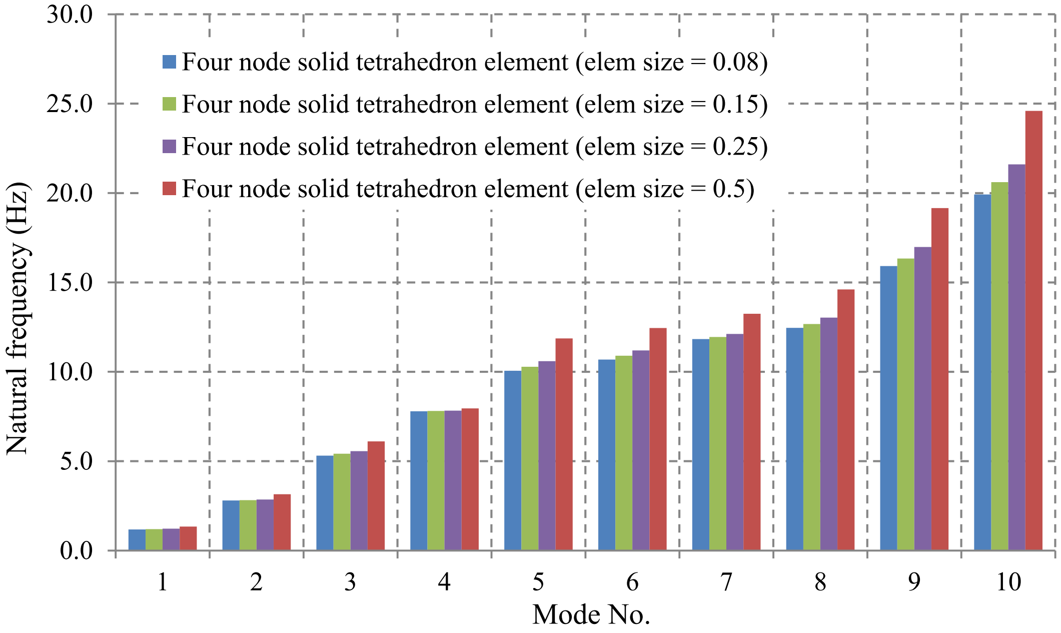


Figure 6-3 Comparisons of natural frequencies of wind turbine with different element sizes

The first three mode shapes are plotted in Figure 6-4, in which the gray shape is the original undeformed model, and the colored shape shows the deformed modes. The first mode is the front-back rigid mode, the second mode is the three blade rotational mode, and the third mode is the three blade bend mode.

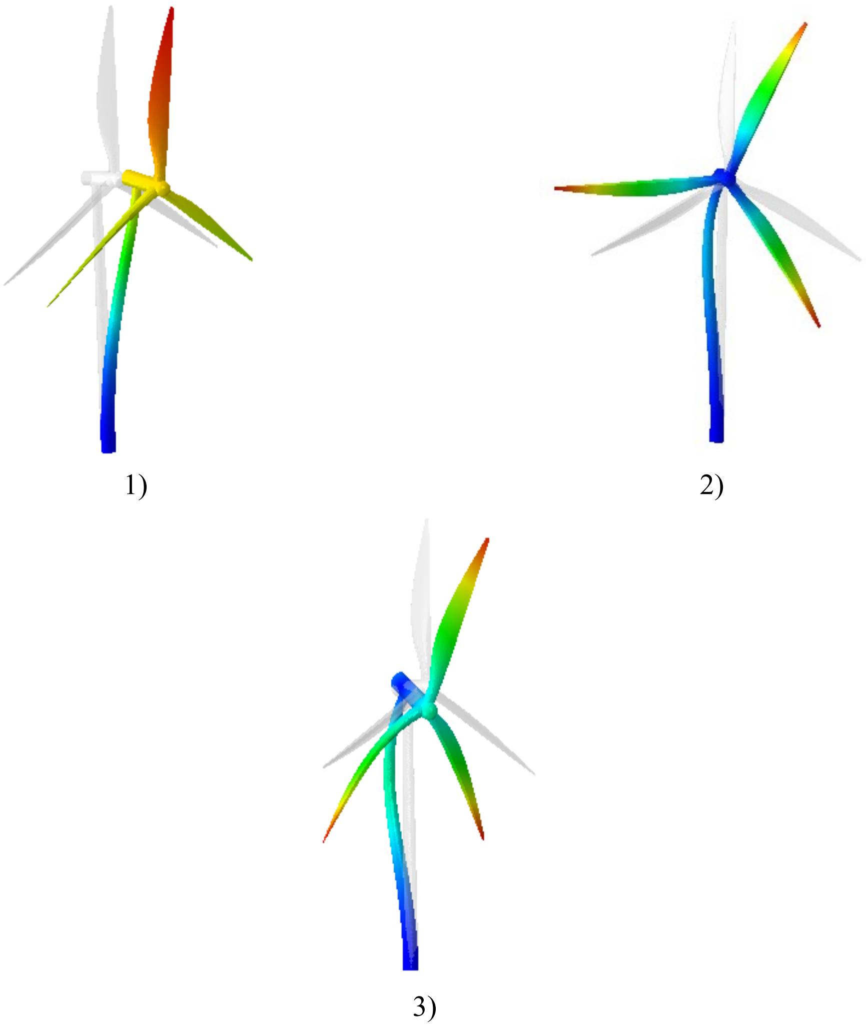


Figure 6-4 First three mode shapes of the structural natural frequency analysis of a wind turbine, 1) front-back rigid mode; 2) three-blade rotational mode; 3) three-blade bend mode

In order to perform the dynamic harmonic analysis, the harmonic pressure with amplitude of 30.0 Pa is added to the three blades' surfaces of the wind turbine model as the dynamic perturbation. The one can determine the velocity result distributions at each of the frequency. Figure 6-5 shows the velocity distributions of the wind turbine model at frequency 6.334 Hz.

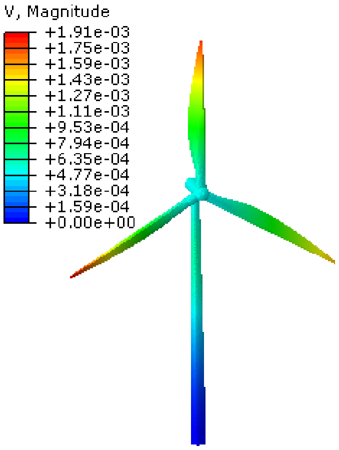


Figure 6-5 Velocity distributions at frequency 6.334 Hz of the wind turbine

6.6 Acoustic Wave Solutions

For this large scale application model, two different solvers of *FastBEM Acoustics* (including fast multipole solver and adaptive cross approximation solver) and LMS Virtual.Lab solver are chosen to solve this problem. In *FastBEM Acoustics*, the *nruleb* is chosen as 3, while the *nrulef* is chosen as 1 and 3. In LMS Virtual.Lab, the quadrature value in the near field is given as 2, and in the far field is given as 1. The frequency - sound power response curves are plotted in Figure 6-6.

The half-space/symmetry modeling technique is used in the BEM analysis. The symmetry infinite and rigid half-space plane is *x*-*y* plane, on which the wind turbine is positioned.

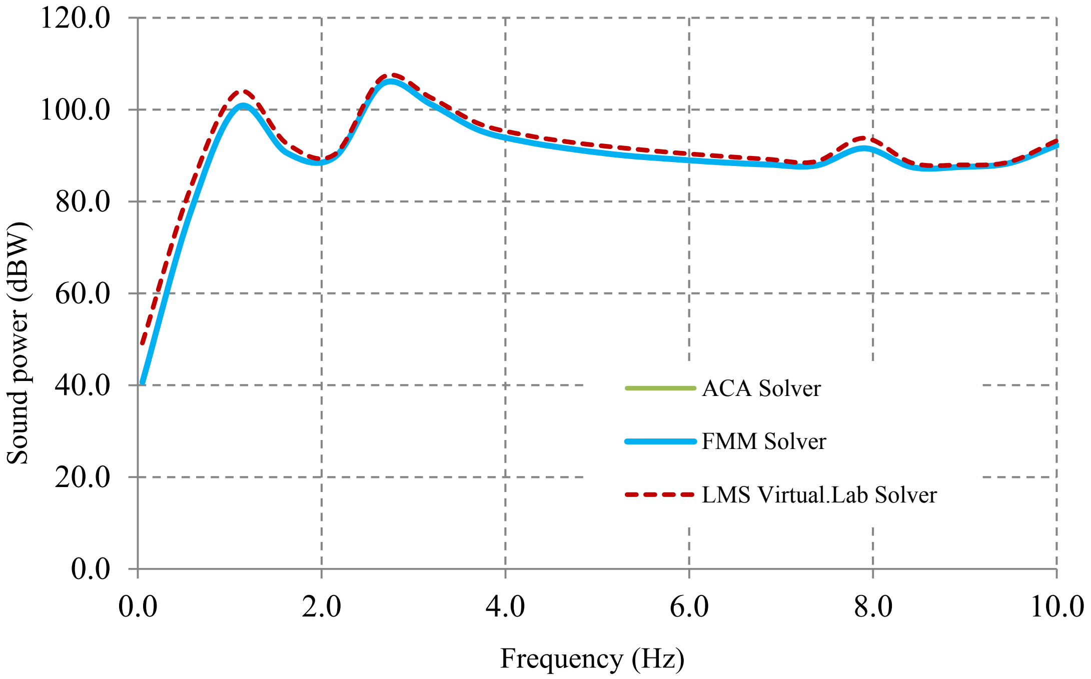


Figure 6-6 Frequency - sound power plot of a wind turbine by FastBEM Acoustics (FMM and ACA) and LMS Virtual.Lab solvers

We can also plot the SPL (dB) on the field surface of the wind turbine model at each frequency of interest. Figure 6-7 shows the SPL on the field surface at the frequency of 2.67 Hz.

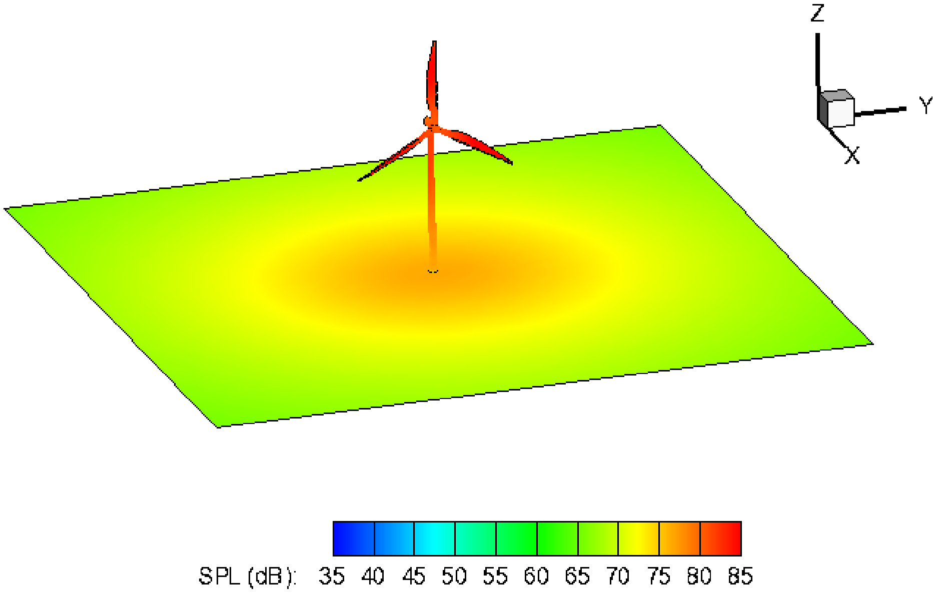


Figure 6-7 Plot of the SPL (dB) on the field surface of the wind turbine model at frequency of 2.67 Hz

The CPU time comparisons of different solver types are shown in Figure 6-8. The CPU time from *FastBEM Acoustics* ACA Solver with parameters *nruleb* = 3, *nrulef* = 1 has the lowest value of 3631.8 second. The CPU time from *FastBEM Acoustics* FMM Solver with *nruleb* = 3, *nrulef* = 1 and LMS Virtual.Lab with quadrature choice of near field = 2, far field =1 have the values of 9,472.2 second and 14,856.7 second respectively for the same BEM model.

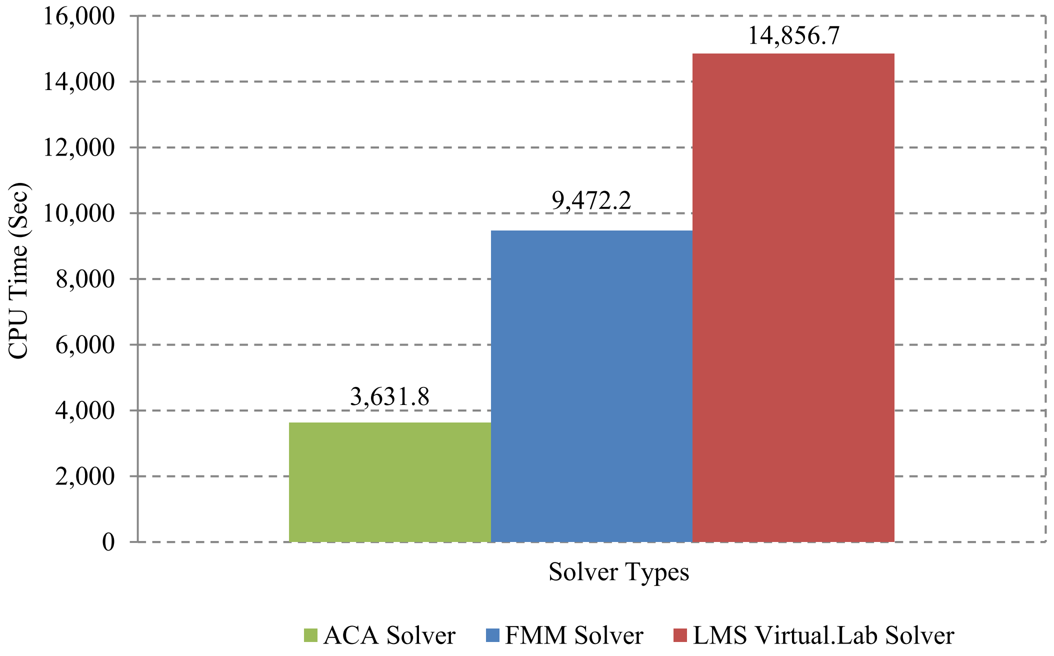


Figure 6-8 CPU Time Comparisons of different solver types

6.7 Solution Discussions

From Figure 6-6, we can see that the extreme values of the sound power are at those frequencies corresponding to the eigen values 1.20 Hz and 2.82 Hz, which means that most resonances of the system are at the natural frequencies responding to the wind pressure perturbation.

The *FastBEM Acoustics* ACA Solver with *nruleb* = 3, *nrulef* = 1 will give both of the accuracy and efficiency from the comparison plots of Figure 6-6 and Figure 6-8.

This application example clearly shows that large-scale noise prediction can be simulated with the FEM and BEM.

7 Discussions

7.1 Further Work

Future research could investigate the possibilities of expanding the application scope to more complex structures.

In this work, assuming a fixed base could potentially overestimate the system stiffness, and thus overestimate the natural frequency.

7.2 Conclusion

Structure acoustic analysis can be performed in a coupled fashion for determining the sound power level due to the vibration of the structure.

The translation code has been developed to connect the ABAQUS and FastBEM Acoustics analyses. The procedure presented in this thesis is suitable for easy implementation and it could be integrated in any of the many existing commercial numerical software packages for evaluating the acoustic field by coupling the FEM and BEM.

With the known normal velocity on the surface of the vibrating structure, the acoustic radiation can be calculated with the same procedure using FastBEM Acoustics. In numerical examples, the computational results of a pulsating spherical model and a plate model were presented, and the validity and feasibility of the proposed method were demonstrated.

Furthermore, a more realistic application of a wind turbine shows the capability of the coupled analysis is well suited for large-scale engineering problems.

8 Bibliography

[1] F. Fahy and P. Gardonio, *Sound and Structural Vibration: Radiation Transmission and Response*. Oxford, UK: Academic Press, 2007.

[2] T. W. Wu, *Boundary Element Acoustics : Fundamentals and Computer Codes*. Boston: WIT, 2000.

[3] M. Bebendorf, "Approximation of boundary element matrices," *Numer. Math.,* vol. 86, pp. 565-589, 2000.

[4] T. Schrefl, D. Süss, and J. Fidler, "Wavelet based matrix compression in numerical micromagnetics," *Technical Proceedings of the 2000 International Conference on Modeling and Simulation of Microsystems*, 2000, pp. 429-432.

[5] W. Hackbusch and Z. P. Nowak, "On the fast matrix multiplication in the boundary element method by panel clustering," *Numerische Mathmatik,* vol. 54, pp. 463-491, 1989.

[6] V. Rokhlin, "Rapid solution of integral equations of classical potential theory," *Computational Physics,* vol. 60, pp. 187-207, 1983.

[7] L. F. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*. Cambridge, MA: The MIT Press, 1988.

[8] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *Computational Physics,* vol. 73, pp. 325-348, 1987.

[9] Y. J. Liu, *Fast Multipole Boundary Element Method - Theory and Applications in Engineering*. Cambridge: Cambridge University Press, 2009.

[10] Y. J. Liu and N. Nishimura, "The fast multipole boundary element method for potential problems: A tutorial," *Engineering Analysis with Boundary Elements,* vol. 30, pp. 371-381, 2006.

[11] C. Bernardi, Y. Maday, and F. Rapetti, "Basics and some applications of the mortar element method," *GAMM-Mitt,* vol. 28, pp. 97-123, 2005.

[12] D. Stefanica, “Domain Decomposition Methods for Mortar Finite Elements,” Ph.D. Dissertation, Courant Institute of Mathematical Sciences, New York University, New York, USA, 1999.

[13] D. Brunner, M. Junge, and L. Gaul, “A comparison of FE-BE coupling schemes for large-scale problems with fluid-structure interaction,” *International Journal for Numerical Methods in Engineering,* vol. 77, pp. 664-688, 2009.

[14] A. J. Burton and G. F. Miller, “The application of integral equation methods to the solution of exterior boundary-value problems,” *Proceedings of the Royal Society A Mathematical Physical and Engineering Sciences,* vol. 323, pp. 201-210, 1971.

[15] L. Gaul and W. Wenzel, “A coupled symmetric BE-FE method for acoustic fluid-structure interaction,” *Engineering Analysis with Boundary Elements,* vol. 26, pp. 629-636, 2002.

[16] A. Frendi and J. Robinson, “Effect of acoustic coupling on random and harmonic plate vibrations,” *The American Institute of Aeronautics and Astronautics,* vol. 31, pp. 1992-1997, 1993.

[17] Y. Y. Lee, “Structural - acoustic coupling effect on the nonlinear natural frequency of a rectangular box with one flexible plate,” *Applied Acoustics,* vol. 63, pp. 1157-1175, 2002.

[18] D. Braess and W. Dahmen, “Stability estimates of the mortar finite element method for 3-dimensional problems,” *East-West J. Numer. Math,* vol. 6, pp. 249-264, 1998.

[19] F. B. Belgacem, “The Mortar finite element method with Lagrange multipliers,” *Numer. Math.,* vol. 84, pp. 173-197, 1999.

[20] *Abaqus Analysis User’s Manual*: Dassault Systèmes Simulia Corp., 2010.

[21] *FastBEM Acoustics V.3.0.0 User Guide*. Advanced CAE Research, LLC, Cincinnati, USA: http://www.fastbem.com/, 2012.

[22] *LMS Virtual.Lab - 3D Simulation*. Leuven, Belgium: LMS International http://www.lmsintl.com/virtuallab, 2011.

[23] R. D. Cook, D. S. Malkus, and M. E. Plesha, *Concepts and Applications of Finite Element Analysis*. New York: John Wiley & Sons, 2001.

[24] M. N. Currey and K. A. tunefare, “The radiation modes of baffled finite plates,” *The Journal of the Acoustical Society of America,* vol. 98, pp. 1570-1580, 1995.

[25] M. M. Sevice, “Technology White Paper on Wind Energy Potential on the U.S. Outer Continental Shelf,” http://ocsenergy.anl.gov, Washington, D.C.2006.

[26] M. AlHamaydeh and S. Hussain, “Optimized frequency-based foundation design for wind turbine towers utilizing soil–structure interaction,” *Journal of the Franklin Institute,* vol. 348, pp. 1470-1487, 2011.

[27] M. Jureczko, M. Pawlak, and A. Mężyk, “Optimisation of wind turbine blades,” *Journal of Materials Processing Technology,* vol. 167, pp. 463-471, 2005.

[28] B. Bi, “CAE Research Lab Report,” University of Cincinnati, Cincinnati, OH, USA, 2008.

[29] *Boundary Element Acoustics*. Leuven Belgium: LMS International http://www.lmsintl.com/acoustic-simulation, 2011.

Appendix A The Sample Input Files of the FEM and BEM Analyses

A.1 A Sample Input File of ABAQUS

A sample input file of ABAQUS for the simply supported plate problem is shown as below.

```
*Heading
*Preprint, echo=NO, model=NO, history=NO, contact=NO
** ----------------------------------------------------------------
**
** PART INSTANCE: Plate-1
**
*Node
      1, 0.0299999993, 0.0166669991, 0.00249999994
      2, 0.0299999993, 0.0500000007, 0.00249999994
      3,           0., 0.0500000007, 0.00249999994
      4,           0., 0.0166669991, 0.00249999994
      5, 0.0299999993, 0.0500000007,  -5.58793532e-11
      6, 0.0299999993, 0.0166669991,  -5.58793532e-11
      7,           0., 0.0166669991,  -5.58793532e-11
      8,           0., 0.0500000007,  -5.58793532e-11
      9, 0.0299999993,           0., 0.00249999994
     10,           0.,           0., 0.00249999994
*Element, type=SC8R
 1,  49, 262,  28,   1,  56, 292,  55,   6
 2, 262, 263,  29,  28, 292, 293,  54,  55
 3, 263, 264,  30,  29, 293, 294,  53,  54
 4, 264, 265,  31,  30, 294, 295,  52,  53
 5, 265, 266,  32,  31, 295, 296,  51,  52
 6, 266, 267,  33,  32, 296, 297,  50,  51
 7, 267,  34,   2,  33, 297,  71,   5,  50
 8,  48, 268, 262,  49,  57, 298, 292,  56
 9, 268, 269, 263, 262, 298, 299, 293, 292
10, 269, 270, 264, 263, 299, 300, 294, 293
*Nset, nset=Plate-1_Set-all, generate
   1,  693,    1
*Elset, elset=Plate-1_Set-all, generate
   1,  400,    1
*Nset, nset=Plate-1_Set-center
 1,
*Nset, nset=Plate-1_Set-edge
   2,   3,   4,   9,  10,  19,  20,  23,  34,  35,  36,  37,  38,  39,  40,  41
  42,  43,  44,  74,  75,  76,  77,  78,  79,  80, 121, 122, 123, 124, 125, 126
 127, 128, 129, 130, 131, 132, 133, 134, 135, 177, 178, 179, 180, 181, 182, 183
 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195
*Nset, nset=Plate-1_Set-corner-points
```

```
 3, 10, 19, 23
*Nset, nset=Plate-1_Set-corner-edge
 3,  8, 10, 14, 16, 18, 19, 22, 23, 25, 26, 27
*Elset, elset=Plate-1_Set-corner-edge
 42,  58, 102, 118, 162, 254, 302, 394
** Section: Section-1
*Shell Section, elset=Plate-1_Set-all, material=Material-1, controls=EC-1
0.005, 5
*System
**
** ELEMENT CONTROLS
**
*Section Controls, name=EC-1, hourglass=RELAX STIFFNESS, second order accuracy=YES
1., 1., 1.
**
** MATERIALS
**
*Material, name=Material-1
*Density
2700.,
*Elastic
 7e+10, 0.33
**
** BOUNDARY CONDITIONS
**
** Name: BC-1 Type: Displacement/Rotation
*Boundary
Plate-1_Set-corner-edge, 1, 1
Plate-1_Set-corner-edge, 2, 2
Plate-1_Set-corner-edge, 3, 3
Plate-1_Set-corner-edge, 4, 4
Plate-1_Set-corner-edge, 5, 5
Plate-1_Set-corner-edge, 6, 6
** ----------------------------------------------------------------
**
** STEP: Freq
**
*Step, name=Freq, perturbation
*Frequency, eigensolver=Lanczos, acoustic coupling=on, normalization=displacement
, 1., 6000., , ,
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT
*End Step
** ----------------------------------------------------------------
**
```

```
** STEP: Ssdd
**
*Step, name=Ssdd, perturbation
*Steady State Dynamics, direct, frequency scale=LINEAR, friction damping=NO
8., 4000., 500, 1.
**
** LOADS
**
** Name: Load-1   Type: Concentrated force
*Cload, load case=1
Plate-1_Set-center, 3, 1.
**
** OUTPUT REQUESTS
**
**
** FIELD OUTPUT: F-Output-3
**
*Output, field
*Node Output
V,
*Output, history, frequency=0
*End Step
```

A.2 A Sample Input File of FastBEM Acoustics

A sample input file of FastBEM Acoustics for the simply supported plate problem is shown as below.

```
A Plate Structure-Acoustic Analysis,             02-May-2012   21:51:51
Complete      1
Full      0       0
 868     436     441     400
0       0
        0
340.0    1.225      2e-05       1e-12
1.0    550.0      10      0      1
1       6     6     0    All
 $ Nodes:
    1   -0.253463   -0.967345   0.000000
    2   -0.253463   0.967345   0.000000
    3   0.000000   -1.000000   0.000000
    4   0.000000   1.000000   0.000000
    5   -0.442599   0.896720   0.000000
    6   -0.613695   0.789544   0.000000
    7   -0.759776   0.650185   0.000000
    8   -0.874888   0.484325   0.000000
    9   -0.954340   0.298724   0.000000
   10   -0.994892   0.100946   0.000000
 $ Elements and Boundary Conditions:
      404          18         2         4         2  ( 0.000000E+00, 4.761421E-02 )
      405           1        17         3         2  ( 0.000000E+00, 4.748958E-02 )
      406          32        18         4         2  ( 0.000000E+00, 4.761421E-02 )
      407          17        31         3         2  ( 0.000000E+00, 4.748958E-02 )
      408          46        32         4         2  ( 0.000000E+00, 4.761421E-02 )
      409          31        45         3         2  ( 0.000000E+00, 4.748958E-02 )
      410          60        46         4         2  ( 0.000000E+00, 4.761421E-02 )
      411          45        59         3         2  ( 0.000000E+00, 4.748958E-02 )
      412          74        60         4         2  ( 0.000000E+00, 4.761421E-02 )
      413          59        73         3         2  ( 0.000000E+00, 4.748958E-02 )
      414          88        74         4         2  ( 0.000000E+00, 4.761421E-02 )
      415          73        87         3         2  ( 0.000000E+00, 4.748958E-02 )
      416         102        88         4         2  ( 0.000000E+00, 4.761421E-02 )
      417          87       101         3         2  ( 0.000000E+00, 4.748958E-02 )
      418         116       102         4         2  ( 0.000000E+00, 4.761421E-02 )
      419         101       115         3         2  ( 0.000000E+00, 4.748958E-02 )
      420         130       116         4         2  ( 0.000000E+00, 4.761421E-02 )
 $ Field Points:
      1   1.000000E+01   1.000000E+01   5.000000E+00
      2   9.000000E+00   1.000000E+01   5.000000E+00
      3   8.000000E+00   1.000000E+01   5.000000E+00
      4   7.000000E+00   1.000000E+01   5.000000E+00
      5   6.000000E+00   1.000000E+01   5.000000E+00
      6   5.000000E+00   1.000000E+01   5.000000E+00
      7   4.000000E+00   1.000000E+01   5.000000E+00
      8   3.000000E+00   1.000000E+01   5.000000E+00
      9   2.000000E+00   1.000000E+01   5.000000E+00
     10   1.000000E+00   1.000000E+01   5.000000E+00
   $ Field Cells:
       1          1        2        23        22
       2          2        3        24        23
       3          3        4        25        24
       4          4        5        26        25
       5          5        6        27        26
       6          6        7        28        27
       7          7        8        29        28
       8          8        9        30        29
       9          9       10        31        30
      10         10       11        32        31
   $ End of the File
```

Appendix B The Translation Programs

B.1 A python Script to Build the FEM Model

The python script *a-build-model.py* will build the geometry, mesh and boundary conditions in ABAQUS used for the preprocessing of the FEM analysis. The whole code is listed below.

```
from abaqus import *
from abaqusConstants import *
from caeModules import *
from driverUtils import executeOnCaeStartup
executeOnCaeStartup()
Mdb()
modelName = 'Model-Plate-SC8R'

mdb.models.changeKey(fromName='Model-1', toName=modelName)
m = mdb.models[modelName]
m.setValues(noPartsInputFile=ON)

## create part
m.ConstrainedSketch(name='__profile__', sheetSize=200.0)
m.sketches['__profile__'].rectangle(point1=(0.0, 0.0),
   point2=(0.1, 0.05))
p = m.Part(dimensionality=THREE_D, name='Part-Plate', type=
   DEFORMABLE_BODY)
p.BaseSolidExtrude(depth=0.005, sketch=
   m.sketches['__profile__'])
del m.sketches['__profile__']

e, v, d = p.edges, p.vertices, p.datums
c = p.cells
cells = c.getSequenceFromMask(mask=('[#1 ]', ), )
p.Set(cells=cells, name='Set-all')

f = p.faces
p.DatumPlaneByOffset(plane=f[0], flip=SIDE2, offset=0.03)
p.DatumPlaneByOffset(plane=f[3], flip=SIDE2, offset=0.016667)

pickedCells = c.getSequenceFromMask(mask=('[#1 ]', ), )
p.PartitionCellByPlaneNormalToEdge(edge=e[3], cells=pickedCells,
   point=p.InterestingPoint(edge=e[3], rule=MIDDLE))
pickedCells = c.getSequenceFromMask(mask=('[#3 ]', ), )
d = p.datums
p.PartitionCellByDatumPlane(datumPlane=d[4], cells=pickedCells)
pickedCells = c.getSequenceFromMask(mask=('[#f ]', ), )
p.PartitionCellByDatumPlane(datumPlane=d[3], cells=pickedCells)
```

```
## create part set
verts = v.getSequenceFromMask(mask=('[#2 ]', ), )
p.Set(vertices=verts, name='Set-center')
edges = e.getSequenceFromMask(mask=('[#4a4000 #2806 ]', ), )
p.Set(edges=edges, name='Set-edge')
verts = v.getSequenceFromMask(mask=('[#409400 ]', ), )
p.Set(vertices=verts, name='Set-corner-points')
e = p.edges
edges = e.getSequenceFromMask(mask=('[#80812000 #9c000 ]', ), )
p.Set(edges=edges, name='Set-corner-edge')

## create material
mat = m.Material(name='Material-1')
mat.Elastic(table=((7.0E10, 0.33), ))
mat.Density(table=((2700, ), ))

## create section
m.HomogeneousShellSection(name='Section-1',
    preIntegrate=OFF, material='Material-1', thicknessType=UNIFORM,
    thickness=0.005, thicknessField='', idealization=NO_IDEALIZATION,
    poissonDefinition=DEFAULT, thicknessModulus=None, temperature=GRADIENT,
    useDensity=OFF, integrationRule=SIMPSON, numIntPts=5)

region = p.sets['Set-all']
p.SectionAssignment(region=region, sectionName='Section-1', offset=0.0,
    offsetType=MIDDLE_SURFACE, offsetField='',
    thicknessAssignment=FROM_SECTION)

## create assembly
a = m.rootAssembly
a.DatumCsysByDefault(CARTESIAN)
a.Instance(name='Plate-1', part=p, dependent=ON)
a.regenerate()

## create mesh
pickedRegions = c.getSequenceFromMask(mask=('[#ff ]', ), )
p.seedPart(size=0.005, deviationFactor=0.1, minSizeFactor=0.1)
p.setMeshControls(regions=pickedRegions, elemShape=HEX_DOMINATED,
    technique=SWEEP, algorithm=ADVANCING_FRONT, allowMapped=True)
elemType1 = mesh.ElemType(elemCode=SC8R, elemLibrary=STANDARD,
    secondOrderAccuracy=OFF, distortionControl=DEFAULT)
elemType2 = mesh.ElemType(elemCode=SC6R, elemLibrary=STANDARD)
elemType3 = mesh.ElemType(elemCode=UNKNOWN_TET, elemLibrary=STANDARD,
    secondOrderAccuracy=OFF, distortionControl=DEFAULT)
cells = c.getSequenceFromMask(mask=('[#ff ]', ), )
pickedRegions =(cells, )
p.setElementType(regions=pickedRegions, elemTypes=(elemType1, elemType2,
    elemType3))
p.generateMesh()

## create step
```

```
m.FrequencyStep(name='Freq', previous='Initial',
   maxEigen=2000.0, limitSavedEigenvectorRegion=None, minEigen=0.0)
m.steps['Freq'].setValues(maxEigen=6000, minEigen=1)
m.SteadyStateDirectStep(name='Ssdd', previous='Freq',
   frequencyRange=((8.0, 4000.0, 500, 1.0), ), scale=LINEAR,
   subdivideUsingEigenfrequencies=OFF)

## create output
m.fieldOutputRequests['F-Output-2'].suppress()
m.historyOutputRequests['H-Output-1'].suppress()
regionDef=m.rootAssembly.instances['Plate-1'].sets['Set-center']
m.FieldOutputRequest(name='F-Output-3',
   createStepName='Ssdd', variables=('V', ))

## BC
region = a.instances['Plate-1'].sets['Set-corner-edge']
m.DisplacementBC(name='BC-1', createStepName='Initial',
   region=region, u1=SET, u2=SET, u3=SET, ur1=SET, ur2=SET, ur3=SET,
   amplitude=UNSET, distributionType=UNIFORM, fieldName='', localCsys=None)

## load
region = a.instances['Plate-1'].sets['Set-center']
m.ConcentratedForce(name='Load-1',
   createStepName='Ssdd', region=region, cf3=1+0j, distributionType=UNIFORM,
   field='', localCsys=None)

## create job
j = mdb.Job(name=modelName, model=modelName, description='', type=ANALYSIS,
   atTime=None, waitMinutes=0, waitHours=0, queue=None, memory=80,
   memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,
   explicitPrecision=SINGLE, nodalOutputPrecision=FULL, echoPrint=OFF,
   modelPrint=OFF, contactPrint=OFF, historyPrint=OFF, userSubroutine='',
   scratch='', multiprocessingMode=DEFAULT, numCpus=1)
j.writeInput(consistencyChecking=OFF)

a.regenerate()
mdb.saveAs(modelName)

print '\nCongratulations, Done! ... \n'
```

B.2 A Python Script to Process the Boundary Elements

The python script *b-get-boundary.py* will convert the boundary mesh to three-node triangular element mesh and flip the normal directions of the boundary elements to make sure the directions are outward to the fluid domain and inward to the structural domain. The whole code is listed below.

```
import csv
import math
from abaqus import *
from abaqusConstants import *
from caeModules import *
from driverUtils import executeOnCaeStartup
executeOnCaeStartup()
Mdb()
modelName = 'Model-Plate-SC8R'

## import the *.inp file
m = mdb.ModelFromInputFile(inputFileName=modelName+'.inp', name=modelName)
p = m.parts['PART-1']
m.setValues(noPartsInputFile=ON)
e = p.elements
n = p.nodes
del mdb.models['Model-1']

nodes = n.getSequenceFromMask(mask=(
    '[#87fffffe #ff7ffe3f #3fffffff #ffffff80 #ffffffff #fffff #ffff8fff',
    ' #7fc01 #ff800000 #ffffffff:2 #ff ]', ), )
p.Set(nodes=nodes, name='Set-boundary-node')

print "offset the node number by 10000"
p.renumberNode(offset=10000)
print "renumber the boundary nodes start from 1"
p.renumberNode(nodes=nodes, startLabel=1, increment=1)

## write out the model with renumbered
j = mdb.Job(name=modelName+'-Renum', model=modelName, description='',
    type=ANALYSIS, atTime=None, waitMinutes=0, waitHours=0, queue=None,
    memory=80, memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,
    explicitPrecision=SINGLE, nodalOutputPrecision=FULL, echoPrint=OFF,
    modelPrint=OFF, contactPrint=OFF, historyPrint=OFF, userSubroutine='',
    scratch='', multiprocessingMode=DEFAULT, numCpus=1)
j.writeInput(consistencyChecking=OFF)
j.submit()
print 'job: ' + modelName + ' -- submitted!'
```

```
## covert solid to boundary shell mesh
p.convertSolidMeshToShell()
print 'Convert all solid mesh to shell...'
## convert four node quad elements to three node tri elements
print 'Convert four node quad elements to three node tri elements...'
try:
    e1 = p.elements
    elems1 = e1[0:2984]
    p.splitElement(elements=elems1)
except:
    print 'NOTHING Converted...'
## renumber the elements
p.renumberElement(startLabel=1, increment=1)

## flip the normal direction
print 'flip the normal direction...'
elements = e.getSequenceFromMask(mask=('[#ffffffff:17 #ffff ]', ), )
regions = regionToolset.Region(elements=elements)
p.flipNormal(regions=regions)
p.regenerate()

## write output csv file
def writeFile(fileName,nodeID,elmCon):
    print 'Writing to output file: \"' + fileName + '\""...'
    csvWriter = csv.writer(open(fileName, 'wb'), delimiter=',')
    csvWriter.writerow(['Total '+str(len(nodeID))+' boundary nodes:'])
    csvWriter.writerow(nodeID)
    csvWriter.writerow(['Total '+str(len(elmCon))+' boundary elements:'])
    for i in range(len(elmCon)):
        csvWriter.writerow(elmCon[i])

## get element connectivities
def getElemCon(allElm, allNode):
    elmList = []
    elmCon = []
    for count in range(len(allElm)):
        elmCon_row = []
        elmList.append(allElm[count].label)
        elmCon_row.append(allElm[count].label)
        connecti = allElm[count].connectivity
        elmCon_row.append(allNode[connecti[0]].label)
        elmCon_row.append(allNode[connecti[1]].label)
        elmCon_row.append(allNode[connecti[2]].label)
        ## get the nomal direction
        local_a = allNode[connecti[0]].coordinates
        local_b = allNode[connecti[1]].coordinates
        local_c = allNode[connecti[2]].coordinates
        ## inward vectors
        vector_p = [local_b[0]-local_a[0], local_b[1]-local_a[1],
                    local_b[2]-local_a[2]]
        vector_q = [local_c[0]-local_b[0], local_c[1]-local_b[1],
                    local_c[2]-local_b[2]]
        normal = [vector_p[1]*vector_q[2] - vector_p[2]*vector_q[1],
                  vector_p[2]*vector_q[0] - vector_p[0]*vector_q[2],
                    vector_p[0]*vector_q[1] - vector_p[1]*vector_q[0]]
        normal_abs = math.sqrt(normal[0]**2 + normal[1]**2 + normal[2]**2)
        ## n -- normalized direction
        n = []
        for i in range(len(normal)):
            n.append(normal[i]/normal_abs)
            elmCon_row.append(normal[i]/normal_abs)
        elmCon.append(elmCon_row)
    return elmCon

## main programme
allNode = p.nodes
print ' ' + str(len(allNode)) + ' nodes found in the boundary shell mesh'
allElm = p.elements
print ' ' + str(len(allElm)) + ' elements found in the boundary shell mesh'
nodeList = []
for count in range(len(allNode)):
    nodeList.append(allNode[count].label)
elmCon = getElemCon(allElm, allNode)

writeFile('boundary.csv', nodeList, elmCon)
print '\nCongratulations, Done! ... \n'
```

B.3 A Python Script to Get the Field Points and Cells

The python script c-*get-field.py* will generate the field points and cells for the BEM acoustic analysis. The whole code is listed below.

```
import csv
from abaqus import *
from caeModules import *
from driverUtils import executeOnCaeStartup
executeOnCaeStartup()
Mdb()

print '\n' + '-'*86
## first import the *.inp file
mdb.ModelFromInputFile(inputFileName=
   'Job-Model-Ssdd-wfield.inp', name='Job-Model-Ssdd-wfield')
p = mdb.models['Job-Model-Ssdd-wfield'].parts['PART-1']
del mdb.models['Model-1']

p.deleteElement(elements=p.sets['SOLID-GEO-1_SET-ALL-SOLID'],
   deleteUnreferencedNodes=ON)
print 'Delete the mesh in the set: \'SOLID-GEO-1_SET-ALL-SOLID\''
p.regenerate()
p.renumberNode(startLabel=1, increment=1)
p.renumberElement(startLabel=1, increment=1)
print 'Renumbering the field points and cells...'
p.regenerate()

def writeFile(fileName,nodeList,elmCon):
   print 'Writing to output file: \''' + fileName + '\''''...'
   fileID=open(fileName,'w')
   format='%8d   %12E   %12E   %12E\n'
   fileID.write(' $ Field Points:\n')
   for i in range(len(nodeList)):
      fileID.write(format %(nodeList[i][0],nodeList[i][1],
                        nodeList[i][2],nodeList[i][3]))
   format='%10d   %10d    %7d    %7d    %7d\n'
   fileID.write(' $ Field Cells:\n')
   for i in range(len(elmCon)):
      fileID.write(format %(elmCon[i][0],elmCon[i][1],
                        elmCon[i][2],elmCon[i][3],
                        elmCon[i][4]))

   fileID.write(' $ End of the File')
   fileID.close

def getElemCon(allElm, allNode):
   elmCon = []
```

```
    for count in range(len(allElm)):
        elmCon_row = []
        elmCon_row.append(allElm[count].label)
        connecti = allElm[count].connectivity
        elmCon_row.append(allNode[connecti[0]].label)
        elmCon_row.append(allNode[connecti[1]].label)
        elmCon_row.append(allNode[connecti[2]].label)
        elmCon_row.append(allNode[connecti[3]].label)
        elmCon.append(elmCon_row)
    print ' ' + str(len(elmCon)) + ' field cells have been processed.'
    return elmCon

## main programme
allfieldNode = p.nodes
print ' ' + str(len(allfieldNode)) + ' nodes found in the field mesh.'
allfieldElm = p.elements
print ' ' + str(len(allfieldElm)) + ' elements found in the field mesh.'
nodeList = []
for count in range(len(allfieldNode)):
    nodeList_row = []
    nodeList_row.append(allfieldNode[count].label)
    nodeList_row.append(allfieldNode[count].coordinates[0])
    nodeList_row.append(allfieldNode[count].coordinates[1])
    nodeList_row.append(allfieldNode[count].coordinates[2])
    nodeList.append(nodeList_row)
print ' ' + str(len(nodeList)) + ' field points have been processed.'

elmCon = getElemCon(allfieldElm, allfieldNode)
writeFile('field.csv', nodeList, elmCon)
print '\nCongratulations, Done! ... \n'
```

B.4 A Python Script to Output the FEM Results and Generate the BEM Input Files

The python script *d-output-odb.py* is developed for the velocity calculations of the boundary triangular elements. Here below lists the *getVel* function in the script, which is the core function in the coupling program to get both the real part and imaginary part of the velocity from the structural analysis.

```
##------------------------------------------------------------------------------
## Purpose: Get velocity for the triangular element
## Input: Boundary nodes and elements information
## Return: Both the real part and imaginary part of the velocity
##------------------------------------------------------------------------------
def getVel(stepName, elmCon, allNode, boundNodes, cspeed):
    myStep = o1.steps[stepName]
    totalframe = myStep.frames
    print '\n' + str(len(totalframe)) + ' frames are found in this step --> ' + stepName
    print str(len(totalframe)-1) + ' frequencies are found in this step --> ' + stepName

    ## open *.jbc file
    fileID = open('input.jbc', 'w')

    for f in range(len(totalframe)):
        if f > 0 and f < len(totalframe):
            fileID.write('$ BC Values for Frequency No. ' + str(f) + ':\n')
            ## current frame of the step
            frame = myStep.frames[f]
            ## current frequency of the frame
            freq = frame.frameValue
            temp = '! Frequency (Hz)'
            fileID.write('  %s              %s\n' %(str(freq), temp))
            fileID.write('$ Boundary Elem No.  BC Type   BC Values\n')
            fields = frame.fieldOutputs
            k = 2*math.pi*freq/cspeed
            print '-'*86
            print 'Processing the Frame: ' + str(f)
            print ' frequency: ' + str(freq)
            print ' ka: ' + str(k)

            nodeVel = []
            for i in range(len(allNode)):
                if allNode[i].label in boundNodes:
                    temp = []
                    vel = fields['V'].getSubset(region=allNode[i])
                    velVal_real = vel.values[0].dataDouble
                    velVal_img = vel.values[0].conjugateDataDouble
                    temp.append(allNode[i].label)
                    temp.append(velVal_real[0])
                    temp.append(velVal_real[1])
                    temp.append(velVal_real[2])
                    temp.append(velVal_img[0])
                    temp.append(velVal_img[1])
                    temp.append(velVal_img[2])
                    nodeVel.append(temp)
            print str(len(nodeVel)) + ' nodes have been used for the velocity ' +\
                    'calculations...'

            print 'Loop on boundary element level, averaging and projecting to normal direction...'
            elmConVel = []
            for j in range(len(elmCon)):
                elmConVel_row = []
                node1 = elmCon[j][1]
                node2 = elmCon[j][2]
                node3 = elmCon[j][3]
                ## normal direction of the element
                normalx = elmCon[j][4]
                normaly = elmCon[j][5]
                normalz = elmCon[j][6]

                ## searchVel - find the velocity from the node label
                ## average the element velocities of three nodes
                ## both real and imagnary parts calculated
                v1_real_avg = (searchVel(node1, nodeVel, 'v1-real') +
                                searchVel(node2, nodeVel, 'v1-real') +
                                searchVel(node3, nodeVel, 'v1-real') )/3.0
                v2_real_avg = (searchVel(node1, nodeVel, 'v2-real') +
                                searchVel(node2, nodeVel, 'v2-real') +
                                searchVel(node3, nodeVel, 'v2-real') )/3.0
                v3_real_avg = (searchVel(node1, nodeVel, 'v3-real') +
                                searchVel(node2, nodeVel, 'v3-real') +
                                searchVel(node3, nodeVel, 'v3-real') )/3.0

                v1_img_avg = (searchVel(node1, nodeVel, 'v1-img') +
                                searchVel(node2, nodeVel, 'v1-img') +
                                searchVel(node3, nodeVel, 'v1-img') )/3.0
                v2_img_avg = (searchVel(node1, nodeVel, 'v2-img') +
                                searchVel(node2, nodeVel, 'v2-img') +
                                searchVel(node3, nodeVel, 'v2-img') )/3.0
                v3_img_avg = (searchVel(node1, nodeVel, 'v3-img') +
                                searchVel(node2, nodeVel, 'v3-img') +
                                searchVel(node3, nodeVel, 'v3-img') )/3.0

                ## project the velocity to the normal direction
                ## the normal direction has already normalized
                mag_real = v1_real_avg*normalx + v2_real_avg*normaly + v3_real_avg*normalz
                mag_img = v1_img_avg*normalx + v2_img_avg*normaly + v3_img_avg*normalz
                vn1_real = mag_real*normalx
                vn2_real = mag_real*normaly
```

```
                vn3_real = mag_real*normalz
                vn1_img = mag_img*normalx
                vn2_img = mag_img*normaly
                vn3_img = mag_img*normalz

                elmConVel_row.append(elmCon[j][0])
                elmConVel_row.append(elmCon[j][1])
                elmConVel_row.append(elmCon[j][2])
                elmConVel_row.append(elmCon[j][3])
                elmConVel_row.append(vn1_real)
                elmConVel_row.append(vn2_real)
                elmConVel_row.append(vn3_real)
                elmConVel_row.append(vn1_img)
                elmConVel_row.append(vn2_img)
                elmConVel_row.append(vn3_img)

                elmConVel.append(elmConVel_row)

            elmVelMag = []
            for k in range(len(elmConVel)):
                elmVelMag_row = []
                realMag = math.sqrt(elmConVel[k][4]**2 + elmConVel[k][5]**2 +
                           elmConVel[k][6]**2)
                imgMag = math.sqrt(elmConVel[k][7]**2 + elmConVel[k][8]**2 +
                           elmConVel[k][9]**2)
                elmVelMag_row.append(elmConVel[k][0])
                elmVelMag_row.append(elmConVel[k][1])
                elmVelMag_row.append(elmConVel[k][2])
                elmVelMag_row.append(elmConVel[k][3])
                elmVelMag_row.append(realMag)
                elmVelMag_row.append(imgMag)

                elmVelMag.append(elmVelMag_row)

                ## write out the *.jbc file
                format='%10d       2       ( %12E, %12E )\n'
                fileID.write(format %(elmVelMag_row[0],
                                  elmVelMag_row[4],elmVelMag_row[5]))

    fileID.write('$ End of the File')
    fileID.close

    return elmVelMag
```

B.5 A Python Script to Output the Parameters for FastBEM Acoustics Solvers

The python script *e-fmm.py* will output the parameters used for the fast multipole, ACA BEM and the iterative linear equation solver. The whole code is listed below.

```
def writeFmm(fileName):
    print 'Writing out --> ' + fileName
    maxl = 100
    levmx = 10
    nexp = 6
    tolerance = 1.E-4
    nstk = 100000000
    iteration = 500
    ncellmx = 60000
    nleafmx = 50000
    maxk = 100
    ifmm = 0
    nexpf = 10
    fileID=open(fileName,'w')
    fileID.write('Parameters for fast multipole method:\n')
    format='%s    %s    %s    %s    %s    %s\n'
    fileID.write(format %(maxl, levmx, nexp, tolerance, nstk, iteration))
    format='%s    %s    %s    %s    %s\n'
    fileID.write(format %(ncellmx, nleafmx, maxk, ifmm, nexpf))

fileName = 'input.fmm'
writeFmm(fileName)
```