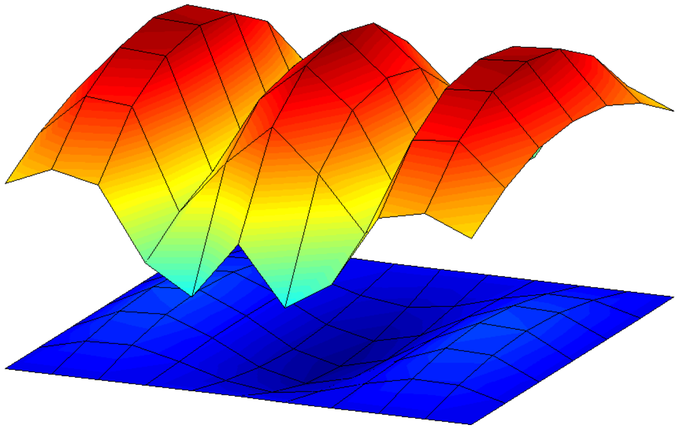
LUND UNIVERSITY



STRUCTURE-ACOUSTIC ANALYSIS USING BEM/FEM; IMPLEMENTATION IN MATLAB

FREDRIK HOLMSTRÖM

Structural Mechanics & Engineering Acoustics

Master's Dissertation

Structural Mechanics

ISRN LUTVDG/TVSM--01/5107--SE (1-100)
ISSN 0281-6679

Engineering Acoustics

ISRN LUTVDG/TVBA--01/5029--SE (1-100)
ISSN 0281-8477

STRUCTURE-ACOUSTIC ANALYSIS USING BEM/FEM; IMPLEMENTATION IN MATLAB

Master's Dissertation by

FREDRIK HOLMSTRÖM

Supervisors

PETER DAVIDSSON,
Structural Mechanics

JONAS BRUNSKOG,
Engineering Acoustics

Printed by KFS i Lund AB, Lund, Sweden, May 2001.

For information, address:

Structural Mechanics, LTH, Lund University, Box 118, SE-221 00 Lund, Sweden.
Homepage: http://www.byggmek.lth.se

Engineering Acoustics, LTH, Lund University, Box 118, SE-221 00 Lund, Sweden.
Homepage: http://www.akustik.lth.se

Abstract

This master dissertation describes the process of implementing a coupling between the boundary element method (BEM) and the finite element method (FEM) for three dimensional time harmonic structure-acoustic models in CALFEM, which is a finite element toolbox to MATLAB. Since no boundary elements earlier have been represented in CALFEM the development and implementation of constant and linear boundary elements is also described in this thesis. To verify the correctness of the implemented functions and to show how they are used three model examples are performed: one using only BEM, and two structure-acoustics.

Keywords: BEM, FEM, Coupled, CALFEM, Vibro-Acoustics, Acoustics, Implementation.

Cover Picture: Sound pressure level on vibrating plate.

Preface

This master dissertation has been performed at the Division of Structural Mechanics and the Division of Engineering Acoustics, Lund University, during the autumn and spring term of year 2000/2001.

Supervisors for this work have been Eng. Lic. Jonas Brunskog and M. Sc. Peter Davidsson. I would like to express my gratitude for their support, during times of both success and frustration. I would also like to thank the personal at Structural Mechanics and at Building Acoustics, that have been involved in my work.

Finally, Andreas, Björn, Anders, and Peter deserves a thank, for keeping the spirit up in the "master's dissertation room".

Lund, May 2001

Fredrik Holmström

Summary

The purpose of this master dissertation is to implement BEM (*Boundary Element Method*) and a coupling between BEM and FEM (*Finite Element Method*) in CALFEM (a FEM toolbox to MATLAB) for structure-acoustic models.

Two different boundary elements, which formulations are based on the Helmholtz integral equation, has been developed: a one node constant, and a four node linear. Both elements are quadrilateral and can take any orientation in the three dimensional space.

The linear boundary element can be coupled with two different four node finite elements: a plate element with 12 degrees of freedom, and a shell element with 24 degrees of freedom.

Since BEM and coupled BEM/FEM problems, modelled with functions developed in this dissertation, can be very time consuming, the most important future improvement is to construct time reducing measures.

Contents

Chapter 1

Introduction

1.1 Background

The boundary element method (BEM) is a numerical method for obtaining approximate solutions to boundary integral equations. Such equations provide a well defined formulation of boundary-value problems in different branches of engineering, e.g. elasticity, plasticity, fracture mechanics, ground water flow, wave propagation and electromagnetic field problems [1]. However, this report only concerns BEM for time-harmonic acoustic problems in fluid domains.

Initially, boundary integral equations were considered to be a different type of analytical method, somewhat unrelated to other approximate methods such as the finite elements. In the 1960's the integral equations had important influence in the development of finite elements. In the late 1970's the expression "Boundary Elements" started to appear in technical research paper and books. Lately, in similarity with FE methods, the interest for BE methods has increased in pace with the development of the computer.[1][2]

In both FEM and BEM the problem domain is divided into finite elements. However, in FEM the entire problem domain is divided into elements, but in BEM only the bounding surface of the domain is divided, Fig. 1.1. The elements are therefore called *boundary elements*.

Basically BEM consists of two different approaches, the indirect (I-BEM) and the direct approach (D-BEM), Fig. 1.2. For D-BEM at least one closed boundary is required and the physical variables (pressure and normal velocity for acoustic problems) can only be considered on one side of the surface. Using I-BEM, both sides of a surface can be considered.

I-BEM can also deal with open boundary problems. Among the two, D-BEM is the most widely spread, and it is also the direct approach that will be used in these thesis. Notice that, when the abbreviation BEM is used in the following text it refers to the direct approach.[3]

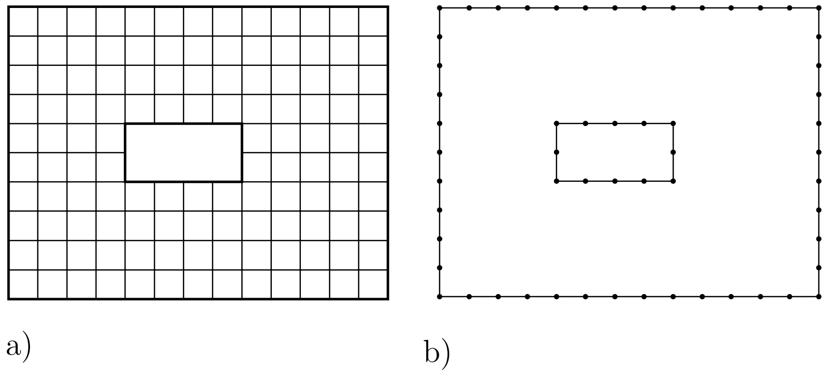


Figure 1.1: *a) Finite element mesh, b) Boundary element mesh*

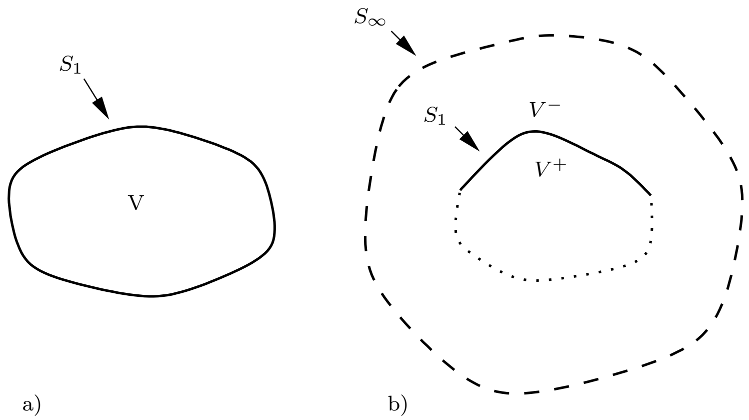


Figure 1.2: *a) Direct BEM, b) Indirect BEM*

1.2 Purpose with the Thesis

Since BEM and FEM both are matrix based methods it is possible to connect them. The purpose with this thesis is to develop and implement BEM functions, and functions that enable connection between BEM and FEM, in CALFEM (Computer Aided Learning of the Finite Element

Method). CALFEM is a toolbox to Matlab and is developed at the Division of Structural Mechanics and the Department of Solid Mechanics at Lund Univeristy for educational purpose [4]. The implementation is performed in Matlab.

1.3 Basic Relationships

As mentioned above this thesis only concerns BEM for time-harmonic acoustic problems in uid domains. The corresponding boundary integral equation for such problems, on which the BE formulation is founded, is the Helmholtz integral equation

$$cp(\mathbf{R}) = \int_S \left(p(\mathbf{R}_0)\frac{\partial g}{\partial \hat{n}_0} - g(|\mathbf{R}-\mathbf{R}_0|)\frac{\partial p}{\partial \hat{n}_0} \right) dS, \tag{1.1}$$

where $\hat{n}_0$ is the surface unit normal vector, and

$$c = \begin{cases} 1, & \mathbf{R} \text{ in fluid domain } V \\ \frac{1}{2}, & \mathbf{R} \text{ on smooth boundary of fluid domain } V \\ \frac{\Omega}{4\pi}, & \mathbf{R} \text{ on nonsmooth boundary of fluid domain } V \\ & (\Omega \text{ is the solid angle}) \\ 0, & \mathbf{R} \text{ outside fluid domain } V. \end{cases}$$

g stands for the free-space Green's function

$$g(|\mathbf{R}-\mathbf{R}_0|) = \frac{e^{-ik|\mathbf{R}-\mathbf{R}_0|}}{4\pi|\mathbf{R}-\mathbf{R}_0|}, \tag{1.2}$$

in which $\mathbf{R}_0$ denotes a point located on the bounding surface S.[5][6]

The bounding surface is submitted to an acoustic pressure and a normal velocity. One of these two physical properties must be known at every point on the bounding surface, so that a BEM problem can be solved. The known property is called *boundary condition*. The specific normal impedance, which states the relationship between pressure and normal velocity, can also serve as a boundary condition [7].

$$\text{pressure:} \quad p \quad \text{on } S_p \tag{1.3}$$

$$\text{normal velocity:} \quad v = \frac{i}{\rho_0\omega}\frac{\partial p}{\partial \hat{n}_0} \quad \text{on } S_v \tag{1.4}$$

$$\text{specific normal impedance:} \quad Z = \frac{p}{v} = -i\rho_0\omega p\frac{\partial \hat{n}_0}{\partial p} \quad \text{on } S_z \tag{1.5}$$

Since the Sommerfeld radiation condition [7], Eq. (1.6), is satisfied in the Helmholtz integral equation (1.1) the BEM can handle problems with unbounded acoustic domains, i.e. infinite and semi-infinite problems, which can not be done with FEM. This ability is one of the points of using BEM.

$$\lim_{|\mathbf{R}-\mathbf{R}_0|\to\infty} |\mathbf{R}-\mathbf{R}_0| \left(\frac{\partial g}{\partial |\mathbf{R}-\mathbf{R}_0|} + ikg \right) = 0. \tag{1.6}$$

1.4 Essential Features

The development of this master dissertation followed a certain procedure which reflects on how the chapters in this report are sorted. The essential features on how the work was performed are listed below:

- Derivation of the Helmholtz integral equation.
- Discretization of the Helmholtz integral equation and development of boundary elements for implementation in CALFEM.
- Performance of a problem, with known analytical solution, with the implemented boundary elements in order to control the correctness of the numerical solution.
- Development and implementation of a method that couples BEM with FEM for structure-acoustic models.
- Performance of a BEM/FEM example to ensure the correctness of the implemented coupling method.

Chapter 2

Fundamental Functions and Equations

The Helmholtz integral Eq. (1.1) and the free-space Green's function, Eq. (1.2), are fundamental for the BEM formulation but, generally, they are seldom derived in an easily comprehensible way in the technical literature. Therefore, in the following sections, effort has been put into an attempt to derive them in a plain manner.

This chapter is mostly based on *Sound, Structures and their Interaction* [8].

2.1 The Helmholtz Equation

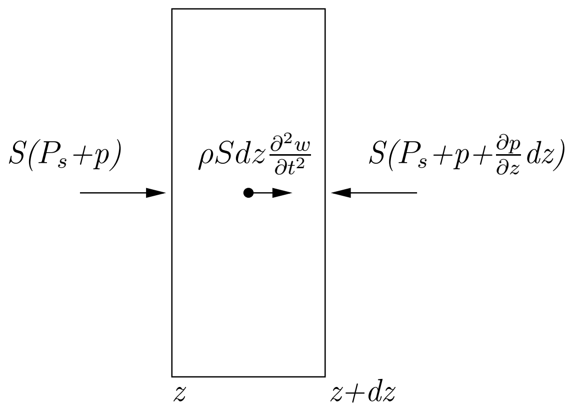


Figure 2.1: *One-dimensional force acting on a fluid volume element*

Consider a volume element of length dz, cross-section area S, and mass $\rho S dz$ located in an effectively infinite, compressible fluid medium of density ρ. In order to fulfil equilibrium the forces from Fig. 2.1 gives:

$$S(P_S + p) - S\left(P_S + p + \frac{\partial p}{\partial z}dz\right) = \left(\frac{\partial^2 w}{\partial t^2}\right)\rho S dz, \tag{2.1}$$

where w is the nodal displacement. By dividing all terms of Eq. (2.1) with the volume $V = Sdz$ the Euler's equation is obtained

$$\frac{\partial p}{\partial z} = -\rho\frac{\partial^2 w}{\partial t^2}. \tag{2.2}$$

The compressibility of the medium is now introduced by defining the bulk modulus B that relates the pressure change p applied to a fluid element and the resulting condensation, or fractional density change, $d\rho/\rho$:

$$p = B\frac{d\rho}{\rho}. \tag{2.3}$$

The requirement that the mass of the volume V remain constant,

$$d(\rho V) = \rho dV + V d\rho = 0, \tag{2.4}$$

gives an alternative formulation of Eq. (2.3):

$$p = -B\frac{dV}{V}. \tag{2.5}$$

Consider the one-dimensional motion of the mass element of Fig. 2.2. The change in volume is

$$dV = S\left[w + \left(\frac{\partial w}{\partial z}\right)dz\right] - Sw. \tag{2.6}$$

Eq. (2.6) substituted in Eq. (2.5) results in

$$p = -B\frac{\partial w}{\partial z}. \tag{2.7}$$

If Eq. (2.2) is differentiated once with respect to z, and Eq. (2.7) twice with respect to t they can be combined and the one-dimensional wave equation is obtained

$$\frac{\partial^2 p}{\partial z^2} - \frac{\rho}{B}\frac{\partial^2 p}{\partial t^2} = 0. \tag{2.8}$$

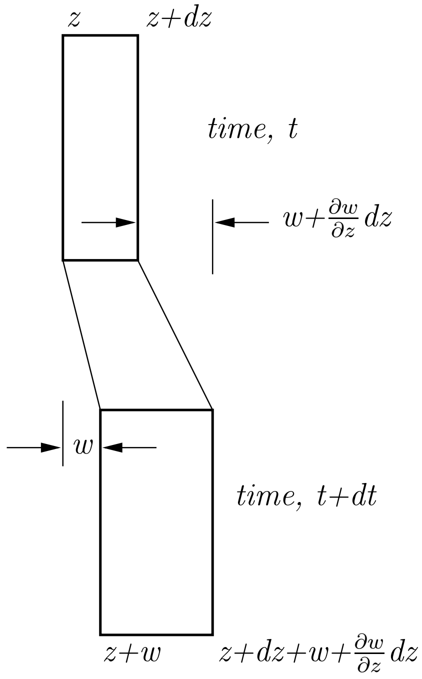


Figure 2.2: *Dilatation of a fluid in a one-dimensional pressure field*

The corresponding wave equation for three dimensions is

$$\nabla^2 p - \frac{\rho}{B}\frac{\partial^2 p}{\partial t^2} = 0. \tag{2.9}$$

In order to satisfy Eq. (2.9) the pressure must be position and time dependent, $p = p(x, y, z, t)$. If the solution is assumed time harmonic, i.e $p = p(x, y, z)e^{(-i\omega t)}$, Eq. (2.9) can be rewritten as,

$$\nabla^2 p + \frac{\omega^2 \rho}{B} p = 0,$$

where ω is the angular frequency. Introducing $c = (B/\rho)^{1/2}$ as the speed of sound in acoustic medium the three-dimensional Helmholtz equation is obtained,

$$(\nabla^2 + k^2)p = 0, \tag{2.10}$$

where k is the wave number expressed as ω/c.

2.2 The Free-Space Green's Function

The free-space Green's function, Eq. (1.2), is a solution of the inhomogeneous Helmholtz equation, which was formulated in homogeneous form in section (2.1):

$$\left(\nabla^2 + k^2\right) g\left(|\mathbf{R} - \mathbf{R}_0|\right) = -\delta\left(\mathbf{R} - \mathbf{R}_0\right). \tag{2.11}$$

Here $\delta\left(\mathbf{R} - \mathbf{R}_0\right)$ is the three-dimensional Dirac delta function defined by the value of its integral when integrated over a volume V:

$$\int_V \Phi(\mathbf{R}_0)\delta(\mathbf{R} - \mathbf{R}_0)dV(\mathbf{R}_0) = \begin{cases} \Phi(\mathbf{R}), & \mathbf{R} \text{ in } V \\ \dfrac{\Phi(\mathbf{R})}{2}, & \mathbf{R} \text{ on boundary of } V \\ 0. & \mathbf{R} \text{ outside } V \end{cases} \tag{2.12}$$

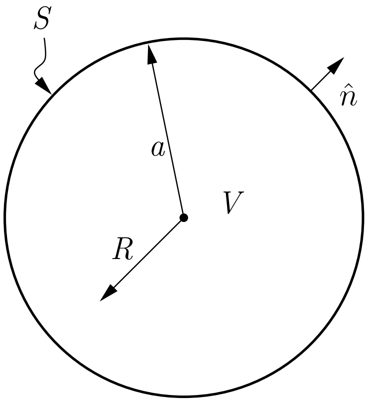


Figure 2.3: *Volume and surface integrals used in the construction of the free-space Green's function*

Green's free-space function must also satisfy the Sommerfeld radiation condition, Eq. (1.6), which will ensure that the Helmhotz integral equation (1.1) represents outward traveling waves. Using spherical coordinates and with R=$|\mathbf{R} - \mathbf{R}_0|$ the homogeneous Helmholtz equation (2.10) can be written as

$$\left(\frac{\partial^2}{\partial R^2} + \frac{2}{R}\frac{\partial}{\partial R} + k^2\right) g(R) = 0, \quad R > 0, \tag{2.13}$$

whose general solution is

$$g(R) = \frac{1}{R}\left(A_+ e^{ikR} + A_- e^{-ikR}\right). \tag{2.14}$$

To satisfy the Sommerfeld boundary condition, Eq. (1.6), A_+ must be set equal to zero. The other term is determined by integrating Eq. (2.11) over a spherical volume element of radius a concentric with the singular point $R = 0$. The definition of δ, Eq. (2.12), indicates that this integral must equal minus unity (notice that $dV = 4\pi R^2 dR$):

$$\lim_{a\to 0} 4\pi \int_0^a (\nabla^2 + k^2) g R^2 dR = -1$$

$$\Leftrightarrow$$

$$\lim_{a\to 0} 4\pi A_- \left(\int_0^a \nabla^2 \left(\frac{e^{-ikR}}{R} \right) R^2 dR + k^2 \int_0^a e^{-ikR} R dR \right) = -1. \quad (2.15)$$

The integration of the second integral results in:

$$k^2 \int_0^{a\to 0} e^{-ikR} R dR = \left[e^{-ikR}(1 + ikR) \right]_0^{a\to 0} = e^{-ika}(1 + ika) - 1 = 0.$$

In order to integrate the integral with the ∇^2 term from Eq. (2.15) Gauss's integral theorem is used,

$$\int_V \nabla \cdot \mathbf{F} dV = \int_S \mathbf{F} \cdot \hat{n} dS, \quad (2.16)$$

where $\hat{n}$ is the unit vector pointing out of volume V, and $\mathbf{F}$ is a vector field. If the vector field $\mathbf{F}$ is expressed as

$$\mathbf{F} \equiv \nabla \left(\frac{A_- e^{-ikR}}{R} \right), \quad (2.17)$$

Eq. (2.15) combined with Eqs. (2.16-2.17) can be written as:

$$\int_S \nabla \left(\frac{A_- e^{-ikR}}{R} \right) \cdot \hat{n} dS = 4\pi R^2 \nabla \left(\frac{A_- e^{-ikR}}{R} \right) \cdot \hat{n}$$
$$= -4\pi A_- e^{-ikR}(ikR + 1).$$

Because $R = a \to 0$ this will result in:

$$-4\pi A_- e^{-ika}(ika + 1) = -4\pi A_- = -1$$
$$\Leftrightarrow$$
$$A_- = \tfrac{1}{4\pi}.$$

When this is inserted in Eq. (2.14) the Green's free-space function is obtained:

$$g(R) = \frac{e^{-ikR}}{4\pi R}.$$

2.3 The Helmholtz Integral Equation

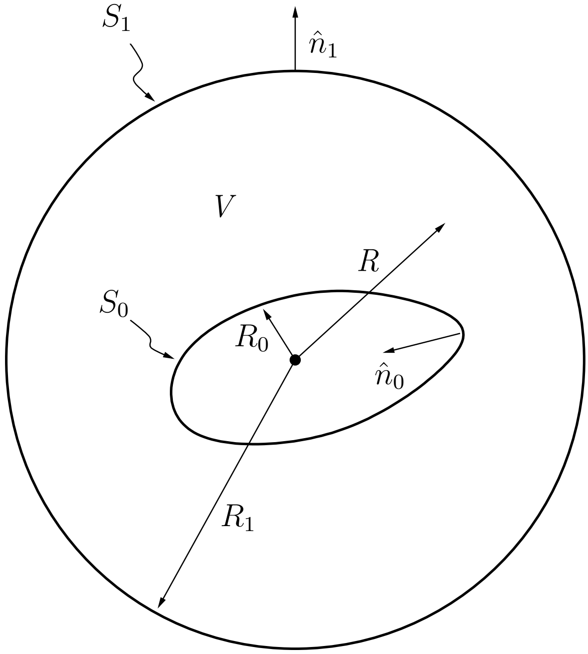


Figure 2.4: *Volume and surface integrals used in the construction of the Helmholtz integral equation*

Fig. 2.4 shows a pressure field $p(\mathbf{R})$ in volume $V(\mathbf{R})$ bounded by the surfaces S_0 and S_1. The pressure field is satisfying the Helmholtz equation (2.10) and the solution is subject to the boundary condition prescribed over the radiating surface S_0 (compare with Eq. (2.2)):

$$\frac{\partial p}{\partial \hat{n}_0} = -\rho \frac{\partial^2 w}{\partial t^2} \qquad \text{on } S_0(\mathbf{R}_0), \tag{2.18}$$

and to the condition that there are no sources except on this surface.

It is desired to obtain an integral representation of the pressure field in the form of Gauss's integral theorem, Eq. (2.16). Formulated in terms of the $\mathbf{R}_0$ coordinate system and with

$$\mathbf{F} \equiv p(\mathbf{R}_0)\nabla_0 g(|\mathbf{R} - \mathbf{R}_0|) - g(|\mathbf{R} - \mathbf{R}_0|)\nabla_0 p(\mathbf{R}_0) \tag{2.19}$$

the integrand of the volume integral, Eq. (2.16), becomes

$$\begin{aligned} \nabla_0 \cdot (p\nabla_0 g - g\nabla_0 p) = \nabla_0 p \cdot \nabla_0 g + p\nabla_0^2 g - \nabla_0 g \cdot \nabla_0 p - g\nabla_0^2 p \\ = p\nabla_0^2 g - g\nabla_0^2 p, \end{aligned}$$

and that of the surface integral

$$\hat{n}_0 \cdot (p\nabla_0 g - g\nabla_0 p) = p\frac{\partial g}{\partial \hat{n}_0} - g\frac{\partial p}{\partial \hat{n}_0}.$$

Substituting these results in Eq. (2.16) Green's identity is obtained:

$$\int_V (p\nabla_0^2 g - g\nabla_0^2 p) dV(\mathbf{R}_0) = \int_S \left(p\frac{\partial g}{\partial \hat{n}_0} - g\frac{\partial p}{\partial \hat{n}_0} \right) dS(\mathbf{R}_0). \tag{2.20}$$

Using the homogeneous Helmholtz equation (2.10), for $\nabla_0^2 p$, and the inhomogeneous Helmholtz equation (2.11), for $\nabla_0^2 g$:

$$\begin{aligned} \nabla_0^2 p &= -k^2 p \\ \nabla_0^2 g &= -k^2 g - \delta(\mathbf{R} - \mathbf{R}_0), \end{aligned}$$

the volume integral in Eq. (2.20) can be developed:

$$\begin{aligned} \int_V (p\nabla_0^2 g - g\nabla_0^2 p) dV(\mathbf{R}_0) &= \int_V (-pk^2 g - p\delta(\mathbf{R} - \mathbf{R}_0) + gk^2 p) dV(\mathbf{R}_0) \\ &= \int_V -p\delta(\mathbf{R} - \mathbf{R}_0) dV(\mathbf{R}_0) = -cp = -cp(\mathbf{R}). \end{aligned}$$

Inserted in Eq. (2.20), and with the normal vector $\hat{n}_0$ reversed this leads to the Helmholtz integral equation (1.1):

$$cp(\mathbf{R}) = \int_S \left(p(\mathbf{R}_0)\frac{\partial g}{\partial \hat{n}_0} - g(|\mathbf{R} - \mathbf{R}_0|)\frac{\partial p}{\partial \hat{n}_0} \right) dS,$$

where

$$c = \begin{cases} 1, & \mathbf{R} \text{ in fluid domain } V \\ \frac{1}{2}, & \mathbf{R} \text{ on smooth boundary of fluid domain } V \\ \frac{\Omega}{4\pi}, & \mathbf{R} \text{ on nonsmooth boundary of fluid domain } V \\ & (\Omega \text{ is the solid angle}) \\ 0, & \mathbf{R} \text{ outside fluid domain } V. \end{cases}$$

Chapter 3

BEM Formulation

In the scope of this thesis two boundary elements, constant and four-node, have been developed. This chapter will show how the elements are formulated.

3.1 General

Without knowing the physical values, i.e. the pressure and the normal velocity, on the boundaries of a problem domain it is not possible to calculate the pressure field inside the domain. Therefore, the boundary values have to be determined before a pressure field can be calculated. As a result of this, BEM consists of three parts: one pre-processing part where an equation system is built up, a second part where the system is solved to obtain the boundary values, and a third post-processing part where the pressure field in a volume V can be calculated.

3.1.1 Pre-Processing

The first step in the development of a BEM formulation is to transform a continuous system into a discrete system, Fig. 3.1. This is done by discretizing the continuous Helmholtz integral equation (1.1) in order to find a system of equations from which the unknown boundary node values can be found. The boundary is divided into N elements and Eq. (1.1) is now discretized for a given node i so that Helmholtz discrete integral

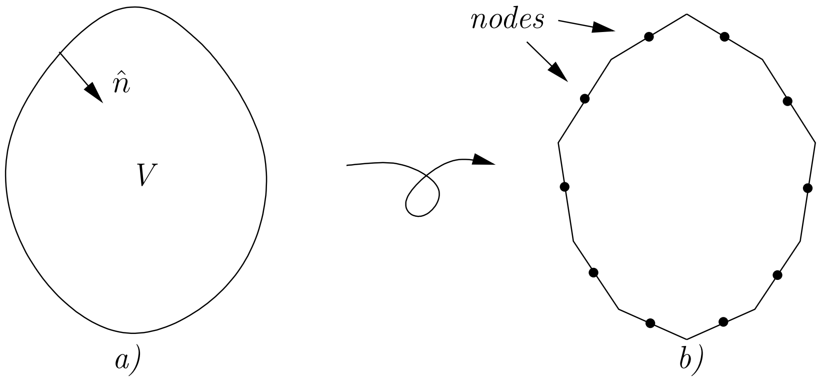


Figure 3.1: *a) continuous system, b) discrete system*

equation is obtained [5]:

$$cp_i - \sum_{j=1}^{N} \int_{S_j} p \frac{\partial g}{\partial \hat{n}} dS = - \sum_{j=1}^{N} \int_{S_j} g \frac{\partial p}{\partial \hat{n}} dS. \tag{3.1}$$

S_j is the surface of element j on the boundary. (For Fig. 3.1 $c = \frac{1}{2}$ because the nodes are placed on a smooth surface.)

For simplicity, the following expression is introduced:

$$\frac{\partial g}{\partial \hat{n}} = \bar{g},$$

so that Eq. (3.1) with use of Eq. (1.4) can be written as:

$$cp_i - \sum_{j=1}^{N} \int_{S_j} p\bar{g} dS = i\rho_0\omega \sum_{j=1}^{N} \int_{S_j} gv dS. \tag{3.2}$$

p and v are functions that originate from node values and shape functions related to surface S_j. Eq. (3.2) is repeated for every node i so that an equation system is obtained [5]. Every node must have a boundary condition, either a prescribed pressure or a prescribed normal velocity, so that the equation system can be solved. The relationship between pressure and normal velocity, Eq. (1.5), also works as a boundary condition if the pressure or the normal velocity is known at, at least, one node. Notice that only one prescribed boundary value, one of Eqs. (1.3-1.5), shall be applied to the same node, otherwise the obtained equation system can be impossible to solve.

When the boundary conditions are accurately applied the, from Eq. (3.2), constructed equation system is solved for the unknown boundary values.

3.1.2 Post-Processing

When all boundary values are known the pressure p for any point inside the fluid domain can be determined in the post-processing part. Eq. (3.2) is used again with the difference that point i is an arbitrary point located in the fluid domain and not coinciding with a boundary node. The constant c here equals one.

3.2 Constant Element

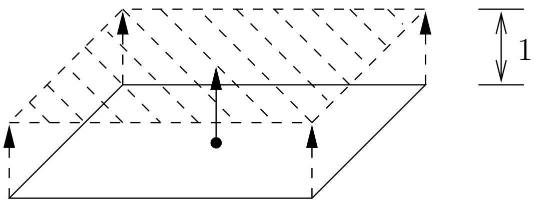


Figure 3.2: *Shape function for a constant element with one node*

As the name constant indicates, there are no magnitude variations of pressure or of normal velocity over the surface of a constant element. Therefore, just one node is necessary to describe the physical variables over an element, Fig. 3.2. The element can have an arbitrary quadrangular shape in which the node is placed in the middle and there are no limitations on the element's orientation in the three-dimensional space.

3.2.1 Pre-Processing

For a constant element, p and v in Eq. (3.2) are assumed to be constant over each element, and therefore they can be taken out of the integrals. They will be called p_j and v_j for element j. Furthermore, the node of a constant element is always placed on a smooth surface, i.e. $c = \frac{1}{2}$ (section 1.3). So,

$$\frac{1}{2}p_i - \sum_{j=1}^{N} \left(\int_{S_j} \bar{g} dS \right) p_j = i\rho_0\omega \sum_{j=1}^{N} \left(\int_{S_j} g dS \right) v_j. \tag{3.3}$$

There are now two types of integrals to be carried out over the elements, i.e. those of following types:

$$\int_{S_j} \bar{g} dS \quad \text{and} \quad \int_{S_j} g dS.$$

These integrals relate the node i, where the fundamental solution is acting, to any other node j. Because of this their resulting values are sometimes called influence coefficients. They will be called $\bar{H}_{ij}$ and G_{ij}, i.e.

$$\bar{H}_{ij} = \int_{S_j} \bar{g} dS; \qquad G_{ij} = \int_{S_j} g dS. \tag{3.4}$$

For a particular point i Eq. (3.3) now takes the form

$$\frac{1}{2} p_i - \sum_{j=1}^{N} \bar{H}_{ij} p_j = i \rho_0 \omega \sum_{j=1}^{N} G_{ij} v_j. \tag{3.5}$$

By introducing

$$H_{ij} = \frac{1}{2} \delta_{ij} - \bar{H}_{ij}, \tag{3.6}$$

where δ_{ij} is Kronecker's delta [9], Eq. (3.5) can now be written on the form

$$\sum_{j=1}^{N} H_{ij} p_j = i \rho_0 \omega \sum_{j=1}^{N} G_{ij} v_j. \tag{3.7}$$

If Eq. (3.7) is repeated for every node point i (that also varies from 1 to N) a system of equations is obtained.[5]

This set of equations can be expressed in matrix form as

$$\mathbf{H}\mathbf{p} = i \rho_0 \omega \mathbf{G}\mathbf{v}, \tag{3.8}$$

where $\mathbf{H}$ and $\mathbf{G}$ are two $N \times N$ matrices and $\mathbf{p}$ and $\mathbf{v}$ are vectors of length N. In order to determine the matrices $\mathbf{H}$ and $\mathbf{G}$ the influence coefficients from Eq. (3.4) has to be developed, starting with G_{ij}:

$$G_{ij} = \int_{S_j} g dS = \int_{S_j} g\left(|\mathbf{R}_j - \mathbf{R}_i|\right) dS = \int_{S_j} \frac{e^{-ik|\mathbf{R}_j - \mathbf{R}_i|}}{4\pi |\mathbf{R}_j - \mathbf{R}_i|} dS. \tag{3.9}$$

In the cartesian coordinate-system, Eq. (3.9) can be expressed as:

$$G_{ij} = \int_{S_j} \frac{e^{-ik\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}}}{4\pi \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}} dS. \tag{3.10}$$

This integral is evaluated by determine the product of the Green's function $g\left(|\mathbf{R}_j - \mathbf{R}_i|\right)$ and the area of element j, where $\mathbf{R}_j$ is the midpoint of element j and $\mathbf{R}_i$ is the location of node i. For the case when $i = j$, $\mathbf{R}_j$ and $\mathbf{R}_i$ coincides, the Green's function becomes singular ($G_{i(j=i)} \to \infty$). To avoid this singularity four Gauss points are used on element j, instead of the midpoint, to calculate the function value. The Gauss points are located at

$$\begin{aligned}
(\xi_1, \eta_1) &= \frac{1}{\sqrt{3}}(1, -1) \\
(\xi_2, \eta_2) &= \frac{1}{\sqrt{3}}(1, 1) \\
(\xi_3, \eta_3) &= \frac{1}{\sqrt{3}}(-1, 1) \\
(\xi_4, \eta_4) &= \frac{1}{\sqrt{3}}(-1, -1),
\end{aligned} \tag{3.11}$$

in the isoparametric $\xi\eta$-coordinate system [10]. G_{ij} is then chosen as the mean value of $G_{ij(\xi_1,\eta_1)}$, $G_{ij(\xi_2,\eta_2)}$, $G_{ij(\xi_3,\eta_3)}$, and $G_{ij(\xi_4,\eta_4)}$.

$\bar{H}_{ij}$ is a bit trickier to determine because of the derivative $\frac{\partial g}{\partial \hat{n}}$ that has to be carried out, i.e.

$$\bar{H}_{ij} = \int_{S_j} \bar{g} dS = \int_{S_j} \frac{\partial g}{\partial \hat{n}} dS = \int_{S_j} \left(\nabla g\right)^T \hat{\mathbf{n}} dS, \tag{3.12}$$

where

$$\hat{\mathbf{n}} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}, \tag{3.13}$$

and

$$\nabla g = \begin{bmatrix} -\frac{x e^{-ik\sqrt{x^2+y^2+z^2}}}{4\pi(x^2+y^2+z^2)} \left(ik + \frac{1}{\sqrt{x^2+y^2+z^2}}\right) \\ -\frac{y e^{-ik\sqrt{x^2+y^2+z^2}}}{4\pi(x^2+y^2+z^2)} \left(ik + \frac{1}{\sqrt{x^2+y^2+z^2}}\right) \\ -\frac{z e^{-ik\sqrt{x^2+y^2+z^2}}}{4\pi(x^2+y^2+z^2)} \left(ik + \frac{1}{\sqrt{x^2+y^2+z^2}}\right) \end{bmatrix}. \tag{3.14}$$

Observe that $(x_j - x_i)$, $(y_j - y_i)$, and $(z_j - z_i)$ has been substituted with x, y, and z in Eq. (3.14) above. The value of $\bar{H}_{ij}$ is now determined by calculating the product of $(\nabla g)^T \hat{\mathbf{n}}$ and the area of surface S_j (compare with the determination of G_{ij}).

H_{ij} is equal with $\bar{H}_{ij}$ when $i \neq j$, Eq. (3.6). When $i = j$ the $\bar{H}_{ii}$ terms are identically zero since the normal $\hat{n}$ and the element coordinates are perpendicular to each other [5]:

$$\bar{H}_{ii} = \int_{S_i} \frac{\partial g}{\partial \hat{n}} dS = \int_{S_i} \frac{\partial g}{\partial r} \frac{\partial r}{\partial \hat{n}} dS \equiv 0, \tag{3.15}$$

Therefore; $H_{ii} = \frac{1}{2}$.

3.2.2 Post Processing

After the pre-processing part, pressure and normal velocity on all boundary elements can be calculated. When all boundary values are known the pressure p can be determined at an arbitrary point inside volume V. Consequently, Eq. (1.1) will be written as

$$p(\mathbf{R}) = \int_S \left(p \frac{\partial g}{\partial \hat{n}_0} - g \frac{\partial p}{\partial \hat{n}_0} \right) dS(\mathbf{R}_0),$$

or in discretized form

$$p_i - \sum_{j=1}^{N} \int_{S_j} p \frac{\partial g}{\partial \hat{n}} dS = - \sum_{j=1}^{N} \int_{S_j} g \frac{\partial p}{\partial \hat{n}} dS. \tag{3.16}$$

In similarity with section 3.2.1 Eq. (3.16) is now rewritten as

$$p_i = \sum_{j=1}^{N} \bar{H}_{ij} p_j + i\rho_0\omega \sum_{j=1}^{N} G_{ij} v_j, \tag{3.17}$$

and p_i can be calculated.

3.3 Four-node Linear Elements

For a four-node isoparametric quadrilateral element, the pressure p and the normal velocity v at any position on the element can be defined by

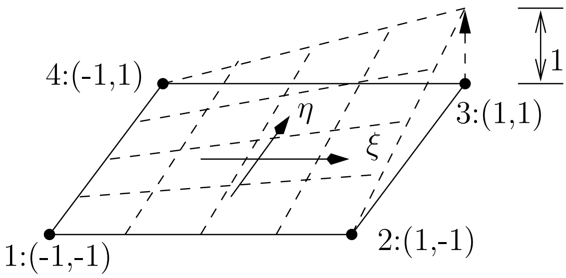


Figure 3.3: *Shape function N_3 for a four-node element*

their nodal values and linear shape functions [5], i.e.

$$
\begin{aligned}
v(\xi,\eta) &= N_1 v_1 + N_2 v_2 + N_3 v_3 + N_4 v_4 \\
&= \begin{bmatrix} N_1 & N_2 & N_3 & N_4 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}, \\
p(\xi,\eta) &= N_1 p_1 + N_2 p_2 + N_3 p_3 + N_4 p_4 \\
&= \begin{bmatrix} N_1 & N_2 & N_3 & N_4 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}, \qquad (3.18)
\end{aligned}
$$

where the shape functions are

$$
\left.\begin{array}{lcl}
N_1 &=& \frac{1}{4}(\xi-1)(\eta-1), \\
N_2 &=& -\frac{1}{4}(\xi+1)(\eta-1), \\
N_3 &=& \frac{1}{4}(\xi+1)(\eta+1), \\
N_4 &=& -\frac{1}{4}(\xi-1)(\eta+1),
\end{array}\right\} \qquad (3.19)
$$

in the $\xi\eta$-coordinate system. Shape function N_3 is shown in Fig. 3.3 [11]. As for the constant element, the four-node element can have an arbitrary quadrangular shape and there are no limitations on the element's orientation in the three-dimensional space.

3.3.1 Pre-Processing

With help of the shape functions, Eq. (3.19), the integral on the left hand side in Eq. (3.2) can be written, considered over one element j, as

$$\int_{S_j} p\bar{g}dS = \int_{S_j} \begin{bmatrix} N_1 & N_2 & N_3 & N_4 \end{bmatrix} \bar{g}dS \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}$$
$$= \begin{bmatrix} \bar{h}^1_{ij} & \bar{h}^2_{ij} & \bar{h}^3_{ij} & \bar{h}^4_{ij} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}, \quad (3.20)$$

where the four influence terms

$$\bar{h}^1_{ij}, .., .., \bar{h}^4_{ij} = \int_{s_j} N_1\bar{g}dS, .., .., \int_{s_j} N_4\bar{g}dS, \quad (3.21)$$

has to be determined [5]. Similarly we have for the right hand side in Eq. (3.2)

$$g^1_{ij}, .., .., g^4_{ij} = \int_{S_j} N_1 g dS, .., .., \int_{S_j} N_4 g dS. \quad (3.22)$$

The integrals in Eq. (3.21) and Eq. (3.22) are carried out by the use of Gauss points. For each element four points are used. The location of the points are shown in Eq. (3.11). The integrals can now be calculated through summations. How the summations are carried through will be explained in section 9.1. Substituting Eq. (3.20) and its corresponding equation for the right hand side, into Eq. (3.2) for all elements j, the following equation for node i is obtained

$$c_i p_i - \sum_{j=1}^{N} \begin{bmatrix} \bar{h}^{n_1}_{ij} & \bar{h}^{n_2}_{ij} & \bar{h}^{n_3}_{ij} & \bar{h}^{n_4}_{ij} \end{bmatrix} \begin{bmatrix} p_{n_1} \\ p_{n_2} \\ p_{n_3} \\ p_{n_4} \end{bmatrix}$$
$$= i\rho_0\omega \sum_{j=1}^{N} \begin{bmatrix} g^{n_1}_{ij} & g^{n_2}_{ij} & g^{n_3}_{ij} & g^{n_4}_{ij} \end{bmatrix} \begin{bmatrix} v_{n_1} \\ v_{n_2} \\ v_{n_3} \\ v_{n_4} \end{bmatrix}, \quad (3.23)$$

where $n_1 - n_4$ refers to the nodes that corresponds to element j. Since the nodes might be located on non-smooth surfaces the constant c_i can take values between 0 and 1 depending on the solid space angle Ω (section 1.3). In similarity with Eq. (3.7), Eq. (3.23) can be rewritten as

$$\sum_{j=1}^{N} \begin{bmatrix} h_{ij}^{n_1} & h_{ij}^{n_2} & h_{ij}^{n_3} & h_{ij}^{n_4} \end{bmatrix} \begin{bmatrix} p_{n_1} \\ p_{n_2} \\ p_{n_3} \\ p_{n_4} \end{bmatrix} = i\rho_0\omega \sum_{j=1}^{N} \begin{bmatrix} g_{ij}^{n_1} & g_{ij}^{n_2} & g_{ij}^{n_3} & g_{ij}^{n_4} \end{bmatrix} \begin{bmatrix} v_{n_1} \\ v_{n_2} \\ v_{n_3} \\ v_{n_4} \end{bmatrix}. \quad (3.24)$$

If Eq. (3.24) is repeated for all nodes the whole system in matrix form becomes

$$\mathbf{Hp} = i\rho_0\omega\mathbf{Gv} \quad (3.25)$$

where $\mathbf{H}$ and $\mathbf{G}$ are two (*number of nodes*)$\times$(*number of nodes*) matrices and $\mathbf{p}$ and $\mathbf{v}$ are vectors of length (*number of nodes*).

3.3.2 Post-Processing

The post-processing is carried through in similar manner to constant elements in section 3.2.2.

Chapter 4

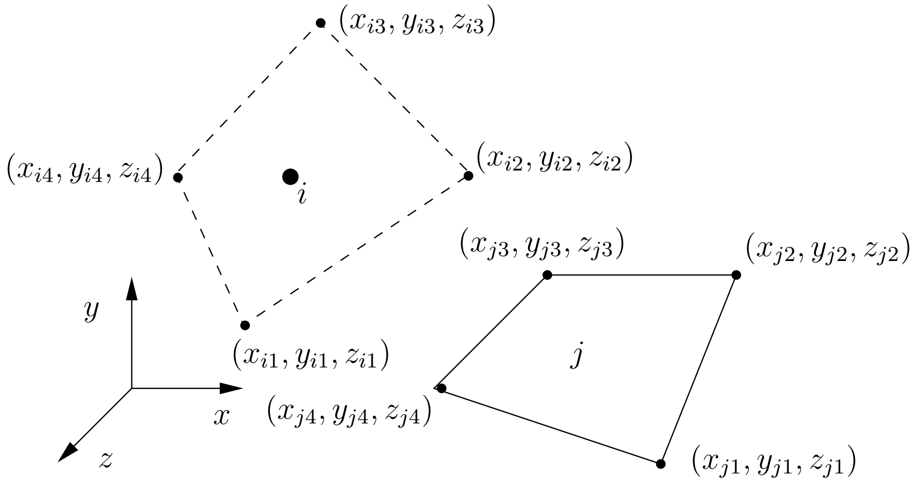
Implemented BEM Functions

Based on the BEM formulation for constant and linear elements, a number of functions have been developed and implemented in order to solve BEM problems for time-harmonic acoustic fluid domains. The programs will be described below and are included in appendix A.

4.1 Bem_infl1q

Purpose:

Compute influence coefficients for a three dimensional quadrilateral constant acoustic boundary element.



Syntax:

[He,Ge]=bem_infl1q(ex,ey,ez,ep,n)
[He,Ge]=bem_infl1q(ex,ey,ez,ep)

Description:

bem_infl1q provides the influence coefficients He and Ge for a constant three dimensional acoustic boundary element j that are influencing a node i located at R_i.
The input variables

$$\mathsf{ex} = \begin{bmatrix} x_{i1} & x_{i2} & x_{i3} & x_{i4} \\ x_{j1} & x_{j2} & x_{j3} & x_{j4} \end{bmatrix}$$

$$\mathsf{ey} = \begin{bmatrix} y_{i1} & y_{i2} & y_{i3} & y_{i4} \\ y_{j1} & y_{j2} & y_{j3} & y_{j4} \end{bmatrix} \qquad \mathsf{ep} = [\ w \quad c \quad rho\ ]$$

$$\mathsf{ez} = \begin{bmatrix} z_{i1} & z_{i2} & z_{i3} & z_{i4} \\ z_{j1} & z_{j2} & z_{j3} & z_{j4} \end{bmatrix}$$

supply the element corner coordinates (where the node i is located on the element midpoint, R_i, in the isoparametric $\xi\eta$-coordinate system), the angular frequency w, the speed of sound in the acoustic medium c, and the density of the acoustic medium rho. n is described in the theory part.

Theory:

A closed boundary surface fulfils the Helmoltz integral equation which in discrete form can be written as

$$\frac{1}{2}p_i - \sum_{j=1}^{N} \int_{S_j} p \frac{\partial g}{\partial \hat{n}} dS = i\rho_0\omega \sum_{j=1}^{N} \int_{S_j} gvdS,$$

in which p_i is the pressure at node i, p and v are pressure and normal velocity distributions over the boundary surface, N is the number of boundary elements. Since p and v are constant over an element the equation above can be written as

$$\frac{1}{2}p_i - \sum_{j=1}^{N} \left(\int_{S_j} \frac{\partial g}{\partial \hat{n}} dS \right) p_j = i\rho_0\omega \sum_{j=1}^{N} \left(\int_{S_j} gdS \right) v_j$$

where the element influence coefficients He and Ge are computed according to

$$\mathsf{He} = -\int_{S_j} (\nabla g)^T \hat{\mathbf{n}} dS, \qquad \text{if} \quad i \neq j$$

$$\mathsf{He} = \frac{1}{2}, \qquad \text{if} \quad i = j$$

$$\mathsf{Ge} = i\rho_0\omega \int_{S_j} g dS,$$

where g is the free-space Green's function

$$g = \frac{e^{-ik|R_j - R_i|}}{4\pi|R_j - R_i|}. \qquad \left(k = \frac{w}{c}\right)$$

$|R_j - R_i|$ is a function of the distance between element j and node i and $\hat{\mathbf{n}}$ is the normal unit vector of element surface j pointing into the fluid domain. For element corner coordinates numbered counter clockwise the vector will be pointing out of surface j which is the default normal direction. For reversed normal direction the input variable n has to be prescribed with the value -1. For default direction n=1.

4.2 Bem_infl4q

Purpose:

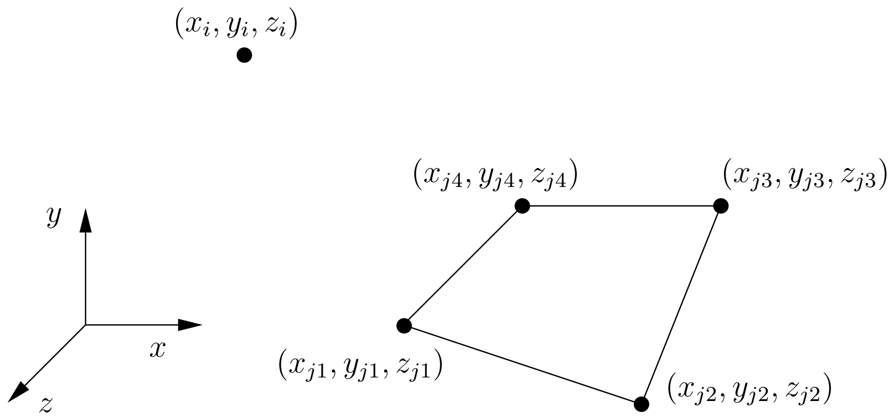
Compute influence vectors for a three dimensional four-node quadrilateral linear acoustic boundary element.

Syntax:

```
[He,Ge]=bem_infl4q(coord,ex,ey,ez,ep,n)
[He,Ge]=bem_infl4q(coord,ex,ey,ez,ep)
```

Description:

bem_infl4q provides the influence vectors He and Ge for a four-node linear three dimensional acoustic boundary element j,



that is influencing a node located at $R_i = (x_i, y_i, z_i)$.
The input variables

$$\text{coord} = [\ x_i \quad y_i \quad z_i\ ] \qquad \begin{aligned} \text{ex} &= [\ x_{j1} \quad x_{j2} \quad x_{j3} \quad x_{j4}\ ] \\ \text{ey} &= [\ y_{j1} \quad y_{j2} \quad y_{j3} \quad y_{j4}\ ] \\ \text{ez} &= [\ z_{j1} \quad z_{j2} \quad z_{j3} \quad z_{j4}\ ] \end{aligned}$$

supply the influenced nodes coordinates coord, and the influencing element's node coordinates ex, ey, ez. The input variables ep and n are explained in bem_infl1q.

Theory:

With the discrete Helmholtz integral equation as a starting point

$$c_i p_i - \sum_{j=1}^{N} \int_{S_j} p \frac{\partial g}{\partial \hat{n}} dS = i \rho_0 \omega \sum_{j=1}^{N} \int_{S_j} g v dS$$

where c_i is computed in bem_spacang, and p_i, p, v and N are explained in bem_infl1q. For a four node element this can be written as

$$c_i p_i - \int_{S_j} [\ N_1 \quad N_2 \quad N_3 \quad N_4\ ] \frac{\partial g}{\partial \hat{n}} dS \begin{bmatrix} p_{j1} \\ p_{j2} \\ p_{j3} \\ p_{j4} \end{bmatrix}$$
$$= i\rho_0\omega \int_{S_j} [\ N_1 \quad N_2 \quad N_3 \quad N_4\ ] gvdS \begin{bmatrix} v_{j1} \\ v_{j2} \\ v_{j3} \\ v_{j4} \end{bmatrix}$$

where the element influence vectors He and Ge are computed according to

$$\mathsf{He} = -\int_{S_j} [N_1 \quad N_2 \quad N_3 \quad N_4]\ (\nabla g)^T \hat{\mathbf{n}} dS$$
$$\mathsf{Ge} = i\rho_0\omega \int_{S_j} [N_1 \quad N_2 \quad N_3 \quad N_4]\ gdS$$

where N_1, N_2, N_3, and N_4 are shape functions,

$$N_1 = \frac{1}{4}(\xi - 1)(\eta - 1), \qquad N_2 = -\frac{1}{4}(\xi + 1)(\eta - 1),$$
$$N_3 = \frac{1}{4}(\xi + 1)(\eta + 1), \qquad N_4 = -\frac{1}{4}(\xi - 1)(\eta + 1).$$

$\hat{\mathbf{n}}$ and g are explained in bem_infl1q.

4.3 Bem_spacang

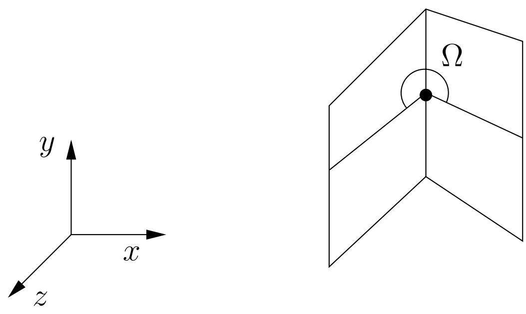
Purpose:

Compute the space angle constant for a node on a non-smooth surface.

Syntax:

Ce=bem_spacang(coord,ex,ey,ez)

Description:



bem_spacang provides the space angle constant Ce, which is the quotient of $\frac{\Omega}{4\pi}$, for a node coinciding with three or four element corners. Ω is the space angle towards the acoustic medium, and the space angle for a sphere is 4π. For a smooth surface the space angle is 2π, and $\mathsf{Ce} = \frac{1}{2}$. bem_spacang can only provide the space angle constant if the angle α between two adjacent elements is $\frac{\pi}{2} \leq \alpha \leq \frac{3\pi}{2}$.
The input variables

$$
\mathsf{coord} = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix} \qquad
\begin{array}{l}
\mathsf{ex} = \begin{bmatrix} x_{j1} & x_{j2} & x_{j3} & x_{j4} \\ \cdot & \cdot & \cdot & \cdot \\ x_{jn} & x_{jn} & x_{jn} & x_{jn} \end{bmatrix} \\
\mathsf{ey} = \begin{bmatrix} y_{j1} & y_{j2} & y_{j3} & y_{j4} \\ \cdot & \cdot & \cdot & \cdot \\ y_{jn} & y_{jn} & y_{jn} & y_{jn} \end{bmatrix} \\
\mathsf{ez} = \begin{bmatrix} z_{j1} & z_{j2} & z_{j3} & z_{j4} \\ \cdot & \cdot & \cdot & \cdot \\ z_{jn} & z_{jn} & z_{jn} & z_{jn} \end{bmatrix}
\end{array}
$$

supply the coordinate of the node where the space angle constant is to be calculated x_i, y_i, and z_i, and the element corner coordinates ex, ey, and ez in which n is the number of elements.

Theory:

In the discrete Helmholtz equation for a four node element

$$c_i p_i - \int_{S_j} [\ N_1 \quad N_2 \quad N_3 \quad N_4\ ] \frac{\partial g}{\partial \hat{n}} dS \begin{bmatrix} p_{j1} \\ p_{j2} \\ p_{j3} \\ p_{j4} \end{bmatrix}$$

$$= i\rho_0\omega \int_{S_j} [\ N_1 \quad N_2 \quad N_3 \quad N_4\ ] g v dS \begin{bmatrix} v_{j1} \\ v_{j2} \\ v_{j3} \\ v_{j4} \end{bmatrix}$$

Ce denotes the constant c_i. For a smooth surface $c_i = \frac{1}{2}$.

4.4 Bem_assem

Purpose:

Assemble acoustic element matrices.

Syntax:

P=bem_assem(edof,P,Pe,n,el)

Description:

bem_assem adds the element influence matrix Pe to the global boundary influence matrix P according to the topology matrix edof.

The topology matrix edof is defined as

$$\text{edof} = \left.\begin{bmatrix} el_1 & eln_1 & eln_2 & . & . & . & eln_{nen} \\ el_2 & eln_1 & eln_2 & . & . & . & eln_{nen} \\ \vdots & \vdots & \vdots & & & & \vdots \\ el_{nel} & eln_1 & eln_2 & . & . & . & eln_{nen} \end{bmatrix}\right\} \begin{array}{l} \textit{one row for} \\ \textit{each element} \end{array}$$

where the first column contains the element numbers, and the columns 2 to (nen+1) contain the corresponding global node numbers (nen = number of element nodes, eln = global element node number). For bem_infl1q elements nen=1 and for bem_infl4q elements nen=4.

The input variables n and el denote the number of the influenced node and the number of the influencing element respectively.

4.5 Bem_solveq

Purpose:

Solve BEM equation system.

Syntax:

[pr,nv]=bem_solveq(G,H,bcpr,bcnv,bcim)

Description:

bem_solveq solves the equation system

$$\mathsf{H}\ \mathsf{pr} = \mathsf{G}\ \mathsf{nv}$$

for acoustic problems where H and G are squared matrices which values are created in bem_infl1q or bem_infl4q. If bem_infl4q is used a diagonal matrix **C** must be added to H so that

$$\mathsf{H} = \mathsf{H} + \mathbf{C}.$$

C can be created with use of bem_spacang.
The output variables pr and nv are vectors providing the node boundary pressure and the node normal velocity.
The input variables

$$\mathsf{bcpr} = \begin{bmatrix} nn_1 & p_1 \\ nn_2 & p_2 \\ \vdots & \vdots \\ nn_n & p_{nbc} \end{bmatrix} \qquad \mathsf{bcim} = \begin{bmatrix} nn_1 & z_1 \\ nn_2 & z_2 \\ \vdots & \vdots \\ nn_n & z_n \end{bmatrix}$$

$$\mathsf{bcnv} = \begin{bmatrix} nn_1 & v_1 \\ nn_2 & v_2 \\ \vdots & \vdots \\ nn_n & v_{nbc} \end{bmatrix}$$

contains prescribed boundary conditions where nn denotes the prescribed node number and pressure p, normal velocity v, and impedance z are prescribed physical properties. Notice that, in order to obtain a solution, no node can have less or more than one prescribed physical property.

4.6 Bem_acouspost

Purpose:

Compute the acoustic pressure at an arbitrary point in a 3D fluid domain.

Syntax:

p=bem_acouspost(coord,ex,ey,ep,pr,nv,edof,n)
p=bem_acouspost(coord,ex,ey,ep,pr,nv,edof)

Description:

bem_acouspost provides the, from a radiating boundary, induced acoustic pressure p at an arbitrary point in a three dimensional fluid domain.
The input parameters

$$\text{coord} = \begin{bmatrix} x & y & z \end{bmatrix} \qquad \text{ex} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & x_{n4} \end{bmatrix} \quad \text{ey} = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} \\ \vdots & \vdots & \vdots & \vdots \\ y_{n1} & y_{n2} & y_{n3} & y_{n4} \end{bmatrix} \quad \text{ez} = \begin{bmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ \vdots & \vdots & \vdots & \vdots \\ z_{n1} & z_{n2} & z_{n3} & z_{n4} \end{bmatrix}$$

supply the coordinates where the pressure is to be calculated, x, y, and z, and the coordinates for the radiating elements. pr and nv are vectors containing pressure and normal velocity at boundary nodes. The input variables ep and n are explained in bem_infl1q. Input matrix edof is explained in bem_assem with the difference that all elements must be represented in edof when used in bem_acouspost.
If both constant and linear elements are used in a BE problem then the rows in the edof matrix, that provides the node number for constant elements must have the same length as rows that provides the node numbers for linear elements. To do this, constant element edof rows are given the value 0 for column positions 3 to 5.

Theory:

The acoustic pressure p is calculated according to the discrete Helmholtz integral equation.

$$p - \sum_{j=1}^{N} \int_{S_j} p \frac{\partial g}{\partial \hat{n}} dS = i\rho_0\omega \sum_{j=1}^{N} \int_{S_j} gvdS$$

For further details see bem_infl1q and bem_infl4q.

Chapter 5

Using BEM

The sections below should be considered to give a better insight in how the BE functions described in chapter 4 should be used.

5.1 Symmetry

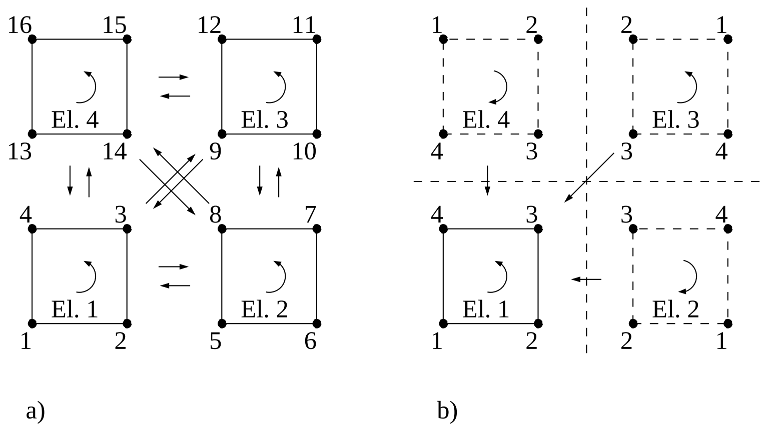


Figure 5.1: *a) no symmetry b) two symmetry lines*

To shorten the calculation time of BE problems, symmetry can be used in the pre-processing stage if suitable. In similarity with FEM, the calculation effort is halved with every symmetry line or plane. Symmetry is quite simple to implement in FE problems because the values in the stiffness, damping, and mass matrices for one element are functions depending on that element only. For BE problems, use of symmetry is more difficult since the values in the influence matrices, **H** and **G**, depend on the entire boundary surface (chapter 3) and not on single elements. So, even if symmetry is used the geometry for the entire boundary surface has to be modelled.

Consider Fig. 5.1 a) with the illustrative non symmetric BE problem. The problem consists of four linear elements with 16 nodes. As indicated by the arrows element 1 to 4 are all influencing each other (and themselves) and their nodes are orientated in a counter clockwise fashion. The orientation of the nodes determine the direction of the element normal vectors that must be pointing into the fluid domain. A counter clockwise orientation gives a normal direction out of the paper.

Fig. 5.1 b) shows the same problem with two symmetry lines. The problem still consists of four elements but the number of nodes has decreased from 16 to 4. Instead of letting all elements influence each other, and themselves, the only influencing allowed is the one from symmetry elements 2, 3, and 4 towards element 1 and the self influence of element 1.

When symmetry is used the orientation of the nodes change, which has happened for the nodes in element 2 and 4, Fig. 5.1 b). A clockwise orientation gives a normal vector that are pointing into the paper and out of the fluid domain. This causes a problem because the normal vector for an element must be directed into the fluid domain. To avoid this problem one has to give element 2 and 4 a normal vector that is pointing into the fluid domain, even if the orientation of the nodes is telling otherwise. (How the default normal direction for function bem_infl1q and bem_infl4q are reversed, see section 4.1 and 4.2.)

How node 1 is influenced by the elements can be shown in the principle equations below. For the non symmetric case we have

$$\begin{aligned} N_1^{ns} = I_{11}^1 f_1 + I_{12}^1 f_2 + I_{13}^1 f_3 + I_{14}^1 f_4 + I_{15}^2 f_5 + I_{16}^2 f_6 \\ + I_{17}^2 f_7 + I_{18}^2 f_8 + I_{19}^3 f_9 + I_{110}^3 f_{10} + I_{111}^3 f_{11} \\ + I_{112}^3 f_{12} + I_{113}^4 f_{13} + I_{114}^4 f_{14} + I_{115}^4 f_{15} + I_{116}^4 f_{16} \end{aligned} \quad (5.1)$$

where I_{1n}^{El} are influencing parameters and f_n are physical properties for the nodes. The same equation for the symmetric case will be written as

$$\begin{aligned} N_1^{s} = I_{11}^1 f_1 + I_{12}^1 f_2 + I_{13}^1 f_3 + I_{14}^1 f_4 + I_{12}^2 f_2 + I_{11}^2 f_1 \\ + I_{14}^2 f_4 + I_{13}^2 f_3 + I_{13}^3 f_3 + I_{14}^3 f_4 + I_{11}^3 f_1 \\ + I_{12}^3 f_2 + I_{14}^4 f_4 + I_{13}^4 f_3 + I_{12}^4 f_2 + I_{11}^4 f_1 \end{aligned} \quad (5.2)$$

or

$$N_1^s = I_{11}^{1+2+3+4} f_1 + I_{12}^{1+2+3+4} f_2 + I_{13}^{1+2+3+4} f_3 + I_{14}^{1+2+3+4} f_4. \qquad (5.3)$$

5.2 Convergence

5.2.1 Mesh Size

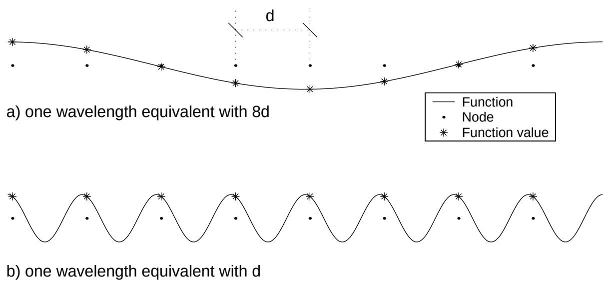


Figure 5.2: *Different wavelengths*

To obtain reliable results, it is important that the boundary mesh is not too course. As illustrated in Fig. 5.2 above, a time harmonic function describes the physical properties over the boundary surface. Because BEM is a numerical method, only a limited number of function values is calculated. If the distances between the nodes are too long in relation to the wavelength, the calculated function values will give a poor estimation of the "true" relationship, Fig. 5.2 b).

For one-node constants and four-node linear boundary elements, convergence is obtained if $\lambda \leq 8d$ where λ is the wavelength and d is the distance between two nodes. For four-node elements the limit can be stretched to $\lambda \leq 4d$ if they are carefully used.

5.2.2 Duplication of Nodes

A commonly used tool to improve convergence rate is duplication of nodes. This is done at edges and corners of the boundary surface, where the normal direction is not uniquely defined and therefore makes the calculation of the normal velocities more sensitive, Fig. 5.3 a). Duplication is also done at the intersecting curve between elements with different boundary conditions, Fig. 5.3 b). [7]

For elements sharing a node at a corner or at an edge, duplication is not necessary if the elements are prescribed a constant normal velocity as a boundary condition.

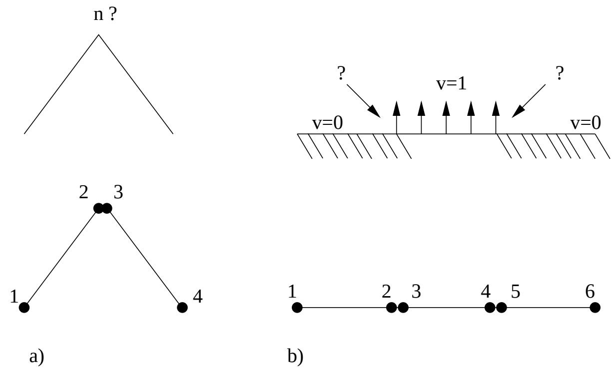


Figure 5.3: *a) Duplication at edges and corners b) Duplication at boundary condition discontinuities*

5.2.3 Numerical Difficulties

Numerical difficulties sometimes occur as a result of badly conditioned influence matrices [13]. If such problems arise when symmetry is used it can sometimes be remedied by removing one or more symmetry line/plane.

5.3 Example with Pulsating Sphere

To verify the correctness of the developed BEM functions described in chapter 4 they are used to examine a pulsating sphere. The exact solution for the acoustic pressure p at a distance r from the center of a sphere of radius a pulsating with uniform radial velocity U_a is:

$$p(r) = \frac{a}{c} U_a Z_0 \frac{ika}{1 + ika} e^{-ik(r-a)}, \tag{5.4}$$

where Z_0 is the acoustic characteristic impedance of the medium and k is the wave number [5].

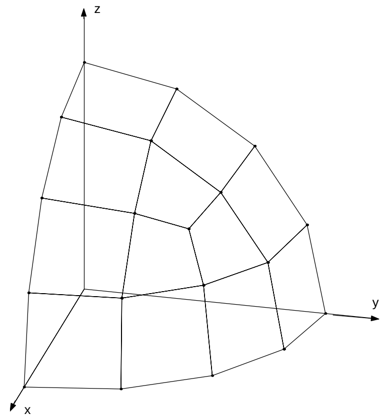


Figure 5.4: *One octant of a pulsating sphere*

Two different meshes are used, constant quadrilateral elements (bem_infl1q) and linear quadrilateral elements (bem_infl4q). The element geometry is shown in figure 5.4.
With U_a, Z_0, and radius r set to unity a comparison between the BEM results and the analytical solution is given in table 5.3. As can be read

	Analytical Solution		Constant Element		Linear Element	
k	real	imag	real	imag	real	imag
0.1	0.0099	0.0990	0.0100	0.0988	0.0096	0.0983
0.5	0.2	0.4	0.2026	0.3998	0.1965	0.4009
1	0.5	0.5	0.5139	0.4996	0.5001	0.5095
2	0.8	0.4	0.8664	0.4161	0.8318	0.3804
4	0.9412	0.2353	0.8506	0.0320	0.8842	0.2452

Table 5.1: *BEM results for pulsating sphere (a=r)*

from the table both constant and linear elements provides a good approximation of the acoustic pressure for wave numbers ≤ 2. For $k = 4$ only linear elements can offer a trustworthy solution.

For this example the relationship between the wavelength, λ, and the node spacing, d, for wave number 2 and 4 is $\lambda \approx 8d$ and $\lambda \approx 4d$ respectively, which refer to the discussion in section 5.2.1.

See appendix B for input to the problem example. Notice that three symmetry planes are used and that no duplication of nodes (section 5.2.2) is necessary since all normal velocities are the same.

Chapter 6

Coupled BEM/FEM

In this chapter it will be explained how the BE method can be connected with the FE method for structure-acoustic applications. The, in the following sections, discussed boundary element is bem_infl4q (section 4.2), and the discussed finite element is a four node quadrilateral shell element with 24 degrees of freedom (dof).

This chapter is mostly based on *Boundary element method in acoustics* [10].

6.1 Coupling Relationship

As established in chapter 3, a BE model has the form

$$\mathbf{H} \cdot \mathbf{p} = i\rho_0\omega\mathbf{G} \cdot \mathbf{v}.$$

For an elastic shell structure the resulting FE model can be represented in the following way,

$$\left(\mathbf{K} + i\omega\mathbf{C} - \omega^2\mathbf{M}\right)\mathbf{w} = \mathbf{F}. \tag{6.1}$$

Where $\mathbf{K}$, $\mathbf{M}$ and $\mathbf{C}$ are structural stiffness, mass and damping matrices. The BE model that is shown in Fig. 6.1 is divided into two parts, a and b.

The boundary elements in part a are connected with the finite elements that represent an elastic shell structure. Notice that the BE nodes on part a must coincide with FE nodes. The node pressure values, $\mathbf{p}$, from the BE model are sorted in two groups, $\mathbf{p}_a$ and $\mathbf{p}_b$, depending on

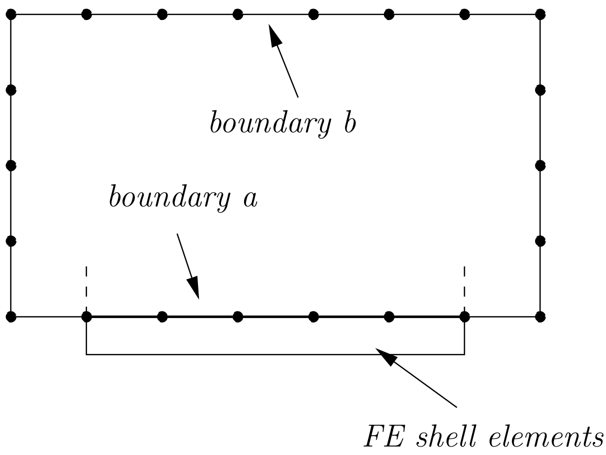


Figure 6.1: *FE structure coupled with boundary elements*

which boundary they are acting on. The same holds for the node normal velocities, $\mathbf{v}$. The BE model can now be rewritten as

$$\begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{p}_a \\ \mathbf{p}_b \end{bmatrix} = i\rho_0\omega \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{v}_a \\ \mathbf{v}_b \end{bmatrix}. \tag{6.2}$$

The force loading of the acoustic pressure, $\mathbf{p}_a$, on the elastic shell structure along the fluid-structure coupling interface in an interior or exterior coupled structure-acoustic system may be regarded as an additional normal load in the resulting FE model, Eq. (6.1), that leads to the modified FE model below,

$$\left(\mathbf{K} + i\omega\mathbf{C} - \omega^2\mathbf{M}\right)\mathbf{w} + \mathbf{L}\cdot\mathbf{p}_a = \mathbf{F}, \tag{6.3}$$

where $\mathbf{L}$ is a coupling matrix of size $m \times n$, where m is the number of FE degrees of freedom and n is the number of BE nodes on the coupled boundary a, which composition will be discussed later in section 6.2.

Regarding the normal fluid velocities and the normal shell translations at the fluid-structure coupling interface a relationship has to be established considering the velocity continuity over the coinciding nodes:

$$\mathbf{v}_a = i\omega\left(\mathbf{T}\cdot\mathbf{w}\right). \tag{6.4}$$

In similarity with $\mathbf{L}$, $\mathbf{T}$ $(n \times m)$ is also a coupling matrix that will be discussed later in section 6.3. With use of the expression above Eq. (6.2) takes the modified form

$$\begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{p}_a \\ \mathbf{p}_b \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} -\rho_0\omega^2(\mathbf{T}\cdot\mathbf{w}) \\ i\rho_0\omega\mathbf{v}_b \end{bmatrix}. \tag{6.5}$$

Combining the modified structural FE model, Eq. (6.3), with the modified acoustic BE model, Eq. (6.5), yields the coupled FE/BE model

$$\begin{bmatrix} \mathbf{K}+i\omega\mathbf{C}-\omega^2\mathbf{M} & \mathbf{L} & \mathbf{0} \\ \rho_0\omega^2\mathbf{G}_{11}\mathbf{T} & \mathbf{H}_{11} & \mathbf{H}_{12} \\ \rho_0\omega^2\mathbf{G}_{21}\mathbf{T} & \mathbf{H}_{12} & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{p}_a \\ \mathbf{p}_b \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{F}_a \\ \mathbf{F}_b \end{bmatrix} \tag{6.6}$$

with

$$\mathbf{F}_a = i\rho_0\omega\mathbf{G}_{12}\mathbf{v}_b, \tag{6.7}$$

$$\mathbf{F}_b = i\rho_0\omega\mathbf{G}_{22}\mathbf{v}_b. \tag{6.8}$$

6.2 Pressure Coupling Matrix, L

The global coupling matrix $\mathbf{L}$ transform the fluid pressure into point forces that act on the nodes of the shell structure, for the entire interface surface a (Fig. 6.1). $\mathbf{L}$ consists of n assembled local transformation matrices $\mathbf{L}_e$, Eq. (6.9), where n is the number of finite or boundary elements on surface a. Each local transformation matrix has the size $dof_{FE} \times dof_{BE}$ (dof_{FE} is the number of degrees of freedom for a finite element and dof_{BE} is the corresponding number for a boundary element) and is determined through the operation

$$\mathbf{L}_e = \int_{S_e} \mathbf{N}_F^T \mathbf{n} \mathbf{N}_B dS, \tag{6.9}$$

in which $\mathbf{N}_F$ is the shape function matrix for the finite element,

$$\mathbf{N}_F = \begin{bmatrix} \mathbf{N}_1 & 0 & 0 & \mathbf{N}_2 & 0 & 0 & \mathbf{N}_3 & 0 & 0 & \mathbf{N}_4 & 0 & 0 \\ 0 & \mathbf{N}_1 & 0 & 0 & \mathbf{N}_2 & 0 & 0 & \mathbf{N}_3 & 0 & 0 & \mathbf{N}_4 & 0 \\ 0 & 0 & \mathbf{N}_1 & 0 & 0 & \mathbf{N}_2 & 0 & 0 & \mathbf{N}_3 & 0 & 0 & \mathbf{N}_4 \end{bmatrix}$$

where

$$\begin{aligned} \mathbf{N}_1 &= [\ N_1 \ \ 0 \ \ 0 \ \ 0\ ] \\ \mathbf{N}_2 &= [\ N_2 \ \ 0 \ \ 0 \ \ 0\ ] \\ \mathbf{N}_3 &= [\ N_3 \ \ 0 \ \ 0 \ \ 0\ ] \\ \mathbf{N}_4 &= [\ N_4 \ \ 0 \ \ 0 \ \ 0\ ], \end{aligned}$$

so that

$$\mathbf{N}_F = \begin{bmatrix} N_1 & 0 & 0 & 0 & 0 & 0 & N_2 & \cdots & 0 & 0 & 0 & 0 \\ 0 & N_1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & N_1 & 0 & 0 & 0 & 0 & \cdots & N_4 & 0 & 0 & 0 \end{bmatrix}.$$

Notice that the rotational parts in $\mathbf{N}_F$ are neglected since rotational effects are small in comparison with translational ones in the coupling between BEM and FEM.

For N_1, N_2, N_3, and N_4 see Eq. (3.19). $\mathbf{n}$ is the element unit normal vector, Eq. (3.13), and $\mathbf{N}_B$ is the shape function vector for the boundary element, Eqs. (3.18-3.19).

6.3 Velocity Coupling Matrix, T

The global velocity coupling matrix $\mathbf{T}$ consists of m assembled local transformation vectors $\mathbf{T}_e$, where m is the number of FE or BE nodes on boundary a. The local transformation matrix has the size 1×3 and is obtained by taking the transponent of the boundary surface unit normal vector.

$$\mathbf{T}_e = \mathbf{n}^T. \tag{6.10}$$

Chapter 7

Implemented BEM/FEM Functions

The described functions in this chapter handle the coupling between the BE method and the FE method. The functions are included in appendix C.

7.1 Bem_velotrans

Purpose:

Compute coupling vector to connect the normal velocity of a BE node with the translation of a FE node.

Syntax:

```
Te=bem_velotrans(ex,ey,ez,n)
Te=bem_velotrans(ex,ey,ez)
```

Description

bem_velotrans provides the coupling vector Te that is coupling the normal velocity of a BE-node with the translation of a FE-node. The input variables

$$\begin{aligned} \mathsf{ex} &= [x_1 \quad x_2 \quad x_3 \quad x_4] \\ \mathsf{ey} &= [y_1 \quad y_2 \quad y_3 \quad y_4] \\ \mathsf{ez} &= [z_1 \quad z_2 \quad z_3 \quad z_4] \end{aligned}$$

supply the corner coordinates for the boundary and the finite element on which the BE and the FE node which are to be connected, are placed. Notice that the boundary and the finite element must be coinciding and therefore have the same coordinates. If n has the value 1 Te is calculated for connection with a 12 dof plate element, i.e. rotation around x and y axis and translation in the z direction, otherwise Te is calculated for connection with a 24 dof shell element, i.e. rotation and translation in all directions.

Theory:

The relationship between BE normal velocity v_{BE} and FE translation degrees of freedom $\mathbf{w}$ can be written as

$$v_{BE} = i\omega(\mathsf{Te} \cdot \mathbf{w})$$

where $i = \sqrt{-1}$, ω is the angular frequency of the acoustic medium, and Te is a unit vector normal to the boundary and the finite element.

7.2 Bem_ptrans

Purpose:

Compute coupling vector to connect the pressure and forces between a boundary element and a finite element.

Syntax:

```
Le=bem_ptrans(ex,ey,ex,n)
Le=bem_ptrans(ex,ey,ez)
```

Description:

bem_ptrans provides the coupling vector Le that is connecting pressure of a boundary element with the forces of a finite element. The input variables are explained in bem_velotrans.

Theory:

The coupling vector Le is obtained by the following expression

$$\mathsf{Le} = \int_{S_e} \mathbf{N}_F^T \mathbf{n} \mathbf{N}_B dS$$

where S_e is a surface, $\mathbf{N}_F$ is the shape function matrix for the finite element, $\mathbf{n}$ is the unit normal vector of surface S_e, and $\mathbf{N}_B$ is the shape function vector for the boundary element (see bem_infl4q). Notice that the boundary and the finite element are coinciding.

7.3 Bem_assempres

Purpose:

Assemble coupling matrices that connects pressure and forces between boundary and finite elements.

Syntax:

L=bem_assempres(L,Le,bedof,fedof,con)

Description

bem_assempres adds the local pressure/force coupling vector Le to the global coupling matrix L according to the vectors bedof and fedof which are describing the topology for the boundary element and the finite element respectively. bedof and fedof are defined as

$$\mathsf{bedof} = [\ elnum \quad dof_1 \quad dof_2 \quad dof_3 \quad dof_{n=4}\ ]$$
$$\mathsf{fedof} = [\ elnum \quad dof_1 \quad dof_2 \quad \cdots \quad dof_n\ ]$$

where *elnum* is the element number and the columns 2 to $(n+1)$ contain the corresponding global degrees of freedom. For the fedof vector n is 12 for plate elements and 24 for shell elements.
con is a vector that contains the numbers of the BE nodes that are connected to FE nodes.
L has the size $k \times l$, where k is the number of FE degrees of freedom and l is the number of BE nodes that are coupled with FE nodes.

7.4 Bem_assemvel

Purpose:

Assemble coupling matrices that connects velocity and translation between BE and FE nodes.

Syntax:

T=bem_assemvel(T,Te,bn,fn,con)

Description:

bem_assemvel adds the local velocity/translation coupling vector Te to the global coupling matrix T according to bn and fn.
bn and fn are defined as

$$\mathsf{bn} = benode$$
$$\mathsf{fn} = [\ x_n \quad y_n \quad z_n\ ]$$

where *benode* is the number of the BE node that is connected to the FE translation degrees of freedom x_n, y_n, and z_n. If a plate element with 12 dof is used fn only contains z_n.
con is explained in bem_assempre.
T has the size $l \times k$ (compare with L in bem_assempres).

7.5 Bem_coupassem

Purpose:

Assemble the coupled BE/FE system.

Syntax:

[Couple,f1,f2]=bem_coupassem(K,C,M,L,T,H,G,ep,con)

Description:

bem_coupassem assembles the FE system matrices (K: stiffness, C: damping M: mass), the coupling matrices (L: pressure/forces, T: velocity/translation), and the BE system matrices (H and G: influence matrices). The assembling results

in three coupled system matrices (Couple, f1, and f2).
The input variable

$$\mathsf{ep} = [\ rho \quad w\ ]$$

provides the density, $rho = \rho_0$, and the angular frequency, $w = \omega$, for the acoustic medium.
con is explained in bem_assempre.

Theory:

For an elastic shell structure the resulting finite element model can be represented in the following way:

$$(\mathsf{K} + i\omega\mathsf{C} - \omega^2\mathsf{M})\mathbf{w} = \mathbf{F},$$

where $\mathbf{w}$ is the displacement vector for the FE model and $\mathbf{F}$ is the corresponding force vector.

An acoustic BE model has the form:

$$\mathsf{H} \cdot \mathbf{p} = \mathsf{G} \cdot \mathbf{v},$$

where $\mathbf{p}$ is the node pressure vector and $\mathbf{v}$ is the node normal velocity vector. If $\mathbf{p}_a$ is the node pressure vector for BE nodes coupled with FE nodes, $\mathbf{p}_b$ is the corresponding vector for uncoupled BE nodes, and $\mathbf{v}_b$ is the normal velocity vector for uncoupled BE nodes, the two equations above can be written as

$$(\mathsf{K} + i\omega\mathsf{C} - \omega^2\mathsf{M})\mathbf{w} + \mathsf{L} \cdot \mathbf{p}_a = \mathbf{F},$$

$$\begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{p}_a \\ \mathbf{p}_b \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} i\omega(\mathsf{T} \cdot \mathbf{w}) \\ \mathbf{v}_b \end{bmatrix}.$$

The coupled BE/FE model now takes the form

$$\begin{bmatrix} \mathsf{K} + i\omega\mathsf{C} - \omega^2\mathsf{M} & \mathsf{L} & \mathbf{0} \\ -i\omega\mathbf{G}_{11}\mathsf{T} & \mathbf{H}_{11} & \mathbf{H}_{12} \\ -i\omega\mathbf{G}_{21}\mathsf{T} & \mathbf{H}_{12} & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{p}_a \\ \mathbf{p}_b \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{G}_{12}\mathbf{v}_b \\ \mathbf{G}_{22}\mathbf{v}_b \end{bmatrix},$$

with

$$\mathsf{f}_1 = \mathbf{G}_{12},$$

$$\mathsf{f}_2 = \mathbf{G}_{22},$$

$$\mathsf{Couple} = \begin{bmatrix} \mathsf{K} + i\omega\mathsf{C} - \omega^2\mathsf{M} & \mathsf{L} & \mathbf{0} \\ -i\omega\mathbf{G}_{11}\mathsf{T} & \mathbf{H}_{11} & \mathbf{H}_{12} \\ -i\omega\mathbf{G}_{21}\mathsf{T} & \mathbf{H}_{12} & \mathbf{H}_{22} \end{bmatrix}.$$

7.6 Bem_coupsolveq

Purpose:

Solve coupled BE/FE equation system.

Syntax:

[pr,nv]=bem_coupsolveq(Couple,f1,f2,T,con,bc,f,bcpr,bcnv,bcim,ep)

Description:

bem_coupsolveq solves the equation system shown in bem_coupassem for pressure pr and normal velocity nv.

$$\mathsf{pr} = \begin{bmatrix} \mathbf{p}_a \\ \mathbf{p}_b \end{bmatrix} \qquad \mathsf{nv} = \begin{bmatrix} iw(\mathsf{T}\mathbf{w}) \\ \mathbf{v}_b \end{bmatrix}$$

The input variables

$$\mathsf{bc} = \begin{bmatrix} dof_1 & \mathrm{w}_1 \\ dof_2 & \mathrm{w}_2 \\ \vdots & \vdots \\ dof_{nbc} & \mathrm{w}_{nbc} \end{bmatrix} \quad \mathsf{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \quad \mathsf{ep} = [w]$$

contain prescribed boundary conditions on the FE structure where *dof* denotes the prescribed degree of freedom and w the prescribed displacement (nbc is the number of boundary conditions), the FE force vector (n is the number of degrees of freedom), and the angular frequency in the acoustic medium w.

Couple, f1, and f2 are output from bem_coupassem. T is output from bem_assemvel and con is explained in bem_assempre. bcpr, bcnv, and bcim are boundary condition vectors for the boundary elements (explained in bem_solveq), notice that only BE nodes that are not connected with FE nodes can have boundary conditions.

Chapter 8

Using Coupled BEM/FEM

8.1 BEM/FEM or FEM/FEM

For coupled structure-acoustic problems one can use acoustic boundary elements or acoustic finite elements. There are pros and cons for both methods, but generally coupled FEM/FEM should be used for internal problems, and coupled BEM/FEM should be used for external problems with unbounded regions, i.e. infinite regions.

Even if a BEM/FEM model usually have less elements than a corresponding FEM/FEM model it does not mean that the BEM/FEM model result in higher computational efficiency. Since the influence matrices in a BEM model are fully populated (chapter 4), and not banded as the stiffness matrix for a FEM model, they can take a considerable amount of time to calculate. So, for a internal problem a FEM/FEM model is often quicker.

For external unbounded problems BEM/FEM comes to its full right, since boundary elements satisfy the Sommerfeld radiation condition, Eq. (1.6), and therefore can handle infinite regions.

An unbounded region is difficult to model with FEM/FEM since the problem region must be demarcated before it is divided into elements. The demarcation creates a boundary which boundary conditions are hard to find. With improper boundary values the solution will be disturbed by the demarcation.

8.2 Example with Vibrating Box

To verify to the correctness of the developed coupling functions described in chapter 7, they are used to examine the sound level in a vibrating box [14], Fig. 8.1. The coupling functions are used together with bem_infl4q (acoustic boundary element) and plateqd (finite plate element), see appendix D. Plateqd is a MATLAB function, implemented in CALFEM, based on the article *Evaluation of a New Quadrilateral Thin Plate Bending Element* [15].

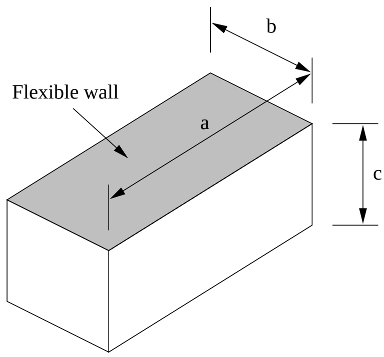


Figure 8.1: *Rectangular box with one flexible wall*

The box consists of five hard walls and one flexible, and has the dimension a × b × c = 304.8 × 152.4 × 152.4 mm. The hard walls are modeled with boundary elements and the flexible, which consists of a 1.63 mm thick undamped aluminum plate, is modeled with coupled boundary and finite elements. The aluminum plate is subjected to an evenly distributed time harmonic pressure load and the acoustic medium inside the box is air.

The coupled BEM/FEM results are compared with results obtained from a coupled FEM/FEM analysis, where the plate element function plateqd has been used together with functions described in *CALFEM, Acoustic and Interface Elements for Structure-Acoustic Analysis* [16].

As can be seen in Fig. 8.2 and 8.3, the sound pressure level is calculated for the box center and on the plate center. The results for the two different methods are very similar at sound level peaks (at resonance frequencies), but for sound level valleys (zero points) they tend to differ a bit. Since sound level valleys occur at different frequencies for different locations, the prediction of them is very sensitive. Prediction of sound

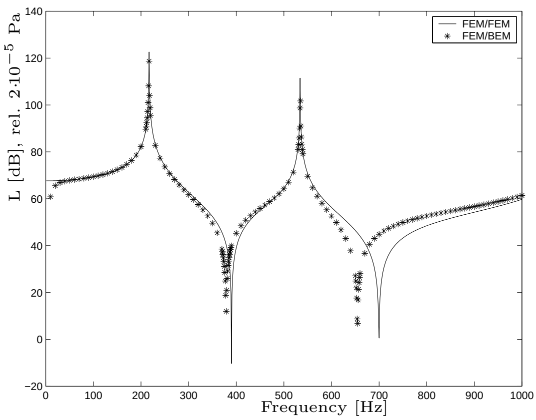


Figure 8.2: *Sound pressure level, L, in box center for time-harmonic waves*

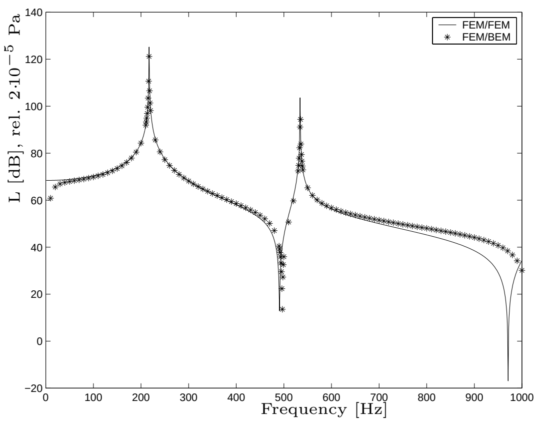


Figure 8.3: *Sound pressure level on plate center for time-harmonic waves*

level peaks is less sensible, as they occur at resonance frequencies and therefore are independent of location.

8.3 Example with Vibrating Plate

In this example the flexible aluminum plate from the section above is placed on a hard, infinite, and plane surface, Fig. 8.4. The surface is in contact with an acoustic medium and the aluminum plate is driven with a time harmonic pressure. Since the hard and infinite plane does not contribute to the solution, only the aluminum plate has to be modeled.

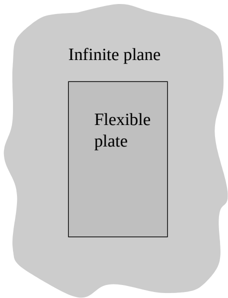


Figure 8.4: *Flexible plate on an infinite plane*

The resulting sound level, for air and water, on the plate center for different frequencies is shown in Fig. 8.5. As can be read from the figure, the resonance frequency for a plate vibrating in a heavy fluid will decrease.

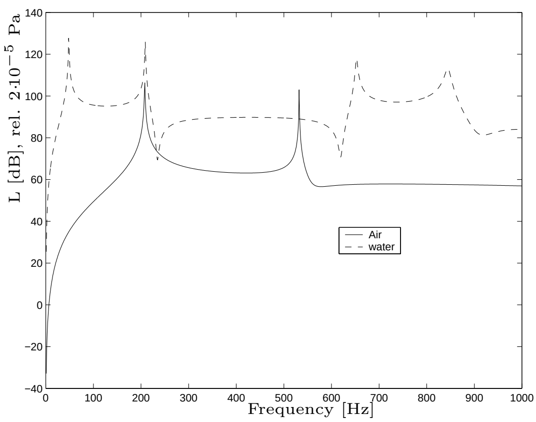


Figure 8.5: *Sound pressure level on plate center for time-harmonic waves*

For the air/plate interaction the first two resonance frequencies occur at 209 and 532 Hz. The corresponding values for the water/plate interaction is 48 and 209 Hz. Without the interaction between fluid and plate, i.e. the plate is vibrating in vacuum, the two first resonance frequencies occur at 210 and 537 Hz. Notice, that the coinciding of the resonance frequency for the air/plate interaction and the water/plate interaction at 209 Hz is accidental.

Often when resonance frequency analyses are performed on structures in contact with air, no interaction is considered. Since air does not affect the result that much this is a fair approximation.

Chapter 9

Comments on Implemented Functions

In this chapter short comments regarding the code of some of the functions that are included in appendix A and C are given. The entire codes are not explained but the sections below might give a hint on how they are written.

9.1 Bem_infl4q

Bem_infl4q calculates the integral Eqs. (3.21-3.22) by the use of a summation based on four Gauss points, as mentioned in section 3.3.1. The resulting summation for $\bar{h}^1_{ij}$, as an example, then takes the form

$$\bar{h}_{ij} = \sum_{k=1}^{4} (\nabla g_k)^T \, \hat{\mathbf{n}} \, \mathrm{A}_k, \tag{9.1}$$

which should be compared with Eq. (3.12). The function ∇g_k has the same appearance as Eq. (3.14) with the difference that the function variables have the form $x_j - x_i^k$, $y_j - y_i^k$, and $z_j - z_j^k$, where (x_i^k, y_i^k, z_i^k) is a Gauss point. x_i^k is calculated through the relation [11]

$$x_i^k = x_i(\xi_k, \eta_k) = \mathbf{N}(\xi_k, \eta_k)\mathbf{x}, \tag{9.2}$$

where $\mathbf{N}$ is the shape function vector, Eq. (3.19), $\mathbf{x}$ is a vector containing the x-position for the element corners, and (ξ_k, η_k) are the Gauss points in the isoparametric coordinate system, Eq. (3.11). y_i^k and z_i^k are calculated in a similar fashion.

The area A_k can be written as

$$\mathrm{A}_k = \det\mathbf{J}_k, \tag{9.3}$$

where

$$\det\mathbf{J}_k = \sqrt{(\det\mathbf{J}_k^{xy})^2 + (\det\mathbf{J}_k^{yz})^2 + (\det\mathbf{J}_k^{zx})^2}. \tag{9.4}$$

J_k^{xy}, J_k^{yz}, and J_k^{zx} is the Jacobian matrix $\mathbf{J}$ [11] reflected on the xy-plane, yz-plane, and zx-plane respectively.

9.2 Bem_solveq

The function bem_solveq solve BEM systems, as the illustrative one below

$$\begin{bmatrix} \mathrm{H}_{11} & \mathrm{H}_{12} & \mathrm{H}_{13} \\ \mathrm{H}_{21} & \mathrm{H}_{22} & \mathrm{H}_{23} \\ \mathrm{H}_{31} & \mathrm{H}_{32} & \mathrm{H}_{33} \end{bmatrix} \begin{bmatrix} p_1^u \\ p_2^k \\ p_3^u \end{bmatrix} = \begin{bmatrix} \mathrm{G}_{11} & \mathrm{G}_{12} & \mathrm{G}_{13} \\ \mathrm{G}_{21} & \mathrm{G}_{22} & \mathrm{G}_{23} \\ \mathrm{G}_{31} & \mathrm{G}_{32} & \mathrm{G}_{33} \end{bmatrix} \begin{bmatrix} v_1^k \\ v_2^u \\ v_3^k \end{bmatrix}, \tag{9.5}$$

where k denotes a prescribed boundary value and u denotes an unknown one. In order to solve the system above all known boundary values must be placed on one side. With all known values on the right hand side this gives the following equation system

$$\begin{bmatrix} \mathrm{H}_{11} & \mathrm{G}_{12} & \mathrm{H}_{13} \\ \mathrm{H}_{21} & \mathrm{G}_{22} & \mathrm{H}_{23} \\ \mathrm{H}_{31} & \mathrm{G}_{32} & \mathrm{H}_{33} \end{bmatrix} \begin{bmatrix} p_1^u \\ -v_2^u \\ p_3^u \end{bmatrix} = \begin{bmatrix} \mathrm{G}_{11} & \mathrm{H}_{12} & \mathrm{G}_{13} \\ \mathrm{G}_{21} & \mathrm{H}_{22} & \mathrm{G}_{23} \\ \mathrm{G}_{31} & \mathrm{H}_{32} & \mathrm{G}_{33} \end{bmatrix} \begin{bmatrix} v_1^k \\ -p_2^k \\ v_3^k \end{bmatrix}, \tag{9.6}$$

which can be solved for the unknown values on the left hand side.
If the impedance is given as a boundary condition (for instance, $z_1^k = \frac{p_1^u}{v_1^u} = 0.5$) in a BEM system it will take the form

$$\begin{bmatrix} \mathrm{H}_{11} & \mathrm{H}_{12} \\ \mathrm{H}_{21} & \mathrm{H}_{22} \end{bmatrix} \begin{bmatrix} 0.5v_1^u \\ p_2^u \end{bmatrix} = \begin{bmatrix} \mathrm{G}_{11} & \mathrm{G}_{12} \\ \mathrm{G}_{21} & \mathrm{G}_{22} \end{bmatrix} \begin{bmatrix} v_1^u \\ v_2^k \end{bmatrix}, \tag{9.7}$$

or

$$\begin{bmatrix} 0.5\mathrm{H}_{11} - \mathrm{G}_{11} & \mathrm{H}_{12} \\ 0.5\mathrm{H}_{21} - \mathrm{G}_{21} & \mathrm{H}_{22} \end{bmatrix} \begin{bmatrix} v_1^u \\ p_2^u \end{bmatrix} = \begin{bmatrix} \mathrm{G}_{12} \\ \mathrm{G}_{22} \end{bmatrix} \begin{bmatrix} v_2^k \end{bmatrix}. \tag{9.8}$$

After the system is solved for the unknown variables, p_1^u is determined.

9.3 Bem_spacang

The function bem_spacang calculates the space angle constant, c_c, for nodes located on a non-smooth surface and it is based on Euler-Eriksson's formula [12]

$$\tan\frac{\Omega}{2} = \frac{|a\cdot(b\times c)|}{1+a\cdot b+b\cdot c+c\cdot a}, \tag{9.9}$$

where Ω is the space angle (away from the fluid domain), a, b, and c are unit vectors (Fig. 9.1), and $c_c = 1-\Omega/2\pi$. The function calculates the

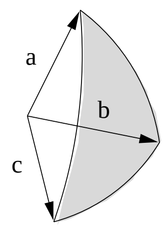


Figure 9.1: *Space angle defined by three unit vectors*

space angle by dividing it into a number of subangles that are determined and added up. Furthermore, bem_spacang can not calculate the space angle if the angle α between two adjacent planes is $\frac{\pi}{2}\leq\alpha\leq\frac{3\pi}{2}$.

9.4 Bem_coupsolveq

The function bem_coupsolveq solve coupled FE/BE systems, Eq. (6.6). With focus on the components that are multiplied with the uncoupled BE node values, $\mathbf{p}_b$ and $\mathbf{v}_b$, Eq. (6.6) is written as (with help of Eqs. (6.7-6.8))

$$\begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \mathbf{H}_{12} \\ \bullet & \bullet & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \bullet \\ \bullet \\ \mathbf{p}_b \end{bmatrix} = \begin{bmatrix} \bullet \\ i\rho_0\omega\mathbf{G}_{12}\mathbf{v}_b \\ i\rho_0\omega\mathbf{G}_{12}\mathbf{v}_b \end{bmatrix}, \tag{9.10}$$

or with the interesting parts taken out of the system equation

$$\begin{bmatrix} \mathbf{H}_{12} \\ \mathbf{H}_{22} \end{bmatrix} [\mathbf{p}_b] \neq i\rho_0\omega \begin{bmatrix} \mathbf{G}_{12} \\ \mathbf{G}_{22} \end{bmatrix} [\mathbf{v}_b]. \tag{9.11}$$

Observe that the left hand side and the right hand side of Eq. (9.11) are not equivalent.

Eq. (9.11) has the same principle appearance as Eq. (9.5) with prescribed and unknown boundary values on both sides. The prescribed boundary values are sorted to the left and the unknown values to the right, in similarity with Eq. (9.6). When Eq. (9.11) has been sorted with respect to the boundary values the entire system, Eq. (9.10), can be solved. If the impedance is given as boundary condition, Eq. (9.11) is sorted in the same manner as Eq. (9.7).

Chapter 10

Concluding Remarks

As the examples in this report show, the functions that have been developed to enable BEM and coupled BEM/FEM problem models works satisfactorily. However, since BEM problems can be very time consuming, my opinion is that future improvements to this thesis in first hand should concern time reducing measures.

10.1 Time Reducing Measures

- Since the free-space Green's function, Eq. (1.2), is frequency dependent the matrix coefficients in boundary element models are frequency dependent. As a result, a boundary element model does not lead to an algebraic eigenvalue problem for the extraction of the natural frequencies. This, of course, makes a frequency analysis very time consuming if the matrix coefficients have to be calculated for every frequency, which is done for the examples in section 8.2 and 8.3.

  The natural frequencies can be calculated in a quicker manner if the frequency dependent Helmholtz problem is decomposed into two subproblems: one consisting of a homogeneous, frequency independent Laplace problem and the other being a Laplace problem, in which the frequency dependence of the Helmholtz problem is incorporated as an inhomogeneous right-hand side excitation. This leads to an algebraic non-symmetric eigenvalue problem for the natural frequencies.[7]

- In this thesis constant and linear boundary elements have been developed. As described in section 5.2.1 there is a limit for how coarse the mesh can be if one wants reliable results.

With quadratic elements the mesh could be coarser, with shorter calculation time as a result, and still provide reliable results. Quadratic elements also makes it possible to model curved boundaries without node duplications at element boarders, which sometimes has to be done for linear elements (section 5.2.2).

Bibliography

[1] Odd Tullberg. *A Study of the Boundary Element Method*. Department of Structural Mechanics, Chalmers University of Technology, 1983.

[2] C.A Brebbia. *Boundary Element Techniques in Computer-Aided Engineering*. Martin Nijhoff Publisher, Dordrecht, 1984.

[3] Pehr Ahlin. *Boundary Element Simulation of a Loudspeaker in a Mobile Telephone*. Master Thesis, Engineering Acoustics LTH, Lund University, Report TVBA-5025, Lund, 1998.

[4] *CALFEM, a Finite Element Toolbox to MATLAB*. Division of Structural Mechanics and Department of Solid Mechanics, Lund University, 1999.

[5] R.D. Ciskowski and C.A. Brebbia. *Boundary Element Methods in Acoustics*. WIT Press, Southampton, 1991.

[6] R. Bai Mingsian. *Study of Acoustic Resonance in Enclosures Using Eigenanalysis Based on Boundary Elements Methods*. The Journal of the Acoustical Society of America, page 2529, Vol. 91, No. 5, May 1992.

[7] W. Desmet. *Boundary Elements Methods in Acoustics*. ISAAC 11, Seminar on Advanced Techniques in Applied & Numerical Acoustics, Katholieke Universiteit Leuven, Belgium, 2000.

[8] Miguel C. Junger and David Felt. *Sound, Structures, and their Interaction*. Published by the Acoustical Society of America through the American Institute of Physics, 1993.

[9] Niels Saabye Ottosen and Matti Ristinmaa. *The Mechanics of Constitutive Modelling.* Department of Solid Mechanics, University of Lund, Lund, 1996.

[10] Klaus-Jürgen Bathe. *Finite Element Procedures.* Prentice Hall, Upper Sadle River, New Jersey.

[11] Niels Saabye Ottosen and Hans Petersson. *Introduction to the Finite Element Method.* Prentice Hall, Europe, 1992.

[12] Lennart Råde and Bertil Westergren. *BETA, Mathematics Handbook for Science and Engineering.* Studentlitteratur, Lund, Sweden, Third edition, 1999.

[13] Torgil Ekman. *Numeriska Metoder på Dator och Dosa.* Datalogi och numerisk analys, Lunds Tekniska Högskola, Andra upplagan, 1995.

[14] A.J. Pretlove. *Forced Vibrations of a Rectangular Panel, Backed by a Closed Redctangular Cavity.* Journal of Sound Vibrations 2, page 252, 1966.

[15] Jean-Louis Batoz and Mabrouk Ben Tahar. *Evaluation of a New Quadrilateral Thin Plate Bending Element.* International Journal for Numerical Methods in Engineering, Vol. 18, page 1655, 1982.

[16] Göran Sandberg. *CALFEM, Acoustic and Interface Elements for Structure-Acoustic Analysis.* Division of Structural Mechanics, LTH, Lund University, Sweden, 2001.

Appendix A

BEM Functions

```
function [He,Ge]=bem_infl1q(ex,ey,ez,ep,n);
% [He,Ge]=bem_infl1q(ex,ey,ez,ep,n)
% [He,Ge]=bem_infl1q(ex,ey,ez,ep)
%-------------------------------------------------------------------
% PURPOSE
%  Compute the influence coefficients He and Ge
%  for a three dimensional constant acoustic element.
%
% INPUT:  ex = [xi1 xi2 xi3 xi4
%               xj1 xj2 xj3 xj4]
%         ey = [yi1 yi2 yi3 yi4
%               yj1 yj2 yj3 yj4]
%         ez = [zi1 zi2 zi3 zi4
%               zj1 zj2 zj3 zj4]  corner coordinates for element i and
%                                 j. The influenced node is located on
%                                 element i and element j is the
%                                 influencing surface.
%
%         ep = [w c rho]          problem properties
%                                 w: angular frequency
%                                 c: speed of sound in acoustic medium
%                                 rho: density of the acoustic medium
%
%         n=[value]               normal direction
%                                 value=1 default
%                                 value=-1 reverse
%
% OUTPUT: He, Ge:                 Influence coefficients
%-------------------------------------------------------------------
rev=1;
if nargin==5
   rev=n;
```

```
end

k=ep(1)/ep(2);
x1=sum(ex(1,:))/4; y1=sum(ey(1,:))/4; z1=sum(ez(1,:))/4;
x2=sum(ex(2,:))/4; y2=sum(ey(2,:))/4; z2=sum(ez(2,:))/4;
diff=[ex(2,:)-ex(1,:) ey(2,:)-ey(1,:) ez(2,:)-ez(1,:)];
exA=ex(2,:); eyA=ey(2,:); ezA=ez(2,:);

%****Element Area****
Axy=1/2*(exA(1)*eyA(2)-exA(1)*eyA(4)-exA(2)*eyA(1)+exA(2)*eyA(3)...
   -exA(3)*eyA(2)+exA(3)*eyA(4)+exA(4)*eyA(1)-exA(4)*eyA(3));
Azx=1/2*(ezA(1)*exA(2)-ezA(1)*exA(4)-ezA(2)*exA(1)+ezA(2)*exA(3)...
    -ezA(3)*exA(2)+ezA(3)*exA(4)+ezA(4)*exA(1)-ezA(4)*exA(3));
Ayz=1/2*(eyA(1)*ezA(2)-eyA(1)*ezA(4)-eyA(2)*ezA(1)+eyA(2)*ezA(3)...
   -eyA(3)*ezA(2)+eyA(3)*ezA(4)+eyA(4)*ezA(1)-eyA(4)*ezA(3));
A=sqrt(Axy^2+Azx^2+Ayz^2);

if diff==0
   %****For Coinciding Elements****
   He=1/2;
   g1=0.577350269189626;
   xi=[-g1; g1; g1;-g1];
   eta=[-g1;-g1; g1; g1];
   rx=1/4*[(xi-1).*(eta-1) -(xi+1).*(eta-1) (xi+1).*...
         (eta+1) -(xi-1).*(eta+1)]*ex(2,:)';
   ry=1/4*[(xi-1).*(eta-1) -(xi+1).*(eta-1) (xi+1).*...
         (eta+1) -(xi-1).*(eta+1)]*ey(2,:)';
   rz=1/4*[(xi-1).*(eta-1) -(xi+1).*(eta-1) (xi+1).*...
         (eta+1) -(xi-1).*(eta+1)]*ez(2,:)';
   dist=sqrt((rx-x1).^2+(ry-y1).^2+(rz-z1).^2);
   dis=sum(dist)/4;
   Ge=i*ep(3)*ep(1)*A*exp(-i*k*dis)/(4*pi*dis);
else

   %****For not Coinciding Elements****
   a=[ex(2,2)-ex(2,1) ey(2,2)-ey(2,1) ez(2,2)-ez(2,1)];
   b=[ex(2,4)-ex(2,1) ey(2,4)-ey(2,1) ez(2,4)-ez(2,1)];
   n=[a(2)*b(3)-a(3)*b(2); a(3)*b(1)-a(1)*b(3); a(1)*b(2)-a(2)*b(1)];
   n=rev*n/sqrt(n'*n);
   dis=sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2);
   h1=-(x2-x1)*exp(-i*k*dis)/(4*pi*dis^2)*(i*k+1/dis);
   h2=-(y2-y1)*exp(-i*k*dis)/(4*pi*dis^2)*(i*k+1/dis);
   h3=-(z2-z1)*exp(-i*k*dis)/(4*pi*dis^2)*(i*k+1/dis);
   He=-[h1 h2 h3]*n*A;
   Ge=i*ep(3)*ep(1)*A*exp(-i*k*dis)/(4*pi*dis);
end
%-----------------------------------end----------------------------------
```

```
function [He,Ge]=bem_infl4q(coord,ex,ey,ez,ep,n)
% [He,Ge]=bem_infl4q(coord,ex,ey,ez,ep,n)
% [He,Ge]=bem_infl4q(coord,ex,ey,ez,ep)
%-------------------------------------------------------------------------
% PURPOSE
%  Compute the element influence matrices He and Ge for a three
%  dimensional four-node quadrilateral acoustic element.
%
% INPUT:   coord=[x y z]     coordinates of the influenced node
%
%          ex=[x1 x2 x3 x4]
%          ey=[y1 y2 y3 y4]
%          ez=[z1 z2 z3 z4]  node coordinates for the influencing
%                            element.
%
%          ep = [w c rho]    problem properties
%                            w: angular frequency
%                            c: speed of sound in acoustic medium
%                            rho: density of acoustic medium
%
%          n=[value]         normal direction
%                            value=1 default
%                                 -1 reverse
%
% OUTPUT:  He, Ge:  Element influence matrices
%-------------------------------------------------------------------------
rev=1;
if nargin==6
   rev=n;
end k=ep(1)/ep(2);

%****Gauss points****
ga=0.577350269189626; x=coord(1); y=coord(2); z=coord(3);
xi=[-ga; ga; ga; -ga];      eta=[-ga; -ga; ga; ga];
N(:,1)=1/4*(xi-1).*(eta-1); N(:,2)=-1/4*(xi+1).*(eta-1);
N(:,3)=1/4*(xi+1).*(eta+1); N(:,4)=-1/4*(xi-1).*(eta+1);
xg=N*ex';   yg=N*ey';   zg=N*ez';

%****Element Area****
dNr(1:2:7,1)=-(1-eta)/4;     dNr(1:2:7,2)= (1-eta)/4;
dNr(1:2:7,3)= (1+eta)/4;     dNr(1:2:7,4)=-(1+eta)/4;
dNr(2:2:8,1)=-(1-xi)/4;      dNr(2:2:8,2)=-(1+xi)/4;
dNr(2:2:8,3)= (1+xi)/4;      dNr(2:2:8,4)= (1-xi)/4;

JTxy=dNr*[ex;ey]'; JTyz=dNr*[ey;ez]'; JTzx=dNr*[ez;ex]';

detJxy=[det(JTxy(1:2,:));det(JTxy(3:4,:));det(JTxy(5:6,:))...
```

```
      ;det(JTxy(7:8,:))];
detJyz=[det(JTyz(1:2,:));det(JTyz(3:4,:));det(JTyz(5:6,:))...
      ;det(JTyz(7:8,:))];
detJzx=[det(JTzx(1:2,:));det(JTzx(3:4,:));det(JTzx(5:6,:))...
      ;det(JTzx(7:8,:))];
A=[sqrt(detJxy.^2+detJyz.^2+detJzx.^2)];

%****Influence Vectors****
xdis=xg-x;  ydis=yg-y;  zdis=zg-z;
dis=sqrt(xdis.^2+ydis.^2+zdis.^2);

g=i*ep(3)*ep(1)*exp(-i*k*dis)./(4*pi*dis);
Ge(1,1)=sum(g.*N(:,1).*A); Ge(1,2)=sum(g.*N(:,2).*A);
Ge(1,3)=sum(g.*N(:,3).*A); Ge(1,4)=sum(g.*N(:,4).*A);

h1=-xdis.*exp(-i*k*dis)./(4*pi*dis.^2).*(i*k+1./dis);
h2=-ydis.*exp(-i*k*dis)./(4*pi*dis.^2).*(i*k+1./dis);
h3=-zdis.*exp(-i*k*dis)./(4*pi*dis.^2).*(i*k+1./dis);

a=[ex(2)-ex(1) ey(2)-ey(1) ez(2)-ez(1)]; b=[ex(4)-ex(1)
ey(4)-ey(1) ez(4)-ez(1)]; n=[a(2)*b(3)-a(3)*b(2);
a(3)*b(1)-a(1)*b(3); a(1)*b(2)-a(2)*b(1)]; n=rev*n/sqrt(n'*n);

h=[h1 h2 h3]*n; He(1,1)=-sum(h.*N(:,1).*A);
He(1,2)=-sum(h.*N(:,2).*A); He(1,3)=-sum(h.*N(:,3).*A);
He(1,4)=-sum(h.*N(:,4).*A);
%-----------------------------------end--------------------------------
```

```
function Ce=bem_spacang(coord,ex,ey,ez)
% Ce=bem_spacang(coord,ex,ey,ez)
%-------------------------------------------------------------------------
% PURPOSE
%  Compute the space angle constant for a node on a non-smooth surface.
%
% INPUT:   coord=[x y z]            coordinates for the node where the
%                                   space angle constant is to be
%                                   calculated
%          ex=[x11 x12 x13 x14
%               .   .   .   .
%              xn1 xn2 xn3 xn4]
%          ey=[y11 y12 y13 y14
%               .   .   .   .
%              yn1 yn2 yn3 yn4]
%          ez=[z11 z12 z13 z14
%               .   .   .   .
%              zn1 zn2 zn3 zn4]     Coordinate matrices for elements
%                                   that coincides with the node of
%                                   interest. n is three or four.
%
% OUTPUT:  Ce:    The space angle constant
%-------------------------------------------------------------------------
[nel,v]=size(ex);
Ex=[ex(:,4) ex ex(:,1)]; Ey=[ey(:,4) ey ey(:,1)];
Ez=[ez(:,4) ez ez(:,1)];

exo=ex-coord(1)*ones(nel,4); eyo=ey-coord(2)*ones(nel,4);
ezo=ez-coord(3)*ones(nel,4);
res=abs(exo)+abs(eyo)+abs(ezo);
[val,col]=min(res');

%****Element border vectors****
for k=1:nel
   A1(k,:)=[Ex(k,col(k))-Ex(k,col(k)+1) Ey(k,col(k))-Ey(k,col(k)+1)...
      Ez(k,col(k))-Ez(k,col(k)+1)];
   A2(k,:)=[Ex(k,col(k)+2)-Ex(k,col(k)+1) Ey(k,col(k)+2)-Ey...
      (k,col(k)+1)Ez(k,col(k)+2)-Ez(k,col(k)+1)];
   a1(k,:)=A1(k,:)/sqrt(sum(A1(k,:).^2));
   a2(k,:)=A2(k,:)/sqrt(sum(A2(k,:).^2));
end

%****Reference normal direction****
n1=a1(1,2)*a2(1,3)-a1(1,3)*a2(1,2);
n2=a1(1,3)*a2(1,1)-a1(1,1)*a2(1,3);
n3=a1(1,1)*a2(1,2)-a1(1,2)*a2(1,1);
n=[n1 n2 n3]/sqrt(n1^2+n2^2+n3^2);
```

```
ang=0;  sa=0;
if nel==3
   a1(4,:)=a1(1,:);
end for k=1:nel
   kr(1,1)=a2(k,2)*a1(k,3)-a2(k,3)*a1(k,2);
   kr(2,1)=a2(k,3)*a1(k,1)-a2(k,1)*a1(k,3);
   kr(3,1)=a2(k,1)*a1(k,2)-a2(k,2)*a1(k,1);

   %****Convex points****
   if (a1(2,:)*n'>-1e-3) & (a1(3,:)*n'>-1e-3) & (a1(4,:)*n'>-1e-3)
      w=abs(n*kr)/(1+n*a2(k,:)'+a2(k,:)*a1(k,:)'+a1(k,:)*n');
      fi=2*atan(w);
      ang=ang+fi;

   %****Concave points****
   elseif (a1(2,:)*n'<1e-3) & (a1(3,:)*n'<1e-3) & (a1(4,:)*n'<1e-3)
      if k==1;
         ang=4*pi;
      end
      n=-n;
      w=abs(n*kr)/(1+n*a2(k,:)'+a2(k,:)*a1(k,:)'+a1(k,:)*n');
      fi=2*atan(w);
      ang=ang-fi;
      n=-n;

   %****Saddle points****
   else
      if k==1
         sa=1;
         w=abs(n*kr)/(1+n*a2(k,:)'+a2(k,:)*a1(k,:)'+a1(k,:)*n');
         fi(1)=2*atan(w);
      elseif (a1(k,:)*n'>=0) & (a2(k,:)*n'>=0)
         w=abs(n*kr)/(1+n*a2(k,:)'+a2(k,:)*a1(k,:)'+a1(k,:)*n');
         fi(2)=2*atan(w);
         down=k;
      elseif (a1(k,:)*n'<=0) & (a2(k,:)*n'<=0)
         n=-n;
         w=abs(n*kr)/(1+n*a2(k,:)'+a2(k,:)*a1(k,:)'+a1(k,:)*n');
         fi(3)=2*atan(w);
         n=-n;
         up=k;
      else
         x0=[Ex(k,col(k)); Ey(k,col(k)); Ez(k,col(k))]-coord';
         G=[-(a1(k,:)-a2(k,:))'/sqrt(sum(((a1(k,:)-a2(k,:)).^2)))...
            a1(1,:)' a2(1,:)'];
         t=G\x0;
```

```
        cut=[Ex(k,col(k)); Ey(k,col(k)); Ez(k,col(k))]-t(1)*G(:,1);
        hv=(cut'-coord)/sqrt(sum((cut'-coord).^2));
      end
   end
end

%****Saddle points****
if sa==1;
   if a1(up,:)*n'<0
      n=-n;
      v=acos(hv*a2(up,:)');
      kr(1,1)=a1(up,2)*hv(3)-a1(up,3)*hv(2);
      kr(2,1)=a1(up,3)*hv(1)-a1(up,1)*hv(3);
      kr(3,1)=a1(up,1)*hv(2)-a1(up,2)*hv(1);
      w=abs(n*kr)/(1+n*hv'+hv*a1(up,:)'+a1(up,:)*n');
      fi(4)=2*atan(w);
      n=-n;
      kr(1,1)=a2(down,2)*hv(3)-a2(down,3)*hv(2);
      kr(2,1)=a2(down,3)*hv(1)-a2(down,1)*hv(3);
      kr(3,1)=a2(down,1)*hv(2)-a2(down,2)*hv(1);
      w=abs(n*kr)/(1+n*hv'+hv*a2(down,:)'+a2(down,:)*n');
      fi(5)=2*atan(w);
   else
      n=-n;
      v=acos(hv*a1(up,:)');
      kr(1,1)=a2(up,2)*hv(3)-a2(up,3)*hv(2);
      kr(2,1)=a2(up,3)*hv(1)-a2(up,1)*hv(3);
      kr(3,1)=a2(up,1)*hv(2)-a2(up,2)*hv(1);
      w=abs(n*kr)/(1+n*hv'+hv*a2(up,:)'+a2(up,:)*n');
      fi(4)=2*atan(w);
      n=-n;
      kr(1,1)=a1(down,2)*hv(3)-a1(down,3)*hv(2);
      kr(2,1)=a1(down,3)*hv(1)-a1(down,1)*hv(3);
      kr(3,1)=a1(down,1)*hv(2)-a1(down,2)*hv(1);
      w=abs(n*kr)/(1+n*hv'+hv*a1(down,:)'+a1(down,:)*n');
      fi(5)=2*atan(w);
   end
   part=v/(2*pi);
   ang=4*pi*part-fi(3)-fi(4)+fi(1)+fi(2)+fi(5);
end
Ce=1-ang/(4*pi);
%------------------------------------end--------------------------------
```

```
function P=bem_assem(edof,P,Pe,n,el)
% P=bem_assem(edof,P,Pe,n,el)
%----------------------------------------------------------------------
% PURPOSE
%  Assemble element influence matrix Pe for acoustic problems into
%  the global influence matrix P according to the topology matrix
%  edof, the influenced node n, and the influencing element el.
%
% INPUT:   edof:     dof topology matrix
%          P:        global influence matrix
%          Pe:       element influence matrix
%          n:        influenced node
%          el:       influencing element
%
% OUTPUT:  P:   New global influence matrix
%----------------------------------------------------------------------
N=size(edof); if N(1,2)==2
   t=abs(edof(:,1)-el);
   [val,p]=min(t);
   P(n,edof(p,2))=P(n,edof(p,2))+Pe;
elseif N(1,2)==5
   t=abs(edof(:,1)-el);
   [val,p]=min(t);
   P(n,edof(p,2:5))=P(n,edof(p,2:5))+Pe;
end
%-------------------------------end------------------------------------
```

```
function [pr,nv]=bem_solveq(G,H,bcpr,bcnv,bcim)
% [pr,nv]=bem_solveq(P,V,bcpr,bcnv,bcim)
%-------------------------------------------------------------------------
% PURPOSE
%  Solve BE-equations considering boundary conditons
%
% INPUT:    G, H:     influence matrices
%           bcpr:     boundary condition matrix (pressure)
%           bcnv:     boundary condition matrix (normal velocity)
%           bcim:     boundary condition matrix (acoustic impedance)
%
% OUTPUT:  pr:  solution including boundary values (pressure)
%          nv:  solution including boundary values (normal velocity)
%-------------------------------------------------------------------------
[nd,nd]=size(G); fpdof=[1:nd]'; fvdof=[1:nd]'; fidof=[1:nd]';
[rowp,colp]=size(bcpr); [rowv,colv]=size(bcnv);
[rowi,coli]=size(bcim);

pr=zeros(size(fpdof)); nv=zeros(size(fvdof));

if rowp~=0
   ppdof=bcpr(:,1);
   prp=bcpr(:,2);
   fpdof(ppdof)=[];
   if rowv~=0
      pvdof=bcnv(:,1);
      nvp=bcnv(:,2);
      fvdof(pvdof)=[];
      if rowi~=0
         pidof=bcim(:,1);
         imp=bcim(:,2);
         HG=G;
         HG(:,pvdof)=0;
         for s=1:rowi
            HG(:,pidof(s))=HG(:,pidof(s))-H(:,pidof(s)).*imp(s);
         end
         HH=H;
         HH(:,fvdof)=0;
         x=(HG-HH)\(H(:,ppdof)*prp-G(:,pvdof)*nvp);
         nv(pvdof)=nvp;
         nv(pidof)=x(pidof);
         nv(fvdof)=x(fvdof);
         pr(ppdof)=prp;
         pr(fpdof)=x(fpdof);
         pr(pidof)=nv(pidof).*imp;
      else
         HG=G;
```

```
         HG(:,pvdof)=0;
         HH=H;
         HH(:,ppdof)=0;
         x=(HG-HH)\(H(:,ppdof)*prp-G(:,pvdof)*nvp);
         pr(ppdof)=prp;
         pr(fpdof)=x(fpdof);
         nv(pvdof)=nvp;
         nv(fvdof)=x(fvdof);
      end
   elseif rowi~=0
      pidof=bcim(:,1);
      imp=bcim(:,2);
      HG=G;
      for s=1:rowi
         HG(:,pidof(s))=HG(:,pidof(s))-H(:,pidof(s)).*imp(s);
      end
      x=HG\(H(:,ppdof)*prp);
      nv=x;
      pr(ppdof)=prp;
      pr(pidof)=nv(pidof).*imp;
   else
      x=G\H*prp;
      pr=prp;
      nv=x;
   end
else
   pvdof=bcnv(:,1);
   nvp=bcnv(:,2);
   fvdof(pvdof)=[];
   if rowi~=0
      pidof=bcim(:,1);
      imp=bcim(:,2);
      HH=H;
      for s=1:rowi
         HH(:,pidof(s))=HH(:,pidof(s))-G(:,pidof(s))./imp(s);
      end
      x=HH\(G(:,pvdof)*nvp);
      pr=x;
      nv(pvdof)=nvp;
      nv(pidof)=pr(pidof)./imp;
   else
      x=H\G*nvp;
      nv=nvp;
      pr=x;
   end
end
%-----------------------------------end--------------------------------
```

```
function p=bem_acouspost(coord,ex,ey,ez,ep,pr,nv,edof,n)
% p=bem_acouspost(coord,ex,ey,ez,ep,pr,nv,edof,n)
% p=bem_acouspost(coord,ex,ey,ez,ep,pr,nv,edof)
%-----------------------------------------------------------------------
% PURPOSE
%  Compute the pressure p at an arbitrary point in the 3D-space.
%
% INPUT:   coord= [x y z]          coordinates for where the pressure p
%                                  will be calculated
%          ex= [x11 x12 x13 x14
%                 .   .   .   .
%               xn1 xn2 xn3 xn4]
%          ey= [y11 y12 y13 y14
%                 .   .   .   .
%               yn1 yn2 yn3 yn4]
%          ez= [z11 z12 z13 z14
%                 .   .   .   .
%               zn1 zn2 zn3 zn4]   element node coordinates
%
%          ep= [w c rho]           problem properties
%                                  w: angular frequency
%                                  c: speed of sound in acoustic medium
%                                  rho: density of acoustic medium
%
%          pr= nel x 1 matrix      pressure at element nodes
%          nv= nel x 1 matrix      normal velocity at element nodes
%        edof= nel x ned+1 matrix  topology matrix
%                                  nel= number of elements
%                                  ned= number of element dof's
%
%          n=[value]               normal direction
%                                  value=1 default
%                                       -1 reverse
%
% OUTPUT:  p:   pressure in the 3D-space
%-----------------------------------------------------------------------
rev=1;
if nargin==9
   rev=n;
end k=ep(1)/ep(2);

%****Gauss points****
x=coord(1); y=coord(2); z=coord(3); ga=0.577350269189626;
xi=[-ga; ga; ga; -ga];      eta=[-ga; -ga; ga; ga];
N(:,1)=1/4*(xi-1).*(eta-1); N(:,2)=-1/4*(xi+1).*(eta-1);
N(:,3)=1/4*(xi+1).*(eta+1); N(:,4)=-1/4*(xi-1).*(eta+1);
dNr(1:2:7,1)=-(1-eta)/4;     dNr(1:2:7,2)= (1-eta)/4;
```

```
dNr(1:2:7,3)= (1+eta)/4;      dNr(1:2:7,4)=-(1+eta)/4;
dNr(2:2:8,1)=-(1-xi)/4;       dNr(2:2:8,2)=-(1+xi)/4;
dNr(2:2:8,3)= (1+xi)/4;       dNr(2:2:8,4)= (1-xi)/4;

[nel,col]=size(edof); if col==2
   edof(:,3:5)=0;
end [dof,col]=size(pr); G=zeros(1,dof); H=zeros(1,dof); for
s=1:nel

   %****Element area****
   JTxy=dNr*[ex(s,:);ey(s,:)]';
   JTyz=dNr*[ey(s,:);ez(s,:)]';
   JTzx=dNr*[ez(s,:);ex(s,:)]';

   detJxy=[det(JTxy(1:2,:));det(JTxy(3:4,:));det(JTxy(5:6,:))...
      ;det(JTxy(7:8,:))];
   detJyz=[det(JTyz(1:2,:));det(JTyz(3:4,:));det(JTyz(5:6,:))...
      ;det(JTyz(7:8,:))];
   detJzx=[det(JTzx(1:2,:));det(JTzx(3:4,:));det(JTzx(5:6,:))...
      ;det(JTzx(7:8,:))];
   A=[sqrt(detJxy.^2+detJyz.^2+detJzx.^2)];

   %****Normal vector****
   a=[ex(s,2)-ex(s,1) ey(s,2)-ey(s,1) ez(s,2)-ez(s,1)];
   b=[ex(s,4)-ex(s,1) ey(s,4)-ey(s,1) ez(s,4)-ez(s,1)];
   n=[a(2)*b(3)-a(3)*b(2); a(3)*b(1)-a(1)*b(3); a(1)*b(2)-a(2)*b(1)];
   n=rev*n/sqrt(n'*n);

   if edof(s,5)==0

      %****Constant elements****
      A=sum(A);
      mid=[sum(ex(s,:))/4 sum(ey(s,:))/4 sum(ez(s,:))/4];
      dis=sqrt((x-mid(1))^2+(y-mid(2))^2+(z-mid(3))^2);
      Ge=i*ep(3)*ep(1)*A*exp(-i*k*dis)/(4*pi*dis);

      h1=-(mid(1)-x)*exp(-i*k*dis)/(4*pi*dis^2)*(i*k+1/dis);
      h2=-(mid(2)-y)*exp(-i*k*dis)/(4*pi*dis^2)*(i*k+1/dis);
      h3=-(mid(3)-z)*exp(-i*k*dis)/(4*pi*dis^2)*(i*k+1/dis);
      He=[h1 h2 h3]*n*A;
      G(edof(s,2))=G(edof(s,2))+Ge;
      H(edof(s,2))=H(edof(s,2))+He;
   else

      %****Linear elements****
      xg=N*ex(s,:)';    yg=N*ey(s,:)';  zg=N*ez(s,:)';
      xdis=xg-x;    ydis=yg-y;  zdis=zg-z;
```

```
      dis=sqrt(xdis.^2+ydis.^2+zdis.^2);
      g=i*ep(3)*ep(1)*exp(-i*k*dis)./(4*pi*dis);
      Ge(1,1)=sum(g.*N(:,1).*A);
      Ge(1,2)=sum(g.*N(:,2).*A);
      Ge(1,3)=sum(g.*N(:,3).*A);
      Ge(1,4)=sum(g.*N(:,4).*A);
      G(edof(s,2:5))=G(edof(s,2:5))+Ge;

      h1=-xdis.*exp(-i*k*dis)./(4*pi*dis.^2).*(i*k+1./dis);
      h2=-ydis.*exp(-i*k*dis)./(4*pi*dis.^2).*(i*k+1./dis);
      h3=-zdis.*exp(-i*k*dis)./(4*pi*dis.^2).*(i*k+1./dis);
      h=[h1 h2 h3]*n;
      He(1,1)=sum(h.*N(:,1).*A);
      He(1,2)=sum(h.*N(:,2).*A);
      He(1,3)=sum(h.*N(:,3).*A);
      He(1,4)=sum(h.*N(:,4).*A);
      H(edof(s,2:5))=H(edof(s,2:5))+He;
   end
end
p=G*nv+H*pr;
%-----------------------------------end-------------------------------
```

Appendix B

BEM Problem

```
%--------------------------Example, BEM-----------------------------%

%******Node coordinates******
coord=[1 0 0;
   0.9239 0.3827 0;
   0.8534 0.3687 0.3687;
   0.9239 0 0.3827;
   0.6587 0.3727 0.6587;
   0.7071 0 0.7071;
   0.3687 0.3687 0.8534;
   0.3827 0 0.9239;
   0 0.3827 0.9239;
   0 0 1;
   0.7071 0.7071 0;
   0.6587 0.6587 0.3727;
   0.5774 0.5774 0.5774;
   0.3727 0.6587 0.6587;
   0 0.7071 0.7071;
   0.3827 0.9239 0;
   0.3687 0.8534 0.3687;
   0 1 0;
   0 0.9239 0.3827];

%******Element coordinates******
ex8=[1 0.9239 0.8534 0.9239;
   0.9239 0.7071 0.6587 0.8534;
   0.7071 0.3827 0.3687 0.6587;
   0.3827 0 0 0.3687;
   0.9239 0.8534 0.6587 0.7071;
   0.8534 0.6587 0.5774 0.6587;
   0.6587 0.3687 0.3727 0.5774;
   0.3687 0 0 0.3727;
```

```
   0.7071 0.6587 0.3687 0.3827;
   0.6587 0.5774 0.3727 0.3687;
   0.3727 0 0 0.3687;
   0.3827 0.3687 0 0];

ex4=[ex8;ex8]; ex2=[ex4;-ex4]; ex=[ex2;ex2];

ey8=[0 0.3827 0.3687 0;
   0.3827 0.7071 0.6587 0.3687;
   0.7071 0.9239 0.8534 0.6587;
   0.9239 1 0.9239 0.8534;
   0 0.3687 0.3727 0;
   0.3687 0.6587 0.5774 0.3727;
   0.6587 0.8534 0.6587 0.5774;
   0.8534 0.9239 0.7071 0.6587;
   0 0.3727 0.3687 0;
   0.3727 0.5774 0.6587 0.3687;
   0.6587 0.7071 0.3827 0.3687;
   0 0.3687 0.3827 0];

ey4=[ey8;-ey8]; ey2=[ey4;ey4]; ey=[ey2;ey2];

ez8=[0 0 0.3687 0.3827;
   0 0 0.3727 0.3687;
   0 0 0.3687 0.3727;
   0 0 0.3827 0.3687;
   0.3827 0.3687 0.6587 0.7071;
   0.3687 0.3727 0.5774 0.6587;
   0.3727 0.3687 0.6587 0.5774;
   0.3687 0.3827 0.7071 0.6587;
   0.7071 0.6587 0.8534 0.9239;
   0.6587 0.5774 0.6587 0.8534;
   0.6587 0.7071 0.9239 0.8534;
   0.9239 0.8534 0.9239 1];

ez4=[ez8;ez8]; ez2=[ez4;ez4]; ez=[ez2;-ez2];

%******Element dof matrix******
edof=[1 1 2 3 4;
   2 2 11 12 3;
   3 11 16 17 12;
   4 16 18 19 17;
   5 4 3 5 6;
   6 3 12 13 5;
   7 12 17 14 13;
   8 17 19 15 14;
   9 6 5 7 8;
```

```
   10 5 13 14 7;
   11 14 15 9 7;
   12 8 7 9 10];

%******Properties of acoustic medium******
%[angular frequency, sound velocity, density]
ep=[3 1 1];

%*****Reversed element normal direction******
n=ones(96,1); n(13:36)=-1; n(49:60)=-1; n(85:96)=-1;

%******Assemble influence matrices******
H=zeros(19);    G=H;
for k=1:19
   r=1;
   for j=1:96
      if r==13
         r=1;
      end
      [He,Ge]=bem_infl4q(coord(k,:),ex(j,:),ey(j,:),ez(j,:),ep,n(j));
      H=bem_assem(edof,H,He,k,r);
      G=bem_assem(edof,G,Ge,k,r);
      r=r+1;
   end
end

%******Space angle constants******
a=bem_spacang(coord(3,:),ex([1 2 5 6],:),ey([1 2 5 6]...
   ,:),ez([1 2 5 6],:));
b=bem_spacang(coord(13,:),ex([6 7 10],:),ey([6 7 10]...
   ,:),ez([6 7 10],:));
c=bem_spacang(coord(5,:),ex([5 6 9 10],:),...
   ey([5 6 9 10],:),ez([5 6 9 10],:));
d=bem_spacang(coord(10,:),ex([12 24 36 48]...
   ,:),ey([12 24 36 48],:),ez([12 24 36 48],:));
e=bem_spacang(coord(8,:),ex([9 12 21 24],:)...
   ,ey([9 12 21 24],:),ez([9 12 21 24],:));
f=bem_spacang(coord(6,:),ex([5 9 17 21],:)...
   ,ey([5 9 17 21],:),ez([5 9 17 21],:));

C(1,1)=d;       C(2,2)=e;       C(3,3)=a;       C(5,5)=c;
C(10,10)=d;     C(4,4)=e;       C(7,7)=a;       C(12,12)=c;
C(18,18)=d;     C(8,8)=e;       C(17,17)=a;     C(14,14)=c;
                C(9,9)=e;
                C(16,16)=e;
                C(19,19)=e;
C(6,6)=f;       C(13,13)=b;
```

```
C(11,11)=f;
C(15,15)=f;
H=H+C;

%******Boundary condition matrices******
for t=1:19
   bcnv(t,1)=t;
end
bcnv(:,2)=1;
bcpr=[];    bcim=[];

%*****Solve the BEM model******
[pr,nv]=bem_solveq(G,H,bcpr,bcnv,bcim);

%-------------------------------end------------------------------------
```

Appendix C

BEM/FEM Functions

```
function [Te]=bem_velotrans(ex,ey,ez,ep)
% [Te]=bem_velotrans(ex,ey,ez,ep)
%-------------------------------------------------------------------
% PURPOSE
%  Compute transformation vector to connect the normal velocity
%  of a BE node with the translation of a FE node.
%
% INPUT:  ex = [x1 x2 x3 x4]   coordinates of the element in
%         ey = [y1 y2 y3 y4]   which the nodes are placed
%         ez = [z1 z2 z3 z4]
%
%         ep = value           value=1: 24 node shell element
%                              value=2: 12 node plate element
%                                       (ez=[0 0 0 0])
%
% OUTPUT: Te:  velocity coupling vector (1 x 3)
%-------------------------------------------------------------------
p(1,:)=[ex(2)-ex(1) ey(2)-ey(1) ez(2)-ez(1)];
p(2,:)=[ex(4)-ex(1) ey(4)-ey(1) ez(4)-ez(1)];
L=sqrt(p*p');
n=[p(1,:)/L(1,1); p(2,:)/L(2,2)];
norm=[n(1,2)*n(2,3)-n(1,3)*n(2,2);
   n(1,3)*n(2,1)-n(1,1)*n(2,3);
   n(1,1)*n(2,2)-n(1,2)*n(2,1)]';
Te=norm/sqrt(norm*norm');
if ep==2
   Te=1;
end
%-------------------------------end---------------------------------
```

```
function Le=bem_ptrans(ex,ey,ez,ep)
% Le=bem_ptrans(ex,ey,ez,ep)
%-----------------------------------------------------------------------
% PURPOSE
%  Compute transformation vector to connect pressure between a BE
%  element and a FE element.
%
% INPUT:  ex = [x1 x2 x3 x4]   Element coordinates for the BE and the
%         ey = [y1 y2 y3 y4]   FE elements. Notice that the coordinates
%         ez = [z1 z2 z3 z4]   are the same for both elements.
%
%         ep = value           value=1: 24 node shell element
%                              value=2: 12 node plate element
%                                       (ez=[0 0 0 0])
%
% OUTPUT: Le:  pressure coupling vector (24 x 4) or (12 x 4)
%-----------------------------------------------------------------------

p1=[ex(2)-ex(1); ey(2)-ey(1); ez(2)-ez(1)];
p2=[ex(4)-ex(1); ey(4)-ey(1); ez(4)-ez(1)];
norm=[p1(2)*p2(3)-p1(3)*p2(2);
p1(3)*p2(1)-p1(1)*p2(3);
p1(1)*p2(2)-p1(2)*p2(1)];
norm=norm/sqrt(norm'*norm);

%****Gauss points****
g1=0.577350269189626;
xi=[-g1; g1; g1;-g1];
eta=[-g1;-g1; g1; g1];

N(:,1)=1/4*(xi-1).*(eta-1); N(:,2)=-1/4*(xi+1).*(eta-1);
N(:,3)=1/4*(xi+1).*(eta+1); N(:,4)=-1/4*(xi-1).*(eta+1);

dNr(1:2:7,1)=-(1-eta)/4;     dNr(1:2:7,2)= (1-eta)/4;
dNr(1:2:7,3)= (1+eta)/4;     dNr(1:2:7,4)=-(1+eta)/4;
dNr(2:2:8,1)=-(1-xi)/4;      dNr(2:2:8,2)=-(1+xi)/4;
dNr(2:2:8,3)= (1+xi)/4;      dNr(2:2:8,4)= (1-xi)/4;

JTxy=dNr*[ex;ey]'; JTyz=dNr*[ey;ez]'; JTzx=dNr*[ez;ex]';

Le=zeros(24,4); for k=1:4
   L=[N(k,1) 0 0;
      0 N(k,1) 0;
      0 0 N(k,1);
      zeros(3);
      N(k,2) 0 0;
      0 N(k,2) 0;
```

```
      0 0 N(k,2);
      zeros(3);
      N(k,3) 0 0;
      0 N(k,3) 0;
      0 0 N(k,3);
      zeros(3)
      N(k,4) 0 0;
      0 N(k,4) 0;
      0 0 N(k,4);
      zeros(3)]*norm*N(k,:);
   indx=[ 2*k-1; 2*k ];
   detJxy=det(JTxy(indx,:));
   detJyz=det(JTyz(indx,:));
   detJzx=det(JTzx(indx,:));
   detJ=sqrt(detJxy^2+detJyz^2+detJzx^2);
   Le=L*detJ+Le;
end
if ep==2
   Le=[Le(3:5,:); Le(9:11,:); Le(15:17,:); Le(21:23,:)];
end
%--------------------------------end------------------------------------
```

```
function L=bem_assempres(L,Le,bedof,fedof,con)
% L=bem_assempres(L,Le,bedof,fedof,con)
%-------------------------------------------------------------------------
% PURPOSE
%  Assemble the transformation matrices that connects pressure between
%  BE and FE elements.
%
% INPUT:    L:      Global transformation matrix
%           Le:     Local transformation matrix
%           bedof:  dof topology vector for the BE element
%           fedof:  dof topology vector for the FE element
%
%           con = (non x 1)   vector containing BE nodes that are
%                             connected with FE nodes
%
% OUTPUT:  L:      New global transformation matrix
%-------------------------------------------------------------------------
[row,col]=size(fedof);
t1=abs(con-bedof(2)); t2=abs(con-bedof(3)); t3=abs(con-bedof(4));
t4=abs(con-bedof(5));
[val,p(1)]=min(t1); [val,p(2)]=min(t2);
[val,p(3)]=min(t3); [val,p(4)]=min(t4);
L(fedof(1,2:col),p)=L(fedof(1,2:col),p)+Le;
%-------------------------------end-------------------------------------
```

```
function T=bem_assemvel(T,Te,bn,fn,con)
% T=bem_assemvel(T,Te,bn,fn,con)
%-----------------------------------------------------------------------
% PURPOSE
%  Assemble the transformation vectors that connects BE-velocity with
%  FE-translations.
%
% INPUT:   T:       Global transformation matrix
%          Te:      Local transformation vector
%          bn:      Number of the actual BE node
%
%          fn = [xn yn zn]    numbers of the FE translation degrees of
%                             freedom
%
%          con = (non x 1)    vector containing BE nodes that are
%                             connected with FE nodes
%
% OUTPUT:   T:    New global transformation matrix
%-----------------------------------------------------------------------
t=abs(con-bn);
[val,p]=min(t);
T(p,fn)=T(p,fn)+Te;
%-------------------------------end------------------------------------
```

```
function [Couple,f1,f2]=bem_coupassem(K,C,M,L,T,H,G,ep,p1)
% [Couple,f1,f2]=bem_coupassem(K,C,M,L,T,H,G,ep,p1)
%-------------------------------------------------------------------------
% PURPOSE
%  Assemble the coupled FE/BE system matrices.
%
% INPUT:  K,C,M:  FE stifness, damping and mass matrices
%         L,T:    FE/BE pressure and velocity coupling matrices
%         H,G:    BE influence matrices
%
%         ep = [rho w]     problem properties
%                          rho: density of acoustic medium
%                          w: angular frequency
%
% OUTPUT:  Couple,f1,f2:    Coupled FE/BE system matrices
%-------------------------------------------------------------------------
rho=ep(1);  w=ep(2); [row1,col1]=size(K); [row2,col2]=size(H);
[row3,col3]=size(L); Couple=zeros(col1+col2);
Couple(1:row1,1:col1)=K+i*w*C-w^2*M;
Couple(1:row1,col1+1:col1+col3)=L;
n=0; m=1; Hnew=H; Gnew=G;
for
k=1:col3
   if k~=p1(k)
      Hnew(:,k)=H(:,p1(k)); Gnew(:,k)=G(:,p1(k));
      if (p1(k)-n)>1
         for p=1:(p1(k)-n-1)
            Hnew(:,col3+m)=H(:,p1(k)-(p1(k)-n)+p);
            Gnew(:,col3+m)=G(:,p1(k)-(p1(k)-n)+p);
            m=m+1;
         end
      end
   end
   n=p1(k);
end
Couple(row1+1:row1+col3,1:col1)=-i*w*Gnew(1:col3,1:col3)*T;
Couple(row1+col3+1:row1+row2,1:col1)=...
   -i*w*Gnew(col3+1:row2,1:col3)*T;
Couple(row1+1:row1+col3,col1+1:col1+col3)=Hnew(1:col3,1:col3);
Couple(row1+1:row1+col3,col1+col3+1:col1+col2)=...
   Hnew(1:col3,col3+1:col2);
Couple(row1+col3+1:row1+row2,col1+1:col1+col3)=...
   Hnew(col3+1:row2,1:col3);
Couple(row1+col3+1:row1+row2,col1+col3+1:col1+col2)=...
   Hnew(col3+1:row2,col3+1:row2);
f1=Gnew(1:col3,col3+1:row2); f2=Gnew(col3+1:row2,col3+1:row2);
%-------------------------------end-------------------------------------
```

```
function
[pr,nv]=bem_coupsolveq(Couple,f1,f2,T,con,bc,f,bcpr,bcnv,bcim,ep)
% [pr,nv]=bem_coupsolveq(Couple,f1,f2,T,con,bc,f,bcpr,bcnv,bcim,ep)
%-------------------------------------------------------------------
% PURPOSE
%  Solve coupled FE/BE equation system
%
% INPUT:   Couple, f1, f2 : output from function bem\_coupassem
%
%          con  : vector giving the number of the BE nodes that are
%                 connected with FE nodes
%            f  : force vector for the FE degrees of freedom
%
%     Boundary conditions on uncoupled BE nodes
%           bcpr:   pressure
%           bcnv:   normal velocity
%           bcim:   acoustic impedance
%
%     Boundary conditions on FE nodes
%           bc
%
% OUTPUT:  pr:  solution including boundary values (pressure)
%          nv:  solution including boundary values (normal velocity)
%-------------------------------------------------------------------
w=ep;
[ro1,co1]=size(Couple); [ro2,co2]=size(f1); [ro3,co3]=size(f2);

[rowp,colp]=size(bcpr);
[rowv,colv]=size(bcnv);
[rowi,coli]=size(bcim);

n=0; m=1; fdof=[1:ro3]';
for k=1:ro2
   [val,pos]=min(abs(fdof-con(k)));
   fdof(pos:ro3)=fdof(pos:ro3)+1;
end
known=fdof;

Couple1=Couple;
if rowp~=0
   for k=1:rowp
      [val,pos]=min(abs(fdof-bcpr(k,1)));
      known(pos,2)=bcpr(k,2);
      Couple(ro1-ro2-ro3+1:ro1-ro3,co1-co3+pos)=-f1(:,pos);
      f1(:,pos)=-Couple1(ro1-ro2-ro3+1:ro1-ro3,co1-co3+pos);
      Couple(ro1-ro3+1:ro1,co1-co3+pos)=-f2(:,pos);
      f2(:,pos)=-Couple1(ro1-ro3+1:ro1,co1-co3+pos);
```

```
   end
end

if rowi~=0
   for k=1:rowi
      [val,pos]=min(abs(fdof-bcim(k,1)));
      Couple(ro1-ro2-ro3+1:ro1-ro3,co1-co3+pos)=...
         Couple(ro1-ro2-ro3+1:ro1-ro3,co1-co3+pos)*bcim(k,2)-f1(:,pos);
      Couple(ro1-ro3+1:ro1,co1-co3+pos)=...
         Couple(ro1-ro3+1:ro1,co1-co3+pos)*bcim(k,2)-f2(:,pos);
   end
end

if rowv~=0
   for k=1:rowv
      [val,pos]=min(abs(fdof-bcnv(k,1)));
      known(pos,2)=bcnv(k,2);
   end
end

if co2==0
   [row4,col4]=size(con);
   f=[f;zeros(row4,1)];
   nv=zeros(row4,1);   pr=zeros(row4,1);
else
   f11=f1*known(:,2);   f22=f2*known(:,2);
   f=[f;f11;f22];
   nv=zeros(ro2+ro3,1);   pr=zeros(ro2+ro3,1);
end

fd=[1:ro1]';
d=zeros(size(fd));
pd=bc(:,1);
dp=bc(:,2);
fd(pd)=[];

s=Couple(fd,fd)\(f(fd)-Couple(fd,pd)*dp);

d(pd)=dp; d(fd)=s;

if co2==0
   nv(con)=i*w*T*d(1:ro1-row4);
   pr(con)=d(ro1-row4+1:ro1);
else
   nv(con)=i*w*T*d(1:ro1-ro2-ro3);
   pr(con)=d(ro1-ro2-ro3+1:ro1-ro3);
end
```

```
for k=1:ro3
   t=fdof(k);
   if rowv~=0 & rowp~=0 & rowi~=0
      [valv,posv]=min(abs(bcnv(:,1)-t));
      [valp,posp]=min(abs(bcpr(:,1)-t));
      [vali,posi]=min(abs(bcim(:,1)-t));
      if valv==0
         nv(t)=bcnv(posv,2);
         pr(t)=d(ro1-ro3+k);
      elseif valp==0
         pr(t)=bcpr(posp,2);
         nv(t)=d(ro1-ro3+k);
      else
         nv(t)=d(ro1-ro3+k);
         pr(t)=nv(t)*bcim(posi,2);
      end
   elseif rowv~=0 & rowp~=0
      [valv,posv]=min(abs(bcnv(:,1)-t));
      [valp,posp]=min(abs(bcpr(:,1)-t));
      if valv==0
         nv(t)=bcnv(posv,2);
         pr(t)=d(ro1-ro3+k);
      else
         pr(t)=bcpr(posp,2);
         nv(t)=d(ro1-ro3+k);
      end
   elseif rowp~=0 & rowi~=0
      [valp,posp]=min(abs(bcpr(:,1)-t));
      [vali,posi]=min(abs(bcim(:,1)-t));
      if valp==0
         pr(t)=bcpr(posp,2);
         nv(t)=d(ro1-ro3+k);
      else
         nv(t)=d(ro1-ro3+k);
         pr(t)=nv(t)*bcim(posi,2);
      end
   elseif rowv~=0 & rowi~=0
      [valv,posv]=min(abs(bcnv(:,1)-t));
      [vali,posi]=min(abs(bcim(:,1)-t));
      if valv==0
         nv(t)=bcnv(posv,2);
         pr(t)=d(ro1-ro3+k);
      else
         nv(t)=d(ro1-ro3+k);
         pr(t)=nv(t)*bcim(posi,2);
      end
```

```
   elseif rowv~=0
      [valv,posv]=min(abs(bcnv(:,1)-t));
      nv(t)=bcnv(posv,2);
      pr(t)=d(ro1-ro3+k);
   elseif rowp~=0
      [valp,posp]=min(abs(bcpr(:,1)-t));
      pr(t)=bcpr(posp,2);
      nv(t)=d(ro1-ro3+k);
   end
end
%--------------------------------end------------------------------------
```

Appendix D

BEM/FEM Problems

```
%--------------------Example FEM/BEM coupling-----------------------%

inch=0.0254;
t=0.064*inch;                              %Plate thickness
E=70e9;                                    %Modulus of elasticity, al
ny=0.3;                                    %Poissons number, al
raa=2690;                                  %Density, al
ir=3;                                      %Integration rule
a0=0;                                      %Damping factor
a1=0;                                      %Damping factor
ep=[t E ny raa ir a0 a1];
Lx=12*inch;Ly=6.001*inch;Lz=5.999*inch; %Box dimensions
nelx=12;nely=6;nelz=6;                     %Number of elements

disx=Lx/nelx; disy=Ly/nely; disz=Lz/nelz;
K=zeros(3*(nelx+1)*(nely+1)); M=K; C=K; el=0; ypos=0;

%******FEM coordinate, and dof matrices******
for i=1:nely
   xpos=0;
   for j=1:nelx
      el=el+1;
      fex(el,:)=[xpos*disx (xpos+1)*disx (xpos+1)*disx xpos*disx];
      fey(el,:)=[ypos*disy ypos*disy (ypos+1)*disy (ypos+1)*disy];
      a=xpos*3+3*(nelx+1)*ypos+1;
      b=a+3*(1+nelx);
      femedof(el,:)=[el a a+1 a+2 a+3 a+4 a+5 b+3 b+4 b+5 b b+1 b+2];
      xpos=xpos+1;
   end
   ypos=ypos+1;
end
```

```
%******FEM stifness and mass matrices******
[Ke,Me]=plateqd(fex(1,:),fey(1,:),ep);
K=assem(femedof,K,Ke);
M=assem(femedof,M,Me);

%******Create BEM dof, coordinate, and coordinate matrices******
                %***(This part is left out)***

%******Number of BEM elements and nodes******
node_n=2*(nelx+1)*(nely+1)+2*(nely+1)*(nelz+1)+2*(nelz+1)*(nelx+1);
el_n=2*nelx*nely+2*nely*nelz+2*nelz*nelx;

am1=(nelx+1)*(nely+1);am2=(nely+1)*(nelz+1);am3=(nelz+1)*(nelx+1);

%******Box corner nodes******
corner(1:4)=[1 nelx+1 am1-nelx am1]; corner(5:8)=am1+[1 nely+1
am2-nely am2]; corner(9:12)=am1+am2+[1 nelx+1 am3-nelx am3];
corner(13:16)=am1+am2+am3+[1 nelx+1 am3-nelx am3];
corner(17:20)=am1+am2+2*am3+[1 nely+1 am2-nely am2];
corner(21:24)=am1+2*am2+2*am3+[1 nelx+1 am1-nelx am1];

%******Box edge nodes******
edgec=0; for i=1:nely+1
   for j=1:nelx+1
      if i==1 | j==1 | i==nely+1 |j==nelx+1
         edgec=edgec+1;
         edge(edgec)=(i-1)*(nelx+1)+j;
      end
   end
end

for i=1:nelz+1
   for j=1:nely+1
      if i==1 | j==1 | i==nelz+1 |j==nely+1
         edgec=edgec+1;
         edge(edgec)=am1+(i-1)*(nely+1)+j;
      end
   end
end

for i=1:nelz+1
   for j=1:nelx+1
      if i==1 | j==1 | i==nelz+1 |j==nelx+1
         edgec=edgec+1;
         edge(edgec)=am1+am2+(i-1)*(nelx+1)+j;
      end
   end
```

```
end edge=[edge am1+am2+am3+edge];

%******FEM boundary condition matrix******
con=0; for a=1:nely+1
   for b=1:nelx+1
      if a==1 | a==nely+1 | b==1 | b==nelx+1
         con=con+1;
         no=(a-1)*(nelx+1)+b;
         bc(con,:)=[no*3-2 0];
      end
   end
end

%******BEM velocity and FEM translation coupling******
T=zeros(am1,3*am1); Te=1; for i=1:am1
   T=bem_assemvel(T,Te,i,i*3-2,[1:am1]');
end

%******BEM pressure and FEM force coupling******
L=zeros(3*am1,am1); for i=1:nelx*nely
   Le=bem_ptrans(bex(i,:),bey(i,:),bez(i,:),2);
   L=bem_assempres(L,Le,bemedof(i,:),femedof(i,:),[1:am1]');
end

%******Force vector******
f=zeros(3*am1,1); p=1; A=disx*disy; f0=p*A; f(1:3:3*am1)=f0;

%******BEM boundary condition matrices******
bcnv(:,1)=[am1+1:2*(am1+am2+am3)]'; bcnv(:,2)=0;

step=10; rev=1000/step; for loop=1:rev

   %******Properties for the acoustic medium******
   %[angular frequency, sound velocity, density]
   bep=[loop*step*2*pi 340 1.21];
   G=zeros(node_n);
   H=G;

   %******Assemble the influence matrices******
   for i=1:node_n
      for j=1:el_n
         [He,Ge]=bem_infl4q(coord(i,:),bex(j,:),bey(j,:),bez(j,:),bep);
         H=bem_assem(bemedof,H,He,i,j);
         G=bem_assem(bemedof,G,Ge,i,j);
      end
   end
   H=H+0.5*diag(ones(node_n,1));
```

```
   for i=1:2*edgec
      H(edge(i),edge(i))=1/4;
   end

   for i=1:24
      H(corner(i),corner(i))=1/8;
   end

   %******Assemble and solve the coupled model******
   [Couple,f1,f2]=bem_coupassem(K,C,M,L,T,H,G,[bep(3) bep(1)],[1:am1]');
   [pr,nv]=bem_coupsolveq(Couple,f1,f2,T,[1:am1]',bc,f,[],bcnv,[],bep(1));
   pres(:,loop)=pr;
   velo(:,loop)=nv;
   p_norm(loop)=pr(round((nelx+1)*(nely+1)/2))*conj(pr(round(...
      (nelx+1)*(nely+1)/2)))/2;
   pDb(loop)=10*log10(p_norm(loop)/2e-5^2);
   p_m=bem_acouspost([Lx/2 Ly/2 Lz/2],bex,bey,bez,bep,pr,nv,bemedof);
   p_mnorm(loop)=p_m*conj(p_m)/2;
   pDb_mitt(loop)=10*log10(p_mnorm(loop)/2e-5^2);
end
%-------------------------------end-----------------------------------
```