2014-06-01

Application of an Isogeometric Boundary Element Method to the Calculation of Acoustic Radiation Modes and Their Efficiencies

Candice Marie Humpherys
Brigham Young University - Provo

Application of an Isogeometric Boundary Element Method

to the Calculation of Acoustic Radiation Modes

and Their Efficiencies

Candice Humpherys

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Scott D. Sommerfeldt, Chair
Derek C. Thomas
Kent L. Gee

Department of Physics and Astronomy

Brigham Young University

June 2014

ABSTRACT

Application of an Isogeometric Boundary Element Method
to the Calculation of Acoustic Radiation Modes
and Their Efficiencies

Candice Humpherys
Department of Physics and Astronomy, BYU
Master of Science

In contrast to the structural modes, which describe the physical motion of vibrating structures, acoustic radiation modes describe the radiated sound power. Radiation modes are beneficial in active noise control because reducing an efficiently radiating radiation mode guarantees the reduction of radiated sound power. Much work has been done to calculate the radiation modes for simple geometries, where analytic solutions are available. In this work, isogeometric analysis (IGA) is used to provide a tool capable of analyzing the radiation modes of arbitrarily complex geometries. IGA offers increased accuracy and efficiency by using basis functions generated from Non-Uniform Rational B-Splines (NURBS) or T-Splines, which can represent geometries exactly. Results showing this increased accuracy and efficiency with IGA using T-Splines are shown for a sphere to validate the method, comparing with the exact analytical solution as well as results from a traditional boundary element method. A free cylindrical shell is also analyzed to show the usefulness of this method. Expected similarities, as well as expected differences, are observed between this free shell and a baffled cylindrical shell.

ACKNOWLEDGMENTS

Contents

List of Figures

Chapter 1

Introduction

The concept of describing the vibration of a structure in terms of acoustic radiation modes has been around since the early 1990s [1]. Radiation modes are useful in active noise control, source reconstruction, and design optimization [2–4]. One benefit of these modes over structural modes is the relatively small number of them that are required to obtain an accurate measure of the radiated sound power [1, 2, 4–8]. Due to the difficulty of accurately representing complex geometries, research has been limited to simple geometries and this field has become mostly dormant as of the late 1990s.

Isogeometric analysis (IGA) was introduced in the early 2000s as a tool to improve the modeling and subsequent analysis of arbitrarily complex geometries [9–11]. Due to its use of a smooth basis, IGA offers improved and more efficient analysis. IGA has been applied - and shown benefits - in areas such as turbulence, fluid-structure interaction, incompressibility, structural analysis, shells, and phase-field analysis, and is still an active area of research [12–14].

1.1 Radiation Modes

The analysis of vibrating structures is easily done in terms of structural modes, also known as the normal modes of vibration. These modes describe the vibrating motion of the structure. However, analysis is often geared towards understanding the radiated sound power from the vibrating structure, and these structural modes do not describe the sound power radiation in a way that allows for quick understanding or control of the radiated sound power.

Radiation modes have been developed as a way to express the sound power radiation in terms of a set of modes that are orthogonal with respect to the sound power radiation, and thus contribute independently to the radiated sound power [1–8, 15–17]. Due to this independence, there is a direct and intuitive connection between a radiation mode and the radiated sound power. There is no coupling between radiation modes like there is between structural modes. All this makes it easier to understand, calculate, and control the radiated power [2, 3, 8].

There are multiple ways of obtaining the acoustic radiation modes for a structure. The most common is the elemental radiator approach, which represents the structure as an array of individual point sources [3, 4]. Another common method is to represent the surface velocity in terms of an arbitrary set of complete, orthogonal basis functions, the most common choice being the structural modes [4, 15, 16, 18, 19]. Yet another approach, though less common in the literature, is to use a boundary element method (BEM) to obtain the needed quantities to determine the radiated sound power in terms of radiation modes [8].

Regardless of the approach used, the radiated sound power can be represented in the form

$$\Pi = \mathbf{v}^H \mathbf{R} \mathbf{v} \tag{1.1}$$

where $\mathbf{v}$ is the surface velocity distribution and $\mathbf{R}$ is a matrix that contains information about the impedance and coupling of the structure. Determining the radiation modes involves the diagonalization of this $\mathbf{R}$ matrix. This decouples the system, representing the radiated sound power in terms

of orthogonal radiation modes (eigenvectors) and their radiation efficiencies (proportional to the eigenvalues) [1–3, 17].

If structural modes are used, an expression similar to Equation (1.1) can be written, where **R** contains information about the coupling between structural modes. This matrix, however, is difficult to calculate, and cannot be diagonalized. Thus, it is not possible to develop a simple expression for calculating the radiated sound power in terms of individual structural modes [18,19].

1.2 Isogeometric Analysis (IGA)
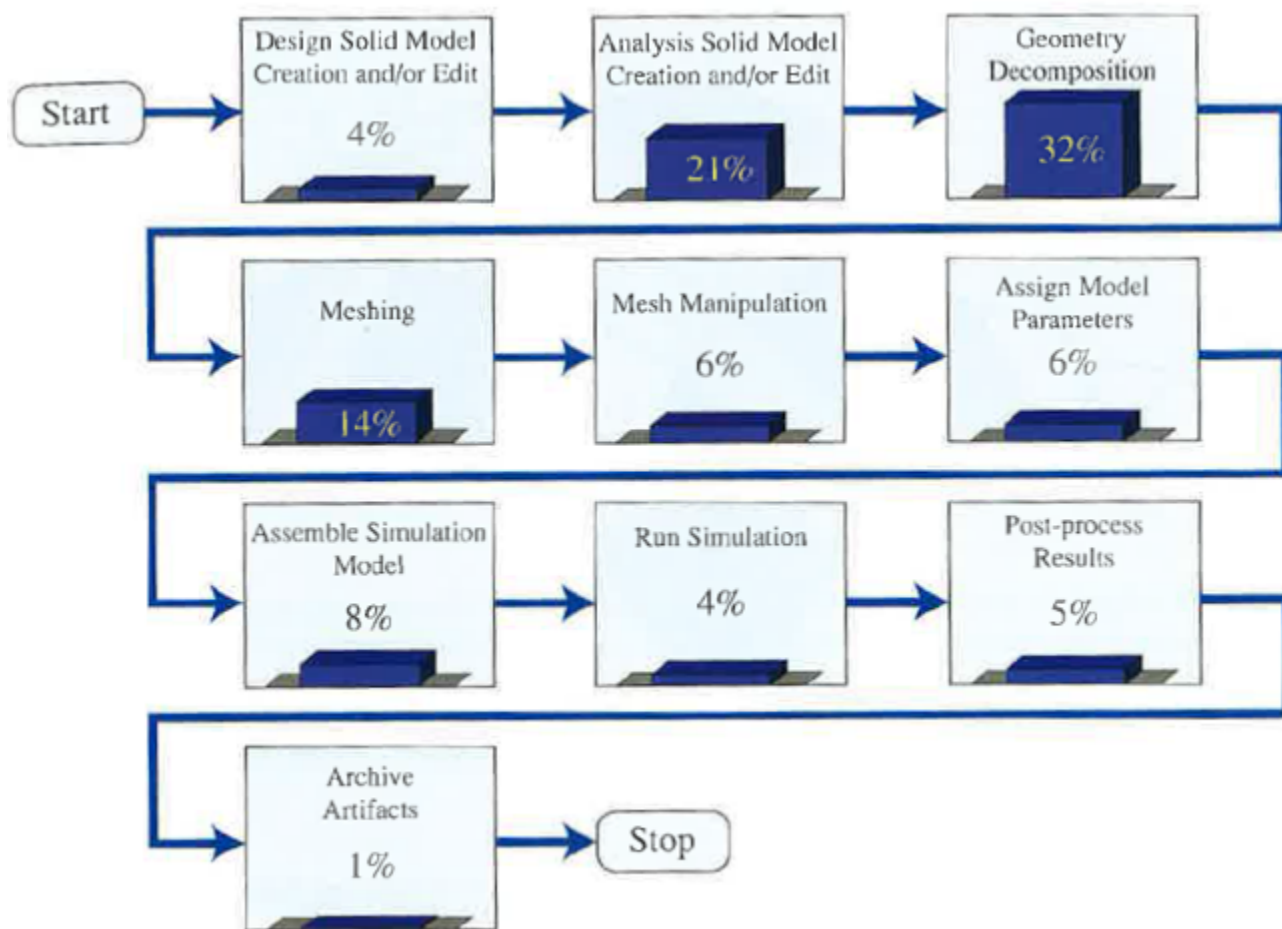
Figure 1.1 Depiction of the overall design and analysis process. See [9]

Figure 1.1 depicts the overall modeling process from design to analysis. As can be seen, the majority of the time is spent translating from the design to an analysis-suitable model. IGA was developed in order to reduce this translation time and cost between design and analysis by using design tools that are already analysis-suitable. IGA offers exact representation of complexities that formerly had to be removed before applying standard finite element analysis (FEA) [9–11]. This lack of approximation in geometry leads to increased accuracy in analysis. In addition to exact geometric representation, IGA requires fewer degrees of freedom, thus offering increased computational efficiency.

IGA achieves these benefits through the choice of basis functions. NURBS (non-uniform rational B-splines) and/or T-splines are used instead of traditional polynomials. Like the polynomials used in FEA, these functions are continuous, but unlike FEA polynomials, they are also smooth. These two characteristics provide the ability to represent complex geometries exactly. These basis functions are used to describe the geometry and then are also used to determine the solution [9–11].

IGA is easily implemented into existing FEA routines, and NURBS are already standard in CAD packages, thus making IGA a very appealing tool. A designer with little to no knowledge of IGA can create a geometry and it is ready for analysis, with only slight modification to current FEA codes [9–11].

Combining IGA with a boundary element method (IGA BEM) offers an additional reduction in degrees of freedom. This is due to the fact that BEM requires only a discretization of the surface rather than the entire volume as required with FEA [8]. IGA BEM has recently been implemented for determining acoustic pressure and compared with traditional BEM [14]. IGA BEM shows significant gains in accuracy and computational efficiency over traditional BEM.

1.3 Contribution

The objective of this project is to apply IGA to the determination of acoustic radiation modes and efficiencies. IGA has been applied to many different fields, and this project will serve to expose the acoustics community to IGA and its benefits.

IGA is used to represent the geometry of a structure, and then the surface velocity is expressed in terms of these basis functions. BEM then uses this surface velocity expression to determine the surface pressure. From this information, the required matrix, denoted as **R** in Equation (1.1), is obtained and diagonalized in order to determine the acoustic radiation modes and efficiencies.

Due to this work, it is possible to compute the acoustic radiation modes and efficiencies from any closed surface designed with T-splines. The IGA BEM tool developed can provide the impetus for research on radiation modes to pick up again, as complex geometries are now more easily represented. This tool may soon have the capability of analyzing structures with vibrating and non-vibrating sections. This feature is not yet complete, but when finished will provide a unique and valuable analysis tool. This work presents results for a sphere and a free cylindrical shell. The sphere results are compared with the analytic solution, and the results for the free cylindrical shell are compared and contrasted with results for a baffled cylindrical shell obtained through an elemental radiator approach.

The rest of this work is laid out as follows. Chapter 2 reviews the theory of acoustic radiation modes. IGA is discussed in Chapter 3, and Chapter 4 details the IGA BEM method used in this work. Results for a sphere and free cylindrical shell are presented in Chapter 5, with concluding remarks and future work in Chapter 6. The pre- and post-processing Python scripts are included in Appendix A.

Chapter 2

Radiation Modes

2.1 Characteristics

Though *structural* modes are powerful in describing structural vibrations, they are not very useful in describing the radiated sound power. Power can be represented, in matrix form, as

$$\Pi = \tilde{\mathbf{v}}^H \mathbf{A} \tilde{\mathbf{v}} \tag{2.1}$$

where $\tilde{\mathbf{v}}$ represents the complex modal (structural) velocities, and $\mathbf{A}$ is the *power transfer matrix*. This matrix contains the coupling information between structural modes, with self-coupling terms on the diagonal and mutual-coupling terms on the off-diagonals. This matrix is dense, and also difficult to evaluate. The structure of $\mathbf{A}$ reveals that the power contribution of each structural mode is dependent on the presence of and interaction with other structural modes [18]. Because of this interdependence, reducing a particular structural mode may actually *increase* the radiated sound power [3,4,16]. As shown in Figure 2.1, reducing the radiated sound power causes some structural modes to be reduced, while others are increased. This lack of intuition associated with structural modes and radiated sound power led to a search for a set of modes that provided a more clear understanding of radiated sound power.

6
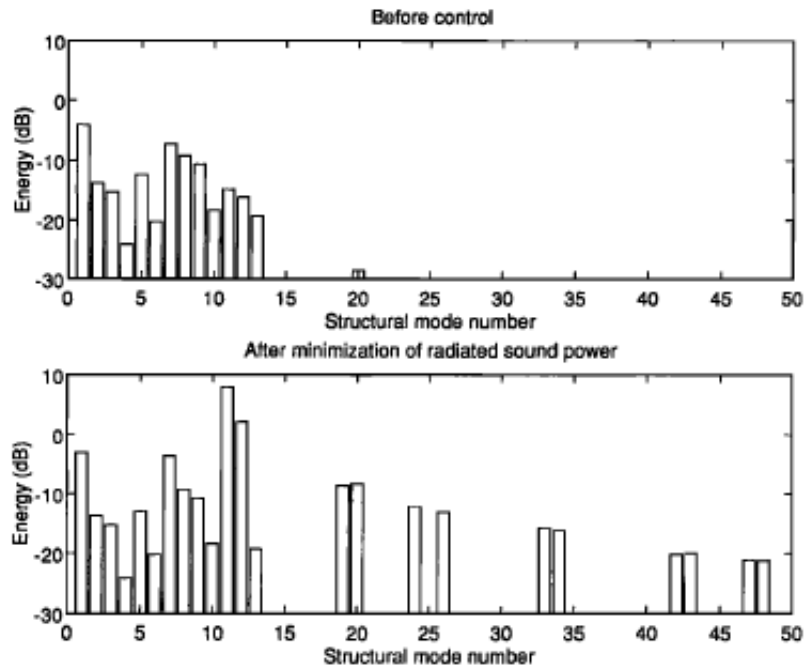
Figure 2.1 Distribution of *structural* modes before and after reducing radiated sound power. Some structural modes actually increase when power is reduced. See [3].

Radiation modes were developed as a more useful way to describe the sound power radiated from a vibrating structure. In contrast to structural modes which are coupled with respect to radiated sound power, these radiation modes contribute independently to the radiated sound power [1–8, 15–17]. This guarantees that a reduction of a particular radiation mode corresponds to a reduction in radiated sound power and vice versa [2, 3, 8]. An example of this is shown in Figure 2.2, where reducing the radiated sound power corresponds to eliminating the first radiation mode. Here we see the intuitive connection between radiation modes and radiated sound power that was non-existent with structural modes.

This characteristic is of great significance for active noise control. Knowledge of the *structural* modes is not necessarily helpful because, as shown in Figure 2.1, reducing a structural mode may actually increase the radiated sound power. But knowing that reducing a *radiation* mode is guaranteed to reduce the radiated sound power, knowledge of the *radiation* modes can guide the position of sensors and control devices to obtain the desired response.
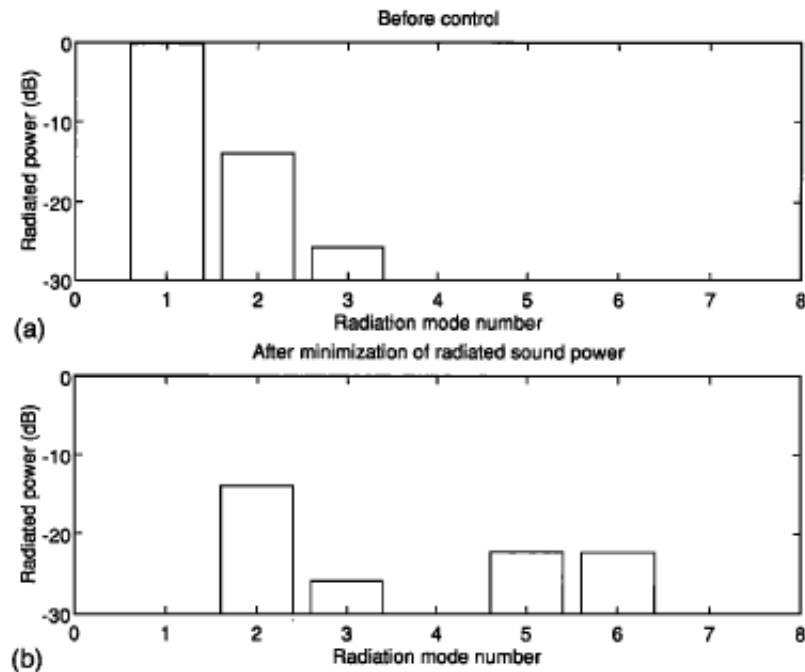
Figure 2.2 Distribution of *radiation* modes before and after reducing radiated sound power. The first mode is completely eliminated, while some higher - and less efficient - modes are increased. See [3].

Another characteristic of radiation modes is that their radiation efficiencies are monotonically decreasing, with the efficiencies of higher modes tapering off to zero, as shown in Figure 2.3. Due
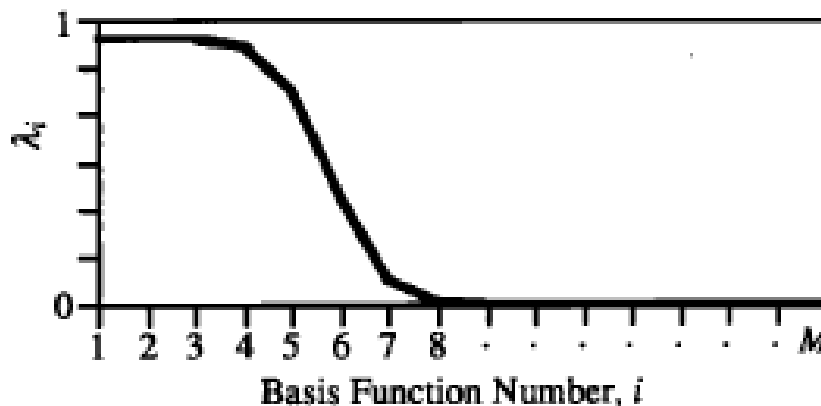
Figure 2.3 Radiation efficiencies of each radiation mode (termed "basis functions"). Only the first few modes are radiate efficiently, contributing to the radiated power. See [2].

to this behavior, it is only necessary to include a relatively small number of radiation modes in the calculation of radiated sound power [1, 2, 4–8]. This is in contrast to a calculation using *structural* modes where a large number of modes may need to be included to obtain an accurate estimate of the radiated sound power.

Radiation modes, and their efficiencies, are dependent on frequency. The lower the frequency, the shorter the plateau in Figure 2.3 becomes. At sufficiently low frequencies, only the first radiation mode has a significant radiation efficiency. Thus the radiated sound power can be calculated using only the first mode, and reducing the power can be achieved by controlling the first radiation mode [3, 4, 8].

This trend of decreasing efficiencies explains Figure 2.2. Even though radiation modes 5 and 6 *increase*, they have a smaller efficiency than the first mode, so the overall radiated sound power is still reduced. This reduction was obtained by controlling only the first radiation mode. Other control techniques could be used to control for multiple radiation modes if desired.

Structural modes are dependent on multiple variables: the geometry and physical properties of the structure, as well as the boundary conditions. Changing any one of these variables requires a recalculation of structural modes. However, *radiation* modes are *only* dependent on the geometry [1, 3–5, 8, 15, 16]. This allows structures with different properties and/or different boundary conditions, but the same geometry, to be analyzed with only one calculation of radiation mode shapes (although the radiation efficiencies will depend on boundary conditions).

These characteristics of radiation modes show their superiority over structural modes. In the next two sections it is shown that not only are they extremely beneficial in describing the radiated sound power, they are also not very difficult to obtain.

2.2   Methods

There are three different approaches to describing a vibrating structure in order to determine the acoustic radiation modes: elementary radiators, velocity expansion, and a boundary element method. These methods can be implemented using a far-field approach or a surface approach. Only the surface approach will be presented.

2.2.1   Elementary Radiators

The most common method to determine the radiation modes of a vibrating structure is to discretize the source into elements, each radiating as a piston-like source, as shown in Figure 2.4 [3,4]. The impedance relationship between each simple source can be determined and, together with knowledge of the velocity of each element, the acoustic radiation modes can be obtained.
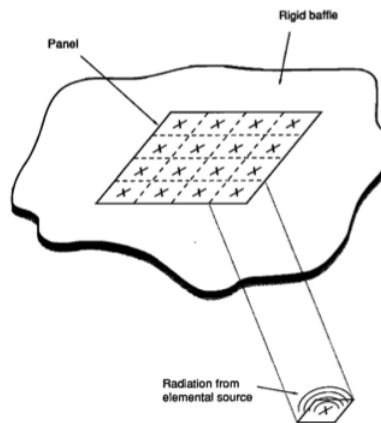
Figure 2.4 Description of the elemental radiator approach. The structure is subdivided into elements and each element radiates as a piston. See [3].

This method is simple, but in order to get accurate results, the element size should be small. This requires more elements to represent the structure, thus increasing the computational cost. Another downside is that it is not known *a priori* how many elements will be needed to obtain a desired level of accuracy.

2.2.2 Velocity Expansion

Another common approach is to represent the surface velocity distribution as a series expansion in terms of a set of basis functions [4, 15, 16, 18, 19]. The structural modes are the most common choice of basis functions, although the choice of basis functions is arbitrary, as long as they form a complete set. Currey and Cunefare [15, 16] have shown that the accuracy and convergence of the radiation efficiencies is dependent on the choice of basis functions and the number of basis functions included in the expansion.

The Rayleigh integral is then used to express the surface pressure in terms of the surface velocity. The coupling between basis functions is represented in a coupling matrix known as the *power transfer matrix* [18, 19]. Though the expression is not difficult to obtain, the power transfer matrix is not an easy calculation.

2.2.3 Boundary Element Method

An alternative approach is the boundary element method (BEM). This method is especially beneficial for arbitrarily shaped three-dimensional sources, where numerical methods are required. BEM is similar to the finite element method (FEM) which, for acoustics problems, is based on a discretization of the Helmholtz equation. However, with BEM, Green's second identity is used to yield the Helmholtz integral equation (HIE):

$$\alpha(\mathbf{r})p(\mathbf{r}) = \int_S p(\mathbf{r}_o)\frac{\partial G(\mathbf{r},\mathbf{r}_o)}{\partial n}\,dS - \int_S G(\mathbf{r},\mathbf{r}_o)\frac{\partial p(\mathbf{r}_o)}{\partial n}\,dS \qquad (2.2)$$

where $\alpha(\mathbf{r})$ is a geometry-related coefficient, $G(\mathbf{r},\mathbf{r}_o)$ represents the free-space Green's function, $\mathbf{r}_o$ is a point on the surface, and $\mathbf{r}$ is a field point.

In obtaining Equation (2.2), volume integrals were reduced to surface integrals. BEM thus only requires a discretization of the surface rather than the whole volumetric domain as is required for FEM, as shown in Figure 2.5. Due to this dimensional reduction, the number of degrees of freedom

is greatly reduced, lowering the computational cost. For exterior problems, BEM also offers the benefits of not needing to approximate infinity, and the Sommerfeld radiation condition at infinity is built in [8, 14].

(a) Exterior problem, FEM approach.   (b) Exterior problem, BEM approach.

(c) Interior problem, FEM approach.   (d) Interior problem, BEM approach.
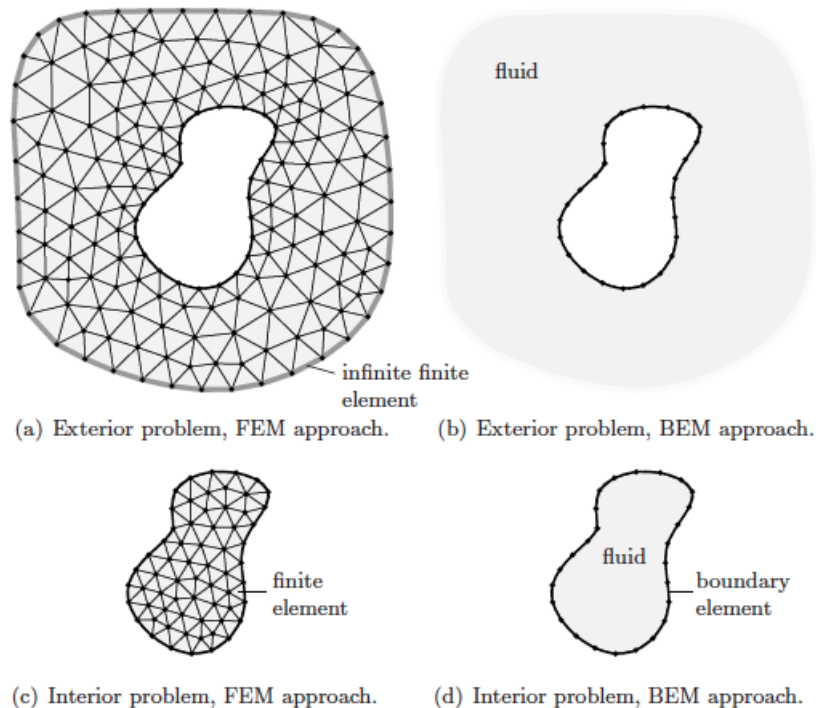
Figure 2.5 Comparison between FEM and BEM for both exterior and interior problems. BEM has fewer degrees of freedom and also no need to approximate infinity in the exterior problem. See [8].

As shown by Equation (2.2), information at the surface of the structure is used to determine information at a field point. However, by choosing the field point to be located on the surface, surface pressure can be obtained from surface velocity, and vice versa, and thus the radiation modes can be obtained as in other methods.

In a recent work, [8], BEM has been used to determine acoustic radiation modes. The surface velocity, and also the surface pressure, are represented as series expansions of basis functions as discussed in Section 2.2.2. The basis functions chosen are the standard shape functions for traditional FEM and BEM.

2.3   Calculation

Regardless of the method used to describe the situation and obtain the necessary quantities, the calculation of radiation modes employs an eigenvalue decomposition or use of the SVD of a radiation operator [1–4, 8, 15, 17]. Due to the existence of other works that present the radiation mode development [1–4, 6, 15–17], only Visser's BEM development [8] will be summarized here, as it serves as the basis for the work that will be presented in Chapter 4.

Using the surface approach, acoustic power is calculated by

$$\Pi = \frac{1}{2} \int\limits_{S} Re[v_n(\mathbf{x})^* p(\mathbf{x})] d\mathbf{x} \tag{2.3}$$

where the integration is over the surface of the structure and $\mathbf{x}$ represents a point on the surface. Writing the normal velocity and pressure as series expansions of basis functions, Equation (2.3) becomes

$$\Pi = \frac{1}{2} \sum_m \int\limits_{S_m} Re[\sum_i (v_n^H)_i N_i^H(\eta, \xi) \sum_j N_j(\eta, \xi) p_j] dS \tag{2.4}$$

where i, j iterate over the number of nodes per element and m runs over the total number of elements.

Noting that the velocity and pressure terms in Equation (2.4) are just coefficients of the basis functions, they can be removed from the integral. The pressure can be written in terms of the velocity using the impedance relationship $p = Zv$. Combining the $\frac{1}{2}$ and the integral over the basis functions into $\mathbf{B}$, Equation (2.4) can be written in matrix form as

$$\Pi = Re[\mathbf{v}_n^H \mathbf{B} \mathbf{Z} \mathbf{v}_n]$$

Due to the fact that $\mathbf{B} \cdot \mathbf{Z}$ is a symmetric matrix [1, 3, 4, 6, 8, 15, 16], this can be rewritten as

$$\Pi = \mathbf{v}_n^H \mathbf{C} \mathbf{v}_n \tag{2.5}$$

where $\mathbf{C} \equiv Re[\mathbf{B} \mathbf{Z}]$, and is called the *radiation resistance operator*.

The majority of the existing work diagonalizes **C** via an eigenvalue decomposition to obtain an expression for the power in terms of radiation modes and their efficiencies, without an explicit connection of the efficiencies to the eigenvalues [2–4, 15–19]. Below is shown the relationship between efficiencies and eigenvalues as presented in [8], which only differs from that presented in [1, 5, 6] due to differences in initial definitions.

The radiation efficiency of an *arbitrary* velocity distribution is defined to be the ratio of the actual power radiated to a reference power. The reference power, Π_o, is chosen to be the power radiated by the same velocity distribution subject to a $\rho_o c$ loading [8]. This can be obtained by substituting $p_j = (\rho_o c) v_j$ into Equation (2.4). The radiation efficiencies can then be expressed as

$$\sigma = \frac{\Pi}{\Pi_o} = \frac{1}{\rho_o c} \frac{\mathbf{v}_n^H \mathbf{C} \mathbf{v}_n}{\mathbf{v}_n^H \mathbf{B} \mathbf{v}_n} \tag{2.6}$$

Rearranging Equation (2.6) yields a generalized eigenvalue problem:

$$\mathbf{C} \mathbf{v}_n = (\rho_o c \sigma) \mathbf{B} \mathbf{v}_n \tag{2.7}$$

It is readily apparent that the radiation efficiencies are related to the eigenvalues of **C** as

$$\sigma_i = \frac{\lambda_i}{\rho_o c} \tag{2.8}$$

To obtain the acoustic radiation modes, the radiation resistance operator, **C**, is diagonalized to rewrite Equation (2.5) as

$$\Pi = \sum_{n=1}^{M} \lambda_i |a_i|^2 \tag{2.9}$$

where **a** represents the projection of the surface velocity distribution onto the radiation modes and *M* is the number of degrees of freedom used. Visser goes through the development of Equations (2.8) and (2.9) in Appendix C of his work [8], with a few minor mistakes. These mistakes are remedied, and a full development of Equation (2.9) is given, in Section 4.2.

Chapter 3

Isogeometric Analysis (IGA)

With models being designed with increasing complexity, making the design suitable for analysis has become increasingly difficult. Unfortunately the design and analysis industries have developed independently of each other, so there is an intermediate translation step required before the design can be analyzed. According to Hughes [11], this intermediate step constitutes about 80% of the total design and analysis time.

3.1   What is IGA

IGA is an analysis framework that has been developed in order to unify the design and analysis processes [9–11, 14, 20]. By modifying the design phase so as to produce analysis-suitable geometries, IGA eliminates the intermediate translation, and thus greatly reduces the analysis time while also offering additional benefits described in Section 3.1.2.

3.1.1   Characteristics

The implementation of IGA differs from standard FEA in two key aspects. The first is the use of the isoparametric concept, in which the solution and geometry are represented by the same set of basis functions. FEA invokes the isoparametric concept by choosing a set of basis functions to represent the solution and using those functions to describe the geometry. This choice however, leads to complex geometries needing to be approximated, as shown in Figure 3.1, in order to avoid nonphysical solutions [9, 10] in the proximity of complexities.
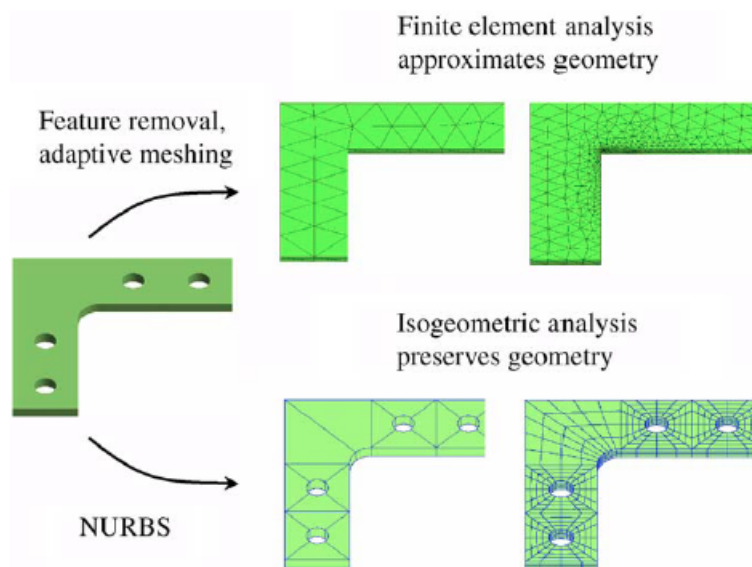
Figure 3.1 Differences in FEA and IGA Modeling. FEA requires geometry approximations, where IGA allows for complexities. See [9, 10].

IGA also uses the isoparametric concept, but in reverse: the same basis functions chosen to represent the geometry are then used to represent the solution [9–14, 20, 21]. This allows for complexities in the geometric design to remain in place for the analysis, as shown in Figure 3.1.

The other difference in implementation between IGA and FEA is the choice of basis functions. How the isoparametric concept is invoked influences this choice. FEA uses piecewise polynomial functions that have compact support (non-zero value over a small range). These functions are

continuous, but are not smooth.

IGA often uses Non-Uniform Rational B-Splines (NURBS) as the basis functions. T-Splines, a superset of NURBS, are another choice that offers greater flexibility than NURBS in the design process, including the capabilities of hanging nodes (T-junctions) and local (as opposed to global) refinement. These choices of basis functions are identical to the constant and linear functions used in FEA, but starting with the quadratic functions, NURBS and T-Splines offer homogeneity between nodes, whereas FEA quadratic functions are different for internal and end nodes [11]. NURBS and T-Splines also have compact support, are continuous, and are also smooth [9–12].

3.1.2 Benefits

Standard computer-aided design (CAD) packages already employ the use of NURBS [9–11]. There is no requisite knowledge about IGA in order for designers to create designs that are analysis-suitable. A T-Spline plug-in has been developed for the Rhino design software to allow for greater flexibility in design and analysis by using T-Splines rather than NURBS. This plug-in allows for a one-click conversion from NURBS to T-Splines.

Due to how IGA implements the isoparametric concept, there is no translation that needs to take place between the design and analysis phases. In addition, complex features in the design do not need to be removed, so there are no approximations of this sort that need to be made, as shown in Figure 3.1. This allows for more accurate analysis to be performed.

Another benefit of NURBS (and T-splines) is the capability of representing geometries exactly [9–11, 14, 20, 21]. The polynomial functions used in FEA are only capable of approximating geometries such as a circle in the limit that the element size approaches zero, as shown in Figure 3.2. This requires a large number of elements, whereas IGA obtains an exact representation with relatively few elements.

IGA achieves this exact representation by use of projective transformations of piecewise B-
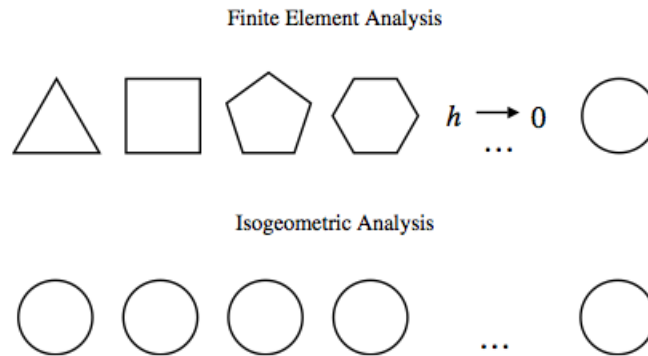
Figure 3.2 FEA only approximates a circle in the limit as the element size approaches zero. IGA represents a circle exactly, regardless of the element size. See [10].

Spline curves in $\mathbb{R}^{d+1}$ space to generate NURBS curves in $\mathbb{R}^d$. This is done by constructing control points in $\mathbb{R}^d$ that are related to the projective control points in $\mathbb{R}^{d+1}$, as shown in Figure 3.3 [9, 11].
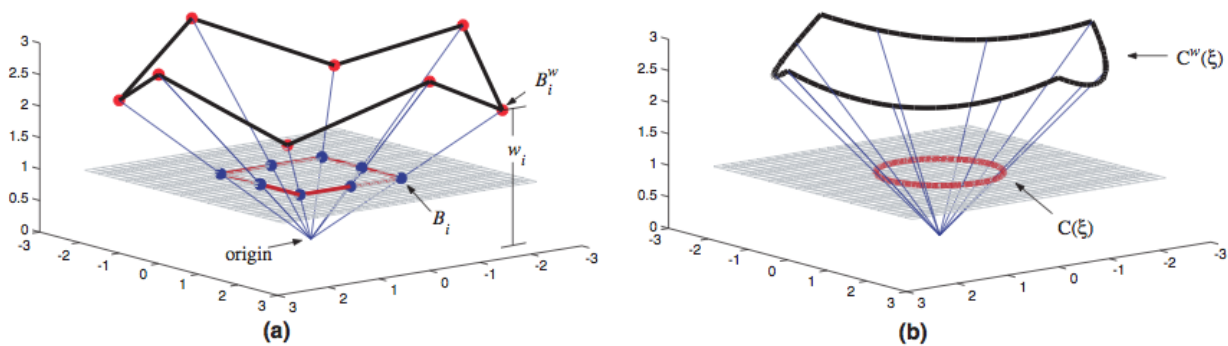
Figure 3.3 Circle in $\mathbb{R}^2$ constructed by projective transformation of piecewise quadratic B-Spline in $\mathbb{R}^3$. (a) Projective transformation of "projective control points" yields control points. Weight, w_i is the z-component of B_i^w. (b) Projective transformation of B-Spline curve $C^w(\xi)$ yields NURBS curve $C(\xi)$. See [11].

Figure 3.4 shows a coarse IGA mesh of a sphere with 8 elements, as well as two meshes obtained through mesh refinement, containing 32 and 128 elements, respectively. With IGA, even the coarsest mesh captures the geometry exactly, in contrast to FEA/BEM, as shown in the meshes Figure 3.5 with 32, 128, and 512 elements, respectively. With FEA/BEM, mesh refinement is employed to obtain a better approximation of the geometry, whereas with IGA, mesh refinement is

(a) Mesh 1   (b) Mesh 2   (c) Mesh 3
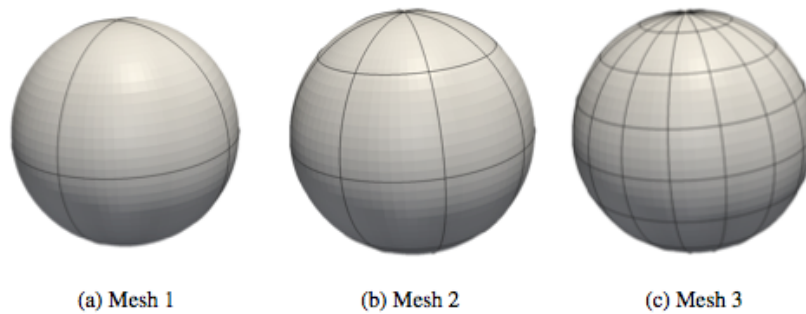
Figure 3.4 Three successive exact IGA meshes with 8, 32, and 128 elements respectively. Each mesh captures the geometry exactly. See [14].

(a) Mesh 1   (b) Mesh 2   (c) Mesh 3
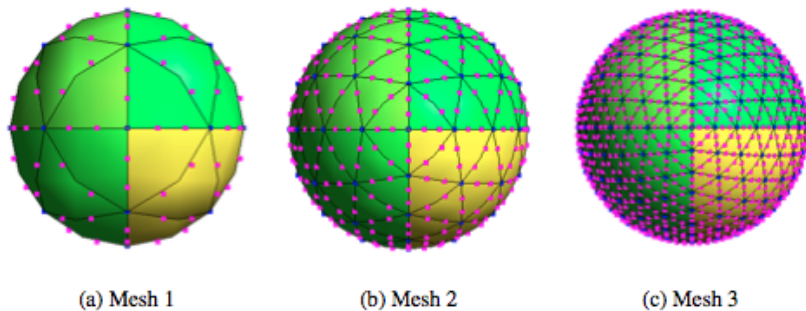
Figure 3.5 Three successive BEM meshes with 32, and 128, and 512 elements respectively. Each mesh is an approximation of the geometry. See [14].

used in order to obtain higher resolution in the solution. IGA offers mesh refinements analogous to the *h*- and *p*-refinements of FEA, as well as an unparalleled *k*-refinement which increases smoothness in the basis functions, which is useful in analysis. This unique *k*-refinement also limits the growth in degrees of freedom [9–11], as displayed in Table 3.1.

IGA mesh refinement has the benefit that it does not change the geometry or parameterization [9–11, 14, 20, 21]. This is a very desirable feature, eliminating the need for communication with the CAD system if mesh refinement is done to increase the accuracy of the solution.

Another feature of IGA is that it can easily be implemented into existing FEA/BEM codes with minor adjustments [11–14]. By using Bézier extraction, a technique detailed in [12, 13],

	IGA (cubic)			BEM (cubic)		
refinement level	**1**	**2**	**3**	**1**	**2**	**3**
# elements	8	32	128	32	128	512
# dofs	62	114	266	114	450	1794

Table 3.1 Computational cost for three meshes for IGA and BEM. The degrees of freedom increases substantially for BEM while IGA increases at a more manageable rate. See [14].

NURBS and T-splines can be defined over Bézier elements that can then be implemented directly in standard FEA/BEM codes. In essence, all that needs to be changed is the shape function routine, which defines the basis functions [9–11].

3.2   Isogeometric Boundary Element Method (IGA BEM)

In a recent paper, Simpson implemented IGA BEM, and applied it to acoustic problems [14]. Simpson demonstrated the superior accuracy of IGA BEM over traditional BEM using Lagrange elements, as shown in Figure 3.6. Both methods used cubic elements.

(a) Traditional BEM, 450 dofs   **(b)** IGA BEM, 62 dofs
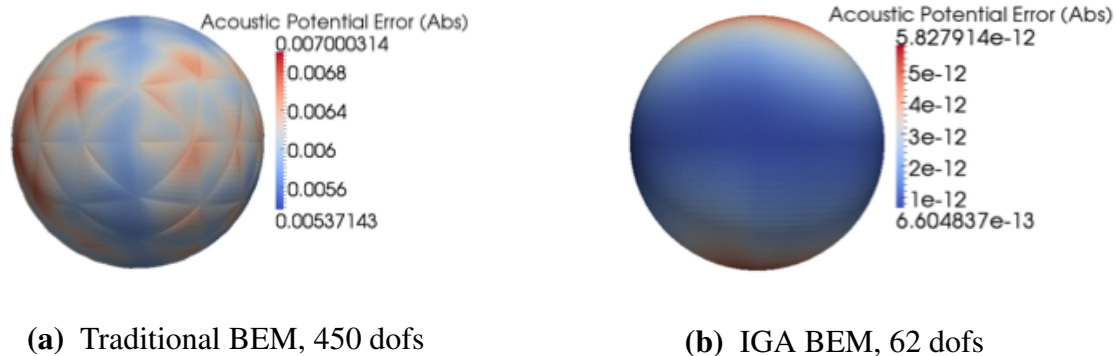
Figure 3.6 Comparing traditional BEM using Lagrange elements with IGA BEM. Both methods use cubic elements (128 and 8 respectively) and 20x20 quadrature points. See [14].

Remark: After comparing the development in [14] with [8], it was determined by Simpson that Figure 3.6 should represent the error in acoustic *pressure*, not the acoustic potential.

As seen in Figure 3.6, IGA BEM is roughly 9 orders of magnitude more accurate than traditional BEM. Simpson showed that this error in traditional BEM is dominated by geometric error, whereas the error in IGA BEM approaches machine precision due to the lack of geometric error [14].

Computational cost is also greatly reduced with IGA BEM. The results in Figure 3.6a were obtained with 128 elements and 450 degrees of freedom. In contrast, the results in Figure 3.6b were obtained with only 8 elements and 62 degrees of freedom [14].

The significant gains in accuracy with decreased computational cost validate the use of IGA BEM in acoustic problems. This IGA BEM implementation forms the basis of the work presented in Chapters 4 and 5.

Chapter 4

IGA BEM Calculation

4.1 Modifications to Existing IGA BEM Code

Simpson's [14] IGA BEM implementation (C++) uses Equation (2.2) to solve for surface variables by specifying a field point on the surface of the structure. By specifying the normal derivative of the surface pressure (at each node), which is proportional to the surface velocity through Euler's relation, the surface pressure (at each node) is obtained. In like manner, the surface pressure could instead be specified to obtain the surface velocity.

4.1.1 Non-Uniform Boundary Condition

Simpson's current implementation only allows for a uniform pressure (or velocity) to be specified. In the .ext input file, there is a node set specifying how many and which nodes are in the node set. The value for the nodes in this set can then be specified in the .xml input file. In the original .ext file, shown (in part) as "Original inverted_sphere_base.ext" below, the node set contains all 62 nodes, corresponding to a uniform surface pressure (or velocity). The input files were modified in this work to allow for non-uniform velocities to be specified. New node sets were defined to allow

for multiple node sets of different sizes. The two modified .ext files are shown (in part) below as "makeBZ inverted_sphere_base.ext" and "rad_modes inverted_sphere_base.ext".

```
───────────────────────────── Original inverted_sphere_base.ext ─────────────────────────────
...
set 62 node all_nodes 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
56 57 58 59 60 61
```

```
───────────────────────────── makeBZ inverted_sphere_base.ext ─────────────────────────────
...
set 1 node active_node 3
set 61 node zero_nodes 0 1 2 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
56 57 58 59 60 61
```

```
───────────────────────────── rad_modes inverted_sphere_base.ext ─────────────────────────────
...
set 1 node node0 0
set 1 node node1 1
set 1 node node2 2
...
set 0 nodes zero_nodes
```

In the "makeBZ" file, node 3 is allowed to have a specified value, while the other 61 nodes are set to zero. A new .ext file was created for each node in turn to be in the active_node node set. In the "rad_modes" file, each node is allowed to have its own specified value, and in this example, there are no nodes set to zero. See Appendix A for the Python scripts that create the modified files.

4.1.2   Redirect Output

Simpson's IGA BEM implementation returns vectors of the surface pressure and its normal derivative directly to the terminal. This output was redirected to a text file for ease of use in the post-processing in Python.

4.1.3   Make B and Z Matrices

In order to determine radiation modes, the impedance matrix, $\mathbf{Z}$, is needed. This can be calculated using knowledge of surface pressure and surface velocity by

$$\mathbf{p} = \mathbf{Zv} \tag{4.1}$$

In Equation (4.1), if $\mathbf{v}$ is the identity matrix, then $\mathbf{p} = \mathbf{Z}$. Using this fact, Simpson's code was run N times, where N is the total number of nodes, each time specifying the velocity as a column of the identity matrix:

$$\mathbf{v}_i = \mathbb{1}(:, i)$$

This is done by looping over the nodes, each time putting one node in the active_node node set and the rest in the zero_nodes node set, as described in the "makeBZ inverted_sphere_base.ext" file in Section 4.1.1. The resulting surface pressure from each specified velocity is stored as a column of $\mathbf{p}$. Once $\mathbf{p}$ is constructed, $\mathbf{p} = \mathbf{Z}$, and $\mathbf{Z}$ is ready for analysis.

In terms of the IGA framework, each node corresponds to one basis function. So each run of the code corresponds to setting one basis function equal to unity and the rest to zero.

In order to use Visser's development as discussed in Sections 2.3 and 4.2, the $\mathbf{B}$ matrix must be obtained, which contains the integration over the basis functions. Simpson's code uses this matrix but it is not stored, and it doesn't include the factor of $\frac{1}{2}$ that Visser includes.

Creating and storing these matrices provides all the necessary information for determining the radiation modes using Visser's development [8].

4.1.4   Selection Sets

The modifications in Sections 4.1.1 - 4.1.3 are necessary for the radiation mode analysis for scenarios in which the whole structure is allowed to vibrate. However, in some situations, it may be interesting to analyze a scenario in which only part of the structure is allowed to vibrate.

In Rhino, the design software for which a T-spline plug-in has been developed, a selection of elements can be grouped into what is called a selection set. Selecting a selection set allows all the elements in that set to be selected with the click of a button. This is useful for the design process so that changes can be made quickly to multiple elements, rather than one at a time.

Selection sets are also useful for analyzing scenarios in which only part of a structure is allowed to vibrate. Creating a selection set adds additional information to the output file (which becomes the input .ext file for the analysis), as shown in the "selset inverted_sphere_base.ext" excerpt below. This information states that there are 4 elements - elements 4, 5, 6, and 7, in the selection set.

```
                          selset inverted_sphere_base.ext
...

set 4 elem SelSet1 4 5 6 7
```

The information of which nodes correspond to which elements is also contained in the .ext file, as shown below in the "node set inverted_sphere_base.ext" excerpt. This shows that there are 13 nodes associated with this element, and lists which nodes those are. Logic was added to the pre-processing to determine the set of valid nodes - nodes that are in the selection set - for which to run the analysis on. The nodes not in the valid node set are constrained to zero in the "rad_modes inverted_sphere_base.ext" file. See Appendix A for the complete Python scripts.

```
                          node set inverted_sphere_base.ext
...

elem 13 3 3

0 1 2 3 4 13 14 15 16 25 26 27 28

...
```

Unfortunately, this tool is not complete yet. The nodes on the boundary between the region of interest of the structure and the non-interesting region are shared by elements in both regions. This sharing causes some spillover effects into the non-interesting region, so it is not completely held to zero.

4.2   Corrections to Existing Radiation Mode Development

This section will flesh out and correct Visser's [8] development of Equations (2.8) and (2.9). The results he presents are accurate, with only minor mistakes in the details. Some notation below is new, but is only used to clarify Visser's development.

Rewriting Equation (2.7) in terms of γ_i, the desired radiation modes, produces

$$\mathbf{C}\gamma_i = (\rho_o c \sigma_i)\mathbf{B}\gamma_i \tag{4.2}$$

Noting that $\mathbf{B}$ is real, symmetric, and positive definite, this generalized eigenvalue problem can be transformed into a standard eigenvalue problem using a Cholesky decomposition of $\mathbf{B}$:

$$\mathbf{B} = \mathbf{U}^T\mathbf{U}$$

Recalling that $\mathbf{C} \equiv Re[\mathbf{BZ}]$, and defining $\gamma_i \equiv \mathbf{U}^{-1}\psi_i$, Equation (4.2) can be rewritten as

$$Re[\mathbf{U}^T\mathbf{U}\mathbf{Z}]\mathbf{U}^{-1}\psi_i = \rho_o c \sigma_i \mathbf{U}^T\mathbf{U}\mathbf{U}^{-1}\psi_i$$

Left-multiplying this by $\mathbf{U}^{-T}$ and defining $\mathbf{Q} \equiv Re[\mathbf{U}\mathbf{Z}\mathbf{U}^{-1}]$ produces the desired standard eigenvalue problem:

$$\mathbf{Q}\psi_i = \rho_o c \sigma_i \psi_i \tag{4.3}$$

Equation (4.3) shows that ψ_i is an eigenvector of $\mathbf{Q}$ with $\lambda_i = \rho_o c \sigma_i$ which is Equation (2.8), i.e. changing the generalized eigenvalue problem to a standard eigenvalue problem does not change

the relationship between the eigenvalues and the efficiencies. The matrix of radiation modes, Γ, is then defined in terms of the eigenvectors of $\mathbf{Q}$:

$$\Gamma \equiv \mathbf{U}^{-1}\Psi \tag{4.4}$$

To obtain the radiated power in terms of the radiation modes, we return to Equation (2.5), reproduced here for convenience:

$$\Pi = \mathbf{v}_n^H \mathbf{C} \mathbf{v}_n \tag{2.5}$$

$\mathbf{C}$ can be written in terms of $\mathbf{Q}$ as $\mathbf{C} = \mathbf{U}^T \mathbf{Q} \mathbf{U}$. Decomposing $\mathbf{Q}$ into eigenvalues and orthogonal eigenvectors, utilizing Equation (4.4), and noting that the radiation modes, γ_i, are orthogonal leads to

$$\Pi = \mathbf{v}_n^H \mathbf{U}^T \Psi \Lambda \Psi^T \mathbf{U} \mathbf{v}_n$$

$$\Pi = \mathbf{v}_n^H \Gamma^{-T} \Lambda \Gamma^{-1} \mathbf{v}_n$$

$$\Pi = \mathbf{v}_n^H \Gamma \Lambda \Gamma^T \mathbf{v}_n$$

Defining $\mathbf{a} \equiv \Gamma^T \mathbf{v}_n$ (the projection of $\mathbf{v}_n$ onto the radiation modes), the result of Equation (2.9) is obtained, and repeated here:

$$\Pi = \sum_{n=1}^{M} \lambda_i |a_i|^2 \tag{2.9}$$

4.3   Visualizing Radiation Mode Shapes

The acoustic radiation modes are related to the eigenvectors of $\mathbf{Q}$ as expressed in Equation (4.4). It is important to remember that in this development, $\Gamma, \Psi, \mathbf{v}_n$, and $\mathbf{a}$ only represent the coefficients of the basis functions. To obtain the full expressions, these coefficients must be projected onto the basis functions.

In order to visualize the acoustic radiation mode shapes, the eigenvectors, Γ, are fed back into Simpson's code one by one as the specified surface velocity. This projects the coefficients onto the

corresponding basis functions and provides a .vtu file to visualize the velocity distribution over the surface of the geometry.

In the case where there is a selection set that doesn't represent the entire structure, care must be taken so as to assign the right coefficients to the right basis functions. Each eigenvector, γ_i, will be the same length as the valid node set, and the first entry corresponds to the first valid node, which is not necessarily node 0. Logic was added to the pre-processing to assign the coefficients in γ_i to the correct nodes and set the rest of the nodes to zero. See Appendix A for the complete Python script.

Modifications were made to Simpson's code in the case of the sphere, where analytic solutions were available, in order to provide comparison between the IGA BEM calculation and the analytic solution.

In addition, the radiation modes were normalized so as to have a maximum value of 1. This normalization matches the normalization of the spherical harmonics that was used in generating the analytic solution for the sphere, allowing for comparisons to be made.

Chapter 5

Examples

Though one of the advantages of using IGA BEM is the ability to handle arbitrarily complex geometries, the results for a sphere and a cylindrical shell are presented. The sphere was chosen to validate the method due to the availability of an analytic solution we could compare with. The cylindrical shell was chosen to demonstrate capability.

5.1 Sphere

The results below are for $a = 0.5$, $k = 3$, $\omega = 1$, and $\rho_o = 1.21$. The parameters are arbitrary, and were chosen to be able to compare with Visser's results [8].

The sphere was analyzed using three methods: IGA BEM, and traditional BEM using linear triangular elements and traditional BEM using cubic triangular elements. IGA BEM uses only 8 cubic elements to represent the sphere exactly, and the analysis is done with 62 degrees of freedom. The traditional BEM uses 128 triangular elements to approximate the sphere, and the analysis is done with 66 and 450 degrees of freedom, for the linear and cubic meshes respectively. This data is summarized in Table 5.1. It is readily seen that IGA BEM offers great benefits in computational cost.

Computation Parameters			
parameter	**IGA BEM** cubic	**BEM** linear	**BEM** cubic
# elements	8	128	128
# dofs	62	66	450

Table 5.1 Comparison of computation parameters for the analysis of a sphere for IGA BEM, traditional BEM with a linear elements, and traditional BEM with cubic elements.

5.1.1   Radiation Mode Shapes

The acoustic radiation modes are related to the spherical harmonics [1, 5, 8, 17]. More specifically, the magnitude of the radiation mode shapes matches the magnitude of the real part of the spherical harmonics. Figures 5.1-5.3 show the spherical harmonics compared with the results for the three methods for the first 9 radiation modes. As noted in [1, 5, 17], the radiation modes match the spherical harmonics, to within a phase difference. The figures presented here have been rotated so the orientation matches among all the methods.

Figure 5.1 Magnitude of $Re\{Y_0^0\}$ compared with the magnitude of radiation mode 1 for IGA BEM, traditional BEM with linear elements, and traditional BEM with cubic elements. See [14].

As can be seen in Figures 5.1-5.3, IGA BEM is superior to traditional BEM in capturing the geometry exactly, whereas traditional BEM is only able to approximate the geometry. IGA BEM obtains this exact representation with only 62 degrees of freedom, as opposed to the traditional BEM with cubic elements which uses 450 degrees of freedom, and still is only an approximation.
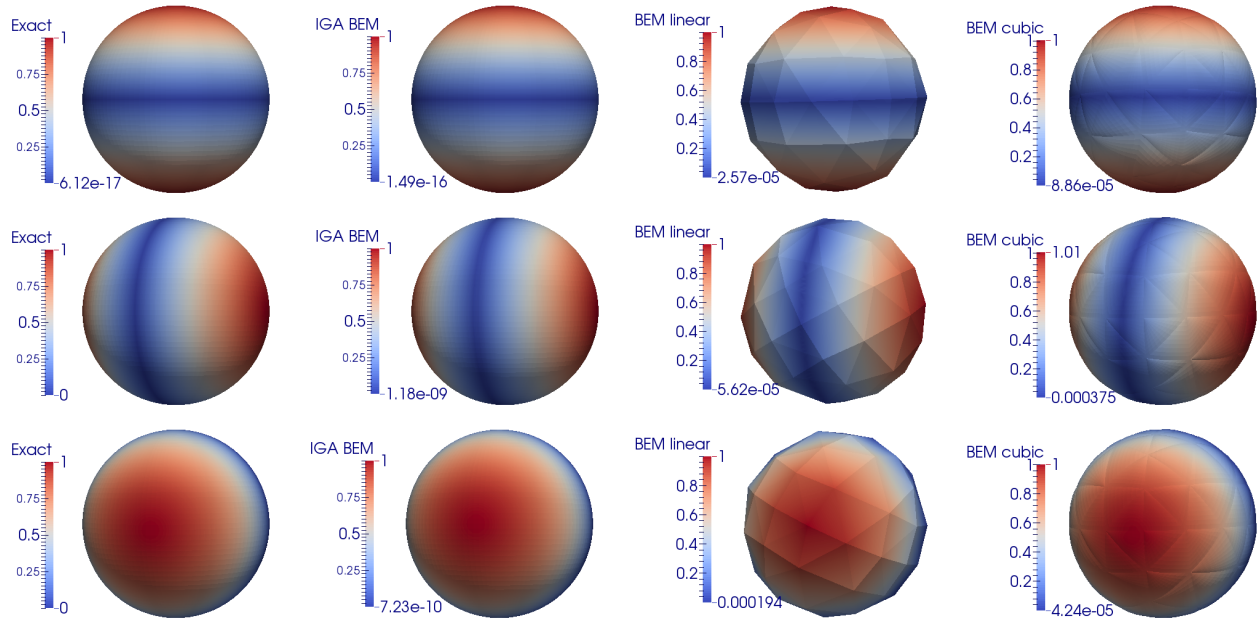
Figure 5.2 Magnitude of $Re\{Y_1^m\}$ compared with the magnitude of radiation modes 2-4 for IGA BEM, traditional BEM with linear elements, and traditional BEM with cubic elements.

As for the radiation mode shapes, all three methods capture the general features. Figures 5.1-5.3 agree with the results presented in [5, 8]. Due to the normalization that was done, all three methods have a maximum value of 1. However, IGA BEM does better than traditional BEM in capturing the minimum extrema, nearing machine precision when the exact solution is zero. Traditional BEM suffers from geometry errors which affect the accuracy of the solution. As for capturing the fine features of the radiation mode shapes, IGA BEM also excels over traditional BEM, as most readily seen in modes 5 and 9 shown Figure 5.3.

Traditional BEM is sufficiently accurate if the interest is only in the general radiation mode shapes. However, the more complex the mode shape becomes, the less accurate traditional BEM becomes, and more complex geometries are likely to have more complex radiation modes. The other downside to traditional BEM is that it is very costly computationally. Using linear elements took about three days, while using cubic elements took six days.
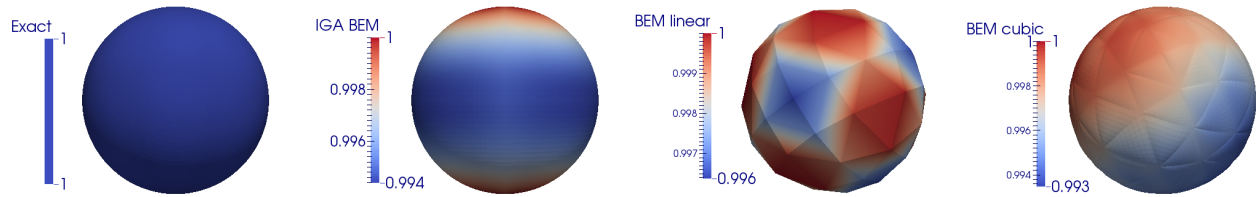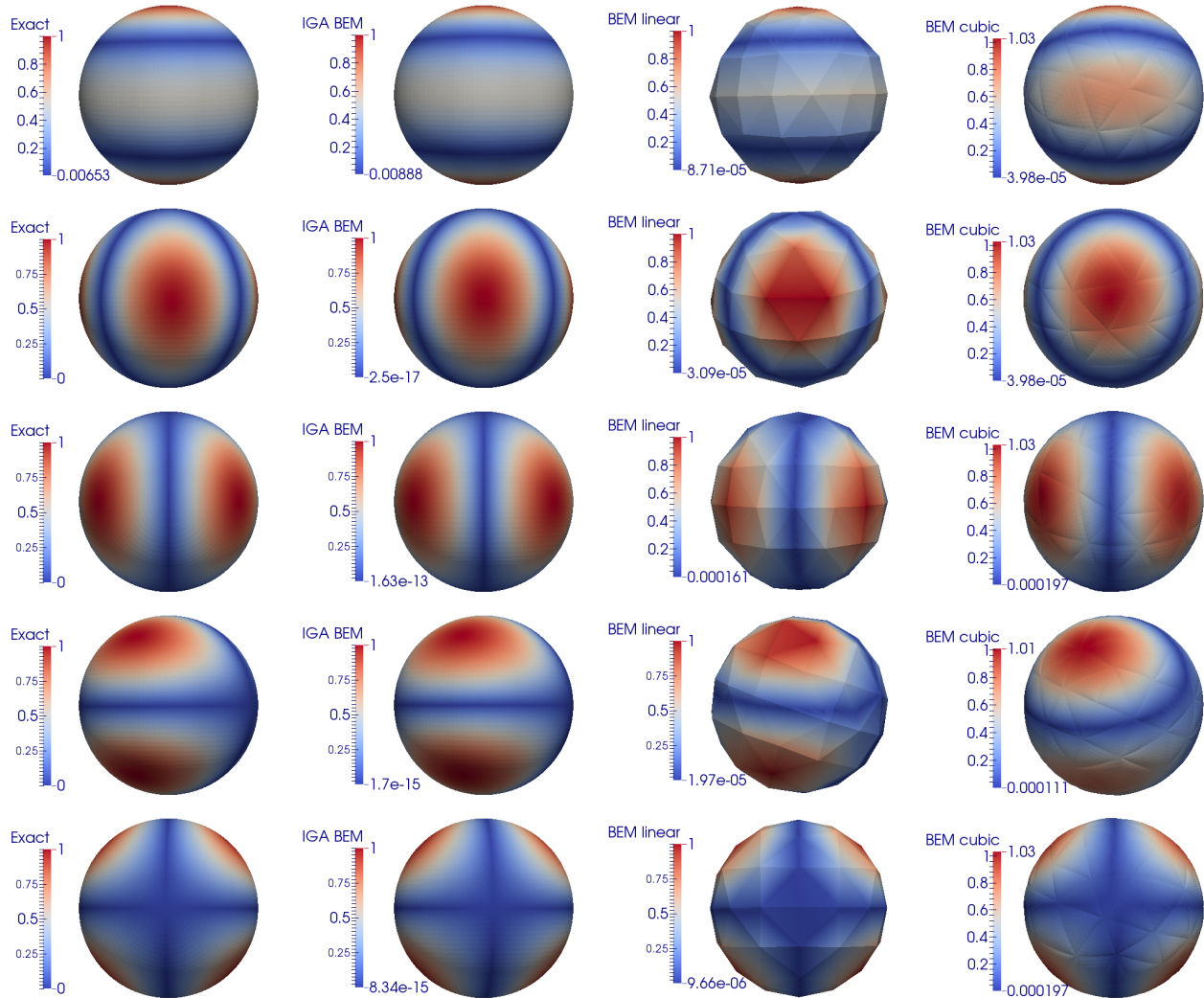
Figure 5.3 Magnitude of $Re\{Y_2^m\}$ compared with the magnitude of radiation modes 5-9 for IGA BEM, traditional BEM with a linear elements, and traditional BEM with cubic elements.

IGA BEM on the other hand provides accurate results for at least the radiation modes of interest. As discussed in Section 2.1, only the first few radiation modes radiate efficiently, and this will be shown for the sphere in Section 5.1.2. IGA BEM is also efficient in that it only takes about 30 minutes to run for the sphere. This increased accuracy and efficiency validates and encourages the use of IGA BEM for obtaining acoustic radiation modes.

5.1.2 Efficiencies

The exact radiation mode efficiencies for the sphere can be calculated via acoustic potential. Acoustic potential, Φ, is defined, and related to the velocity and pressure, as:

$$\Phi(r) = c_1 h_n^{(2)}(kr)e^{-\iota m \psi} P_n^m(cos\theta)$$

$$v = \frac{\partial \Phi(r)}{\partial r}\Big|_{r=a}$$

$$p = -\iota \omega \rho_o \Phi(a)$$

where $h_n^{(2)}$ is a spherical Hankel function of the second kind and c_1 is a scaling factor, and the $e^{\iota \omega t}$ time dependence is assumed [22]. Knowing that the velocity distribution will be the spherical harmonics for the radiation modes, c_1 can be determined. These expressions for v and p are then substituted into Equation (2.3) to obtain the radiated power, where the integration is performed in spherical coordinates. The reference power is again the power due to the specified velocity pattern under a $\rho_o c$ loading.

The radiation mode efficiencies are numerically determined using Equation (2.8). The efficiencies for the first 20 (10) radiation modes using IGA BEM and the two element types for traditional BEM are compared with the exact solution in Figure 5.4 and Table 5.2.

Figure 5.4 shows good agreement in the distribution of the efficiencies and matches that shown in [8]. The efficiency grouping, first discussed by [5], is readily seen with the first 9 modes. It is clear that the efficiencies are degenerate with respect to the integer m in the spherical harmonics, Y_n^m.

Traditional BEM with linear elements is able to capture the distribution of the efficiencies, but only the efficiency of the first mode is accurate. Traditional BEM with cubic elements is more accurate, but IGA BEM is the most accurate, as shown more clearly in Table 5.3.
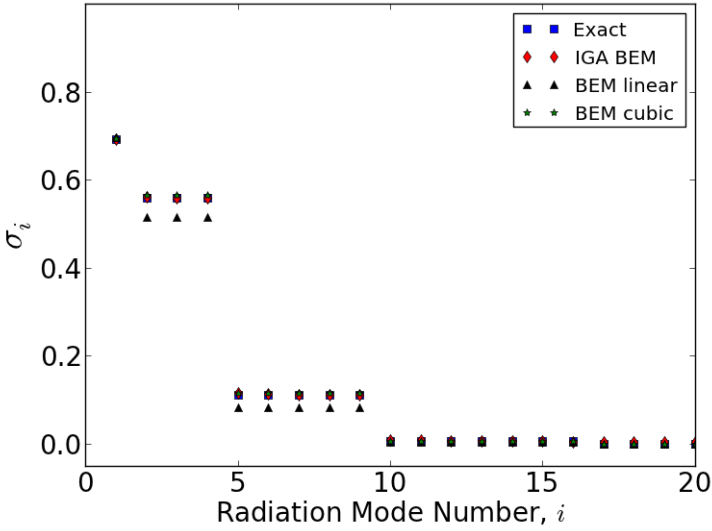
Figure 5.4 The first 20 radiation mode efficiencies for the sphere, exact and numerical.

Radiation Mode Efficiencies - Sphere				
Mode #	Exact	IGA BEM cubic	BEM linear	BEM cubic
1	0.6923	0.6919	0.6960	0.6944
2	0.5586	0.5604	0.5152	0.5649
3	0.5586	0.5592	0.5152	0.5647
4	0.5586	0.5592	0.5152	0.5645
5	0.1111	0.1156	0.0827	0.1148
6	0.1111	0.1132	0.0827	0.1148
7	0.1111	0.1116	0.0827	0.1148
8	0.1111	0.1107	0.0820	0.1145
9	0.1111	0.1107	0.0820	0.1145
10	0.00579	0.00796	0.00357	0.00613

Table 5.2 The first 10 radiation mode efficiencies for the sphere, exact and numerical.

Percent Error in Efficiencies - Sphere			
Mode #	IGA BEM cubic	BEM linear	BEM cubic
1	0.0587	0.5346	0.2967
2	0.3219	7.7680	1.1170
3	0.1077	7.7680	1.0841
4	0.1077	7.7680	1.0608
5	4.0029	25.5618	3.3160
6	1.9200	25.5618	3.3160
7	0.4812	25.5618	3.2766
8	0.3982	26.1618	3.0911
9	0.3982	26.1618	3.0911
10	37.3779	38.4304	5.8576

Table 5.3 Comparison of percent error in the radiation mode efficiencies of a sphere for IGA BEM and traditional BEM with linear and cubic elements, respectively.

As shown in Table 5.3, the percent error for IGA BEM for the first 9 modes stays below 1% for all but two modes. Modes 5 and 6 have a higher error, but still under 5%. Traditional BEM with linear elements captures the first efficiency, but then the error jumps up to around 7% and then 26%. Traditional BEM with cubic elements does better, keeping the error below 5% for the first 9 modes, but this is still not as good as IGA BEM.

For the 10th mode, the percent error for IGA BEM and traditional BEM with linear elements reaches almost 40% for mode 10, whereas the traditional BEM with cubic elements maintains a reasonable amount of error. The error for the BEM with linear elements sees a steady upward trend as the mode number increases and is due mostly to geometry error.

The jump in error for IGA BEM is due to the coarseness of the mesh used. Only 8 elements are used, and whereas this is sufficient to represent the geometry exactly, it is not sufficient to obtain the resolution in the solution that is needed for the higher modes. Using a finer mesh, as discussed in Section 3.1.2, will increase the resolution. A mesh with 32 elements and 114 nodes reduces the percent error in the 10th mode to 32.66% and a mesh with 128 elements and 266 nodes reduces it to 13.53%. This is still higher than the BEM with cubic elements, and though the finer mesh with IGA BEM does increase computational cost, it is still about 6 times faster than BEM with cubic elements.

Another factor in the error in the 10th mode can be attributed to the fact that the eigenvalue for this mode is on the order of 10^{-3}. With a value this small, numerical error is likely to come into play. Fortunately though, such a low eigenvalue corresponds to a low efficiency and so this mode - and all higher modes - are not really of interest because they do not contribute significantly to the radiated sound power.

5.2   Cylindrical Shell

Analysis was done on a cylindrical shell with $a = 0.2$, $L = 1$, $t = 0.002$, $f = 600$, $c = 343$, and $\rho_o = 1.21$. These parameters were chosen to match concurrent radiation mode work being done at BYU with a cylindrical shell in an infinite baffle [23]. The IGA mesh that describes the geometry contains 80 elements and 80 degrees of freedom. This is in contrast to the concurrent work, which uses an elemental radiator approach, and has 900 elements and 900 degrees of freedom.

Even though radiation modes are independent of boundary conditions, the two analyses are *not* identical. The analysis of the cylindrical shell in the infinite baffle essentially ignores the contributions of the interior surface of the shell to the radiation, whereas the cylindrical shell analyzed with IGA BEM *does* account for the interior surface. As an infinite baffle is a physical impossibility, IGA BEM is not able to model this situation easily. Though these two analyses are different, comparing these two problems is instructive since there is no analytic solution for the radiations modes for a cylindrical shell.

5.2.1   Radiation Mode Shapes

The radiation modes for the free cylindrical shell using IGA BEM are presented in Figure 5.5. The radiation modes for the cylindrical shell in the infinite baffle using elemental radiators are presented in Figure 5.6.

Modes 1 and 3 from the free shell match modes 3 and 1 from the baffled shell, respectively. Modes 2 and 7 are the same between the two shells, and modes 4 and 5 from the free shell matches mode 6 of the baffled shell. As expected, the rest of the modes are different between the two shells, due to the interior surface not contributing to the radiated field in the baffled shell analysis.

Mode 1 Mode 2 Mode 3 Mode 4

Mode 5 Mode 6 Mode 7 Mode 8

Mode 9 Mode 10 Mode 11 Mode 12
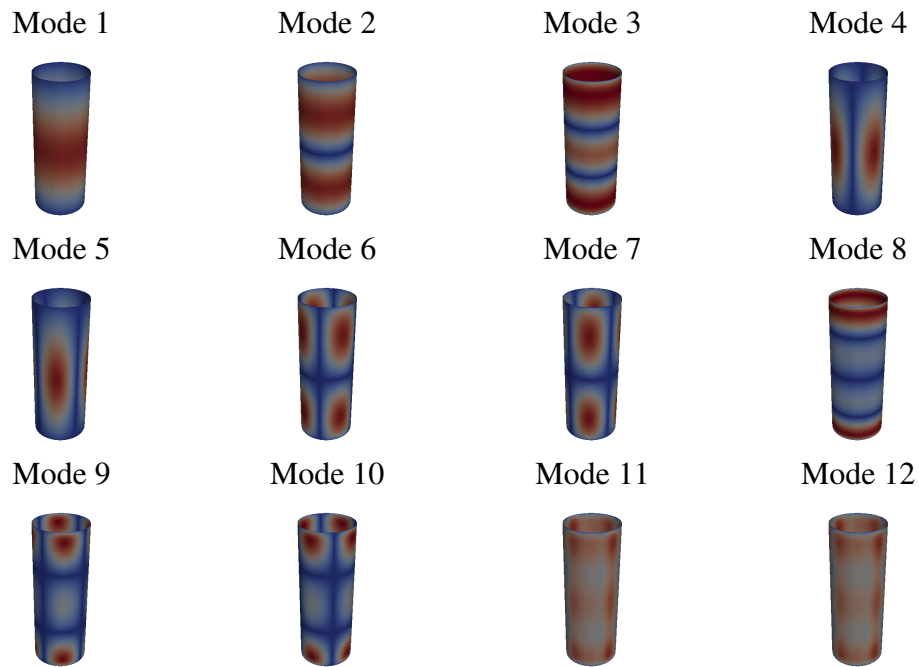
Figure 5.5 First 12 radiation modes for a free cylindrical shell, determined by IGA BEM

Mode 1 Mode 2 Mode 3 Mode 4

Mode 5 Mode 6 Mode 7 Mode 8

Mode 9 Mode 10 Mode 11 Mode 12
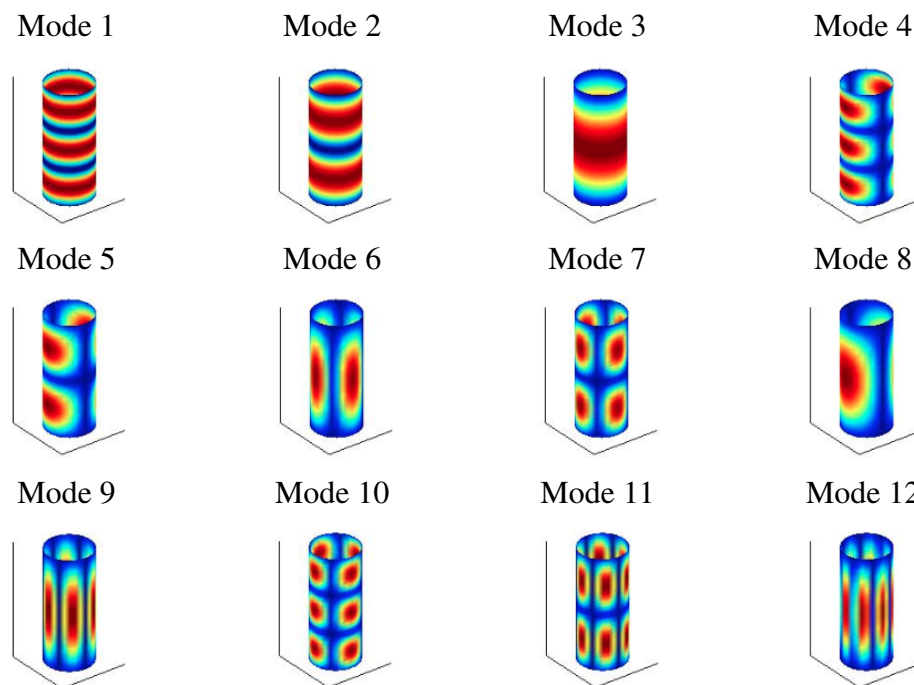
Figure 5.6 First 12 radiation modes for a baffled cylindrical shell, determined by elementary radiators

It is interesting that modes 1 and 3 are interchanged between the free shell and baffled shell. The ordering of the first three modes for the free shell seem to make more sense, with the (1,0) mode appearing before the (3,0) mode. In order to investigate this, the IGA BEM code was run on a free cylindrical shell twice as long. This increases the size of the shell relative to a wavelength. To keep the same resolution in the solution, this longer shell is described by 144 elements and 144 nodes. The first 12 radiation modes for this cylinder are shown in Figure 5.7.

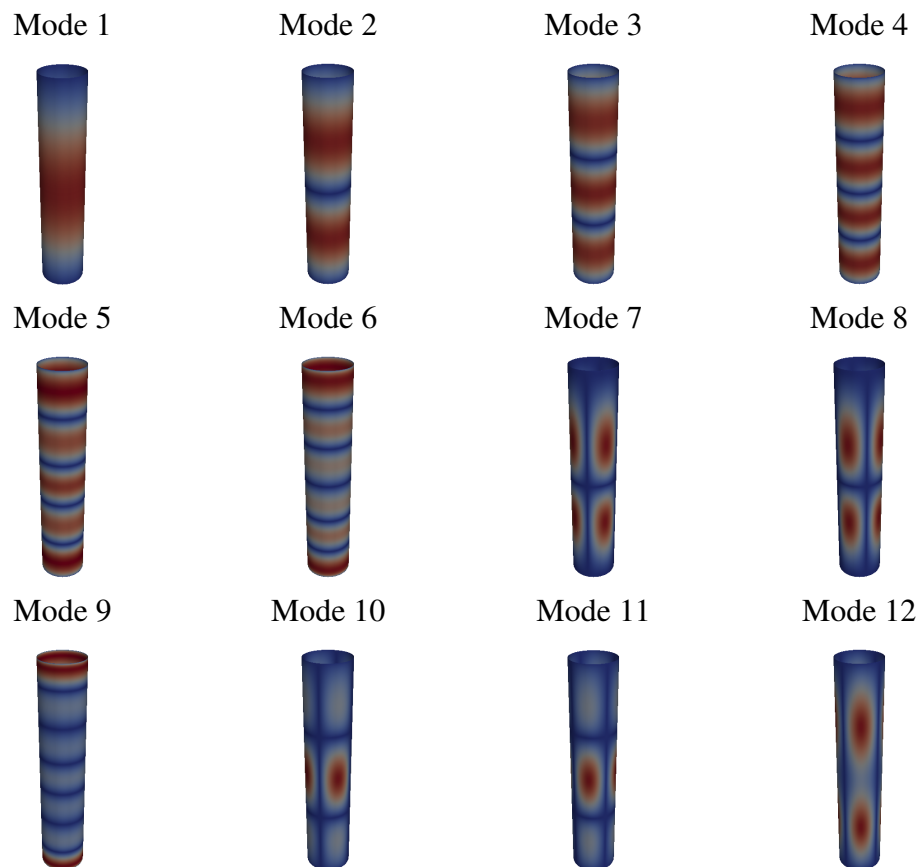**Figure 5.7** First 12 radiation modes for a free cylindrical shell twice as long, determined by IGA BEM

The same ordering of modes 1-3 is seen in the longer shell as in the original shell with IGA BEM. The change in the dimensions also introduced a reordering of the modes. Modes 4-6 are now more efficient than mode 12, while they were less efficient than mode 12 for the original shell.

5.2.2 Efficiencies

The first 20 radiation mode efficiencies for the free cylindrical shell are shown in Figure 5.8a. As can be seen, this shell is *not* a very efficient radiator. This leads to very small eigenvalues, on the order of 10^{-5} by only the 7th eigenvalue. Due to such small values, numerical error comes into play and the results are meaningless for the higher modes as the eigenvalues actually become negative (which is unphysical). This is one possible explanation for modes 11 and 12 appearing "washed out" in Figure 5.5. This is not of great concern, however, as Figure 5.8a shows that most of the radiated sound power is quantified by only the first two or three radiation modes.
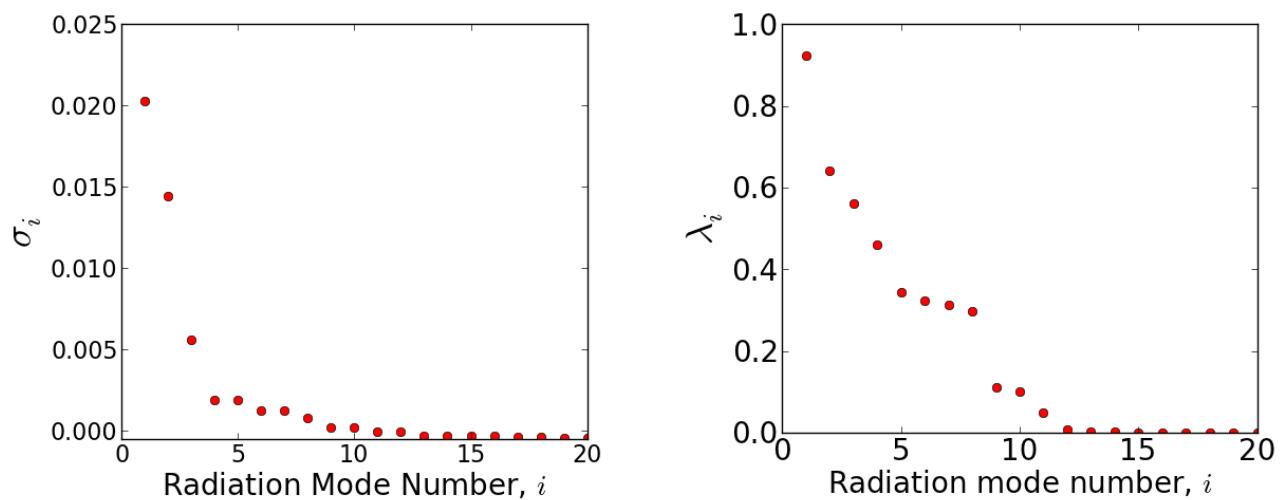
(a) Efficiencies for free cylindrical shell

(b) *Eigenvalues* for baffled cylindrical shell

Figure 5.8 The first 20 efficiencies for the free cylindrical shell are visually compared with the eigenvalues of the radiation resistance matrix for the baffled cylindrical shell.

The first 20 *eigenvalues* of the radiation resistance matrix for the baffled shell are shown in Figure 5.8b. There is no exact relationship between the eigenvalues and efficiencies defined in the literature, though they are proportional [7]. Without an explicit relationship between the eigenvalues and efficiencies for the elemental radiator approach, all that can really be compared between the free and baffled shells is the number of modes that radiate "efficiently". The baffled cylinder

seems to have more radiation modes that contribute to the radiated power. For both shells, only 11 modes would need to be included to quantify essentially all of the radiated power, and using fewer modes would provide a decent approximation of the radiated power.
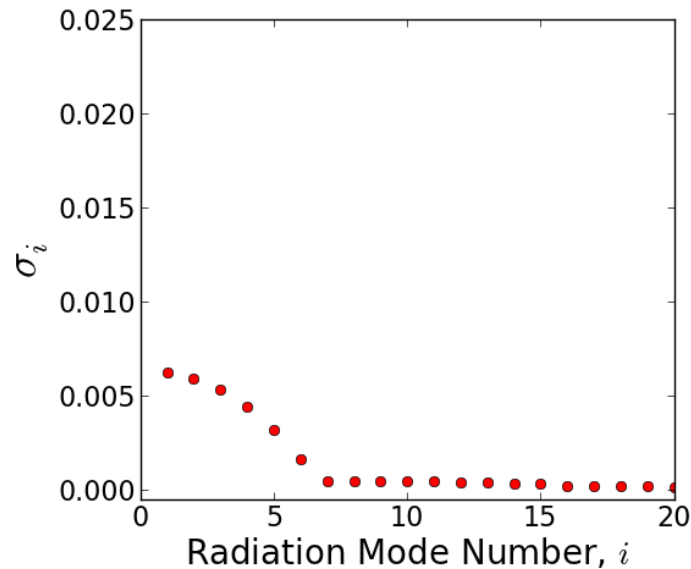
Figure 5.9 Efficiencies for longer free cylindrical shell

The first 20 radiation mode efficiencies for the longer free cylindrical shell are shown in Figure 5.9. The efficiencies are lower than for the original free shell, and there are also fewer modes that radiate. Only 6 modes would need to be included to quantify essentially all of the radiated power, but with the efficiencies so low, even a smaller number of modes could be used for a calculation.

It is hypothesized that the efficiencies are lower for the longer shell due to the increased area. This shows up in the **B** matrix in the denominator of Equation (2.6). Due to the presence of the **Z** matrix in the numerator, the effects of the area don't completely cancel out. Thus, the reference power is higher for the longer cylinder, contributing to lower radiation efficiencies.

Chapter 6

Conclusion and Future Work

By utilizing NURBS and T-Splines as basis functions, IGA unifies the design and analysis phases and offers dual benefits of increased accuracy and efficiency. Arbitrarily complex geometries are represented exactly and with fewer elements. This eliminates geometric error and approximations and reduces computational cost.

In the case of determining the acoustic radiation modes for a sphere, IGA BEM outperformed traditional BEM in capturing details, though both methods were able to capture overall shape. The main benefit of IGA BEM in this case is the computational cost. IGA BEM requires just over 30 minutes for computation, whereas traditional BEM requires multiple days, depending on which type of elements are used.

As for the radiation efficiencies for the sphere, IGA BEM shows both increased accuracy and computational efficiency. Traditional BEM with a similar number of degrees of freedom (using linear elements) is able to capture the distribution of radiation efficiencies, though the magnitude of these efficiencies is not accurate. Traditional BEM with more degrees of freedom (using cubic elements) becomes more accurate, but again, at the expense of computational cost.

A free cylindrical shell was able to be analyzed using IGA BEM. There is no analytic solution available, so numerical methods must be used. IGA again provides reduced computational cost

through exact geometry representation. The results for the free cylindrical shell were compared with an elemental radiator approach for a baffled cylindrical shell. Some similar radiation modes were obtained, seeming to validate the results obtained for each shell.

IGA BEM offers the additional benefit of ease of implementation. With the T-Spline plug-in for Rhino, a designer can quickly and easily create a design and export it for immediate analysis with the IGA BEM code. This will allow for quick and accurate analysis of radiated sound power for arbitrary geometries.

This work demonstrates the benefits of using IGA BEM to determine acoustic radiation modes. Additional work on error comparison, selection sets, analyzing complex shapes, and calculating radiated sound power will prove to make the tools developed here more robust.

The comparison between the spherical harmonics and the radiation mode shapes for the sphere using IGA BEM at this point is only visual. An error calculation was added, however it is not running properly yet due to the normalization that is done. Getting this to work properly will provide more conclusive evidence of just how accurate IGA BEM is in capturing the radiation mode shapes.

As mentioned, using the selection sets in Rhino to analyze partially vibrating sources is a potentially useful tool but it is not complete yet. There are two possibilities for the completion of this tool. The first is to specify boundary conditions on a set of elements, as opposed to a set of nodes. The code now accepts element sets, but undesirable tapering happens in order to force the elements not in the selection set to be held at zero. This problem is the opposite of the spillover seen when specifying node sets.

The second, and more promising, possibility is to use a discontinuous basis to handle the boundary between the vibrating and non-vibrating portions of the structure. This approach would still use the node set boundary conditions. Work needs to be done to implement this discontinuous basis, and it is unclear at this time whether using a discontinuous basis is consistent with the

requirements for the radiation mode development used here or whether modifications would need to be made.

It took longer than anticipated to get the code working for the sphere and the free cylindrical shell. Because of this, no other geometries were able to be analyzed. The geometry must be created in Rhino with T-Splines, and Rhino presents a steep learning curve to someone with no design experience. Having someone with design experience create more complex geometries will allow the code developed here to be applied to more interesting cases, which will strengthen the case for using IGA BEM to determine the radiation modes.

The power formulation in terms of the radiation modes and their efficiencies has been described, but no power calculation was done. To further show the utility of IGA BEM, power can be calculated for a specified surface velocity distribution, and the accuracy and convergence rate can be compared with that of traditional BEM as well as elemental radiator methods.

Appendix A

Python Scripts

Below are the Python scripts that were developed for pre- and post-processing of the IGA BEM C++ code that Simpson developed. Listing A.1 creates and saves the **B** and **Z** matrices. Listing A.2 solves for the efficiencies and radiation modes (coefficients of the basis functions), then feeds these coefficients back into Simpson's code in order to visualize the mode shapes.

Listing A.1 makeBZ.py

```python
shape = raw_input('sphere, bem, or cylinder? ')

if shape != 'bem':
  # open original .ext input file
  ext = open('../input/inverted_'+shape+'_base.ext',"r")
  kind = ''                # element kind. only used for BEM
  node = []               # string of node sets
  nodesets = []       # array of node sets
  selset = None      # selection set
  valid = []               # valid nodes (not in selection set)

  # read info from .ext file
  line = ext.readline()
  while line:
    if 'SelSet' in line:
      selset = line             # list of elements in selection set
    elif 'elemN' in line:
      elems = line              # number of elements
    elif 'nodeN' in line:
      nodes = line              # number of nodes
    elif 'elem ' in line:
```

```python
      line  = ext. readline ()
      node.append( line )            # write  the  line  with  the  element node set
    line  = ext. readline ()
  ext. close ()

  elems = [ int ( i ) for  i  in  elems. split (' ' )[1:]][0]     # convert  to  integer
  nodes = [ int ( i ) for  i  in  nodes. split (' ' )[1:]][0]     # convert  to  integer

  if  selset  != None:
    selset  = [ int ( i ) for  i  in  selset . split (' ' )[4:]]        # array  of  elements  in  selection  set
  else :
    selset  =  list ( range (0, elems ))    # all  elements  in  selection  set  if  no  set  specified

  # store  the  nodesets  for  each  element
  for  i  in  range (0, elems ):
    nodesets . append([ int ( s ) for  s  in  node[ i ]. split (' ' )])

  # use  only  the  nodes  that  are  in  the  selection  set
  validnodesets  = [ nodesets [ i ] for  i  in  selset ]
  for  i  in  validnodesets :
    if  i  not  in  valid :
      valid . extend( i )        # store  unique  values  of  validnodesets  in  valid
  valid  = array ( sorted ( set ( valid )))

# for  BEM
else :
  kind = raw_input(' linear ,  quad, or  cubic? ')         # element  kind

  # open and  read  info  from  original  .msh input  file
  msh = open('sphere_'+kind+'.msh')
  line  = msh. readline ()
  while  line :
    if  '$Nodes' in  line :
      line  = msh. readline ()
      nodes = int ( line )  − 1      # number of nodes
      break
    line  = msh. readline ()
  msh.close ()
  valid  = array ( range (2, nodes+2))    # nodes  numbered  starting  with  2
save (' valid '+shape, valid )

# define  acoustic  parameters
if  shape == ' cylinder ':
  w = 1
  c  = 343/(2∗pi∗600)
else :
  w = 1
  c  = 1./3
rho  = 1.21
k = w/c
```

```python
# remove .txt  file
try :
  os.remove(" potential . txt ")
except  OSError:
  pass

#  initialize   pressure  and impedance matrices
pressmat  = zeros (shape=(nodes,nodes), dtype='c16')
Z = matrix ( zeros (shape=(len ( valid ), len ( valid )), dtype='c16' ))

# set  each  valid  node to  unit   velocity   one at  a time .
# nodes not  in  valid  array  will  be  left  at  0.
for  n  in   valid :
  # open original  and create  new .ext input   files  for  each  BC
  if  shape != 'bem':
    ext  = open(" ../ input / inverted_ "+shape+"_base.ext","r")
    myext = open(" inverted_ "+shape+"_base.ext","w")

    # read  info  from  . ext
    for  line  in  ext . readlines ():
      myext.write ( line )
    myext.write ('\n')
    ext . close ()

    # create  the  new node sets  − 1 active  node set  to 1 and the  rest  set  to 0
    myext.write ( "' set  1 node active_node  '" )
    myext.write ( str (n)  + "\n")
    myext.write ( "' set   '" )
    myext.write ( str (nodes−1))
    myext.write ( "'  node zero_nodes  '" )
    for  node in  range (0, nodes):
      # don't  include  the  current  node in  the  other_nodes
      if  node != n:
        myext.write ( str (node)  + " ")
    myext.close ()

    # open original  and  create  new .xml input  file
    xml = open(" ../ input / input_sphere_acoustic .xml","r")
    myxml = open("input_"+shape+".xml", "w")

    for  line  in  xml. readlines ():
      # choose input  . ext  file
      if  ' inverted '  in  line :
        myxml.write( "' <Parameter name="base" type=" string " value=" inverted_  '" )
        myxml.write(shape)
        myxml.write( "' _base"/> '" )
        myxml.write('\n')
      # rename . vtu  file  to  reflect  geometry type
      elif  ' sphere_results '  in  line :
        myxml.write( "' <Parameter name="base" type=" string " value=" '" )
```

```python
      myxml.write(shape)
      myxml.write( '"  _results "/> "' )
      myxml.write('\n')
   # specify wavenumber, k
    elif 'name="k"' in line :
      myxml.write( '" <Parameter name="k" type="double" value=" "' )
      myxml.write( str (k))
      myxml.write( '" "/> "' )
   # write the BC. dp/dn = -j*rho*w*v. v = 1
    elif 'all_nodes' in line :
      myxml.write( '" <BC name="active_node" dof="0" input="complex" realvalue ="0.0" imagvalue=" "' )
      myxml.write(str (-rho*w))
      myxml.write( '" "/>
<BC name="zero_nodes" dof="0" input="complex" realvalue="0.0" imagvalue="0.0"/> "' )
      myxml.write('\n')
   # write the rest of the file exactly
    else :
      myxml.write( line )
  xml.close ()
  myxml.close()

# for BEM
else :
  # open and create new .als input file
  als  = open(" acoustic_sphere . als ","r")
  myals = open("unitBC.als ","w")

  for line in als . readlines ():
    # define which .msh input file to read from
    if 'sphere' in line :
      myals. write ( '" sphere_ "' )
      myals. write (kind+'.msh continuous\n')
    # define first (active) node set of 1 node
    elif '{{ first ' in line :
      myals. write ( '" {{ first_node_set 1 "' )
      myals. write ( str (n) + " }\n")
    # define second node set of remaining nodes set to 0
    elif '{second' in line :
      myals. write ( '" {second_node_set "' )
      myals. write ( str (nodes-1) + ' ' )
      for node in range (2, nodes+2):
        # don't include the current node in the zero node
        if node != n:
          myals. write ( str (node) + " ")
      myals. write ('}}\n')
    # set the wavenumber, k
    elif 'wavenumber' in line :
      myals. write ('matprop wavenumber '+str(k)+'\n')
```

```python
    # write the rest exactly
     elif 'node_set' not in line :
        myals.write ( line )
   # set the BC. dp/dn = 1
   myals.write ( ''' bc neumann first_node_set  0  1
bc neumann second_node_set 0 0 ''' )
   als . close ()
   myals.close ()

press = []         # empty array for pressure data
if shape != 'bem':
   # run the solver
   print '\ndof = ', where( valid==n)[0][0]+1, '/', size ( valid )
   subprocess . call ([ "./tsbem", "input_"+shape+".xml"])

   # read the pressure from the output of the solver
   Press = genfromtxt(' potential . txt ', dtype='c16', delimiter =' ', skip_header=1, skip_footer =2)

   # combine real and imag parts into one entry
   for m in arange (1, nodes ∗2,2):
      press . append(Press[m] + Press[m+1]∗1j)

   # store pressure as column of pressure matrix
   pressmat [:, n] = press

   # clean up files
   os.remove("input_"+shape+".xml")
   os.remove("inverted_"+shape+"_base.ext")
   os.remove(shape+"_results . vtu")

# for BEM
else :
   # run solver
   print '\ndof =',n−1,'/', size ( valid )
   subprocess . call ([ "./stdbem", "unitBC. als "])

   # read the pressure from the output of the solver
   Press = genfromtxt(' potential . txt ',dtype='c16', delimiter =' ', skip_header=3)

   # combine real and imag parts into one entry
   for m in arange (0, nodes ):
      press . append(Press[m]+Press[m+nodes]∗1j)

   # store pressure as column of pressure matrix (n starts at 2)
   pressmat [:, n−2] = press

   # clean up files
   os.remove("unitBC. als")
os.remove(" potential . txt ")
```

```python
if shape != 'bem':
  # only keep the rows/cols associated with valid nodes
  pressmat = pressmat[valid ,:][:, valid]

  # P = Zv. Due to BCs, v is identity
  Z = pressmat

  B = matrix(zeros(shape=(nodes,nodes)))
  # read number of non-zero entries in B matrix
  with open('B matrix.txt') as file:
    NZ = int(genfromtxt(itertools.islice(file,5))[-1,-1])

  for n in range(0,NZ):
    # only read in one row at a time and construct B
    info = genfromtxt('B matrix.txt', dtype=float, skip_header=18+n, skip_footer=NZ-1-n)
    row = info[1]
    col = info[2]
    val = info[3]
    B[row,col] = val

  # only keep the rows/cols associated with valid nodes
  B = B[valid ,:][:, valid]

# for BEM
else:
  # only keep the rows/cols associated with valid nodes
  pressmat = pressmat[valid -2,:][:, valid -2]

  # P = Zv. Due to BCs, v is identity /(-j*w*rho)
  Z = -1j*w*rho*pressmat

  # read in B matrix and only keep rows/cols associated with valid nodes
  B = matrix(genfromtxt('B Matrix.txt', dtype=float, skip_header=6))
  B = B[valid -2,:][:, valid -2]

# save B and Z matrices
save('Z_'+shape+'_'+kind,Z)
save('B_'+shape+'_'+kind,B)
```

Listing A.2 rad_modes.py

```python
shape = raw_input('sphere, bem, or cylinder? ')

if shape != 'bem':
  kind = ''      # element kind. only used for BEM

  # open and read info from original .ext input file
  ext = open('../input/inverted_'+shape+'_base.ext',"r")
  line = ext.readline()
  while line:
    if 'nodeN' in line:
      nodes = line      # number of nodes
        break
    line = ext.readline()
  ext.close()
  nodes = [int(i) for i in nodes.split(' ')[1:]][0]     # convert to integer
else:
  kind = raw_input('linear, quad, or cubic? ')     # element kind

  # open and read info from .msh input file
  msh = open('sphere_'+kind+'.msh')
  line = msh.readline()
  while line:
    if '$Nodes' in line:
      line = msh.readline()
      nodes = int(line) - 1      # number of nodes
      break
      line = msh.readline()
  msh.close()

# define acoustic parameters
if shape == 'cylinder':
  c = 343/(2*pi*600)
  w = 1
else:
  c = 1./3
  w = 1
rho = 1.21
k = w/c

# load valid nodes, B, and Z
valid = load('valid'+shape+'.npy')
Z = matrix(load('Z_'+shape+'_'+kind+'.npy'))
B = .5*matrix(load('B_'+shape+'_'+kind+'.npy'))

# cholesky decomposition of B
L = matrix(cholesky(B))

# normalized radiation matrix
Q = (L.T*Z*inv(L).T).real
```

```python
# eigenanalysis
vals , psi = eig(Q)

# sort eigenvectors by decreasing eigenvalues
ind = list ( argsort ( vals ))
ind . reverse ()
psi = psi [:, ind ]
vals = sorted ( vals , reverse =True)

# back−transformed eigenvectors
gamma = inv(L).T∗psi

# plot efficiencies (lambda/rho/c)
figure ()
plot ( matrix ( vals ). T. real /( rho∗c ), 'ro')
xlabel ('Radiation Mode Number, $i$')
ylabel ('$\sigma_{i}$')

# now use eigenvectors as input BC to visualize mode shapes
if shape != 'bem':
  # open and create new .ext input files for each BC
  ext = open(" ../ input / inverted_ "+shape+"_base.ext","r")
  myext = open("inverted_ "+shape+"_base.ext","w")

  # read info from .ext
  for line in ext . readlines ():
    myext. write ( line )
  myext. write ('\n')
  ext . close ()

  # create the new node sets − each node in its own node set
  for m in valid :
    myext. write ( "' set 1 node node '" )
    myext. write ( str (m) + " " + str (m) + "\n")

  # set any non−valid nodes to zero
  myext. write ( "' set '" )
  myext. write ( str (nodes−len(valid )) + " node zero_nodes ")
  for node in range (0, nodes ):
    if node not in valid :
      myext. write ( str (node) + " ")
  myext. close ()

# visualize the first 20 modes
for n in range (0,20):
  if shape != 'bem':
    # open original and create new .xml input file
    xml = open(" ../ input / input_sphere_acoustic .xml","r")
    myxml = open("input_"+shape+".xml", "w")
```

```python
    # read / write  info  from . ext  file
    for  line  in  xml. readlines ():
      # choose  input  . ext  file
      if  ' inverted '  in  line :
        myxml.write( '''         <Parameter name="base" type=" string "  value=" inverted_  ''' )
        myxml.write(shape)
        myxml.write( ''' _base"/> ''' )
        myxml.write('\n')
      # rename . vtu  file  to  reflect  geometry type  and mode number
      elif  ' sphere_results '  in  line :
        myxml.write( '''         <Parameter name="base" type=" string "  value=" ''' )
        myxml.write(shape)
        myxml.write( ''' _results  ''' )
        myxml.write( str (n))
        myxml.write( ''' "/> ''' )
        myxml.write('\n')
      # specify  wavenumber, k
      elif  'name="k"'  in  line :
        myxml.write( '''         <Parameter name="k" type="double" value=" ''' )
        myxml.write( str (k))
        myxml.write( ''' "/> ''' )
      # write  the  BC. Each valid  node gets  a   coefficient   from gamma
      # each non−valid node is  set  to  0
      # IGA BEM code normalizes so max value = 1
      elif  ' all_nodes '  in  line :
        q = 0
        for  m in  valid :
          myxml.write( ''' <BC name="node"''' )
          myxml.write( str (m))
          myxml.write( ''' "  dof="0"  input="complex"  imagvalue=" ''' )
          myxml.write( str (gamma[q,n].imag))
          myxml.write( ''' "   realvalue =" ''' )
          myxml.write( str (−gamma[q,n].real))
          myxml.write( ''' "/> ''' )
          myxml.write('\n')
          q  += 1
        myxml.write( ''' <BC name="zero_nodes" dof="0" input="complex"
          realvalue ="0.0"  imagvalue="0.0"/> ''' )
        myxml.write('\n')
      # write  the  rest  of  the  file  exactly
      else :
        myxml.write(line )
    xml. close ()
    myxml.close()

  # for  BEM
  else :
    # open original  and create  new . als  input  file
    als  = open(" acoustic_sphere . als ", "r")
    myals = open("gammaBC.als","w")
```

```python
    # define which .msh input file to read from
    myals.write('" sphere_ "')
    myals.write(kind+'.msh continuous')
    myals.write('\n{')
    # create node sets of one node each
    for m in range(1,nodes+1):
      myals.write('{node_set'+str(m)+' 1 ')
      myals.write(str(m+1))
      if m != nodes:
        myals.write('" }\n "')
      else:
        myals.write('" }}\n "')

    for line in als.readlines():
      # specify wavenumber, k
      if 'wavenumber' in line:
        myals.write('matprop wavenumber '+str(k) + '\n')
      # write the rest of the file exactly
      elif 'sphere' not in line and 'node_set' not in line:
        myals.write(line)
    # set BC. each node gets a coefficient from gamma. normalize so max value = 1
    for m in range(0,nodes):
      myals.write('" bc neumann node_set "')
      myals.write(str(m+1) + ' ' + str(0) + ' ' + \
          str((gamma[m,n].real)/float(max(abs(gamma[:,n].real)))) + '\n')
    als.close()
    myals.close()

  # run the solver
  print '\ndof =',n
  if shape == 'bem':
    subprocess.call(["./stdbem", "gammaBC.als"])
    os.rename('solution.vtu','solution_'+shape+'_'+kind+'_'+str(n)+'.vtu')
  else:
    subprocess.call(["./tsbem", "input_"+shape+".xml"])

  # clean up files
  try:
    os.remove("input_"+shape+".xml")
  except OSError:
    pass
  try:
    os.remove("gammaBC.als")
  except OSError:
    pass
  os.remove("potential.txt")
try:
  os.remove("inverted_"+shape+"_base.ext")
except OSError:
  pass
```

Bibliography

[1] G. V. Borgiotti, "The Power Radiated By a Vibrating Body in an Acoustic Fluid and its Determination From Boundary Measurements," J. Acoust. Soc. Am. **88,** 1884–1893 (1990).

[2] K. Naghshineh, "Active Control of Sound Power Using Acoustic Basis Functions as Surface Velocity Filters," J. Acoust. Soc. Am. **93,** 2740–2752 (1993).

[3] M. E. Johnson and S. J. Elliott, "Active Control of Sound Radiation Using Volume Velocity Cancellation," J. Acoust. Soc. Am. **98,** 2174–2186 (1995).

[4] S. J. Elliott and M. E. Johnson, "Radiation Modes and the Active Control of Sound Power," J. Acoust. Soc. Am. **94,** 2194–2204 (1993).

[5] K. A. Cunefare and M. N. Currey, "The Radiation Efficiency Grouping of Free-Space Acoustic Radiation Modes," J. Acoust. Soc. Am. **109,** 203–215 (2001).

[6] H. Wu and W. Jiang, "Analyzing Acoustic Radiation Modes of Baffled Plates with a Fast Multipole Boundary Element Method," J. Vib. Acoust. 135 (2013).

[7] A. Sarkissian, "Acoustic Radiation From Finite Structures," J. Acoust. Soc. Am. **90,** 574–578 (1991).

[8] R. Visser, Ph.D. thesis, University of Twente, 2004.

[9] J. A. Cottrell, T. J. R. Hughes, and B. Yuri, *Isogeometric Analysis: Toward Integration of CAD and FEA* (Wiley, 2009).

[10] J. A. Cottrell, A. Reali, Y. Bazilevs, and T. J. R. Hughes, "Isogeometric Analysis of Structural Vibrations," Comput. Methods Appl. Mech. Engrg. **195,** 5257–5296 (2006).

[11] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs, "Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry and Mesh Refinement," Comput. Methods Appl. Mech. Engrg. **194,** 4135–4195 (2005).

[12] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, and T. J. R. Huges, "Isogeometric Finite Element Data Structures Based on Bézier Extraction of T-Splines," Int. J. Numer. Meth. Engng **88,** 126–156 (2011).

[13] M. J. Borden, M. A. Scott, J. A. Evans, and T. J. R. Hughes, "Isogeometric finite element data structures based on Bézier extraction of NURBS," Int. J. Numer. Meth. Engng. **87,** 15–47 (2011).

[14] R. N. Simpson, M. A. Scott, M. Taus, D. Thomas, and H. Lian, "Acoustic Isogeometric Boundary Element Analysis," Comput. Methods Appl. Mech. Engrg. **269,** 265–290 (2014).

[15] K. A. Cunefare and M. N. Currey, "On the Exterior Acoustic Radiation Modes of Structures," J. Acoust. Soc. Am. **96,** 2302–2312 (1996).

[16] M. N. Currey and K. A. Cunefare, "The Radiation Modes of Baffled Finite Plates," J. Acoust. Soc. Am. **98,** 1570–1580 (1995).

[17] D. M. Photiadis, "The Relationship of Singular Value Decomposition to Wave-Vector Filtering in Sound Radiation Problems," J. Acoust. Soc. Am. **88,** 1152–1159 (1990).

[18] S. D. Snyder and N. Tanaka, "Calculating Total Acoustic Power Output Using Modal Radiation Efficiencies," J. Acoust. Soc. Am. **97,** 1702–1709 (1995).

[19] F. Fahy and P. Gardonio, *Sound and Structural Vibration*, 2nd ed. (Academic Press, 2007).

[20] A. Buffa, J. Rivas, G. Sangalli, and R. Vazquez, "Isogeometric Discrete Differential Forms in Three Dimensions," SIAM J. Numer. Anal. **49,** 818–844 (2011).

[21] G. Buffa, A.and Sangalli and R. Vazquez, "Isogeometric analysis in electromagnetics: B-splines approximation," Comput. Methods Appl. Mech. Engrg. **199,** 1143–1152 (2010).

[22] D. T. Blackstock, *Fundamentals of Physical Acoustics* (Wiley-Interscience, 2000).

[23] P. Aslani and S. D. Sommerfeldt, "Development of radiation mode shapes for cylindrical shells," J. Acoust. Soc. Am **134,** 4191 (2013).