The Boundary Element Method for Linear Acoustic Systems

Simon de Leon

Music Technology Area, Department of Theory
Schulich School of Music
McGill University
Montreal, Canada

February 2008

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Arts in Music Technology.

Abstract

A technique based on the transient boundary element method for the numerical simulation of three-dimensional linear acoustic systems is presented. Using surface meshes generated by `Distmesh` [1], pressure is extracted from points within the enclosed volume of the mesh by integration of retarded potentials across the boundary. As a result, mesh junction calculations and topology-related inaccuracies within the volume are avoided. For simulations of struck instruments, membrane motion is modeled using two-dimensional digital waveguide techniques adapted for an unstructured triangular topology and coupled with the internal pressure field. The method is also used to extract response characteristics of an open, unflanged pipe and the results are compared to a one-dimensional waveguide model. Finally, possible future work using the freely available software developed for this project is discussed.

Sommaire

Une technique de simulation numérique des systèmes acoustiques linéaires tridimensionnels basée sur la méthode d'éléments de frontière transitoires est présentée. La librairie Matlab `Distmesh` [1] a été employée pour générer des maillages représentant la surface de tels objets. La pression en tout point du volume est obtenue par intégration des potentiels tout au long du maillage. Ainsi les calculs aux noeuds du maillage de même que les inexactitudes de topologies connexes dans le volume sont évités. Dans le cadre de la simulation d'instruments percussifs, le mouvement de la membrane est modélisé au moyen de guides d'ondes bidimensionnels particulièrement bien adaptés à une topologie triangulaire non-structurée. L'effet de couplage dû aux pressions internes est aussi pris en compte. D'autre part, la réponse caractéristique d'un tube droit, ouvert aux deux bouts obtenue par cette méthode est comparée aux résultats d'une méthode basée sur les guides d'ondes monodimensionnels. Enfin, de possibles applications du logiciel developpé pendant ce projet (et librement disponible) sont présentées.

Acknowledgments

I would like to gratefully acknowledge Prof. Gary Scavone for his supervision, guidance, and expertise during this work. I also thank the students, faculty and staff of the Music Technology Area at McGill University for their advice, insight and friendship. In particular, I would like to thank the members of the McGill Image to Audio Conversion project, the members of the McGill Digital Orchestra project, as well as Darryl Cameron, Beinan Li and Andrey R. da Silva for their assistance and support throughout my graduate work. Thank you to William Chapin and Joel Storckman of AuSIM Inc. for the gracious use of their facilities and services during the final stages of this work. Special thanks to Bertrand Scherrer for providing some last minute editing work.

I would also like to thank my friends and family for providing me with endless memories and inspiration throughout my studies. I cannot thank my parents Jose and Natividad de Leon enough with words alone. Your dedication, perseverence, and love remain with me always and guide me through my most difficult times, even from many miles away. Finally, I am forever indebted to Emily Ng for her tireless support, understanding, and encouragement throughout my entire academic career.

Contents

List of Figures

List of Acronyms

BEM	Boundary Element Method
TBEM	Transient/Time-Domain Boundary Element Method
FBEM	Frequency-Domain Boundary Element Method
FEM	Finite Element Method
LBM	Lattice Boltzmann Method
CFD	Computational Fluid Dynamics

Chapter 1

Introduction

The boundary element method (BEM) is a technique used for the numerical simulation of field problems described by boundary integral equations. In engineering applications from geomechanics to elastodynamics, BEM has enjoyed popularity as an alternative to the finite element method (FEM) where the discretization of the entire region of interest is impractical or unnecessary. The time domain variants of BEM in acoustics are based on Kirchhoff's method of integration, where an appropriate Green's function is applied to the inhomogeneous scalar wave equation to arrive at an integral solution [2]. This allows the pressure variation in a volume to be determined completely from time delayed boundary values and avoids the need to discretize the entire volume. The retarded potential technique approximates the numerical solution to the classical Kirchhoff solution to the wave equation and is used as the basis for this study [3].

Although BEM is widely utilized in industrial acoustics, including automotive and architectural applications, it has received less attention in musical contexts. This is primarily due to the much greater computational effort required in comparison with other physical modeling methods optimized for realtime applications, such as digital waveguide and wave digital techniques [4], and the functional transformation method [5]. Additionally, the implementation efforts for meshing and BEM algorithms in arbitrary domains are considerable and commercial software packages are costly. The software developed for this project is freely available under the GNU General Public License (GPL) and is intended as an initial framework for simulations of the generation and propagation of sound. More importantly, it serves as a generalizable technique that is less dependent on ad hoc constructions and

empirically tuned filters.

The objective of this study is to demonstrate that our variation of the retarded potential technique is a viable method for the simulation of three-dimensional linear acoustic systems and that it can be modified to suit various applications. To accomplish this, the simulation of an unflanged cylinder is presented and compared against the result of a 1D waveguide model. By making boundary condition assumptions at the open end, it is shown to be in general agreement with the 1D approximation. Further, a 2D digital waveguide drum membrane model is coupled with the retarded potential technique to illustrate its application to struck instrument simulations. These examples demonstrate the suitability of such techniques for multimedia and virtual reality applications as computational capacity increases.

1.1 Context and Motivation

Modeling physical objects for sound generation is becoming increasingly popular as computational capacity increases. For musical applications, this is apparent in the number of commercially available software packages that serve to emulate other instruments and acoustic spaces. This offers composers the ability to generate realistic compositions from a single computer. For video games and virtual reality applications, the motivation for physically-based sound generation comes from the growing difficulty in managing and maintaining the ever-increasing sound databases that are associated and triggered with specific events. Since these applications are interactive, a seemingly infinite number of sounds could be generated by the user when interacting with objects in the virtual environment.

Physical models developed for musical applications have traditionally focused on techniques capable of realtime sound generation. For this purpose, waveguide theory is the most widespread technique for computationally efficient models of musical instruments. These structures are typically application-specific filters and are difficult to adapt to gross changes in shape or material without a re-design of the filter structure. Furthermore, several components of waveguide models often require measurements of the system being modeled and thus lose relevance in predictive applications where one would be developing a prototype or interacting with an arbitrary object.

Computational fluid dynamics (CFD) offers several alternatives for modeling acoustics but sacrifices computational efficiency for accuracy. Finite element modeling (FEM) is a

highly utilized numerical method often employed to solve the Navier-Stokes equations in arbitrary domains and has come to be accepted as the de facto standard in aeroacoustics, among other fields. BEM is also popular in predictive applications of acoustics due to its ability to handle large volumes much more efficiently than FEM by reducing the dimensions of the problem space. However, this reduction is made possible by making simplifying assumptions of the problem space, thus limiting BEM to linear applications with homogeneous properties of the fluid. Its frequency domain variant is most popular in engineering applications.

We are interested in merging concepts from the time-domain variant of BEM from the CFD realm and physical modeling of musical instruments in order to develop a generalizable framework for wave propagation simulation in future multimedia applications. These applications include animated film, video games and virtual reality, as well as reverberation and instrument emulation. In such scenarios, there may be an infinite number of sound generation scenarios due to user interaction, and thus a CFD engine capable of generating sound in arbitrary volumes is necessary. Furthermore, the target applications are concerned with transient effects as opposed to steady state analysis, so a time-domain BEM algorithm is chosen. Where the motion of an object would be fast enough to generate perceivable audio, the waveguide junction method [4] is utilized due to its analogous relationship with wire frame meshes used in computer animation.

1.2 Project Overview

This study implements software to demonstrate the application of time-domain BEM to multimedia applications. The meshing algorithm used to define the surfaces of objects is based on `Distmesh` [1], a set of open-source `MATLAB` scripts that generates unstructured triangular meshes. Two main simulations were performed from the set of `MATLAB` scripts developed to accomplish this:

- The simulation of an unflanged pipe with a closed and open end. An initial plane wave propagates from the closed end towards the open end. This demonstrates the accuracy of the method chosen.
- The simulation of a djembe drum in response to a centered strike on its membrane. This demonstrates coupled simulation with mesh motion objects similar to the wire

frame models used in animation.

Simplicity and generality were emphasized during the development of the software and as such, should serve as a starting point for those with specific applications in mind. As a result, the software is made freely available under the GNU GPL to encourage further development and optimization work.

1.3 Thesis Overview

The remainder of this thesis is organized into four chapters. As mentioned previously, the software developed for this project is available under the GNU GPL and can be obtained from `http://www.music.mcgill.ca/~deleon/research.html`. After familiarizing oneself with the theoretical and practical aspects of this work, the software implementation is an excellent resource for those interested in three-dimensional wave propagation.

Chapter 2 covers the technical background necessary to understand the transient boundary element method, the retarded potential technique, and waveguide junction modeling. The chapter concludes by discussing related work that draws from computational fluid dynamics for applications in music.

Chapter 3 discusses our variation of the retarded potential technique in detail. Specifically, we outline several shortcomings in the retarded potential technique that motivated the modifications to the algorithm. Additional topics covered in this chapter include an adaptation of the waveguide junction technique for unstructured triangular topologies, the quadrature technique used for element integration, and ray-triangle intersection used to determine line-of-sight connectivity between mesh elements.

Chapter 4 presents the results for the pipe and djembe simulations. These exercises highlighted several technical issues which are discussed in detail. Insight into the results and technical issues encountered during the development of this project lead to the evaluation and future work discussions in the fifth and final chapter.

1.4 Contributions and Limitations

The simulation of arbitrary acoustic objects with the transient boundary element method is theoretically limited to homogeneous, linear and adiabatic volumes. These limitations

are implied by the use of the inhomogeneous scalar wave equation. It should be noted that the term inhomogeneous in regards to the scalar wave equation refers to its inhomogeneous property as a differential equation and should not be confused with a physical property of the volume.

As such, the transient boundary element method as it is implemented here should not normally be coupled with surfaces exhibiting perforations. As discussed briefly in Chapter 3, perforated surfaces should be treated with indirect BEM solvers such as those detailed in [6] and [7]. However, for the limited scope of this research, we chose to approximate perforated surfaces with a zero pressure boundary condition. This allowed us to verify that solving the three-dimensional wave equation was inline with the results of empirically tuned solutions, but the results should not be considered rigorous. Furthermore, all BEM methods are linear and cannot account for higher order phenomena such as turbulence. Therefore, cases that would require such precision should consider a Navier-Stokes solver.

This research extends the physical modeling properties of digital waveguides into three dimensions and allows sound extraction of three-dimensional modes directly from physical variables (such as those commonly used in virtual reality simulations) which do not have direct physical correlation with one-dimensional methods. This is at the cost of computational effort, but is much more suitable in the near-future for sub-laminar flow than the use of a Navier-Stokes solver due to its reduction of the integration space by one dimension. Simulations of bodies with perforations would either need to make use of empirically tuned filters (such as those used with digital waveguides) at each mesh element belonging to a perforation, use an indirect BEM solver, or bidirectionally couple the BEM inside the body to a BEM model outside of the body. If inhomogeneous properties of the volume around the perforation are to be considered, it is also possible to couple the BEM with a Navier-Stokes solver in the region of interest.

The key technical and innovative contributions of this research are the augmentation of the retarded potential technique with Gaussian surface integration and unstructured triangular topology meshing, as well as its coupling with a dynamically moving mesh. This allows pressure and surface mesh motion to influence each other. Additionally, the mesh variable structure allows the user to assign unique and independent surface characteristics to each mesh element to determine its response to both motion variables (waveguide junction method) and pressure (retarded potential technique). Furthermore, the unstructured topology of the mesh allows one to mesh model three-dimensional objects more easily than

with fixed meshes. Further innovation includes extending the digital waveguide junction technique to an unstructured triangular topology with interpolation and developing linear connectivity solutions between mesh elements.

With these limitations in mind, it is clear that the retarded potential technique is unsuitable for realtime sound computation with present computers. However, its direct correspondence with three-dimensional variables allows it to predict the response of a linear, homogeneous and adiabatic system much more efficiently than a Navier-Stokes solver. Examples of present day industry use include interior cabin noise modeling for vehicles and evaluating impulse response characteristics of diffusers and performance theatres at different seating locations.

Chapter 2

Background

This chapter presents the mathematical background for the numerical method implemented in the acoustic simulations of this project. Since the goal of the software developed for this work is to remain generally applicable for arbitrary acoustic objects, it is necessary to understand the fundamental mathematical concepts and their physical meanings. Without a proper understanding of the underlying mechanics of the method, and most importantly, the approximations and assumptions made, it will be difficult to adapt and optimize the numerical method properly to a specific simulation.

The software used in this project determines the pressure field within an acoustic body via several repetitive iterations for each unit of time. Therefore, it is an open form solution to an equation describing the pressure within the body as a function of location and time. In our case, this is a partial differential equation that was found by making several simplifying assumptions about the propagation of pressure within the acoustic body. The translation of the continuous mathematical representation of pressure over a duration of time within a volume into a numerical method suitable for iterative calculation on a computer is known as the transient boundary element method.

The transient boundary element method calculates the time domain solution to the inhomogeneous scalar wave equation. At any point in a volume, the solution is determined from the past values along the volume's boundary. In other words, the solution to the inhomogeneous scalar wave equation is an integral function of the retarded potentials across the bounding surface.

The integral solution to the inhomogeneous scalar wave equation originates in electro-

magnetics and is known as the Kirchhoff method of integration. By making simplifying assumptions of the volume containing the point of observation, an appropriate Green's function is applied to the inhomogeneous scalar wave equation to arrive at the Kirchhoff formula. Mitzner [3] developed a numerical approach to the classical Kirchhoff equation known as the retarded potential technique without using matrix inversions. It was used primarily to predict transient scattering of acoustic waves from arbitrary hard surfaces.

The software implementation of this study couples a modified retarded potential technique with a bounding unstructured triangular mesh generated by `Distmesh`. In situations where the mesh is in motion, its movement is governed by two-dimensional digital waveguide meshes propagating velocity waves. This coupling of mesh motion with pressure propagation allows us to simulate struck drum membranes and provides the capability to visualize the motion of the membrane in three dimensions. An overview of the necessary background for understanding our modifications are contained in the following subsections, with interested readers encouraged to further pursue references [2], [3], [8], and [7] as starting points regarding BEM and [4] regarding digital waveguides.

2.1 The Kirchhoff Equation

The following derivation of the Kirchhoff equation can be found in its entirety in Electromagnetic Theory by Julius Adams Stratton [2]. To describe linear acoustics within a homogeneous volume of air contained in a volume, we start from the inhomogeneous scalar wave equation with density source function q

$$\nabla^2\psi(r,t) - \frac{1}{c^2}\frac{\partial^2\psi(r,t)}{\partial t^2} = -q(r,t) \tag{2.1}$$

where ψ is the pressure field, r is the distance to the fixed point of observation, t is time, and c is the speed of sound in the medium. Secondly, we make the assumption of an arbitrary, spherically symmetric solution

$$\phi(r,t) = \frac{1}{r}f\left(t + \frac{r}{c}\right) \tag{2.2}$$

as implied by the homogeneous wave equation

$$\nabla^2\phi(r,t) - \frac{1}{c^2}\frac{\partial^2\phi(r,t)}{\partial t^2} = 0. \tag{2.3}$$

The method for reducing the dimensions of the problem for a twice-differentiable, continuous pressure field ψ at position r_o and time t_o follows by substituting $\nabla^2\psi(r,t) = -q(r,t) + -(1)/(c^2)(\partial^2\psi(r,t))/(\partial t^2)$ and $\nabla^2\phi(r,t) = (1)/(c^2)(\partial^2\phi(r,t))/(\partial t^2)$ into the second Green's identity

$$\int_V (\phi\nabla^2\psi - \psi\nabla^2\phi)dV = \int_S \left(\phi\frac{\partial\psi}{\partial n} - \psi\frac{\partial\phi}{\partial n}\right) dS \tag{2.4}$$

to arrive at

$$-\int_V \phi q dV + \frac{1}{c^2}\int_V (\phi\frac{\partial^2\psi}{\partial t^2} - \psi\frac{\partial^2\phi}{\partial t^2})dv = \int_S (\phi\frac{\partial\psi}{\partial n} - \psi\frac{\partial\phi}{\partial n})dS. \tag{2.5}$$

Given the previous assumption of a spherically symmetric solution, we choose the function

$$f\left(t + \frac{r}{c}\right) = \frac{1}{\sqrt{2\pi}}\frac{e^{\frac{-(t+\frac{r}{c})^2}{2\delta^2}}}{\delta} \tag{2.6}$$

with the special property of vanishing everywhere except at $t = -r/c$. After substitution, this allows all terms to disappear upon integration over the entire duration of time

$$\int\int_S \left(\phi\frac{\partial\psi}{\partial n} - \psi\frac{\partial\phi}{\partial n}\right) dSdt = -\int\int_V \left(f(t+\frac{r}{c}\right)\frac{q}{r}dVdt \tag{2.7}$$

$$+\frac{1}{c^2}\int\int_V \left(\frac{f(t+\frac{r}{c})}{r} - \frac{\psi}{r}\frac{\partial^2}{\partial t^2}f\left(t+\frac{r}{c}\right)\right) dVdt \tag{2.8}$$

$$-\int_V q(x,y,z,-\frac{-r}{c})dV \tag{2.9}$$

The remaining ϕ terms on the left-hand side of the equation are eliminated by bounding the singularity at $r = 0$ caused by $\phi = (1/r)f(t + r/c)$. To accomplish this, we bound the surface S inside by surface S_1 with an inner radius of r_1

$$0 = \int_V \frac{1}{r}q(x,y,z,\frac{-r}{c})dV + \int\int_S (\phi\frac{\partial\psi}{\partial n} - \psi\frac{\partial\phi}{\partial n})dSdt + \int\int_{S_1} (\phi\frac{\partial\psi}{\partial n} - \psi\frac{\partial\phi}{\partial n})dSdt \tag{2.10}$$

As r_1 of S_1 approaches 0, the sampling property of $f\,(t + r/c)$ simplifies the surface integral term upon integration over time

$$\int\int_{S_1}\left(\phi\frac{\partial\psi}{\partial n}-\psi\frac{\partial\phi}{\partial n}\right)dSdt=-4\pi\psi(x',y',z',0). \tag{2.11}$$

Finally, we isolate for the pressure field ψ and substitute $\phi=(1/r)f(t+r/c)$ into the remaining time integral and arrive at the Kirchhoff equation at time $t=0$

$$\psi(x',y',z',0)=\frac{1}{4\pi}\int_V\frac{1}{r}q\left(x,y,z,\frac{-r}{c}\right)dV+\frac{1}{4\pi}\int_S\left(\frac{1}{r}\frac{\partial\psi}{\partial n}-\frac{\partial}{\partial n}\frac{1}{r}\psi+\frac{1}{cr}\frac{\partial r}{\partial n}\frac{\partial\psi}{\partial t}\right)_{t=\frac{-r}{c}}dS. \tag{2.12}$$

Recognizing that the right-hand side of the above equation consists entirely of terms at $t=-r/c$, the equation can be rewritten for an arbitrary time t'

$$\psi(x',y',z',t')=\frac{1}{4\pi}\int_V\frac{1}{r}q\left(x,y,z,t'-\frac{r}{c}\right)dV+\frac{1}{4\pi}\int_S\left(\frac{1}{r}\frac{\partial\psi}{\partial n}-\frac{\partial}{\partial n}\frac{1}{r}\psi+\frac{1}{cr}\frac{\partial r}{\partial n}\frac{\partial\psi}{\partial t}\right)_{t=t'-\frac{r}{c}}dS \tag{2.13}$$

The first term on the right-hand side represents the source excitation, while the second term on the right-hand side contributes to ψ at the observation position and time as a function of past values of ψ along the bounding surface. Therefore, we are able to obtain our solution by making simplifying assumptions of the inner volume and ensuring the wave equation is satisfied at the boundaries. This is known as a direct boundary element method, since ψ can only be determined in either the inner or outer volume of a closed surface S, and cannot be known in both volumes simultaneously.

2.2 The Retarded Potential Technique

Mitzner [3] derives the discrete approximation to the classical Kirchhoff equation by dividing the bounding surface S into K elements and approximating the time derivative term with a three-point backward difference formula [9]. To obtain the collocation scheme for a transient scattered field, we first simplify the source excitation to ψ_o and combine the first two terms inside the surface integral from the quotient rule

$$\frac{1}{r}\frac{\partial\psi}{\partial n}-\frac{\partial}{\partial n}\frac{1}{r}\psi=\left(\frac{\partial\psi}{\partial n}\frac{\partial r}{\partial n}\right)\left(\frac{1}{r^2}\right) \tag{2.14}$$

and factor to obtain the expression for the Kirchhoff equation in K discrete zones

$$\psi = \psi_o + \frac{1}{4\pi}\sum_K \int_{S_K} \left(\frac{1}{r^2}\frac{\partial r}{\partial n}\left(\frac{\partial \psi}{\partial n} + \frac{r}{c}\frac{\partial \psi}{\partial t}\right)\right) dS. \tag{2.15}$$

When the point of observation is directly on the surface S, the surface area of the infinitesimal sphere bounding this point of observation is halved. This is the result of the sphere being split into halves by the surface upon which the point of observation resides, and therefore restricting the point to observe the half of the sphere that is above the surface only. Thus, the influence of the retarded potentials in Equation 2.15 is doubled as shown in Equation 2.16:

$$\frac{1}{2}\psi = \psi_o + \frac{1}{4\pi}\sum_K \int_{S_K} \left(\frac{1}{r^2}\frac{\partial r}{\partial n}\left(\frac{\partial \psi}{\partial n} + \frac{r}{c}\frac{\partial \psi}{\partial t}\right)\right) dS. \tag{2.16}$$

As an example, a collocation point at the inner corner of a perfect box would require a factor of 1/8 by the same argument.

Mitzner's retarded potential approach is based on subdividing the S_K elements until they are small enough such that the second factor in the integrand can be approximated by

$$\frac{\partial \psi_K}{\partial n} + \frac{r_K}{c}\frac{\partial \psi_K}{\partial t} \approx \psi_K\left(t - \frac{r_K}{c}\right) + t_K \frac{\partial \psi_K\left(t - \frac{r}{c}\right)}{\partial t} \tag{2.17}$$

and is therefore pulled out of the integrand.

The remaining integral $\int_{S_K}(1/{r_K}^2)(\partial r_K/\partial n)dS$ is the negative of the solid angle Ω subtended by S_K as observed at a distance r_K. As a result, the scattered field from collocation point j located on a smooth, continuous surface is

$$\frac{1}{2}\psi_j - \psi_{oj} \approx -\frac{1}{4\pi}\Omega\left(\psi_K\left(t - \frac{r_K}{c}\right) + t_K\frac{\partial \psi_K\left(t - \frac{r}{c}\right)}{\partial t}\right). \tag{2.18}$$

For both ψ_K terms on the right-hand side of the equation, we can linearly interpolate between time steps for τ a non-integer multiple of r/c

$$\psi_K\left(t_o + m\tau - \frac{r_K}{c}\right) = (1-\gamma)\psi_K(m-n) + \gamma\psi_K(m-n-1), \tag{2.19}$$

$$\frac{\partial \psi_K \left(t_o + m\tau - \frac{r_K}{c}\right)}{\partial t} = (1-\gamma)\frac{\partial \psi_K(m-n)}{\partial t} + \frac{\partial \gamma \psi_K(m-n-1)}{\partial t}. \tag{2.20}$$

Furthermore, the differentiation in time term $\frac{\partial \psi_K}{\partial t}$ can be approximated with the previously mentioned three-point backward difference formula $C_0 = 3/2, C_1 = 2, C_2 = 1/2$ as shown below

$$\frac{\partial \psi_K}{\partial t} \approx \frac{1}{\tau}\left(\frac{3}{2}\psi_K(m-0) - 2\psi_K(m-1) + \frac{1}{2}\psi_K(m-2)\right). \tag{2.21}$$

By substituting equations 2.19, 2.20, and 2.21 into equation 2.18, we can generalize the equation for the scattered value from surface element j at time m with influence coefficient β_{ij}, which accounts for the influence of the k-th element on element j at a time delay of i units as shown below

$$2\pi\psi_j^m = 4\pi\psi_{oj}^m - \sum_{k=1}^{K}\sum_{i=1}^{I}\beta_{ij}^k \psi_k^{(m-i)}. \tag{2.22}$$

2.3 Waveguide Junction Meshes

Mesh motion for membrane modeling is based on velocity wave propagation in N-junction loaded digital waveguides as described in [10]. This provides our software the capability to simulate dynamic drums that have been struck anywhere on the surface, and also the ability to visualize the motion in three dimensions. The number of digital waveguide elements, N, intersecting at each junction in an unstructured mesh varies, so the number of neighboring nodes must be pre-computed. The scattering relations for the junction velocity at junction j are shown below:

$$V_j = V_i^+ + V_i^- \tag{2.23}$$

$$V_j = 2\left(R_j + \sum_{i=1}^{N} R_i\right)^{-1} \sum_{i=1}^{N} R_i V_i^+ \tag{2.24}$$

where V_i^+ is the incoming velocity wave, V_i^- is the outgoing velocity wave, R_i is the wave impedance of the medium and R_j is the load impedance of the i-th digital waveguide on the

j-th junction [4]. In order to provide control over the tuning of the membrane as the mesh granularity changes at different sampling rates, the motion of the junctions can be tuned by an interpolation ratio. From the velocity of the mesh junctions, the normal pressure gradients for each surface element are established and pressure variation is injected into the retarded potential system as a Neumann boundary condition. In a similar manner, the retarded potential system is coupled to the digital waveguide junction through the total junction force

$$\sum_{i=1}^{N} F_i(s) = F_j(s) = V_j(s)R_j(s). \tag{2.25}$$

Each collocation point along the boundary is associated with an area A determined by its Gauss-Legendre weight and the element area, thus providing the feedback force $F_{fdbk} = \psi A$ that can be scaled appropriately. It should also be noted that when using unstructured meshes, a uniform distance option in `Distmesh` is used to force the mesh generation algorithm to maintain a relatively uniform distribution of junctions throughout the mesh.

2.4 Related Work

The theoretical foundation for this study as briefly described in the first two sections of this chapter originates from electromagnetics, as detailed by Stratton [2]. The numerical method based on the classical formulation of the Kirchhoff equation in the context of scattered acoustic waves was first demonstrated by Mitzner [3]. Friedman [8] and Shaw [11] similarly demonstrated scattered acoustic waves from arbitrary objects, but with additional residual terms to account for discontinuous wavefronts.

More recently, Kawai and Terai [6] formulated an indirect transient boundary element method suitable for scattering from thin plates. The frequency-domain version of the boundary element method was used to model a resonating guitar body, achieving good correlation with measured results [7]. In the same work, a FEM model of a timpani drum was coupled with a BEM model of the surrounding air and was shown to achieve good correlation with measured results. Additional work in realtime applications has seen FEM and BEM being used to pre-compute filters for use in animated shorts [12].

Chapter 3

Design and Implementation

This chapter describes the `MATLAB` toolbox developed for this study. In general, the toolbox implements Mitzner's retarded potential technique with added modifications and improvements in a number of areas, including direction dependence, object obstruction, solid angle calculation, element integration, and object representation via meshing. The meshing algorithm is based on the GPL toolbox `Distmesh` [1], also with added functionality. Furthermore, it will be made clear that the majority of this work was conducted to support a generalized framework, so that it could be easily understood and adopted for a single, optimized application. It should be kept in mind, though, that the direct boundary element method, in addition to its simplifying homogeneous assumptions of the problem space, requires the user to define the bounding surface, and thus assumptions must be made concerning the scattering coefficients of the surface along with its other characteristics . In general, perforated surfaces should be treated with indirect BEM solvers such as those detailed in [6] and [7]. These works, among other relatively recent advances, are discussed in the future work section in Chapter 5.

3.1 Mesh Generation

`Distmesh` was developed by Per-Olof Persson and Gilbert Strang from the Department of Mathematics at MIT. It produces unstructured triangular meshes in two-dimensions and unstructured tetrahedral meshes in three-dimensions, but is also capable of generating general N-simplex meshes in N-space, where an N-simplex is the N-dimensional analogue of a triangle. The unstructured designation means that the simplex elements are formed

without strict topological limitations such as element size or number of neighbours. In this sense, it presents some unique difficulties when used as the basis for a waveguide membrane, as discussed later in this chapter. However, `Distmesh` provides the option to force the algorithm to make efforts to preserve uniform distance between nodes. This uniform option was used for all `Distmesh` related discussion in this work.

The geometry of each mesh is based on a user-defined signed distance function, which is negative inside the mesh region and positive outside. By intersection, differencing, and union of distance functions, complex geometries can be created. It is necessary that the user of our boundary element software first include the `Distmesh` folder, which is included with its distribution, into `MATLAB`'s internal search path variable. Alternatively, the `Distmesh` software can be downloaded separately from the web address for `Distmesh` [13]. An example of a tetrahedral mesh used to model a sphere in three dimensions is shown in Figure 3.1, where the distance function for this case would simply be $\sqrt{(x^2 - r) + (y^2 - r) + (z^2 - r)}$, with x, y, and z being the Cartesian mesh node coordinates and r being the radius of the sphere.

3.1.1 Generating Surface Meshes

`Distmesh` uses a tetrahedral topology throughout the volume for three-dimensional meshes. Using the `Distmesh` function `distmeshnd`, we generate two vectors p and t, which are the Cartesian mesh node coordinates and adjacency matrix, respectively. The adjacency matrix is a list of pointers into p in groupings of $N + 1$ for an N-dimensional mesh, where each grouping represents the vertices of the simplex elements. For a tetrahedral mesh, the adjacency matrix t would consist of pointers in groups of four, with each group representing a unique tetrahedron in three-dimensional Euclidean space.

It is necessary to remove the internal nodes to generate a surface mesh for the boundary element method since it requires only the bounding surface of the modelled object. To accomplish this, an algorithm was developed to count the occurrences of the four triangular faces of each tetrahedral element of the mesh within the adjacency matrix. Internal triangular faces are shared between adjacent tetrahedrons, and therefore occur twice. Conversely, the triangular elements facing outwards are associated with only one tetrahedron. These triangular elements are preserved and all others purged. After removing the inner node coordinates from p, the adjacency matrix t is re-aligned to the new coordinate indices in

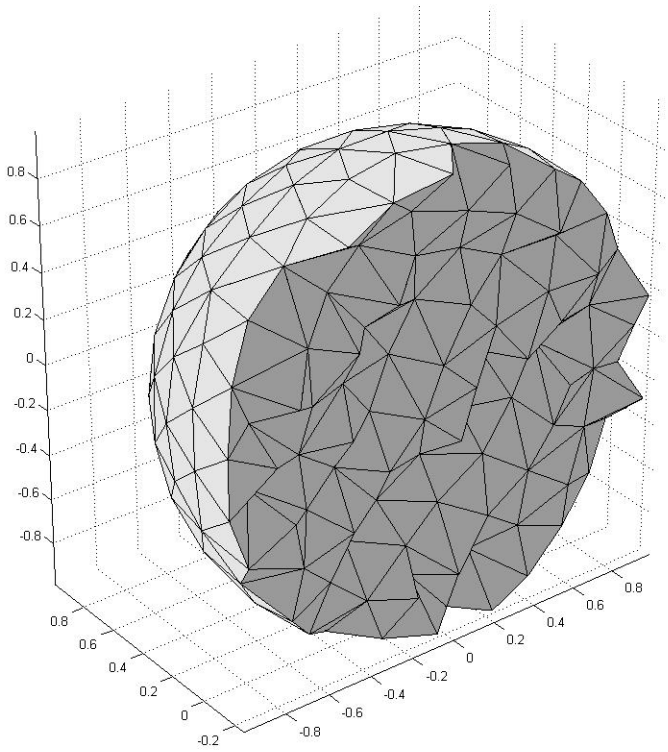

Fig. 3.1 Half of a sphere generated with `Distmesh` in 3D using tetrahedral mesh elements

p and reduces to a list of 3-element, triangular groupings. This functionality is performed in the function `vol2surf`. Finally, a new graphing function `simpplotsurf` based on the `Distmesh` function `simpplot` was developed in order to display the surface. An example of `vol2surf` being applied to the tetrahedral mesh in Figure 3.1 is shown in Figure 3.2.

3.1.2 Orienting Surface Normals

In order to make several key surface calculations such as vector reflection and Gaussian element integration, the normals of each triangular element must be computed and oriented inwards as shown in Figure 3.3. The unstructured topology does not orient the node indices in the adjacency matrix in a fixed manner to easily permit orientation, and it was important

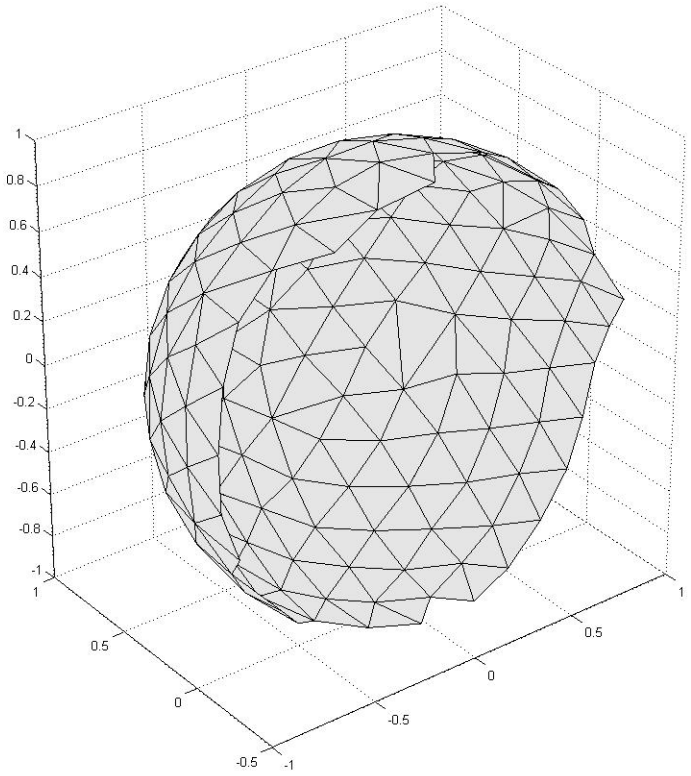

Fig. 3.2 Half of a sphere surface generated with `Distmesh` in 3D using tetrahedral mesh elements, after being processed by `vol2surf`

to maintain generality and to allow for complex geometries, including meshes with folds. As a result, Jordan's curve theorem is applied before any perforation of the mesh by associating an odd number of surface intersections with the infinitely extended inward normal [14]. This follows from the theorem, which states that any continuous simple closed curve in the plane separates the plane into two disjoint regions, the inside and the outside.

The number of intersections was determined by extending the normal infinitely from the center of the triangle and applying an intersection algorithm [15] with all other triangular elements contained in the mesh. For this purpose, the distance t for the normal $N(t) = O + tD$ and the barycentric coordinates (u, v) of its possible intersection with triangle T

with vertices V_1, V_2, and V_3 are determined below:

$$N(t) = T(u, v) \tag{3.1}$$
$$= (1 - u - v)V_0 + uV_1 + vV_2. \tag{3.2}$$

By applying Cramer's rule to 3.2, we can determine t, u, and v according to Equation 3.5:

$$E_1 = V_1 - V_0 \tag{3.3}$$

$$E_2 = V_2 - V_0 \tag{3.4}$$

$$\begin{pmatrix} t \\ u \\ v \end{pmatrix} = \frac{1}{(D \times E_2) \cdot E_1} \begin{pmatrix} ((O - V_0) \times E_1) \cdot E_2 \\ (D \times E_2) \cdot (O - V_0) \\ ((O - V_0) \times E_1) \cdot D \end{pmatrix}. \tag{3.5}$$

This functionality is implemented in the function `intersecttriangle`, which includes handlers for several degenerate cases. This includes handling intersections with vertices and shared edges. The encapsulating function for determining surface normals is contained in the function `normmeshnd`.

3.1.3 Characterizing Surfaces

For simulations involving objects such as unflanged open-end pipes and hand drums, triangular elements of the surface mesh need to be removed at the location of the desired perforations. The technique to accomplish this with a triangular surface mesh is similar to the tetrahedral-to-surface mesh conversion method, where the nodes belonging to the perforation are removed from p, with t modified accordingly afterwards. The nodes to be removed are specified by a user-defined signed distance function. It should be noted that nodes designated as removed in the perforation operation by function `vol2surf` are retained in our implementation of the retarded potential technique and are set to invert incoming pressure waves. This sets the boundary condition at the opening as pressure $p = 0$, which is a low-frequency approximation.

In general, all surface properties are localized to the elements of interest by implementing user-defined signed distance functions. This includes mechanical impedance for waveguide

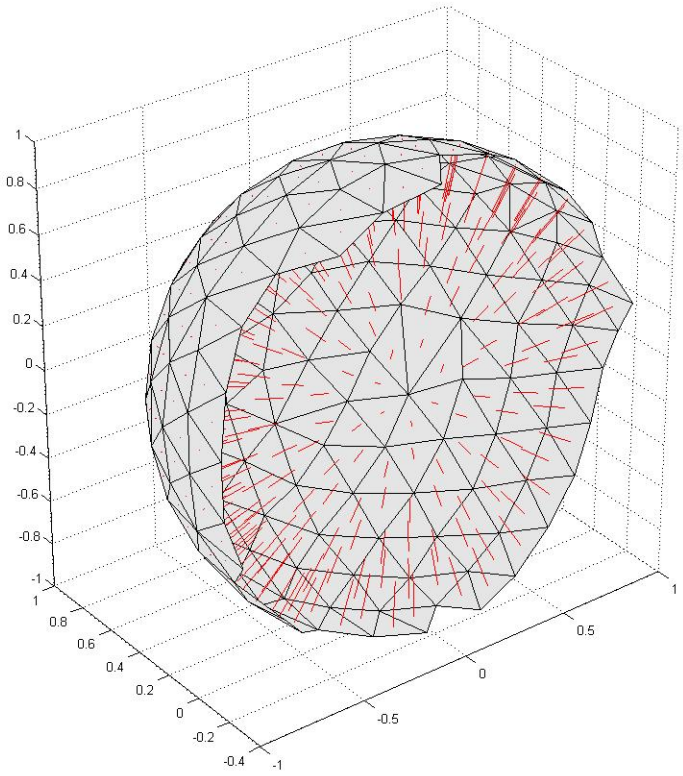

Fig. 3.3 Half of a sphere surface generated with `Distmesh` in 3D using tetrahedral mesh elements and displaying inward oriented surface normals, after being processed by `vol2surf` and `normmeshnd`

simulation, scattering coefficients for the vectorized retarded potential scheme, and the placement of initializing pressure origins, such as plane wave excitation in a closed-open cylinder experiment. These surface characteristics are described in more detail in section 3.2.

3.2 The Vectorized Retarded Potential Technique

In this section, the differences between Mitzner's retarded potential technique and our own implementation are outlined.

3.2.1 Element Integration

Whereas Mitzner zones the surface elements by using co-ordinate curves, we elect to integrate across the elements using four-point Gauss-Legendre cubature [16]. This allows us to express a definite integral as a weighted sum of four function values within each triangular element

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{4} w_i f(x_i). \tag{3.6}$$

In our case, the Gaussian weights are used to determine solid angle values which in turn are a key component of the influence coefficient, as discussed in section 2.2.

3.2.2 Solid Angle Calculation

With weighted Gaussian points for each triangular surface element, we can divide the exact solid angle calculation for the entire triangle among the Gaussian points when computing the influence coefficients. This is accomplished by using the Gaussian weights to divide the solid angle determined by the method described in [17], which is an exact solid angle calculation for planar triangles. In contrast, Mitzner approximates the solid angle used in equation 2.22. The exact formula used to determine the solid angle in the vectorized retarded potential technique is given by:

$$\Omega = 2\arctan\left(\frac{det\begin{pmatrix} a \\ b \\ c \end{pmatrix}}{abc + (a\cdot b)c + (a\cdot c)b + (b\cdot c)a}\right), \tag{3.7}$$

where a, b, and c are vectors from the observation point to the vertices of the triangle.

3.2.3 Connectivity

The boundary element method assumes the volume to be homogeneous and therefore does not model diffraction. When computing the influence coefficients β in equation 2.22, the

previously mentioned `intersecttriangle` routine is used to determine line-of-sight connectivity between surface elements. This is especially important with complex geometries, where there would be no direct interaction between elements at either end of a complex object such as a twisted tube. Note that connectivity, along with the other components used to compute the influence coefficients, should be re-computed each time step for situations where the bounding surface or objects within would be in motion.

3.2.4 Vector Fields

Along with connectivity, Mitzner's formulation of the retarded potential technique makes no consideration for the origin of the presure wave or the orientation of the surface. Therefore, the original implementation would scatter a surface value to all other surface elements S_K regardless of their location or the angle of incidence of which the exciting pressure made contact with the surface. To resolve this problem, a scattering coefficient is used to determine the maximum deviation angle from which a reflected vector can intersect with another surface element. The reflected vector is calculated by equating the angle of incidence between the incident vector and the inward-oriented normal of S_K with the reflected angle.

The collocation scheme for a vector field is much more memory intensive and requires a number of unique vectors to be stored at each surface element S_K. For a mesh consisting of N elements, there can be vectors originating from up to $N-1$ other elements in the mesh which are required to be stored at each time step. The space for these vectors are dynamically allocated during the simulation. Over several time steps, the number of unique vectors stored in each element will increase dramatically since each element will transmit vectors of different magnitudes to several other elements in a one-to-many fashion. To avoid memory leak issues, the list of vectors for each surface element S_K are optimized such that vectors pointing in the same direction are merged. Therefore, for a mesh consisting of N triangular elements, each element may be required to store as many as $N-1$ unique vectors for each time step.

3.2.5 Observation Mesh

With Mitzner's formulation, the observation point is a single point in space

$$2\pi\psi_j^m = 4\pi\psi_{oj}^m - \sum_{k=1}^{K}\sum_{i=1}^{I}\beta_{ij}^k\psi_k^{(m-i)}. \tag{3.8}$$

However, the main observation variable, pressure $p = F/A$, is associated with surface areas, where F is force and A is area.

As a consequence, we choose to take an alternate approach based on listening meshes such as the one shown in Figure 3.4. This allows for results that are averaged over a larger, more relevant surface. A separate function `makept` uses the two-dimensional `Distmesh` function `distmesh2d` to create a circular observation mesh at a user-specified location and orientation.

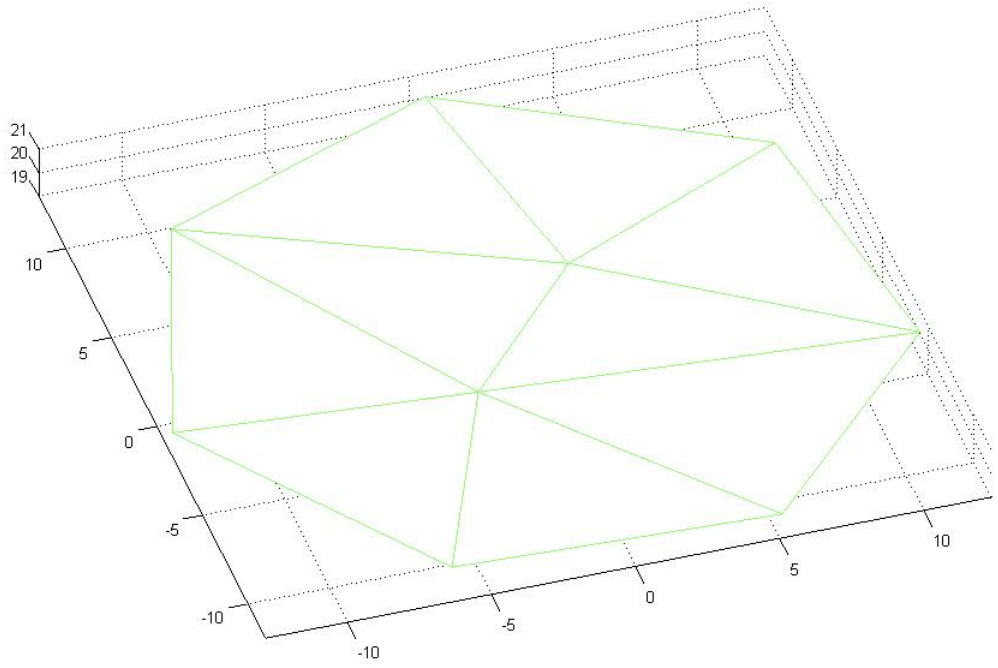

Fig. 3.4 Listening mesh oriented with normal vector (x,y,z) = [0 0 1] and elevated to 20 mm height

The main simulation implementing the vectorized retarded potential technique takes place in the function `tbem`. The listening mesh is transparent and does not propagate or reflect pressure waves.

3.2.6 Waveguide Junctions With Distmesh

Mesh motion for membrane modeling is based on velocity wave propagation in N-junction loaded waveguides and coupled with the internal pressure field generated by the vectorized retarded potential technique, as previously described in Section 2.3. We present several figures showing the time evolution of a center-struck circular drum membrane using the waveguide junction technique as shown in Figure 3.5.

3.3 Software Description

Appendix A provides a brief list and description of the software modules developed as part of the `MATLAB` toolbox for this study. For further details such as input and output argument requirements, type `help [functionname]` in the command line of `MATLAB` after navigating to the directory containing the source code.

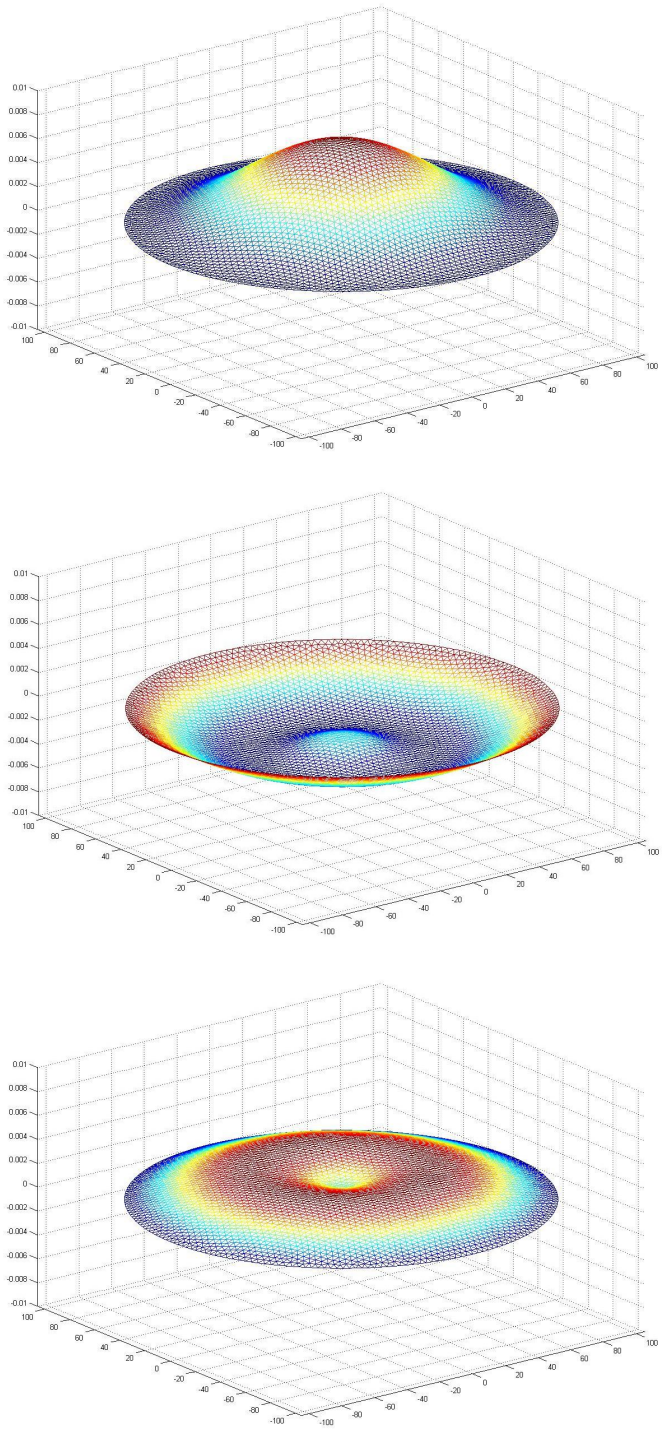

Fig. 3.5 Drum membrane modeled with a waveguide junction mesh adapted to an unstructured triangular topology, from top to bottom: timestep n=1, 30, and 60

Chapter 4

Results and Discussion

This chapter presents the results obtained for this study and discusses in detail various technical issues of particular importance. Specifically, the experiments performed with the vectorized retarded potential technique are:

- The simulation of an unflanged pipe with a closed and open end. An initial plane wave propagates from the closed end towards the open end. This demonstrates the accuracy of the method chosen.
- The simulation of a djembe drum in response to centered strike on its membrane. This demonstrates coupled simulation with mesh motion objects similar to the wire frame models used in animation.

It was found that the TBEM simulation of the unflanged pipe was in general agreement with the lossy waveguide model. No actual losses were implemented in the vectorized retarded potential simulation though some losses are inherent in the numerical technique. Furthermore, it is believed that the transfer of energy from the 1D propagating plane wave to higher-order radial modes contributed to the low-pass filtered effect.

4.1 Unflanged Cylindrical Pipe Simulation

A cylindrical pipe of 80 mm length and 18 mm radius was simulated with an observation point at its center. The input end was rigidly terminated (an infinite impedance) and an ideal open-end boundary condition was applied at the other end, as shown in Figure 4.1. A

circular observation mesh was placed in the middle of the pipe. A plane wave was initialized at the closed end. An ideal, open-end boundary condition was used (zero pressure) and the cylindrical walls were assumed to be infinitely rigid.

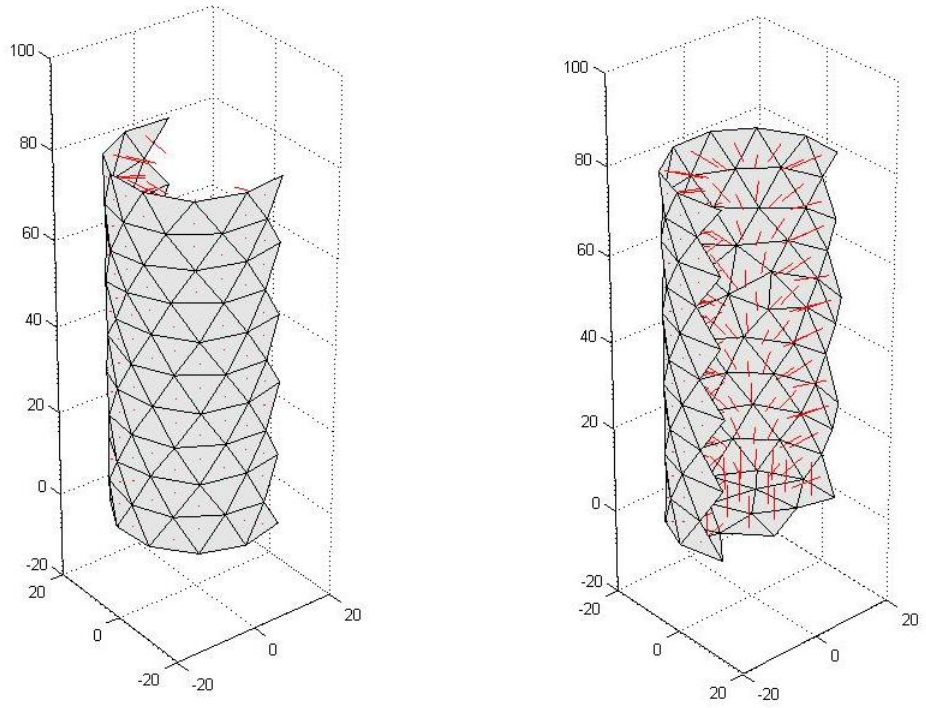

Fig. 4.1 Front and back perspectives of half of an open-closed unflanged pipe with length of 80 mm and radius of 18 mm, created with `Distmesh`, inner view includes inward oriented surface normals

The time response and frequency response at the mid-point observation circular mesh is shown in Figures 4.2 and 4.3 and compared with a 1D waveguide model as described in [18]. The results are in general agreement although there are obvious discrepancies between the responses and the number of dimensions modeled. While both models implement an ideal zero-pressure boundary condition at the open end, an indirect BEM method should be implemented for a more rigorous model of thin boundaries and perforated surfaces. Also note that the 1D waveguide response includes filters to model attenuation and phase delay due to viscous and thermal losses along the sidewalls that are not accounted for in the 3D simulation [18]. While the TBEM results are not intended to exhibit these losses, it

is inherently lossy due to leakage resulting from the implementation of a minimal vector magnitude constraint in the `intersecttriange` routine. For extremely small vectors, it was found that precision errors would occur in the routine and cause false intersections or result in no intersections. Since the algorithm is iterative, these losses are cumulative and are non-negligible. There is a tuneable constant `EPSILON` in the routine `tbem` which can be decreased to approach ideal lossless computation, but as this becomes extremely small it significantly slows the simulation and increases the memory requirements due to the large number of vectors which would not be truncated. It is also believed that the higher-mode interference causes a smoothing effect as the pulses propagate longitudinally, radially, and so forth. In effect, this is a lowpass filter since high frequency energy is transferred to other modes. A more thorough investigation of this effect is suggested in section 5.2 of this work.

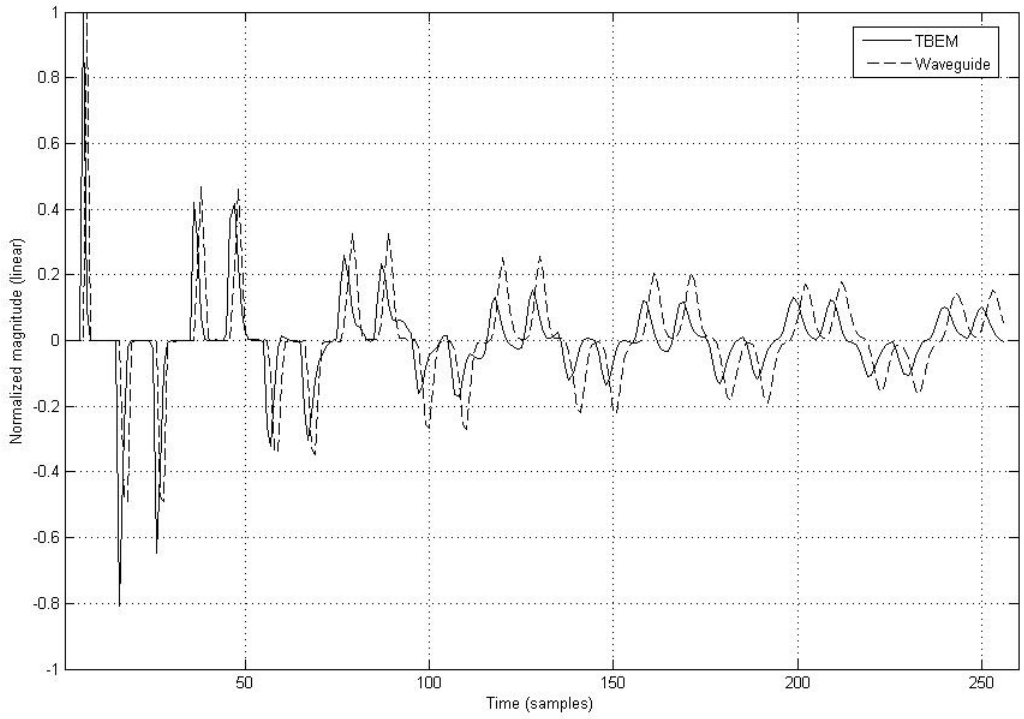

Fig. 4.2 Normalized pressure versus time taken at the midpoint of the simulation of an open-closed unflanged pipe with length 80 mm and radius 18 mm. TBEM results (solid) and waveguide results (dashed)

Another approximation present in the TBEM routine is attributed to Gaussian integra-

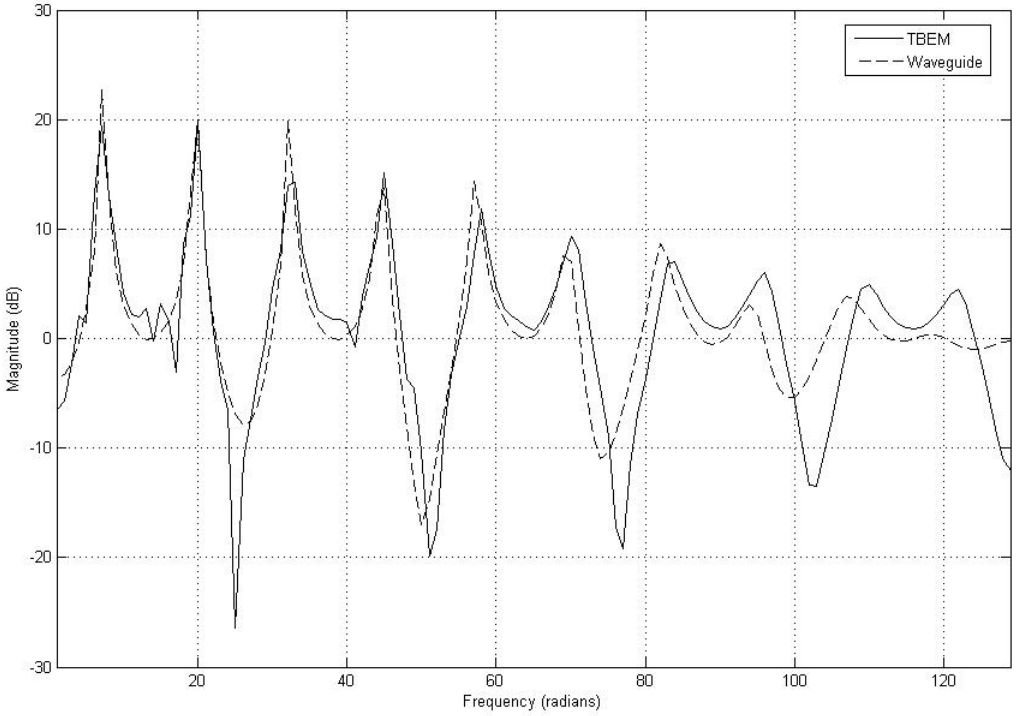

Fig. 4.3 Magnitude (dB) of pressure in the frequency domain taken at the midpoint of the simulation of an open-closed unflanged pipe with length 80 mm and radius 18 mm. TBEM results (solid) and waveguide results (dashed)

tion and its relation to the spatial sampling distance. For boundary elements to communicate in time, their distance must remain greater than c/f_s, where c is the speed of sound and f_s is the sampling frequency. When this is violated between weighted Gaussian points, the distance was rounded to the nearest valid value. This partially accounts for the slight timing differences in the time simulation (Figure 4.2) and the slight high frequency peak differences in the frequency domain results (Figure 4.3). Finally, differences between the waveguide and TBEM simulations can also be attributed to the fact that a pipe generated with `Distmesh` will exhibit geometric non-idealities and affect the results. These geometric imperfections are due to the fact that the mesh generated with `Distmesh` only approximates the actual shape of a cylinder with triangular elements.

4.2 Djembe Drum

A djembe drum of height 400 mm with a membrane diameter of 210 mm was modeled in a fully coupled simulation. The mesh models are shown in Figures 4.5 and 4.6. The initial excitation was modeled as a raised cosine distribution of velocity at the membrane waveguide junctions. Once again, a rigid boundary condition was set along the surface of the drum with an ideal open-end boundary condition at the bottom hole. The first 14 ms of the simulation results are shown in Figure 4.4. This simulation demonstrates the applicability of BEM-based methods for physically-based sound generation in animation games.

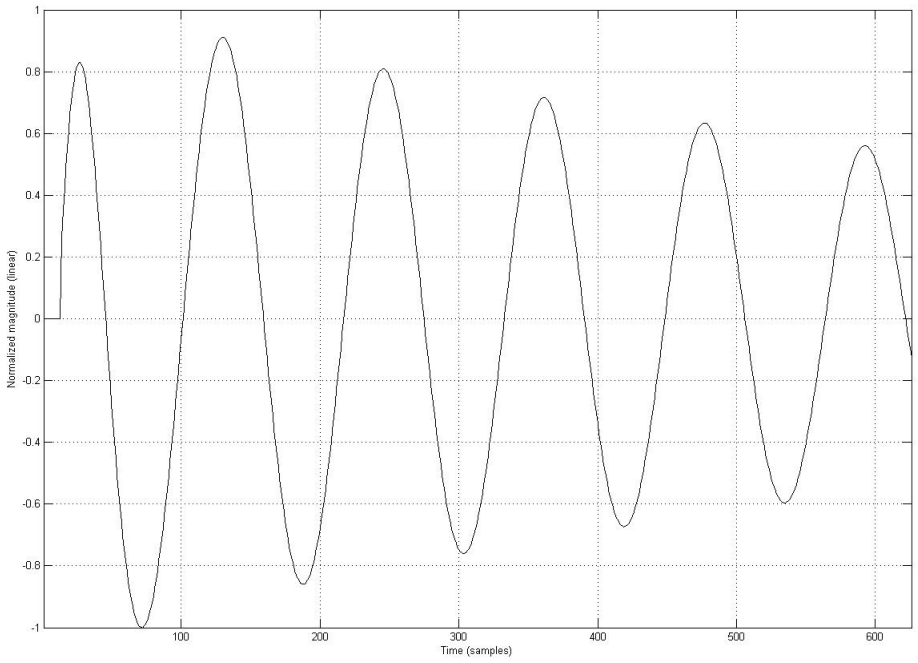

Fig. 4.4 Initial 14 ms of the pressure within a fully-coupled djembe drum simulation after a centered strike of the membrane, taken 10 mm above the bottom exit hole center

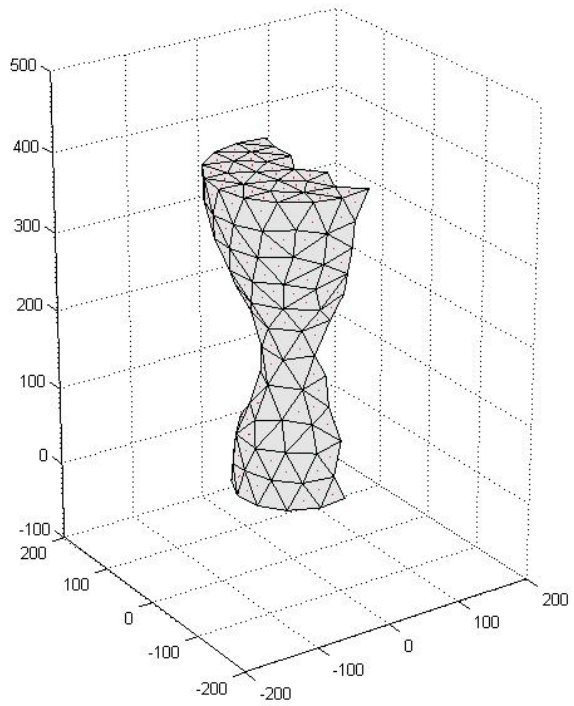

Fig. 4.5 Half of djembe drum mesh model created with `Distmesh`, outer view

4.3 Discussion

One of the drawbacks with implementing Gaussian quadrature on a pointwise basis is that the distance between Gaussian points must remain greater than c/F_s, where c is the speed of sound and f_s is the sampling rate. In the case that the exception to this rule is not handled, the vectors belonging to a source element would partially be written into the delay structure of the `tbem` routine at the current time-step. In a physical sense, this corresponds to a generation of energy and leads to unstable simulations.

One work-around for this situation is to locate troublesome Gaussian point locations and manually place or integrate them as exceptions. Another option is to completely disregard the propagation between the violating elements, but this would cause energy leaks resulting in an excessive damping of the results when simulating a lossless system. Since these occurrences were generally rare and took place primarily in sharp corners, we

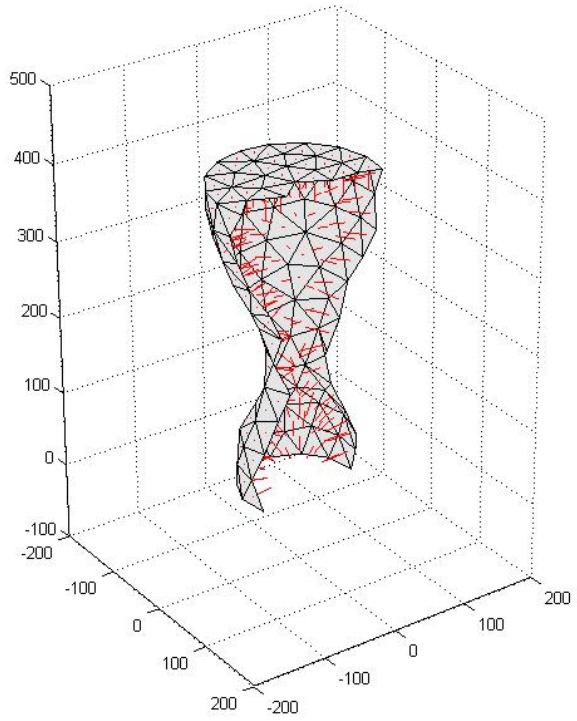

Fig. 4.6 Half of djembe drum mesh model created with `Distmesh`, inner view includes inward oriented surface normals

elected to take the simple solution of rounding the spatial sampling distance to unity in order to avoid instability. However, this inaccuracy in potential retardation can cause slight discrepancies in observed resonant modes upon discrete Fourier transform analysis in comparison with waveguide models.

The simulation stores a list of unique vectors at each Gaussian point. It is extremely important to sort and merge this vector list into a unique list of vectors since their growth can outpace the rate of the time-step zeroing of past elements and lead to memory leaks. As discussed earlier, the memory leak can occur when the vectors transmitted between mesh elements at each time step are dynamically allocated and are not merged according to the orientation of the vectors.

Chapter 5

Conclusions and Future Work

This chapter summarizes the work presented and makes suggestions for future work. Overall, the simulations were successful but will require much more optimization in terms of computational complexity to be feasible in the immediate future.

5.1 Summary and Conclusions

A variation of the retarded potential technique was presented and merged with the meshing capability of `Distmesh` and demonstrated to be a generalizable framework for a number of simulations. It is extensible to suit applications for prototyping, measurement, instrument synthesis and immersive sound where inhomogeneous qualities of the volume can be neglected. Therefore, it should only be utilized where accurate modeling of diffraction is not necessary.

Several improvements to the retarded potential method proposed by Mitzner [3] were implemented, including an improvement to the solid angle calculation and a vectorization of the propagation variables to support direction dependence. This allows one to solve for integral solutions of scalar fields such as pressure. Furthermore, the generality of the mesh model was kept at the forefront as line-of-sight connectivity is determined during each time step in support of the influence coefficient calculations.

An cylinder experiment was conducted and compared with the one-dimensional waveguide model in order to verify the accuracy of the vectorized retarded potential technique. Results show show promise for further development of the technique for pipe simulations.

In order to demonstrate the generality and applicability of the vectorized retarded poten-

tial technique in animated multimedia applications, a fully coupled simulation of a djembe drum strike was studied. While too computationally intensive for realtime applications, it is hoped that the GNU GPL availability and inevitable increase in computational capacity will further the interest in transient BEM in musical applications.

5.2 Future Work

The Green's function, as shown in Equation 2.4, employed to derive the Kirchhoff integral solution to the inhomogeneous wave equation, constrains the pressure to be continuous and twice differentiable in the volume. Other formulations of the transient BEM add additional residual terms to account for discontinuous wavefronts and should be considered for more rigorous predictive applications [8], [11]. A discontinuous wavefront occurs when the limit of the function describing the wavefront does not satisfy the Cauchy definition of continuity. A step function is an example of a discontinuous function.

Also, the retarded potential technique is a direct BEM method, and thus it is not strictly correct to simulate perforated bodies or slender membranes with it. A more rigorous approach is to apply the indirect boundary element method that integrates the pressure jump across the boundary [7], [6].

Additional refinements can also be made in the numerical method of element integration. While Gauss-Legendre cubature is effective, error increases as the distance between the observation point and the boundary element decreases due to the (1/r) weak singularity expression in the kernel of the integral. Other methods have been explored in the context of transient acoustic scattering to account for these problems and would be useful as an addition [6]. It should be noted that in the current implementation of the vectorized retarded potential technique, distances less than the spatial sampling distance $\frac{c}{F_s}$ (where c is the speed of sound and f_s is the sampling rate) would lead to instabilities. Therefore, these distances are rounded up to a unity spatial sampling distance.

It is also possible to simulate loss at surface perforations and to incorporate losses at the sidewalls of an object in order to obtain a more realistic simulation of an acoustic body. The primarily difficulty with this approach is that such losses are not directly inherent in the underlying mathematical model of the vectorized retarded potential technique, and therefore would require tuned constants or an augmentation to the fundamental equations.

Finally, it should be noted that for virtual reality simulations where the listening or

collocation points would move, the influence coefficients of the retarded potential technique would have to be dynamically updated for each time step.

Appendix A

Software Modules

- *animate*
  Displays motion of mesh each time step.
- *anglebetween*
  Calculates the angle between two vectors in Cartesian space.
- *boundedges3d*
  Determines the boundary edges of a surface mesh.
- *dcube*
  Distance function for a cube.
- *demotbem*
  Demonstration module containing the setup calls to the make* functions to create various meshes, listening points, etc.
- *distmesndlim*
  A version of `distmeshnd` that will force convergence after a certain number of iterations. This is useful when using functions such as `dcube` or implementing very complicated distance functions.
- *findnbours*
  Finds the neighbours of a node in a surface mesh. Used to determine the adjacency matrix for velocity wave propagation in waveguide junctions.

- *findreflection*
  Determines the reflection vector from a specified surface element under the assumption that the angle of incidence always equals the angle of reflectance.

- *findrim*
  Locates the boundary nodes and edges of a membrane as specified by the mechanical impedance setting in the function `makemesh`.

- *findscatter*
  Determines connectivity based on current stored vector in the source element based on scatter coefficient and `intersecttriangle` function.

- *findscatter_listen*
  Similar to `findscatter`, but for the listening mesh.

- *fixmeshsurf*
  The same as `fixmesh` as developed in the `Distmesh` toolbox, but adapted for surface meshes.

- *gaussquad*
  Calculates the gaussian points for a triangular element after rotating to the XY plane using Rx, Ry, and Rz functions.

- *getgauss*
  Wrapper function for `gaussquad`.

- *graphnorm*
  Displays a surface mesh with calculated surface normals as red lines.

- *impmeshnd*
  Uses a distance function to set the mechanical impedance of a section of a surface mesh.

- *intersecttriangle*
  Determines line-of-sight connectivity between mesh elements.

- *isconnect*
  Wrapper function for `intersecttriangle` and performs some checking for degenerate intersection cases.

- *makeadj*
  Finds adjacency matrix for velocity wave propagation in waveguide junction implementation of vibrating membranes.

- *makeg*
  Creates the global structure variable.

- *makemesh*
  Creates the surface mesh and all related structure fields for use in the `tbem` routine.

- *makept*
  Creates the listening mesh and related structure fields for the simulation.

- *mergevectors*
  Determines vectors in a list that are oriented in the same direction and merges them together in order to minimize memory requirements during the `tbem` routine.

- *meshmotion*
  Adds motion vectors to the nodal positions of a membrane in motion for display.

- *normmeshnd*
  Locates and calculates the normals of a given triangular surface mesh.

- *projectvector*
  Calculates the projection distance of a vector in a particular direction.

- *Rx*
  Rotation of an image about the x-axis

- *Ry*
  Rotation of an image about the y-axis

- *Rz*
  Rotation of an image about the z-axis

- *sculptmeshnd*
  Uses signed distance function to remove nodes from a given surface mesh and re-aligns the adjacency matrix.

- *setuptbem*
  Precalculation for the tbem routine, such as inter-nodal distances and solid angle.

- *sharedge*
  Determines which nodes in a surface mesh are sharing an edge. Used in locating boundary edges.

- *simpplotsurf*
  Modification of the `Distmesh` function simpplot in order to display surface meshes rather than tetrahedral volumes.

- *solidangletri*
  Determines which nodes in a surface mesh are sharing an edge. Used in locating boundary edges.

- *strike*
  Calculates the initial displacement vectors.

References

[1] P. Persson and G. Strang, "A simple mesh generator in `MATLAB`," *SIAM Review*, vol. 46, no. 2, pp. 329–345, 2004.

[2] J. Stratton, *Electromagnetic Theory*. McGraw-Hill Book Co., 1941.

[3] K. Mitzner, "Numerical solution for transient scattering from a hard surface of arbitrary shape - retarded potential technique," *Journal of the Acoustical Society of America*, vol. 42, no. 2, pp. 391–397, 1967.

[4] J. Smith, *Physical Audio Signal Processing for Virtual Music Instruments and Digital Audio Effects*. W3K Publishing, 2005.

[5] S. Petrausch and R. Rabenstein, "Wave field simulation with the functional transformation method," in *Int. Conf. on Audio, Speech, and Signal Processing*, pp. 1–4, 2006.

[6] Y. Kawai and T. Terai, "A numerical method for the calculation of transient acoustic scattering from thin rigid plates," *Journal of Sound and Vibration*, vol. 141, no. 1, pp. 83–96, 1990.

[7] O. von Estorff, ed., *Boundary Elements in Acoustics: Advances and Applications*. WIT Press, 2000.

[8] M. Friedman and R. Shaw, "Diffraction of pulses by cylindrical obstacles of arbitrary cross section," *Journal of Applied Mechanics*, vol. 29, no. 1, pp. 40–46, 1962.

[9] F. Hildebrand, *Introduction to Numerical Analysis*. McGraw-Hill Book Co., 1956.

[10] S. V. Duyne and J. Smith, "Physical modeling with the 2-d digital waveguide mesh," in *Proceedings of the International Computer Music Conference*, pp. 40–47, 1994.

[11] R. Shaw, "Diffraction of plane acoustic pulses by obstacles of arbitrary cross section with an impedance boundary condition," *Journal of the Acoustical Society of America*, vol. 44, no. 4, pp. 1062–1068, 1967.

[12] D. James, J. Barbi, and D. Pai, "Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources," in *ACM SIGGRAPH*, pp. 987–995, 2006.

[13] P. Persson, "Distmesh - a simple mesh generator in matlab: `http://www-math.mit.edu/~persson/mesh/`," July 2007.

[14] M. de Berg, M. van Krevald, M. Overmars, and O. Schwarkopf, *Computational Geometry - Algorithms and Applications*. Springer-Verlag, 2nd ed., 1998.

[15] T. Möller and B. Trumbore, "Fast, minimum storage ray/triangle intersection," *J. Graphics Tools*, vol. 2, no. 1, pp. 21–28, 1997.

[16] J. Lyness and R. Cools, "A survey of numerical cubature over triangles," in *Proceedings of the Symposia in Applied Mathematics*, pp. 127–150, 1994.

[17] A. van Oosterom and A. Strackee, "The solid angle of a plane triangle," *IEEE Transactions on Biomedical Engineering*, vol. 30, no. 2, pp. 125–126, 1983.

[18] G. Scavone, *An acoustic analysis of single-reed woodwind instruments with an emphasis on design and performance issues and digital waveguide modeling techniques*. PhD thesis, Stanford University, 1997.