BRIGHAM YOUNG UNIVERSITY - IDAHO

DEPARTMENT APPROVAL

of a senior thesis submitted by

Ryan Flamm

This thesis has been reviewed by the research committee, senior thesis coordinator, and department chair and has been found to be satisfactory.

Date

Jon Paul Johnson, Advisor

Date

Ryan Nielson, Committee Member

Date

David Oliphant, Senior Thesis Coordinator

Date

Stephen McNeil, Department Chair
The goal of this study was to evaluate the capability of the open source software called OpenFOAM for the purposes of acoustical studies. To do this the software was tested in both linear and nonlinear cases. For the linear portion of the study a room acoustics case was used. The software was able to accurately simulate wave propagation including wave speed and reflections. However, the fast fourier transform of the pressure data did not match well with physical data. This however, is likely due to sampling limitations resulting in low nyquist frequencies. For the nonlinear case it was desired to see a sinusoidal pressure wave of high amplitude transform into a saw-tooth wave. The software successfully produced such a result. Between both cases it was determined that OpenFOAM is capable of advanced studies in acoustics.
ACKNOWLEDGMENTS

I would like to thank Professor Jon Paul Johnson for his help in bringing this study to pass. I also wish to thank all of my physics professors for helping me to gain knowledge in physics, which ultimately came to fruition in this research.
# Contents

Table of Contents vi

List of Figures vii

1 Introduction 1
   1.1 Fluid Dynamics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
   1.2 Acoustics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
   1.3 OpenFOAM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

2 Methods 9
   2.1 Room Acoustics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
      2.1.1 Grid . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
      2.1.2 Boundary Conditions . . . . . . . . . . . . . . . . . . . . . . . 11
      2.1.3 Initial Conditions . . . . . . . . . . . . . . . . . . . . . . . . 12
      2.1.4 Solver . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
      2.1.5 2D vs. 3D . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
   2.2 Non-linear Acoustics . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
   2.3 Data Visualization & Analysis . . . . . . . . . . . . . . . . . . . . . 18

3 Results & Conclusion 19

Bibliography 25

A Case Files 27
   A.1 2D Room Acoustics Case Files . . . . . . . . . . . . . . . . . . . . . 27
   A.2 3D Room Acoustics Case Files . . . . . . . . . . . . . . . . . . . . . 33
   A.3 NonLinear Case Files . . . . . . . . . . . . . . . . . . . . . . . . . . 39

B MatLab code 47

Index 49
## List of Figures

1.1 Bonding of hydrogen atom occurs when the distance between atoms is such that the attractive and repulsive forces are balanced. .................................................. 2
1.2 The no-slip condition at a fluid-solid interface such that the velocity at the interface is zero. ................................................................. 3
1.3 Longitudinal wave of sound in air. As the molecules group together the pressure increases, while where they are more sparse the pressure is lower. ................................................................. 5
1.4 Under non-linear conditions, a sinusoidal sound wave will transform into a saw wave as it travels. .................................................. 6
2.1 a) 2D room acoustics case grid with 200 grid points in the $x$ direction and 120 in the $y$ direction for a total of 24,000 grid points. In addition, the grid is stretched such that the grid points are concentrated in the center and at the walls. b) the sound impulse with radius of .03m placed at the center of the grid. .................................................. 10
2.2 The boundaries of the grid were set such that the 4 faces orthogonal to the $xy$ plane were set to be walls while the parallel faces were set to empty. ................................................................. 11
2.3 Table of settings for the Boundary Conditions of the 2D case. .................................................. 12
2.4 a) the grid mesh of the the 3D room acoustics case shown in the $xz$ and $yz$ planes. b) the 3D grid. ................................................................. 15
2.5 The nonlinear case was set to have an inlet on the left face, outlets on the top and bottom faces, and a wall at the far right. .................................................. 16
2.6 Table of settings for the Boundary Conditions of the nonlinear case. .................................................. 17
3.1 a) the 3D room acoustics case at .01s, showing the acoustic wave reaching the closest walls. b) the 3D room acoustics case at .016s showing the first reflections off the walls. ................................................................. 20
3.2 The left column shows the results of the 3D simulation. The top image being the pressure data, while the bottom is the frequency spectrum. The right column shows the physical data taken in a BYU-I raquetball court room. Again the top is the pressure while the bottom is the frequency spectrum. .................................................. 21
3.3 The simulated frequency data zoomed in to see the peaks in the range of 0 Hz to 100 Hz.

3.4 On left is a flood plot of the pressure at .054s of the simulation. On the right is a line plot showing the pressure wave over distance. The first oscillation of the sine wave has already begun taking a saw tooth wave form.

3.5 A FFT of the nonlinear case ran at 0,4,8,12,16, and 20 meters. It is seen that as the sinusoid travels the FFT gets more jagged as the sawtooth wave forms.

3.6 As a side by side comparison, it is clear that the temperature and pressure are positively correlated.
Chapter 1

Introduction

1.1 Fluid Dynamics

All matter can be categorized as being either a solid or a fluid. In a solid the molecules are bound together, keeping relative fixed positions. Conversely, in a fluid the molecules are not bound and as such are free to move relative to each other. We can, however, sub-divide the category of fluids even further, into gasses and liquids.

Fluid dynamics is the branch of fluid mechanics dealing with the properties of fluids in motion. Some of the essential metrics of fluid dynamics are: velocity, pressure, and density. To understand fluid dynamics it is important to first understand the underlying physics, which covers scales from atomic to macroscopic.

On a basic level, fluids are substances comprised of molecules. These molecules exhibit and are subject to two basic forces; attraction and repulsion. As the molecules move closer together the attractive force gets stronger according to the square of the distance between molecules. [1] However, once the molecules get too close, the repulsive force becomes strong enough to counter the attractive force. In a solid, the molecules sit in an equilibrium state, bound by the attractive and repulsive forces.
1.1 Fluid Dynamics

![Diagram showing energy vs interatomic distance]

**Figure 1.1** Bonding of hydrogen atom occurs when the distance between atoms is such that the attractive and repulsive forces are balanced.

This behavior is exemplified in hydrogen bonding as seen in Figure 1.1. In fluids however, the kinetic energy is sufficient to avoid getting trapped in such an equilibrium state. Instead, when the molecules converge they can be modeled as perfectly elastic spheres, colliding with the repulsive force and ricocheting away from one another.

When observing fluids, we often look at how they interact with other fluids and solids. For instance, it is common for a fluid to flow past a solid boundary, such as a river and the ground containing it. To understand the fluid-solid interface we should first define adhesion and cohesion. Adhesion is the attraction of different molecules, while cohesion is the attraction of identical molecules. Because different types of molecules have different net charges, the forces of adhesion and cohesion will differ. In the case of the river, the force of attraction between the water and the ground is greater than the force of cohesion. This phenomenon leads to a fundamental boundary condition in fluid dynamics called the no-slip condition. This boundary condition affects the velocity profile of the fluid by essentially locking the molecules close to the surface in place, such that the velocity at the boundary is zero as seen in Figure 1.2.

Some of the most important equations to the study of fluid mechanics are the
1.2 Acoustics

Figure 1.2 The no-slip condition at a fluid-solid interface such that the velocity at the interface is zero.

Navier-Stokes equations. They are second-order nonlinear partial differential equations. Given the complexity of these equations, they are still a subject of much study even after over 100 years of use.

Fluid flow is found all across nature, from the blood in our veins to ocean currents. Understanding these flows allows us the ability to accomplish some extraordinary engineering feats. Some technological advancements we owe at least in part to an understanding of fluid flow are: airplanes, submarines, and accurate weather prediction.

1.2 Acoustics

Acoustics is traditionally described as the study of sound, or audible frequencies. However, in modern science the range of acoustics has been expanded to include lower and higher frequencies than the audible range. As such acoustics can be defined as the propagation of mechanical waves in gases, liquids, and solids. In this study the portion of acoustics that was studied was that of sound propagation in fluids. Restricting our view of acoustics in this way, it is inherently a part of fluid dynamics.
Acoustics can be divided into two main categories: linear and nonlinear. When sound propagates through a medium it compresses and rarefies the medium as a longitudinal wave as seen in Figure 1.3. When the compression, also known as condensation, reaches certain levels, the sound becomes non-linear. The wave will have greater values of condensation as the sound amplitude increases. Nonlinear effects take place around 200 dB, which is well above the threshold of pain at 130 dB. Typical sources of non-linear effects would be explosions, sonic booms, and other extraordinarily loud events. [2]

A common study of linear acoustics can be seen in the subfield of room acoustics. This is an important area of acoustics in that nowadays a large number of events take place in closed spaces, and so room acoustics plays an important role in our daily life. [3] Applying room acoustics in engineering offers society a wide range of benefits. For example, room acoustics helps us to isolate and soften loud noises inside buildings to protect peoples ears and improve work environments. Another common application is in optimizing speech intelligibility in spaces such as auditoriums and conference rooms. In particular, room acoustics is interesting due to the confined reflections of sound waves off of walls, ceilings, and floors.

An important aspect of study in room acoustics is that of room modes. These modes are a collection of resonances that exist in almost all practical enclosed spaces excited by an acoustic source such as a loudspeaker. In physics, resonance is the tendency of a system to oscillate with greater amplitude at some frequencies than at others. Room resonances occur at frequencies being related to one or more of the room dimensions. [4] The modes are related to constructive interference of the wave reflections, and as such often relate to the distance between parallel walls. When these resonances fall within the audible range it can have undesired effects on sound clarity. [5]
1.2 Acoustics

Increased Pressure
Decreased Pressure
Atmospheric Pressure

Figure 1.3 longitudinal wave of sound in air. As the molecules group together the pressure increases, while where they are more sparse the pressure is lower.

In nonlinear acoustics, disturbances with a sawtooth shaped wave pressure profile are typical. This is because in a nondispersive medium, any periodic perturbation turns into a saw at long distance, where each period contains a discontinuity (shock) and a rectilinear segment of the profile. [6]

Normally, sound we encounter results from from small pressure fluctuations. [7] This means that the density and, by extension, the temperature are relatively constant. As a result of a constant temperature, a uniform speed of sound can be assumed. However, with very loud sound amplitudes, the air compresses enough that the temperature increases sufficiently to noticeably affect the speed of the sound wave. Following the equation for the speed of sound,

\[ c_0 \approx 331.4 + 0.6T, \]  

where T is temperature, the tops of the sine wave, being at the compressed sites, travel faster than the troughs, at the rarefied sites. Over time this results in a discontinuity in the pressure, which is visually represented as a saw-tooth wave as seen in Figure 1.4. This discontinuity occurs at what is called the shock formation or discontinuity distance and is written as
Figure 1.4 Under non-linear conditions, a sinusoidal sound wave will transform into a saw wave as it travels.

\[ \bar{p} = \frac{\rho_0 c_0^2}{p_{\text{init}} k}, \]  

where \( p_{\text{init}} \) is the pressure amplitude of the initial sinusoid, \( \rho \) is the density, \( k \) is the wave number, and \( \beta \) is the coefficient of non-linearity of the medium (\( \beta = 1.201 \) in air). [8]

Acoustics is a rich subject with connections to many different disciplines. By understanding acoustics, we have been able to make many advancements and improvements in our quality of life. For example, we are able to improve speech intelligibility in auditoriums, create noise cancelling devices for personal and industrial settings, minimize noise pollution, use ultrasound imaging, and use ultrasound based therapies. The military is particularly interested in the study of nonlinear acoustics due to the effects of high-intensity sound on servicemen. As we continue to investigate acoustical phenomena we will be able to accomplish even more.
1.3 OpenFOAM

Before the advent of computational modeling, acoustical studies typically relied on using fundamental equations and averages of acoustical data to calculate characteristics of sound propagation. These traditional methods have had a great deal of success in helping to understand and model acoustical spaces. However, these methods often do not sufficiently describe many acoustical phenomena and as such require a more detailed investigation. [9] Through the use of computational modeling, a greater level of understanding can be achieved.

In the field of acoustics there are many different programs one can use to model sound propagation. However these programs can be quite expensive. There is, however, an open source program called OpenFOAM (Open Field Operation and Manipulation) that may be of use in Acoustics. The program is a fluid dynamics software that focuses on fluid flow and the variables and parameters necessary to model them. OpenFOAM offers many features to solve complex fluid flows including for example turbulences and chemical reactions up to solid dynamics and electromagnetic and multiphysics applications. [10] Though the program has been mostly used in other areas of fluid dynamics, acoustics is inherently a fluid dynamics problem that deals primarily with pressure. OpenFOAM does handle pressures of fluid flows among other variables and therefore should be capable of modeling acoustics. This work describes an effort to evaluate OpenFOAM for serious acoustics research.

OpenFOAM offers many features that are desirable in the study of acoustics. One benefit is that it supports a visual representation of the computational data, allowing one to watch the sound waves propagate and study their behavior. In addition, since the program was built for a much broader study of fluid dynamics, it tracks more information than typical acoustical software would. For example, it can track
velocities and thermophysical properties. In addition, the program can handle many
different types of fluid situations such as laminar and turbulent, steady and transient,
as well as compressible and incompressible flows. Moreover, since OpenFOAM is open
source; anyone can build custom solvers and boundary conditions to meet new needs.
When all these benefits are accounted for, it is easy to see the possibility of a much
broader, advanced, and cost effective acoustical study than other programs would
permit.

To ascertain whether OpenFOAM is suitable for acoustical studies, it was tested
in both linear and nonlinear cases. For linear acoustics, a room acoustics case was
tested, being a common application of acoustics in the work field. Secondly, it was
desirable to evaluate whether or not the software would be able to explore more
complicated areas of acoustics. Therefore, a second case was chosen in which the
goal was to observe the transformation of sinusoidal waves into saw-tooth waves in a
nonlinear acoustical situation.
Chapter 2

Methods

2.1 Room Acoustics

2.1.1 Grid

The room acoustics case was done first in 2D, with dimensions of 12 by 6 meters. To build a grid in OpenFOAM, one has to first define all the vertices of the grid. The vertices can be defined according to a wide range of cell types such as hexahedron, wedge, prism, pyramid, and tetrahedron. Because the room acoustics case requires a rectangular prism, the hexahedron cell type was used with vertices in Cartesian coordinates. Once the vertices were defined they were grouped together in what are called blocks. Being a simple 2D square room, only one block was required for the case. Then the faces of the block had to be defined and assigned a type. In this case the four faces on the $x$-$y$ plane were categorized as walls, while the remaining two faces were assigned empty as seen in Figure 2.2.

OpenFoam is inherently designed to use three dimensional grids, and as such one must use a 3D grid and treat it as being two dimensional. In order to make the 3D grid two dimensional, the faces corresponding to the undesired dimension were
2.1 Room Acoustics

Figure 2.1 a) 2D room acoustics case grid with 200 grid points in the $x$ direction and 120 in the $y$ direction for a total of 24,000 grid points. In addition, the grid is stretched such that the grid points are concentrated in the center and at the walls. b) the sound impulse with radius of .03m placed at the center of the grid.

assigned the type called *empty*. This boundary condition tells OpenFOAM to ignore any information at that face as though it didn’t exist at all. In addition, the grid is made to be very small in that direction. In this case a $z$ dimension of .01 m was used, which is inconsequential compared to the 12 m used in the $x$ and 6 m in the $y$ dimension.

For the grid, a total of 24,000 grid points were used – 200 points in the $x$ and 120 and $y$ dimensions. In addition, the grid was refined such that the grid points were concentrated in the middle and borders as seen in Figure 2.1. This refinement was chosen to accommodate the initial conditions as well as the complex wave superpositions and reflections, particularly near the walls. Though a uniform spacing is ideal, it was not possible for this case. In order to have the points sufficiently refined in the middle and the edges of a uniform grid, significantly more points would have been needed. This increase in grid points would dramatically increase the computational time needed to run the simulation.
2.1 Room Acoustics

Figure 2.2 The boundaries of the grid were set such that the 4 faces orthogonal to the $xy$ plane were set to be walls while the parallel faces were set to empty.

2.1.2 Boundary Conditions

In OpenFOAM, one has the freedom to assign each face of the grid a boundary condition. These conditions tell the program how different parameters should behave at those boundaries. Given that the grid was created for the purpose of room acoustics, all sides were designated as solid walls, thus allowing for wave reflections. For proper physical behavior at the walls, each key parameter must be defined with an appropriate boundary condition. These key parameters include: velocity, pressure, and temperature.

For velocity, a no-slip condition was used due to the fluid-solid interface. For the temperature, the walls were set to zeroGradient. This boundary condition, applied to temperature, designates that there is no temperature gradient orthogonal to the wall. In other words, it makes the assumption that the walls are adiabatic. [11] Though there will always be some level of heat transfer in acoustics, it is a negligible amount and so a zero gradient boundary condition can be used.

For the pressure, a boundary condition called waveTransmissive was used. This condition is an approximation of the NavierStokes characteristic boundary conditions (NSCBC) applied to the pressure. This condition was chosen because of its ability
2.1 Room Acoustics

<table>
<thead>
<tr>
<th></th>
<th>p</th>
<th>U</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>walls</td>
<td>waveTransmissive</td>
<td>noSlip</td>
<td>zeroGradient</td>
</tr>
<tr>
<td>Front &amp; Back</td>
<td>empty</td>
<td>empty</td>
<td>empty</td>
</tr>
</tbody>
</table>

**Figure 2.3** Table of settings for the Boundary Conditions of the 2D case.

to adjust the reflectivity of the walls. This boundary condition requires 4 different pieces of information: gamma, fieldInf, lInf, and value. Gamma refers to the ratio of specific heats, fieldInf refers to far field value to be applied to the pressure, lInf refers to how far away the far-field condition should be, and value is the initial field pressure. These values were set to 1.4, 101325 Pa, .5, and 101325 Pa respectively.
The reflectivity of the boundary can be adjusted with the lInf value. The smaller the value, the more reflective the boundary will be. [12] All boundary conditions can be seen in Figure 2.3

2.1.3 Initial Conditions

The initial field values of the case were selected to mimic typical static atmospheric air conditions. The temperature was set to 300 K, the velocity was set to zero in all directions, and the pressure was set to atmospheric pressure (101325 Pa). It is worth noting that depending on the solver being used, the pressure might be read in kinematic form, which is pressure in pascals divided by the density of the medium. This unit is used in OpenFOAM purely for mathematical purposes. Incorporating the density into the pressure allows the Navier-Stokes equations to be solved without explicitly showing the constant fluid density. Having used a compressible solver, a constant density cannot be assumed and as such the solver calls for pressure in pascals.

In addition to the key parameters of the problem being set and in order for Open-
FOAM to run an accurate simulation, the thermophysical properties also needed to be defined. These were set in the thermoPhysicalProperties dictionary. In this dictionary, the number of moles was set to 1, with a molecular weight of 28.9, according to typical air. Specific heat was set to 1004.9, heat fusion to 2.544E06, dynamic viscosity to 1.846E−05, and a Prandtl number of 0.707.

Once the base characteristics of the fluid were set, a sound impulse was created. In room acoustics it is common to use a sound impulse to analyze acoustical properties of a room. To produce a sound impulse, a region at the center of the grid was set to have a higher pressure value than the rest of the room. To do this, the terminal command $setFields$ was used, which calls the dictionary called $setFieldsDict$ to alter the initial conditions. The dictionary was set to call a function named sphereToCell, which as the name implies allows one to impose new conditions in a specified spherical region. Since the simulation was to be in 2D, the sphere was simply placed on that 2D plane such that a circular slice would be imposed on the plane. The rest of the sphere would fall outside of the simulation and thus be void. The spherical region to be adjusted was placed at the center of the grid (6, 3, .05) and given a radius of .03 m. The specified region was given a pressure value of 123,765 Pa. This value represents a sound impulse from a starter pistol. The peak sound pressure level reached by a starter pistol is 181 dB. [13] This decibel value corresponds to an overpressure of 22440 Pa, and when added to the base atmospheric pressure gives the value of 123,765 Pa. Though a quieter impulse could have been used, a large differential was desired for easier visualization.

2.1.4 Solver

Due to the broad study of fluid dynamics that OpenFOAM supports, there exist many different solvers. Each solver is based on different mathematical and physical princi-
2.1 Room Acoustics

samples and assumptions. This is done in order to find the right balance of computational speed and accuracy for various simulations. These include general categories such as incompressible, compressible, multiphase, combustion, heat transfer and buoyancy driven flows, particle-tracking, electromagnetics, stress analysis, and even finance. Given the broad range of applications it is important to use the right solver.

Though gas is compressible, for the purpose of room acoustics it is safe to assume the gas is incompressible. [14] Even so, it was decided that a compressible flow solver would be used. This was done because the computational cost of being a little more accurate was thought to be negligible for the room acoustics case.

Of the available compressible flow solvers, the rhoPimpleFoam solver was selected. The rhoPimpleFoam solver being a transient solver for turbulent flow of compressible fluids. Therefore, the solver would be able to sufficiently solve the room acoustics case.

2.1.5 2D vs. 3D

Once the simulation was successfully run in 2D, a 3D room was made. The 3D grid was given the dimensions of $12 \times 6 \times 12$ meters. These dimensions were made to mimic one of the racquetball courts on the BYU-Idaho campus. This particular room was chosen so that the simulation results could be compared with previously taken physical data in said room. Though the model is not an exact match, due to things such as a missing segment of wall where spectators can watch, it was deemed sufficient to give a sense of the programs capabilities. For this, the same grid refinement method was used.

Due to limitations of computational power, it was necessary to use less grid points than in the 2D model. So, 90 grid points were used in the $x$ and $z$ dimensions and 45 in the $y$. This gave a total of 364,500 grid points. This would give the grid a spatial
2.1 Room Acoustics

resolution of approximately 421 grid points in every cubic meter prior to any grid
stretching. Because the grid points are concentrated in the center and by the edges
(as seen in Figure 2.4), the actual number of grid points per cubic meter varies. In the
most unrefined sections of the grid there are approximately 216 grid points per cubic
meter. This is believed to be a sufficient grid refinement to see accurate acoustical
wave propagation while reducing computational time from that of a uniform grid with
more grid points. If the wave were to propagate incorrectly, it would likely be due to
the drastic change in grid points from the edge to the center. As the wave propagates,
it is possible that the speed of sound would get distorted in the transition.

The boundary conditions for the 3D room were nearly identical to that of the 2D
case. The main difference was to change the empty faces to be walls, like the other
tables, and in doing so convert the grid from 2D to 3D. The initial conditions were
also nearly identical to the 2D case. The only difference being in the coordinates of
the impulse origin being moved from (5,5,0.5) to (6,3,6). Additionally, being placed
in true 3D grid, the impulse was permitted to be spherical.

Figure 2.4 a) the grid mesh of the the 3D room acoustics case shown in the
xz and yz planes. b) the 3D grid.
2.2 Non-linear Acoustics

To see if OpenFOAM could simulate nonlinear acoustics, it was desired to see if sinusoidal pressure wave of large amplitude over a distance would transform to saw-tooth waves. To model this, a simple 2D grid was created, the dimensions of which were $100 \times 5 \times .1 \text{ m}$. The same trick of categorizing the faces orthogonal to the z axis as empty was used as before to treat it as 2D. However, now the left face was set to be an inlet, the top and bottom as outlets, and the right face set to be a wall as depicted in Figure 2.5. This was done so that the sinusoidal pressure wave could be introduced on the left and then reflected off the right wall. This reflection was put in place for the sole purpose of allowing the wave to travel across the grid twice. This, in theory, could give it sufficient time to transform while not slowing down computational time with an excessively large grid.

The grid was set to have a total of 6,000 grid points. This was done with 300 grid points in the $x$ direction and 20 in the $y$. Most of the grid points were put in the $x$ dimension because the simulation was primarily looking at the progression and changes in the wave in the $x$ direction.

In the velocity file, the far right was set to noSlip. The inlet was set to pressureInletOutletVelocity. The top and bottom outlets were set to inletOutlet. In the pressure file, the outlets were set to zeroGradient. The reflecting wall was also set to zeroGradient. The inlet was set to uniformTotalPressure.

The uniformTotalPressure boundary condition allows one to insert a table of val-
2.2 Non-linear Acoustics

<table>
<thead>
<tr>
<th></th>
<th>Inlet</th>
<th>Outlets</th>
<th>Wall</th>
<th>Front &amp; Back</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>uniformTotalPressure</td>
<td>zeroGradient</td>
<td>zeroGradient</td>
<td>empty</td>
</tr>
<tr>
<td>U</td>
<td>pressureInletOutletVelocity</td>
<td>inletOutlet</td>
<td>noSlip</td>
<td>empty</td>
</tr>
<tr>
<td>T</td>
<td>zeroGradient</td>
<td>zeroGradient</td>
<td>zeroGradient</td>
<td>empty</td>
</tr>
</tbody>
</table>

**Figure 2.6** Table of settings for the Boundary Conditions of the nonlinear case.

ues. The table has two columns, one specifying a time and the other a pressure value. The pressure value at the boundary will then change according to the table provided to it. The sine wave values were created in Excel and imported into the OpenFOAM pressure file. The sine wave was created with a frequency of 30 Hz and an amplitude of 200,000 Pa and written out every .001 seconds. This amplitude corresponds to a sound pressure level of 200 dB. The amplitude of the wave was increased by 1 atm to fit into the simulation. It was then increased an additional 100,000 Pa to avoid negative numbers. If any part of the sinusoid were negative OpenFOAM would crash. As for the other initial conditions, the uniform field pressure was set to atmospheric plus 100,000 Pa for a value of 201,325 Pa, temperature to 300 K, and velocity to 0 m/s. All boundary conditions can be seen in Figure 2.6.

For the solver, *sonicFoam* was chosen. The solver is defined as a transient solver for trans-sonic/supersonic turbulent flow. Because nonlinear acoustics deals with high sound amplitudes, it is not uncommon for this area of acoustics to reach the sonic level. As such it was decided that sonicFoam would be an appropriate solver.
2.3 Data Visualization & Analysis

In this study, different methods were used to visualize and analyze the data. In every case a 3D visualization through ParaFoam was used. For the room acoustics cases, a frequency analysis in MATLAB was also used.

ParaFoam is the principal post-processing tool included in the OpenFOAM package. This tool is actually a wrapper for the third-party product named ParaView; which is an open-source, multi-platform data analysis, and visualization application. For the purposes of this study this tool was used primarily for 3D visualization, though it can do much more. Some of its other features include line plots, contours, and glyphs. To access the simulation data in paraFoam one must simply use the terminal command \texttt{paraFoam} while in the corresponding directory of the simulation data. Then, by applying filters to the data through paraFoams GUI, one can view the simulation at any timestep or play through all time steps as an animation. For this study, dealing primarily with the pressure, the model was filtered to show the pressure data at every grid point over time. In this way the state of the acoustical waves could be seen at every time step. Additionally videos were made to see the waves propagate in real time. As needed, particularly in the 3D case, the model was broken down into slices to better visualize how things were progressing and changing at different locations in the 3D model.

MATLAB was used to take the Fast Fourier Transform (FFT) of both the simulated and physical data of the room acoustics case. By this means the frequency spectrum of both data sets could be compared. To do this both data sets were loaded into a MATLAB code. Then by using MATLABs \texttt{fft} command and plotting the output, the frequencies could be viewed side by side and compared. The MATLAB code can be seen in the Appendix.
Chapter 3

Results & Conclusion

In the room acoustics cases, it was seen that the software was able to seemingly model sound wave propagation accurately. It was found, as shown in Figure 3.1, that the sound wave propagated from the center to the walls (a distance of 3 meters) in approximately .009 seconds, giving a speed of sound of about $333 \text{ m/s}$. In addition, at .016 seconds it can be seen that the wave reflected off the walls orthogonal to the $zx$ plane. In seeing the proper speed of sound as well as acoustical reflections, it appears that OpenFOAM is capable of producing accurate visualizations of room acoustics.

However, in looking at the FFT of the simulation, it was seen that the frequency spectrum was quite a bit different from the expected values. In particular, it is noted that the frequency range was confined to around 500 Hz. In contrast, the physical data had a frequency range reaching upwards of 10,000 Hz. In addition, the room modes or resonant frequencies of the simulation do not appear to match that of the physical data in the lower frequencies. Both frequency spectra can be seen in Figure 3.2. To better see some of the individual frequency peaks of the simulated data, the plot was zoomed in to look at frequency range from 0 Hz to 100 Hz as seen in Figure 3.3. The simulation appears to have been affected by sampling limitations. The temporal
Figure 3.1 a) the 3D room acoustics case at .01s, showing the acoustic wave reaching the closest walls. b) the 3D room acoustics case at .016s showing the first reflections off the walls.

nyquist frequency of the simulation was 750 Hz while the average spatial nyquist frequency was about 644 Hz. These values suggest that with a higher sampling rate and a more refined grid, the frequency results would likely change and possibly better match the physical data. To do this, greater computational power than what was available in this study would be needed.

It is also worth noting some of the assumptions and simplifications that were made that likely affected the frequency spectrum. The room dimensions of the racquetball court were not taken accurately but rather estimated. In addition, the simulation did not account for the section of wall missing for the use of spectators. The amplitude of the impulse was not matched between the simulation and the physical data. Also, to lower computational time, the radius of the impulse was made bigger than what it probably should have been. Though the goal of this study was to get a general sense of the capabilities of OpenFOAM in acoustics, a more thorough methodology may be needed to further examine the accuracy of the software.

In the nonlinear case, a saw-toothed wave was achieved as seen in Figure 3.4. In the Figure, the first sine wave oscillation has taken a saw toothed shape while the following oscillation has yet to travel far enough to exhibit such a change. In
Figure 3.2 The left column shows the results of the 3D simulation. The top image being the pressure data, while the bottom is the frequency spectrum. The right column shows the physical data taken in a BYU-I raquetball court room. Again the top is the pressure while the bottom is the frequency spectrum.

Figure 3.3 The simulated frequency data zoomed in to see the peaks in the range of 0 Hz to 100 Hz.
addition, it was seen that the frequency spectrum became more jagged as the sinusoid transformed into a sawtooth wave. This result is an FFT was taken to show this at 0, 4, 8, 12, 16, and 20 meters as seen in Figure 3.5. Important to the formation of a saw-tooth wave is the capability of OpenFOAM to keep track of the thermophysical properties. In Figure 3.6 it is seen through a side by side comparison of the pressure and temperature that there is a positive correlation between the two variables. This is what is expected of a high amplitude pressure wave due to compression of the air as the wave propagates. This proves that the software is capable of following these parameters and by extension may have use in complex acoustical studies.

In general, it was seen that OpenFOAM has potential for studying acoustics. The 3D visualization tools built into OpenFOAM offer insights into the interpretation of the simulated data. In addition, the software is built to follow a large range of variables and even permits the development of new implementations to meet the needs of current and future studies. Though the software is very powerful and versatile, it does have limitations. Because OpenFOAM keeps track of so much data, it can have a very long computational runtime. For very complex simulations it may be necessary
Figure 3.5 A FFT of the nonlinear case ran at 0, 4, 8, 12, 16, and 20 meters. It is seen that as the sinusoid travels the FFT gets more jagged as the sawtooth wave forms.

to run the software on a supercomputer. Depending on the complexity of study being done, a more simplified and streamlined software may be more advantageous.

To better evaluate the full potential of OpenFOAM in acoustics, future work is needed. While this study verified OpenFOAM’s abilities to model acoustics in a general sense, work should be done to verify its accuracy. In particular, better grid refinement and by extension, more computational power, would be needed to better test the software’s accuracy. In addition, it would be beneficial to write new code specifically for the purpose of studying acoustics in this software. This could include a new solver that implements more accurate acoustical analogies or new boundary conditions that are more suited to acoustics.
Figure 3.6 As a side by side comparison, it is clear that the temperature and pressure are positively correlated.
Bibliography


Appendix A

Case Files

A.1 2D Room Acoustics Case Files

FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    object blockMeshDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
convertToMeters 1;

vertices
(
(0 0 0) //0
(12 0 0)//1
(12 6 0)//2
(0 6 0)//3
(0 0 .1)//4
(12 0 .1)//5
(12 6 .1)//6
(0 6 .1)//7
blocks
{
  hex (0 1 2 3 4 5 6 7) (200 120 1) simpleGrading (  
  (.2 .2 2)  
  (.3 .3 .5)  
  (.3 .3 2)  
  (.2 .2 .5)  
  )  
  (.2 .2 2)  
  (.3 .3 .5)  
  (.3 .3 2)  
  (.2 .2 .5)  
  )  
  1  
  )
);

edges
(
);

boundary
{
  fixedWalls  
  {
    type wall;  
    faces  
    (  
      (1 2 6 5)  
      (2 3 7 6)  
      (4 7 3 0)  
      (0 1 5 4)  
    );  
  }
  frontAndBack  
  {
    type empty;  
    faces  
    (  
      (0 3 2 1) //bottom  
      (4 5 6 7)//top  
    );  
  }
};
mergePatchPairs
(
);

// *************************************************************************
/*--------------------------------*- C++ -*----------------------------------*\
| ========= | |
| \ / F ield | OpenFOAM: The Open Source CFD Toolbox      | |
| \ / O peration | Version: 4.1                | |
| \ / A nd | Web: www.OpenFOAM.org           | |
| \/-- M anipulation | |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version 2.0;
    format ascii;
    class volScalarField;
    object p;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

dimensions [1 -1 -2 0 0 0 0];

internalField uniform 101325;

boundaryField
{
    fixedWalls
    {
        type waveTransmissive;
        field p;
        psi thermo:psi;
        gamma 1.4;
        fieldInf 101325;
        lInf 3;
        value uniform 101325;
    }

    frontAndBack
    {
        type empty;
    }
}

// *************************************************************************
FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    location "system";
    object setFieldsDict;
}

// ************************************************************************* //
/*------------------------------------- C++ --------------------------------*/
| ===== === | OpenFOAM: The Open Source CFD Toolbox |
| ===== === | Version: 4.1 |
| ===== === | Web: www.OpenFOAM.org |
| ===== === |

FoamFile
{
    version 2.0;
    format ascii;
    class volScalarField;
    object T;
}
dimensions [0 0 1 0 0 0];

internalField uniform 300;

boundaryField
{
    fixedWalls
    {
        type zeroGradient;
    }

    frontAndBack
    {
        type empty;
    }
}

dimensions [0 0 1 0 0 0 0];

internalField uniform (0 0 0);

boundaryField
{
    fixedWalls
    {
        type noSlip;
    }

    frontAndBack
    {
        type empty;
    }
}
FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    location "system";
    object controlDict;
}

application rhoPimpleFoam;
startFrom startTime;
startTime 0;
stopAt endTime;
endTime .3;
deltaT .0002;
writeControl timeStep;
writeInterval 10;
purgeWrite 0;
writeFormat ascii;
writePrecision 6;
writeCompression off;
timeFormat general;
timePrecision 6;
runTimeModifiable true;
adjustTimeStep no;

maxCo 0.5;

functions {
    #includeFunc residuals
    #include "probe"
}

// ************************************************************************* //

A.2 3D Room Acoustics Case Files

FoamFile {
    version 2.0;
    format ascii;
    class dictionary;
    object blockMeshDict;
}

convertToMeters 1;

vertices {
    (0 0 0) //0
    (12 0 0) //1
    (12 6 0) //2
    (0 6 0) //3
    (0 0 12) //4
    (12 0 12) //5
    (12 6 12) //6
    (0 6 12) //7
};

blocks {
    hex (0 1 2 3 4 5 6 7) (90 45 90) simpleGrading (
( (.2 .2 .2) (.3 .3 .5) (.3 .3 .2) (.2 .2 .5) )
( (.2 .2 .2) (.3 .3 .5) (.3 .3 .2) (.2 .2 .5) )
( (.2 .2 .2) (.3 .3 .5) (.3 .3 .2) (.2 .2 .5) )
);

edges
(
);

boundary
(
    fixedWalls
    {
        type wall;
        faces
        (
            (1 2 6 5)
            (2 3 7 6)
            (4 7 3 0)
            (0 1 5 4)
            (0 3 2 1) //bottom
            (4 5 6 7)//top
        );
    }
    frontAndBack
    {
        type empty;
        faces
        ( );
    }
);
A.2 3D Room Acoustics Case Files

FoamFile
{
  version 2.0;
  format ascii;
  class volScalarField;
  object p;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions [1 -1 -2 0 0 0 0];

internalField uniform 101325;

boundaryField
{

  fixedWalls
  {
    type waveTransmissive;
    field p;
    psi thermo:psi;
    gamma 1.4;
    fieldInf 101325;
    lInf 3;
    value uniform 101325;
  }

  frontAndBack
  {
    type empty;
  }
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
A.2 3D Room Acoustics Case Files

FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    location "system";
    object setFieldsDict;
}

// ************************************************************************* //

// ************************************************************************* //
dimensions       [0 0 0 1 0 0 0];

internalField   uniform 300;

boundaryField   
  
    
    fixedWalls
    
    
      type      zeroGradient;
    
    
    frontAndBack
    
    
      type      empty;
    
  

    
// ************************************************************************* //
/*--------------------------------*- C++ -*----------------------------------*
| ========= | |
| \ / F ield | OpenFOAM: The Open Source CFD Toolbox |
| \ / O peration | Version: 4.1 |
| \ / A nd | Web: www.OpenFOAM.org |
| \// M anipulation | |

 FoamFile
  
    
    version 2.0;
    
    format  ascii;
    
    class  volVectorField;
    
    location  "0";
    
    object  U;
  
  
  
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions       [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField   
  
    
    fixedWalls
    
    
      type      noSlip;
    
    
    frontAndBack
    
    
      type      empty;
    
  
}
/ ************************************************************************* //
/*--------------------------------*- C++ -*----------------------------------*\
| ========= | |
| \ / F ield | OpenFOAM: The Open Source CFD Toolbox |
| \ / O peration | Version: 4.1 |
| \ / A nd | Web: www.OpenFOAM.org |
| \// M anipulation | |
\---------------------------------------------------------------------------*/
FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    location "system";
    object controlDict;
}

// *********************************************************************** //
application rhoPimpleFoam;
startFrom startTime;
startTime 0;
stopAt endTime;
endTime .3;
deltaT .0002;
writeControl timeStep;
writeInterval 10;
purgeWrite 0;
writeFormat ascii;
writePrecision 6;
writeCompression off;
timeFormat general;
timePrecision 6;
runTimeModifiable true;
adjustTimeStep no;
maxCo 0.5;

functions
{
#includeFunc residuals
#include "probe"
}

FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    object blockMeshDict;
}

convertToMeters 1;

vertices
(
    (0 0 0) //0
    (100 0 0) //1
    (100 5 0) //2
    (0 5 0) //3
    (0 0 .1) //4
    (100 0 .1) //5
    (100 5 .1) //6
    (0 5 .1) //7
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (300 20 1) simpleGrading (1 1 1)
);
edges
(
);

boundary
(
    fixedWalls
    {
        type wall;
        faces
        (
            (1 2 6 5) //right face
        );
    }
    inlet
    {
        type patch;
        faces
        (
            (4 7 3 0) //left face
        );
    }
    outlet
    {
        type patch;
        faces
        (
            (2 3 7 6) //Top face
            (0 1 5 4) //Bottom face
        );
    }
    frontAndBack
    {
        type empty;
        faces
        (
            (0 3 2 1) //back
            (4 5 6 7) //front
        );
    }
);

mergePatchPairs
(
);

// ************************************************************************* //
/*--------------------------------*- C++ -*----------------------------------*
/------------------------------------------------------------------------------------------------*/
FoamFile
{
  version 2.0;
  format ascii;
  class volScalarField;
  object p;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions [1 -1 -2 0 0 0 0];

internalField uniform 201325;

boundaryField
{
  fixedWalls
  {
    type zeroGradient;
  }

  inlet
  {
    type uniformTotalPressure;
    p0 table
    
      // sine wave frequency of 30Hz and amplitude of 200,000 Pa
      // adjusted pressure by 100000pa to avoid negative numbers
      // (time amplitude)
      
      (0 201325)
      (0.001 238801.262917145)
      (0.002 274949.910536936)
      (0.003 308490.358995799)
      (0.004 338234.421185738)
      (0.005 363128.398874989)
      (0.006 382290.410493204)
      (0.007 395041.632225726)
      (0.008 400930.345685654)
      (0.009 399747.940262896)
      (0.01 391536.303259031)
      (0.011 376586.336008773)
      (0.012 355427.648555158)
      (0.013 328809.797949738)
      (0.014 297675.734820343)
      (0.015 263128.398874989)
      (0.016 226391.646712861)
  }
}
A.3 NonLinear Case Files

(0.017 188766.896094137)
(0.018 151587.022567029)
(0.019 116169.141686986)
(0.02 83767.9495415051)
(0.021 55531.2745157174)
(0.022 32459.4148995968)
(0.023 15369.7028223496)
(0.024 4867.5498542622)
(0.025 1325)
(0.026 4867.5498542624)
(0.027 15369.7028223499)
(0.028 32459.4148995973)
(0.029 55531.2745157181)
(0.03 83767.9495415059)
(0.031 116169.141686986)
(0.032 15369.7028223496)
(0.033 188766.896094138)
(0.034 226391.646712862)
(0.035 263128.39887499)
(0.036 297675.734820344)
(0.037 328809.797949738)
(0.038 355427.648555158)
(0.039 376586.336008773)
(0.04 391536.303259031)
(0.041 399747.940262896)
(0.042 400930.345685654)
(0.043 395041.632225726)
(0.044 382290.410493204)
(0.045 363128.398874989)
(0.046 338234.421185737)
(0.047 308490.358995798)
(0.048 274949.910530035)
(0.049 238801.262917144)
(0.05 201324.999999999)
(0.051 163848.737082854)
(0.052 127700.089463063)
(0.053 94159.6410041996)
(0.054 64415.5788142614)
(0.055 39521.6011250097)
(0.056 20359.5895067956)
(0.057 7608.367742735)
(0.058 1719.6543143456)
(0.059 2902.0597371046)
(0.06 11113.6967409697)
(0.061 26063.663991228)
(0.062 47222.3514448432)
(0.063 73840.2020502631)
(0.064 104974.265179658)
(0.065 139521.601125012)
(0.066 176258.353287141)
(0.067 201325)
outOfBounds clamp;
}

outlet
{
    type zeroGradient;
}

frontAndBack
{
    type empty;
}

FoamFile
{
    version 2.0;
    format ascii;
    class volScalarField;
    object T;
}

dimensions [0 0 0 1 0 0 0];

internalField uniform 300;

boundaryField
{
    fixedWalls
    {
        type zeroGradient;
    }
    inlet
    {
        type zeroGradient;
    }
    outlet
    {
        type zeroGradient;
    }
A.3 NonLinear Case Files

```
frontAndBack
{
    type empty;
}
}

// ************************************************************************* //
/*--------------------------------*- C++ -*----------------------------------*
| ========= | |
| \ / F ield | OpenFOAM: The Open Source CFD Toolbox |
| \ / O peration | Version: 4.1 |
| \ / A nd | Web: www.OpenFOAM.org |
| \/ M anipulation | |
\*--------------------------------------------------------------------------*/
FoamFile
{
    version 2.0;
    format ascii;
    class volVectorField;
    location "0";
    object U;
}

dimensions [0 1 -1 0 0 0 0];

internalField uniform (0 0 0);

boundaryField
{
    fixedWalls
    {
        type noSlip;
    }

    inlet
    {
        type pressureInletOutletVelocity;
        value uniform (0 0 0);
    }

    outlet
    {
        type inletOutlet;
        inletValue uniform (0 0 0);
        value uniform (0 0 0);
    }

    frontAndBack
    {
```
A.3 NonLinear Case Files

type empty;
}
}

FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    location "system";
    object controlDict;
}

application sonicFoam;

startFrom startTime;

startTime 0;

stopAt endTime;

endTime .3;

deltaT 0.0002;

writeControl timeStep;

writeInterval 10;

purgeWrite 0;

writeFormat ascii;

writePrecision 6;

writeCompression off;

timeFormat general;
timePrecision  6;

runTimeModifiable true;

functions
{
  #includeFunc residuals
  #include "probe"
}

// ************************************************************************* //
Appendix B

MatLab code

close all
cal=csvread('calibration114dB.csv');
r4=csvread('racquetball4.csv');

% use cac{1,1} to call time column, cac{1,2} for center probe, etc...
fid = fopen('p3DGood');%('pRefine3DGrid'); %
cac = textscan(fid,'%f %f %f %f','Delimiter',',');
fclose(fid);

% set variables
N=150;
fs=50000;
tcal=0:1/fs:(N-1)/fs;
start=105000;%105000;
stop=120000;%120000
NN=1e6;
t=0:1/fs:(stop-start)/fs;

% fft of the physical presure data
% x axis in Hz
k=fft(r4(start:stop));
pk=(conj(k).*k).^2;
dfk=fs/length(r4(start:stop));
fk=0:dfk:(fs-dfk);
%figure
%plot(fk(1:3000),pk(1:3000))

% plot simulated vs actual pressure
figure %100=.02 seconds
subplot(1,2,1)
% Simulation of Pressure Data
plot(cac{1,1}(50:1400), cac{1,2}(50:1400))
title('Simulated Pressure Data')
xlabel('Time (s)')
ylabel('Kinematic Pressure')
subplot(1,2,2)
plot(t, r4((start):stop))
xlabel('Time (s)')
title('Physical Pressure Data')
ylabel('Pressure (Pa)')

% FFT of simulated data
b = fft(cac{1,2});
pb = (conj(b).*b).^2;
dfb = .0002; % Step size
fb = 0:2144; % Frequency domain

% Frequency comparison
figure
subplot(1,2,1)
plot(fb(2:500), pb(2:500))
title('Simulated FFT Data')
xlabel('Hz')
subplot(1,2,2)
plot(fk(1:3000), pk(1:3000))
title('Physical FFT Data')
xlabel('Hz')

%%%% % Simulation of Pressure Data
figure
subplot(2,2,1)
plot(cac{1,1}(50:1400), cac{1,2}(50:1400))
title('Simulated Pressure Data')
xlabel('Time (s)')
ylabel('Pressure (Pa)')
subplot(2,2,2)
plot(t, r4((start):stop))
xlabel('Time (s)')
title('Physical Pressure Data')
ylabel('Pressure (Pa)')
subplot(2,2,3)
plot(fb(2:500), pb(2:500))
title('Simulated FFT Data')
xlabel('Hz')
subplot(2,2,4)
plot(fk(1:3000), pk(1:3000))
title('Physical FFT Data')
xlabel('Hz')
Index

2D3D, 14
Acoustics, 3
BC, 11
Fluid Dynamics, 1
Grid, 9
IC, 12
nonlinear, 16
OpenFOAM, 7
results, 19
Solver, 13
visualization, 18