An Investigation and Application of the Finite Difference Time Domain Method as a Tool for Solving Equalization Problems in Acoustics

by
Ryan J. Matheson

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Physics

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

Abstract

This thesis investigates the issues in deriving the Finite Difference Time Domain Method, including the derivation of a unique method for exciting an FDTD system that is physically realistic in terms of acoustics. It is also the goal of this thesis to use the FDTD method as a tool for investigating various speaker placement configurations for use in bass equalization. A demerit function is then developed in order to assess how well a particular equalization method performs relative to any others.

Acknowledgements

I would like to thank my advisor John Vanderkooy and co-advisor Stanly Lipshitz for taking me under their wings and guiding me through the incredible field that is Acoustics. I would like to thank my collegue and friend Scott Mallais who worked with me throughout this masters thesis. I also would like to thank William Lewis for his invaluable advice that lead to the optimization of the FDTD code in MATLAB.

Dedication

I would like to dedicate this thesis to my entire family for always being there for me on the sidelines cheering me on, picking me up when I've been knocked down, and giving me a swift kick in the ass when I needed it.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The goals of this masters project can be summed up in two main foci. The first, is the development of a Finite Difference Time Domian (FDTD) acoustic simulation program and its use to investigate and assess the FDTD method's ability to properly simulate the physical characteristics involved in room acoustics. The second, is to investigate different equalization methods that have been shown to improve the room's frequency response in the bass region and to develop an appropriate measure of demerit to properly determine the effectiveness of an equalization technique. Both of these foci have the same single, overall goal, which is the equalization of bass.

1.1 Bass Equalization

Since the development of the first closed-box loudspeaker to be used in home audio systems and it being made available to the common public, the problem of proper placement and equalization has arisen. In a typical rectangular listening room, the hard, parallel surfaces of the walls, floor and ceiling cause the primary acoustic resonances due to the interaction of the loudspeaker coupling to the room. The resonances occur in each of the three dimensions of the room as standing waves. The most notable of the resonances are the ones at low frequencies that reflect from two parallel surfaces only (front-back walls, left-right walls and ceiling-floor walls). The other resonance modes are more complex and involve as many as all six boundary surfaces. In general though the rule of thumb is is that the more surfaces involved in a resonance the more that resonance is absorbed by the boundary from which it is reflecting, which is why most of the concern is with the resonances that occur between the fewest number of surfaces. This point will be evident when looking at room responses to various speaker placements. It is important to note that the lowest frequencies are

most excited in rooms where the wavelength is so long that the sound waves are not significantly affected by the objects in the room.

The two most noticeable effects that these resonances create are,

- the average sound pressure level at low frequencies has great variation depending on the listener position
- the sound level for a low frequency resonance decays at a noticeably slower rate than the rest of the sound spectrum.

Thus for several decades much time and research has been devoted to solving these issues that affect primarily sounds in the bass region. One of the earliest papers that the author has come across that attempts to address the issue of bass equalization is "High-Fidelity Sound System Equalization by Analysis of Standing Waves" by Allen R. Groh in 1974 [1]. The paper recognizes the shortcomings of electronic equalization at the time and attempts to utilize a physical analysis of standing waves to create an optimization method for speaker placement. The method is based primarily on moving the speaker toward a position where there is a pressure node that corresponds to the frequency someone would wish to minimize. From the paper the method works reasonably well. But the method does have shortcomings as well. When moving a speaker around in the room other modes of vibration may be excited while attempting to minimize the resonance of interest. It also might not be practical to place a speaker in its optimized position depending on what the room is used for. Thus something more robust is required.

The works of Welti [2, 3] help show that increasing the number of sub-woofers combined with appropriate placement of the additional woofers can yield a reduction in the resonant effects in the room. The work of Santillan [5] also deals with placing multiple sub-woofers and describes how a plane wave can be created in a listening room. This then leads into the works of J. Abildgaard Pedersen [4] and Celestinos (in three conference papers [6, 7, 8] and in two journal papers [9, 10]). The work of Celestinos was critical to both the placement and equalization portion of this thesis, as well as the creation of the simulation program and was the basis for this Masters project.

1.2 The Finite Difference Time Domain Method (FDTD) Simulator

The wave equation for acoustics, like most other wave equations, is difficult to solve analytically when not dealing with ideal boundary shapes and conditions. Thus like

most problems in physics we turn to solving the problem numerically. For acoustics some of the common numerical methods used for solving real world problems are the Finite Element Method(FEM), Boundary Element Method(BEM) and Ray Tracing. These methods make solving the wave equation easier by transposing the problem from the time domain into the frequency domain. Then after the problem is solved in the frequency domain the result has to be transposed again back to the time domain. Although these methods have advantages, the method of Finite Difference Time Domain has two significant advantages the others do not. The first, is that it computes the solution to the wave equation directly in the time domain as real time data. This allows for a real time picture to be made of what is happening in the room and helps in the development of a deeper intuition into more complex problems. It also prevents the need to transform the data back and forth between the time and frequency domains. The second advantage, is that it is accurate for low frequencies and becomes more accurate at higher frequencies when smaller grid spacings are used. The accuracy of the other methods on the low end of the spectrum is not well known by the author. The general feeling obtained from researching and reading about the FDTD method from other authors has been that the other methods are not well suited to low frequency analysis. This is especially true in the case of Ray Tracing where it is assumed the wavelengths are small and the waves can be treated as particle like beams bouncing around the room. Thus the FDTD method is very appropriate for simulating low frequencies, which happens to be exactly why it has been chosen for this area of research. The only hindrance to the method when it was first conceived in 1966 in a paper written by Kane Yee[37], is that the modern computer at that time, was not powerful enough to handle computing solutions using this method in a time efficient way.

Today's CPUs are much faster and can now make FDTD computations of reasonable spaces with reasonable frequency resolution on common home computers.

As a starting point for this research the papers presented by Adrian Celestinos [6, 7, 8] laid the ground work for learning about both equalization techniques and creating the FDTD simulation program. Although the papers provided most of the equations needed to make the simulation program, their derivation and application was left as a mystery. The paper that made it very clear how the FDTD method for acoustics is derived was by Soren Krarup Olesen in his paper 'Low Frequency Room Simulation Using Finite Difference Equations' [11]. This paper alone answered nearly every question about how the derivation was done and provided insight into how Celestinos performed his derivation. The difference between Olesen's equations and Celestinos' were that Olesen used h and k as the spatial and temporal intervals, whereas Celestinos used h/2 and k/2. This led to much confusion on the authors

part because either way did not yield a change in the constants for the differencing equations because the 2's eventually canceled each other out. The difference is that by doing half steps we can calculate both pressure and velocity in one interval of k with Celestinos' equations instead of in two with Olesen's. Thus the difference is only in making it easier to program the scheme. Olesen's paper also made it quite clear as to how the boundary condition was applied and made the assumptions being made, much more clear than in other papers. It is truly a must read for anyone doing FDTD research.

The only mystery that Olesen's paper does not shed a significant amount of light on is in the derivation and application of a source term to properly excite the system. The author then turned to other papers like Botteldooren [12, 13], and Lopez [14] and books like [15, 16, 17]. While these references do contain a wealth of knowledge it was still frustrating to the author that not one of these references contained information on how to properly excite the system with a source. Three papers found that did happen to include a method for how to excite the system are [18, 19, 20]. These three papers illustrated the source discussed in sub-section 2.4.3, which turns out to not be a physically realistic way to excite a system, and not really applicable for our application. A different way to excite the system was then discovered in a MATLAB script written by Nick Clark [21] on the Mathworks website that is then discussed in sub-section 2.4.2. This source term is assumed to be what most FDTD users are probably using to excite the system given the vague descriptions given in the preceding references. It is also not physical, with respect to the fact that it does not respect the acoustical equations used in the FDTD method and forces a pressure point to adhere to a particular function. This then led the author, under the instruction of his adviser Professor John Vanderkooy, to derive a method based on the complete form of the continuity equation as it is given in Morse & Ingard [22]. A derivation and full discussion of the results are given in sub-section 2.4.4. This "new method" is a unique achievement that has resulted from this work.

As an observation of the author's is it is seen that a majority of the papers reviewed spend most of their efforts on refining the boundary conditions in an attempt to capture the frequency dependence of a typical wall. This is strongly observed in the papers [23, 24].

Chapter 2

The FDTD method for Acoustics

The Finite Difference Time Domain method, or FDTD, is as powerful as it is simple. It can easily be applied to the wave equation for acoustics and be used to solve problems and situations that analytic methods would find impossible.

In this Chapter we go over the finer points of deriving the FDTD method for acoustics and its implementation in computer code.

2.1 Derivation[1]

2.1.1 The starting point - The centered difference equation

Before we dive right into the derivation of the FDTD method we need to recall some familiar formulas that most would have seen in any introductory Calculus class. The following are three variations of the definition of the derivative given by first principles.

$$\begin{aligned}
\frac{df}{dx}(x_0) &= \lim_{\triangle x \to 0} \frac{f(x_0 + \triangle x) - f(x_0)}{\triangle x} \\
\frac{df}{dx}(x_0) &= \lim_{\triangle x \to 0} \frac{f(x_0) - f(x_0 - \triangle x)}{\triangle x} \\
\frac{df}{dx}(x_0) &= \lim_{\triangle x \to 0} \frac{f(x_0 + \triangle x) - f(x_0 - \triangle x)}{2\triangle x}
\end{aligned}$$

In a computational application $\triangle x$ is a small finite value which leads to the following approximations,

[1] everything in this section is a variation of the derivation found in Olesen [11]

$$\frac{df}{dx}(x_0) \approx \frac{f(x_0 + \triangle x) - f(x_0)}{\triangle x} \tag{2.1}$$

$$\frac{df}{dx}(x_0) \approx \frac{f(x_0) - f(x_0 - \triangle x)}{\triangle x} \tag{2.2}$$

$$\frac{df}{dx}(x_0) \approx \frac{f(x_0 + \triangle x) - f(x_0 - \triangle x)}{2\triangle x} \tag{2.3}$$

Equations 2.1 and 2.2 are commonly called the Right and Left Difference equations, respectively. Equation 2.3 is called the Centered Difference equation, and is the key equation that will be used in our derivation. The other two equations will be useful when deriving the boundary conditions.

2.1.2 The Acoustic Equations

In deriving the FDTD method for acoustics 3 essential equations are required. The first, is the Linear Inviscid Force Equation, or Newton's Equation,

$$\nabla p = -\rho \frac{\partial}{\partial t} \vec{u}. \tag{2.4}$$

The second, is the Equation of Continuity,

$$\frac{1}{\rho c^2} \frac{\partial}{\partial t} p = \frac{q(r,t)}{\rho} - \nabla \cdot \vec{u}, \tag{2.5}$$

where p is the sound pressure, c is the speed of sound in the medium, $\vec{u}$ is the particle velocity vector, ρ is the density of the medium and $q(r,t)$ is the function that defines the rate of creation of fluid in $\frac{kg}{m^3 s}$.

Commonly in most of the literature equation 2.5 is written without the function $q(r,t)$ as,

$$\frac{1}{\rho c^2} \frac{\partial}{\partial t} p = -\nabla \cdot \vec{u} \tag{2.6}$$

and will be the form of the continuity equation that is employed for the main body of the FDTD method, but we will see that equation 2.5 is essential for deriving a physically realistic source model.

Applying equation 2.3 to equations 2.4 and 2.6 will result in the creation of four separate matrices. The first three will be the x, y and z components of the velocity vector, while the 4th matrix will be the pressure in the room. These four matrices

will be computed in a leap frog pattern where, for example, the components of the velocity are calculated first for a time t, and then the pressure matrix is calculated afterwards at time $t+1$. The reason for this will become clear after the derivation.

An interesting point to note is that there is another method for the derivation that only involves the Acoustic Wave equation,

$$\nabla^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0$$

Upon discretizing this equation we obtain only the pressure matrices, but for three adjacent time steps. The benefit here is that one only has to store three matrices instead of four. This certainly cuts down on the amount of memory needed and can also increase the calculation speed. It is not applied here because in order to derive a proper source term we need a q(r,t) that includes the rate of creation of fluid as given in equation 2.5.

2.1.3 Velocity Matrix Derivation

In light of the fact that velocity is a vector, $\vec{u} = u_x\hat{x} + u_y\hat{y} + u_z\hat{z}$, we must consider each component of the velocity separately. We apply equation 2.3 to each component of equation 2.4, where we substitute[2] $\frac{h}{2}$ and $\frac{k}{2}$ for $\triangle x$ in equation 2.3 when applying to a spatial derivative and time derivative, respectively. Thus the x component should look as follows,

$$\frac{p\left(x+\frac{h}{2},y,z,t\right) - p\left(x-\frac{h}{2},y,z,t\right)}{h} = -\frac{\rho\left(u_x\left(x,y,z,t+\frac{k}{2}\right) - u_x\left(x,y,z,t-\frac{k}{2}\right)\right)}{k}$$

Thus we re-arrange and solve for, in this case, $u_x\left(x,y,z,t+\frac{k}{2}\right)$ and time shift $t+\frac{k}{2} = T$, which gives us the following for all 3 components,

$$u_x(x,y,z,T) = u_x(x,y,z,T-k) + \ldots \\ \frac{k}{\rho h}\left(p\left(x-\frac{h}{2},y,z,T-\frac{k}{2}\right) - p\left(x+\frac{h}{2},y,z,T-\frac{k}{2}\right)\right) \quad (2.7)$$

[2]For this derivation we used the half step method which uses $\frac{h}{2}$ instead of h and $\frac{k}{2}$ instead of k for the approximation to the derivative. The reason for this is for ease of programming later on where we wish to have computed both $\hat{u}$ and p in one interval of k.

$$u_y(x,y,z,T) = u_y(x,y,z,T-k) + \ldots \\ \frac{k}{\rho h}\left(p\left(x, y-\frac{h}{2}, z, T-\frac{k}{2}\right) - p\left(x, y+\frac{h}{2}, z, T-\frac{k}{2}\right)\right) \quad (2.8)$$

$$u_z(x,y,z,T) = u_z(x,y,z,T-k) + \ldots \\ \frac{k}{\rho h}\left(p\left(x, y, z-\frac{h}{2}, T-\frac{k}{2}\right) - p\left(x, y, z+\frac{h}{2}, T-\frac{k}{2}\right)\right) \quad (2.9)$$

2.1.4 Pressure Matrix Derivation

Here we observe that there are only scalars in equation 2.6 which means we can derive the whole thing in one giant step. Apply equation 2.3 to both sides of 2.6 in the same way as we did with the velocity matrices. Resulting in,

$$\frac{u_x\left(x+\frac{h}{2},y,z,t\right) - u_x\left(x-\frac{h}{2},y,z,t\right)}{h} + \frac{u_y\left(x,y+\frac{h}{2},z,t\right) - u_y\left(x,y-\frac{h}{2},z,t\right)}{h} \ldots \\ + \frac{u_z\left(x,y,z-\frac{h}{2},t\right) - u_z\left(x,y,z+\frac{h}{2},t\right)}{h} = -\frac{1}{\rho c^2}\frac{p\left(x,y,z,t+\frac{k}{2}\right) - p\left(x,y,z,t-\frac{k}{2}\right)}{k}$$

followed by re-arranging and solving for $p\left(x,y,z,t+\frac{k}{2}\right)$ and time shift $t+\frac{k}{2}=T$ to obtain

$$p(x,y,z,T) = p(x,y,z,T-k) + \frac{\rho k c^2}{h}\left[u_x\left(x+\frac{h}{2},y,z,T-\frac{k}{2}\right) - u_x\left(x-\frac{h}{2},y,z,T-\frac{k}{2}\right)\right] \ldots \\ + \frac{\rho k c^2}{h}\left[u_y\left(x,y+\frac{h}{2},z,T-\frac{k}{2}\right) - u_y\left(x,y-\frac{h}{2},z,T-\frac{k}{2}\right)\right] \ldots \\ + \frac{\rho k c^2}{h}\left[u_z\left(x,y,z+\frac{h}{2},T-\frac{k}{2}\right) - u_z\left(x,y,z-\frac{h}{2},T-\frac{k}{2}\right)\right] \quad (2.10)$$

2.1.5 Results of Derivation

Considering the x-component for the velocity, $u_x(x,y,z,T)$ is calculated based on the velocity from $t = T-k$ seconds ago and the two pressures at $\left(x \pm \frac{h}{2}, y, z\right)$ from

$t = T - \frac{k}{2}$ seconds ago. This is a similar situation when $p(x, y, z, T)$ is calculated, except it is required that all three x,y,z-components are done as well.

What does this tell us? It means that we can use a leapfrog scheme for our calculations in one time step (1t) calculating all the p's at $t = T$, then calculate all the u's at $t = T + \frac{k}{2}$. Then again calculate all the p's again at $t = T + k$ based on the previous u's and p's we just calculated at T and $T + \frac{k}{2}$. The derivation can also be performed with a non-zero q which will be the main focus when we develop how to properly add a source term in the FDTD scheme.

2.2 Parameter Determination

2.2.1 Cell Size Choice

We must choose the size of the cell to be much much smaller than the smallest wavelength we wish, in order to have accurate results in our simulation. This concept differs from the idea of the Nyquist sampling rate where it is needed to sample twice as fast as the fastest frequency. The reason for this is that for spatial sampling we require that we have enough grid points close enough together that it can sample the wave at a minimum of 5 points.

Common choices range from between $\frac{1}{5}\lambda$ to $\frac{1}{10}\lambda$. The smaller the fraction we choose the more accurate the results will be but at the sacrifice of computational speed. For example, if we wish to accurately simulate 600Hz, which has a wavelength of 0.5733m, we may choose $\frac{1}{5}\lambda$ which requires a grid spacing of approximately 11.47cm. Thus choosing 10cm for our value of h (or$\triangle x$) to use as the grid spacing is even better.

2.2.2 Time Step Choice (stability)

Now that a spatial sampling step has been chosen we now need to choose a temporal sampling or time step. Because we are discretizing a partial differential equation we need to make use of the Courant Stability Condition, which says,

$$c \cdot \frac{\triangle t}{\triangle x} \leq 1 \tag{2.11}$$

Where $\triangle x$ is our h value and $\triangle t$ will be our k value. Equation 2.11 in 3-D looks like (see [12]),

$$c \cdot \triangle t \cdot \sqrt{\frac{1}{\triangle x^2} + \frac{1}{\triangle y^2} + \frac{1}{\triangle z^2}} \leq 1.$$

If $h = \triangle x = \triangle y = \triangle z$,

$$k = \triangle t \leq \frac{h}{c \cdot \sqrt{3}} \tag{2.12}$$

As a common example let's choose h=10cm. We then obtain, from equation 2.12, a $k = 1.678 \times 10^{-4} s$. Because of the less than or equal sign we can choose k to be a more managable number like 1.25×10^{-4} which corresponds to a time sampling frequency of 8kHz. Thus because of the Courant condition we are forced to sample in time at a much faster rate than is required by the Nyquist Theorem, which for 600Hz would have been 1.2kHz.

2.2.3 Parameter Verification/Exploration

In this subsection we wish to investigate the consequences of violating the Courant condition by using a time step choice that is larger than $k = 1.678 \times 10^{-4}(s)$. Also we wish to see the effects of what happens to a pulse that contains frequencies greater than those that are theoretically available to be calculated given our choice of h=10cm.

2.2.3.1 Time Step

In order to investigate what happens to the stability of the system a pulse of finite duration is sent into the system for various values of $k \geq 1.678 \times 10^{-4}$and the results of how stable the system is can be observed from the three plots in figure 2.1.

As can be observed, both cases for $k > 1.678 \times 10^{-4}(s)$yields a system that becomes unstable. The closer the k value is to 1.678×10^{-4} the longer it takes for the system to become unstable. But only the value $k = 1.678 \times 10^{-4}$ remains stable for all time.

2.2.3.2 Cell Size

The size of the cell that we choose directly affects the allowed frequencies that will propagate in the system without any numerical errors. The most significant numerical error that is suffered in the FDTD method is that of numerical dispersion. Numerical dispersion is totally analogous to the type of dispersion that is suffered in Solid State physics, see [34], where the wave speed of higher frequency components is reduced relative to the lower frequency components. For example a wave packet or impulse would be observed to be spreading out as it travels. Using the pulse that will be derived in section 2.4.4 we can test this phenomenon by changing the length of a squared raised-cosine pulse and observe it as it travels through the system at various distances. By making the pulse longer we are thus reducing the number of higher

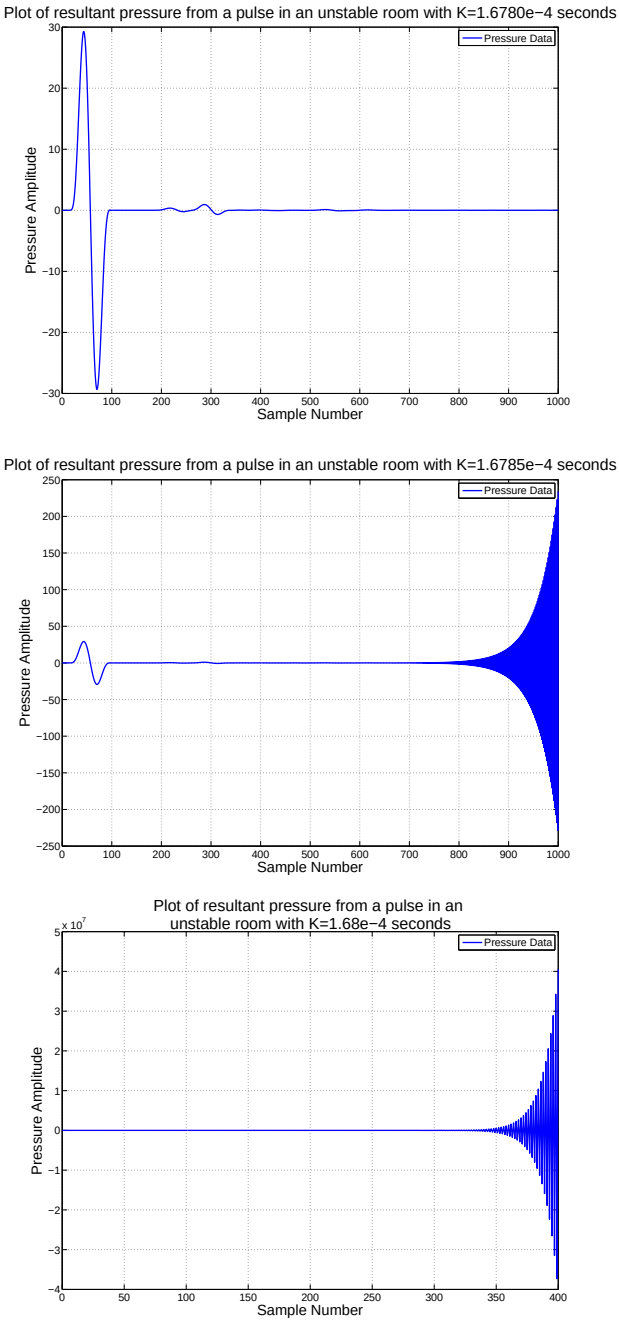

Figure 2.1: Figures depicting change in stability as K approaches the Courant Condition. Note the small reflection in the upper plot.

frequency components that are present in the pulse. The specification of the pulse in each case is given by its total length. The actual half-width of the pulse is a bit less than half of this value.

The parameters for this experiment are a $k = 1.25 \times 10^{-4} seconds$ and an h=10cm. The length of the pulses under investigation will range between 30*k to 160*k. The pulses will be observed at 1, 2.5, 5, and 10metres away from the source. The results are displayed in figure 2.2.

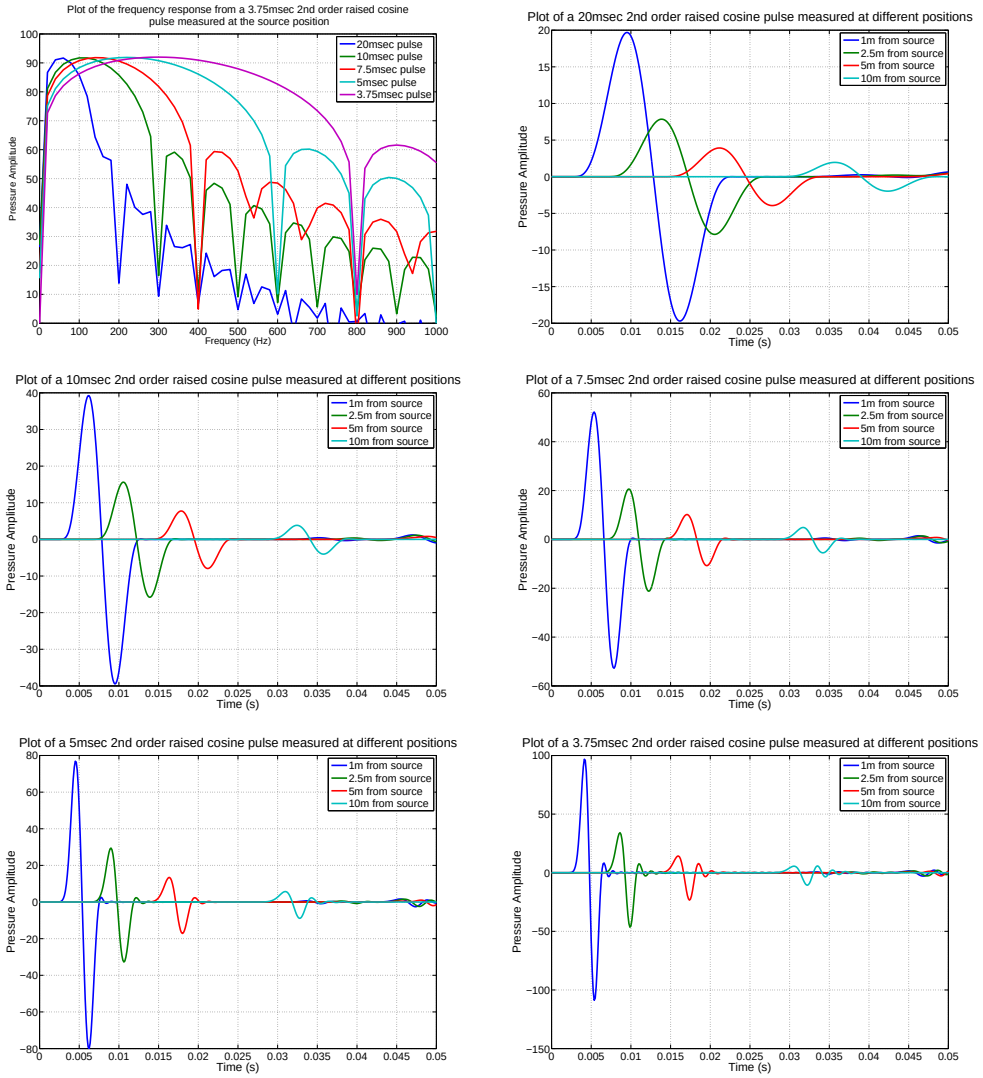

Figure 2.2: Collection of figures illustrating the numerical dispersion issue.

As can be observed in the figures from the 20msec pulse to the 3.75msec pulse, that once a pulse has more and more higher frequencies in it, its shape changes. This is most easily observed by the slight bump after the pulse in the shorter pulse plots. It can also be observed by noting that the amplitude of the negative peak of the pulse is getting larger relative to the positive peak as the amount of dispersion increases. Only the 20msec pulse appears free of dispersive elements. The only problem with using a 20msec pulse is that the decibel level for frequencies above 90Hz or so sharply decays. This is the region where in most small listening rooms there is still a lot of reverberant behavior. Thus perhaps a 10msec or 5msec pulse would be more ideal. A 10msec pulse should provide ample level for the frequencies of interest and still have a low amount of dispersion. To examine how much the dispersion affects the frequency response of our simulated rooms, we shall perform the following:

- Create in the simulator a room with dimensions L=5.6m, W=4.2m and H=2.4m
- apply the source term in 2.4.4 and place it in the corner of the simulation to excite maximum number of room modes
- plot frequency responses for 10, 5, and 2.5cm grid spacings and compare the results to determine effect of dispersion on the frequency responses.

Halving the grid spacing doubles the number of frequencies that will be dispersion free. Our area of interest is below 200Hz, thus we can determine whether or not a 10cm grid is appropriate for these kinds of measurements.

From the plot a response for a 10cm grid with a 16000Hz sampling is also introduced to show that there is no effect from simply changing the sampling frequency because it is observed that both 10cm grid plots lie directly on top on one another. It is also observed that for low frequencies all the plots lay essentially on top of one another, with the exception of small amplitude changes. It is assumed that these minor changes are due to the fact that when the grid spacing is made smaller the virtual sub-woofer gets closer to the simulated corner of the room, which is discussed in section 5.2.1. For now it should be safe to assume the small amplitude variations are due to the slight change in position of the sub-woofer, whereas numerical dispersion would show up as frequency shifts due to the speed change of the higher frequency components. From the plot this effect is observed and becomes stronger as frequency increases. Up to 200Hz the dispersion effect on the frequency response is extremely small to non-existent, thus we can conclude that both a 10cm grid and a 10msec squared cosine pulse together produce a minimum of numerical error in the simulation for the region we are interested in.

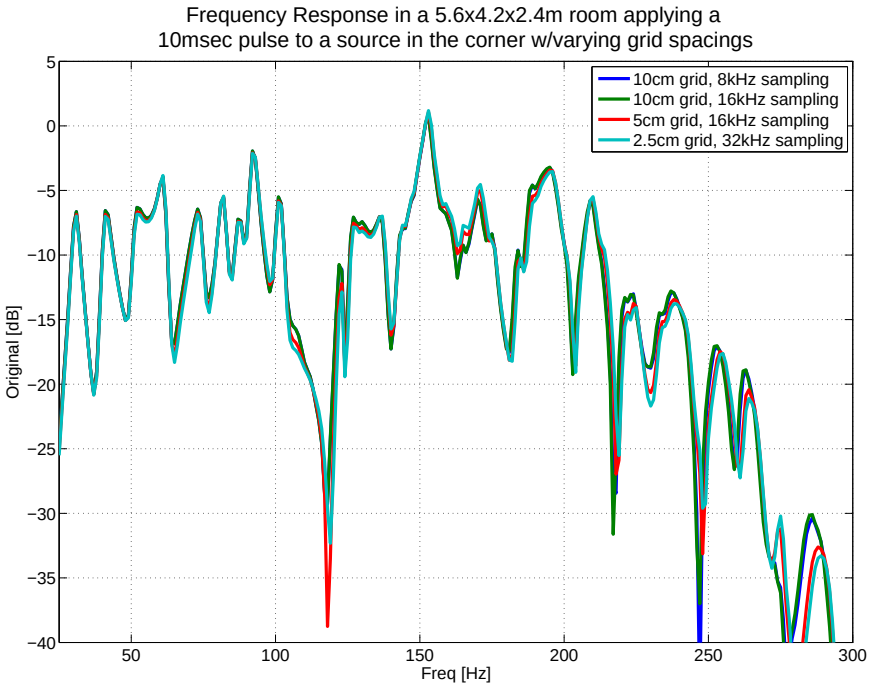

Figure 2.3: Plot displaying the effect numerical dispersion has at low frequencies from varying grid sizes from 10 to 2.5cm.

2.3 Boundary Conditions

In the bulk part of our grid when we calculate a pressure point or a velocity point that requires knowledge of the points around it. For instance, if we calculate the x-component of the velocity at (x,y,z), then we require the pressure points at $x \pm \frac{h}{2}$. This situation poses a problem once we reach a wall, or boundary. At a wall there is no pressure point for us to use in our calculation, thus we have to re-derive equation 2.7 with equations 2.1 and 2.2, depending on whether we encounter a left handed wall or a right handed wall.

The next piece we will need to re-derive our formula is the characteristic wall impedance Z, where

$$Z = \frac{p}{\vec{u} \cdot \hat{n}}, \tag{2.13}$$

where $\hat{n}$ the unit normal vector to the surface.

Here we make the assumption the Z is real, because for most wall impedances

the real part significantly dominates (according to Olesen [1]). Thus we can use the following as an assumption,

$$Z = \rho \cdot c \left(\frac{1 + \sqrt{1-\alpha}}{1 - \sqrt{1-\alpha}} \right). \tag{2.14}$$

Here α is the real absorption co-efficient. The last piece of our puzzle is not an obvious approximation but will be necessary later on. It is an approximation for the velocity given an earlier time and a later time,

$$u\left(x, y, z, t - \frac{k}{2}\right) \approx \frac{[u(x, y, z, t) + u(x, y, z, t-k)]}{2} \tag{2.15}$$

2.3.1 Right-Hand Wall (RHW) Boundary Derivation

For a RHW we wish to apply 2.3 to the right hand side of 2.4 as usual. This time we wish to simultaneously apply 2.2 to the left hand side of 2.4 instead. This yields,

$$\frac{p(x, y, z, t) - p(x - \frac{h}{2}, y, z, t)}{\frac{h}{2}} = -\rho \left(\frac{u_x(x, y, z, t + \frac{k}{2}) - u_x(x, y, z, t - \frac{k}{2})}{k} \right)$$

then, after a time shift of $T = t + \frac{k}{2}$, solve for $u_x(x, y, z, T)$ to obtain,

$$u_x(x, y, z, T) = u_x(x, y, z, T-k) - \frac{2k}{\rho h} \left[p\left(x, y, z, T - \frac{k}{2}\right) - p\left(x - \frac{h}{2}, y, z, T - \frac{k}{2}\right) \right]$$

We note that in this equation we have u and the first p occur both at position x, but according to our grid that's just not possible. We can only have u's at position x so we need to apply 2.13 to $p(x, y, z, T - \frac{k}{2})$ to obtain,

$$u_x(x, y, z, T) = u_x(x, y, z, T-k) - \frac{2k}{\rho h} \left[Z u_x(x, y, z, T - \frac{k}{2}) - p\left(x - \frac{h}{2}, y, z, T - \frac{k}{2}\right) \right].$$

This creates another issue. Now we have a u_x at time $T - \frac{k}{2}$. This directly violates our leap frog scheme. In order to rectify this we must now apply 2.15.

$$\begin{aligned} u_x(x, y, z, T) \;=\; & u_x(x, y, z, T-k) - \frac{2k}{\rho h} \cdots \\ & \left[Z \left(\frac{u_x(x, y, z, T) + u_x(x, y, z, T-k)}{2} \right) - p\left(x - \frac{h}{2}, y, z, T - \frac{k}{2}\right) \right] \end{aligned}$$

After collecting like terms we thus arrive at our equation for the boundary condition at a right-hand wall.

$$\begin{aligned} u_x(x,y,z,T) &= \frac{(\rho\cdot h - k\cdot Z)}{(\rho\cdot h + k\cdot Z)} u_x(x,y,z,T-k)\cdots \\ &\quad + \frac{2\cdot k}{(\rho\cdot h + k\cdot Z)} p\left(x-\frac{h}{2},y,z,T-\frac{k}{2}\right) \end{aligned} \tag{2.16}$$

2.3.2 Left-Hand Wall (LHW) Boundary Derivation

For the LHW the derivation is nearly identical except for the fact that seeing as we are approaching the wall from the right we now wish to apply equation 2.1 to the left hand side of 2.4 instead of equation 2.2. Thus we get,

$$\frac{p(x+\frac{h}{2},y,z,t)-p(x,y,z,t)}{\frac{h}{2}} = -\rho\left(\frac{u_x(x,y,z,t+\frac{k}{2})-u_x(x,y,z,t-\frac{k}{2})}{k}\right).$$

We thus repeat the same procedure as before but note that we obtain the following before collecting like terms[3].

$$\begin{aligned} u_x(x,y,z,T) &= u_x(x,y,z,T-k)\cdots \\ &\quad -\frac{2k}{\rho h}\left[p\left(x+\frac{h}{2},y,z,T-\frac{k}{2}\right) - Z\left(\frac{u_x(x,y,z,T)+u_x(x,y,z,T-k)}{2}\right)\right] \end{aligned}$$

We then obtain as the final equation for the LHW boundary,

$$\begin{aligned} u_x(x,y,z,T) &= \frac{(\rho\cdot h + k\cdot Z)}{(\rho\cdot h - k\cdot Z)} u_x(x,y,z,T-k)\cdots \\ &\quad - \frac{2\cdot k}{\rho\cdot h - k\cdot Z} p\left(x+\frac{h}{2},y,z,T-\frac{k}{2}\right) \end{aligned} \tag{2.17}$$

[3] An important point here is that because it is a LHW the dot product in equation 2.13 is negative, whereas before for a RHW it was positive. This point was crucial in obtaining the correct signs in the end result.

2.4 Adding in a Source Term

Typically when in a room we think of a speaker as creating some pressure at a point in the room. In most cases what is done is that at a desired position a pressure point is made to reflect a signal by forcing that point to adhere to a predefined function. This procedure may be OK in terms of observing how signals travel throughout the room but is not a physical way of exciting the room. Although at a point in our room a pressure is influenced by an external force, like a speaker, the pressure at that point is also affected by the air around that point reacting to the force being created by the source. Thus I would like to present the "not so good way" to excite the room and "the better way" to excite the room in an FDTD scheme.

2.4.1 Sources at Low Frequencies

The main focus of this research has been on the response in a listening room based on various speaker placements. Thus it is only sensible to have a source term that simulates the physics of a conventional loudspeaker. The response from a typical closed box loudspeaker (which will be our focus) is dependent on the frequencies it is trying to reproduce. At high frequencies closed box loudspeakers have a very directional response where most of the sound is projected forward in a beam like fashion. On the other hand, low frequencies are omni-directional and act like approximate point sources[4] that obey the compact source model described in [5]. The essence of the compact source model is that the point source is creating spherical waves whose pressure is described by the relation,

$$p(r) = \frac{\rho \cdot A(t)}{4\pi r} \tag{2.18}$$

where A(t) is the volume acceleration of the speaker cone. What closed box loudspeakers actually do is they produce volume velocity (the area of the speaker cone multiplied by the velocity of the cone), denoted Q(t), and the pressure a distance r away is directly proportional to the first derivative of Q(t), which is A(t). Thus the two qualities we are looking for from a source term are:

- Creates spherical waves that have a 1/r relationship
- the shape of the wave a distance r away is a derivative of the inputted volume velocity.

[4]This fact is shown strongly in the works of [6].

2.4.2 The Hard Source[5] ("not so good way")

This source model involves taking a point in the pressure matrix and applying a Gaussian pulse to it (we are not limited to only using a pulse, but can excite with any type of signal). For example,

$$p(x, y, z, t) = e^{-\frac{(t-t_0)^2}{\sigma^2}}.$$

This is a pulse centered at time t_0 and the width of the pulse is defined by σ. The issue when using a pulse like this is that it does not involve the velocity points around it in the grid to update itself based on how the pressure in the rest of the room is affecting it. It simply maintains a p irrespective of the acoustic equations. This might not seem like a big deal because all the other points around the source are being updated by the source itself. But, in fact this is a very big deal! Once the pulse has reached zero the pulse has begun to spread out into the room and there are now two choices.

1. Set p(x,y,z,t) back to equation 2.10
2. Leave p(x,y,z,t)=0 for all t.

To most people option 1 seems like the obvious and logical choice. If the room size is sufficiently large or the pulse sufficiently narrow and has returned to zero before any reflections come back then why not simply revert back to equation 2.10 and continue to calculate normally? Although this sounds like a simple solution it turns out to be very troublesome and fundamentally unusable. What happens is that this will introduce an unexpected discontinuity that introduces errors into the calculation scheme.

The second option is the stable option here except for the fact that we are constantly giving a point in the room a value of 0 for all time. Physically what this does is create an area of "quiet" in the room, where the size of this area is dependent on the size of the grid spacing. Clearly this isn't physically possible, but one could imagine that if the region is sufficiently small then the error this introduces may well be just as small. One could compare this method and what one would get with our model that we derive from q(r,t) in 2.5 that we will introduce shortly.

[5]The "hard source" is a term encountered while browsing the Mathworks website looking for a FDTD example. http://www.mathworks.com/matlabcentral/fileexchange/21000.

2.4.3 Area Pulse[6]

Another source model, that couldn't even really be classified as a source model (but for completeness I have included it), is what I have nicknamed the Area Pulse or Spatial Pulse. Instead of being a pulse that evolves in time like that Hard Source it is a pulse that is defined over an area as an initial condition. The program calculates the net result based on this initial condition. An example of the Area Pulse is given by,

$$p(r) = \begin{cases} \frac{1}{2}\left(1 + cos\left(\frac{\pi r}{R}\right)\right) & r \leq R \\ 0 & r > R \end{cases}.$$

As with the other methods for exciting the FDTD system there are also problems with this choice as well. For instance, this pulse is only defined in the pressure matrix with nothing said about what the initial condition might be in the velocity matrix, although upon observation when applying this pulse this doesn't seem to yield any major issues except that it is physically impossible. The second problem with this method is that it is limited only to being able to define an initial condition and not be able to excite the system past t=0.

2.4.4 New pulse

So far we have seen two methods of exciting the simulated room. But how do we know how that room will react to a specific speaker? One method can be to measure the speaker's impulse response in an anechoic situation and apply its transfer function to the source. But is there an easier way (or just some way to avoid having to have measured data all the time)? It can be shown that by solving some simple coupled differential equations that it is possible to work out a speaker cone's volume velocity and thus relate it to the function q(r,t) as seen in equation 2.5[7]. Thus we can relate q(r,t) to the voltage that is applied from an amplifier, which can be known with great precision, to properly simulate an actual speaker in our FDTD model by using our knowledge of the speaker's parameters. But for now let us focus on creating a source that acts like the a simple source as discussed in section 2.4.1.

We begin by re-deriving equation 2.10[8] from equation 2.5 instead of from equation 2.6, which has a non-zero q(r,t) term. Recall that we apply 2.3 to 2.5 to now obtain,

[6] see references [18, 19, 20].

[7] The method of which I am talking about can be seen in [4].

[8] For the sake of simplicity I will only derive with the x-component present. To include the other two components follow the same procedure.

$$\frac{q(x,t)}{\rho} - \frac{\left(u_x\left(x+\frac{h}{2},t\right) - u_x\left(x-\frac{h}{2},t\right)\right)}{h} = \frac{1}{c^2\rho}\frac{\left(p\left(x,t+\frac{k}{2}\right) - p\left(x,t-\frac{k}{2}\right)\right)}{k}.$$

After time shifting and re-arranging in the same way that was done in 2.1.4 we obtain the following,

$$p(x,T) = p(x,T-k) + kc^2 \cdot q\left(x, T-\frac{k}{2}\right) - \frac{c^2\rho k}{h}\left[u_x\left(x+\frac{h}{2},T-\frac{k}{2}\right) - u_x\left(x-\frac{h}{2},T-\frac{k}{2}\right)\right]. \quad (2.19)$$

Equation 2.19 contains the intensive function q which is the rate of creation of fluid in the system, which has units of $\frac{kg}{m^3 s}$. But we want a relation that contains the extensive volume velocity function Q, which has units of $\frac{m^3}{s}$.Therefore we can relate q and Q by multiplying q by the volume of the cell about the pressure point of interest, and then divide it by the density of the medium. Thus

$$Q = \frac{\triangle V}{\rho} \cdot q \quad or \quad q = \frac{\rho}{\triangle V} \cdot Q. \quad (2.20)$$

Here $\triangle V = \triangle x \cdot \triangle y \cdot \triangle z = h^3$. Substituting 2.20 into 2.19 yields

$$p(x,T) = p(x,T-k) + \frac{kc^2\rho}{h^3} \cdot q\left(x, T-\frac{k}{2}\right) - \frac{c^2\rho k}{h}\left[u_x\left(x+\frac{h}{2},T-\frac{k}{2}\right) - u_x\left(x-\frac{h}{2},T-\frac{k}{2}\right)\right]. \quad (2.21)$$

2.4.5 New Pulse Testing

If we review our criteria for what makes an acceptable pulse from section 2.4.1 we recognize that a physical spherical pulse will have a 1/r dependence. Thus by measuring the peak pressure at two different locations and comparing we should obtain results that correspond to the theory. We do this by using the formula

$$\frac{P_1}{P_2} = \frac{r_2}{r_1} \Rightarrow P_2 = \frac{P_1 r_1}{r_2}. \quad (2.22)$$

Using the data found in the plot entitled "Plot of a 5msec squared raised-cosine pulse measured at different positions" in section 2.2.3.2 we assume the peak pressure

for the pulse measured at 1m away from the source is $P_1$and is used as a reference to see if the measured peak values at 2.5, 5 and 10 metres agree with what is expected in equation 2.22. The results are summarized in the following table.

Table 2.1: Table of Expected and Measured Pressure values from a 1/r spherical pulse using a 5msec pulse.

r(m)	2.5	5	10
Expected Pmax	30.8178	15.4089	7.7044
Measured Pmax	29.4736	13.4615	5.7319
error(%)	4.3616	12.6380	25.6026

The experiment is also repeated for the data using the 10msec and 20msec pulses.

Table 2.2: Table of Expected and Measured Pressure values from a 1/r spherical pulse using a 10msec pulse.

r(m)	2.5	5	10
Expected Pmax	15.7068	7.8534	3.9267
Measured Pmax	15.6220	7.7520	3.8362
error(%)	0.5396	1.2909	2.3045

Table 2.3: Table of Expected and Measured Pressure values from a 1/r spherical pulse using a 20msec pulse.

r(m)	2.5	5	10
Expected Pmax	7.8757	3.9379	1.9689
Measured Pmax	7.8554	3.9236	1.9588
error(%)	0.2580	0.3621	0.5145

As can be observed from the data in the tables the error that is observed increases over the distance the pulse travels and also increases as the pulse length gets smaller, or in other words, as there are more higher frequencies present in the pulse. A most likely conclusion that can be drawn from this is that the result of the increasing error over distance and the increasing error with shorter pulse widths may be due to the numerical dispersion, where the higher frequencies are traveling at different speeds than the rest of the pulse, thus changing its shape over time. Thus the shorter the pulse the more distortion of the pulse and the greater the error.

2.4.6 New Pulse Drawbacks

In a perfect setting it would be nice to be able to excite our system with a sharp impulse that had a flat response for all frequencies. Given that there are dispersion effects at higher frequencies a sharp impulse is not an option for us. The reader will have noticed that we have already used a squared raised-cosine pulse. The reason for this is that a raised-cosine pulse starts off gently and finishes gently so that there are no sharp changes that would introduce unwanted high frequeny content. The raised-cosine pulse on its own would be an ideal pulse to use because its frequency response is reasonably flat from 0Hz up to some cut-off frequency(based on the length of the pulse). Unfortunatly when using the raised-cosine pulse for the function Q(t) we end up exciting the system with dQ(t)/dt as we have already observed, which does not have a flat response. In order to have dQ(t)/dt equal to a raised-cosine pulse we have to set Q(t) equal to the initial first half of the cosine pulse and then hold it at its peak value. This does not prove to be very helpful for two reasons. The first, is that this situation is unphysical. Holding Q(t) at a positive value implies that the speaker cone is travelling with a constant velocity, which is just not possible physically. The second reason is that the room becomes pressurized. The pressure level rises to a fixed value and never decays. This is problematic because we are unable to take a proper FFT of something that does not decay to zero unless we apply a window to it. This is a very bad idea because the frequency response at the freqencies of interest will be affected by the window and will make the measurement very difficult to interpret. This is why we use raised-cosine pulses. We could have used a Gaussian pulse as well.

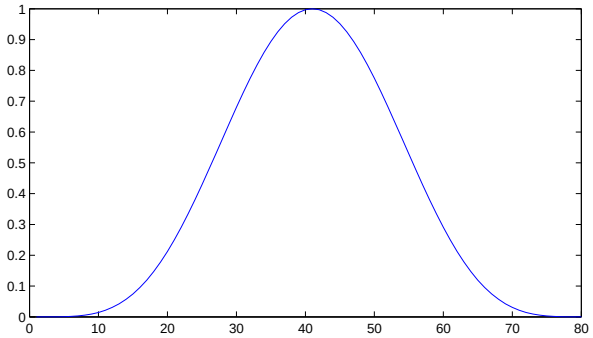

Figure 2.4: Example of a squared raised-cosine pulse

Chapter 3

Code Implementation

Now that we have the equations that define the FDTD method we need to implement the method into computer code. The following will be a brief overview of how to implement the equations in general into any sort of programming language using pseudo code and one-dimensional examples. A full disclosure of the final implementation of the computer code in MATLAB script used to generate the results in this thesis is included in the Appendix. This section can be skipped because the sections that follow are not dependent on any of this information. Along the way the author has included programming tips. This was done because the code contained in the appendix is the evolved form of the basic implementation that is described in this chapter. The tips are helpful things that the author has used, and in some cases thought of using but didn't have the time, and are briefly described in the tips sub-sections.

3.1 Initial conditions and pre-allocation

By examining equations 2.7 to 2.10 it is clear that we have a few constants that need to be defined initially. They are,

- k, the time step or sampling rate. Commonly set to 1.25e-4 seconds
- h, the spatial step or spatial sampling spacing. Commonly set to 0.1 metres
- c, the speed of sound. Commonly set to 344 m/s
- ρ, the density of air. Commonly set to 1.21 kg/m^3.

These constants make up the constant factors found in equations 2.7 to 2.10. Those factors will simply be referred to as F_V and F_P for simplicity. F_V is the factor used in the velocity equations and F_P is the factor used in the pressure equations.

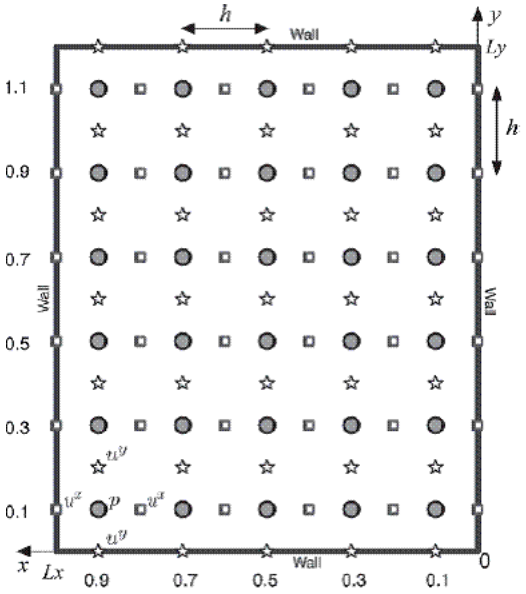

Figure 3.1: Example FDTD grid setup. Circles-pressure points; Stars-particle velocity in the y direction; Squares-particle velocity in the x direction.

The next set of constants that have to be defined are the dimensions of the matrices that will be used. What needs to be kept in mind is that on the boundaries there are velocity matrix points only, thus the velocity matrix will always have one index larger than the pressure matrix. As an example let's assume we wish to define a three dimensional situation with variables M, N, and O for the dimensions. We would allocate space for the pressure matrix as P(M-1, N-1, O-1) because in all three dimensions the pressure matrix has to be one unit smaller. For the velocity matrices only, the dimension that that matrix corresponds to is kept at the full value of the variables. For example the velocity in the x direction is allocated as Ux(M, N-1, O-1), where the x component is kept at M while the y and z components each have a "-1". This point is visually understood by looking at the 2D layout of the grid in figure 3.1[1]. Considering the fact that for now we just want to look at the one dimensional case we set M=50. Given that our value of h is 0.1m then the distance between velocity points is h. This means that if the indices of the matrix start at zero then the size of this room dimension is 5m. In the case of MATLAB the indices start at 1 and thus the size of the room dimension is only 4.9m. It is crucial to be aware of this.

[1]This figure is taken directly from [8].

In general the FDTD method requires knowledge of the previous values when pre-defining the matrices that will be used in the scheme. We need to define two matrices for each acoustic variable we wish to calculate. Thus, we pre-allocate the following matrices;

P_old=zeros(M-1)

P_new=zeros(M-1)

Ux_old=zeros(M)

Ux_new=zeros(M).

The zeros command is used to set the matrices initially as having all of the values inside equal to zero as an initial condition.

3.1.1 Programming Tip #1

Keeping around two full sized arrays for the acoustic pressure and all three of the particle velocity components is very expensive in terms of memory, especially if working with very large arrays. What can be done in MATLAB is the use of recursion to eliminate half of the matrices that need to be stored. For example, using $P = P + F_p * (\mathit{'velocity\,components'})$ calls P to re-calculate itself based on its current values instead of using $P_{new} = P_{old} + F_p * (\mathit{'velocity\,components'})$ which depends on two stored matrices.

3.2 Implementation of equations 2.7 to 2.10

For the moment, let's forget about the boundary conditions and any sources that we need to implement and just focus on the raw calculation of the body of the matrices. Firstly we recall that in the FDTD equations we have components that depend on three separate instances of time (T, T-k/2 and T-k). For the initial condition the components that depend on T-k we can assume to be zero initially because they are earlier versions of the values being calculated at T. For an example let us use the one dimensional version of 2.10 and 2.7;

$$p_{new}(x,T) = p_{old}(x,T-k) + F_P \left[u_x \left(x + \frac{h}{2}, T - \frac{k}{2} \right) - u_x \left(x - \frac{h}{2}, T - \frac{k}{2} \right) \right] \quad (3.1)$$

$$u_{new}(x,T) = u_{old}(x,T-k) + F_V\left[p\left(x+\frac{h}{2},T-\frac{k}{2}\right) - p\left(x-\frac{h}{2},T-\frac{k}{2}\right)\right]. \quad (3.2)$$

The time difference between $T-k$ and T is the interval k. So from one instance of k, to say $k+1$, we need to calculate velocity first (at $k+1/2$) and then pressure afterwards (at k) in one single interval of k (this sounds more confusing than it is). This is the point for using the half step method mentioned in the footnote in subsection 2.1.3. If it were not used, then one would be calculating velocity and pressure in the interval of 2k and not in the interval of k. This led to a lot of confusion to the author as to why the velocity of sound appeared to be c/2 in some early versions of the code. Thus, for the calculation a for-loop is used where inside the loop velocity is calculated first and then pressure afterward, i.e.,

for i=0 to N

calculate Velocity Matrix

calculate Pressure Matrix

end

The two calculations inside this for-loop are the main body of the whole program.

3.3 Pressure and Velocity calculation

Now that we have our main for-loop we need to go about calculating the velocity and pressure matrices. As a matter of preference, the velocity matrix is chosen to be calculated first. The values for the newly calculated velocity are then used immediately afterwards to calculate the pressure matrix. Thus the pseudo code inside the main for-loop should resemble[2];

for i=1 to M

calculate $u_{new}(i,T-\frac{k}{2}) = u_{old}(i,T-\frac{3k}{2}) + F_V\left[p_{old}\left(i+\frac{h}{2},T-k\right) - p_{old}\left(i-\frac{h}{2},T-k\right)\right]$

end

[2]It should be noted that in real code it would be $p_{old}(i+1) - p_{old}(i)$ and not $p_{old}\left(i+\frac{h}{2}\right) - p_{old}\left(i-\frac{h}{2}\right)$ because the u and p matrices are separate from each other, but using h/2 helps illustrate the point that they are the values on either side of the u value being calculated.

for i=1 to M-1

calculate $p_{new}(i,T) = p_{old}(i,T-k) + F_V\left[u_{old}\left(i+\frac{h}{2},T-\frac{k}{2}\right) - u_{old}\left(i-\frac{h}{2},T-\frac{k}{2}\right)\right]$

end

As you can see in the two for-loops we are indeed calculating both u and p in one iteration of the main for-loop and that the pressure matrix only goes out to M-1.

3.3.1 Programming tip #2

For the sake of speeding up the calculation of the simulation program we can leverage the fact that when we calculate the values in the x-component velocity matrix (equation 2.7) they only depend on its previous values and the previous values of the pressure matrix, completely independent of the other components of velocity. Thus, for multi-core CPU systems the three components of the velocity can all be calculated at once instead of one at a time, thus saving computation time. Unfortunately, this cannot be done with the pressure matrix as there is only a single matrix. The way around this is to parse all 4 matrices into separate parts and have each core of the CPU work on one chunk at a time instead. The implementation for the first method is more straight forward (in the author's opinion) with using the function "parfor" in MATLAB.

3.3.2 Programming tip #3

Another technique to increase the speed of calculation is to eliminate the two for-loops by using vectorization. Vectorization is something that is built into MATLAB and can be done in C. Instead of iterating through each individual value one at a time, it can tackle the whole matrix all at once, thus saving a considerable amount of time.

3.3.3 Programming tip #4

This tip may be MATLAB specific. If we note the term in brackets in equation 3.1 we are taking the difference. By using the diff function in MATLAB there is an increase in speed because the function has been optimized in MATLAB. Therefore, whenever possible, use a "built-in" function to do a job for speed gains.

3.4 Boundary Conditions

If we observe the last bit of pseudo code and focus on the velocity calculation we will realize that there is a problem. When the for-loop reaches the value of i=M then the equation wants to calculate using a p at $M + h/2$, which does not exist. Similarly, the same thing happens when we calculate at i=1. This is because those indices represent the boundary and need to be calculated separately by the boundary condition equations for right-hand walls (equation 2.16) and left-hand walls (equation 2.17). Thus the pseudo code changes to adapt to these changes as follows;

for i=2 to M-1

calculate $u_{new}(i, T-\frac{k}{2}) = u_{old}(i, T-\frac{3k}{2}) + F_V \left[p_{old}\left(i+\frac{h}{2}, T-k\right) - p_{old}\left(i-\frac{h}{2}, T-k\right)\right]$

end

calculate $u_{new}(1, T-\frac{k}{2}) = F_{Z1*} u_{old}(1, T-\frac{3k}{2}) - F_{Z2}\left[p_{old}\left(1+\frac{h}{2}, T-k\right)\right]$

calculate $u_{new}(M, T-\frac{k}{2}) = F_{Z1*} u_{old}(M, T-\frac{3k}{2}) + F_{Z2}\left[p_{old}\left(M-\frac{h}{2}, T-k\right)\right]$

In the above pseudo code, F_{Z1} and F_{Z2} are just the constants in equations 2.16 and 2.17. This one dimensional case only has to update the end points. But if we are talking larger dimensions then the last two lines will have to be calculated in their own for-loops that calculate the outer shell of the three dimensional room. Given that the shell of the room is made up of slices then programming tip #3 works very well in this instance.

3.5 Source Terms

In order to add a source term we examine equation 2.21 and notice that it is identical to equation 2.10 with the addition of the function $q(x, T-\frac{k}{2})$ multiplied by a constant. For now, let us denote the q function and its constant by s(t) and its position is at index s. In order to add in the source term we just have to define where we want it in our pressure matrix and have that point be calculated with equation 2.21 instead of 2.10. There is more than one way to do this. If we are using for-loops (what has been shown so far) then it is possible to add a conditional if-statement in the for-loop to switch to equation 2.21. Or we can simply calculate the whole matrix normally and then calculate the source equation immediately afterwards replacing the value that was just calculated. This second method is the preferred method because the first creates significant difficulties for performing vectorization later on. The second

method works better because the pressure at the point where the source is going to be added, is only dependent on its previous value and the particle velocity around it. So as long as the previous pressure at the source has not been thrown away then it can simply be recalculated before the next iteration of the main loop begins. The pseudo code is as follows:

for i=1 to M-1

calculate $p_{new}(i,T) = p_{old}(i,T-k) + F_V\left[u_{old}\left(i+\frac{h}{2},T-\frac{k}{2}\right) - u_{old}\left(i-\frac{h}{2},T-\frac{k}{2}\right)\right]$

end

calculate $p_{new}(s,T) = p_{oldsource}(s,T-k)+s(t)+F_V\left[u_{old}\left(s+\frac{h}{2},T-\frac{k}{2}\right) - u_{old}\left(s-\frac{h}{2},T-\frac{k}{2}\right)\right]$

store $p_{oldsource} = p_{new}(s,T)$.

3.6 L-Shaped Room Creation

In order to talk about L-shaped rooms we need to move to thinking about things in two dimensions briefly. There are two ways to construct an L-shaped room in an FDTD calculation. The first is a brute force way in which the programmer creates a massive square room and then takes a subset of the points in the matrix in the shape of a smaller square and sets those points to be left out of the calculation process. This is accomplished by simply setting up the points around the boundary of the smaller square as boundary points. This method is very easy to program but wastes a lot of memory in the process.

The second and preferred method is to create a separate sub-room and attach it to the main room. This is slightly more complicated than it sounds. The sub-room and the main room have to share the boundary that separates the main room from the sub-room. This shared boundary is a set of particle velocity points that are set up to be calculated as boundary points. Instead they must now be setup to calculate just like the other points that are governed by equation 3.2. But now one of the pressure points has to come from the main room and the other comes from the new sub-room in order to properly connect the two rooms together. This means that along the dimension that is perpendicular to the boundary to which we are attaching the sub-room, the size of the pressure matrix and velocity matrix will be the same size. This differs from the normal case of always having the velocity matrix one dimension larger than the pressure matrix. The following is a one dimensional example of pseudo-code for connecting two linear strips together. The new sub-strip is added at the left hand

side of the original and has N points. Capital letters are used to denote the variables from the sub-room:

for i=2 to M-1

calculate $u_{new}(i, T-\frac{k}{2}) = u_{old}(i, T-\frac{3k}{2})+F_V\left[p_{old}\left(i+\frac{h}{2}, T-k\right) - p_{old}\left(i-\frac{h}{2}, T-k\right)\right]$

end

calculate $u_{new}(M, T-\frac{k}{2}) = F_{Z1*}u_{old}(M, T-\frac{3k}{2}) + F_{Z2}\left[p_{old}\left(M-\frac{h}{2}, T-k\right)\right]$

calculate $u_{new}(1, T-\frac{k}{2}) = u(1, T-\frac{3k}{2}) + F_V\left[p_{old}\left(1, T-k\right) - P_{old}(N, T-k)\right]$

for j=2 to N-1

calculate $U_{new}(j, T-\frac{k}{2}) = U_{old}(j, T-\frac{3k}{2})+F_V\left[P_{old}\left(j+\frac{h}{2}, T-k\right) - P_{old}\left(j-\frac{h}{2}, T-k\right)\right]$

end

calculate $U_{new}(1, T-\frac{k}{2}) = F_{Z1*}U_{old}(1, T-\frac{3k}{2}) - F_{Z2}\left[P_{old}\left(1+\frac{h}{2}, T-k\right)\right]$.

As can be observed, the 5th line is where the two strips are connected to one another and depends on the pressures from both the main room ($p_{old}(1, T-k)$) and the sub-room ($P_{old}(N, T-k)$).

Chapter 4

Boundary Condition Testing

One of the most worked-on areas in acoustic FDTD research has been in modeling the boundary conditions. The work is mainly focused on modeling the boundaries in such a way that they reflect the nature of common acoustic surfaces. Most surfaces have varying acoustic impedances at different frequencies.

Considering the low frequency (20Hz-200Hz) nature of our research, we do not necessarily care about how the absorption co-efficient changes with respect to frequency because it varies little in the frequencies of direct interest. This fact strengthens the justification of our choice of using the boundary condition stated in section 2.3, which upon investigation does not incorporate any obvious knowledge about the frequencies that are impinging on the boundary.

Equation 2.14 gives us a relationship for the Characteristic Wall Impedance which is what we use to approximate our boundary condition. But before going any further, it should be noted how it can be obtained from the acoustic theory. We turn to section 6.3 in [5] entitled "Reflection from a locally reacting surface". A locally reacting surface is a surface whose various parts are not strongly coupled together, thus the motion normal to a certain portion of the surface is dependent only on the acoustic pressure incident on that portion, and independent of all other motions occurring elsewhere in the surface. At the boundary for a situation like this, the impedance is equal to $(p/-u_y)$[1] where;

$$\begin{aligned} p(x,0) &= -\rho c(1-C_r)\psi_i'(x\,sin\vartheta - ct) \\ u_y(x,0) &= cos\theta(1-C_r)\psi_i'(x\,sin\vartheta - ct) \end{aligned}$$

[1]Equation2.13 applies to both RHW and LHW. The negative sign appears when the normal vector is properly taken into account.

Thus $p/-u_y = Z$ gives us,

$$Z = \frac{\rho c(1+C_r)}{cos\theta(1-C_r)} \tag{4.1}$$

The variable C_r is defined in [5] as the reflection coefficient, which is the ratio between incident and reflected pressure waves. If we consider normal incidence then cos(0)=1 and equation 4.1 looks identical to equation 2.14, given that $C_r = \sqrt{1-\alpha}$, which is the common definition.

4.1 Direct Incidence Testing

In testing to determine whether or not the simulation program's boundary condition is sufficient we need to test what happens to a pressure pulse before and after it hits the boundary. The factor C_r is defined as being the ratio between incident and reflected pressure waves. The theory for Z and how it relates to C_r is derived right at the boundary, which presents a problem for measuring this phenomenon. If data is taken right against the wall as the pressure pulse hits, then separating the waves from the data may be possible but difficult. Another way to look at the problem is to consider the wave before, and after, it encounters the boundary. This is acceptable, because the reflected wave will travel through the medium but with a reduced amplitude factor given by C_r. Given that the simulator uses a point source to produce a pulse then the waves will be spherical in nature. Thus, as a wave is sent at the wall, the source sends another identical wave into the open room. This second wave can be measured at a distance equal to the path length that the reflected wave travels to be measured at the virtual mic. Thus, the second wave that has been unhindered by any boundary shall be thought of as being a reference to the incident wave that corresponds to the reflected wave having traveled the same distance. Therefore we can measure how much absorption occurred by measuring α from the pressure amplitudes of the two waves in the following formula.

$$\alpha = 1 - C_r^2 = 1 - \frac{P_r^2}{P_i^2}. \tag{4.2}$$

The results are tabulated in table 4.1. The conditions for this data are that a 10cm grid and a 7.5msec pulse were used.

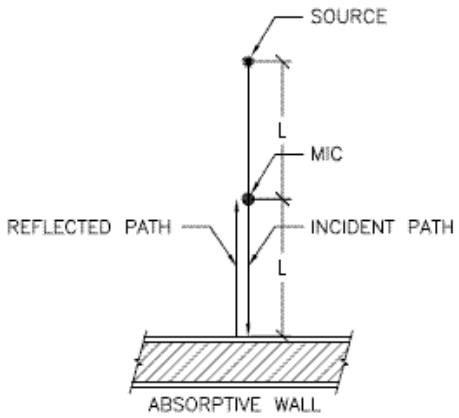

Figure 4.1: Diagram depicting the mic and source placement for measuring the amount of absorption on the boundary.

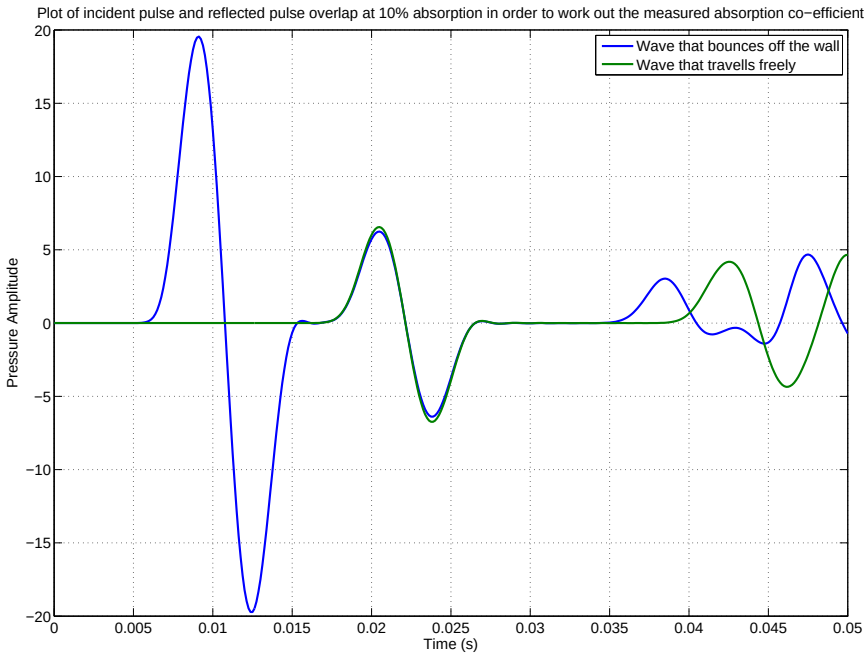

Figure 4.2: Example providing data for calculating the absorption coefficient.

Absorption Co-efficient	P_r^2/P_i^2
.01	.0202
.1	.1001
.2	.1897
.3	.2806
.4	.3727
.5	.4663
.6	.5616
.7	.65910
.8	.7596
.9	.86480

Table 4.1: Table showing absorption coefficient data.

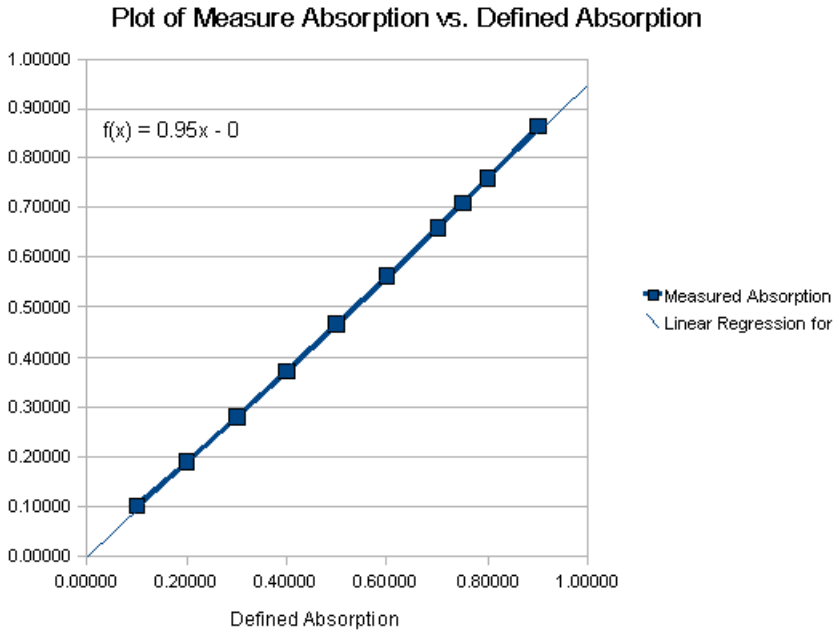

Figure 4.3: Plot of data in table 4.1.

As can be seen from the plotted data, we get a line whose slope is close to unity and has a very small intercept, which is expected from a plot of α vs. P_r^2/P_i^2. Thus, it can be concluded that for direct incidence the boundary condition is functioning

correctly. Smaller grid spacings seem to make the resultant slope tend closer to unity, but we speculate that the slope is not exactly unity because the reflected wave is not plane. The spherical character changes the absorption at different portions of the wave surface.

4.2 Angular Absorption

We have now seen that the boundary condition is a reasonable approximation to the theory posed in [5] for the case of zero angle incidence. Considering that the angle isn't directly factored into the derivation of the boundary condition we should now try and observe how well the physics is preserved off axis.

For this experiment, the same technique as for direct incidence is applied again, except now the mic that measures the reflected pulse will be the same distance away from the wall as the source, but now it is a known distance to the left of the source in the room. The mic that measures the incident pulse is placed a distance away that is equal to the path length of the reflected wave away from the source, as depicted in figure 4.4.

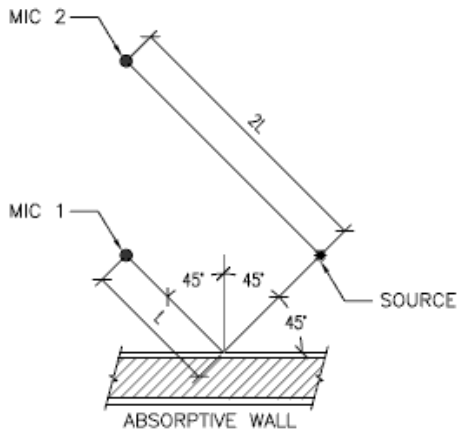

Figure 4.4: Diagram depicting the mic and source placement for measuring the amount of absorption on the boundary at various angles.

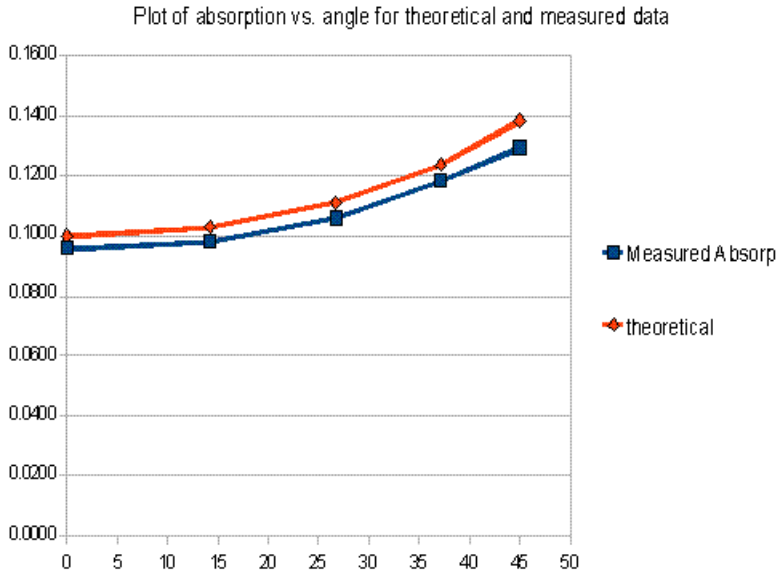

Figure 4.5: Example plot showing trend of how the amount of absorption changes with the angle in the simulation relative to the theory.

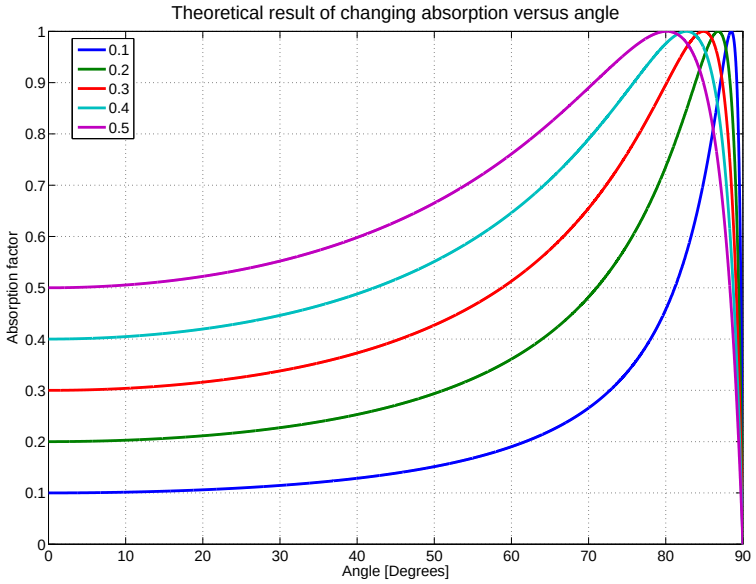

Figure 4.6: Figure depicting the theoretical change in the amount of absorption as a function of angle for various levels of the value alpha.

Alpha=.1			
Angle	Measured Absorption	Theoretical Absorption	%difference
0	.0958	.1000	4.20
14.17	.0982	.1030	4.66
26.8	.1060	.1113	4.76
37.15	.1185	.1238	4.28
45	.1293	.1385	6.64
Alpha=.2			
Angle	Measured Absorption	Theoretical Absorption	%difference
0	.1904	.2000	4.80
14.17	.1963	.2056	4.52
26.8	.2114	.2212	4.43
37.15	.2334	.2443	4.46
45	.2564	.2709	5.35
Alpha=.5			
Angle	Measured Absorption	Theoretical Absorption	%difference
0	.4787	.5000	4.26
14.17	.4925	.5110	3.62
26.8	.5219	.5409	3.51
37.15	.5640	.5830	3.26
45	.6075	.6285	3.34

Table 4.2: Data for changing absorption as the angle theta is changed.

Given equation 4.1 and the relationship $C_r = \sqrt{1-\alpha}$ we can calculate alpha as,

$$\alpha = 1 - \left(\frac{Z\,cos\theta - \rho c}{Z\,cos\theta + \rho c}\right)^2. \tag{4.3}$$

Thus, if we want to investigate theoretically how alpha varies with angle we choose an alpha and input it into equation 2.14 to first obtain a value for Z. We then put it into equation 4.3 in order to compute values for alpha as theta is changing[2]. A sample of some of the data taken, is in table 4.2. From table 4.2 it is clear that the measured absorption in the simulated room mimics the trend given by equation 4.3 and observed in figure 4.5. The theoretical calculation for all angles is shown

[2]As a check it can be seen that for theta equal to zero gives back the original value of alpha that was used to compute Z from equation 2.14

in figure 4.6. We can conclude that even though the angular dependence of the absorption coefficient was not directly incorporated in the derivation, the physics is still preserved. This is because even though the angle of the wave is unknown, the proportion of the perpendicular velocity component is known and its change in size relative to the other components is what allows the program to obey the physics in equation 4.1.

Chapter 5

Rectangular Room analysis

The most popular of all room shapes is the rectangular room.

5.1 The Groh Room

The paper by Groh [1] illustrates how adjusting the position of a speaker changes the response at the listening position. The dimensions of the room are Height=2.4m, Width=4.2m, and Length=5.6m. The author will use the term "Groh Room" throughout this thesis to refer to the room dimensions given.

5.1.1 Common Acoustic Room Measures

In order to better assess whether or not our program is properly simulating the Groh Room we need to asses the expected acoustic properties of such a room based on the common acoustic equations that are used to assess rectangular rooms. The following equations were chosen based on their use in the paper by Siegfried Linkwitz [28].

The first equation is the reverberation time. It is the most well known and least disputed measure in room acoustics. This makes it an important measure to help us indicate whether or not our simulated room resembles an actual listening room or not. The reverberation time in [29] says "The reverberation time T is defined as the time required for the sound energy density to decay 60dB, that is, to 10^{-6} of its original value." In most cases the reverberation time is denoted as T_{60}, and is computed via the following formula[1].

[1]For completeness it should be pointed out that in general $\alpha S = \Sigma \alpha_i S_i$, but this is not used because for simplicity we make all the $\alpha's$ the same.

$$T_{60} = \frac{60V}{1.085c\alpha S}, \tag{5.1}$$

where V is the volume of the room, c is the velocity of sound in air, α is the average absorption coefficient in the room, and S is the sum of all the surface areas in the room. The T_{60} measurement will be focused on more intensely later on in the chapter.

The next equation is called the Rise Time and is directly related to the reverberation time. Unlike the reverberation time the rise time describes the amount of time it takes for a room to go from 10% to 90% of its steady state sound level when being excited by a sound source. The equation is given by,

$$T_{rise} = 0.32\, T_{60}. \tag{5.2}$$

The following formula is one that describes the various eigenfrequencies, or resonance modes, which can occur in rectangular shaped rooms and is simply known as the Eigen-Frequency formula,

$$f = \frac{\omega}{2\pi} = \frac{c}{2}\sqrt{\left(\frac{n_x}{l_x}\right)^2 + \left(\frac{n_y}{l_y}\right)^2 + \left(\frac{n_z}{l_z}\right)^2} \tag{5.3}$$

where $n_{x,y,z}$ are the room indices and $l_{x,y,z}$ are the room dimensions.

The spacing between resonance modes in a room is initially irregular, as seen in table 5.2, but the spacing between these modes gets smaller and smaller as the frequency increases. The equation that is used to estimate the number of modes of vibration below an upper frequency limit, say $f_m = 150Hz$, is given by,

$$N = \frac{4\pi}{3}V\left(\frac{f_m}{c}\right)^3 + \frac{\pi}{4}S\left(\frac{f_m}{c}\right)^2 + \frac{1}{8}L_e\left(\frac{f_m}{c}\right) \tag{5.4}$$

where

$$\begin{aligned} V &= Volume = LWH\,[m^3] \\ S &= Surface\,Area = 2(LW + LH + WH)\,[m^2] \\ L_e &= Length\,of\,Edges = 4(L + W + H)\,[m] \end{aligned}$$

and the average separation, df, between the resonance frequencies at f_m is given by,

$$df = \frac{c^3}{4\pi V f_m^2}\,[Hz]. \tag{5.5}$$

The estimation of the bandwidth, BW, for the resonance frequencies is based on the calculation for the reverberation time and is given by,

$$BW = 2.2/T_{60} = 0.7/T_{rise}\,[Hz]. \tag{5.6}$$

The last acoustic measure that is used in table 5.1 is called the Schroeder Cutoff Frequency. I personally feel that its explanation given in [32] on pages 293-294, best describes this measure and is quoted as follows:

> When the resonance peaks are closer together than the bandwidth associated with any one peak, the resonances are less evident. If the average spacing $(\triangle f)_{mode}$ between peaks is of the order of or less than, say, $\frac{1}{3}(\triangle f)_{res}$, the resonance peaks may be regarded as a smoothed-out continuum. Since the average spacing $(\triangle f)_{mode}$ decreases with increasing frequency, there is a frequency f_{sch} (Schroeder cutoff frequency) below which $(\triangle f)_{res} > 3(\triangle f)_{mode}$ is not satisfied and above which it is. This frequency is identified as,

$$f_{sch} = \left(\frac{c^3}{4ln10}\right)^{1/2}\left(\frac{T_{60}}{V}\right)^{1/2}. \tag{5.7}$$

The important point to understand from this equation is that it is simply telling us below what frequency the modes are not significantly overlapping one another. This is important because of how we hear. The more overlap in the mode the less our brain distinguishes that a resonance is present. Since there is very little overlap between the modes in the bass region below f_{sch} then our brains easily distinguish one mode from another. If c=340m/s then the equation reduces to,

$$f_{sch} \backsimeq 2000\left(\frac{T_{60}}{V}\right)^{1/2}. \tag{5.8}$$

For the Groh Room with the common parameters that are assumed, the frequency $f_{sch}$is about 260Hz. Given that the room is reverberant up to this point then it would be ideal to equalize up to that frequency. But it should be noted that the frequencies below 200Hz (depending on the room) are in general the most troublesome.

Most of these measures will not be examined in great detail but it is worth noting that most, in some way, depend on the reverberation time, thus that measure will be looked at more closely in section 5.3.

Table 5.1 describes the common acoustic attributes associated with a room with these dimensions.

Groh Room Calculations		
Assumed Average Wall Absorption=	.1	
Room Dimensions in metres	L=	5.6
	W=	4.2
	H=	2.4
Speed of sound c (m/s)=	343	
Floor area A(m^2)=	23.52	
Volume V(m^3)=	56.45	
Surface area S(m^2)=	94.08	
Edge Length Le(m)=	48.8	
Below Frequency fm(Hz)	150	
Total number of modes N=	36.57	
Avg. Mode spacing df(@fm)=	2.53Hz	
Estimated reverberation time T60(s)=	.98	
Resonance bandwidth BW(Hz)=	2.25	
Rise time Trise(s)	.31	
Schroeder frequency f_{sch}(Hz)=	263.5	

Table 5.1: Table displaying the common acoustic measures for the Groh Room.

Modes in Groh Room (sorted by Freq.)	n_x	n_y	n_z	freq[Hz]
1	1	0	0	30.63
2	0	1	0	40.83
3	1	1	0	51.04
4	2	0	0	61.25
5	0	0	1	71.46
6	2	1	0	73.61
7	1	0	1	77.74
8	0	2	0	81.67
9	0	1	1	82.3
10	1	1	1	87.82
11	3	0	0	91.88
12	3	1	0	100.54
13	2	1	1	102.59
14	0	2	1	108.52
15	1	2	1	112.75
16	4	0	0	122.5
17	0	3	0	122.5
18	3	1	1	123.35
19	4	1	0	129.13
20	0	3	1	141.82
21	0	0	2	142.92
22	1	3	1	145.09
23	1	0	2	146.16
24	4	1	1	147.58
25	1	1	2	151.76
26	5	0	0	153.13
27	5	1	0	158.48
28	0	4	0	163.33
29	5	1	1	173.84
30	0	4	1	178.28
31	1	2	1	180.89
32	6	0	0	183.75
33	6	1	0	188.23
34	6	1	1	201.34

Table 5.2: Table of modal frequencies in the Groh Room.

5.2 Verification of Room Size

Given that our simulation method is based on a number of assumptions and approximations it is important to know that when we specify our room dimensions that what we put in is what we get out again. The following are two ways in which we test the size of our simulated room.

5.2.1 Frequency Response Method

In rectangular rooms it is possible to use the following formula to determine all the possible resonance frequencies that are potentially present in a listening room.

$$f = \frac{\omega}{2\pi} = \frac{c}{2}\sqrt{\left(\frac{n_x}{l_x}\right)^2 + \left(\frac{n_y}{l_y}\right)^2 + \left(\frac{n_z}{l_z}\right)^2} \tag{5.9}$$

The $n_{x,y,z}$ are the room modes and $l_{x,y,z}$ are the room dimensions. In rectangular listening rooms it is possible to excite all possible room modes, as calculated by 5.9, by placing a sub-woofer in the corner of the room. By placing a source term in a corner of our pressure matrix, say at (1,1,1), it is possible to observe this effect of exciting all possible room modes. Based on the nature of our grid, and the fact that the source term is a pressure point, then choosing (1,1,1) does not exactly place the source in the corner because the velocity points are on the boundaries and not the pressure points, but with no actual point in the corner. Thus, the true distance from the assumed corner will be dependent on the grid spacing and equal to $\frac{\sqrt{3}h}{2}$. For example if h=10cm then the distance from the source to the corner will be 8.66cm.

As example we shall use the Groh Room. The lowest possible resonance mode will correspond to the largest room dimension, l_x=5.6m. Given $n_x = 1$, and $n_y = n_z = 0$, or n(1,0,0) mode, then 5.9 reduces to $f_{1,0,0} = \frac{c}{2}\frac{1}{l_x}$, thus resulting in $f_{1,0,0} = 30.71Hz$.

In this test of the FDTD program we use the following parameters,

- pulse length of 10msec
- absorption coefficients on all surfaces set to 10%
- h=10cm and 5cm with K=1.25×10^{-4}s (fs=8000Hz) and $6.25\times10^{-5}s$ (fs=16000Hz) respectively
- 1second of data to be collected.
- source placement at (1,1,1)

- mic placement at (M,N,O), where M,N, and O correspond to the maximum dimensions of the pressure matrix, putting the mic in the opposite corner relative to the source.

The reason for taking data with h=10cm and 5cm is to see whether or not our results change when the size of the grid spacing is made smaller. In general making the grid spacing smaller suggests achieving greater accuracy in the FDTD calculations. Below in figure 5.1 we see the raw pressure data taken by our microphone. It should be pointed out that the blue curve is almost completely covered by the green curve showing how little difference there is.

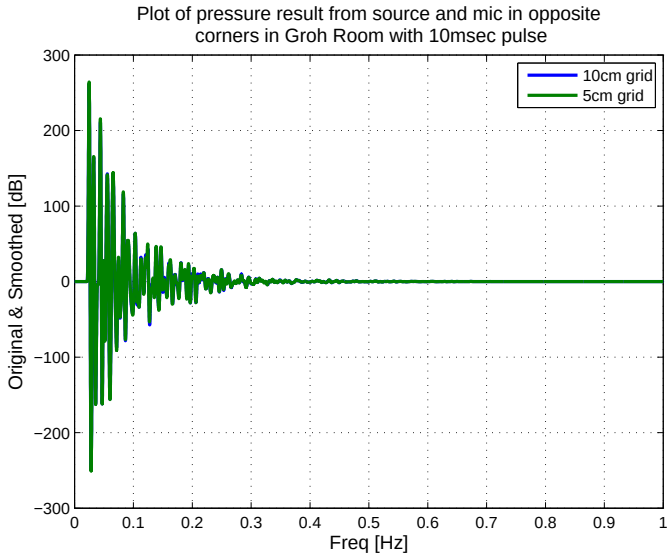

Figure 5.1: Raw pressure data showing that the 10cm and 5cm grid curves lie practically on top of one another. Note, the horizontal axis should be time (s).

In order to obtain greater accuracy in determining the frequency of the lowest frequency mode from taking the FFT of our data, we need to consider zero-packing our data. It should be noted that the data has decayed to zero in 1 second, thus it is possible to use zero-packing, without the need for windowing. Simply taking the number of points, n, obtained from the data and multiplying by a factor of 4 yields acceptable results[2] as plotted in figure 5.2.

[2]The use of factors larger than 4 seem to yield no new information, which is why we stopped at 4.

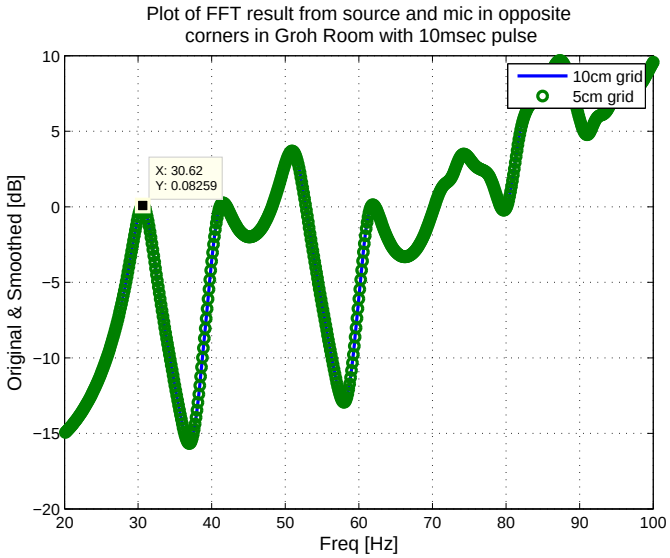

Figure 5.2: fft output of raw pressure data from figure 5.1.

The location of the lowest frequency mode given by the plot is found to be 30.62Hz. In order to be more accurate the five values about the peak are taken and a squared least squares fit[3], is applied to those points. From the resulting polynomial the frequency of the first peak is obtained. For the 10cm grid we obtain $f_{1,0,0} = 30.6314$Hz and for the 5cm grid we obtain $f_{1,0,0} = 30.6340$Hz. These values correspond to l_x values of 5.6151m and 5.6146m which differs from the inputted value of 5.60m by less than 2cm. This corresponds to a percent difference of 0.31%. This is certainly an acceptable degree of error. But when comparing the difference to the size of the grid spacing we see that the difference for the 10cm grid is 15.1% of a grid spacing and for the 5cm grid is 29.2% of a grid spacing. This result is clearly not acceptable.

In order to obtain a more accurate result it is necessary to lower the value of the absorption coefficient to 1% (which also creates the need to take a longer data set). This creates an incredibly reverberant environment which will sharpen the resonance peaks and is observed in figure 5.3.

From figure 5.3 we can certainly see that the room is now very reverberant and the resonance peaks have indeed been sharpened. For the 10cm grid we obtain $f_{1,0,0} = 30.7101$Hz and for the 5cm grid we obtain $f_{1,0,0} = 30.7126$Hz. These values correspond to l_x values of 5.6007m and 5.6003m which differs from the inputted value of 5.60m by less than 0.1cm. This corresponds to a percent difference of less than 0.018%.

[3]for the least squares fitting the software package known as Maple was used here

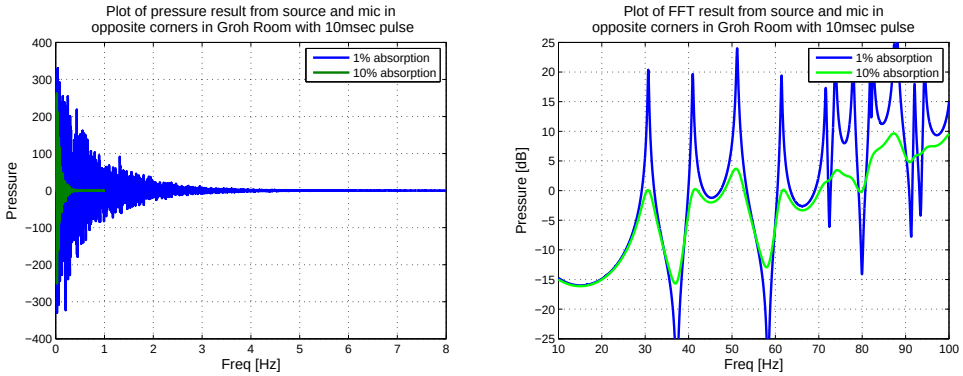

Figure 5.3: Comparison of data taken in the Groh room at 10% and 1% absorption. Note, the horizontal axis on the left hand plot should be time (s).

This is certainly an acceptable degree of error. When comparing to the difference of the size of the grid spacing we see that the difference for the 10cm grid has become 0.7% of a grid spacing and for the 5cm grid is 0.6% of a grid spacing. This result is now exactly what we expected.

This experiment is repeated for two additional, different sized rooms. The first one with dimensions[4] of $l_x = 6.30m$, $l_y = 3.50m$, and $l_z = 2.70m$. The second room has dimensions that are double that of the Groh Room. ($l_x = 11.20m$, $l_y = 8.40m$ and $l_z = 4.80m$). The results are tabulated in the following table.

[4]Theses room dimensions are as an example and are similar to the dimensions of room P374 in the physics building at the University of Waterloo.

Table 5.3: Tabulated Data for the three different room sizes and their measured $f_{1,0,0}$ frequency.

Room Dimensions(m)	Grid Spacing	absorption	Expected Frequency(Hz)	Measured Frequency	%difference
5.6x4.2x2.4	10cm	10%	30.7143	30.6314	.2699
	10cm	1%		30.7102	.0133
	5cm	10%		30.6340	.2614
	5cm	1%		30.7126	.0055
6.3x3.5x2.7	10cm	1%	27.3016	27.2990	.0095
	5cm	1%		27.3007	.0033
11.2x8.4x4.8	10cm	1%	15.3571	15.3562	.0059
	5cm	1%		15.3564	.0046

Given that the results differ by fractions of a percent from what the expected $f_{1,0,0}$ value should be, it can be said that the FDTD simulator is accurately simulating rooms that are the same size as what are being specified.

5.3 Reverberation Time

From table 5.1 we can see that the calculated reverberation time is .98 seconds for the Groh Room given an assumed absorption of 0.1 and the speed of sound being 344m/s. The question now remains, "How do we measure the reverberation time in the simulated room?". Thankfully the FDTD method gives us pressure data in the time domain instantly and from that we can easily work out the reverberation time using two different methods.

5.3.1 Beranek Method

The Beranek Method is one that is adapted from the book Acoustical Measurements by Leo L. Beranek [29]. In this method a room is excited by a band limited excitation signal that excites the room to its steady state level. The sound source is then removed and the sound level then begins to decay in the room. The time it takes for the signal to decay 60dB from its steady state is thus the reverberation time we are looking for. In the simulation we don't worry about band-limiting our sound source because it has been shown in figure 2.2 that the 20msec pulse has very little frequency content above 100Hz, which will be used instead of a 10msec pulse. What can be done is to

just excite the room with one cosine pulse and then just observe the decay in pressure in the room and extrapolate the rate from that. An example of the raw pressure data from the Groh Room being excited by a sub-woofer in the corner is presented in figure 5.4.

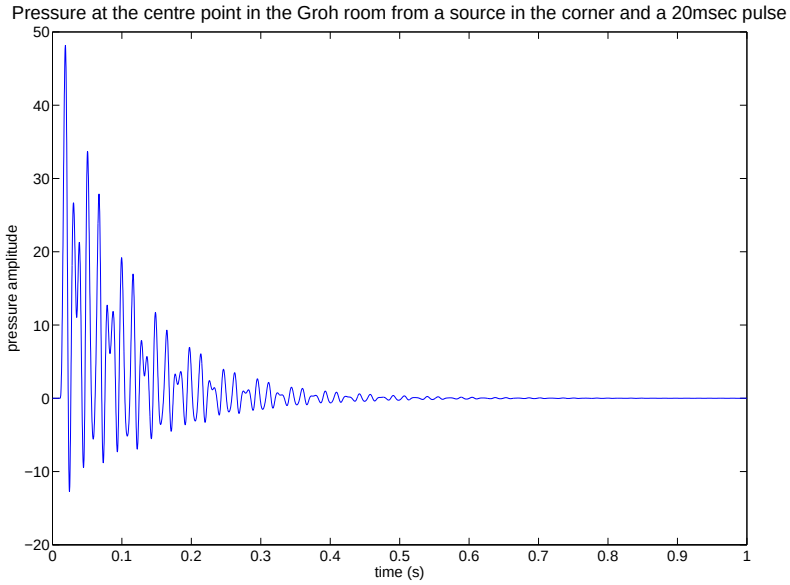

Figure 5.4: Raw pressure data taken from a mic in the centre of the Groh Room after being excited by a 10msec source in the corner.

From the raw pressure data in figure 5.4 there is an observable envelope of an exponential decay as time evolves. In order to properly measure this decay we need to plot $10log_{10}(Pressure^2)$ versus time in order to obtain the decay rate. This data is plotted in the left-hand plot of figure 5.5. The idea would be to apply a linear fit to this data and extrapolate the reverberation time from this. Because of the numerous sharp dips in the data the result of a linear fit is skewed. In order to obtain a better result from this data only the peak values are taken and used for the linear fit instead of all the raw data, and is shown in the right of figure 5.5. This omission of some of the data is justified in light of the fact that it is the envelope of the decay that is of importance and not the points that are oscillating about zero. The results from the linear fit is an equation for the line as $time = -80.295 * (dBlevel) + 27.819$. By taking 60dB and dividing by the slope the reverberation time is found to be 0.747 seconds.

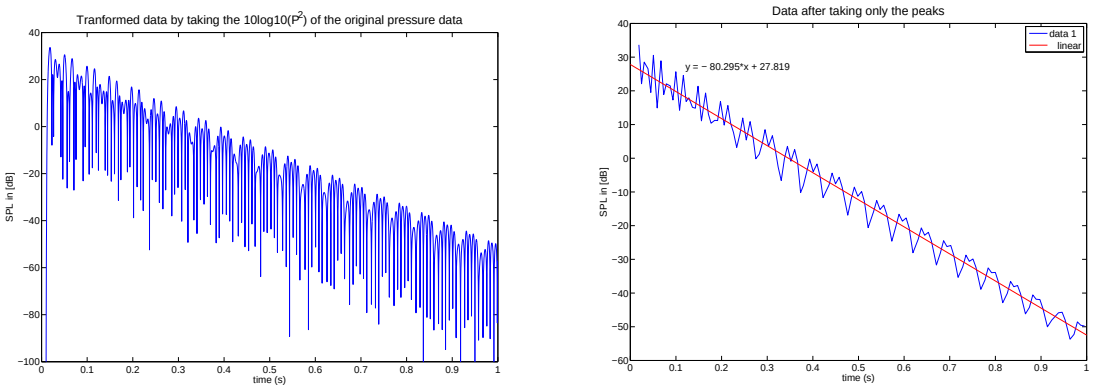

Figure 5.5: Figures depicting the transformed pressure data ($10log_{10}(Pressure^2)$). Left: Transformed data. Right: Plot of only the peaks taken from the transformed data.

5.3.2 Schroeder Method

The Schroeder Method was developed by M. R. Schroeder and presented in his paper "New method for measuring reverberation time" in 1965 [30], and then explored in [31]. This method works out the reverberation time by using reverse integration over the impulse response. We can use the results in the room from the 20msec, squared raised-cosine pulse because its frequency response is very similar to that of a low pass filtered impulse and in the time domain is essentially a pulse. The response from our band-limited impulse will yield an acceptable impulse response (h_n), to be used for the Schroeder Method given below.

$$\left\langle s^2_{N,T}(t) \right\rangle = N \int_t^T h^2_N(\tau) d\tau \tag{5.10}$$

A plot of the result $\langle s^2 \rangle$ of this reverse integration applied to the same data shown in figure 5.4 is shown in figure 5.6. The reverberation time calculated from the slope of the Schroeder decay plot is 0.736 seconds.

5.3.3 Discussion

In light of the closeness in the results from both methods for determining the reverberation time (Beranek=0.747s and Schroeder=0.736s) we can safely say that the reverberation time in the Groh room with alpha=0.1 on all walls is approximately

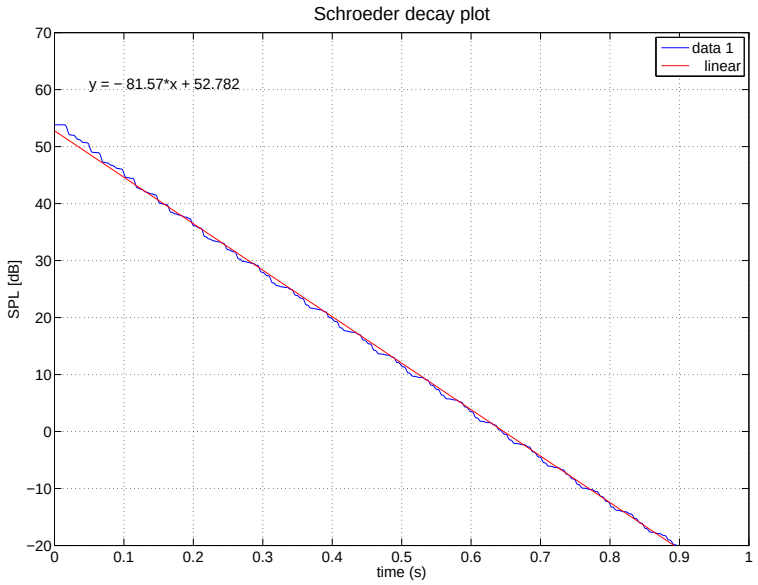

Figure 5.6: Plot of the Schroeder decay plot from a 20msec pulse in the Groh room with a source in the corner.

0.74 seconds and that either method is appropriate for determining the reverberation time in the simulation. A difference of 0.1 seconds in reverberation time is not very significant, thus the values for the Beranek and Shroeder methods are essentially in agreement.

But there is still an issue. The expected reverberation time in the Groh room with alpha=0.1 as computed from equation 5.1 and shown in table 5.1 is 0.98 seconds and not 0.74 seconds. This clearly needs to be explained. If we return to section 4.2 we recall that it has been shown that the absorption increases as a function of angle. Given that the source is a spherical source the waves come out and bounce all around the room at all available angles. The case of direct incidence is in fact a rare occurrence, and it is more likely that most reflections would have some sort of angular component. Therefore it can be said that the average absorption in the room will be larger than alpha=0.1. In order to get an idea of what the average absorption is we take our 0.74 reverberation time and put it back into equation 5.1 and solve for alpha. We obtain the value 0.1363, which from figure 4.5 doesn't seem like an unreasonable result.

It is possible to use equation 4.3 and perform an integral over the solid angle to find the average absorption value. If the rays are assumed to be coming in equally

Perpendicular Alpha Value	Avg. Absorption
0.1	0.2839
0.2	0.4453
0.3	0.5652
0.4	0.6591
0.5	0.7334
0.6	0.7913
0.7	0.8339
0.8	0.8599
0.9	0.8634
1.0	0.7734

Table 5.4: Table depicting results of integration over solid angle to determine the average absorption value.

from all directions a calculation (performed by JV) shows that the effective α should be 0.284 (table 5.4 shows the values from this calculation for other alpha values). This value is much higher than what we are seeing due to the fact that each angle is weighted equally, and in a realistic situation some angles may be much more common than others. This is probably due to the fact that the modes in the room are not very uniform at these lower frequencies. Suffice it to say, the problem is a very complex one. It should also be noted that the value of the Schroeder frequency also becomes lower now with alpha equal to 0.1363.

Chapter 6

Plane Wave Creation

In undergraduate electrodynamics, students learn that when an infinite line of point charges is equally spaced, the electric field a distance away from these point charges becomes very similar to the electric field of a continuous line of charge the further one moves away. The same thing applies in 3 dimensions. If we have an infinite number of charges, all equally spaced out along a plane, then the further away one travels from this plane the more the electric field looks like a sheet of charge. This effect is also true in acoustics. The interesting difference between the field away from a spherical point source and that of a plane of sources is that the plane of sources creates a plane wave that does not decay as 1/r and does not depend on A(t)=dQ/dt. Instead, the plane wave will maintain the same shape as the function Q(t) that is inputted into the source, and will not decay as 1/r.

6.1 Single Source Plane Wave

In our simulation we are dealing with a finite space and can use the image method to observe a single source creating a plane wave. When a source is placed in a room with reflecting walls, then the reflections appear to an observer as individual image sources that in turn have reflections of their own creating an infinite number of sources. If the source is placed directly on one of the reflecting boundaries, then it will appear that there are only image sources directly in the plane of that boundary. If the boundary is a square wall, then placing the source in the centre will make it such that the distance between the source and all of its images are equal to each other and equal to the dimension of that square wall. It has been shown in [5] that when there is an array of sources has seperation distance d, then there is a response change or equalization effect below the frequency given by,

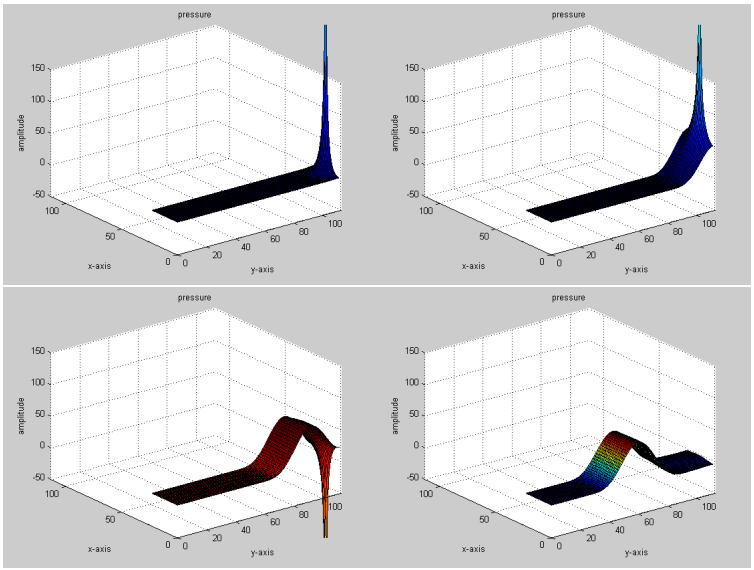

Figure 6.1: Series of plots showing the time evolution of the point source emitting a plane wave.

$$f_{max} = \frac{c}{d} - \triangle\epsilon \tag{6.1}$$

where c is the speed of sound and $\triangle\epsilon$[1] is a frequency shift that depends on the damping of the room. The equalization only affects the standing waves that setup parallel to the plane the source is in and the longitudinal standing waves that travel down the length of the room will be relatively unaffected. Actually they may be re-enforced and are observed in section 7.2.

As an experiment to test our source term and how it couples to the room to create a plane wave, we can set up an extremely long room with a relatively small square wall at either end and measure the pulse at the source and at a point in the centre of the room to observe the change in the shape of the pressure wave. The front wall where the source will be placed is a 2.4m square and the length of the room is 28m. Since the dimension of the wall is 2.4m then $f_{max} \simeq 143Hz$. In light of this value for f_{max} the pulse that will be used is the 20msec squared raised-cosine pulse because its frequency content is primarily below 140Hz. Also an alpha=0.1 was used.

It is observed in figure 6.1 and 6.2 that the pressure at the source is indeed proportional to dQ/dt as we have seen before and that the pressure does in fact

[1]it is not made clear in [5] how to calculate this value.

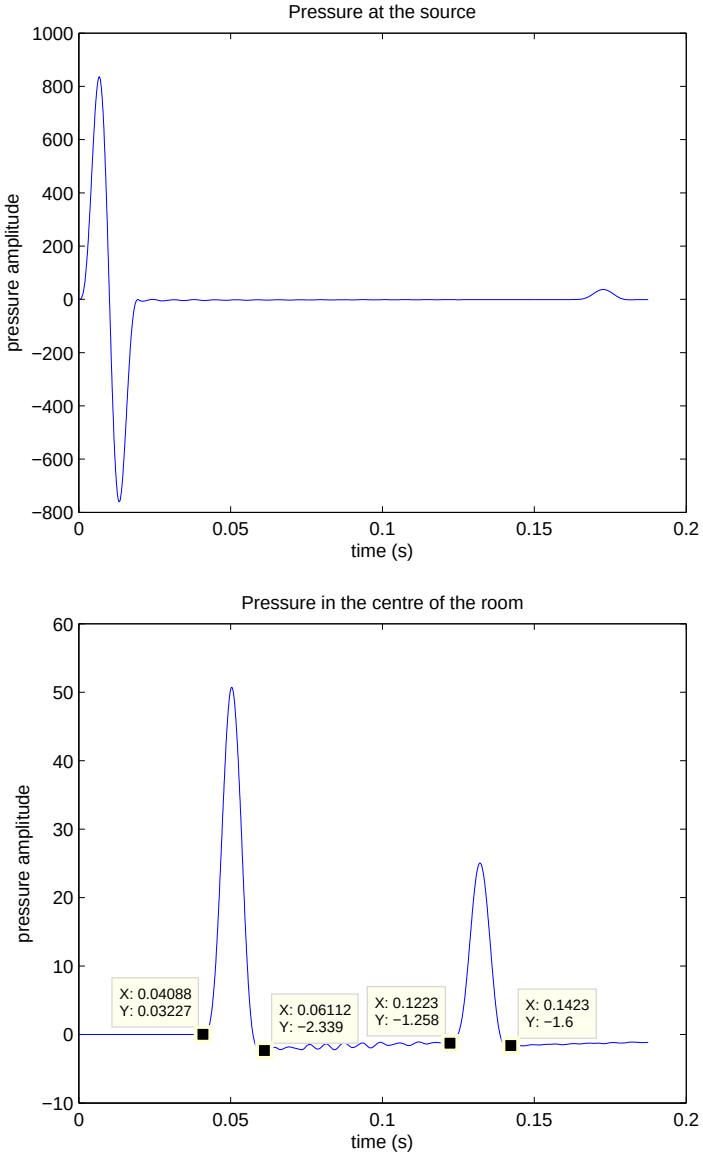

Figure 6.2: Plots depicting the change of the spherical source pressure into a plane wave. Top: Pressure at the source; Bottom: Pressure measured in the centre of the room showing incident and reflected pulse.

become a plane wave when observed away from the source in the centre of the room. It is also interesting to point out, that the length of the pulses measured in the centre of the room (the first pulse is an incident pulse and the second is a reflected pulse from the back wall) are in fact 20msec which is exactly what was put into the room. Thus, it has been shown that a plane wave was created from our source model and gave the expected result. The characteristic of the plane wave that was not observed, was how much it was decaying as it travelled, but regardless does not cast doubt on the fact that what was observed is indeed a plane wave. There was absorption in the room and it should be clear from the bottom plot in figure 6.2 that the pressure does not depend on 1/r.

6.2 Plane Wave In a Realistic Room

The idea behind suppressing a particular mode of vibration, is to move the speaker towards a node for that mode of vibration. This simply prevents that mode from establishing itself in the room because there is no coupling to the mode. This was the whole idea behind Groh's paper [1]. In the previous subsection, it was shown that a plane wave could be established by placing a source in the middle of the front wall. This made the distance between the source and all the images the same so that the lower-frequency tangential modes would not be established and only the longitudinal modes are excited, creating a plane wave. This worked well for one source with frequencies below 140Hz because the distance between source and image sources was small (see equation 6.1) due to the dimensions of the front wall. The larger the dimensions of the wall, according to equation 6.1, the lower the f_{max} is. This means that the frequency limit at which the tangential waves are suppressed decreases. In order to raise up f_{max} in the case of larger wall dimensions it has been shown in [5] that additional speakers can be placed on the front wall.

For practical reasons the work of Celestinos in [6, 7, 8, 9, 10] used only two sources on the front wall. The speakers were placed half way up the height of the wall and moved in by a quarter distance in from the side walls. This positioning maintains equal distance between sources and image sources in the y-direction. However, the distances may differ in the x and z directions if the width of the room is not exactly twice the size of the height[2]. In the case of the Groh Room the front wall dimensions do not form a perfect square and the width is not exactly twice the height. Given the placement of the speakers equation 6.1 needs to be adjusted to read

[2]Having the width = 2(height) is the only condition that would allow a two speaker set up to have the images and sources equally spaced in the x and z directions.

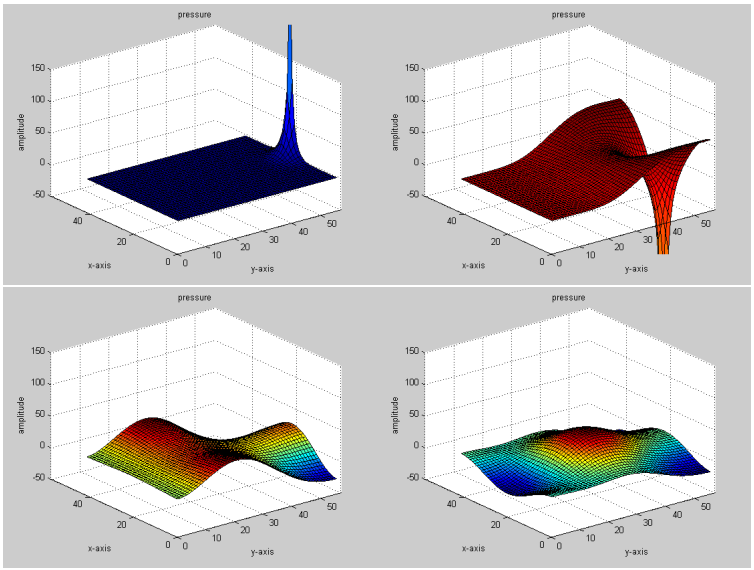

Figure 6.3: Series of plots showing the time evolution of the point source in the centre of the front wall displaying that a clean plane wave is not created in the Groh Room using 20msec pulses.

$$f_{max} = \frac{c}{max\left\{\frac{W}{2}, H\right\}} - \triangle\epsilon \tag{6.2}$$

and in the case of the Groh Room H is the larger of the two and again $f_{max} = 143Hz$. Suppressing the tangential modes with this speaker setup leaves the longitudinal modes unaffected. The sources are thus placed in the Groh Room at (L,W/4,H/2) and (L,3W/4,H/2) and the results are observed in figures 7.7 to 7.11 in section 7.2. This placement and the creation of plane waves below f_{max} is an essential element for the cancellation process seen in section 7.3.

Figures 6.3 shows a single sub-woofer generating a complicated wave pattern that is not quite a plane wave. The exact same source is used for the room of figure 6.1, which then creates a plane wave since that room is much narrower. Figure 6.4 shows the same Groh Room as figure 6.3, but now with two sub-woofers optimally placed.

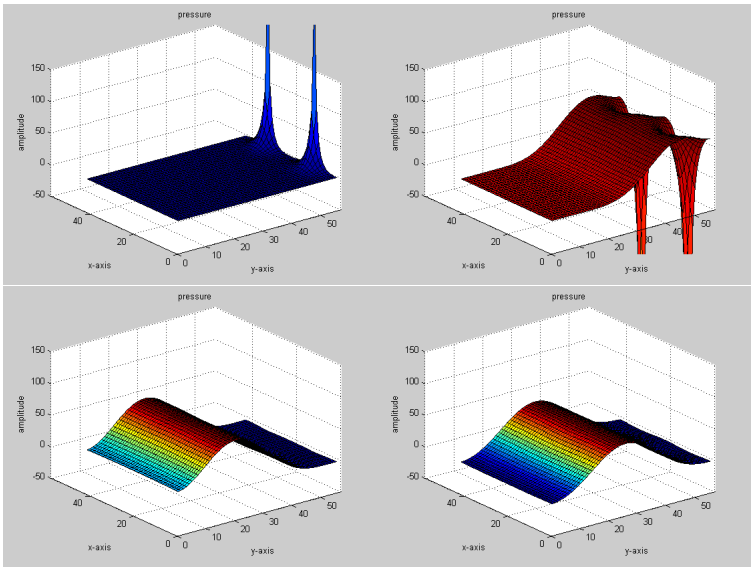

Figure 6.4: Series of plots showing the time evolution of two point sources at (L,W/4,H/2) and (L,3W/4,H/2) in the Groh Room creating a clean plane wave using 20msec pulses.

Chapter 7

Speaker Placement and Rear Cancellation

In an ideal world, when an architect would begin designing a home, he/she would consider that in one part of the home the family that would one day choose to have a home audio system installed. Knowing this, the architect would factor in this information when laying out the floor plan for what would be the living room or rec room in the home. The architect would consider both the dimensions of the room as well as the building materials being used in the room's construction in order to provide the basis for what would eventually be an acoustically suitable space for a new home audio system.

Unfortunately, in most cases in home construction, the acoustic suitability of the living/rec room isn't considered. The norm is that most people buy their homes pre-made without such attentions to detail in their listening rooms and have to deal with the room as it is. So then the question is asked, "Now that I have a listening room, how do I make it better?". The options available are limited. Acoustic absorbing material on the walls, floor and ceiling is certainly a possibility as well as acoustic traps. Both can be expensive. So, instead, it is better to ask the question, "Is there a way I can set up my home audio system to make the combination of room and speakers sound better?". In an attempt to answer this question, we have to start at the beginning for what is done with typical sound systems.

The following sections will be examples of typical sub-woofer placements. The finer details of how much improvement is made, will be looked at when we have developed a proper demerit function. Also, for the analysis of each speaker position, an array of 25 mics equally spaced out over a 2m square region half way up the room and in the xy-plane is used. Five plots will then be presented displaying each row

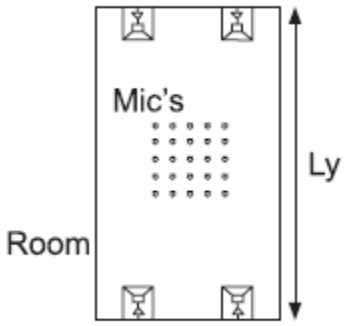

Figure 7.1: Diagram depicting an example of the 25 mics in the room.

of the 5 mics from the front row to the back row . In certain occasions it will also be necessary to employ a mic at approximately (1.5m, 2m, H/2) when comparing different speaker set-ups because that position picks up most of the resonance modes.

7.1 Sub-woofer in the Corner

In a typical 5.1 sound system the sub-woofer is placed in the corner. The benefit of this is that due to the immediate reflections from the floor and the adjacent walls, there is an 8 times increase in power output. The unintended side effect of this, is that from this position all room modes are excited, which gives rise to unwanted room resonances.

By placing a source term in the approximate corner of the room we can measure the frequency response resulting from a 10msec squared raised-cosine pulse in the Groh Room.

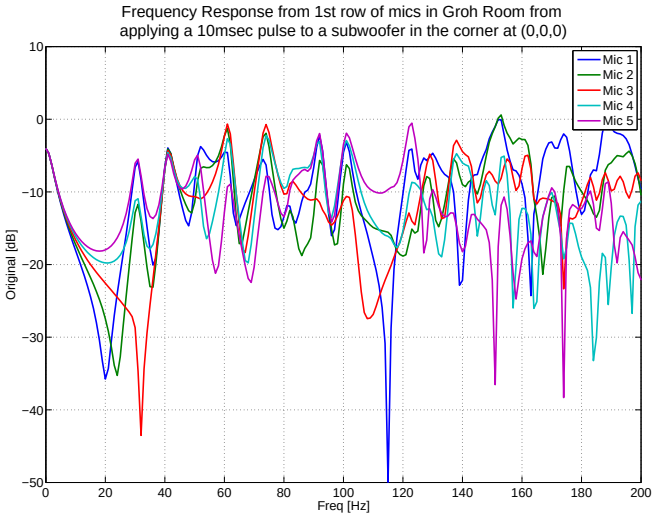

Figure 7.2: Plot of frequency response from the 1st row of mics in the 25 mic array for a sub woofer at (0,0,0).

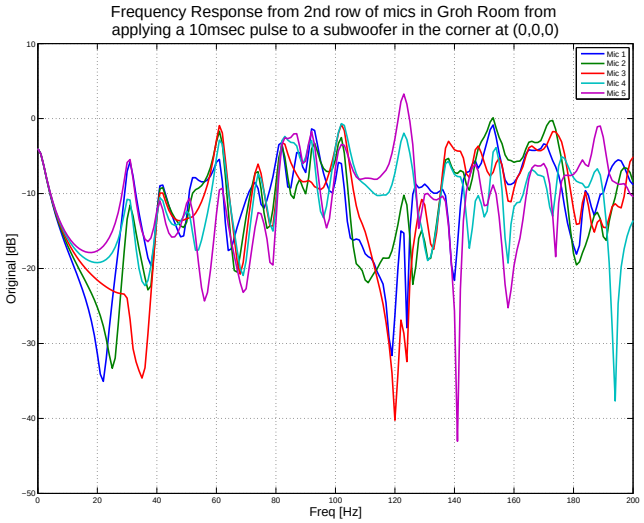

Figure 7.3: Plot of frequency response from the 2nd row of mics in the 25 mic array for a sub-woofer at (0,0,0).

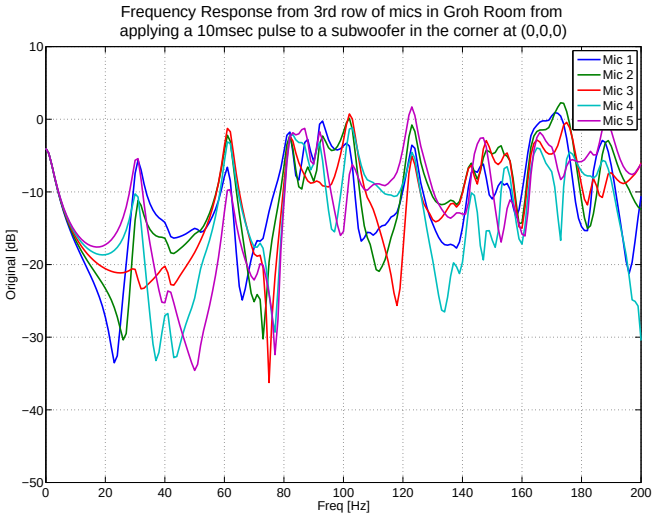

Figure 7.4: Plot of frequency response from the 3rd row of mics in the 25 mic array for a sub-woofer at (0,0,0).

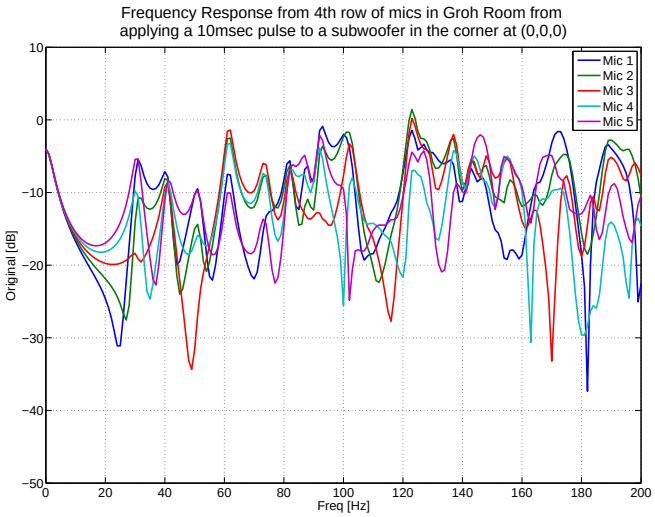

Figure 7.5: Plot of frequency response from the 4th row of mics in the 25 mic array for a sub-woofer at (0,0,0).

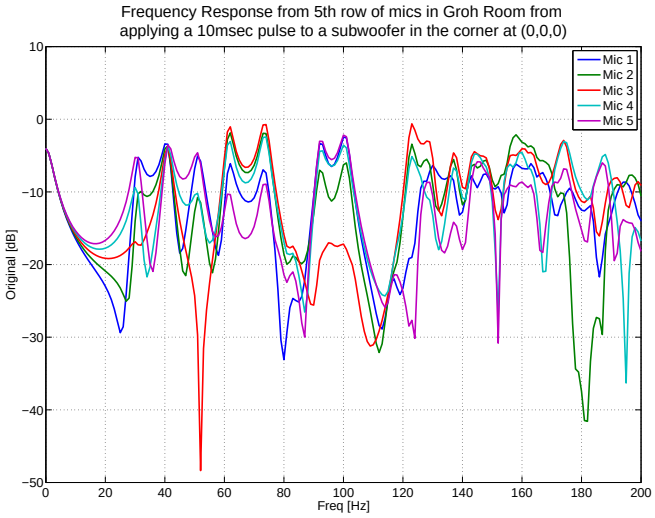

Figure 7.6: Plot of frequency response from the 5th row of mics in the 25 mic array for a sub-woofer at (0,0,0).

As can be seen in the plots there are many low frequency resonances present that correspond to many of the lowest modes seen in table 5.1. The reason not all of them are present is because many lie close to frequencies where there is also an anti-resonance, which are left to the reader to verify. It should also be noted that there is a lot of variability in the frequency responses. This illustrates the other problem that this particular speaker configuration creates. Each sitting position in the area of interest, receives varying frequency responses creating different versions of the performance depending on where a listener is sitting. The ideal would be to receive a similar, or the same, response at all positions.

7.2 Two Front Sub-woofers Tangential Adjustment

As explained in section 6.2, by placing sources at (L,W/4,H/2) and (L,3W/4,H/2) a clean plane wave is created for frequencies below f_{max} ($f_{max} = 143Hz$ in the Groh Room), using a 20msec wide pulse. In figures 7.7 to 7.11 we can see that there is an abrupt change in the distribution of modes, before, and after 143Hz. Below 143Hz the modes are less dense because this is the region where the tangential modes of vibration have not been established, and only the longitudinal modes remain. We observe that above 143Hz the modes are abruptly more dense due to the addtion of the tangential modes. It is also observed that the plane wave created a more uniform

response across all the microphone positions by noting that the responses in all 5 plots appear to be much more uniform than the responses in figures 7.2 to 7.6.

At first glance, someone may conclude that the response due to the plane wave creation is more desirable based on the uniformity of the response in the microphone array area. This is because having random variations in the listening area is one of the problems caused by low frequency resonances in the first place. But at the same time it is observed that the peaks for the resonances that remain appear to be larger. One would wonder if having less resonances that have an elevated level is more appealing to the listener than having several more with lower level. This result is bitter sweet because without doing rigourus listening tests it is impossible to say for sure that this plane wave equalization technique "sounds better" than the single subwoofer in the corner. All that can be said is that the perfomance over the region is more uniform and that is a desired affect.

This method could become even more effective if coupled with a filter to surpress the specific longitudinal modes that are observed in the plots.

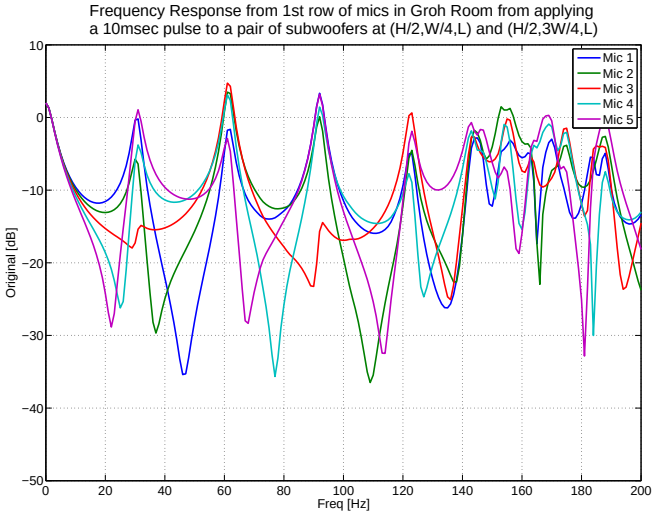

Figure 7.7: Plot of frequency response, from the 1st row of mics in the 25 mic array for sub-woofers w/tangential adjustment.

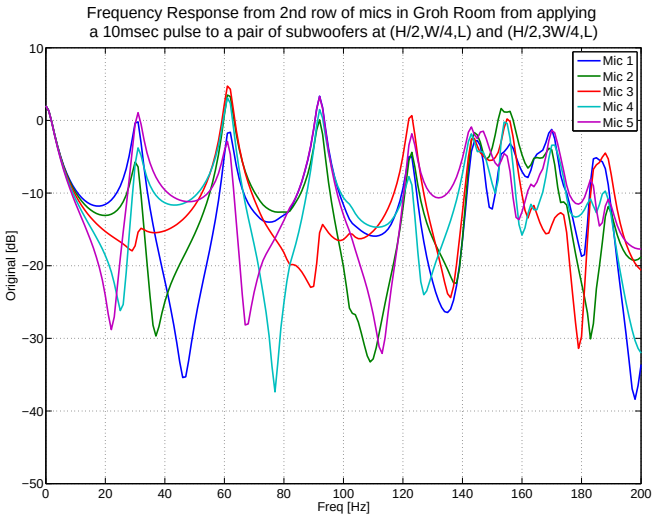

Figure 7.8: Plot of frequency response, from the 2nd row of mics in the 25 mic array for sub-woofers w/tangential adjustment.

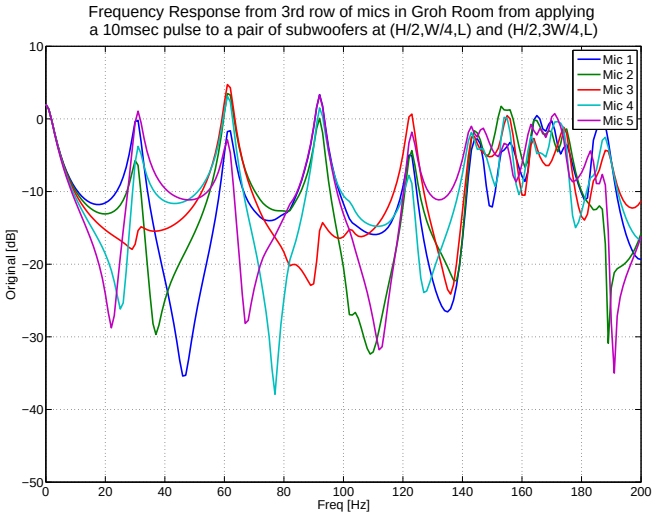

Figure 7.9: Plot of frequency response, from the 3rd row of mics in the 25 mic array for sub-woofers w/tangential adjustment.

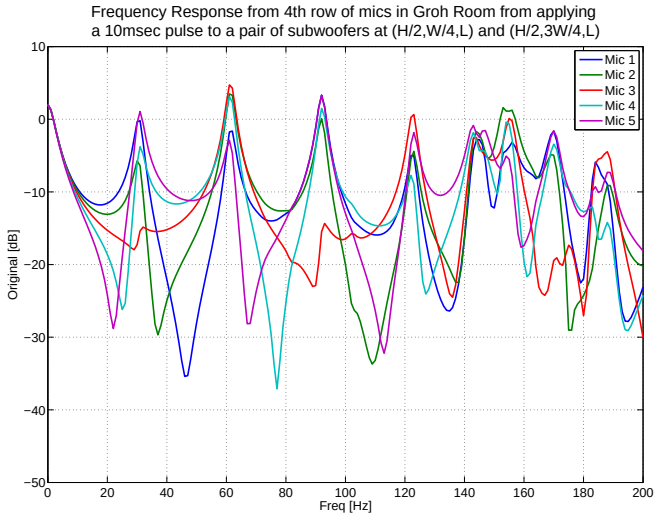

Figure 7.10: Plot of frequency response, from the 4th row of mics in the 25 mic array for sub-woofers w/tangential adjustment.

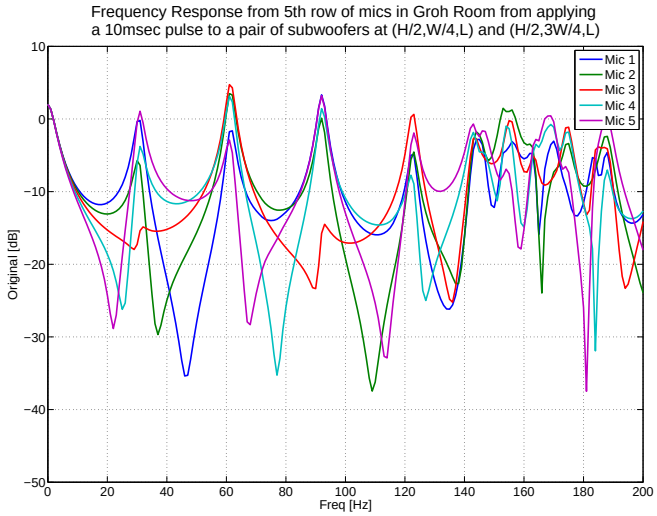

Figure 7.11: Plot of frequency response, from the 5th row of mics in the 25 mic array for sub-woofers w/tangential adjustment.

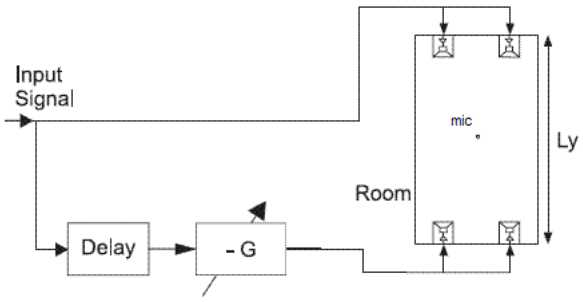

Figure 7.12: Diagram depicting the CABS system.

7.3 Rear Cancellation (aka CABS[1])

The idea behind rear cancellation is to create a plane wave down the room and allow it to pass the listener. Then after the wave has passed by the listener it can then be canceled. This is done by placing the sub-woofers at the front of the room as in section 7.2. Now there is also an identical pair of sub-woofers at the back of the room at (1,W/4,H/2) and (1,3W/4,H/2). The rear speakers are then fed an inverted and delayed version of the signal that is sent to the front speakers(see figure 7.12). The amplitude is determined via some trial and error. This trial and error is made easier with the use of the demerit that will be described in section 9.4. So for now, let's assume that we require an 85% reduction on the amplitude of the cancellation pulses sent to the rear speakers. The time delay is simply calculated by taking the length of the room and dividing by the speed of sound i.e., $time\,delay = L/c$. Thus for a room with L=5.6m the time delay will be 16.279msec. Considering that we have finite time intervals between calculations of .125msec(which corresponds to using a 10cm grid) then we cannot have the time delay be exact. The closest we can get is 130 time steps, or 16.250msec.

As can be observed in figure 7.13 the lowest frequency modes have now been equalized and the response is relatively flat up to about 130HZ. This result is an extreme improvement in room response when comparred to the case of when there was a lone sub-woofer in the corner. It is also very clear that the responses are very similar at each location and one would be getting the same result throughout the listening area.

[1]CABS stands for Controlled Acoustic Bass System and the acronym was coined by Adrian Celestinos in two AES journal papers [8] that were the result of the research outlined in greater detail in these 3 AES convention papers [7]

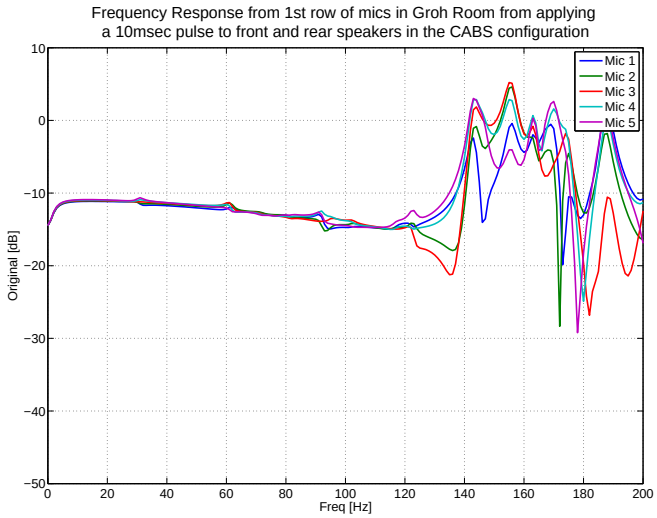

Figure 7.13: Plot of frequency response from the 1st row of mics in the 25 mic array for sub-woofers w/rear cancellation.

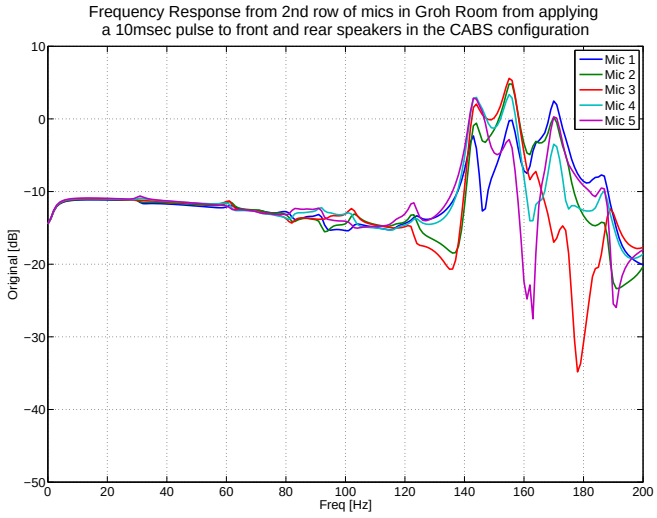

Figure 7.14: Plot of frequency response from the 2nd row of mics in the 25 mic array for sub-woofers w/rear cancellation.

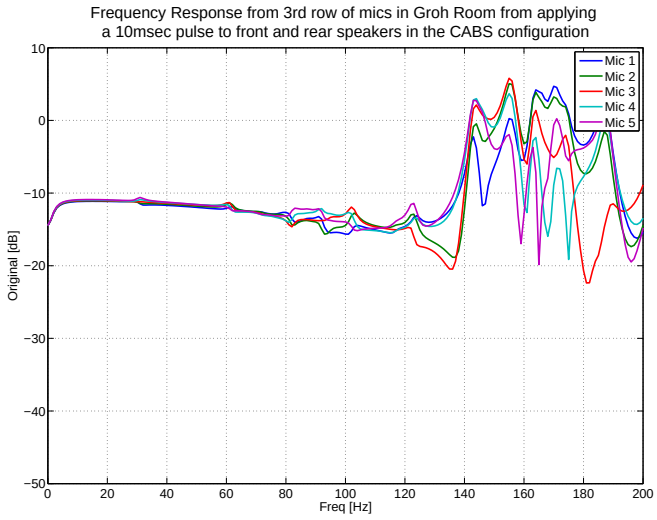

Figure 7.15: Plot of frequency response from the 3rd row of mics in the 25 mic array for sub-woofers w/rear cancellation.

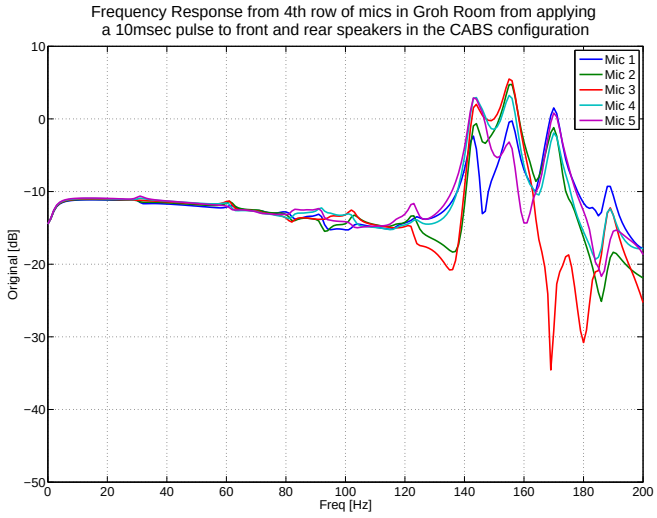

Figure 7.16: Plot of frequency response from the 4th row of mics in the 25 mic array for sub-woofers w/rear cancellation.

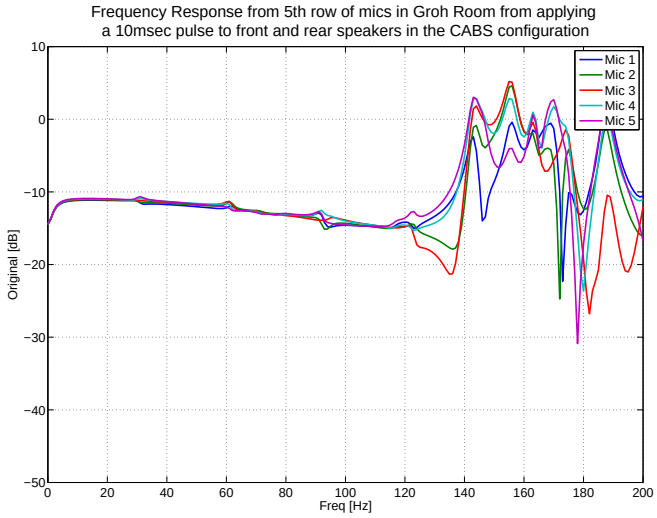

Figure 7.17: Plot of frequency response from the 5th row of mics in the 25 mic array for sub-woofers w/rear cancellation.

7.4 Further Discussion

In order to further analyze the changes made by each speaker configuration the reference mics are used and plotted in figure 7.18. Looking at the blue curve in figure 7.18 we can compare the resonance peaks to the expected resonances predicted by equation 5.9 and displayed in table 5.2. Focusing on the region below 110Hz the peaks are very distinct and peaks #1 to #13 are all present in the plot, with the exception of peak #7 which corresponds to 77Hz. Zooming in on the plot in the region between 20Hz and 100Hz we can see that when the tangential adjustment is made, that a large number of resonances are canceled. The only resonance peaks that remain are the #1, #4 and #11 from the table. If we examine the table closely, we notice that the peaks that were canceled all had n_y and n_z components, but the ones with n_x components alone are indeed the ones left behind. This is an expected result given the positions into which we moved the sub-woofers. Another expected result, is the gain on the resonances of #1, 4 and 11 from the blue to the green curve. It is noted that the gain varies between 6 and 4.5dB. At low frequencies, when an additional speaker is added to a room, the increase in gain is theoretically supposed to be 6dB, which is close to what we are observing in the simulation. It is also important to note that the response observed in the green curve is only obtainable by using two

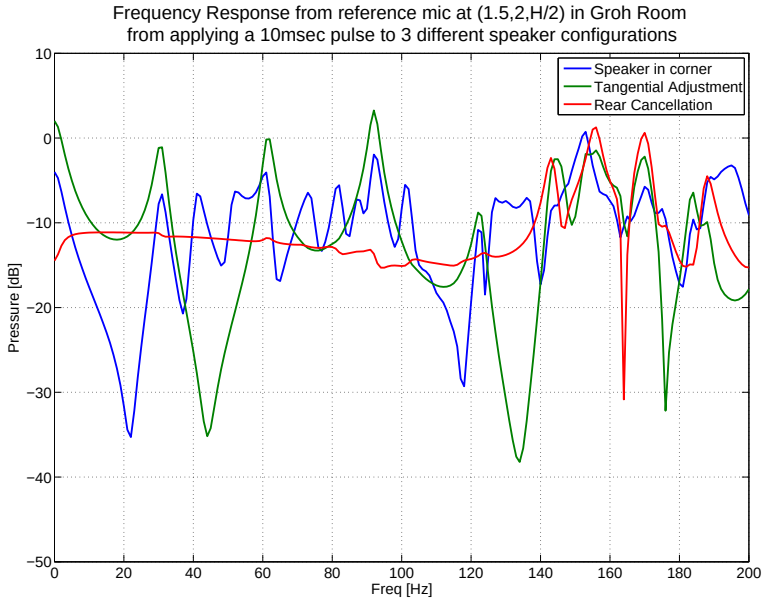

Figure 7.18: Comparison of Speaker Configurations.

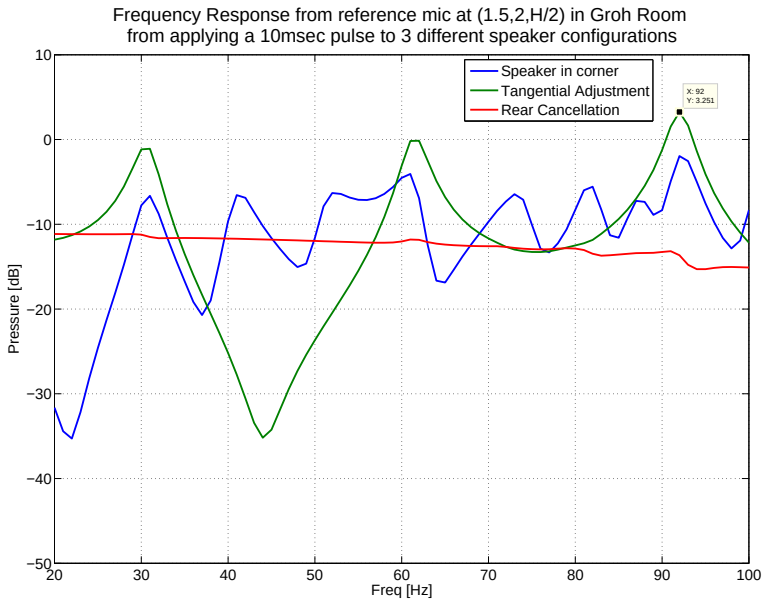

Figure 7.19: Zoomed in plot of figure 7.18.

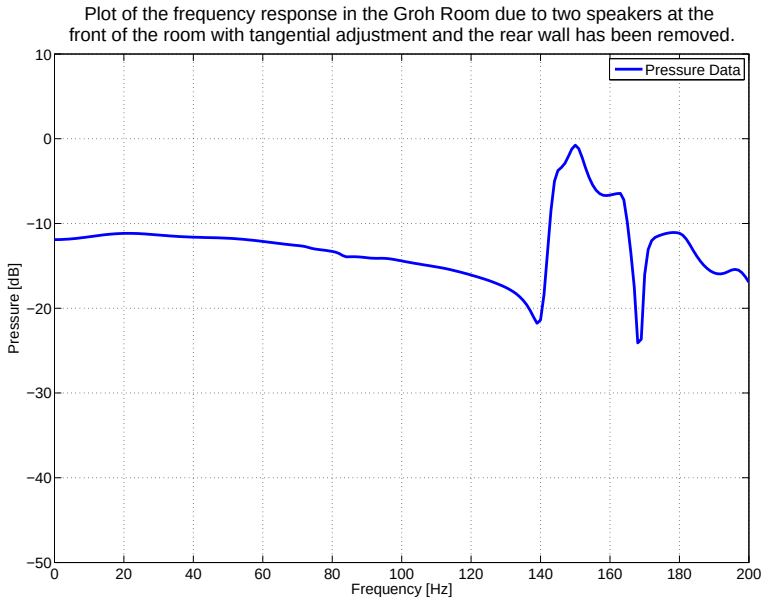

Figure 7.20: Plot illustrating effect of removing the rear wall (setting $\alpha \simeq 1$) in the simulation.

speakers both moved in to W/4 and 3W/4. One speaker alone just doesn't cut it!

The red curve in figure 7.18, involving the cancellation of the plane wave by use of the rear speakers, inverted and delayed signal (CABS) is observed to be canceling the longitudinal waves that correspond to the resonances of #1, 4 and 11. The rear cancellation method accomplishes this because the waves are canceled before they bounce off the back wall and travel back up the room. This is a lot like physically removing the back wall of the room and not allowing any of the waves to bounce off of it. This in practice can't often be done, but we can do it in the simulation by setting the back wall to have an absorption coefficient of $\alpha \simeq 1$. The results are observed in figure 7.20. Clearly the plots are not identical, but the trend from 0 to 140Hz is the same. It is also interesting to note that if the speakers are only up at H/2 and put into the corners instead of moved in from the side walls by W/4 then the length of the flat region is shortened, as observed in figure 7.21. Thus the rear cancellation method does still create an ideal effect but it is not optimized. It is interesting to note that the resonance peaks that had an n_x component are still canceled, which is expected, but the region below 80Hz contains resonances that do not depend on n_x at all and are still being canceled regardless. This is made possible because of what was observed in chapter 6, except the spacing between the sources and images in the

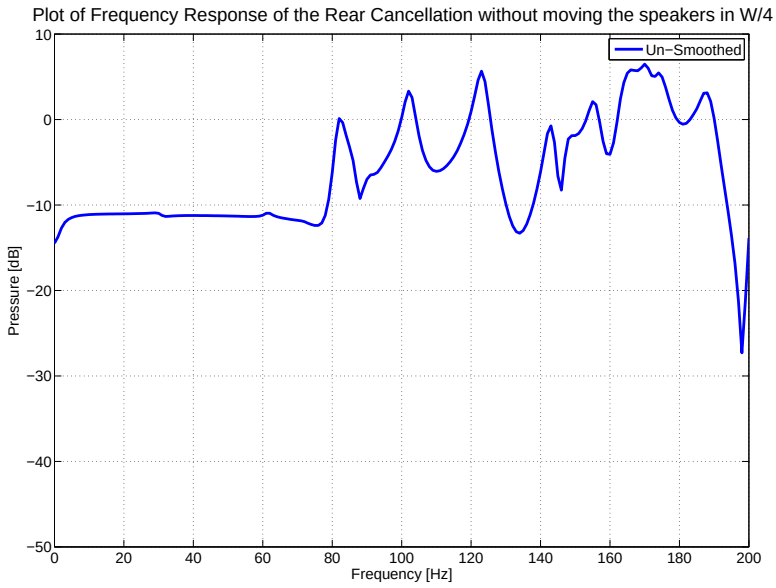

Figure 7.21: Plot of the response of the Rear Cancellation with having the speakers up H/2 but not in from the side walls by W/4.

y direction is now larger. A plane wave is then created for frequencies below f_{max} which is now lower than 143Hz and corresponds to equation 6.1. In this case, d is equal to the width of the Groh Room (4.2m) making $f_{max} = 81.90Hz$, which is close to the cutoff observed in figure 7.21.

Chapter 8

L-shaped Room Analysis

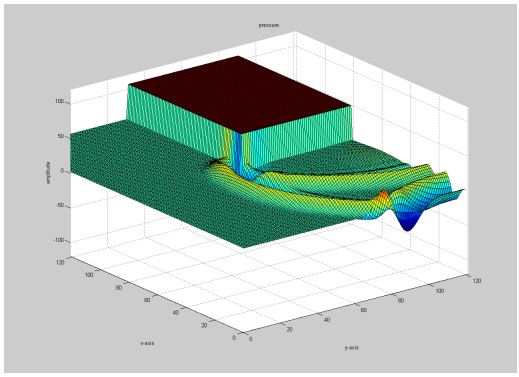

Figure 8.1: An example plot of an L-shaped room in the simulation program

8.1 The L-shaped Room

Although rectangular rooms are the most popular shape for listening rooms, they are not the only shape that people will encounter. In more realistic situations the room may be connected to the outside by a window, an open door or a long hallway. Things like windows and doors that open up to the outside world have an effect on the frequency response in the room. More interesting situations are when that door connects the listening room to another closed room, or the case of a hallway connecting two rooms together. In situations like this we have coupling effects between the two

rooms that may be observed and may cause issues in our attempts to equalize the room. As an example let's look at the effect of adding a different sized rectangular room onto our existing Groh Room. The dimensions of this new 'sub-room' will be Height=same as Groh Room, Length=5m and Width=2.8m(which is half as long as the existing Groh Room). Figure 8.2 depicts the frequency response of the new L-shaped room as a result of a sub-woofer placed in the bottom corner of the room at (0,0,0), the same as in section 7.1.

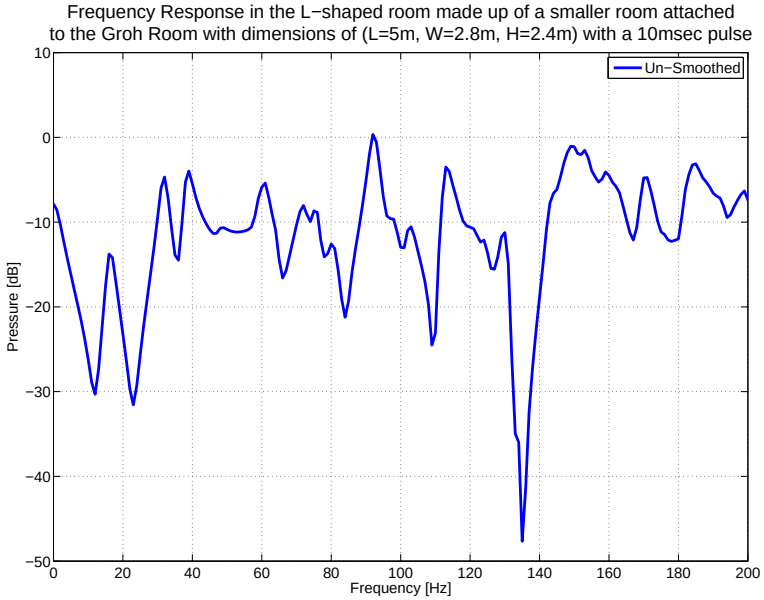

Figure 8.2: Frequency of coupled Groh Room with a sub-woofer in the bottom corner at (0,0,0).

The most notable change is that there is a resonance mode now at approximately 18Hz that was not there before. This new resonance peak, as well as some others, are explained by the eigen-frequency equation 5.9, because of the length 9.2m. In general what is happening is that when the two rooms are coupled together they act like coupled oscillators do. But unlike simple springs, the strength of the coupling has been observed in the simulations thus far, to be similar to that for weakly coupled springs as explained from [5], where μ is the coupling coefficient between the two resonance frequencies, ω_1 and ω_2,

> We have already mentioned that when the coupling spring is weak, so that $\mu^2 \ll \omega_1^2$or ω_2^2, the system behaves like two separate simple oscillators,

#	lx	ly	lz	Frequency (Hz)
1	1	0	0	18.64
2	2	0	0	37.28
3	3	0	0	55.92
4	0	1	0	61.25
5	1	1	0	64.02
6	0	0	1	71.46
7	2	1	0	71.7
8	1	0	1	73.85
9	4	0	0	74.57
10	3	1	0	82.94
11	5	0	0	93.21
12	0	1	1	94.12
13	1	1	1	95.94
14	4	1	0	96.5
15	2	1	1	101.23
16	3	1	1	109.48
17	5	1	0	111.53
18	6	0	0	111.85
19	4	1	1	120.07
20	0	2	0	122.5
21	6	1	0	127.52
22	5	1	1	132.46
23	0	2	1	141.82
24	0	0	2	142.92
25	1	2	1	143.04
26	1	0	2	144.13
27	6	1	1	146.18
28	1	1	2	156.6
29	0	3	0	183.75

Table 8.1: Table displaying the Eigen-Frequencies that are possible in a 9.2x2.8x2.4m room.

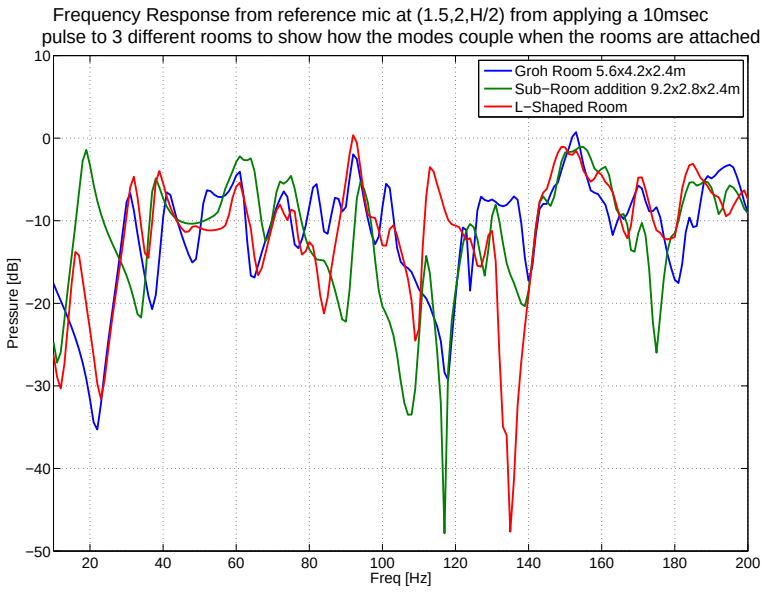

Figure 8.3: Figure displaying the responses of a room that is 9.2x2.8x2.4m(green curve) and comparing to the Groh Room(blue curve) and an L-shaped room(red curve) that is coupled with the new room and the Groh Room.

each slightly influencing the motion of the other.

This point is strongly observed in the region of the resonance peaks from all three curves in figure 8.3, by first focusing on the region about 40Hz. Here the green curve for the sub room has a peak at 38Hz, and the blue curve for the Groh Room has a peak at 41Hz. These two frequencies couple together and result in the stronger 38Hz peak shifting slightly to 39Hz. Also, we can see that the peak for the Groh Room at 31Hz has been shifted to 32Hz and the large 19Hz peak in the sub-room has been reduced in amplitude and shifted backwards to 16Hz. From this we can conclude that the lower in frequency the stronger the coupling.

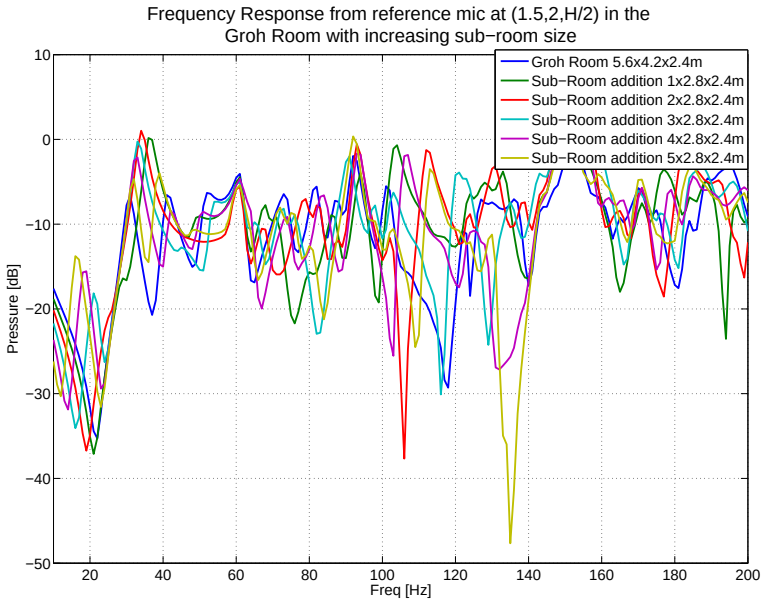

Figure 8.4: Figure illustrating the effect of increasing the length of the sub-room attached to the Groh Room.

Figure 8.3 illustrates the increased modal density when the length of the sub-room is increased.

8.2 L-shaped room with CABS cancellation

Additional frequencies have been introduced by the addition of the sub-room. We also have an extended back wall. In light of the new variables the question arises,

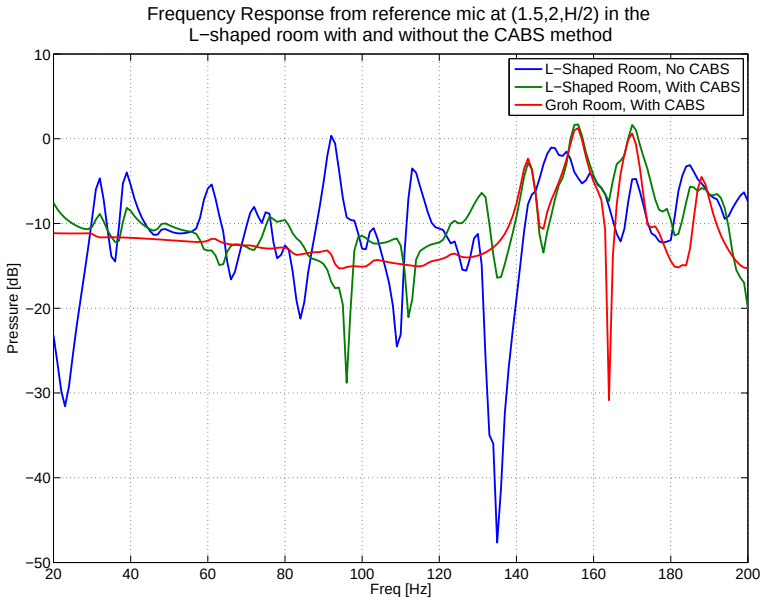

Figure 8.5: Comparison of the L-Shaped room with and without the CABS method.

"how well does the CABS method work in this new situation?" The easiest way to answer that is to run the simulation for this situation and observe the results. The result from the reference mic is presented in figure 8.5.

As can be observed the CABS method does in fact provide a large level of cancellation in the case of an L-shaped room, but the results are not as appealing as the rectangular case. As expected the CABS method still equalized the modes corresponding to the original rectangular room but was not set up to handle the modes corresponding to the addition of the sub room, whose features are still present even with equalization. The degree of cancellation that has occurred will be discussed in the next chapter.

Chapter 9

Room Demerit

In room acoustics there are all sorts of different measurements and calculations one can do to assess how well a room will perform in terms of a particular performance. Whether listening to music from a home stereo or a presentation being given by a guest speaker to a small audience, there are different "rules of thumb" as to how to determine optimum room building parameters. Things like the room dimensions, building materials, and acoustic absorbers may come under consideration for the room's construction. As has been the intention thus far, let us consider the average home listening room that has been constructed using a calculation based on a room quality index (I do not wish to quote a specific one because their validity is not certain). The room is then built and speakers are placed in it. We measure the room's frequency response and plot it along with a smoothed version of itself. Upon comparing both plots in figure 9.1 it can be said that at first glance the one on the right may look more desirable than the other one. This qualitative way of looking

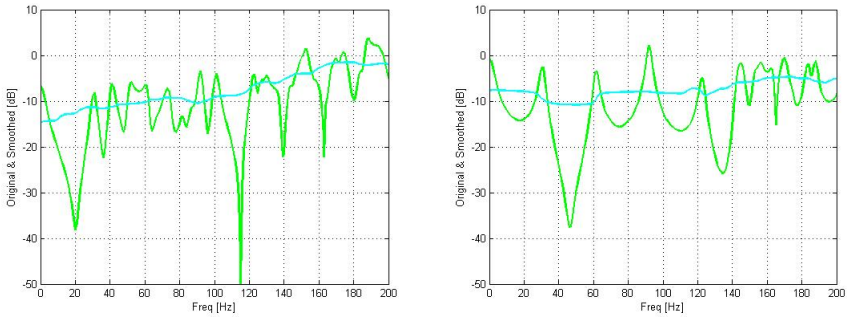

Figure 9.1: Example room response plots

at different room response plots is the common method for measuring whether or not a particular speaker placement is more effective in that location. This method clearly lacks any sort of quantitative measure, which is more desired. The lack of a quantitative measure is what brings up the need of a room demerit function. D2 is given in [36].

If we consider that the smoothed response from the frequency response plots are indeed the desired response of our system, then the deviations from that smoothed response are what needs to be measured. The deviations can then be used to determine how well the room, plus speaker system, are working together to create (or destroy) a pleasurable listening experience.

Thus, the following two formulas are presented as possible candidates for what we would call a "Room Demerit" calculation.

$$D1 = \frac{1}{N}\sum_{n=1}^{N}\left[\left|\frac{P_{unsmoothed}[n] - P_{smoothed}[n]}{P_{smoothed}[n]}\right|\right] \tag{9.1}$$

$$D2 = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{(P_{unsmoothed}[n] - P_{smoothed}[n])^2}{(P_{smoothed}[n])^2}\right] \tag{9.2}$$

There is very little difference between D1 and D2 except one may say that D2 is more mathematically defensible by taking the squared difference, but in the author's personal opinion, D1 is more intuitive and the results are more realistic in how a demerit formula should respond when assessing different rooms. The differences become clearer when looking at computed data, which will be seen later on. For now, let's just say that we have two possible candidates for a demerit function.

It should be noted that the smoothed function is one that uses fixed bandwidth smoothing instead of some sort of factional octave smoothing. At the low end of the spectrum, fractional octave smoothing is using a very small amount data. This results in the need for a very large amount of smoothing (1 to 2 octaves) in order to obtain an ideal smooth curve in the bass region. The reason for this is that at frequencies below the Schroeder frequency, the spacing between peaks is large compared to frequencies after it. The spacing between peaks below 100Hz are especially large. When smoothing begins with points that are approaching the Schroeder frequency the smoothed curve jumps up in level at a point before the rest of the room does. Thus the smoothing is only ever applied using a fixed bandwidth. This is also psycho-acoustically defensible because it has been shown that the critical bandwidth at very low frequencies is almost independent of frequency [35].

9.1 Demerit Function Application to Multiple Mics

Applying the demerit function to a single microphone is very straight forward but, in order to assess how a room is behaving it is important to have multiple microphone positions. When applying the demerit to more than one microphone we are still left with the problem from before with the multiple plots that had to be compared, but now we have 25 different demerit values that we have to scan through and assess for each mic. Thus, it might make sense to take the average of those values and find an average room demerit, $\overline{D}$. The smaller value for $\overline{D}$ we obtain, then the less deviation there is from the smoothed curves. But it has been shown that, in particular, for the case when a sub-woofer is placed in the corner of the room there is great variability across the listening area. This logically means that even the smoothed curves at each mic could be different from one another depending on where the microphone is. It is observed though that the smoothed curves for each mic position do differ a little but not significantly. Thus data will be presented where the demerit calculation is performed based on (1) individual smoothed curves and (2) an average smoothed curve. Logic would suggest that it would be best for the demerit calculation to be based on a single reference smoothed curve that is the average of all the smoothed curves, instead of individual smoothed curves. This is justifiable because one of the goals has been to make the listening experience uniform across the whole area so it makes sense to use a single reference curve for this exerecise.

Along with taking the average demerit it then only makes sense to try and make use of the other statistical tools that go along with calculating an average. These tools are the Standard Deviation (SD) and the Standard Deviation of the Mean (SDOM).

9.1.1 Statistics Tools[1]

Before we get into the SD and SDOM we have to first properly define exactly what the mean is. The mean value for any set of values, x_i, is given by the formula,

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i = mean \tag{9.3}$$

which is assumed to be familiar to the readers.

The standard deviation is given by the formula,

$$\sigma_x = \sqrt{\frac{1}{N-1}\sum(x_i - \overline{x})^2} = standard\,deviation,\ or\ SD. \tag{9.4}$$

[1]The statistical tools and their definitions are taken directly from Taylor [33].

According to [33] the standard deviation characterizes the average uncertainty in the measurements of x_i. In the case of the demerit, the standard deviation characterizes the average variability between the individual measured responses and the average response, which gives a measure for how different one mic is from the average. Smaller numbers indicate greater conformity to the average value across all the mics.

The last statistical tool, although not as useful or important for completeness, should be mentioned. Simply put, the standard deviation of the mean is a measure of the uncertainty in the value of the mean as being the true value out of a series of separate measurements.

$$\sigma_{\overline{x}} = \frac{\sigma_x}{\sqrt{N}} = standard\,deviation\,of\,mean,\,or\,SDOM. \tag{9.5}$$

9.1.2 Demerit Parameter Choice

Two things need to be known. The first, is how large should the smoothing bandwidth be. The second, is at what frequency do we stop calculating the demerit.

It was previously mentioned, that for the smoothing we are using a constant frequency bandwidth because at low frequencies fractional octave smoothing typically smooths too little below 100Hz. What needs to be determined, is how big the bandwidth should be. We need to use a large enough bandwidth so that it uses enough data to create a flat curve, but at the same time we do not wish to use a bandwidth that is so large that it uses too much data (for reasons that will be made clear with the CABS method example that follows) and is excessively smoothed, to be explained momentarily. The curves that are in section 7.2 will be used because the separation between peaks is greatest for these plots and will require a larger smoothing bandwidth than the case for when the speaker is in the corner. By changing the bandwidth from 30Hz to 60Hz and observing the change in how smooth the smoothed curve became, it was decided that the 60Hz bandwidth worked the best. This is shown in figure 9.2.

With the bandwidth now chosen the demerit functions can now be applied to real data. Ideally, what we would want is to calculate the demerit from 0Hz to the Schroeder frequency in order to assess the room in the region where the resonance modes are still significantly spaced apart. Unfortunately, in the case of the Groh Room this does not yield good results. The reason is, when we observe the plots in section 7.3, we see that the cancellation only works up to 130Hz, which is well below the Schroeder frequency. Therefore, assessing the region above 130Hz, that has not been equalized will only make the value of the demerit larger and not properly asses how much change has happened in the region where there is a significant amount of

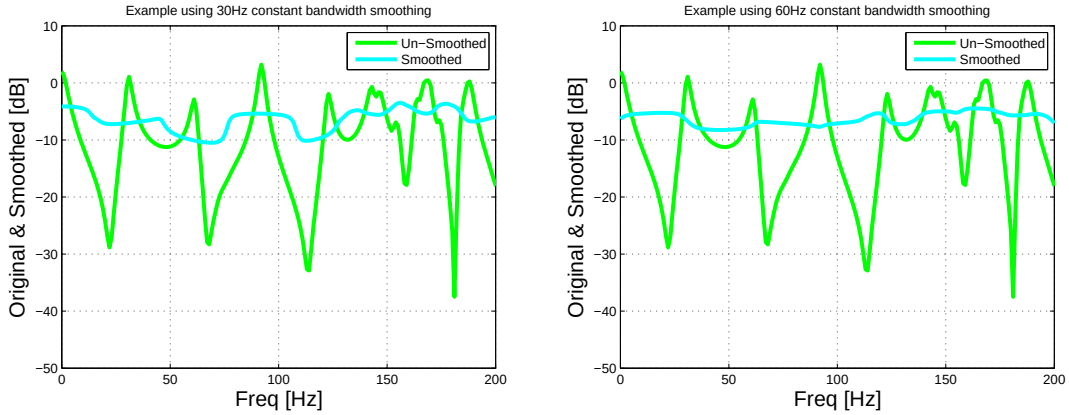

Figure 9.2: Example plots of smoothing performed on the same data using constant bandwidth smoothing with a 30Hz bandwidth (left) and a 60Hz bandwidth (right).

equalization going on. This gives rise to the reasoning that, in the case for the Groh Room, we should stop calculating the demerit at least below 130Hz. This is in fact still too high and we have to observe an example curve from the resultant CABS method to understand why.

As can be observed from figure 9.3, the smoothed curve deviates strongly from the un-smoothed curve just after 110Hz. This deviation is because the smoothing is now beginning to take into account the data after 140Hz, where there is little equalization and the average level jumps up by nearly 10dB. If octave smoothing were being used, this deviation occurs much earlier and is another reason to support constant bandwidth smoothing. This deviation may not occur in the same place and is dependent partly on the room dimensions. So for now, we can say that in order to have a proper assessment of the amount of equalization in the Groh Room we have to stop calculating the demerit at 100Hz.

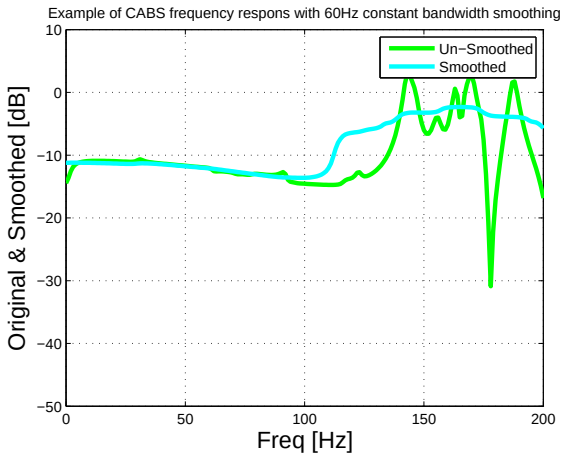

Figure 9.3: Example of 60Hz constant bandwidth smoothing with CABS.

9.2 Rectangular Rooms Revisited

Back in chapter 7, when we took a look at the rectangular room and the various speaker configurations that were used, we noticed that the frequency responses showed fewer resonance peaks and more uniformity as we went from a single sub in the corner to using the CABS method. We would expect that the demerit values should reflect what was observed in the plots with decreasing values of both $\overline{D}$ and σ_D. The following tables are summaries of the demerit values taken for the different changes made in the Groh Room for both the individually smoothed and average smoothed curves.

	Sub in Corner	2 subs 1/4width, 1/2height adjustment	CABS
$\overline{D}1$	0.9269	1.1799	0.0814
$\overline{D}2$	1.7001	3.7937	0.0135
σ_{D1}	0.0922	0.0443	0.0112
σ_{D2}	0.3068	0.5136	0.0022
$\sigma_{\overline{D}1}$	0.0092	0.0044	0.0011
$\sigma_{\overline{D}2}$	0.0307	0.0514	2.2272e-4

Table 9.1: Demerit values taken from the Groh Room with individually smoothed curves.

	Sub in Corner	1/4width 1/2height adjustment	CABS
$\overline{D}1$	0.9538	1.2043	0.0840
$\overline{D}2$	1.7563	4.2016	0.0140
σ_{D1}	0.1088	0.0430	0.0137
σ_{D2}	0.3427	0.6272	0.0029
$\sigma_{\overline{D}1}$	0.0109	0.0043	0.0014
$\sigma_{\overline{D}2}$	0.0343	0.0627	2.9016e-4

Table 9.2: Demerit values taken from the Groh Room with a single average smoothed curve for each speaker configuration.

As we can see from the tables, the values are not very dissimilar between using individual smoothed curves and using an average smoothed curve.

If we compare the "1/4width 1/2height adjustment" column with the CABS column we can see that the demerit values decrease by a significant amount for the CABS column. Both versions of the demerit show a decrease in the average demerit

value as well as a much smaller value for the Standard Deviation which indicates both smoothness and uniformity of the response across all 25 mics. This is indeed the predicted result we expected to see earlier. What was not expected though, was the results comparing the "Sub in Corner" column with the "1/4width 1/2height adjustment" column. It is expected that all the values for the demerit measures should have decreased when applying the 1/4width 1/2height adjustment. In order to understand this result let's focus on figure 9.4 taken from chapter 7. After making the 1/4width 1/2height adjustment or tangential adjustment, we can see from the figure that the peaks and dips from the green curve have a much larger magnitude than those of the blue curve that correspond to the single speaker in the corner. So even though the number of resonances has been reduced, the remaining modes of vibration have been excited more strongly and this explains the higher $\overline{D}$ values. But, if we recall, the tangential adjustment was supposed to create a plane wave which should result in a more uniform response across the listening area. We can see this effect by comparing the plots for the sub in the corner (figures 7.2 to 7.6) to the plots for the tangential adjustment (figures 7.7 to 7.11). It was observed that the plots for the tangential adjustment are more similar to one another than the plots for the sub in the corner. This trend is reflected in the tables where the value of σ_{D1} does in fact get smaller (about half the size) for the tangential adjustment response. It is puzzling as to why the σ_{D2} values do not share this trend. Given the fact that the D1 demerit values conform to what is being observed, then this makes a case for the validity of equation 9.1 over equation 9.2.

9.3 L-Shaped Room Revisited

It has already been observed that the CABS method does indeed equalize the L-shaped room to a large degree, although not as well as the simple rectangular case. Now that we have defined demerit functions that appear to work reasonably well for obtaining a value as to how well the equalization is working, we can have some value to help us judge how bad the equalization is in the L-shaped room compared to the typical Groh Room.

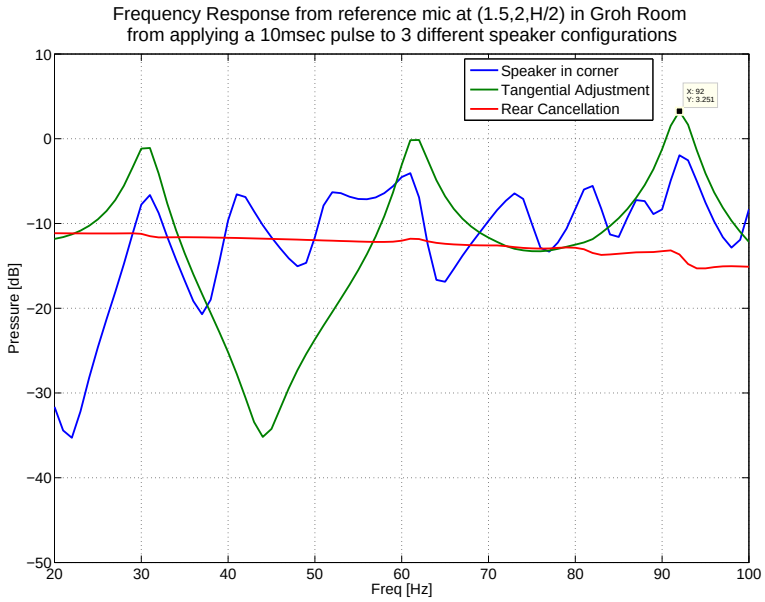

Figure 9.4: Zoomed in plot of figure 7.18.

	Sub in Corner	1/4width 1/2height adjustment	CABS
$\overline{D}1$	0.8760	1.0379	0.5665
$\overline{D}2$	1.5043	2.5736	0.9097
σ_{D1}	0.1067	0.0461	0.0421
σ_{D2}	0.4345	0.4324	0.1816
$\sigma_{\overline{D}1}$	0.0107	0.0046	0.0042
$\sigma_{\overline{D}2}$	0.0435	0.0432	0.0182

Table 9.3: Demerit values taken from the L-shaped room with individually smoothed curves.

	Sub in Corner	1/4width 1/2height adjustment	CABS
$\overline{D}1$	0.9538	1.0514	0.5811
$\overline{D}2$	1.7563	2.7512	1.0835
σ_{D1}	0.1088	0.0815	0.0490
σ_{D2}	0.3427	0.7958	0.5113
$\sigma_{\overline{D}1}$	0.0109	0.0081	0.0049
$\sigma_{\overline{D}2}$	0.0343	0.0796	0.0511

Table 9.4: Demerit values taken from the L-shaped room with a single average smoothed curve for each speaker configuration.

From the tables we can see that the trends are very similar to what was observed in the Groh Room. As expected, the values in the CABS columns are much smaller than the others, but not nearly as small as the CABS columns for the Groh Room. This indicates that the CABS method is still working reasonably well even though we do not have a closed off rectangular room.

9.4 Amplitude Selection for CABS using Demerit.

In section 7.3 it was mentioned that the amplitude used for the rear speakers that perform the cancellation was chosen based on trial and error. This is because the amplitude for the rear cancellation not only depends on the room dimensions, but also the amount of absorption in the room. This makes its determination by means other than using trial and error difficult and is a problem that is left unsolved here.

It is very easy to run a simulation and look at the plot and say, "yeah, that looks like it's working pretty well". But now with a demerit function we can make the process all that much more accurate.

Alpha=0.01				Alpha=0.1		
amplitude(%)	$\overline{D}1$	$\overline{D}2$		amplitude(%)	$\overline{D}1$	$\overline{D}2$
100%	0.1416	0.0454		90%	0.1123	0.0293
98.75	**0.1311**	**0.0412**		87.5	0.0876	0.0185
97.5	0.1329	0.0616		**85**	**0.0840**	**0.0140**
95	0.2081	0.3148		82.5	0.0988	0.0189
92.5	0.3528	1.1062		80	0.1202	0.0362
90	0.5315	2.6192		75	0.2069	0.1186

Alpha=0.5				Alpha=0.1	4.2x11.2x2.4	
amplitude(%)	$\overline{D}1$	$\overline{D}2$		amplitude(%)	$\overline{D}1$	$\overline{D}2$
40	0.2018	0.0619		90	0.1988	0.0672
37.5	0.1945	0.0570		80	0.0899	0.0176
35	0.1907	**0.0545**		**77.5**	**0.0780**	**0.0141**
32.5	**0.1906**	0.0546		75	0.0880	0.0167
30	0.1932	0.0573		72.5	0.1111	0.0265
25	0.2056	0.0703		70	0.1415	0.0446

Table 9.5: Tables showing the change in desired rear cancellation amplitude with different alpha values and change in size of the room dimensions.

Table 9.5 is a sample of data that was used in determining the correct amplitude for the CABS method. In general, it is observed that when either the value of alpha increases or the size of the room increases, the required amplitude to perform the CABS method decreases. This is expected.

There are a few interesting things to note about the data in table 9.5. For starters, when alpha is 0.01 the ideal amplitude to use turns out to be extremely close to 100%. This is interesting, in that it points out the fact that the plane wave that is created, decays very little as it travels down the length of the room in the absence of a significant degree of absorption on the walls. This is to be expected, since plane waves do not decay at all, whereas spherical waves decay like 1/r. Alpha=0.01 is also a very unrealistic level of absorption in the first place. Another point is the fact that for alpha=0.5 we notice that the results from the two demerit functions disagree slightly. To have an alpha=0.5 is a very unrealistic situation as well, and the response in the room without the CABS method is already reasonably flat (see figure below) and has a $\overline{D}1 = 0.4130$ and $\overline{D}2 = 0.2556$.

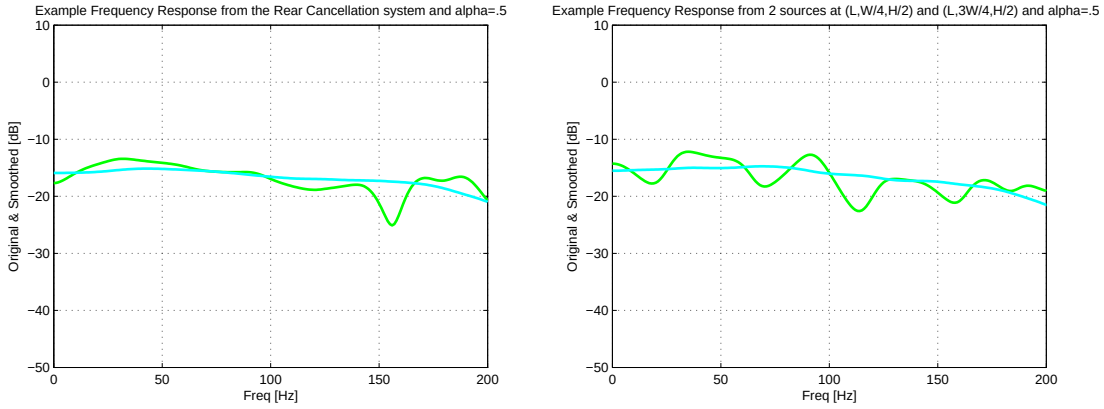

Figure 9.5: Comparison of optimized CABS method(left) to two front sources(right) with an alpha of 0.5.

Chapter 10

Summary

In this thesis we investigated several aspects that pertained to two main topics, the first is the development of the FDTD method, and the second is the bass equalization and development of a demerit function.

10.1 FDTD Method

The Finite Difference Time Domanin method and how it is applied to the linear acoutic equations was the first topic of interest. We needed to develop and explore the FDTD method and its applicability to acoustics because it would be used as a tool to help propel this research poject forward into other areas of investigation. This subject brought up two important questions. How accurate is the FDTD method for simulating room acoustics, and how do we properly excite the system?

The first question was tackled both directly and indirectly throughout the thesis. We saw specifically in chapters 4 and 5 the FDTD simulation program, that was written and is contained in the appendix, provided physically realistic results in testing the boundary condidtions, verifying the room size, and the observed reverberation time. As well we saw in chapter 2 that a spherical pulse did in fact obey a 1/r law as predicted by the physics in [22]. It was also observed, that other expected physical results were re-created throughout chapters 5 through 8. For example, the observation of a 6dB increase in sound pressure level when going from a single speaker setup to a two speaker setup. The most sigificant example was the creation of a plane wave from a single source and its images. From these observations it can be concluded that the FDTD method is indeed an accurate method for simulating room acoustics.

The second question, of how we properly excite the system, was one that came about out of necessity. Given that this was a subject that seemed to be almost

completely passed over in much of the literature, the derivation of a proper source term is a very important accomplishment, even though it was never meant to be an important focus for this masters project. The source derived in chapter 2 comes out directly from the physics of acoustics in the continuity equation, and acts identically like a closed box loudspeaker at low frequencies (a spherical point source). Thus we can conclude from this that the source that we developed is a proper way to excite the system.

Overall we can see that the FDTD method, even with very basic boundary conditions, is a very valuable and powerful tool for doing low frequency research for reasonable sized listening rooms.

10.2 Bass Equalization and Demerit

From developing the FDTD simulation program in MATLAB we now have a tool that allows us to look at various speaker placements and cancellation techniques, as well as the ability to observe some special acoustic situations that would not normally be observed in real world situations (i.e., removing the back wall of the listening room to observe that there are no longitudinal modes of vibration). In order to assess whether or not one equalization technique was performing better than another a demerit function was required. The original demerit that was used was found in [36] and with some slight modifications, the two demerit functions presented in chapter 9 were then able to be applied to 25 virtual microphones in the simulation program.

The first equalization method that was looked at was the creation of plane waves down the room. This had the effect of reducing the number of modes present in the room but also had the effect of increasing the level of the left over resonances, which is not a desired result. The increased level was reflected in larger demerit values when compared to the values obtained from the reference frequency response obtained from placing a single sub-woofer in the corner of the room. The other effect that the plane wave had, was that the frequency responses were observed to be more uniform across the 25 microphone positions. This is certainly a desired result given the goal of creating an "acoustically sweet area" instead of just having a single "sweet spot". This result was reflected in a smaller SD value for the D1 demerit function, but was not observed in the SD value for the D2 demerit function. The reason for this remains unknown and leads us to consider that perhaps D1 is a more valid fuction for calculating the demerit than D2 is. To recap, this equalization has a larger demerit value due to the larger resonance peaks but the response is more consistent. Whether this creates an overall improvment in listening experience is unknown, given the conflicting results. However, it should be noted that the creation of the plane wave

is essential for achieving the significant equalization found in the next equalization method.

The second equalization method that was tested was the rear cancellation method that was originally presented by Adrian Celestinos in [6, 7, 8, 9, 10]. As expected this equalization technique works extremely well, creating a flat response in the region below f_{max}. This result is also reflected in the much smaller values for the demerit and the SD. The interesting result observed from this equalization technique is that it is rather robust and achieves very good results when applied to the L-shaped room, and can presumably may work well in other situations.

The demerit function is in general used as a measure of how well the room responds to a speaker setup. In the case of rear cancellation it is also used as a tool to help in determining the correct amplitude for the rear speakers. This requires being able to sense and differentiate between small changes in the response. Both demerit functions have been shown to be sensitive enough for this job.

Overall the dermerit functions (in particular D1) have been observed to reflect the changes in the frequency response over the 25 mic positions with trends that coincide with what is being observed, thus connecting the qualitative to the quantitative results.

10.3 Future Work

10.3.1 Auralization

One of the interesting points to note about the pulse that we developed, is the fact that it directly employs the volume velocity function Q(t). This function is directly derivable from the physics applied to loudspeakers as seen in [26]. In the paper the function for the displacement of the cone, $X(\omega)$, is given and is a function of the voltage applied across the speaker terminals, as well as the other common Small-Thiele parameters that are commonly known for speakers. Since $Q(t) = v(t) \cdot Cone\,Area = \frac{dx(t)}{dt} \cdot Cone\,Area$, then the function Q(t) can be obtained from the signal voltage and the loudspeaker model. What this does is it allows us to simulate any speaker in the FDTD simulation and observe how it couples to the room if we are aware of its Small-Thiele parameters and the signal that would be applied to the speaker. Since this is all done in the time domain, we can even feed the simulated speaker an actual audio file and record the pressure output at the listening position, in order to see how a particular speaker might sound in the room we are interested in constructing.

The drawback to this method, is that because of the fact that we are oversampling in time we have to up sample whatever sound file we wish to play. This means that if

we wish to input a raw music file that has a length of 30seconds containing frequencies higher than 10kHz then we need a much smaller grid spacing and thousands of iterations. It was calculated that in order to do this, on a conventional PC, it would take months. In order to get around this problem it was concieved that we could filter the sound file we wish to test, inputting the low frequency content that would be affected by the room modes and storing the high frequency portion to be added back in afterwards. This would allow us to use a coarser grid spacing and save alot of computational time. In addition to splitting the frequencies of the sound file up, we could also just obtain an impulse response for the grid spacing of interest instead of playing the whole 30second sound file in the simulation. We then would convolve the results to get back what the sound file would sound like in the room.

In order to obtain an impulse response from the room that is suitable for auralization, a very fine time and spatial step FDTD simulation would need to be done. This is computationally challenging. Thus, we would look for ways of modelling the low frequencies and adding a simpler model for the high frequencies.

This technique seemed a good idea but it required that a reference be made. A reference that contains an impulse for the simulated room for an very small grid spacing would be needed in order to tell if the sound artifacts we would hear in the recording are from the errors and dispersion related to the FDTD method, and not from our filtering and equalization. In the authors opinion, in order to obtain a satisfactory reference it would have been a large undertaking and taken over a month to calculate on the current hardware. This led the author to abandon this aspect of the project in hopes of continuinig this work in the future.

10.3.2 Boundary Conditions

From the investigation in chapter 4, we have seen that the boundary condition that we chose to use was appropriate, by mimicing the physical results one would expect for a locally reacting surface at low frequencies. The drawback here, is that not all surfaces are locally re-acting and the amount of absorption can vary with frequency. Given that the work we were performing only had to do with low frequencies, the varying absorption was not so much of an issue, but it is foreseeable that one would like to work with situations where the boundaries are not locally re-acting. Thus, a more robust boundary condition is certainly worth adopting into the simulation program. Examples of different boundary conditions can be found in [23, 24].

10.3.3 CUDA(Nvidia) and Stream(ATI) programming

Problems being solved using numerical methods are becoming more and more complex and require more and more computer power to solve them. A perfect example, is the issue that came about when we attempted to try Auralization. The speed of single CPUs have roughly hit a limit, so smarter more clever ways have been developed by scientists to tackle today's intense numerical problems. The most popular is to break up the problem and have it solved by more than one CPU at a time. This is called parallel computing. This new way of solving large numerical problems changed the direction of CPU manufactuers to, instead of making faster CPUs, to now put multiple CPUs, or "cores", on a single chip. The current CPU architecture is limited by the maximum number of CPU cores that can be put on a single chip. This gave way to looking to the GPU, or graphics processor to handle the parallel processing work. GPU's are made up of hundreds of smaller, less powerful cores instead of a small handful of really powerful ones. The idea and strength behind using GPU's instead of CPU's is that we are taking on the problem with thousands of little chisels instead of one big sledge hammer.

Programming on a GPU is different than programming on a CPU, and thus, Nvidia and ATI (the two largest GPU manufacturers) have come up with their own C like programming languages. Nvidia introduced CUDA (Compute Unified Device Architecture) and ATI introduced Stream, in order to leverage the power of the manufacturers' respective graphics processors. There are also different programs and plugins for existing applications, like MATLAB, that allow one to seamlessly carry out calculations in a programming language that is familiar to the user. An example of this can be found at http://gp-you.org/, where a plugin for MATLAB has been developed to allow it to access Nvidia GPU's directly and perform computations. The author has tried this plugin and observed increases in the speed of computation over his CPU by more than a factor of ten using a conventional graphics processor (Nvidia GeForece 9600GT).

The FDTD method is a prime candidate for parallelization because it can be broken up into pieces and the values only depend on previous values that are stored and not immediately on the values around that particular piece. Research grade GPU's boast speed increases of up to 30 times that of a high end CPU. Thus, in order to handle higher and higher frequencies directly and to do auralization, some sort of parallel processing using CUDA or Stream would be a great asset and should strongly be considered for the future.

Bibliography

[1] Allen R. Groh. "High-Fidelity Sound System Equalization by Analysis of Standing Waves". J. Audio Eng. Soc., Vol 22, No. 10, 1974 December.

[2] Todd Welti & Allan Devantier. "Low-Frequency Optimization Using Multiple Subwoofers". J. Audio Eng. Soc., Vol 54, No. 5, 2006 May.

[3] Todd Welti. "How Many Subwoofers are Enough". Presented at the 112th Convention of the Audio Engineering Society, Munich, Germany, May 10-13 2002, paper 5602.

[4] J. Abildgaard Pedersen, "Adjusting a loudspeaker to its acoustic environment", Proc. AES 115th Convention, New York, convention paper 5880, (October 2003).

[5] Arturo O. Santillan. "Spatially extended sound equalization in rectangular rooms". J. Acoust. Soc. Am. 110 (4), October 2001.

[6] Adrian Celestinos and Sofus Birkedal Nielsen. "Multi-source low frequency room simulation using finite difference time domain approximations". Presented at the 117th Convention of the Audio Engineering Society, San Fransisco, CA, USA, October 28-31 2004.

[7] Adrian Celestinos and Sofus Birkedal Nielsen. "Optimizing placement and equalization of multiple low frequency loudspeakers in rooms". Presented at the 119th Convention of the Audio Engineering Society, New York, NY, USA, October 7-10 2005.

[8] Adrian Celestinos and Sofus Birkedal Nielsen. "Low frequency sound field enhancement system for rectangular rooms using multiple low frequency loudspeakers". Presented at the 120th Convention of the Audio Engineering Society, Paris, France, May 20-23 2006.

[9] Adrian Celestinos and Sofus Birkedal Nielsen. "Low-Frequency Loudspeaker–Room Simulation Using Finite Differences in the Time Domain—Part 1: Analysis". J. Audio Eng. Soc., Vol 56, No. 10, 2008 October.

[10] Adrian Celestinos and Sofus Birkedal Nielsen. "Controlled Acoustic Bass System (CABS) A Method to Achieve Uniform Sound Field Distribution at Low Frequencies in Rectangular Rooms". J. Audio Eng. Soc., Vol 56, No. 11, 2008 November.

[11] Soren Krarup Olesen. "Low Frequency Room Simulation Using Finite Difference Equations". Presented at the 102nd Convention of the Audio Engineering Society, Munich, Germany, March 22-25 1997.

[12] D. Botteldooren. "Acoustical Finite-Difference Time-Domain Simulation in Quasi-Cartesian Grid". J. Acoust. Soc. Am., 96, pp. 2313-2319. (May 1994).

[13] D. Botteldooren. "Finite-Difference Time Domain Simulation of Low-Frequency Room Acoustic Problems". J. Acoust. Soc. Am., 98, pp. 3302-3309. (December 1994).

[14] Jose J. Lopez, Jose Escolano, and Basilio Pueo. "On the Implementation of a Room Acoustics Modeling Software using Finite-Difference Time-Domain Methods". Presented at the 122nd Convention of the Audio Engineering Society, Vienna, Austria, May 5-8 2007.

[15] K.S. Kunz. *The finite difference time domain method for electromagnetism.* CRC Press. (1993).

[16] Dale R. Durran. *Numerical Methods for Wave Equations in Geophysical Fluid Dynamics.* Springer (1999).

[17] Stefan Bilbao. *Wave and Scattering Methods for Numerical Simulations.* Wiley (2004).

[18] Takatoshi Yokota, Shinichi Sakamoto, and Kideki Tachibana. "Visualization of sound propagation and scattering in rooms". Acoust. Sci. & Tech. 23, 1 (2002).

[19] Shinichi Sakamoto, Takuma Seimiya and Hideki Tachibana. "Visualization of sound reflection and diffraction using finite difference time domain method". Acoust. Sci. & Tech. 23, 1 (2002)

[20] Hisaharu Suzuki, Akira Omoto, and Kyoji Fujiwara. "Treatment of boundary conditions by finite difference time domain method". Acoust. Sci. & Tech. 28, 1 (2007).

[21] Nick Clark. "Tiny FDTD v1.0". Mathworks website. URL: http://www.mathworks.com/matlabcentral/fileexchange/21000-tiny-fdtd-v1-0.

[22] Philip M. Morse and K. Uno Ingard. *Theoretical Acoustics.* Princeton (1986).

[23] Chengbin Peng and M. Nafi Toksoz. "An optimal absorbing boundary condition for finite difference modeling of acoustic and elastic wave propagation". J. Acoust. Soc. Am. 95 (2), February 1994.

[24] Konrad Kowalczyk and Maarten van Walstijn. "Modelling Frequency-Dependent Boundaries as Digital Impedance Filters in FDTD and K-DWM Room Acoustics Simulations". Presented at the 124th Convention of the Audio Engineering Society, Amsterdam, The Netherlands, May 17-20 2008.

[25] Leo L. Beranek. *Acoustics*, 1993 edition. Acoustical Society of America.

[26] John Vanderkooy, Paul M. Boers and Ronald M. Aarts. "Direct-Radiator Loudspeaker Systems with High Bl". Presented at the 114th Convention of the Audio Engineering Society, Amsterdam, the Netherlands, 2003 March 22-25; revised 2003 May 26.

[27] John Vanderkooy. "The Acoustic Center: A New Concept for Loudspeakers at Low Frequencies". Presented at the 121st Convention of the Audio Engineering Society. San Francisco, CA, USA. 2006 October 5-8.

[28] Siegfried Linkwitz. "Investigation of Sound Quality Differences between Monopolar and Dipolar Woofers in Small Rooms". Presented at the 105th Convention of the Audio Engineering Society, San Francisco, CA, USA, 1998 September 26-29, paper #4786.

[29] Leo L. Beranek. *Acoustical Measurements: Revised Edition.* Acoustical Society of America (1988).

[30] M. R. Schroeder. "New method for measuring reverberation time". J. Acoust. Soc. Am., vol. 37, pp. 409-412, 1965.

[31] Laurent Faiget, Robert Ruiz and Claude Legros. "The True Duration of the Impulse Response used to Estimate Reverberation Time". IEEE manuscript 0-7803-3192-3/96, 1996.

[32] Allen D. Pierce. *Acoustics: An introduction to its Physical Principles and Applications.* Acoustical Society of America (1989).

[33] John R. Taylor. *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements (Second Edition).* University Science Books 1997.

[34] Neil W. Ashcroft & N. David Merman. *Solid State Physics.* Brooks Cole (1976).

[35] Thomas D. Rossing, Richard F. Moore, and Paul A. Wheeler. T*he Science of Sound (3rd edition).* Addison Wesley (2001).

[36] John Vanderkooy. "Multi-Source Room Equalization: Reducing Room Resonances". Presented at the 123rd Convention of the Audio Engineering Society, New York, NY, USA, 2007 October 5-8.

[37] Kane Yee. "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media". Antennas and Propagation, IEEE Transactions on 14: 302–307, 1996.

Appendix A

MATLAB Code

```
%this is the beginning of my new 5.1 test program
%this is a 3-d FDTD simulation
%This program will start off by exciting a pressure point of the
  system temporally

clear all; close all;
tic;

%Settings for plotting
Plot=0;           %1=plot, 0=no plot
Plotinterval=100; %determines the interval between interations to
  plot

%Settings for Single or Double Precision
chooseSingle=1;      %1=Single precision, 0=Double factor=1;

%Defining the time step, spacial step and speed of sound constants
   K=1/factor*1.25e-4;  %this corresponds to a sampling frequency
  of 8000Hz h=1/factor*.1;        %this is a spatial sampling of 10
  cm
c=344;                %speed of sound
rho=1.21;             %density of air
fs=1/K;

%n Sets number of time steps
n=8000*factor;
T=0:n;

%This sets the size of the room %assume a P matrix with MxNx0
```

```
M=42*factor+1;    %42width 1 to 42, thus 41 points always one
  bigger because there is not zero position, array starts at 1
N=56*factor+1;    %56depth
O=24*factor+1;    %24height

%This will add in a seperate room connected to the main room
sub_room_flag=0;
M_sub=50*factor;    %you are re-using a Ux from the main room so
  you don't need to account for 0 position being 1
N_sub=28*factor+1;
O_sub=0;%<----DO NOT CHANGE, I have not accounted for any other
  case

%This will be where we add a doorway
Rightside_doorway_flag=0;

%Activating the speakers you want to test
%1=active and all other values = deactive
sourceon=0;
centreon=0;
lefton=1;
righton=1;
rlefton=1;
rrighton=1;
subwooferon=0;

%defining the time delay and a relative amplitude for the rear
  speakers timedelay=130*factor;
amp=-0.85;
amp_RearLeft=amp;
amp_RearRight=amp;

%determins parameters for plotting with correct aspect ratios
if M>N
    largest_dim=M;
elseif N>M
    largest_dim=N;
elseif M==N
    largest_dim=M;
end

%pre-allocating arrays
P=zeros(M-1,N-1,O-1);
Ux=zeros(M,N-1,O-1);
Uy=zeros(M-1,N,O-1);
Uz=zeros(M-1,N-1,O);
```

```
67
68 if chooseSingle==1 %changes double arrays to single
69 P=single(P);
70 Ux=single(Ux);
71 Uy=single(Uy);
72 Uz=single(Uz);
73 end
74
75 %pre-allocating speakers
76 Pold_source=0;
77 Pold_centre=0;
78 Pold_left=0;
79 Pold_right=0;
80 Pold_rleft=0;
81 Pold_rleft=0;
82 Pold_subwoofer=0;
83
84 %pre-allocating the mic array %data stored as single precision to
     save space Psource=zeros(n+1,1,'single');
85 Pmic1=zeros(n+1,1,'single');
86 Pmic2=zeros(n+1,1,'single');
87 Pmic3=zeros(n+1,1,'single');
88 Pmicarray=zeros(5,5,n+1,'single');
89
90 %this will be for adding in the initial pulse
91 pulsewidth=80*factor; %in samples
92 offset=0;
93 Q=(0.5*(1-cos(2*pi*(T)/pulsewidth))).^2; dQ=(pi/pulsewidth)*sin(2*
     pi*(T)/pulsewidth).*(1-cos(2*pi*(T)/pulsewidth));
94
95 if chooseSingle==1
96 Q=single(Q);
97 end
98
99 for i=pulsewidth+1+offset:n+1;
100 Q(i)=0;
101 dQ(i+1)=0;
102 end
103
104 if chooseSingle==1
105 Q=single(Q);
106 end
107
108 Qfft=1/fs*fft(Q,n);
109 f=fs/n*(0:n/2);
110 figure(1)
```

```
111 Qplot=semilogx(f,20*log10(abs(Qfft(1:n/2+1))));
112 ylabel('[dB]');xlabel('Freq[Hz]');
113 grid on
114 axis([1 3000 -100 -30])
115 set(Qplot,'linewidth',2)
116 drawnow;
117
118 f=fs/n*(0:n-1);
119 Qfft2=f.*Qfft;
120 figure(2)
121 Qplot=semilogx(f(1:n/2+1),20*log10(abs(Qfft2(1:n/2+1))));
122 ylabel('[dB]');
123 xlabel('Freq [Hz]');
124 grid on axis([1 3000 -100 -30])
125 set(Qplot,'linewidth',2)
126 drawnow;
127
128 dQfft=1/fs*fft(dQ,n);
129 f=fs/n*(0:n/2);
130 figure(3)
131 dQplot=semilogx(f,20*log10(abs(dQfft(1:n/2+1))));
132 ylabel('[dB]');xlabel('Freq[Hz]');
133 grid on
134 axis([1 3000 -100 -30])
135 set(dQplot,'linewidth',2)
136 drawnow;
137
138 %this is only active with the sub-room switch
139 if sub_room_flag==1
140 P_sub=zeros(M_sub,N_sub-1,O_sub-1);
141 Ux_sub=zeros(M_sub,N_sub-1,O_sub-1);
142 Uy_sub=zeros(M_sub,N_sub,O_sub-1);
143 Uz_sub=zeros(M_sub,N_sub-1,O_sub);
144 Ux_subroom_old=Ux(M,1:N_sub-1,:);
145 new_dimx=(M-1+M_sub);
146 P_plot=50*ones(new_dimx,N-1);
147
148     if chooseSingle==1 %changes double arrays to single
149         P_sub=single(P_sub);
150         Ux_sub=single(Ux_sub);
151         Uy_sub=single(Uy_sub);
152         Uz_sub=single(Uz_sub);
153     end
154 end
155
156 %DOORWAY
```

```
157 if Rightside_doorway_flag==1
158 heightstart=1;%<----DO NOT CHANGE, I have not accounted for any
      other case heightfinish=20;
159 widthstart=17;
160 widthfinish=37;
161 Ux_doorway_old=Ux(M,widthstart:widthfinish,heightstart:
      heightfinish);
162 end
163
164 %determining the centre of the room for reference
165 roomcentrex=round((M)/2);
166 roomcentrey=round((N)/2);
167 roomcentrez=round((O)/2);
168
169 %dummy source position in centre of the room
170 xsource=roomcentrex;
171 ysource=roomcentrey;
172 zsource=roomcentrez;
173
174 %set Centre Channel position
175 xcentre=round(M/2);
176 ycentre=N-1;
177 zcentre=round(O/2);
178 move=0;
179
180 %set Right Channel Position
181 xright=(round((M)/4))-move;
182 yright=N-1;
183 zright=round(O/2);
184
185 %set Left Channel Position
186 xleft=M-xright+move;
187 yleft=N-1;
188 zleft=round(O/2);
189
190 %set Rear-Right Channel Position
191 xrright=xright;
192 yrright=1;
193 zrright=round(O/2);
194
195 %set Rear-Left Channel Position
196 xrleft=xleft;
197 yrleft=1;
198 zrleft=round(O/2);
199
200 %set Subwoofer Position
```

```
xsubwoofer=1;
ysubwoofer=1;
zsubwoofer=1;

%defining the absorption coefficients and wall impedances
alpha_right=.1;
alpha_left=.1;
alpha_front=.1;
alpha_back=.1;
alpha_roof=.1;
alpha_floor=.1;

Z_right=rho*c*((1+sqrt(1-alpha_right))/(1-sqrt(1-alpha_right)));
  Z_left=rho*c*((1+sqrt(1-alpha_left))/(1-sqrt(1-alpha_left)));
  Z_front=rho*c*((1+sqrt(1-alpha_front))/(1-sqrt(1-alpha_front)));
  Z_back=rho*c*((1+sqrt(1-alpha_back))/(1-sqrt(1-alpha_back)));
  Z_roof=rho*c*((1+sqrt(1-alpha_roof))/(1-sqrt(1-alpha_roof)));
  Z_floor=rho*c*((1+sqrt(1-alpha_floor))/(1-sqrt(1-alpha_floor)));
Z_open_doorway=rho*c;

startloopscheckflag=1
timestamp1=0
Pmicarray=zeros(5,5,n+1,'single'); %allocating mic array

%
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Setting the constants
%
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

UxleftwallU=(rho*h-K*Z_left)/(rho*h+K*Z_left);
UxleftwallP=2*K/(rho*h+K*Z_left);
UxrightwallU=(rho*h-K*Z_right)/(rho*h+K*Z_right);
UxrightwallP=2*K/(rho*h+K*Z_right); UxdoorwayU=(rho*h-K*
  Z_open_doorway)/(rho*h+K*Z_open_doorway); UxdoorwayP=2*K/(rho*h+K
  *Z_open_doorway);
UybackwallU=(rho*h-K*Z_back)/(rho*h+K*Z_back);
UybackwallP=2*K/(rho*h+K*Z_back);
UyfrontwallU=(rho*h-K*Z_front)/(rho*h+K*Z_front);
UyfrontwallP=2*K/(rho*h+K*Z_front);
UzfloorU=(rho*h-K*Z_floor)/(rho*h+K*Z_floor);
UzfloorP=2*K/(rho*h+K*Z_floor);
UzroofU=(rho*h-K*Z_roof)/(rho*h+K*Z_roof);
UzroofP=2*K/(rho*h+K*Z_roof);

```

```
for T=0:n;
%begin velocity iterations

%side walls
    %left hand wall
    Ux(1,:,:)=UxleftwallU*Ux(1,:,:)-UxleftwallP*P(1,:,:);
    %right hand wall
    Ux(M,:,:)=UxrightwallU*Ux(M,:,:)+UxrightwallP*P(M-1,:,:);
if Rightside_doorway_flag==1                Ux(M,widthstart:
  widthfinish,heightstart:heightfinish)=...
  UxdoorwayU*Ux_doorway_old+UxdoorwayP*...                    P(M
  -1,widthstart:widthfinish,heightstart:heightfinish);
                  Ux_doorway_old=Ux(M,widthstart:widthfinish,
  heightstart:heightfinish);          elseif sub_room_flag==1
                Ux(M,1:N_sub-1,:)=Ux_subroom_old+K/rho/h*(P(M-1,1:
  N_sub-1,:)-P_sub(1,1:N_sub-1,:));  Ux_subroom_old=Ux(M,1:N_sub
  -1,:);
end
%the rest
Ux(2:M-1,:,:)=Ux(2:M-1,:,:)+K/rho/h*(-diff(P(2-1:M-1,:,:),1,1));

%front and back walls
    %left hand wall
    Uy(:,1,:)=UybackwallU*Uy(:,1,:)-UybackwallP*P(:,1,:);
    %right hand wall              Uy(:,N,:)=UyfrontwallU*Uy(:,N,:)+
      UyfrontwallP*P(:,N-1,:);
    %the rest
    Uy(:,2:N-1,:)=Uy(:,2:N-1,:)+K/rho/h*(-diff(P(:,1:N-1,:),1,2));

%roof and floor
    %left hand wall
    Uz(:,:,1)=UzfloorU*Uz(:,:,1)-UzfloorP*P(:,:,1);
    %right hand wall
    Uz(:,:,O)=UzroofU*Uz(:,:,O)+UzroofP*P(:,:,O-1);
    %the rest
    Uz(:,:,2:O-1)=Uz(:,:,2:O-1)+K/rho/h*(-diff(P(:,:,1:O-1),1,3));

%now do the Pressure Matrix
P = P + c*c*rho*K/h*(-diff(Ux,1,1)-diff(Uy,1,2)-diff(Uz,1,3));

%this section adds in the speakers into the system with our
  defined Q
if sourceon==1              P(xsource,ysource,zsource)=K*c*c*rho/(h
  *h*h)*Q(T+1)+Pold_source+c*c*rho*K/h*(Ux(xsource,ysource,zsource)
  -Ux(xsource+1,ysource,zsource)+Uy(xsource,ysource,zsource)-Uy(
  xsource,ysource+1,zsource)+Uz(xsource,ysource,zsource)-Uz(xsource
```

```
  ,ysource,zsource+1));
end

if centreon==1              P(xcentre,ycentre,zcentre)=K*c*c*rho/(h*
  h*h)*Q(T+1)+Pold_centre+c*c*rho*K/h*(Ux(xcentre,ycentre,zcentre)-
  Ux(xcentre+1,ycentre,zcentre)+Uy(xcentre,ycentre,zcentre)-Uy(
  xcentre,ycentre+1,zcentre)+Uz(xcentre,ycentre,zcentre)-Uz(xcentre
  ,ycentre,zcentre+1));
end

if lefton==1              P(xleft,yleft,zleft)=K*c*c*rho/(h*h*h)*Q(T
  +1)+Pold_left+c*c*rho*K/h*(Ux(xleft,yleft,zleft)-Ux(xleft+1,yleft
  ,zleft)+Uy(xleft,yleft,zleft)-Uy(xleft,yleft+1,zleft)+Uz(xleft,
  yleft,zleft)-Uz(xleft,yleft,zleft+1));
end

if righton==1           P(xright,yright,zright)=K*c*c*rho/(h*h*h)*Q
  (T+1)+Pold_right+c*c*rho*K/h*(Ux(xright,yright,zright)-Ux(xright
  +1,yright,zright)+Uy(xright,yright,zright)-Uy(xright,yright+1,
  zright)+Uz(xright,yright,zright)-Uz(xright,yright,zright+1));
end

if rlefton==1 && T>=timedelay          P(xrleft,yrleft,zrleft)=K*c
  *c*rho/(h*h*h)*amp_RearLeft*Q(T+1-timedelay)+Pold_rleft+c*c*rho*K
  /h*(Ux(xrleft,yrleft,zrleft)-Ux(xrleft+1,yrleft,zrleft)+Uy(xrleft
  ,yrleft,zrleft)-Uy(xrleft,yrleft+1,zrleft)+Uz(xrleft,yrleft,
  zrleft)-Uz(xrleft,yrleft,zrleft+1));
end

if rrighton==1 && T>=timedelay           P(xrright,yrright,zrright)
  =K*c*c*rho/(h*h*h)*amp_RearRight*Q(T+1-timedelay)+Pold_rright+c*c
  *rho*K/h*(Ux(xrright,yrright,zrright)-Ux(xrright+1,yrright,
  zrright)+Uy(xrright,yrright,zrright)-Uy(xrright,yrright+1,zrright
  )+Uz(xrright,yrright,zrright)-Uz(xrright,yrright,zrright+1));
end

if subwooferon==1           P(xsubwoofer,ysubwoofer,zsubwoofer)=K*c
  *c*rho/(h*h*h)*Q(T+1)+Pold_subwoofer+c*c*rho*K/h*(Ux(xsubwoofer,
  ysubwoofer,zsubwoofer)-Ux(xsubwoofer+1,ysubwoofer,zsubwoofer)+Uy(
  xsubwoofer,ysubwoofer,zsubwoofer)-Uy(xsubwoofer,ysubwoofer+1,
  zsubwoofer)+Uz(xsubwoofer,ysubwoofer,zsubwoofer)-Uz(xsubwoofer,
  ysubwoofer,zsubwoofer+1));
end

%grabs and stores old source term data
Pold_source=P(xsource,ysource,zsource);
```

```
Pold_centre=P(xcentre,ycentre,zcentre);
Pold_left=P(xleft,yleft,zleft);
Pold_right=P(xright,yright,zright);
Pold_rleft=P(xrleft,yrleft,zrleft);
Pold_rright=P(xrright,yrright,zrright); Pold_subwoofer=P(
  xsubwoofer,ysubwoofer,zsubwoofer);

%Section that works out the sub-room
if sub_room_flag==1
Ux_sub(M_sub,:,:)=(rho*h-K*Z_right)/(rho*h+K*Z_right)*Ux_sub(M_sub
  ,:,:)+2*K/(rho*h+K*Z_right)*P_sub(M_sub,:,:);
Ux_sub(1:M_sub-1,:,:)=Ux_sub(1:M_sub-1,:,:)+K/rho/h*(-diff(P_sub
  (1:M_sub,:,:),1,1));
%front and back walls
    %left hand wall
    Uy_sub(:,1,:)=(rho*h-K*Z_back)/(rho*h+K*Z_back)*Uy_sub(:,1,:)
      -2*K/(rho*h+K*Z_back)*P_sub(:,1,:);
    %right hand wall
    Uy_sub(:,N_sub,:)=(rho*h-K*Z_front)/(rho*h+K*Z_front)*Uy_sub
      (:,N_sub,:)+2*K/(rho*h+K*Z_front)*P_sub(:,N_sub-1,:);
    %the rest
    Uy_sub(:,2:N_sub-1,:)=Uy_sub(:,2:N_sub-1,:)+K/rho/h*(-diff(
      P_sub(:,1:N_sub-1,:),1,2));

%roof and floor
    %left hand wall
    Uz_sub(:,:,1)=(rho*h-K*Z_floor)/(rho*h+K*Z_floor)*Uz_sub
      (:,:,1)-2*K/(rho*h+K*Z_floor)*P_sub(:,:,1);
    %right hand wall
    Uz_sub(:,:,O_sub)=(rho*h-K*Z_roof)/(rho*h+K*Z_roof)*Uz_sub
      (:,:,O_sub)+2*K/(rho*h+K*Z_roof)*P_sub(:,:,O_sub-1);
    %the rest
    Uz_sub(:,:,2:O_sub-1)=Uz_sub(:,:,2:O_sub-1)+K/rho/h*(-diff(
      P_sub(:,:,1:O_sub-1),1,3));
    P_sub(1,:,:)=P_sub(1,:,:)+c*c*rho*K/h*(Ux(M,1:N_sub-1,1:O_sub
      -1)-Ux_sub(1,:,:)-diff(Uy_sub(1,:,:),1,2)-diff(Uz_sub(1,:,:)
      ,1,3));
    P_sub(2:M_sub,:,:)=P_sub(2:M_sub,:,:) + c*c*rho*K/h*(-diff(
      Ux_sub,1,1)-diff(Uy_sub(2:M_sub,:,:),1,2)-diff(Uz_sub(2:M_sub
      ,:,:),1,3));
end

%collecting mic data
%Source is the geometrical centre of the room
Psource(T+1)=P(xsource,ysource,zsource);
Pmic1(T+1)=P(15*factor,15*factor,zsource);
```

```
Pmic2(T+1)=P(15*factor,20*factor,zsource);
Pmic3(T+1)=P(20*factor,15*factor,zsource);
Pmic4(T+1)=P(20*factor,20*factor,zsource);

%mic array a=5*factor;
%width spacing in h cm increments b=5*factor; %depth spacing
%defining the mic array based on the approximate centre of the
  room
for i=-2:2
    for j=-2:2
        Pmicarray(3+i,3+j,T+1)=P(roomcentrex-i*a,roomcentrey-j*b,
          roomcentrez);      end
end

%plotting commands if rem(T,Plotinterval) == 0
if sub_room_flag==1
    P_plot(1:M-1,1:N-1)=P(:,:,zsource);             P_plot(M:M-1+
      M_sub,1:N_sub-1)=P_sub(:,:,zsource);
else
    P_plot=P(:,:,zsource);
end

timestamp2=toc;
howmuchdone=T/n*100
timeperiteration=(timestamp2-timestamp1)/Plotinterval
timestamp1=timestamp2

if Plot==1
    if chooseSingle==1
    P_plot=double(P_plot);
    end
figure(4)
surf(P_plot);
title('pressure');ylabel('x-axis');xlabel('y-axis');zlabel('
  amplitude')
axis([0 largest_dim 0 largest_dim -50 150]); drawnow;
end
end
end

%plotting mic and source responses for T=0:n
t(T+1)=(T*K);
end

figure(5)
plot(t,Psource);
```

```
366 title('Pressure at the Centre Point');
367 ylabel('pressure amplitude');xlabel('time (s)'); %axis([0 .7 0
    170]) %grid on
368
369 figure(6)
370 plot(t,Pmic1);
371 title('Pressure at the 1st mic');
372 ylabel('pressure amplitude');xlabel('time (s)');
373
374 figure(7)
375 plot(t,Pmic2);
376 title('Pressure at the 2nd Mic');
377 ylabel('pressure amplitude');xlabel('time (s)');
378
379 figure(8)
380 plot(t,Pmic3);
381 title('Pressure at the 3rd Mic');
382 ylabel('pressure amplitude');xlabel('time (s)');
383
384 figure(9)
385 plot(t,Pmic4);
386 title('Pressure at the 4th Mic');
387 ylabel('pressure amplitude');xlabel('time (s)');
388
389 Pmic1fft=1/fs*fft(Pmic1.*Pmic1,n);
390 figure(10)
391 Pmic1plot=plot(f,(abs(Pmic1fft(1:n/2+1))));
392 ylabel('[dB]');xlabel('Freq [Hz]');
393 grid on
394 axis([0 50 0 1200])
395 set(Pmic1plot,'linewidth',2)
396
397 Pmic1fft=1/fs*fft(Pmic1,n);
398 figure(100)
399 Pmic1plot=plot(f,(20*log10(Pmic1fft(1:n/2+1))));
400 %title('Frequency Response in Groh Room from a sub-woofer in the
    corner with a 10msec pulse','FontSize',20)
401 ylabel('Pressure [dB]','FontSize',16);xlabel('Frequency [Hz]','
    FontSize',16);
402 grid on
403 axis([0 200 -50 10])
404 legend('Un-Smoothed'); set(Pmic1plot,'linewidth',3)
405
406 Pmic2fft=1/fs*fft(Pmic2,n);
407 figure(200)
408 Pmic2plot=plot(f,(20*log10(Pmic2fft(1:n/2+1))));
```

```
%title('Frequency Response in Groh Room from a sub-woofer in the
  corner with a 10msec pulse','FontSize',20)
ylabel('Pressure [dB]','FontSize',16);xlabel('Frequency [Hz]','
  FontSize',16);
grid on
axis([0 200 -50 10])
legend('Un-Smoothed'); set(Pmic2plot,'linewidth',3)

Pmic3fft=1/fs*fft(Pmic3,n);
figure(300)
Pmic3plot=plot(f,(20*log10(Pmic3fft(1:n/2+1))));
%title('Frequency Response in Groh Room from a sub-woofer in the
  corner with a 10msec pulse','FontSize',20)
ylabel('Pressure [dB]','FontSize',16);xlabel('Frequency [Hz]','
  FontSize',16);
grid on
axis([0 200 -50 10])
legend('Un-Smoothed'); set(Pmic3plot,'linewidth',3)

Pmic4fft=1/fs*fft(Pmic4,n);
figure(400)
Pmic4plot=plot(f,(20*log10(Pmic4fft(1:n/2+1))));
%title('Frequency Response in Groh Room from a sub-woofer in the
  corner with a 10msec pulse','FontSize',20)
ylabel('Pressure [dB]','FontSize',16);xlabel('Frequency [Hz]','
  FontSize',16);
grid on
axis([0 200 -50 10])
legend('Un-Smoothed'); set(Pmic4plot,'linewidth',3)

%%%Johns schroeder plot
Pintegral=0.002; %change this to straighten Schroeder plot
  Pschroeder(n+1)=Pintegral;
    for i=1:n
        Pintegral=Psource(n+1-i).^2 + Pintegral;
        Pschroeder(n+1-i)=Pintegral;
    end
Pschroeder=10*log10(Pschroeder);
figure(11)
plot(t,Pschroeder);
grid on
title('Schroeder decay plot');
ylabel('SPL [dB]');xlabel('time (s)');
axis([0 1 -20 70])
%%%end of johns shroeder plot

```

```
448 %beginning of my own reverberation method
449 figure(12)
450 Reverbtime(:,1)=t;
451 Reverbtime(:,2)=10*log10(Psource.^2);
452 plot(Reverbtime(:,1),Reverbtime(:,2));
453 axis([0 1 -100 40]);
454 [X Y]=findpeaks(Reverbtime(:,2));
455 Y=Y*K;
456
457 figure(13)
458 plot(Y,X);
459 drawnow
460 %end of my reverberation
461
462 fexpected=1/(pulsewidth*K)
463 load chirp;
464 sound(y,Fs)
465
466 %figure(4)
467 %surf(P);
468 %title('pressure');ylabel('time');xlabel('position');zlabel('
    amplitude') toc
469
470
471 %% This version of Demerit uses constant band smoothing
472 tic
473 load 'PmicarrayGrohReference10cm' flag=0;
474 bandwidth=60;   %for now this is just the number of points. If
    Fres=1 then it is the bandwidth
475 cuttoff=100;    %this is the cuttoff frequency where the demerit
    calculation stops
476 halfbandwidth=round(bandwidth/2);
477 realbandwidth=2*halfbandwidth+1
478
479 %% lets begin by extracting one mic out of the mic array
480 for i=1:5
481     for j=1:5
482         tempequalized=Pmicarray(6-i,6-j,:);
              tempequalized=squeeze(tempequalized);
              tempunequalized=PmicarrayGrohReference10cm(6-i,6-j,:);
                              tempunequalized=squeeze(tempunequalized);
483         n1=length(tempequalized)-1; n2=length(tempunequalized)-1;
484     if n1 ~= n2
485         return
486     else
487         n=n1;
```

```
488     end
489 Y=1/fs*fft(tempequalized,n);%./transpose(dQfft); X=1/fs*fft(
      tempunequalized,n);%./transpose(dQfft);
490 flag=flag+1;
491 f=fs/n*(0:n/2);
492
493 for k=1:n/2+1
494 low_bin=k-halfbandwidth;
495 high_bin=k+halfbandwidth;
496     if low_bin<1
497         power_sum=sum((X(1:high_bin).^2))+sum((X(1:abs(low_bin)+1)
              .^2));
498     else
499         power_sum=sum((X(low_bin:high_bin).^2));
500     end
501     orig_smooth_power(k)=power_sum/(high_bin-low_bin+1);
502 end
503
504 UnSmoothCollector(flag,:)=(X);
505 SmoothCollector(flag,:)=orig_smooth_power;
506
507 for k=1:n/2+1
508     low_bin=k-halfbandwidth;
509     high_bin=k+halfbandwidth;
510     if low_bin<1
511         power_sum2=sum((abs(Y(1:high_bin)).^2))+sum((abs(Y(1:abs(
              low_bin)+1).^2)));
512     else
513         power_sum2=sum((abs(Y(low_bin:high_bin)).^2));
514     end
515     orig_smooth_power2(k)=power_sum2/(high_bin-low_bin+1);
516 end
517 UnSmoothCollector2(flag,:)=Y;
518 SmoothCollector2(flag,:)=orig_smooth_power2;
519
520 figure(flag+13) h=plot(f,20*log10(abs(Y(1:n/2+1))),'g',f,10*log10(
      orig_smooth_power2(1:n/2+1)),'c');
521 %f,20*log10(abs(X(1:n/2+1))),f,10*log10(orig_smooth_power(1:n/2+1)
      ),'r', %title('Frequency Response in Groh Room from appling the
      Rear Cancellation Technique with a 10msec pulse','FontSize',20)
522 ylabel('Original & Smoothed [dB]','FontSize',16);xlabel('Freq [Hz]
      ','FontSize',16);
523 grid on axis([0 200 -50 10]) legend('Un-Smoothed','Smoothed');
524 set(h,'linewidth',3)
525
```

```
526 P_uneq_unsmoothed=transpose((abs(X(1:n/2+1))).^2); %check that
    matrix dimensions match
527 P_uneq_smoothed=(orig_smooth_power(1:n/2+1)); P_eq_unsmoothed=
    transpose((abs(Y(1:n/2+1))).^2); P_eq_smoothed=(
    orig_smooth_power2(1:n/2+1));
528 Fres=fs/n;  %the frequency resolution of the FFT
529 N=round(cuttoff/Fres);  %determines the #of bins between 0Hz and
    200Hz
530 Demerit_eq1(flag)=1*(1/N)*sum(((abs((P_eq_unsmoothed(1:N+1)-
    P_eq_smoothed(1:N+1))./P_eq_smoothed(1:N+1)))));
531 Demerit_eq2(flag)=1*(1/N)*sum(((((P_eq_unsmoothed(1:N+1)-
    P_eq_smoothed(1:N+1))./P_eq_smoothed(1:N+1)).^2)));
532     end
533 end
534 Mean_eq1 = mean(Demerit_eq1)
535 Mean_eq2 = mean(Demerit_eq2)
536 std_eq1 = std(Demerit_eq1)
537 std_eq2 = std(Demerit_eq2)
538 sdom_eq1 = std_eq1/sqrt(N)
539 sdom_eq2 = std_eq2/sqrt(N)
540
541 %% Averaged Smoothing
542 flag=0;
543 for i=1:25
544 flag=flag+1;
545     for j=1:4001
546         SmoothSum(j)=sum(SmoothCollector(:,j))/25;
              SmoothSum2(j)=sum(SmoothCollector2(:,j))/25;
547     end
548 figure(50) h=plot(f,20*log10(abs(UnSmoothCollector2(flag,(1:n/2+1)
    ))),'g',f,10*log10(SmoothSum2(1:n/2+1)),'c');
549 ylabel('Original & Smoothed [dB]');xlabel('Freq [Hz]');
550 axis([0 200 -50 10])
551 %semilogx(f,10*log10(orig_smooth_power(1:n/2+1)),'r',f,20*log10(
    abs(Y(1:n/2+1))),'g');ylabel('Original & Smoothed [dB]');xlabel('
    Freq [Hz]');
552 grid on
553 set(h,'linewidth',2)
554
555 P_eq_unsmoothed=((abs(UnSmoothCollector2(flag,(1:n/2+1)))).^2);
    P_eq_smoothed=(SmoothSum2(1:n/2+1));
556 Fres=fs/n;  %the frequency resolution of the FFT
557 N=round(cuttoff/Fres);  %determines the #of bins between 0Hz and
    200Hz
558 Demerit_eq1(flag)=1*(1/N)*sum(((abs((P_eq_unsmoothed(1:N+1)-
    P_eq_smoothed(1:N+1))./P_eq_smoothed(1:N+1)))));
```

```
559 Demerit_eq2(flag)=1*(1/N)*sum(((((P_eq_unsmoothed(1:N+1)-
      P_eq_smoothed(1:N+1))./P_eq_smoothed(1:N+1)).^2)));
560 end
561 toc
562
563 Mean_eq1 = mean(Demerit_eq1)
564 Mean_eq2 = mean(Demerit_eq2)
565 std_eq1 = std(Demerit_eq1)
566 std_eq2 = std(Demerit_eq2)
567 sdom_eq1 = std_eq1/sqrt(N)
568 sdom_eq2 = std_eq2/sqrt(N)
```