Efficient Acoustic Modelling of Large Spaces using Time Domain Methods

Simon Durbridge

May 10, 2017

Contents

Chapter 1

Introduction

Modelling and replicating the effects of acoustic systems has been of continued interest to a number of industries, from the creation of scale models for concert halls to the architecture of individualised audio in video games. Acoustic modelling is not only a tool for those wishing to design acoustic systems, but may be of increasing interest for those wishing to experience an acoustic system[1] while in another environment. Simulating the acoustic behaviour of large systems with multiple sources and receivers may not be a trivial undertaking, with significant computational resources required to model such systems.

1.1 Problem Definition

Current commercially available acoustic modelling tools for large (cathedrals, arenas, video game maps etc) electro acoustic simulations rely on assumptions based around plane wave propagation. These models are only accurate when assuming that detail of the domain features are significantly larger than the wavelengths of interested, and that no diffraction effects occur. These ray based methods such as the image source method, ray or beam tracing methods approximate the performance of the system and simulation domain without solving the full physics of an acoustic system. These methods have been successfully used to approximate large systems at mid and high frequencies, but may not accurately simulate low frequency wave propagation and wave interaction.

Wave based acoustic modelling methods have been previously implemented in commercial software packages with great success, and are often used to model complex acoustic systems such as loudspeakers and other transducers. However, these packages are difficult to apply to simulating large domain problems, due to some limitations with numerical solutions to wave based methods. This is due in part to the extreme memory requirements of simulating large domains, but also due to the significant computation time required to 'crunch the numbers' and solve for adequate amounts of time. One such package known as Comsol that is a finite element solver, has recently added ray tracing to the internal tools, potentially to accommodate this restriction.

[1] In this context we consider an acoustic system as whole i.e. sound sources, receivers, the propagation medium and the domain boundaries

1.2 Aim of the Study

The aim of this study is to test three numerical methods of solving the acoustic wave equation that result in time domain solutions. These methods will be tested for speed of execution with consecutively large domains. The methods of interested in this studying are the finite-difference time domain method, the sparse finite difference time domain method and the pseudo-spectral time domain method. The outcome will be the identification of a method that may warrant further optimisation and could be potentially used for real-time solving in the future.

1.3 Format of the Report

At first we will discuss the acoustic wave equation, and some properties of acoustics that we might expect to approximate in a wave equation solving model. Following this, we will introduce the finite difference time domain and pseudo-spectral time domain methods for solving the acoustic wave equation, and we will discuss semi-empirical partially absorbing boundary conditions applied to these methods. While discussing the finite difference time domain method, we will introduce the idea of the sparse finite difference time domain method. Finally We will attempt to validate the models as they were developed, and review the execution speed of each method. We will then discuss some concepts that may significantly improve the execution of these methods.

Chapter 2

Acoustic Principals

Acoustics is a branch of physics[1] that aims to characterise Newton's law of motion applied to mechanical wave propagation, while obeying the physical conservation law and often focussing on propagation in an audible spectrum. This characterisation of sound propagation is intrinsically linked to many other disciplines of science and engineering, as well as psychological and perceptual study. In this section we will review the acoustic wave equation, and discuss some properties of interest in acoustic modelling.

2.1 The Acoustic Wave Equation

In the Mcgraw-Hill Electronic and Electrical Engineering Series of books, the late Leo Beranek authored the Acoustics volume [1]. This volume contains an elegant summary of the wave equation, that will be the subject of paraphrase in the following section.

Acoustic waves are classified as fluctuations of pressure in a given medium, manifesting as longitudinal waves of high and low pressure and density of air molecules. Often these fluctuations are cyclical in nature around an ambient pressure, though jets are often described in aeroacoustic study. Similar to the behaviour of heat convection or fluid diffusion, these cyclical fluctuations propagate and spread through the medium of interest, converging towards an entropic steady state. It is possible to calculate an approximate solution to the propagation of pressure through a space, by solving a system of second order partial differential equations that can be collected into a 'Wave Equation'. Below, we will introduce the three building blocks of the wave equation in both one dimension, and three dimensions (based on vector notation). These building blocks are Newton's Second Law of Motion, the gas law, and the laws of conservation of mass.

To consider the wave equation, we should use the analogy of a small[2] volume of gas, within a larger homogeneous medium. The faces of the volume are frictionless, and only the pressure at any face impacts on the gas inside the volume.

[1] though often considered to be interdisciplinary
[2] rectilinear

One Dimension Standard	Three Dimension Vector
Sound pressure p propagates across the medium like a plane wave, from one side to the other in the x direction at a rate equal to the change in space $\frac{\delta p}{\delta x}$	Sound pressure p propagates across the medium like a spherical wave, from one side to the other at a rate of **grad** $p = \mathbf{i}\frac{\delta p}{\delta x} + \mathbf{j}\frac{\delta p}{\delta y} + \mathbf{k}\frac{\delta p}{\delta z}$ where i, j and k are unit vectors in the directions x, y and z.
Force acting on the volume in the positive x direction can thus be described as $-(\frac{\delta p}{\delta x}\Delta x)\Delta y\Delta z$	Force acting on the volume in the positive x direction can thus be described as $-[i(\frac{\delta p}{\delta x}\Delta x)\Delta y\Delta z) + j(\frac{\delta p}{\delta y}\Delta y)\Delta x\Delta z) + k(\frac{\delta p}{\delta z}\Delta z)\Delta x\Delta y)]$
A positive gradient causes acceleration in the $-x$ direction	$\leftarrow$
Force per unit volume is given by dividing both sides of the previous equation by the volume V, $\frac{f}{V} = -\frac{\delta p}{\delta x}$	Force per unit volume is given by dividing both sides of the previous equation by the volume V, $\frac{f}{V} = -\mathbf{grad}p$
Newton's second law of motion dictates that the rate of change of momentum in the volume must balance with force per unit volume, and we can assume the mass of gas in the volume is constant.	$\leftarrow$
The force mass balance can be described as $\frac{f}{V} = -\frac{\delta p}{\delta x} = \frac{M}{M}\frac{\delta u}{\delta t} = \rho'\frac{\delta u}{\delta t}$	The force mass balance can be described as $\frac{f}{V} = -\mathbf{grad}p = \frac{M}{M}\frac{Dq}{Dt} = \rho'\frac{Dq}{Dt}$
u is the velocity of gas in the volume, ρ' is the density of the gas, and $M = \rho' V$ is the mass of gas in the volume.	where q is the vector velocity, ρ' is the density of gas in the volume, $M = \rho' V$ is the total mass of gas in the volume. $\frac{D}{Dt}$ represents the total rate of change of velocity of a section of gas in the volume, and can be composed as $\frac{Dq}{Dt} = \frac{\delta q}{\delta t} + q_x\frac{\delta q}{\delta x} + q_y\frac{\delta q}{\delta y} + q_z\frac{\delta q}{\delta z}$ where q_x, q_y and q_z are the components of the particle velocity q in each direction. As this is a linear wave equation approximation, these velocity components have no cross terms.

If the change in density of gas in the volume is sufficiently small, the ρ' will be approximately equal to the average density ρ_0, thus simplifying the equations above to $-\frac{\delta p}{\delta x} = \rho_0 \frac{\delta u}{\delta t}$	If the particle velocity vector is sufficiently small, the change of momentum of the gas is approximately the same as the momentum of the volume at any arbitrary point, and the density of gas within the volume ρ' will be approximately equal to the average density ρ_0. Thus the above can be written as $-grad\, p = \rho_0 \frac{\delta q}{\delta t}$
This kind of approximation may be appropriate as long as the maximum pressure is appropriately low, so that the behaviour of the air is linear, often quoted to be at or under the threshold of pain for human hearing or 120dB SIL.	$\leftarrow$
$\rightarrow$	Assuming that the gas of the volume is ideal, then the gas law $PV = RT$ should hold true. Here, T is the temperature in degrees Kelvin, and R is a constant based on the mass of the gas. For this approximation we assume that the system is adiabatic, and that T and R are lumped into a gas constant which for air is $\gamma = 1.4$.
In differential form, the relationship between pressure and volume for an adiabatic expansion the volume is $\frac{dP}{P} = \frac{-\gamma dV}{V}$ i.e. changes in pressure scale with changes in volume by this γ value.	$\leftarrow$
$\rightarrow$	If perturbations in pressure and volume due to a sound wave, p for pressure and τ for volume respectively, are sufficiently small compared to the rest values P_0 and V_0; the time based derivative of the above equation can be written as follows: $\frac{1}{P_0}\frac{\delta p}{\delta t} = \frac{-\gamma}{V_0}\frac{\delta \tau}{\delta t}$

As the wave equation being derived is concerned with the transport of pressure within a volume, a continuity expression must be applied. The conservation of mass states that the total mass of gas in the volume must remain constant. This conservation law brings a unique relationship between discrete velocities at the boundary of the volume:	$\leftarrow$
If the volume is displaced by some rate ε_x, air particles at either boundary of the volume must be displaced at an equal rate for the mass of the volume to remain constant. As such if the left side of the volume is displaced with a velocity, in a given time step the particles at the right hand boundary must also be displaced. This can be written as $\varepsilon_x + \frac{\delta\varepsilon_x}{\delta x}\Delta x$ The difference between this velocity and a subsequent change in volume τ multiplied by the volume gives $\tau = V_0 \frac{\delta\varepsilon_x}{\delta x}$.	If the mass of gas within the box must remain constant, the vector displacement will directly change the volume by some rate, as the two must balance to satisfy the continuity equation. This can be written as $\tau = V_0\ div\ \varepsilon$
Differentiating this with respect to time gives: $\frac{\delta\tau}{\delta t} = V_0 \frac{\delta u}{\delta x}$ where u is the instantaneous particle velocity	Differentiating this with respect to time gives: $\frac{\delta\tau}{\delta t} = V_0\ div\ q$ where q is the instantaneous particle velocity
The one dimensional wave equation in rectangular coordinates can be created by combining the above statements about the equation of motion, the gas law and the continuity equation. The combination of the gas law and continuity equation gives $\frac{\delta p}{\delta t} = -\gamma P_0 \frac{\delta u}{\delta x}$	The three dimensional wave equation in rectangular coordinates can be created by combining the above statements about the equation of motion, the gas law and the continuity equation. The combination of the gas law and continuity equation gives $\frac{\delta p}{\delta t} = -\gamma P_0 div\mathbf{q}$
When differentiated with respect to time, this gives: $\frac{\delta^2 p}{\delta t^2} = -\gamma P_0 \frac{\delta^2 u}{\delta t \delta x}$	When differentiated with respect to time this gives: $\frac{\delta^2 p}{\delta t^2} = -\gamma P_0 div \frac{\delta q}{\delta t}$

Differentiating the momentum equation derived above with respect to time gives $-\frac{\delta^2 p}{\delta t^2} = \rho_0 \frac{\delta^2 u}{\delta x \delta t}$	The divergence of the momentum equation derived above gives: $-div = \rho_0 div \frac{\delta q}{\delta t}$ Replacing the divergence $(grad p)$ term with the Lapacian operator$\nabla^2 p$ produces $-\nabla^2 p = \rho_0 div \frac{\delta^2 p}{\delta t}$
Combining the above equations gives: $\frac{\delta^2 p}{\delta x^2} = \frac{\rho_0}{\gamma P_0} \frac{\delta^2 p}{\delta t^2}$	Combining the above equations gives: $\nabla^2 p = \frac{\rho_0}{\gamma P_0} \frac{\delta^2 p}{\delta t^2}$
If we define c as the speed of propagation in the medium of interest, then $c^2 \approx \frac{\gamma P_0}{\rho_0}$ due to the fact that the speed of sound $c \approx (1.4 \frac{10^5}{1.18})^{\frac{1}{2}}$ where the ambient air pressure at sea level is $10^5 Pa$, 1.4 is the adiabatic constant γ (ratio of specific heats) for air, and ρ_0 is the density of air is approximately $1.8 kg/m^3$	$\leftarrow$
Finally we find that the 1 dimensional wave equation is: $\frac{\delta^2 p}{\delta x^2} = \frac{1}{c^2} \frac{\delta^2 p}{\delta t^2}$	Finally we find that the 3 dimensional wave equation is: $\nabla^2 p = \frac{1}{c^2} \frac{\delta^2 p}{\delta t^2}$ An explicit 3 dimensional expression of the pressure component of this equation is: $\nabla^2 p = \frac{\delta^2 p}{\delta x^2} + \frac{\delta^2 p}{\delta y^2} + \frac{\delta^2 p}{\delta z^2}$
This equation can also be expressed in terms of the instantaneous velocity in the volume as: $\frac{\delta^2 u}{\delta x^2} = \frac{1}{c^2} \frac{\delta^2 u}{\delta t^2}$	This equation can be expressed velocity vector $\nabla^2 q = \frac{1}{c^2} \frac{\delta^2 q}{\delta t^2}$ where $\nabla^2 q$ represents the gradient of pressure (velocity) in the volume.

In the above table we have derived wave equations, with forms of velocity and pressure as the independent variables. We have also shown that pressure, velocity, displacement and density are related within the system of equations, by differentiating and integrating with respect to space and time. As these forms of the wave equation are intrinsically coupled, it is possible to leverage this coupling when generating a numerical solution to the wave equation. It is also important to note that a significant number of assumptions have been taken when deriving these equations, and any solution to these equations may only be accurate when simulating a loss free, frictionless, homogeneous, ideal gas medium, where all perturbations are sufficiently small and fast that it is possible to reduce the complexity of the system.

2.2 Acoustic Properties of Interest in This Study

Now that we have an understanding of the mathematics behind sound propagation from the wave equation, it is important to have an understanding of what acoustic phenomena can be observed through solving the wave equation. In the next section we shall discuss three components of acoustics behaviour, two of which are intrinsic to the acoustics of rooms and one is more general.

2.2.1 Inverse Square Law & Propagation time

As previously noted, sound propagates as longitudinal waves through a medium such as air or water. These waves are often conceptualised as simple rays [1] travelling through a space[3], much like planar waves. However, the properties of a sound source such as the directivity and shape can have a significant effect on the behaviour of sound wave propagation. An example of this is the difference in energy spread over distance for theoretically ideal point and line sources. Ideal point sources that propagate sound omni-directionally obey the inverse square law and propagate sound spherically, and ideal line sources do not as they propagate sound cylindrically. The inverse square law is sound propagations is defined as:

$$I = \frac{P}{4\pi r^2}$$

Where I is the intensity over the area of the sphere, P is the propagated energy at the source and r is the radius of the sphere i.e. the distance between the source and the point of inquiry. This equation denotes that as an acoustic pressure wave radiates outward like an expanding sphere, as the area of the surface of the sphere increase the energy-per-unit-area of the surface of the sphere decreases. That is, as r increases, I decreases, assuming P is a constant pressure of interest.
As ideal line sources propagate pressure waves cylindrically, the equation above can be modified to account for this change:

$$I = \frac{P}{2\pi r}$$

These two similar equations show a change in relationship between acoustic power over distance for different acoustic sources. If you were to evaluate the change of I for different values of r with the point source equation, you would find that as r is doubled I decreases by $6dB$. Doing the same for the line source equation would yield a $3dB$ change. This is in part due to the fact that we assume the cylinder is infinitely long, and thus we are evaluating a 2D simplification of a 3D problem. The area of the cylinder for any value of r is less than the equivalent sphere, and so the theoretical distribution of energy is also reduced. This may be an important concept when considering the 2D approximation of 3D simulations in acoustic studies. Below is a graph showing the

[3] It may be appropriate to often consider space to be 3 dimensional (3D), or a lower order approximation of a 3D space

difference between intensity over distance for an ideal point and line source:

Although the wave equation considered and solved in this study is lossless i.e. we do not consider viscous or thermal losses in the basic linearised acoustic wave equation, we would expect to see a reduction in absolute pressure between a source and receiver. As sound travels at some finite distance over time c, we would also expect to see a uniform time between a wave being radiated from a source, and being recorded at some receiver location for all simulation methods[4].

2.2.2 Reverberation

For this study we will consider spaces or domains of finite size. These domains have boundaries, and those boundaries will either absorb or reflect sound waves by some ratio. In acoustic engineering the proportion of sound energy absorbed or reflected by a material is often described as an absorption or reflection coefficient α, which is often expressed as a normalised value between 0 (totally reflecting) and 1 (totally absorbing) [1].
If a sound source propagates a signal of appropriate speed and amplitude, the sound wave will reach the boundaries and be partially absorbed or reflected in reciprocal directions. These reflections will continue to scatter and will eventually decay beyond audibility. The reverberant sound field is the steady-state of diffusely scattered sound energy (reflections), due to perturbation by a sound source in bounded space. The amplitude of the sound source and diffusely scattered reflections balance with the rate of decay (diffusion and absorption) of the sound field [2].

Acoustic Absorption

The decay rate of a reverberant sound field is often quantified by the time taken for a steady state sound field to reduce in level by $60dB$, once the sound source has finished propagating. This is defined as the RT_{60} of the domain, and was first proposed by WC Sabine in 1900 [2]. There have been a multitude of expansions on Sabines original formula, notably in this study by Eyring, who expanded the denominator of the reverberation time equation to calculate more realistically for average absorption values above 0.1. The Eyring reverberation time equation is as follows:

$$T = 0.161 \frac{V}{-\ln(1-\alpha)S}$$

Where S is the surface area, V is the volume of the domain and α is the average absorption coefficient and can be calculated as such:

$$\alpha = \left(\left(S_{leftwall}\alpha_{leftwall}\right) + \left(S_{rightwall}\alpha_{rightwall}\right) + ...\right) / S_{total}$$

[4]For an interesting review of the relationship between 1D, 2D and 3D sound propagation being derivative, please see the appendices

The use of RT_{60} as the preferred metric of decay time is valid, assuming that the acoustics system is linear and time-invariant. A more comprehensive description of reverberation and overview of the associated parameters is given by Rossing [**?**].

Low order reflections often described as early reflections in relation to psychoacoustics, may occur above the steady state amplitude (echos) [2] if the steady state amplitude decreases appropriately. Early and strong reflections are of significant interest in acoustic modelling, and the auralization and perception of sound fields due to the cues humans receive from perception of them e.g. room size and source direction information.

2.2.3 Room Modes

As the domains of interest in this study are fully bounded (much like a room), sound waves propagating in the domain are subject to periodicity relative to the dimensions of the domain. That is at wavelengths relative to the dimensions of the domain, standing waves may occur within the domain i.e. there will be regions maximal and minimal pressure change at points $\frac{\lambda}{4}$ relative to a dimension of the domain, where λ is the spatial dimension of one cycle otherwise known as a wavelength. These standing waves are often called room modes, and for oblique modes in a 3 dimensional rectangular domain the frequencies of the modes can be calculated as such:

$$f_{n_x n_y n_z} = \frac{c}{2}\sqrt{\left(\frac{n_x}{l_x}\right)^2 + \left(\frac{n_y}{l_y}\right)^2 + \left(\frac{n_z}{l_z}\right)^2}$$

Where n_x, n_y, n_z is the order of the standing wave in the dimension.

These modes are defined in spatial reciprocity as axial, tangential and oblique, depending on the order of dimensions involved in the periodicity i.e. When using such an equation as that above to calculate the theoretical modes of a rectangular domain, a state table such as the one below may be used to define the $n_{x/y/z}$ order [5] component of each term:

Mode Type	x order	y order	z order
Axial	0	0	1
Axial	0	1	0
Axial	1	0	0
Tangential	0	1	1
Tangential	1	1	0
Tangential	1	0	1
Oblique	1	1	1
Oblique	1	1	2
Oblique	1	2	1

[5] number of cycles within the dimension

In a room of the dimensions $5m$ by $4m$ by $3m$, modes up to the 10th order may occur at the following frequencies:

As modes occur at higher orders, the spatial change of the intensity of a particular frequency due to modes decreases to the point where the human ear may be insensitive these changes. Further, at higher orders of modes the frequency density of modes may tend to converge, such that many modes occur around the same frequencies and so may appear to be diffusely occurring. The frequency at which modes tend to become difficult to observe by ear due to these properties for a particular domain is known as the Schroeder frequency, and that is calculated by:

$$f_{Schroeder} = 2000\sqrt{\frac{RT_{60}}{V}}$$

When considering the use of wave equation based solvers to compute large domains, it may be appropriate to consider calculating only up to a frequency of interest such as the Schroeder frequency. This will reduce the required spatial resolution for the solution and thus will reduce total memory used and the time to completion per-time-step. However, calculating low frequency propagation may require to solve for longer total time to remain accurate. In environments such as stadia, arenas and cathedrals, RT_{60} may vary from 1.2 to beyond $10s$. The mesh plot below shows Schroeder frequency as a function of RT_{60} and room volume:

Chapter 3

Finite Difference Time Domain Method

The Finite Difference Time Domain Method is a numerical method for solving partial differential equations. The power of this method lies in its simplicity and flexibility, and it can be used to solve partial differential equations of varying complexity. In this chapter we will discuss the application of the finite difference time domain method to the acoustic wave equation, including the application of empirical partially absorbing boundary conditions.

3.1 Introduction to the Finite Difference Time Domain Method

Methods for solving partial differential equations have been of significant and continued research since the early 1900's; with mathematicians such as Courant, Fiedrichs and Hrennikof undertaking seminal work in the early 1920s, that formed a base for much of the finite methods used today. The Finite Difference Time Domain Method (FDTD) is a numerical method for solving time domain problems (often wave equations) with localised handling of spatial derivatives, and was first introduced for solving Maxwell's equations to simulate electromagnetic wave propagation by Yee [3].

Yee proposed a method for which Maxwell's equations in partial differential form were applied to matrices staggered in partial steps in time and space, these matrices representing the magnetic (H) and electric (E) fields. In this explicit formulation, partial derivatives were used to solve H and E contiguously in a 'leapfrog' style, executing two sets of computations to solve for one time step. Multiple time steps would be solved from current time $t = 0$, in steps of dt to the end of simulation time T.Each field is solved at half steps in time from each-other, thus H for a current time step $t + \delta t$ is calculated using the H values one time step ago t, and the E values half a time step ago $t + \frac{\delta t}{2}$. These two fields are also solved using central finite differences in space, in a staggered grid format i.e. E at index x at time $t + \delta t$ is calculated using E at index x at time t, and the finite difference between the local discrete values of H at $x - \frac{\delta x}{2}$ and at $x + \frac{\delta x}{2}$ at time $t + \frac{\delta t}{2}$. As such, it is possible to apply a simple kernel across many discretised points of a domain (H and E) to simulate electromagnetic wave propagation.

In acoustics, FDTD can be used to simulate a wide range of problems such as diffraction and diffusion, aeroacoustics, meteorological & environmental and mixed medium, without having to perform multiple simulations for different frequencies or geometry characteristics [1].

3.2 The Finite Difference Time Domain Method Applied To The Acoustic Wave Equation

The FDTD method applied to solving the acoustic wave equation, follows an almost identical form to that of solving Maxwells Equations with FDTD [4][2]. Bottledooren's [5] seminal work applied the FDTD method to the acoustic wave equations for both Cartesian and quasi-Cartesian grid systems. As previously described in the room acoustics section, the linear acoustic wave equation is based on Newton's second law of motion, the gas law and the continuity equation, and follows the form for the changes in the pressure and velocity respectively within a volume:

$$\frac{\delta^2 p}{\delta t^2} = \frac{1}{c^2}\frac{\delta^2 p}{\delta t^2}$$
$$\frac{\delta^2 u}{\delta t^2} = \frac{1}{c^2}\frac{\delta^2 u}{\delta t^2}$$

As pressure (p) and velocity (u) have a reciprocal relationship in a similar way to H and E, it is possible to rearrange the acoustic wave equation to reflect this relationship for a FDTD computation.

3.2.1 Field Calculation

When treating the 1 dimensional linear acoustic wave equation with the FDTD method, it is possible to treat the p and u terms separately in time using the opposing terms for reciprocal calculation. As such, the p and u terms are reformulated as follows:

$$\frac{\delta^2 p}{\delta t^2} = p - \frac{\delta t}{\rho_0 \delta x}\frac{\delta^2 u}{\delta t^2}$$
$$\frac{\delta^2 u}{\delta t^2} = u - \frac{\delta t}{\rho_0 \delta x}\frac{\delta^2 u}{\delta t^2}$$

However, this formulation is incomplete as it does not consider spatial or temporal discretisation of the field of interest, when applying the FDTD method. As the FDTD method relies on solving local finite difference approximations across a domain of interest, it is important to define a space and time index referencing method. In many mathematical texts, time step indexing is often represented by an i value, and spatial indexing often uses a j,k,l or l,m,n convention. For the aim of simplicity and as we will not directly address other forms of input output system in this text, we will use t for the time step indexing, and x, y and z for spatial indexing in each dimension. Following an implementation of the acoustic FDTD method by Hill [6], we can generate the following p and u equations for FDTD applied to the acoustic wave equation:

[1] as would have to be required in frequency domain simulations such as some Finite Element and Boundary Element simulations

[2] In fact, the equations follow an almost identical form

$$u_x^{t+\frac{\delta t}{2}} = u_x^{t-\frac{\delta t}{2}} - \frac{\delta t}{\rho \delta x}\left[p^t_{x+\frac{\delta x}{2}} - p^t_{x-\frac{\delta x}{2}}\right]$$
$$p_x^{t+\frac{\delta t}{2}} = p_x^{t-\frac{\delta t}{2}} - \frac{c^2 \rho \delta t}{\delta x}\left[u^t_{x+\frac{\delta x}{2}} - u^t_{x-\frac{\delta x}{2}}\right]$$

3.2.2 Boundary Handling

As a significant part of room acoustics involves analysing the effects of reverberation, it is important to be able to handle semi-absorbing boundary conditions in an acoustic simulation. That is, to model a boundary (wall) that will absorb and reflect some proportion of energy that is at the boundary. This can be handled by calculating semi-derivatives at the boundaries of the domain based on the acoustic impedance of the boundaries [7] [6]. p, u and impedance (z) are often applied in a relationship similar to Ohms law $v = i * r$.The absorbing and reflecting properties of boundaries in acoustics are often empirically defined as normalised quantities (between 0.0 and 1.0), related to the loss in energy when a portion of the material is tested under particular conditions such as energy loss modulation when placed in a reverberation chamber. The equation to calculate acoustic impedance based on absorption coefficient is as follows:

$$z = \rho c \frac{1+\sqrt{1-a}}{1-\sqrt{1-a}}$$

Due to the spatially staggered grids in FDTD, it is possible to handle the boundaries only in the velocity components by increasing the size of the velocity matrices by 1 in the direction parallel to the axis of the velocity i.e. the length of a 3 dimensional u_x matrix would be $u_{x_{x,y,z}} = (x = N+1, y = N, z = N)$ where the size of the pressure matrix is $p_{x,y,z} = N : N : N$. For convenience and simplicity, local constant terms for the boundary can be lumped into an R parameter $R = \frac{\rho \delta x}{0.5 \delta t}$. Rearranging the form of the velocity equation to include a semi-derivative acoustic impedance component at the negative x boundary can be given as follows:

$$u_x^{t+\frac{\delta t}{2}} = \frac{R-Z}{R+Z} u_x^{t-\frac{\delta t}{2}} - \frac{2}{R+Z} p^t_{x+\frac{\delta x}{2}}$$

3.2.3 Example Function for Solving

Below, is a function written in the Matlab ®language, used to solve one time step of the wave equation using the FDTD method, in 3 dimensions:

```
function [p, ux, uy, uz] = FDTD3Dfun(p, pCx, pCy, pCz, ux, uy, uz, uCx ,...
    uCy, uCz, Rx, Ry, Rz, ZxN, ZxP, ZyN, ZyP, ZzN, ZzP)
% Function that performs one timestep of FDTD method for acoustic simulation.
%
% This function performs central finite difference calculations on
% matricies that represent pressure and velocity. This function assumes
% that a linear acoustic wave equation is being solved, and so assumes that
```

```
% the velocity terms are orthoganal and there are no cross-terms. This
% function solves empirical semi-absorbing boundary conditions, using
    the
% acoustic impedance of the boundary based on a normalised aproximation
     of
% absorption coefficient.
%
% Takes the following arguments:
% p = N:N:N matrix of pressure values
% ux = N:N+1:N matrix of velocity values
% uy = N+1:N:N matrix of velocity values
% uz = N:N:N+1 matrix of velocity values
% pCx = constant related to pressure calculation in x direction
% pCy = constant related to pressure calculation in y direction
% pCz = constant related to pressure calculation in z direction
% uCx = constant related to velocity calculation in x direction
% uCy = constant related to velocity calculation in y direction
% uCz = constant related to velocity calculation in z direction
% Rx = (rho0*dx)/(0.5*dt) Constant related to field constants
% Ry = (rho0*dy)/(0.5*dt) Constant related to field constants
% Rz = (rho0*dz)/(0.5*dt) Constant related to field constants
% ZxN = acoutsitc impedance term at boundary in -x direction
% ZxP = acoutsitc impedance term at boundary in +x direction
% ZyN = acoutsitc impedance term at boundary in -y direction
% ZyP = acoutsitc impedance term at boundary in +y direction
% ZzN = acoutsitc impedance term at boundary in -z direction
% ZzP = acoutsitc impedance term at boundary in +z direction
%
% This functions returns the pressure and velocity field matricies
%

    % Calculate central difference aproximation to velocity field
    % Velocity in a direction at current timestep excluding the
    boundarys
    % = velocity 1 time step ago - constants * pressure
    % differential half a time step ago in that direction
    ux(:, 2:end-1, :) = ux(:, 2:end-1,:) - uCx*(p(:, 2:end,:) - p(:, 1:
    end-1, :));
    uy(2:end-1, :, :) = uy(2:end-1, :, :) - uCy*(p(2:end, :, :) - p(1:
    end-1, :, :));
    uz(:, :, 2:end-1) = uz(:, :, 2:end-1) - uCz*(p(:, :, 2:end) - p(:,
    :, 1:end-1));

    % update the velocity at the negative x boundary
    % Velocity at this boundary for all of y and z = time and space
    step
    % normalised by the lovel impedance condition * current velocity
    values
    % - 2 / time and space discretization * local pressure value
    ux(:, 1, :) = ((Rx - ZxN)/(Rx + ZxN))*ux(:, 1, :)...
        - (2/(Rx + ZxN))*p(:, 1, :);

    % update the velocity at the positive x boundary
    ux(:, end, :) = ((Rx - ZxP)/(Rx + ZxP))*ux(:, end, :) ...
        + (2/(Rx + ZxP))*p(:, end, :);

    % update the velocity at the negative y boundary
```

```
    uy(1, :, :) = ((Ry - ZyN)/(Ry + ZyN))*uy(1, :, :)...
        - (2/(Ry + ZyN))*p(1, :, :);

    % update the velocity at the positive y boundary
    uy(end, :, :) = ((Ry - ZyP)/(Ry + ZyP))*uy(end, :, :) ...
        + (2/(Ry + ZyP))*p(end, :, :);

    % update the velocity at the negative z boundary
    uz(:, :, 1) = ((Rz - ZzN)/(Rz + ZzN))*uz(:, :, 1)...
        -(2/(Rz + ZzN))*p(:, :, 1);

    % update the velocity at the positive z boundary
    uz(:, :, end) = ((Rz - ZzP)/(Rz + ZzP))*uz(:, :, end)...
        +(2/(Rz + ZzP))*p(:, :, end);

    % update the pressure at all nodes
    % new pressure across domain = pressure across domain 1 time step
    ago -
    % (space,time and wave speed constant) * central difference of
    % velocities half a time step  ago in all three dimensions
    p = p - pCx*(ux(:, 2:end, :) - ux(:, 1:end-1, :))...
        - pCy*(uy(2:end, :, :) - uy(1:end-1, :, :))...
        - pCz*(uz(:, :, 2:end) - uz(:, :, 1:end-1));
end
```

3.2.4 Stability

Surrounding this formulation of the FDTD method for the acoustic wave equation, it may be important to ensure appropriate conditions are met for a converging and stable solution. As this is an explicit time marching method, the Courant-Friedrichs-Lewy (CFL) stability condition may provide a guide for generating appropriate spatial and temporal discretisation steps. The CFL condition implies that spatial δx and temporal δt discretization of a wave propagation model must be sufficiently small, that a single step in time is equal to or smaller than the time required for a wave to cross a spatial discretization step. This concerns both the speed of wave propagation c, the number of dimensions N_D and maximum simulation frequency f_{max}. The 2 dimensional CFL condition can be computed as such, where the CFL limit C_{max} is approximately 1 due to the use of an explicit time stepping solver:

$$CFL = c\frac{\delta t}{\sqrt{\Sigma_1^{N_D}\delta {N_D}^2}} \leq C_{max}$$

However, although having a CFL that is less than the C_{max} of 1 is a necessary condition to satisfy, this does not guarantee numerical stability. As this acoustic simulation is a discrete computation of a continuous system, the Nyquist sampling theorem must be considered. This suggests and $\delta t \leq \frac{f_{max}}{2}$ and as δx and δt are linked by the CFL condition, $\delta x \leq c\delta t C_{max}$. Although some stability analysis techniques are available for analysing the stability of simply shaped unbounded models such as VonNeuman analysis, such a tool is not appropriate for analysing domains with partially absorbing boundary conditions. Some sources such as Celestinos and Murphy suggest δx should be between 5 and 10 points per smallest wavelength (λ) of interest. As such, the following equations can be used to calculate δx and δt terms for stable simulation:

$$\delta x = \frac{1}{6}\frac{c}{f_{max}}$$
$$\delta t = \delta x \frac{C_{max}}{c}$$

Further study of the Bilbao FVTD thing and VonNeuman analysis is necesarry to get a better stability condition that a fifth of lambda.

3.3 Sparse FDTD

The sparse FDTD method (SFDTD) is a variant of the FDTD method proposed by Doerr [8] for use in the modelling of optical problems with significantly large domains such as for PIC micro-controllers. This is not to be confused with sparse matrix solvers used for decomposing large sparse matrices in implicit FDTD methods. The SFDTD method relies on setting an appropriate threshold, and uses this threshold to compute points in the simulation domain that should be solved, and points that should be ignored. This is analogous to applying a gate or window to the domain being computed, where ignoring parts of the domain with sufficient energy may significantly reduce computation time.

The approach suggested by Doerr is similar to the moving window FDTD method implemented by *Schuster et al* [9], in that the number of computations undertaken at any one time is significantly reduced, and thus may improve computation time in a large simulation. However unlike moving window FDTD, the SFDTD implementation suggested by Doerr dynamically accommodates high and low energy points as the simulation continues. This is achieved by maintaining a set of lists of currently active points, previously active points and an array that parallels the field and contains list indices. However Doerr's method relies on constantly maintaining lists, and a pointing array that is the same size as the domain. Maintaining long and large lists in memory may be detrimental to the largest domain a system can simulate, and may be significant when implementing this method for the acoustic wave equation.
something about electromagnetic and acoustic waves being of different types.

3.3.1 2D implementation

The implementation of the sparse FDTD method (SFDTD) for 2D simulation in this study attempts to leverage some signal processing techniques instead of search algorithms or individual checks like Doerrs method, in order to generate an indexing matrix that is used as opposed to having an indexing matrix and lists. The aim of this implementation is to create a single array of points that can be used as a mask, in less time than it would take to compute a full field for the time of propagation of wave-fronts.

Below a function is presented for calculating such a matrix:

```
function [idx] = SPARSEfun2D(p, thresholddB, p0)
% Convert threshold from dB to Pa
threshold = p0 * 10^(thresholddB/20);
% Pad edge of p with 0s to accomodate truncation
p(end+1,1:end) = 0;
p(1:end,end+1) = 0;

% Decimate matrix to operate on fewer points, and to smooth
% Decimate p in x direction
for i = 1 : size(p, 1)
temp(i,:) = decimate(p(i,:), 2);
end
% Decimate p in y direction
for i = 1 : size(temp, 2)
temp2(:,i) = decimate(temp(:,i), 2);
end
% Normalise array by threshold
temp3 = abs(temp2) ./ threshold;
% Cut out low levels
temp3 = floor(temp3);
% Bring index of interest to 1
temp3(temp3 > 1) = 1;
% Interp to complete smoothing and bring back array scale
temp4 = ceil(interp2(temp3));
% Bring back to size of p
idx = temp4(1:end-1, 1:end-1);
end
```

An implemented FDTD algorithm can then be adjusted to read through this matrix and operate at non-zero coordinates, calculating not only the regions with appropriate amounts of power but also the surrounding cells.

Depending on the intention of the persons implementing the simulation and thus the level of the threshold value, if may be possible to set the threshold low enough to allow a diffuse field to be calculated. However if an appropriate lossy wave equation was implemented, it may be possible to use a relatively high threshold to compute propagation loss for wavefronts such as strong and early reflections.

Chapter 4

Pseudo-Spectral Time Domain Method

The Fourier Pseudo-spectral Time Domain Method [PSTD] is a numerical method that can be used for solving partial differential equations. The advantage of this method lies in leveraging the computational speed of performing a discrete Fourier transform, both providing fast frequency domain differentiation and differentiation with higher order accuracy than the FDTD method. In this chapter we will discuss the application of the PSTD method to the acoustic wave equation, including the use of empirical partially absorbing boundary conditions and the perfectly matched layer (PML).

4.1 A Background to the Pseudo-Spectral Time Domain Method

The PSTD method is of a branch of spectral methods that are useful for solving some hyperbolic partial differential equations, and was first proposed by Orszag [10], and was further expanded by Kriess and Oliger [11]. Fourier Pseudospectral methods have been advanced considerably since then, and have found applications in weather prediction particle physics, electromagnetics and acoustics. More recently Trefethen [12] presented a classic text showcasing both the power of spectral methods and how simply they could be implemented. The Fourier PSTD method used in this study is advanced from that presented by Angus and Caunce [13], with expansion into 2 and 3 dimensions and implementation of partially absorbing boundary conditions.

4.2 The Pseudospectral Time Domain Method Applied To The Wave Equation

The acoustic wave equation has been previously defined with two resolving parts:

$$\frac{\delta^2 p}{\delta t^2} = \frac{1}{c^2}\frac{\delta^2 p}{\delta t^2}$$
$$\frac{\delta^2 u}{\delta t^2} = \frac{1}{c^2}\frac{\delta^2 u}{\delta t^2}$$

Applying a continuous time Euler solving method to the above relationship with respect to space brings the following:

$$\rho_0 \frac{\delta}{\delta x}\left[\frac{\delta u}{\delta t}\right] = \frac{1}{c^2}\frac{\delta^2 p}{\delta t^2}$$

Implementing a discrete time and space version of this equation using an FDTD scheme yields:

$$u_x^{t+\frac{\delta t}{2}} = u_x^{t-\frac{\delta t}{2}} - \frac{\delta t}{\rho\delta x}\left[p^t_{x+\frac{\delta x}{2}} - p^t_{x-\frac{\delta x}{2}}\right]$$
$$p_x^{t+\frac{\delta t}{2}} = p_x^{t-\frac{\delta t}{2}} - \frac{c^2\rho\delta t}{\delta x}\left[u^t_{x+\frac{\delta x}{2}} - u^t_{x-\frac{\delta x}{2}}\right]$$

The PSTD method applies differentiation in the frequency or $k-space$ domain. This can be represented as:

$$u_x^{t+\frac{\delta t}{2}} = u_x^{t-\frac{\delta t}{2}} - \frac{\delta t}{\rho\delta x}\boldsymbol{F}^{-1}\left(\varepsilon\boldsymbol{F}\left[p^t\right]\right)$$
$$p_x^{t+\frac{\delta t}{2}} = p_x^{t-\frac{\delta t}{2}} - \frac{c^2\rho\delta t}{\delta x}\boldsymbol{F}^{-1}\left(\varepsilon\boldsymbol{F}\left[u^t\right]\right)$$

Where $\boldsymbol{F}$ represents the forward and inverse Fourier Transforms respectively, and ε is a differentiating function representing:

$$\mathbf{JK}_N \exp^{-jk_N\frac{\delta x}{2}}$$

Which is the impulse response of a differentiating function in the complex domain, where N is the 1D size of the domain in the dimension of interest i.e. each dimension requires a differentiator function. This is compounded by velocity components in each dimension not having cross terms.

4.2.1 Absorbing Boundary Conditions

The Fourier PSTD is fast and performs well for problems with smoothly varying properties. However, this method suffers from Gibbs phenomenon as the domain is periodic and has discontinuity at its boundaries. This is manifested as aliasing in the domain. A way to reduce this aliasing is to increase the area of the domain and implement a perfectly matched layer (PML). A PML is a totally absorbing boundary condition that absorbs waves travelling into it without reflection, as opposed to a more simple boundary condition such as Dirchlet (fixed) that will cause reflections. The PML was first developed for Maxwell's Equations in Computational Electromagnetics by Berenger [14], and was quickly developed for other applications such as acoustic FDTD and FE [15].

Three kinds of PML available are the split field PML, Uniaxial PML and the Convolutional PML. For the sake of time saving and simplicity, the uniaxial perfectly matched layer is implemented in this study. The PML is implemented as a matrix with the same dimensions as the domain, which has been extended in each dimension by the number of cells matching the desired depth of the PML N_{pml}. In the PML region, the value of the PML contribution to the p and u update equations σ, reduces in value from 1 to 0 towards the final boudnary of the domain, continuously and smoothly impeding acoustic waves in any direction within the PML, thus causing no reflection of waves from the PML back into the domain proper.

The modified 1D update equation for this is as follows:

$$
\begin{aligned}
u_x^{t+\frac{\delta t}{2}} &= u_x^{t-\frac{\delta t}{2}} \sigma_a - \tfrac{\delta t}{\rho \delta x} \sigma_b \boldsymbol{F}^{-1} \left(\varepsilon \boldsymbol{F} \left[p^t \right] \right) \\
p_x^{t+\frac{\delta t}{2}} &= p_x^{t-\frac{\delta t}{2}} \sigma_a - \tfrac{c^2 \rho \delta t}{\delta x} \sigma_b \boldsymbol{F}^{-1} \left(\varepsilon \boldsymbol{F} \left[u^t \right] \right)
\end{aligned}
$$

Where:

$$
\begin{aligned}
\sigma_a &= \frac{1-a}{1+a} \\
\sigma_b &= \frac{1}{1+a} \\
d &= PMLDepth \\
N &= TotalArrayLength \\
i &= 1, 2 \ldots N-1 \\
i < d \quad a &= \frac{1}{3} \left(\frac{i}{d} \right)^3 \\
d < i < N-d \quad a &= 0 \\
i > N-d \quad a &= \frac{1}{2} \left(\frac{N-i}{d} \right)^3
\end{aligned}
\tag{4.1}
$$

As the maximum number in the matrix is 1, a multidimensional implementation of the PML regions involved creating orthogonal arrays of these 1D sections and applying an average summation of the regions values i.e. sum of squares in 2D and a sum of 3D matrices divided by the number of matrices.

4.2.2 Partially Absorbing Boundary Conditions

Partially absorbing boundary conditions for PSTD are implemented using the methods explored by *Spa et al.* [16], where a real, normalised value can be defined and used to define a frequency independent absorption characteristic for acoustic PSTD simulations. This method applies a weighting to the relationship between pressure and velocity at a point in the grid, reflecting and passing a proportion of energy.

At the point where the partially absorbing boundary occurs, the scaling term ξ is set to either scale the p or u value depending on the value if ξ at that point. The value of ξ is determined by normalising the relationship between specified absorption value α, and the numerical stability of the simulation$S$:

$$
\begin{aligned}
S &= \frac{\delta t}{\delta x} \\
\xi_n &= 1 - \alpha \\
\xi &= \frac{(1+\xi_n)}{(1+\xi_n - 2 * S * \xi_n)}
\end{aligned}
\tag{4.2}
$$

The update equations are then modified to handle ξ at the point of interest at the boundary of the domain:

$$
\begin{aligned}
&\text{For} \xi \leq 1 : \\
&\quad p_x^{t+\frac{\delta t}{2}} = \xi \left[p_x^{t-\frac{\delta t}{2}} \sigma_a - \frac{c^2 \rho \delta t}{\delta x} \sigma_b \boldsymbol{F}^{-1} \left(\varepsilon \boldsymbol{F} \left[u^t \right] \right) \right] \\
&\text{For} \xi \geq 1 : \\
&\quad u_x^{t+\frac{\delta t}{2}} = \frac{1}{\xi} \left[u_x^{t-\frac{\delta t}{2}} \sigma_a - \frac{\delta t}{\rho \delta x} \sigma_b \boldsymbol{F}^{-1} \left(\varepsilon \boldsymbol{F} \left[p^t \right] \right) \right]
\end{aligned}
\tag{4.3}
$$

Chapter 5

Validation

While it may be beneficial and interesting to review the behaviour of a wave propagating in a fictitious or simulated domain, model validation could be considered an important step towards creating a such a robust and useful tool. Below we shall discuss a scenario that is used for the validation of the simulation tools described in this study, and we shall review the performance of such tools in comparison to the hand calculated properties of the scenario and the results of an Image Source model.

5.1 A Model Environment

The model environment used for validation in this study shall be a fully enclosed room of the following dimensions:

Dimension	Length (m)
x	5
y	4
z	3

This gives a volume of $v = 60m^3$, and a boundary surface area of $S = 94m^2$.
The boundaries shall have a uniform absorption coefficient of $\alpha = 0.45$. As the boundaries are uniformly absorbing and the coefficient average is above 0.1, it may be appropriate to use the Eyring reverberation time equation which yields $RT_{60} = 0.1719s$.
The average number of reflections before the energy of a wave-front has decayed below the noise floor will be $N_{reflections} = 30.7$, and the mean free path between reflections will be $MFP = 2.55m$.
The Schroeder frequency of the room will be $f_{schroeder} = 107Hz$, and the axial, tangential and oblique modes below the Schroeder frequency are calculated as follows:

5.2 Results

Chapter 6

Execution Time and Analysis

The interest of this study is to analyse the execution times of the three algorithms described above, to determine which method executes in the fastest time and thus might be most appropriate for using to solve large problems. The execution times will be measured for each method for domains of the following sizes at 10kHz sample rate:

Dimension (m^2)	FDTD Cells	SFDTDCells	PSTD Cells
2	121104	121104	24025
4	483025	483025	62500
8	1932100	1932100	192721
16	7722841	7722841	667489
32	30880249	30880249	2474329
64	123498769	123498769	9517225
128	493950625	493950625	37319881
256	1975802500	1975802500	147816964
512	7903210000	7903210000	588353536

As the implementation of SFDTD has not been successfully implemented in 3D, all simulations will be performed in 2D. Matlabs Tic/Toc methods are used to calculate execution times within the loop, for 100*ms* of calculation time. The time steps for each method are:

6.1 Results

The average execution time for each domain is given below:

As domains get bigger, sfdtd and pstd dominate.

Conclusion

We introduced FDTD, PSTD and SFDTD.

We couldn't get it all going in 3D, but I tested SFDTD in 2D in as big a domain

One of them is faster, but fdtd isnt bad at all!

Further work would successfully get sfdtd going in 3d, and would really optimise the whole thing.

Bibliography

[1] L L Beranek. *Acoustics*. McGraw-Hill electrical and electronic engineering series. McGraw-Hill, 1st edition, 1954.

[2] F. Alton Everest and Neil A. Shaw. *Master Handbook of Acoustics, 5th Edition*, volume 110. 5th edition, 2009.

[3] Kane S Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, 14(3):302–307, 1966.

[4] John B Schneider. Understanding the Finite-Difference Time-Domain Method. pages 1–403, 2015.

[5] D Botteldooren. Acoustical finite-difference time-domain simulation in a quasi-Cartesian grid. *Acoustical Society of America*, 95(5):2313–2319, 1993.

[6] Adam J Hill. Analysis , Modeling and Wide-Area Spatiotemporal Control of Low-Frequency Sound Reproduction. (January), 2012.

[7] Soren Krarup Olesen. Low Frequency Room Simulation using Finite Difference Equations. *Proceedings of the 102nd Audio Engineering Society Convention*, 1997.

[8] Christopher Doerr. Sparse Finite Difference Time Domain Method. *IEEE Photonics Technology Letters*, 25(23):1, 2013.

[9] J.W. Schuster, K.C. Wu, R.R. Ohs, and R.J. Luebbers. Application of moving window FDTD to predicting path loss over forest covered irregular terrain. *IEEE Antennas and Propagation Society Symposium, 2004.*, pages 1607–1610 Vol.2, 2004.

[10] Steven a. Orszag. Numerical simulation of incompressible flows within simple boundaries: accuracy. *Journal of Fluid Mechanics*, 49(01):75, 1971.

[11] Heinz-Otto Kreiss and Joseph Oliger. Comparison of accurate methods for the integration of hyperbolic equations. *Tellus*, 24(3):199–215, 1972.

[12] Lloyd N Trefethen. Spectral Methods in Matlab. *Lloydia Cincinnati*, 10:184, 2000.

[13] Jamie A S Angus and Andrew Caunce. A GPGPU Approach to Improved Acoustic Finite Difference Time Domain Calculations. *128th Audio Engineering Society Convention*, 2010.

[14] Jean P Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114:185–200, 1994.

[15] Qing-Huo Liu and Jianping Tao. The perfectly matched layer for acoustic waves. *J. Acoust. Soc. Am.*, 102(4):2072–2082, 1997.

[16] Carlos Spa, Jose Escolano, and Adan Garriga. Semi-empirical boundary conditions for the linearized acoustic Euler equations using Pseudo-Spectral Time-Domain methods. *Applied Acoustics*, 72(4):226–230, 2011.