UNIVERSITY OF THE WITWATERSRAND

The application of non-linear partial differential equations for the removal of noise in audio signal processing.

by

Jarrod Jay Shipton 407406

Supervisor: Dr. B. Jacobs

A thesis submitted in fulfillment for the
degree of Masters of Science

in the
Faculty of Science
School of Computer Science and Applied Mathematics

October 2017

Declaration of Authorship

I, Jarrod Jay Shipton, declare that this thesis titled, 'The application of non-linear partial differential equations for the removal of noise in audio signal processing.' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"Mathematics is music for the mind; music is mathematics for the soul."

Anonymous

UNIVERSITY OF THE WITWATERSRAND

Abstract

Faculty of Science
School of Computer Science and Applied Mathematics

Masters of Science

by Jarrod Jay Shipton 407406

Supervisor: Dr. B. Jacobs

This work explores a new method of applying partial differential equations to audio signal processing, particularly that of noise removal. Two methods are explored and compared to the method of noise removal used in the free software Audacity(R). The first of these methods uses a non-linear variation of the diffusion equation in two dimensions, coupled with a non-linear sink/source term, in order to filter the imaginary and real components of an array of overlapping windows of the signal's Fourier transform. The second model is that of a non-linear diffusion function applied to the magnitude of the Fourier transform in order to estimate the noise power spectrum to be used in a spectral subtraction noise removal technique. The technique in this work features finite difference methods to approximate the solutions of each of the models.

Acknowledgements

I would like to thank my supervisor, Dr Byron Jacobs, for standing by me and advising me through it all, my collegues and university friends for always having the time to discuss problems, my family for always supporting me.

J. J. S. and B. A. J. acknowledges the National Research Foundation of South Africa under grant number 94005.

Opinions expressed and conclusions arrived at are those of the author and are not necessarily to be attributed to the CoE-MaSS.

Contents

List of Figures

List of Tables

For my late brother Russell who was always a positive influence in my life and always made me strive for my best even when I wasn't confident in my abilities.

Chapter 1

Introduction

1.1 Introduction

In audio signal processing there is an unwanted phenomena known as noise. It is detrimental to the intelligibility of speech and can often mask important information within a signal. Noise removal is used in various situations in industry, such as in hearing aids, in music recording studios and many other commercial industries, in many cases it is required to remove echo or a humming sound in the background. However in this work we will mainly be dealing with additive noise, which, mathematically, can be displayed via the following equation:

$$X(t) = S(t) + N(t), \tag{1.1}$$

where $X(t)$ is the observed signal, comprised of the true signal, $S(t)$, and the noise process, $N(t)$.

There are a number of noise categories in which $N(t)$ can fall into. In this work we will be using additive noise and explain a few forms in which additive noise can be found. White or Gaussian noise is completely random and features equal amplitude noise at each frequency band, that is its spectral density has the same power in any given bandwidth. This is the most common type of noise referred to in literature. It is completely uncorrelated and thus unpredictable. Pink noise is similar except it has a spectral density inversely proportional to its frequency, that is a density proportional to $1/f$, where f is the frequency. Brown or Brownian noise is produced by Brownian motion. Its spectral density is inversely proportional to the square of its frequency, that is a density proportional to $1/f^2$. Blue noise is similar to Pink noise except its spectral density is proportional to its frequency, that is a density proportional to f. Violet noise is similar to Brown noise in the same way Blue is to Pink. Its spectral

density is proportional to the square of its frequency that is a density proportional to f^2. Grey noise is White noise with some sort of weighting applied to it, eg. an inverse A-weighting curve. The A-weighting curve was designed to reflect the response of the human ear to each frequency, that is, the human ear is less sensitive to frequencies not in the range of 500 Hz and 6 kHz and this curve reflects such a response by the human ear. The A-weighting curve covers the range of 20 Hz to 20 kHz as noted by Moller and Pedersen as well as various other authors in [1, 2, 3].

However, in literature White noise is typically the noise type of choice when simulating noise since it is similar to a large number of background noises such as wind, thermal noise in electronic components and machine hum. Thus this work will focus on the removal of such noise. Various methods for the removal of this specific additive noise exist and are used in music and speech enhancement. A number of these methods will be reviewed.

A number of methods for noise removal in audio signal processing exist which use statistical methods to estimate the noise profile and then use spectral subtraction, the process of subtracting from the short-time power spectrum (the square of the magnitude of the short-time Fourier transform) of the contaminated audio, P_y, an estimate of the noise short-time power spectrum, P_n, resulting in a "clean" audio short-time power spectrum, P_x. The basic process is describe by the following equation:

$$P_x = P_y - \alpha P_n, \tag{1.2}$$

where α is a scaling factor for the noise. These are then transformed back to the time domain using the noisy phase since the phase is deemed perceptually unimportant.

Ephraim and Malah developed a method based on a minimum mean-square error (MMSE) short-time spectral amplitude estimator in [4]. This work was later expanded in [5] resulting in minimum mean-square error log-spectral amplitude estimator as described in the paper.

Wolfe and Godsill, as seen in [6], propose alternatives to Ephraim and Malah by using a Bayesian estimation model while extensive expansions on this approach for removing noise from audio data is covered by Godsill and Rayner [7] where they ultimately lead to a fully Bayesian restoration method using expectation-maximisation (EM) algorithms and Markov Chain Monte Carlo (MCMC) methods. Godsill and Rayner further discuss the EM methods in [8, 9] while MCMC methods and their various applications to audio signal processing are discussed by Carlin, Polson and Stoffer and McCulloch and Tsay in [10, 11, 12], where they explain how to make predictions for the mean and variance shifts in autoregressive time series data such as noise. Ruanaidh and Fitzgerald discuss

using interpolation of missing samples in [13], which can also be useful in the estimation of a noise profile, as well as solving the problem of missing samples in audio.

There are a number of various other approaches which all use a form of the MMSE spectral amplitude estimator technique. Specifically Martin and Lotter and Vary use supergaussian priors to estimate the noise profile in [14, 15]. Andrianakis and White use spectral amplitude estimators with chi and gamma priors in [16]. Chen and Loizou use a laplacian based spectral amplitude estimator in [17]. Erkelens, Hendriks and Jensen get MMSE estimation of the discrete fourier coefficients using gamma priors in [18].

These statistical methods all have strengths in the fact that they are unsupervised and have relatively low computational complexity compared to other noise removal algorithms. Generally the only required knowledge is that of the speech and noise variance, which can be estimated from the noisy recordings. The weaknesses in these methods are in the nature of the noise profile being non-static in time while the estimation parameters are. This implies that over time the noise profile changes and the estimation parameters do not which impairs the audio and speech enhancement.

Methods using complex wavelets for noise reduction which work in the time-scale representations of audio have shown to be excellent noise reduction techniques. These methods are described by Wolfe and Godsill in [19] and were later expanded on by Yu, Bacry and Mallat in [20] where they create an adaptive block attenuation technique based on the dyadic classification and regression trees (CART) algorithm. The dyadic CART algorithm is discussed by Breiman et al. in [21].

Another approach for speech enhancement, or audio separation (when there are multiple signal sources and we require a specific signal and regard the others as the "noise", thus separate the signal from the "noise"), is using non-negative matrix factorization (NMF). This approach tackles the problem in the Bayesian approaches of not being able to deal with non-stationary noise profiles. The idea of the algorithm is discussed by Lee and Sebastian as well as Févotte, Bertin and Durrieu in [22, 23]. More specific applications of this approach are discussed by various authors, notably Schmidt and Larsen in [24, 25] where they with wind noise reduction using non-negative sparse coding. Duan et al. achieve enhancement of speech from non-stationary noise environments by using non-negative spectrogram decomposition in [26]. Mohammadiha et al. use prediction based filtering based on temporal dependencies in NMF as described in [27]. Scmidt and Olsson and Virtanen achieve audio separation in [28, 29] via NMF. While Mysore and Smaragdis use a NMF semi-supervised separation process in [30]. As noted above this approach better deals with non-stationary noise profiles and it does enhancements across frequency bins jointly unlike the Bayesian approaches. It however has a worse computational complexity in comparison and has poor statistical estimation criteria for the profiles.

The next, and more recent approach for speech enhancement is that of ideal masking (IM). This approach uses the time-frequency representation and is particularly effective audio separation. Han and Wang describe a support vector machine (SVM) based classification approach to speech separation us IM in [31]. Y. Wang and D. Wang improve this by employing pre-trained deep neural networks in [32]. Chen, Y. Wang, and D. Wang expand further by doing speech separation at low signal to noise ratios in [33] while Y. Wang, Narayanan and D. Wang, describe supervised speech separation in [34]. IM follows a feature extraction process, generally this is determining whether a specific segment of the data is noise dominant or speech dominant, then a mask estimation process and it uses this for the re-synthesis of the signal. For the feature extraction process this method requires a Deep Neural Network which is a form of machine learning and thus requires a training process (which takes a large amount of time and resources) and database which is a clear weakness. It does however have huge improvements in intelligibility over the NMF and Bayesian approaches.

The next approach is to observe what is being done in fields outside of strictly audio signal processing and expand to that of image processing. Image processing and audio signal processing are both sub-fields of the same field of work, signal processing, with image processing being done mostly in two dimensions and audio signal processing done mostly in one dimension. However we can justify using two dimensional processing on audio since the spectrogram of an audio signal can be seen as an image. A distinct difference between images and that of a spectrogram can be seen however. In images we desire to preserve what is known as the edges and in the spectrogram we want to keep the fundamental frequencies of the signal.

In image processing Gaussian noise is often removed using spatial filters such as the Median or Gaussian filters as described by Jain et al. in [35]. Whilst this does prove useful, it does come with the cost of blurring the images at edges, which are points in the image with high gradients. A novel approach that has been used for noise removal in image processing is that of using the diffusion equation to remove noise as suggested by Perona and Malik in [36]. In this paper, it is suggested that using a correctly chosen diffusion coefficient (which in this case is a non-negative monotonically decreasing function of the gradient of the image's intensity) makes it possible to remove noise from smooth areas in the image whilst still maintaining the edges. Weaknesses occur when the magnitude of the noise that is present in the image is comparable to the magnitude of the features of the image which creates the illusion of an edge being present, as described by Perona and Malik and Yu et al. in [36, 37]. An extension of this thought process has already been demonstrated by Dugnol et al. where they use a partial differential equation model which is based on processing the spectrogram of the signal using a smoothing and edge detecting nonlinear diffusion equation in [38].

The partial differential equation is originally attributed to Alvarez, Lions and Morel and described in [39]. Jacobs and Momoniat use a partial differential equation for text binarization in [40] which uses diffusion with a sink/source term in order to push any pixel in an image to either its maximum or minimum value based on a threshold value that is between the two and considered to be above the noise threshold, thus removing noise from a text image. This is a useful basis for an approach to remove any frequencies below a certain threshold in a spectrogram however it would not maintain power any of the frequencies above that threshold, it would also cause "blurring" of the frequencies, that is, it would add to unwanted frequencies and remove power from frequencies that we want to preserve.

1.1.1 Outline of work

In this section I will give an outline of each chapter in this thesis. In Chapter 2, I will highlight a number of important methods and concepts that will be used throughout the work. A brief outline of each method and a number of reasons why other methods were not used in their place.

In Chapter 3 I will examine a non-linear diffusion equation applied to the magnitude of the Fourier transform of a signal at different time windows in order to approximate the noise profile so as to use spectral subtraction to remove the noise. The magnitude of the Fourier transform is again a reasonable representation of the signal since it is a representation of the magnitude of the frequencies present in the signal and if we take its squared value we obtain the power spectrum, which is what is used in the process of spectral subtraction.

In Chapter 4 I will examine a non-linear partial differential equation and explain how it is applied to the spectrogram of the audio signal in order remove noise. The spectrogram is a reasonable representation of the audio signal since it is a representation of the power of the different frequencies present in the signal at different time frames. This gives us the ability to reduce the power of the noise at each frequency.

In Chapter 5 I compare the results of the models discussed in Chapters 4 and 3 with that of the results of the noise removal process used in Audacity(R), for this work I used version 2.1.3 of Audacity(R) recording and editing software [41], a free audio signal processing software, and reformatted any results using mathematica. This will be to show that the results are comparable to that of a method being used in industry.

In Chapter 6 I summarise and highlight the work done though out this thesis.

Chapter 2

Methodology

2.1 Introduction

In this chapter I will introduce some of the concepts, methodologies and notation that are used throughout the work. It should serve as a starting point or reference for anyone who is not familiar with the concepts, terms or notation used.

2.2 Diffusion Equation

The standard diffusion equation is of the following form

$$\frac{\partial u(t,X)}{\partial t} = \nabla \cdot [D(u,t,X)\ \nabla u(t,X)], \tag{2.1}$$

where $u(t,X)$ is the density, D is the diffusivity coefficient, with the diffusion being isotropic, X is the position variable and t is the time variable.

For the purpose of this thesis we will be using subscript notation. This is an alternative way to display the different differentials in a given equation. For example:

$$\frac{\partial u(t,X)}{\partial t} = u_t, \tag{2.2}$$

would be one way to display the first derivative with respect to time,

$$\frac{\partial^2 u(t,X)}{\partial x^2} = u_{xx}, \tag{2.3}$$

would be the second derivative with respect to the variable x. From this point on in this work we will default to using u as the variable for our partial differential equations. To

demonstrate an example of this notation I will use the diffusion equation in one spatial variable with a constant diffusivity coefficient. It would be of the following form:

$$u_t = k\, u_{xx}, \tag{2.4}$$

where u_t is the first order time derivative, u_{xx} is second order spatial derivative and k is the constant diffusivity coefficient.

The reason the diffusion equation is the equation of interest in this work is the nature in which it "smooths" out a given initial condition. If we were to use equation (2.4), with $k = 1$, an initial condition of $u = \sin(x)$, where $x \in [0, 2\pi]$, and simple boundary conditions of $u(t, 0) = u(t, 2\pi) = 0$ (this concept is expanded upon in sections 2.3), we would find that as the time evolution of the solution went on, as t grows larger, that the solution would slowly evolve to be a straight line with $u = 0$ when it eventually reaches it's steady state. Another important property to note is that it deals with discontinuities in the initial condition as soon as $t > t_0$. In other words, if we instead had our initial condition as

$$u = \begin{cases} 1, & \text{if } 0 \leq x \leq 0.5 \\ -1, & \text{if } 0.5 < x \leq 1 \end{cases}, \tag{2.5}$$

where $x \in [0, 1]$ and boundary conditions of $u(t, 0) = u(t, 1) = 0$, we would find that u would very quickly become smooth and approach the steady state, which is again $u = 0$.

The properties described above are incredibly useful when we are thinking about using them for the removal of noise, which essentially manifests itself as random fluctuations in the initial condition, which can create steps such as those described by equation (2.5), which we wish to "smooth" out.

In this section I have presented what we would expect of a simple linear diffusion process and why these solutions have desirable qualities in noise removal, however the steady state of such an equation is not entirely the desired result when we are removing noise from a signal, since it would remove the signal in its entirety. Thus, we will want to find a way for the steady state to be reach, such that it still contains the signal. Essentially we only wish to remove what is regarded as the random fluctuations from the noise. I will discuss, in Chapters 3 and 4, non-linear variations of the diffusion equation and how they will be used, in conjunction with the various transformations, which are introduced in sections 2.5 and 2.6, to remove this noise.

2.3 Boundary Conditions

The boundary conditions for any chosen partial differential equation have to make sense to its real world application. As it is noted by various authors in [1, 2, 3], the human hearing range is $20hz$ to $20khz$. This leads to the belief that any frequency below $20hz$ and any frequency above $20khz$ is not important to human hearing and can be disregarded in our analysis of audio signals. Thus if our non-linear diffusion model was to be used on any data containing the frequencies (such as the fourier transform of the data) we could make our boundary conditions for u such that there are zeros at either end, under the assumption that the lowest frequency is below $20hz$ and the highest is above $20khz$, in cases where this is not true we will keep this as a simplifying assumption since it would only affect the last measurable frequency. If we were to use the actual signal as our data, we could assume that the audio signal has no sound at the start, nor at the end of the clip. This is a reasonable assumption since one would not usually record an audio signal when the sound has already started and would usually record the sound up to its completion. While arguments can be made that this wouldn't always be the case, it is mostly a simplifying assumption. This would again leave us with the boundary conditions of u being zero at either end of the data. We would also have to note that in the Fourier transform of the signal the size of our domain would be prescribed to us by the sampling rate of the signal.

This leaves us with the notion that Dirichlet boundary conditions are the best to be used to describe the real world scenario in this work, under the assumptions made above. They will be of the form,

$$\begin{aligned} u(t,-1,y) &= 0, \\ u(t,1,y) &= 0, \\ u(t,x,-1) &= 0, \\ u(t,x,1) &= 0. \end{aligned} \tag{2.6}$$

2.4 Solution approximation schemes

It is important to note that throughout the work the partial differential equations to be used are non-linear, and thus, in general are not representable as elementary equations. As such, there are a number of different approximations for finding solutions of partial differential equations that can be used. Each of them have reasons for why they should be used and when they should be used. I will describe a few of these and then explain why they would, or would not be suitable, and why I chose the scheme used in this work.

2.4.1 Finite Element Method

Finite element methods are used to approximate the solutions of partial differential equations.

To examine how these methods work let us define a parabolic partial differential equation, such as the diffusion equation, as follows,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(x,t), \tag{2.7}$$

where $x \in (0,1)$ and $t \in (0,T]$. The boundary conditions are $u(0,t) = 0$ and $u(1,t) = 0$ and the initial condition is $u(x,0) = U_0$ In order to solve this with the finite element method we would need to begin by defining a mesh, $Q = [0,1] \times [0,T]$. We would define the mesh-size in the x direction as h=1/N and the mesh-size in the t direction as $\Delta t = T/M$ where $N \geq 2$ and $M \geq 1$. Thus we will have have the uniform mesh

$$Q_h^{\Delta t} = \{(x_j, t^i) : x_j = jh, 0 \leq j \leq N; t^i = i\ \Delta t, 0 \leq i \leq M\}. \tag{2.8}$$

Next we have to define the set $\phi_h \in V_h$ to be a set of continuous piecewise linear functions on the x mesh, that is $0 = x_0 < ... < x_N = 1$, which all vanish at the end points.

From here I will present how the forward Euler scheme works for solving this problem. From here it is important to note that the inner product $\langle u\,, v\rangle$ in this problem is defined by

$$\langle u\,, v\rangle = \int_0^1 v(x)w(x)dx. \tag{2.9}$$

We will have to find $u_h^i \in V_h$ such that

$$\begin{aligned} &\left\langle \frac{u_h^{m+1} - u_h^m}{\Delta t}, v_h \right\rangle + a(u_h^m, v_h) = \langle (f(x,t^i), v_h\rangle \quad \forall v_h \in V_h \\ &(u_h^0 - U_0, v_h) = 0 \quad \forall v_h \in V_h, \end{aligned} \tag{2.10}$$

where u_h^i is the approximation of $u(x,t^i)$ and

$$a(w,v) = \int_0^1 w'(x)v'(x)dx \tag{2.11}$$

Thus given u_h^i we can find u_h^{i+1} by solving the system of linear equations with the positive definite matrix M of size $(N-1) \times (N-1)$, with the entries $\langle \phi_k, \phi_q\rangle$ where ϕ_k is the piecewise linear finite element function associated with the point x_k.

The advantages of this method is that the mesh can be designed to work with a problem with a boundary of any shape quite easily, however, it is not very amendable to the use of discontinuous data such as that of the audio sound clips and their discrete Fourier transform which we will be using in this work.

2.4.2 Collocation Methods

Collocation methods are another type of solution approximation scheme that can be used. To demonstrate the idea of these I will explain using Chebyshev collocation. The basic idea is to of this method is divide the domain we are working on into a a non-uniform grid over the basis of the Chebyshev polynomials, which is on [-1,1]. This might require a mapping of our domain to be in the domain [-1,1]. When we create our mesh we use Chebyshev-Gauss-Labatto points, that is

$$x_j = \cos\left(\frac{j\pi}{N}\right), \quad j = 0, 1, ..., N, \tag{2.12}$$

The differentiation matrix in this method is defined as

$$d_{j,k} = \begin{cases} \dfrac{c_i(-1)^{j+k}}{c_k(x_j - x_k)}, & j \neq k \\ -\dfrac{x_j}{2(1 - {x_j}^2)}, & j = k \\ \dfrac{1}{6}(2N_x^2 + 1), & j = k = 0 \\ -\dfrac{1}{6}(2N_x^2 + 1), & j = k = N, \end{cases} \tag{2.13}$$

where

$$c_j = \begin{cases} 2, & j = 0, N \\ 1, & j = 1, ..., N - 1. \end{cases} \tag{2.14}$$

A useful feature of this differentiation matrix is that if we wish to get the second order differentiation matrix it is as simple $D^{(2)} = D^{(1)}.D^{(1)}$ where $D^{(1)} = d_{j,k}$. We can then use these to solve the diffusion equation as done by Golbabai in [42].

However we won't use collocation methods for our work since we have fixed grid spacing within our data and the for collocation methods to be the most effective they can be, they require the points for their grid to correspond to quadrature formulas, and we cannot change the grid spacing of this work since the Fourier transform is given in a specific form.

2.4.3 Finite-Difference Method

Finite-difference methods are used to approximate the values of the derivatives of functions. There are a number of different finite-difference schemes which can be implemented. Among these are various explicit, eg. FTCS (Foward in time and central in space), and implicit, eg. Crank-Nicolson, schemes. While there are benefits to using implicit schemes, such as their unconditional stability, they can be cumbersome to set up. The FTCS method, while having the issue of stability conditions imposed by the scheme, especially in higher dimensions, and for non-linear equations, being explicit, is easy to set up for various different equations. Since this research is mainly focused on the design of partial differential equations to be applied in audio signal processing and many different iterations of the variations of the equations presented in this paper were applied to the processes before arriving at the final resultant equations I have chosen to use the method which is easier to set up. With this being said I acknowledge there may be some benefit to using an implicit method to take advantage of the unconditional stability criteria, however, the FTCS method demonstrates the ability of the partial differential equations, applied in the methods described in this work, to remove noise reasonably adequately.

Thus I will only cover the approximations that will be used in the explicit finite-difference scheme known as FTCS. This scheme allows us to work with discrete data that has discontinuities, such as that used in digitally recorded audio. However it does imply conditions onto certain parameters in order to maintain stability. These conditions will be calculated when the finite-difference scheme is applied to a specific equation.

The first of these approximations is the forward finite-difference approximation for the first derivative. For this we will demonstrate using the first derivative in time for a function $u(t)$. The derivative approximation is

$$\frac{du}{dt} = \frac{u(t+\Delta t) - u(t)}{\Delta t} + \mathcal{O}(\Delta t) = \frac{u_{i+1} - u_i}{\Delta t} + \mathcal{O}(\Delta t), \tag{2.15}$$

where a Δt change time is equivalent to a unit change in i.

Next I will describe the central finite-difference approximation using spatial derivatives. I will cover both the first order and second order derivatives. For this we will be using the multivariate function $u(x, y, t)$. The first derivatives in the x and y directions are

$$\frac{\partial u}{\partial x} = \frac{u(t, x-\Delta x, y) - u(t, x+\Delta x, y)}{2\Delta x} + \mathcal{O}(\Delta x^2) = \frac{u^i_{j-1,k} - u^i_{j+1,k}}{2\Delta x} + \mathcal{O}(\Delta x^2), \tag{2.16}$$

and

$$\frac{\partial u}{\partial y} = \frac{u(t,x,y-\Delta y) - u(t,x,y+\Delta y)}{2\Delta y} + \mathcal{O}(\Delta y^2) = \frac{u^i_{j,k-1} - u^i_{j,k+1}}{2\Delta y} + \mathcal{O}(\Delta y^2), \quad (2.17)$$

respectively where a Δx change in x is equivalent to a unit change in j and a Δy change in y is equivalent to a unit change in k. The second order approximations will use the same notation and are

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} &= \frac{u(t,x-\Delta x,y) - 2u(t,x,y) + u(t,x+\Delta x,y)}{\Delta x^2} + \mathcal{O}(\Delta x^2) \\ &= \frac{u^i_{j-1,k} - 2u^i_{j,k} + u^i_{j+1,k}}{\Delta x^2} + \mathcal{O}(\Delta x^2) \end{aligned} \quad (2.18)$$

and

$$\begin{aligned} \frac{\partial^2 u}{\partial y^2} &= \frac{u(t,x,y-\Delta y) - 2u(t,x,y) + u(t,x,y+\Delta y)}{\Delta y^2} + \mathcal{O}(\Delta y^2) \\ &= \frac{u^i_{j-1,k} - 2u^i_{j,k} + u^i_{j+1,k}}{\Delta y^2} + \mathcal{O}(\Delta y^2) \end{aligned} \quad (2.19)$$

respectively. Since it is available here, in future chapters the order of error notation will be omitted for ease of reading.

The boundary conditions listed in section 2.3 can be discretized to,

$$\begin{aligned} u^i_{0,k} &= 0, \\ u^i_{N,k} &= 0, \\ u^i_{j,0} &= 0, \\ u^i_{j,M} &= 0, \end{aligned} \quad (2.20)$$

where N and M are the last values for x and y respectively.

2.5 DFT

In order to perform a Fourier transform for discrete data it is important to understand the discrete Fourier transform (DFT). Given a list of data $f(r)$ we can perform a DFT to get the Fourier transform $\hat{f}(s)$ by using the following equation

$$\hat{f}(s) = \frac{1}{\sqrt{n}} \sum_{r=1}^{n} f(r) e^{2\pi i (r-1)(s-1)/n}, \quad (2.21)$$

where n is the total number of data points are r and s are the positions of the data points in $f(r)$ and $\hat{f}(s)$ respectively. The inverse DFT is done via the following equation

$$f(r) = \frac{1}{\sqrt{n}} \sum_{r=1}^{n} \hat{f}(s) e^{-2\pi i (r-1)(s-1)/n}, \tag{2.22}$$

where n is the total number of data points and r and s are the positions of the data points in $f(r)$ and $\hat{f}(s)$ respectively.

Throughout this work I will be scaling this data to fit any assumptions made in either of the models in Chapters 3 and 4, later rescaling when the inverse transform is taken in order to keep the correct amplitude levels. The reasoning for the scaling will explained in the various PDE models.

2.6 Windowing Algorithm and Short Time Fourier Transform

2.6.1 Windowing

When working with a signal and its DFT it is important to note that the amount of data we transform gives different resolutions of the frequencies present in the section of data we transform. That is, the more data we use from the signal in the temporal domain, the higher the frequency resolution in the Fourier domain However, if we use all the data from our signal in our DFT we can end up losing a lot of information in the temporal domain if we change that data too much. This introduces the idea of windowing the data so that changes in the Fourier domain only affect small sections of the signal, with smaller windows affecting the integrity of the signal less than bigger windows. This introduces a trade off between the resolution of the frequencies and the integrity of the signal in the temporal domain. Thus, it is required to pick a window small enough to keep the signal intact, but large enough to obtain enough information about the frequencies. Another factor to consider is that if these windows are simply adjacent to one another discontinuities will again be created in the data when each window is transformed back to the temporal domain, thus there needs to be a level of overlap in order to maintain continuity.

Another type of problem is spectral leakage. This is a phenomenon that is present due to the nature of the DFT. The DFT assumes that the signal it relates to can be looped infinitely without discontinuities forming, which is often not the case for a signal which is evolving in time, especially when a window of the signal is taken.

When the signal can't be looped infinitely the frequencies around the actual frequency of interest are represented as having an amplitude higher than they should. This is clearly an undesirable situation since it means our DFT is not necessarily an accurate representation of the data.

This leads to the discussion of the different types of window functions and the windows desirable size. I will only discuss a few to introduce the ideas behind them, and then explain the reasoning as to why the window we used was chosen. To start this discussion off it is important to make it clear that the values of the DFT when window functions are applied are the result of each window of data being multiplied by the function that describes the window type and then only taking the DFT. That is we take the DFT of

$$\bar{f}(n) = \xi(n) f(n), \tag{2.23}$$

where $\bar{f}(n)$ is the signal window with the windowing function applied to it, $\xi(n)$ is the windowing function, $f(n)$ is the signal window and n is the data point position, starting at 0 and ending at N.

2.6.1.1 Rectangular window

The rectangular window effectively takes the values of the data in the window and doesn't scale them at all. That is

$$\xi(n) = \begin{cases} 0, & n < 0 \\ 1, & 0 \leq n \leq N-1 \\ 0, & n \geq N \end{cases} . \tag{2.24}$$

This windowing function leaves most of the spectral leakage in the DFT but it has minimal effect on the noise profile.

2.6.1.2 Sine window

This is the window that best describes the reasoning as to why different window types were designed. Since the DFT assumes that the signal repeats infinitely if the signal starts at zero and ends at zero there would be no discontinuity if we looped it. Thus a window function that is of the form

$$\xi(n) = \sin\left(\frac{2\pi n}{N-1}\right) \tag{2.25}$$

would create a situation where the signal appears to be able to be looped infinitely, fitting the Fourier transforms model. This reduces the spectral leakage, however, this would change the noise profile since we lose information at the edges of the windows.

2.6.1.3 Gaussian window

The Gaussian window can be described as follows,

$$\xi(n) = e^{-\frac{1}{2}\left(\frac{n-(N-1)/2}{\sigma(N-1)-2}\right)^2}, \tag{2.26}$$

where $\sigma \leq 0.5$. The Gaussian window also has the ability to successfully reduce spectral leakage with an appropriately chosen value of σ but still losing too much information to do with the noise profile which the models in this research need.

2.6.1.4 Hamming window

The Hamming window can be describe as follows,

$$\xi(n) = \alpha - \beta \cos\left(\frac{2\pi n}{N-1}\right), \tag{2.27}$$

where $\alpha = 25/46$ and $\beta = 1 - \alpha$. This window doesn't make the start and end points of the window identically equal to zero, given that they aren't in fact zero, but rather scales them to a value in a range between $-\alpha \times f(n)$ and $\alpha \times f(n)$. This brings us closer to the assumption that the windows signal can be looped infinitely without a discontinuity forming, but not exactly. This window type also reduces the spectral leakage (less so than the sine), but has less of an effect on the noise profile than the Sine window.

While spectral leakage is a problem, for our models to have an assumption about the noise profile which we can work with, we chose to work with the Hamming window. Since spectral leakage manifests itself as a form of noise and our model is designed to remove noise we would essentially be removing the spectral leakage in our processing. Thus we can justify keeping the some spectral leakage in order to keep more knowledge about the noise profile.

As for the window size and percentage of overlap for each window, these were left as another parameter to optimize in the models, as they can affect the different models results.

2.6.2 The windowing algorithm

In order to produce each of the windows with the chosen window size and level of overlap and then undo the process the following two algorithms, algorithm 1 and 2, were used. These algorithms keep track of the maximum value for two reasons, the first being for any scaling and rescaling we might have assumed in our models, and secondly, if we use a part of the signal that is deemed to be only noise we can get a good estimate of the noise profile. It is important to note that in the algorithms that WF is the windowing function, which is a hamming window function as stated.

Algorithm 1 Windowing

```
procedure WINDOWING
    i ← 0
    Position ← 0
    while i × WindowSize ≤ ClipLen do
        Window ← WF × SoundClip(Position : Position + WindowSize)
        FourierWindow ← Fourier[Window]
        RePart[i] ← Real[FourierWindow]
        ImPart[i] ← Imaginary[FourierWindow]
        Max ← Max[Abs[FourierWindow]]
        if Max > MaxValue then
            MaxValue ← Max
        i ← i + 1
        Position ← Position + WindowSize × Overlap
    return [RePart, ImPart, MaxValue]
```

Algorithm 2 UnWindow

```
procedure UNWINDOW
    Position ← 0
    for i in RePart, ImPart do
        TempClip ←InverseFourier[MaxValue×RePart[i]+MaxValue×ImPart[i]×j]
        Ave ← Average[Clip(Position:Position+WindowSize×Overlap), TempClip(0 :
    WindowSize × Overlap)]
        NonOverlapSec ← TempClip( WindowSize × Overlap : end)
        Clip(Position:Position+WindowSize) ← Ave + NonOverlapSec
        Position ← Position + WindowSize × Overlap
    return Clip
```

Once we have the fourier windows in their real and imaginary parts, RePart and ImPart respectively as the algorithm state it or $Re(STFT)$ and $Im(STFT)$ as seen in equation (2.28), which algorithm 1 outputs, it is a simple process to obtain the spectrogram at this point. The modulus of the STFT (short-time fourier transform) is described by the following equation,

$$|STFT| = (Re(STFT)^2 + Im(STFT)^2)^{\frac{1}{2}} \tag{2.28}$$

and the spectrogram is simply $|STFT|^2$.

2.7 SNR

Before we can define the SNR (signal to noise ratio), which has been briefly mentioned before, we need to define what we mean by the average power of the signal. To get the average power of a signal we take its Fourier transform, take the absolute value of each data point (this is to obtain the the magnitude/amplitude), square the values (converting it from magnitude/amplitude to power) and then take the average of the values in the bandwidth of frequencies we are interested. For this work we will use the entire available bandwidth.

The signal to noise ratio is the ratio of the average power of the signal to that of the average power of the noise, where this average is over the same bandwidth of frequencies. This can be measured by measuring the average power of the signal in a "silent" section and comparing it to the average power of the signal in a non-silent section with the following equation,

$$SNR = \frac{P_{signal}}{P_{noise}}, \tag{2.29}$$

where P_{signal} and P_{noise} are the average power of the signal and noise respectively. To convert this to decibels however we would have to covert this to a logarithmic scale as follows,

$$SNR_{dB} = 10\log_{10}(P_{signal}) - 10\log_{10}(P_{noise}). \tag{2.30}$$

This is the measure of the SNR I will be using throughout this work.

2.8 Conclusion

In this chapter I have presented various methods I will be using throughout this work. I have explained how they work and given reasoning, where a choice could be made, as to why each methods were chosen. As such, this chapter should be used as a reference point for the methodology used in this work.

Chapter 3

PDE Spectral Subtraction Model

3.1 Introduction

To adequately describe this method we are required to refer to equation (1.2). This was an equation used to describe a concept known as spectral subtraction. As a reminder, that is

$$P_x = P_y - \alpha P_n,$$

where P_x is the power spectrum of the a signal which has had noise removed from it, P_y is the power spectrum of the signal with noise, P_n, is an estimate of the power spectrum of the noise and α is simply a scaling factor for the noise estimate.

In order to use this concept we would be required to find a reasonable estimate of the noise profile. That is, we would have to find as accurate a representation as possible of the power spectrum of the noise as we can. In order to do this our approach we will be taking knowledge of the noise profile in order to diffuse the short-time Fourier transform in such a manner as to get rid of the signal and be left with an accurate prediction of the noise profile. Thus for this model we will be designing a partial differential equation which will remove the signal and leave us with an estimate of the noise profile.

3.2 Non-linear Diffusion Equation

The non-linear diffusion equation chosen and designed for the process of removing the signal is,

$$u_t = q_\omega \, C(u) \, u_{\omega\omega}, \tag{3.1}$$

where

$$C(u) = \frac{1}{1 + e^{-k(u-\eta)}}, \tag{3.2}$$

q_ω is the scaling factor for the diffusive term $u_{\omega\omega}$, and η is an estimate the maximum value of the noise in the frequency domain. For this model, since we are trying to obtain an estimate of the noise's power spectrum, we have chosen u to represent the data in the power spectrum of our signal.

The boundary conditions are,

$$\begin{aligned} u(t, 0, \tau) &= 0, \\ u(t, N, \tau) &= 0, \\ u(t, \omega, 0) &= 0, \\ u(t, \omega, M) &= 0, \end{aligned} \tag{3.3}$$

where,

$$\begin{aligned} \omega &\in [0, N], \\ \tau &\in [0, M]. \end{aligned} \tag{3.4}$$

This is assuming that the signal starts and ends in silence and that frequencies at the start and end of each Fourier window are inaudible and thus are negligible, thus all the boundary conditions can be set to zero.

This partial differential equation is ultimately a non-linear diffusion equation without a sink/source term, the only part of this that is out of the ordinary is the equation is making use of C, as described in equation (3.2) and known as, or rather a variation of, the logistics function, in its diffusion term coefficient. This equation has a number of interesting features. It, in this variation, has a minimum of 0 and a maximum of 1, and its shape is that of an "S" (sigmoid curve). In order to understand its significance we need to discuss the parameters in this function, η dictates where the sigmoid's midpoint, or the point of inflection, the value of k will dictate the steepness of the curve and if set high enough will let this function be an estimate of a step function where the step is at η. That is it behaves, with a high enough value for k (eg. $k = 10^5$), like

$$F(u) = \begin{cases} 1, & u \geq \eta \\ 0, & u < \eta. \end{cases} \tag{3.5}$$

Ultimately this function takes advantage of the fact that this variation of the diffusion equation acts in a desirable way on data that is in a certain scaling, this scaling is between 0 and 1, and thus our data will be scaled to fit this model. That is the the logistic function in the coefficient will only allow diffusion of values of u between η and 1, if k is sufficiently large. If k is small enough then it will allow for some diffusion between 0 and η. This is where it becomes clear that we could avoid this scaling process but

this leads to the issue of having to use the maximum value of the data in the logistics function instead of a 1 in the numerator. The main point of the scaling is to simplify the process because it is easier to scale the data that to constantly recheck certain values.

Thus we have a diffusion equation which leaves the data unchanged if the value of u is less than that of η, which leaves an accurate estimate of the noise. However, we will not always chose the value of k to be so high that it behaves like the step function described in equation (3.5), leaving some leniency for u to be affected in a small range around η unlike a pure threshold take at η. This is where the process gains an advantage over other methods which use a pure threshold taken at η, which only keeps data that is of a value below this level and sets any data higher than this level to be equal to the the threshold value, that is,

$$T(u) = \begin{cases} \eta, & u \geq \eta \\ u, & u < \eta. \end{cases} \tag{3.6}$$

This hard threshold function would cause extra loss of signal in many positions in the data where $u \geq \eta$ since it is common for the magnitude of the noise level at a particular frequency to not reach the maximum value we have estimated from the noise profile taken in a "silent" section of the signal. The best way to demonstrate this concept would be to show a comparison of the actual noise that is added, an estimate of the noise profile obtained via a hard threshold function, and the estimate of this profile obtained by applying equation (3.1) to a single window of the Fourier domain. This can be seen in figure 3.1 where, the green function shown in the graph represents a much better estimate of the actual noise profile, which is shown by the blue function, than that of the threshold function, shown by the yellow function in the graph. The reason for some of the values being lower than a simple threshold is because of the diffusion process, if the signal frequency is surrounded by frequencies which have little to no magnitude, then these will cause a reduction in the energy (magnitude) of that particular frequency, if the value of k is low enough so that the logistics function doesn't behave exactly like a step function.

In this model there are three parameters which require optimization, namely q_ω, η and k. This search space can be greatly reduced since in reality the only parameter which needs optimizing is k. This is because the scaling factor, q_ω, should be chosen in such a way that is the largest value it can be which will still satisfy the stability conditions of this model, which are to be shown in section 3.4. That is, if we pick a value for q_ω which is as high as possible, to speed up the process, we will get the solution required as long as k is at a value that is desirable. If we pick k to be too high we will simply be doing a hard threshold process with the threshold being η. The value of η, which is essentially the maximum value of the noise profile, is obtained from a "silent" section,

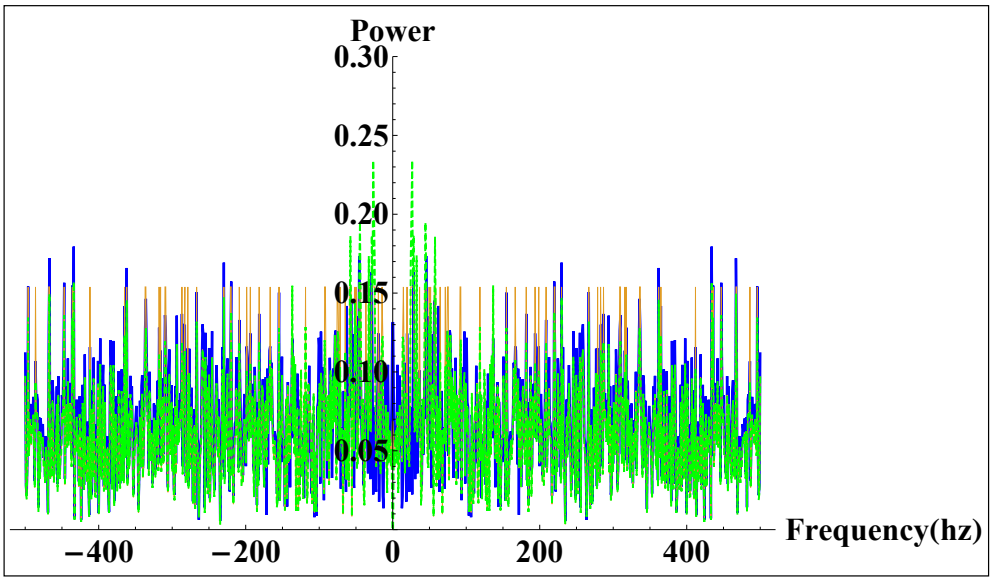

FIGURE 3.1: Actual noise profile (Blue) compared to the estimates of the noise profile produced by the threshold function noise profile (yellow) and the noise profile produced by the non-linear diffusion equation (green).

a section of the signal with only noise and seemingly no signal. Thus the optimisation process only requires that we increase k from 1 until we find a solution that is desirable and better than that of a hard threshold process. If have a situation where we have a noisy signal with no "silent" section and we have access to a signal which has no noise, we can simulate a noise profile which behaves in a similar fashion to the noise which is being observed. This will give us an estimate of a signal we can then use to run a process to find an optimal value of k, which is extremely easy to do due to a very small parameter.

3.3 Implementation

We will be using an explicit finite difference scheme, FTCS, in order to approximate a solution for our equation, (3.1). This will imply that a first order time derivative, where we will be using $\bar{u}$ as the actual value for our data and u for approximations, will be the following approximation form:

$$\frac{\partial \bar{u}}{\partial t} \approx \frac{u(t+\Delta t,\omega,\tau)-u(t,\omega,\tau)}{\Delta t} = \frac{u_{j,k}^{i+1}-u_{j,k}^{i}}{\Delta t}, \tag{3.7}$$

where i will be used as the time index (a 1 unit change in i will equate to a Δt change in time) and j and k will be used as the spatial indices (a 1 unit change in j and k will equate to a $\Delta\omega$ and $\Delta\tau$ change in space respectively). The central difference approximation for the second order spatial derivative with respect to ω will have the

same indexing and will have the form as follows:

$$\frac{\partial^2 \bar{u}}{\partial \omega^2} \approx \frac{u(t, \omega - \Delta, \tau\omega, \tau) - 2u(t, \omega, \tau) + u(t, \omega + \Delta\omega, \tau)}{\Delta\omega^2} = \frac{u^i_{j-1,k} - 2u^i_{j,k} + u^i_{j+1,k}}{\Delta\omega^2}. \tag{3.8}$$

Since there is no second order derivative in the direction of τ I will not show it's approximation.

While this equation can be done as a one dimensional diffusion process over multiple windows I have chosen to keep it over the short-time Fourier transform and thus have to account for the temporal direction, that is the time of the signal and not the time evolution of the diffusion equation.

If we replace the relative terms in equation (3.1) with the relative approximations above, while rearranging to keep the terms from each time period on the same side of the equation, we get the following:

$$u^{i+1}_{j,k} = \frac{q_\omega \Delta t}{\Delta\omega^2} C(u^i_{j,k}) \left(u^i_{j-1,k} - 2u^i_{j,k} + u^i_{j+1,k}\right) \tag{3.9}$$

,

which approximates the value of the equation 3.1 function at each time step.

The initial values for u will be the short-time Fourier transform of the noisy audio sound clip. That is,

$$u(0, \omega, \tau) = |STFT|^2, \tag{3.10}$$

where $STFT$ is the short-time Fourier transform of the audio sound clip.

The boundary conditions are stated in equation (3.3) and these are discretized to

$$\begin{aligned} u^i_{0,k} &= 0, \\ u^i_{N,k} &= 0, \\ u^i_{j,0} &= 0, \\ u^i_{j,M} &= 0. \end{aligned} \tag{3.11}$$

Using this as the FTCS approximation of we will apply algorithm 3 in section 3.3.1

3.3.1 Algorithm

The basic idea of the noise removal algorithm is shown in algorithm 3 where PDESignal-Removal is simply the FTCS approximation of equation (3.1) as described in section 3.3,

specifically equation (3.9) with equation (3.10) as the initial condition and the boundary conditions as in equation (3.11), run to a desired steady state.

Algorithm 3 Main

procedure MAIN:SPECTRAL SUBTRACTION
    SoundClip ← import the sound clip
    ClipLen ← length of *SoundClip*
    WindowSize ← Desired window size
    Overlap ← 1 − desired overlap percentage
    [*RePart, ImPart, MaxValue*] ←
      Windowing(*SoundClip ,WindowSize, ClipLen, Overlap*)
    $STFT \leftarrow (RePart^2 + ImPart^2)^{\frac{1}{2}}$
    $Phase \leftarrow Arg[RePart + i\ ImPart]$
    STFT2 ← PDESignalRemoval[STFT/MaxValue]
    STFT ← STFT - STFT2
    $RePart \leftarrow STFT \times \cos(Phase)$
    $ImPart \leftarrow STFT \times \sin(Phase)$
    SoundClip ← Unwindow[*RePart,ImPart,MaxValue*]

3.4 Boundedness conditions

We have a partial differential equation which takes advantage of a scaling of its data. That is, all the values of u are scaled between 0 and 1. In order to maintain this scaling we need to find conditions on the parameters to keep the solution in the bounds at each time step. If we were to prove that it stays in the bounds for one time we would inductively prove that it would stay within the bound for all of time. This doubles as our stability conditions since if we prove that the solution is bounded for all of time then the solution is stable. Thus if we take the implementation of the FTCS method on this equation and we bound the solution by the assumed bounds we can find a set of constraints on the parameters that for each time step the solution would be bounded,

$$0 \leq\ u_{j,k}^{i+1} = \frac{q_\omega \Delta t}{\Delta \omega^2}\, C(u_{j,k}^i)\, (u_{j-1,k}^i - 2u_{j,k}^i + u_{j+1,k}^i) \quad \leq 1. \tag{3.12}$$

With these bounds implemented, we will drop the $u_{j,k}^{i+1}$ term since it has an equation that is its equivalent, which is ultimately what we will be using to prove that $u_{j,k}^{i+1}$ is bounded under a set of constraints. From this point we will consider the different cases for which this equation may be faced. That is, when $u_{j,k}^i = 1$ and $u_{j,k}^i = 0$. Each of these cases will give us and/or confirm the constraints, given Δt and $\Delta\omega$, q_ω.

Let us however, before these cases are considered, discuss the logistics function in our equation. That is, let us discuss $C(u_{j,k}^i)$ and its extreme values. We know that this

function is bounded above by 1 and below by 0, that is, it is bounded by its maximum and minimum values as discussed in section 3.2. We can with this knowledge simplify our problem. Since $C(u)$ is bounded below by 0 and above by 1 we know that it will scale any number it is multiplied with between 0 and itself, and since this is the case we can leave it out of this analysis since it would keep the bounds we have assumed are needed for the equation as long as the rest of the equation obeyed those bounds. This leaves us with the equation in the following form,

$$0 \leq \frac{q_\omega \Delta t}{\Delta \omega^2} \left(u^i_{j-1,k} - 2u^i_{j,k} + u^i_{j+1,k}\right) \leq 1. \tag{3.13}$$

Let us first consider $u^i_{j,k} = 0$. In this case the problem becomes a rather trivial one. That is it simplifies to,

$$0 \leq \frac{q_\omega \Delta t}{\Delta \omega^2} \left(u^i_{j-1,k} + u^i_{j+1,k}\right) \leq 1. \tag{3.14}$$

From here we are simply left with the problem of the possible values of $(u^i_{j-1,k} + u^i_{j+1,k})$, that is, it can either be at a maximum of 2 or a minimum of 0 with both terms equal to 1 or 0 respectively. Since the solution for this term being zero is trivial and doesn't give any information we will only consider when it is equal to its maximum since this will give us the most extreme constraint possible. Thus, we have found the first constraint on q_ω.

$$0 \leq q_\omega \leq \frac{\Delta \omega^2}{2\Delta t}. \tag{3.15}$$

We however only care about the upper bound on the constraint since making q_ω negative would change the nature of the partial differential equation.

For the case of $u^i_{j,k} = 1$ we find that it provides the same constraint as in equation (3.15) through similar reasoning. When we use this case value we find the equation simplifies to,

$$0 \leq \frac{q_\omega \Delta t}{\Delta \omega^2} \left(u^i_{j-1,k} - 2 + u^i_{j+1,k}\right) + 1 \leq 1. \tag{3.16}$$

We find that $(u^i_{j-1,k} - 2 + u^i_{j+1,k})$ has a strictly negative value, between its maximum of 0 and minimum of -2, when both of the unknown terms are 1 or 0 respectively. Substituting this in and rearranging leaves us with the exact same constraint as in equation (3.15).

3.5 Conclusion

In this chapter I have presented a partial differential equation which, when using a audio signal's short-time Fourier transform, can obtain an estimate of the noise profile

of noisy signal in order to perform a spectral subtraction process to rid the originally observed noisy signal of noise. A discussion on how and why this method works as well as a method of how to implement this process using a finite difference scheme, FTCS, and the corresponding stability criteria for this method have been shown. This method is easy to implement and requires very little work to find the optimal values for the parameters. Besides the value of the parameter which is to be optimized, k, the only required knowledge of the noisy signal is a value, η, which is approximately equal to the maximum power of the noise in the short-time Fourier transform.

Ultimately a viable method, using partial differential equations, to obtain an estimate of the noise profile in order to perform a spectral subtraction process has been presented.

Chapter 4

Diffusion of the Spectrogram Model

4.1 Introduction

When we move back to the notion of audio signals in mathematical notation, a function as seen in equation (1.1), we see that noise manifests itself as random fluctuations in $X(t)$. This begs the question of what is the best way to deal with a function that should be seemingly smooth without these fluctuations, a function that looks more like $S(t)$. In section 2.2 I discussed two useful properties of the diffusion equation, namely its ability to remove these random fluctuations and discontinuities. The problem with the linear diffusion equation was that its steady state, given the implied boundary conditions from section 2.3, removed the signal in its entirety. However, there exists an optimal time, t_{op}, in the evolution of the diffusion equation at a point between the initial starting time, when $t = 0$, and the final time of the steady state, that is $\lim_{t\to\infty} u(t, \omega, \tau)$. This leads to the hypothesis that either at some given t_{op} or the use of some non-linear variation of the diffusion equation which would keep the integrity of the signal intact at its steady state, we would be able to obtain a smoothed out version of the initial condition, the signal without the noise.

A simple test of using the linear diffusion equation with the initial condition being the signal, with the noise already added in, and attempting to find t_{op} yields that the diffusion equation applied in this manner would simply act as a low pass filter. The reason for this is as the time evolution of the diffusion equation happens we find that the smaller fluctuations in the data are removed first. These fluctuations directly correspond to the higher frequencies. If we let the process run for too long we find that the signal is

removed, which is undesirable. Thus the best result we can find from the linear diffusion in the temporal domain is, as stated before, a low pass filter.

Since these were such poor results, in the sense that low pass filters exist and do the job better, we decided to move to the Fourier domain. When using the simple diffusion equation with one window of the Fourier domain as the initial condition it causes "blurring" between frequencies, adding to unwanted frequencies and subtracting from wanted ones due to the transfer of energy from one frequency to another. All of these being undesired situations.

However, this lead to the hypothesis which suggests it is possible to design a variation of the non-linear diffusion equation with a sink/source term, the non-linearity is to preserve the frequencies above a certain power level and the sink/source term is to remove signals below a certain threshold. This would be able to take advantage of the diffusion equation's smoothing nature while keeping the signal's frequencies if we were to work in the with the short time Fourier transform.

Thus with the correct form variation of the non-linear diffusion equation we could remove unwanted frequencies using the spectrogram of the audio clip as the initial condition. In this domain we have more variables to work with, thus we can diffuse the frequencies in the temporal direction. In other words, with a few modifications to the diffusion equation in two dimensions it is potentially possible to find coefficients of diffusion for each direction, time and frequency, and use a sink/source term to remove a large number of unwanted frequencies in this domain. After this removal process we then transform the data back to the temporal domain we would have a signal either completely devoid of, or with far less, noise.

4.2 Phase issues

In industry the general procedure for any processing done to the Fourier transform of an audio clip involves working with the phase and magnitude components. The processing, if any is done, is only to the magnitude component and the phase is generally considered unimportant, since perceptually it is deemed not noticeable by the majority of the signal processing community as suggested by Kim in [43] where he discusses the community's perception and his contrary results that results can be improved with better phase reconstruction. I have found similar results to Kim. If we are to take the Fourier transform of a test signal with and without additive Gaussian noise we can show that noise can be reduced in a simple process of taking the Fourier transform of the noisy signal, replacing its phase with the noiseless phase of the original signal and then taking

the inverse Fourier transform of the noisy signal's magnitude and the original signal's phase. This is shown in figures 4.1, 4.2 and 4.3, which are the spectrograms of the noiseless signal, the noisy signal and the noisy signal with a replaced phase, that is, it has been transformed to the Fourier domain and then we replaced its noisy phase with the phase of the noiseless signal, respectively. It can be clearly seen that certain sections of the noise are lower than others when the noiseless phase is used instead of the noisy phase. It is however very difficult to process the phase, and thus we have clear motivation as to why it is important to use a method which does not need to use the phase.

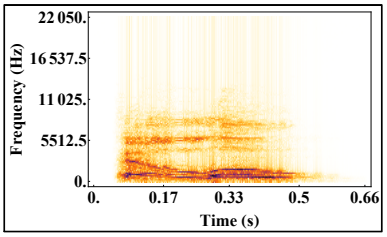

FIGURE 4.1: Spectrogram of noiseless signal.

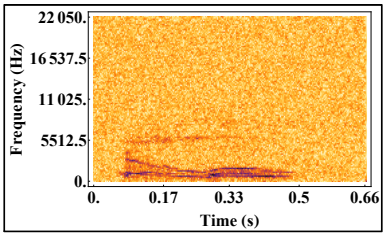

FIGURE 4.2: Spectrogram of the noisy signal.

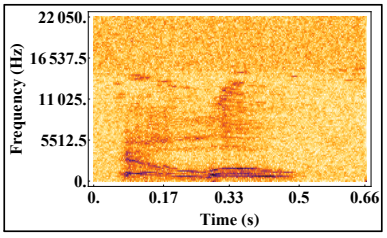

FIGURE 4.3: Spectrogram when the noiseless phase was used to recreate the sound from the noisy signal.

Since it becomes near impossible to process the phase while keeping valuable information, we will instead show the relation of the magnitude and phase to that of the real and imaginary components. The phase and magnitude can be represented by the following two equations,

$$P = \arccos\left(\frac{Re}{Im}\right), \tag{4.1}$$

$$M = |Re + Im|, \tag{4.2}$$

where P is the phase, M is the magnitude, Re is the real component of the Fourier transform and Im is the imaginary component of the Fourier transform. The inverse of this relation is given by:

$$Re + Im = M\ \cos(P) + M\ i\ \sin(P). \tag{4.3}$$

Now that we have a relation between the magnitude and phase and the real and imaginary components we can ignore working with the magnitude and phase. Instead we will just use the imaginary and real components of the Fourier transform in our spectrogram since it gives us something that is manageable to process, unlike the phase itself.

We will use a different type of spectrogram for the sake of this work. Instead of the spectrograms representing the actual frequency magnitudes it will use the two different representations, using the real part and imaginary parts of the Fourier transforms respectively. I will typically refer to both of these representations of the spectrograms as the $Im(STFT)$ and $Re(STFT)$ spectrograms. These two representations will have both positive and negative values unlike a normal spectrogram. However the magnitude is still the most intuitive representation of the data and is used to display the results.

The $Im(STFT)$ and $Re(STFT)$ will be used as initial conditions in our processing via the diffusion equation discussed in Section 4.3. In other words we will process two sets of data with the equation, the imaginary spectrogram and the real spectrogram. Our initial condition will be

$$u_1(0, \omega, \tau) = Re(STFT), \tag{4.4}$$

and

$$u_2(0, \omega, \tau) = Im(STFT), \tag{4.5}$$

where u_1 is used when processing $Re(STFT)$ and u_2 is when $Im(STFT)$. Both of these are a $t = 0$ and ω and τ are the directional variables, relating to frequency and temporal directions respectively. However, since the same process is taken on both we will collectively state that u is our data, meaning that it is being done to both the real and imaginary spectrograms in parallel. Taking this approach will remove the issue of not being able to obtain the correct noiseless phase of the Fourier transform since we can obtain the inverse fourier transform with just this information.

4.3 Non-linear Diffusion Equation

The non-linear diffusion equation chosen and designed to be used in the procedure to remove the noise is,

$$u_t = q_\omega(1-|u|)u_{\omega\omega} + q_\tau(1-|u|)u_{\tau\tau} + q_s\Gamma(u)u(1-u)(1+u)(u-\eta)(u+\eta), \tag{4.6}$$

where

$$\Gamma(u) = e^{-\beta|u|exp(-\gamma\,\eta)}, \tag{4.7}$$

u_t is the first derivative of u with respect to t (time evolution of the non-linear diffusion equation), $u_{\omega\omega}$ is the second derivative with respect to ω (frequency), $u_{\tau\tau}$ is the second derivative with respect to τ (time evolution of the audio signal's frequencies, or temporal direction), q_ω is the scaling factor in the diffusion coefficient of the $u_{\omega\omega}$ term, q_τ is the scaling factor in the diffusion coefficient of the $u_{\tau\tau}$ term, q_s is the scaling factor in coefficient of the sink term, and η is an estimate the maximum value of the noise in the frequency domain. The boundary conditions are,

$$\begin{aligned} u(t,0,\tau) &= 0, \\ u(t,N,\tau) &= 0, \\ u(t,\omega,0) &= 0, \\ u(t,\omega,M) &= 0, \end{aligned} \tag{4.8}$$

where,

$$\begin{aligned} \omega &\in [0,N], \\ \tau &\in [0,M]. \end{aligned} \tag{4.9}$$

Each term will be discussed in detail and the purpose of each variable made clear. It is important to note that all the data that will be processed via equation (4.6) is to be scaled between -1 and 1, which is fundamentally important to this model. The first term in equation (4.6) captures the diffusion in the ω (frequency) direction and has three components: the scaling factor q_ω, the non-linearity $(1-|u|)$ and the diffusion itself $u_{\omega\omega}$. The scaling factor is simply a parameter which is chosen according to the given information about the noise levels, specifically the SNR. The higher the SNR the higher the value of the coefficient. In general this parameter's value will be close to or in fact zero as this would cause 'blurring' of energy between the different frequencies that should not be there as described in Section 4.1. The diffusion component has been chosen due to its smoothing nature which is ultimately what is removing the noise. This is because it smooths out the function by removing the random fluctuations in the data, this alone however cannot solve the issue and thus there is a sink/source term in the diffusion equation.

The most interesting part of this term is $(1-|u|)$. It has the purpose of further controlling

the level of diffusion based on what is effectively a value proportional to that of the SNR for a particular frequency. I say effectively, but it is not completely proportional to the SNR. Due to the scaling of the data to be between -1 and 1 this is essentially a scaling factor which lowers the value of the term when the absolute vale of u is close to 1. That is, if the value of u is close to the maximum value of the data, this implies there is a large amount of signal at that frequency and it is less likely to be noise. The closer it is to that maximum value the more likely it is to be signal and less likely to be noise, thus closely related to a high SNR, thus, it is desirable for there to be less diffusion in that particular frequency. In other frequencies where the value of u is closer to zero there is a higher likelihood of it being noise, thus more diffusion is desired, and thus far less scaling down of the term.

The second term in equation (4.6) is designed in a similar fashion to that of the first term, however this term captures the diffusion in the τ (time evolution of the audio signal's frequencies, or temporal direction) direction. It has similar components to the first term: the the scaling factor q_τ, the non-linearity $(1 - |u|)$ and the diffusion itself $u_{\tau\tau}$. The difference here is that blurring, or smoothing, of the frequency in its temporal direction is desired and thus having a higher value for the scaling factor is desirable. It is however still chosen according to the SNR as less diffusion is desired when there is less noise. The rest of the term behaves the same as the first term, except that its direction is different.

The sink/source term is designed to work so as to add or subtract energy depending on the energy level of u. If its energy level is considered to be in the noisy band, $-\eta \leq u \leq \eta$, as to get this level to zero. If its energy level is outside of that band, $-1 \leq u \leq -\eta$ or $\eta \leq u \leq 1$, then it will subtract or add energy, respectively, as to drop or raise these energy levels to 1 or -1 respectively. A better grasp of this can be taken from figure 4.4 which shows the direction of the energy flow, $f(u)$, for the different values of u when the the parameters in the sink/source term are $\gamma = 6$, $\beta = 100$, $\eta = 0.5$ and the product of q_s and Δt is equal to 10, that is $q_s \times \Delta t = 10$.

This term requires the careful picking of values for 3 variables, namely β, γ and q_s. When correctly chosen the coefficient of this term, $\Gamma(u)$, coupled with q_s, will be extremely small or zero when the energy levels of u are greater, in absolute value, than η and has a significant value when the energy level of u is smaller, in absolute value, than η. The reasoning behind this is to keep signal unaffected when the signal is considered to have more energy than that of the noise.

Since this sink/source term was designed specifically for the Ir-Re model that was mentioned in Section 4.2 it might be useful to note that with a slight adaptation, since it would no longer be required to deal with negative values, one could use another form

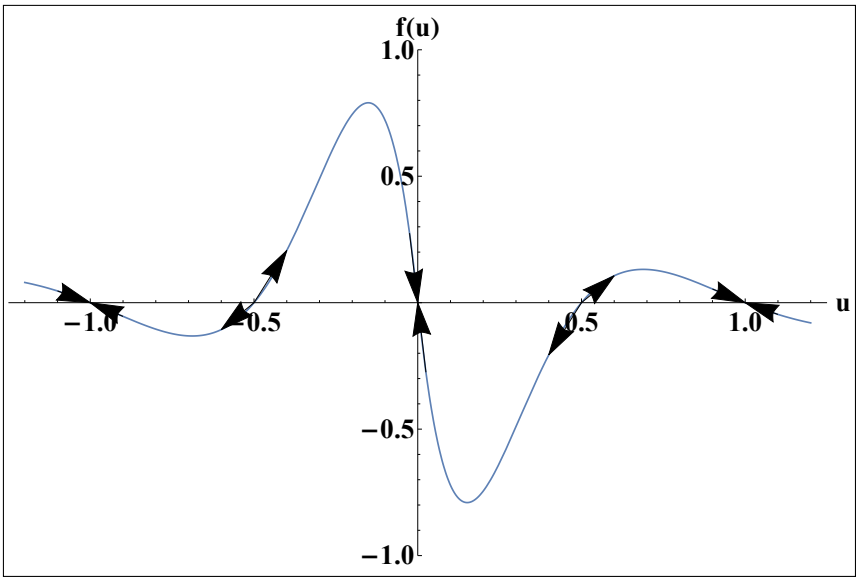

FIGURE 4.4: Phase of sink term.

of the sink/source term to be used with the ordinary spectrogram. This would however require that we would have to use the noisy signals phase. This was found to have inferior results and has the same pitfalls as described in Section 4.2, thus it has been omitted in favour of this model.

A few assumptions are made on the audio clip to handle the boundary conditions. The human ear typically only hears sounds between the frequency range of $20hz$ to $20\,000hz$. This leads to the assumption that the frequencies outside of this range can be set to 0 with little to no repercussions. Not all the windows used in processing will have the resolution of the $20\,000hz$ frequency and thus a simplification of assuming the last frequency available in the window's resolution will be outside of this range. Taking these assumptions into account we can impose the Dirichlet boundary condition of $u = 0$ when $\omega = 0$ or $\omega = N$, where N is the last available frequency in the window's resolution. Assuming the audio clip starts off with silence and ends with silence we can impose the Dirichlet boundary condition of $u = 0$ when $\tau = 0$ and $\tau = M$, where M is the last available time window. This choice of boundary condition also helps with the attenuation of edges of the different windows, which is the reason for the rectangular window, as previously discussed in Section 2.6.

4.3.1 Implementation

Since we will be doing an explicit scheme, forward difference in time and central difference in space scheme (FTCS), we would have to convert the problem into its approximation form. This will imply that a first order time derivative, where we will be using $\bar{u}$ as the

actual value for our data and u for approximations, will have the following approximation form:

$$\frac{\partial \bar{u}}{\partial t} \approx \frac{u(t+\Delta t,\omega,\tau)-u(t,\omega,\tau)}{\Delta t} = \frac{u^{i+1}_{j,k}-u^{i}_{j,k}}{\Delta t}, \tag{4.10}$$

where i will be used as the time index (a 1 unit change in i will equate to a Δt change in time) and j and k will be used as the spatial indices (a 1 unit change in j and k will equate to a $\Delta\omega$ and $\Delta\tau$ change in space respectively). The central difference approximation for the second order spatial derivative with respect to ω will have the same indexing and will have the form as follows:

$$\frac{\partial^2 \bar{u}}{\partial \omega^2} \approx \frac{u(t,\omega-\Delta\omega,\tau)-2u(t,\omega,\tau)+u(t,\omega+\Delta\omega,\tau)}{\Delta\omega^2} = \frac{u^{i}_{j-1,k}-2u^{i}_{j,k}+u^{i}_{j+1,k}}{\Delta\omega^2}. \tag{4.11}$$

The central difference approximation for the second order spatial derivative with respect to τ will have the same indexing and will have the form as follows:

$$\frac{\partial^2 \bar{u}}{\partial \tau^2} \approx \frac{u(t,\omega,\tau-\Delta\tau)-2u(t,\omega,\tau)+u(t,\omega,\tau+\Delta\tau)}{\Delta\tau^2} = \frac{u^{i}_{j,k-1}-2u^{i}_{j,k}+u^{i}_{j,k+1}}{\Delta\tau^2}. \tag{4.12}$$

If we replace the relative terms in equation (4.6) with the relative approximations above, while rearranging to keep the terms from each time period on the same side of the equation, we get the following:

$$\begin{aligned} u^{i+1}_{j,k} = {} & \frac{q_\omega \Delta t}{\Delta\omega^2}\left(1-|u^{i}_{j,k}|\right)\left(u^{i}_{j-1,k}-2u^{i}_{j,k}+u^{i}_{j+1,k}\right) \\ & + \frac{q_\tau \Delta t}{\Delta\tau^2}\left(1-|u^{i}_{j,k}|\right)\left(u^{i}_{j,k-1}-2u^{i}_{j,k}+u^{i}_{j,k+1}\right) \\ & + \Delta t\, q_s \Gamma(u^{i}_{j,k}) u^{i}_{j,k}(1-u^{i}_{j,k})(1+u^{i}_{j,k})(u^{i}_{j,k}-\eta)(u^{i}_{j,k}+\eta) \\ & + u^{i}_{j,k}, \end{aligned} \tag{4.13}$$

which approximates the value of the non-linear diffusion function at each time step.

The initial values for u were discussed in section 4.2. Specifically that of equations (4.4) and (4.5).

The boundary conditions are stated in equation (4.8) and these are discretized to

$$\begin{aligned} u^{i}_{0,k} &= 0, \\ u^{i}_{N,k} &= 0, \\ u^{i}_{j,0} &= 0, \\ u^{i}_{j,M} &= 0. \end{aligned} \tag{4.14}$$

With all this information we can perform the algorithm described in section 4.3.2.

4.3.2 Algorithm

The basic idea of the noise removal algorithm is shown in algorithm 4 where PDENoiseRemoval is simply the FTCS approximation of equation (4.3) given by equations (4.13) and (4.14) as described in section 4.3.1 run to a desired steady state.

Algorithm 4 Main

```
1: procedure MAIN:SPECTROGRAM MODEL
2:     SoundClip ← import the sound clip
3:     ClipLen ← length of SoundClip
4:     WindowSize ← Desired window size
5:     Overlap ← 1 − desired overlap percentage
6:     [RePart, ImPart, MaxValue] ←
7:         Windowing(SoundClip ,WindowSize, ClipLen, Overlap)
8:     [RePart, ImPart] ← PDENoiseRemoval[RePart/MaxValue, ImPart/MaxValue]
9:     SoundClip ← UnWindow[RePart, ImPart, MaxValue]
```

4.3.3 Conditions for boundedness

Since equation 4.6 was designed to take advantage of a scaling of the data such that $-1 \leq u \leq 1$ there is need to provide the conditions for boundedness to any parameters chosen and a derivation thereof.

If we are to find suitable constraints to apply to each of these terms we will have to do an in depth analysis of the numerical approximation we will be using to estimate the values of the data at each time step.

We have the assumption in our model that u needs to be between -1 and 1 and we are given our changes in time and space (Δt, $\Delta\omega$ and $\Delta\tau$), this leaves one factor to be obtained, the constraints on the scaling factors q_ω, q_τ and q_s. If we were to prove that with a given constraint on each of these variables that at the next time step would stay in the given bounds of u we would have an inductive proof that it would stay in those bounds for each time step. This serves a double purpose of proving both this boundedness and thus, as a consequence of this boundedness, we have that the stability of the method is implied. Thus we look to the following inequality to find conditions for these constraints:

$$\begin{aligned} -1 \leq & \frac{q_\omega \Delta t}{\Delta\omega^2}\left(1-|u^i_{j,k}|\right)\left(u^i_{j-1,k}-2u^i_{j,k}+u^i_{j+1,k}\right) \\ & + \frac{q_\tau \Delta t}{\Delta\tau^2}\left(1-|u^i_{j,k}|\right)\left(u^i_{j,k-1}-2u^i_{j,k}+u^i_{j,k+1}\right) \\ & + \Delta t\, q_s \Gamma(u^i_{j,k}) u^i_{j,k}(1-u^i_{j,k})(1+u^i_{j,k})(u^i_{j,k}-\eta)(u^i_{j,k}+\eta) \\ & + u^i_{j,k} \qquad\qquad \leq 1. \end{aligned} \tag{4.15}$$

From this inequality the necessary constraints on q_ω, q_τ and q_s can be found.

From this point second order derivative approximations will be abbreviated with $(u^i_{j-1,k} - 2u^i_{j,k} + u^i_{j+1,k}) = \delta^2_\omega$ and $(u^i_{j,k-1} - 2u^i_{j,k} + u^i_{j,k+1}) = \delta^2_\tau$.

The inequality (4.15) gives rise to a number of extreme situations. In particular when the various values of $u^i_{j,k}$ are -1, 0 or 1 respectively.

If $u^i_{j,k} = 1$ or $u^i_{j,k} = -1$ there is a trivial solution where $u^{i+1}_{j,k} = u^i_{j,k}$. This preserves the boundedness.

When $u^i_{j,k} = 0$ the scaling factor is immediately equal to 1. This leads to the following relationship the constraint between the two variables q_ω and q_τ.

$$\frac{-\Delta\omega^2}{\Delta t\delta_\omega} - \frac{q_\tau\Delta\omega^2}{\Delta\tau^2\delta_\omega}\,\delta_\tau \le q_\omega \quad \le \frac{\Delta\omega^2}{\Delta t\delta_\omega} - \frac{q_\tau\Delta\omega^2}{\Delta\tau^2\delta_\omega}\,\delta_\tau. \tag{4.16}$$

$$\frac{-\Delta\tau^2}{\Delta t\delta_\tau} - \frac{q_\omega\Delta\tau^2}{\Delta\omega^2\delta_\tau}\,\delta_\omega \le q_\tau \quad \le \frac{\Delta\tau^2}{\Delta t\delta_\tau} - \frac{q_\omega\Delta\tau^2}{\Delta\omega^2\delta_\tau}\,\delta_\omega. \tag{4.17}$$

Taking into account the most limiting extreme case is where both δ_ω and δ_τ are equal to 2 and assuming that both the diffusion coefficients must stay positive we find that the imposed constraints are:

$$q_\omega \le \frac{\Delta\omega^2}{2\Delta t} - \frac{q_\tau\Delta\omega^2}{\Delta\tau^2} \text{ for } q_\tau \in [0, \frac{\Delta\tau^2}{2\Delta t}]. \tag{4.18}$$

$$q_\tau \le \frac{\Delta\tau^2}{2\Delta t} - \frac{q_\omega\Delta\tau^2}{\Delta\omega^2} \text{ for } q_\omega \in [0, \frac{\Delta\omega^2}{2\Delta t}]. \tag{4.19}$$

The last of these cases is when is when $u^i_{j,k}$ is not necessarily identically equal to any of -1, 0 or 1. Let us consider the original numerical approximation again:

$$\begin{aligned} u^{i+1}_{j,k} = {} & \frac{q_\omega\Delta t}{\Delta\omega^2}\,(1 - |u^i_{j,k}|)\,(u^i_{j-1,k} - 2u^i_{j,k} + u^i_{j+1,k}) \\ & + \frac{q_\tau\Delta t}{\Delta\tau^2}\,(1 - |u^i_{j,k}|)\,(u^i_{j,k-1} - 2u^i_{j,k} + u^i_{j,k+1}) \\ & + \Delta t\; q_s\Gamma(u^i_{j,k})u^i_{j,k}(1 - u^i_{j,k})(1 + u^i_{j,k})(u^i_{j,k} - \eta)(u^i_{j,k} + \eta) \\ & + u^i_{j,k}, \end{aligned} \tag{4.20}$$

In our work we have the implied bounds from the scaling factor and thus it will be sufficient to find the implied constraints when $-1 \le u^{i+1}_{j,k} \le 1$. We will be splitting this up into two cases.

Case 1: $u^i_{j,k} > 0$

$$\begin{aligned}
&\frac{q_\omega \Delta t}{\Delta \omega^2}\,(1-u^i_{j,k})\,(u^i_{j-1,k}-2u^i_{j,k}+u^i_{j+1,k})\\
&+\frac{q_\tau \Delta t}{\Delta \tau^2}\,(1-u^i_{j,k})\,(u^i_{j,k-1}-2u^i_{j,k}+u^i_{j,k+1})\\
&+\Delta t\; q_s \Gamma(u^i_{j,k})u^i_{j,k}(1-u^i_{j,k})(1+u^i_{j,k})(u^i_{j,k}-\eta)(u^i_{j,k}+\eta)\\
&+u^i_{j,k}.
\end{aligned} \tag{4.21}$$

Case 2: $u^i_{j,k} < 0$

$$\begin{aligned}
&\frac{q_\omega \Delta t}{\Delta \omega^2}\,(1+u^i_{j,k})\,(u^i_{j-1,k}-2u^i_{j,k}+u^i_{j+1,k})\\
&+\frac{q_\tau \Delta t}{\Delta \tau^2}\,(1+u^i_{j,k})\,(u^i_{j,k-1}-2u^i_{j,k}+u^i_{j,k+1})\\
&+\Delta t\; q_s \Gamma(u^i_{j,k})u^i_{j,k}(1-u^i_{j,k})(1+u^i_{j,k})(u^i_{j,k}-\eta)(u^i_{j,k}+\eta)\\
&+u^i_{j,k}.
\end{aligned} \tag{4.22}$$

From here we will find the constraints. We will show that we have a constraint in the form of an upper bound for q_s in case 1 by showing that $u^{i+1}_{j,k}$ is bounded above by 1 and that this constraint is reiterated in case 2 by showing $u^{i+1}_{j,k}$ is bounded from below by -1. The constraint that q_s is bounded below by zero is implied by the assumptions of the model.

Case 1: We will assume $u^i_{j,k}$ is bounded, by the assumptions of the model, and show that $u^{i+1}_{j,k}$ is bounded as an inductive step. Factorizing the right hand terms of equation (4.21) we find:

$$\begin{aligned}
&(1-u^i_{j,k})(\frac{q_\omega \Delta t}{\Delta \omega^2}\,(u^i_{j-1,k}-2u^i_{j,k}+u^i_{j+1,k})+\frac{q_\tau \Delta t}{\Delta \tau^2}(u^i_{j,k-1}-2u^i_{j,k}+u^i_{j,k+1})\\
&+\Delta t\; q_s \Gamma(u^i_{j,k})u^i_{j,k}(1+u^i_{j,k})(u^i_{j,k}-\eta)(u^i_{j,k}+\eta))+u^i_{j,k}.
\end{aligned} \tag{4.23}$$

For ease of notation we will use the following equation:

$$(1-u^i_{j,k})A+u^i_{j,k}, \tag{4.24}$$

where,

$$\begin{aligned}
A=(&\frac{q_\omega \Delta t}{\Delta \omega^2}\,(u^i_{j-1,k}-2u^i_{j,k}+u^i_{j+1,k})+\frac{q_\tau \Delta t}{\Delta \tau^2}(u^i_{j,k-1}-2u^i_{j,k}+u^i_{j,k+1})\\
&+\Delta t\; q_s \Gamma(u^i_{j,k})u^i_{j,k}(1+u^i_{j,k})(u^i_{j,k}-\eta)(u^i_{j,k}+\eta)).
\end{aligned} \tag{4.25}$$

It is sufficient to show that if $0 \leq A \leq 1$ then $0 \leq u_{j,k}^{i+1} \leq 1$. Since we are seeking an upper bound we are able to consider the triangle inequality,

$$||a + b + c|| \leq ||a|| + ||b|| + ||c||. \tag{4.26}$$

The non-linear polynomial, $u_{j,k}^{i}(1 + u_{j,k}^{i})(u_{j,k}^{i} - \eta)(u_{j,k}^{i} + \eta))$, is bounded above by 2 (when $u = 1$ and $\eta = 0$).

$$\begin{aligned} &\frac{q_\omega \Delta t}{\Delta \omega^2}\,(u_{j-1,k}^{i} - 2u_{j,k}^{i} + u_{j+1,k}^{i}) + \frac{q_\tau \Delta t}{\Delta \tau^2}(u_{j,k-1}^{i} - 2u_{j,k}^{i} + u_{j,k+1}^{i}) \\ &+ \Delta t\; q_s \Gamma(u_{j,k}^{i}) u_{j,k}^{i}(1 + u_{j,k}^{i})(u_{j,k}^{i} - \eta)(u_{j,k}^{i} + \eta) \\ &\leq \frac{q_\omega \Delta t}{\Delta \omega^2}\,(u_{j-1,k}^{i} - 2u_{j,k}^{i} + u_{j+1,k}^{i}) + \frac{q_\tau \Delta t}{\Delta \tau^2}(u_{j,k-1}^{i} - 2u_{j,k}^{i} + u_{j,k+1}^{i}) \\ &+ 2\Delta t\; q_s \Gamma(u_{j,k}^{i}). \end{aligned} \tag{4.27}$$

At this point it becomes important to find an upper bound for $\Gamma(u)$. We know that

$$\Gamma(u) = e^{-\beta |u| exp(-\gamma\, \eta)},$$

thus if we can prove that the exponent of e is always negative, then we know that $\Gamma(u)$ has an upper bound of 1. Since β is always chosen to be positive, the absolute value of u, by definition, is positive, and $exp(-\gamma\ \eta)$ is always positive, then the exponent will always be negative. Thus we know that $\Gamma(u)$ is bounded above by 1, and we can use this value in our equation,

$$\begin{aligned} &\frac{q_\omega \Delta t}{\Delta \omega^2}\,(u_{j-1,k}^{i} - 2u_{j,k}^{i} + u_{j+1,k}^{i}) + \frac{q_\tau \Delta t}{\Delta \tau^2}(u_{j,k-1}^{i} - 2u_{j,k}^{i} + u_{j,k+1}^{i}) + 2\Delta t\; q_s \Gamma(u_{j,k}^{i}) \\ &\leq \frac{q_\omega \Delta t}{\Delta \omega^2}\,(u_{j-1,k}^{i} - 2u_{j,k}^{i} + u_{j+1,k}^{i}) + \frac{q_\tau \Delta t}{\Delta \tau^2}(u_{j,k-1}^{i} - 2u_{j,k}^{i} + u_{j,k+1}^{i}) + 2\Delta t\; q_s. \end{aligned} \tag{4.28}$$

Since $u_{j,k}^{i} > 0$, we know $(u_{j-1,k}^{i} - 2u_{j,k}^{i} + u_{j+1,k}^{i}) \leq 2$ and $(u_{j,k-1}^{i} - 2u_{j,k}^{i} + u_{j,k+1}^{i}) \leq 2$, hence,

$$\begin{aligned} &\frac{q_\omega \Delta t}{\Delta \omega^2}\,(u_{j-1,k}^{i} - 2u_{j,k}^{i} + u_{j+1,k}^{i}) + \frac{q_\tau \Delta t}{\Delta \tau^2}(u_{j,k-1}^{i} - 2u_{j,k}^{i} + u_{j,k+1}^{i}) + 2\Delta t\; q_s \\ &\leq 2\frac{q_\omega \Delta t}{\Delta \omega^2}\; + 2\frac{q_\tau \Delta t}{\Delta \tau^2} + 2\Delta t\; q_s \leq 1. \end{aligned} \tag{4.29}$$

This leaves us with an upper bound of q_s which is dependent on the choices of both q_ω and q_τ.

$$q_s \leq \frac{1}{2\Delta t} - \frac{q_\omega}{\Delta \omega^2} - \frac{q_\tau}{\Delta \tau^2}. \tag{4.30}$$

Case 2: We will assume $u_{j,k}^{i}$ is bounded, by the assumptions of the model, and show that $u_{j,k}^{i+1}$ is bounded as an inductive step. Factorizing the right hand terms of equation

(4.22) we find:

$$\begin{aligned}(1+u^i_{j,k})(\frac{q_\omega \Delta t}{\Delta\omega^2}\ (u^i_{j-1,k}-2u^i_{j,k}+u^i_{j+1,k})+\frac{q_\tau \Delta t}{\Delta\tau^2}(u^i_{j,k-1}-2u^i_{j,k}+u^i_{j,k+1})\\ +\Delta t\ q_s\Gamma(u^i_{j,k})u^i_{j,k}(1-u^i_{j,k})(u^i_{j,k}-\eta)(u^i_{j,k}+\eta))+u^i_{j,k}.\end{aligned} \tag{4.31}$$

For ease of notation we will use the following equation:

$$(1+u^i_{j,k})B+u^i_{j,k}, \tag{4.32}$$

where,

$$\begin{aligned}B=(\frac{q_\omega \Delta t}{\Delta\omega^2}\ (u^i_{j-1,k}-2u^i_{j,k}+u^i_{j+1,k})+\frac{q_\tau \Delta t}{\Delta\tau^2}(u^i_{j,k-1}-2u^i_{j,k}+u^i_{j,k+1})\\ +\Delta t\ q_s\Gamma(u^i_{j,k})u^i_{j,k}(1-u^i_{j,k})(u^i_{j,k}-\eta)(u^i_{j,k}+\eta)).\end{aligned} \tag{4.33}$$

By similar logic of that used in Case 1, it is sufficient to show that if $-1 \leq B \leq 0$ then $-1 \leq u^{i+1}_{j,k} \leq 0$.

The non-linear polynomial, $u^i_{j,k}(1-u^i_{j,k})(u^i_{j,k}-\eta)(u^i_{j,k}+\eta))$, is bounded below by -2 (when $u=-1$ and $\eta=0$).

$$\begin{aligned}&\frac{q_\omega \Delta t}{\Delta\omega^2}\ (u^i_{j-1,k}-2u^i_{j,k}+u^i_{j+1,k})+\frac{q_\tau \Delta t}{\Delta\tau^2}(u^i_{j,k-1}-2u^i_{j,k}+u^i_{j,k+1})\\ &+\Delta t\ q_s\Gamma(u^i_{j,k})u^i_{j,k}(1+u^i_{j,k})(u^i_{j,k}-\eta)(u^i_{j,k}+\eta)\\ &\geq \frac{q_\omega \Delta t}{\Delta\omega^2}\ (u^i_{j-1,k}-2u^i_{j,k}+u^i_{j+1,k})+\frac{q_\tau \Delta t}{\Delta\tau^2}(u^i_{j,k-1}-2u^i_{j,k}+u^i_{j,k+1})\\ &-2\Delta t\ q_s\Gamma(u^i_{j,k}).\end{aligned} \tag{4.34}$$

We know that $\Gamma(u)$ is bounded above by 1,

$$\begin{aligned}&\frac{q_\omega \Delta t}{\Delta\omega^2}\ (u^i_{j-1,k}-2u^i_{j,k}+u^i_{j+1,k})+\frac{q_\tau \Delta t}{\Delta\tau^2}(u^i_{j,k-1}-2u^i_{j,k}+u^i_{j,k+1})-2\Delta t\ q_s\Gamma(u^i_{j,k})\\ &\geq \frac{q_\omega \Delta t}{\Delta\omega^2}\ (u^i_{j-1,k}-2u^i_{j,k}+u^i_{j+1,k})+\frac{q_\tau \Delta t}{\Delta\tau^2}(u^i_{j,k-1}-2u^i_{j,k}+u^i_{j,k+1})\\ &-2\Delta t\ q_s.\end{aligned} \tag{4.35}$$

Since $u^i_{j,k}<0$, we know $(u^i_{j-1,k}-2u^i_{j,k}+u^i_{j+1,k}) \geq -2$ and $(u^i_{j,k-1}-2u^i_{j,k}+u^i_{j,k+1}) \geq -2$, hence,

$$\begin{aligned}&\frac{q_\omega \Delta t}{\Delta\omega^2}\ (u^i_{j-1,k}-2u^i_{j,k}+u^i_{j+1,k})+\frac{q_\tau \Delta t}{\Delta\tau^2}(u^i_{j,k-1}-2u^i_{j,k}+u^i_{j,k+1})-2\Delta t\ q_s\\ &\geq -2\frac{q_\omega \Delta t}{\Delta\omega^2}\ -2\frac{q_\tau \Delta t}{\Delta\tau^2}-2\Delta t\ q_s \geq -1.\end{aligned} \tag{4.36}$$

This leaves us with the same upper bound for q_s as in case 1.

$$q_s \leq \frac{1}{2\Delta t} - \frac{q_\omega}{\Delta\omega^2} - \frac{q_\tau}{\Delta\tau^2}. \tag{4.37}$$

Thus in both case 1 and case 2 we have the same constraint implied on q_s from above and the constraint of zero from below from the assumptions of the model.

While this formulation of the constraint on q_s is useful and gives us a proof for constraints which keep the model in the bounds that are required from the assumptions of the model it is useful to note that this constraint is much more limiting than it should be due to the nature of γ, β and η being unknown and thus it can be relaxed. That is, when γ, β and η are known we can create the constraint in the following form,

$$q_s \leq \frac{1}{\aleph}\left(\frac{1}{\Delta t} - \frac{2q_\omega\Delta t}{\Delta\omega^2} - \frac{2q_\tau 2\Delta t}{\Delta\tau^2}\right), \tag{4.38}$$

where

$$\aleph = \mathrm{Max}[\Gamma(u^i_{j,k})\ u^i_{j,k}(1+u^i_{j,k})(u^i_{j,k}-\eta)(u^i_{j,k}+\eta))]. \tag{4.39}$$

It is important to note that γ and β are chosen so that $\aleph$ is generally very large making the constraint on q_s much less stringent. Examples of these values as is used in this work are $\gamma = 6$ and $\beta = 250$.

4.4 Method for finding optimal zones for parameters

This model requires that a lot of values for the different parameters are chosen correctly. This leads us to the problem of picking a value from the correct range for each of these parameters.

Each of the parameters' values need to be chosen to match up with the noise profile, to do this we will need to experimentally explore different values for each of the parameters in the model, that is, for each threshold value of the noise we need to find an optimal range for the parameters. We do this by running the process for a sample audio signal, to which we have added noise, over different ranges of values for each of the parameters. We are using a signal which is not contaminated by noise as to have a benchmark to which we can compare our results to. By doing this we have a error margin which we can minimize in order to show that there is an optimal range for the parameters for each different noise threshold. In this work the error margin we chose to minimize was the l^2-norm of the difference between the spectrogram of the original, uncontaminated, signal and the signal we had processed after adding noise to it.

To reduce the parameter search space for the parameters optimized we fixed some of less important parameters to be a scaled version of another, more important and related, parameter. For example, in equation (4.6) the two diffusion scaling factors are closely related, however the q_ω coefficient is far less important than the q_τ coefficient and was set to be scaled as follows: $q_\omega = q_\tau \times 10^{-9}$, this is because no diffusion in the ω direction was found to be sub-optimal, however the diffusion in this direction had to be a much smaller than in the temporal direction since it would cause "blurring" of the frequencies, which is in direct conflict with the assumptions of the model. Optimal values for the parameters in the $\Gamma(u)$ coefficient in equation (4.6) are much easier to find since they rely completely on the value of η and not the error margin, these were fixed to be $\gamma = 6$ and $\beta = 250$. This results in the phase portrait of the sink/source term (which assumes a η value of 0.5) as seen in 4.5. From this figure it can be seen that the only time the sink/source term, with the give β and γ values, affects the power of u is when the value of u is between $-\eta$ and η.

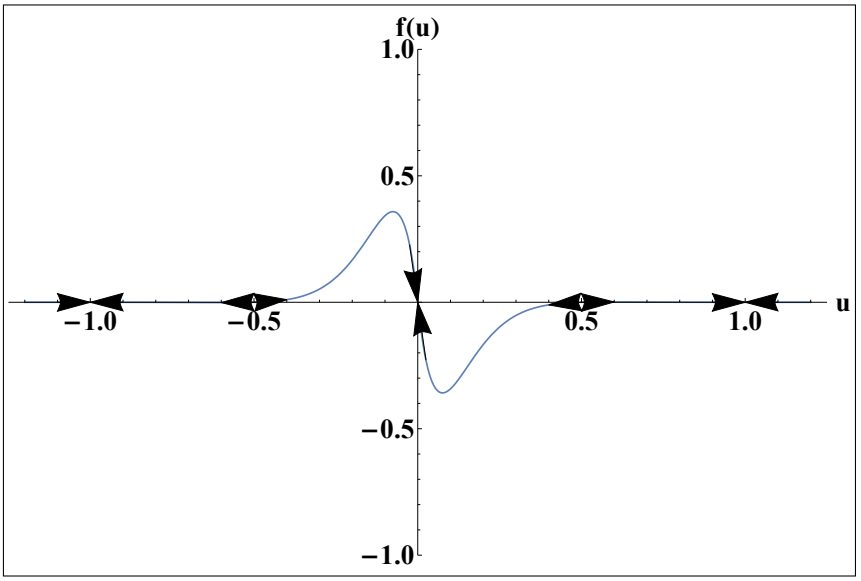

FIGURE 4.5: Phase of sink term, with optimal γ and β.

The value of q_s was dependent on the error margin was chosen as the second parameter to find an optimal range for. Thus, the problem has been simplified to only require an optimization in two of the parameters. The results of one of these optimization processes, when the noise level in percentage of the signal was approximately 7%, can be seen in figure 4.6 where it is clearly shown that the value for q_τ should be below 0.00213 and the value for q_s should be below 6×10^8, the exact values being $q_\tau = 0.002$ and $q_s = 3.4 \times 10^8$, while still keeping inside the constraints discussed in Section 4.3.3.

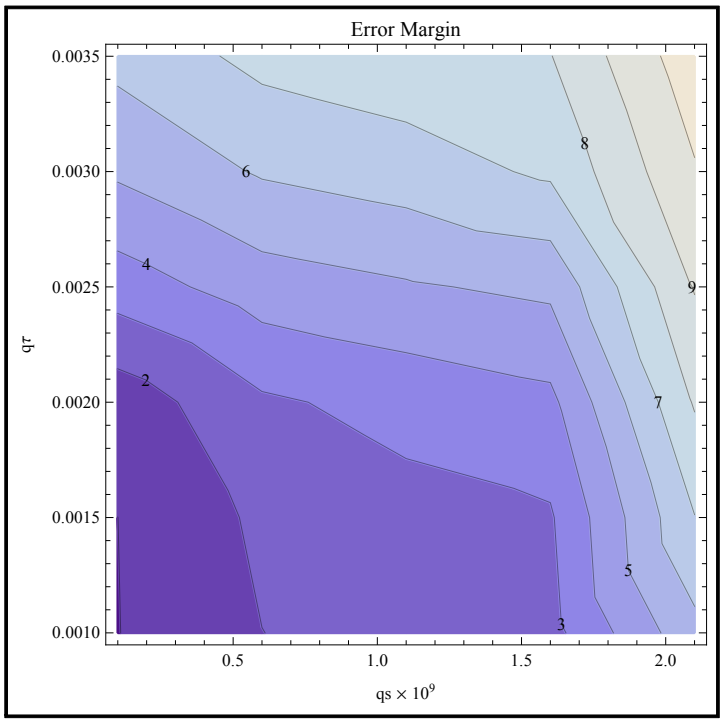

FIGURE 4.6: The Error Margin and ranges for the chosen variables q_τ and q_s

4.5 Conclusion

In this chapter I have presented a partial differential equation which, when using a audio signal's short-time Fourier transform, can diffuse a noisy signal to a point where there is little to no noise left in the signal. A discussion on how and why this method works as well as a method of how to implement this process using a finite difference scheme, FTCS, and the corresponding stability criteria for this method have been shown. This method requires optimization in a large number of parameters in order to get the desired result. The only knowledge required, besides the parameters which are to be optimized, is that of an estimate of the maximum power of the noise in the short-time Fourier transform.

This chapter ultimately presents a method of noise removal using a partial differential equation on the short-time Fourier transform. The method is dynamic and robust but requires a fair amount of processing power for the optimization process.

Chapter 5

Results and Discussion

5.1 A comparison of results

We will compare and contrast the results of filtering an audio clip with different levels of artificially added white noise via the algorithms presented in Chapters 4 (diffusion of the spectrogram model) and 3 (spectral subtraction model) and also in a free audio processing software, Audacity(R), as to compare the results with a method that is used in industry. The window size and overlap used in the diffusion of the spectrogram model and spectral subtraction model was 1000 and 95% respectively.

While the measure of the SNR, in decibels, is given, one cannot say they intuitively understand this unless they know the average power of the section of the signal and the section of the noise it was measured over. Thus, instead of listing all three values I will be listing the SNR and the maximum percentage of the signal which is noise, noise percentage, that is a value that is the maximum noise value measured in a "silent" section compared to the maximum value of the signal in temporal domain. I find this value is easier for anyone not familiar with the decibel SNR scale to understand intuitively.

5.1.1 Results Table

Here I have tabulated the results of the above experiments in table 5.1 for easy comparison, these results are discussed in sections 5.1.3, 5.1.4 and 5.1.5. I will list the results in the form the SNR for each method at each noise percentage level.

The parameters used for each of these methods will be tabulated in table 5.2. Once again, these are the values which are referred to in sections 5.1.3, 5.1.4 and 5.1.5.

TABLE 5.1: Results

Noise percentage level	2%	7%	20%
Noisy SNR	$6.96dB$	$3.006dB$	$1.28dB$
Audacity filtered SNR	$25.47dB$	$21.1dB$	$16.05dB$
Diffusion of spectrogram filter SNR	$42.21dB$	$32.3dB$	$32.03dB$
Spectral subtraction filter SNR	$36.59dB$	$58.7dB$	$23.35dB$

A comparison of the resulting SNR of the different methods at different noise levels.

TABLE 5.2: Parameter values

Noise percentage level	2%	7%	20%
η value	0.0158	0.0514	0.156
Diffusion of spectrogram filter q_s	2×10^9	3.4×10^8	3×10^7
Diffusion of spectrogram filter q_τ	0.002	0.002	0.002
Diffusion of spectrogram filter q_ω	2×10^{-9}	2×10^{-9}	2×10^{-9}
Spectral subtraction filter k	1000	800	500
Spectral subtraction filter q_τ	0.4	0.4	0.4

A comparison of the parameter values used for each of the different methods at the different noise levels.

It might be noted that difference between the coefficients in the equations is large which constrains the time step size, which can be obtained in the analysis of the FTCS scheme in Chapter 4, specifically given these values of q_ω, q_s and q_τ we can calculate the exact value of the time step we can use, which is extremely small, this dramatically slows down computation. An implicit finite-difference scheme would improve the speed, making the time step size less of an obstacle to the computation times.

5.1.2 Noiseless signal

To begin with I will show the signal with no noise in a spectrogram, using the Hamming window, and the plot of the actual signal as to compare to the filtered signals. These can be seen in figures 5.1 and 5.2 respectively.

All processing done via the method presented in Chapter 4 use values of $\gamma = 6$ and $\beta = 250$ where the other parameters are specified in each of the cases.

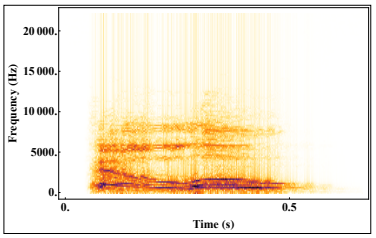

FIGURE 5.1: Spectrogram of the noiseless sound clip.

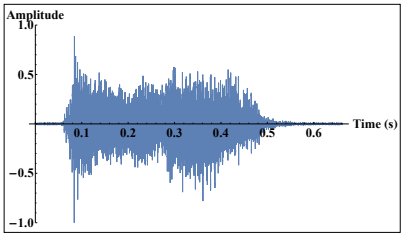

FIGURE 5.2: Temporal plot of the noiseless sound clip.

5.1.3 Comparison at 2% Noise

The first experiment was run over the audio clip containing a SNR of $6.96dB$ and a noise percentage of 2%. This was the lowest level of noise I used and can be seen in figures 5.3 and 5.4. The measured value of η in this method was 0.0158. There are no truly noticeable differences in the solutions by the different models at this noise level except the measured SNR after being filtered however the results are still shown to show that each of the methods for noise removal do work at low noise levels. The SNR was improved to $25.47dB$ by Audacity(R), and the results can be seen in figures 5.5 and 5.6. The spectral subtraction model improved the SNR to $36.59dB$ with $k = 1000$ and $qt = 0.4$. These results can be seen in figures 5.7 and 5.8. The SNR was improved to $42.21dB$ by the diffusion of the spectrogram model, using $q_t = 0.002$, $q_\omega = 2 \times 10^{-9}$ and $q_s = 2 \times 10^9$. These results can be seen in figures 5.9 and 5.10.

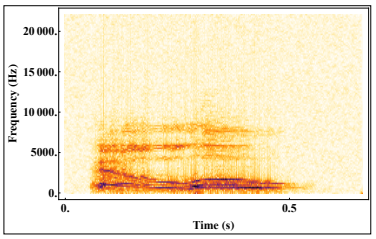

FIGURE 5.3: Spectrogram of the sound clip with a $6.96dB$ SNR.

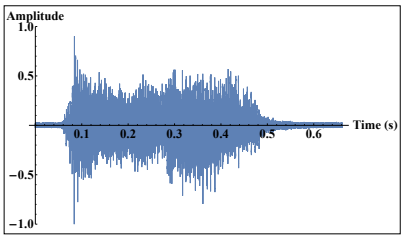

FIGURE 5.4: Temporal plot of the sound clip with a $6.96dB$ SNR.

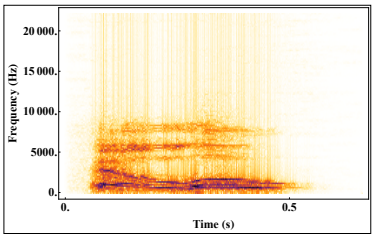

FIGURE 5.5: Spectrogram of the sound clip with a $25.47dB$ SNR after being filtered in Audacity(R).

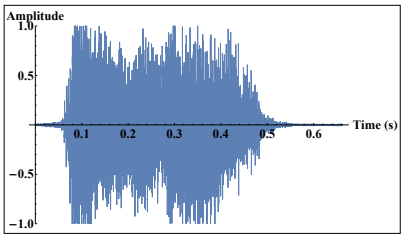

FIGURE 5.6: Temporal plot of the sound clip with a $25.47dB$ SNR after being filtered in Audacity(R).

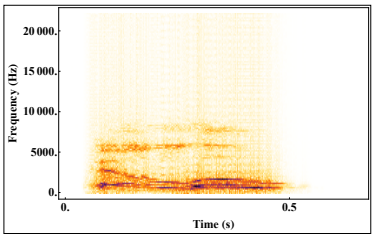

FIGURE 5.7: Spectrogram of the sound clip with a 36.59dB SNR after being filtered by the spectral subtraction model.

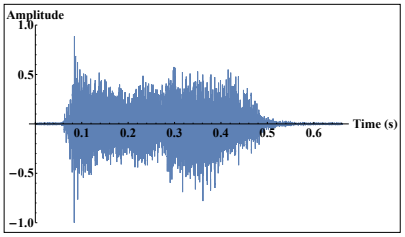

FIGURE 5.8: Temporal plot of the sound clip with a 36.59dB SNR after being filtered by the spectral subtraction model.

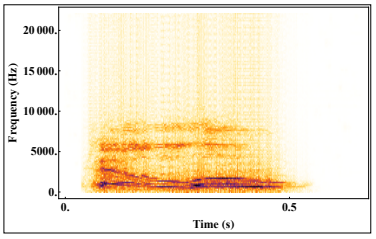

FIGURE 5.9: Spectrogram of the sound clip with a 42.21dB SNR after being filtered by the diffusion of the Ir-Re model.

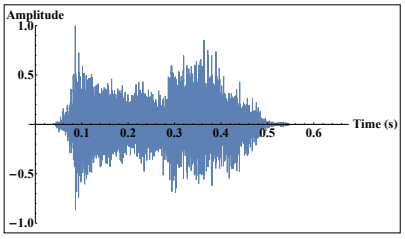

FIGURE 5.10: Temporal plot of the sound clip with a 42.21dB SNR after being filtered by the diffusion of the Ir-Re model.

In figures 5.11, 5.12, 5.13 and 5.14 we can see the spectrograms of the noise itself and the difference between the different filters and the noisy signals. It is clearly evident that each of the filtering processes remove parts of the signal itself, but figure 5.12 clearly shows that less of the signal is removed as described earlier.

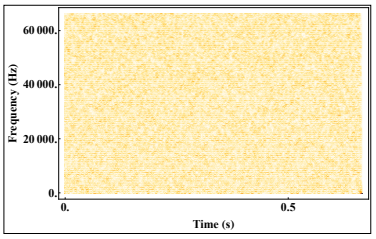

FIGURE 5.11: The spectrogram of the noise of the sound clip which initially had a 2% noise ratio.

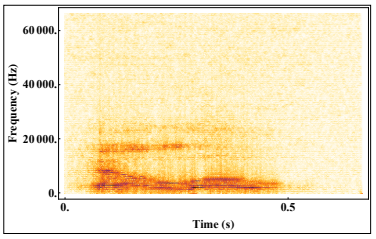

FIGURE 5.12: The difference of the spectrogram of the sound clip, which initially had a 2% noise ratio, after being filtered by the Audacity(R) and the spectrogram of the its respective noisy signal.

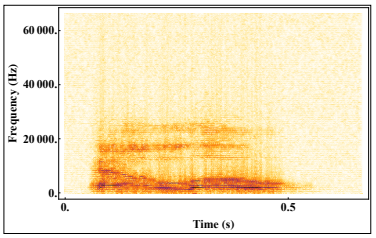

FIGURE 5.13: The difference of the spectrogram of the sound clip, which initially had a 2% noise ratio, after being filtered by the spectral subtraction model and the spectrogram of the its respective noisy signal.

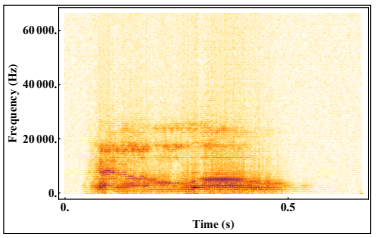

FIGURE 5.14: The difference of the spectrogram of the sound clip, which initially had a 2% noise ratio, after being filtered by the diffusion of the Ir-Re model and the spectrogram of the its respective noisy signal.

5.1.4 Comparison at 7% Noise

The second experiment was run over the audio clip containing a SNR of $3.006dB$ and a noise percentage of 7%. This was the second lowest level of noise I used and can be seen in figures 5.15 and 5.16. The measured value of η in this method was 0.0514. At this noise level there were noticeable in the signals filtered by the different processes. The SNR was improved to $25.47dB$ by Audacity(R), and the results can be seen in figures 5.17 and 5.18. In figure 5.17 we notice that in the "silent" sections, specifically 0-0.06s and 0.54-0.66s, that the noise is still visibly present throughout all frequencies when contrasted to 5.1, in the section with the signal, specifically 0.06-0.54s, the signal in the frequency band between $6000hz$ and $8000hz$ is similar to that of the noiseless spectrogram, with slightly elevated power levels in some frequencies being the only noticeable difference, this is an important point to note since when we compare it to the other two methods we will notice this is an advantage to this method even though the SNR of the method is worse. The spectral subtraction model improved the SNR to $58.7dB$ with $k = 800$ and $qt = 0.4$. These results can be seen in figures 5.19 and 5.20. This method has a far better SNR than Audacity(R)'s method, however when we compare the frequencies' power in the band $6000hz$ to $8000hz$, in the time period 0.06-0.54s, we notice we lose a fair amount the signal when compared to the noiseless signal, we do however have the rest of the frequencies are far more comparable, that is they are closer in value, to the noiseless signal throughout the entire time period. The SNR was improved to $32.3dB$ by the diffusion of the spectrogram model, using $q_t = 0.002$, $q_\omega = 2 \times 10^{-9}$ and $q_s = 3.4 \times 10^8$. These results can be seen in figures 5.21 and 5.22. This method has a mixture of the advantages of both the other methods however it does come with a mixture of both of their disadvantages. It has a good SNR, which is between the others, it keeps a large amount of the power of the frequencies present in the band $6000hz$ to $8000hz$, in the time period 0.06-0.54s, but not as much as the Audacity(R) filtered signal, the power of the frequencies throughout the rest of the spectrogram are close to

the noiseless spectrogram, but not as close as that of the spectral subtraction model's filtered signal. At this noise level it appears that Audacity(R) preserves the signal the best but leaves the highest level of noise behind. The spectral subtraction model gets rid of the most noise but loses some information of the signal when the values of the signal are close to the noise level. The diffusion of the spectrogram model is a mixture of both of these.

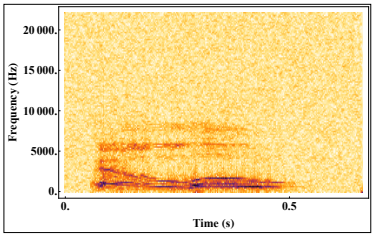

FIGURE 5.15: Spectrogram of the sound clip with a 3.006dB SNR.

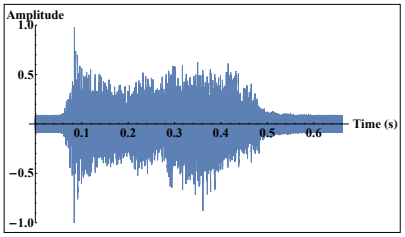

FIGURE 5.16: Temporal plot of the sound clip with a 3.006dB SNR.

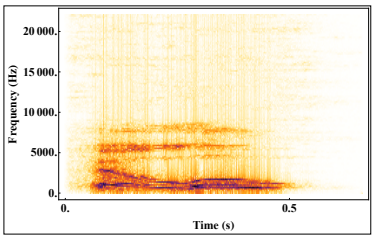

FIGURE 5.17: Spectrogram of the sound clip with a 21.1dB SNR after being filtered in Audacity(R).

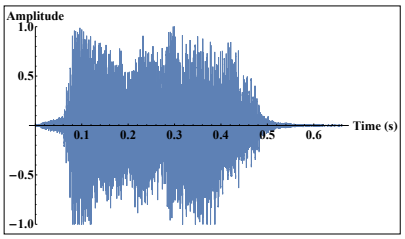

FIGURE 5.18: Temporal plot of the sound clip with a 21.1dB SNR after being filtered in Audacity(R).

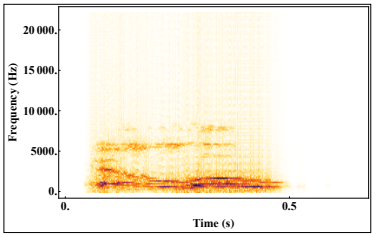

FIGURE 5.19: Spectrogram of the sound clip with a 58.7dB SNR after being filtered by the spectral subtraction model.

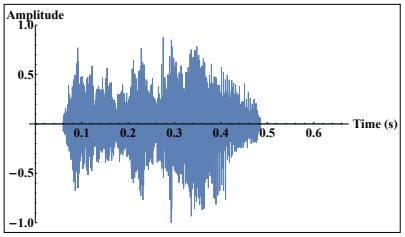

FIGURE 5.20: Temporal plot of the sound clip with a 58.7dB SNR after being filtered by the spectral subtraction model.

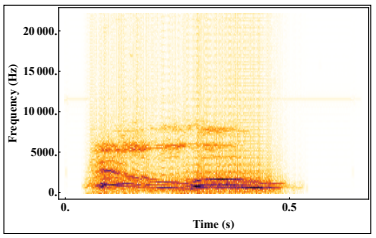

FIGURE 5.21: Spectrogram of the sound clip with a 32.3dB SNR after being filtered by the diffusion of the Ir-Re model.

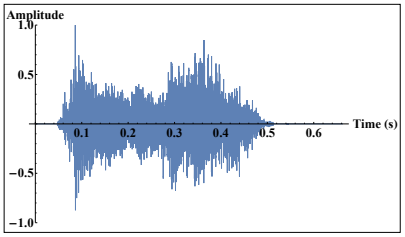

FIGURE 5.22: Temporal plot of the sound clip with a 32.3dB SNR after being filtered by the diffusion of the Ir-Re model.

In figures 5.23, 5.24, 5.25 and 5.26 we can see the spectrograms of the noise itself and the difference between the different filters and the noisy signals. It is clearly evident that each of the filtering processes remove parts of the signal itself, but figure 5.24 clearly shows that less of the signal is removed as described earlier.

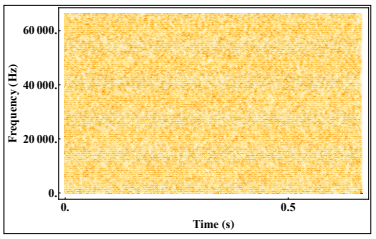

FIGURE 5.23: The spectrogram of the noise of the sound clip which initially had a 7% noise ratio.

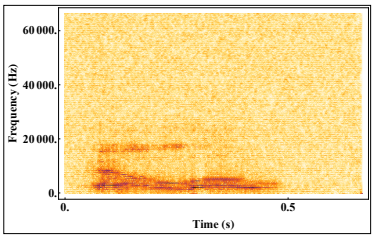

FIGURE 5.24: The difference of the spectrogram of the sound clip, which initially had a 7% noise ratio, after being filtered by the Audacity(R) and the spectrogram of the its respective noisy signal.

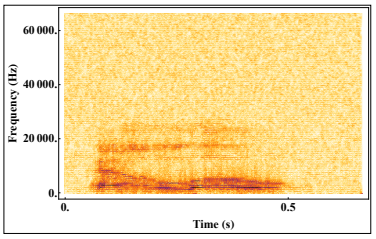

FIGURE 5.25: The difference of the spectrogram of the sound clip, which initially had a 7% noise ratio, after being filtered by the spectral subtraction model and the spectrogram of the its respective noisy signal.

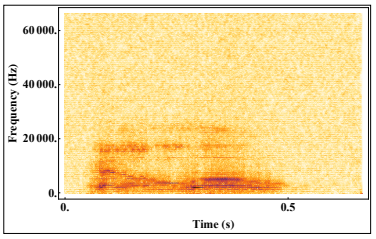

FIGURE 5.26: The difference of the spectrogram of the sound clip, which initially had a 7% noise ratio, after being filtered by the diffusion of the Ir-Re model and the spectrogram of the its respective noisy signal.

5.1.5 Comparison at 20% Noise

The third and final experiment run over the audio clip containing a SNR of 1.28dB and a noise percentage of 20%. This was the highest level of noise I used and can be

seen in figures 5.27 and 5.28. The measured value of η in this method was 0.156. It can be seen in figure 5.27 in particular that the noise in this example almost entirely drowns out the signal in majority of the frequencies. At this noise level the advantages and disadvantages, as discussed in the previous noise level, become quite clear. The measured improvement in the SNR for Audacity(R)'s filter is $16.05dB$ and the results can be seen in figures 5.29 and 5.30. While at this noise level the signal's power in the frequency band $6000hz$ to $8000hz$, in the time period 0.06-0.54s, is lost almost entirely we do however notice the strength of this method in the frequency band $5000hz$ to $6000hz$ in the same time period when we contrast this to the other two methods. We do notice the weakness here is much more visible, that is, the noise throughout the entire spectrogram is still clearly visible. The spectral subtraction model improved the SNR to $23.35dB$ with $k = 1000$ and $qt = 0.4$ and the results can be seen in figures 5.31 and 5.32. We do however notice that there is a scattering of noise throughout the spectrogram which is comparable to that of the actual signal in power. This method is keeping certain sections of the signal in the frequency band $6000hz$ to $8000hz$, in the time period 0.06-0.54s, but loses more in the signal in the frequency band $5000hz$ to $6000hz$ in the same time period when we contrast this to the Audacity(R) filtered signal. Overall the advantage that it removes more noise than the Audacity(R) filter can be seen so this is still consistent. The SNR was improved to $42.21dB$ by the diffusion of the spectrogram model, using $q_t = 0.002$, $q_\omega = 2 \times 10^{-9}$ and $q_s = 3 \times 10^7$. These results can be seen in figures 5.33 and 5.34. Here we notice that this method has the best noise removal for this level of noise, not only by the best SNR of the three methods but it is also clearly visible in its spectrograms when it is compared to the other two and that of the noiseless spectrogram. It has the same disadvantages as before in the fact that it has lost the power level required in the frequency band $6000hz$ to $8000hz$, in the time period 0.06-0.54s, as well as in the frequency band $5000hz$ to $6000hz$ in a similar fashion to that of the spectral subtraction method. When we compare figure 5.34 and 5.30 we can clearly see the difference in the noise removal in the time segments 0-0.06s and 0.54-0.66s.

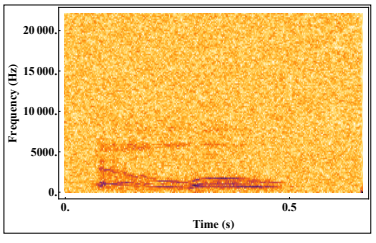

FIGURE 5.27: Spectrogram of the sound clip with a $1.28dB$ SNR.

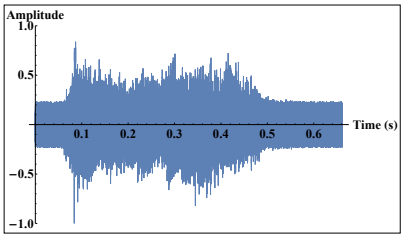

FIGURE 5.28: Temporal plot of the sound clip with a $1.28dB$ SNR.

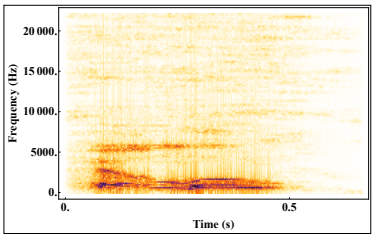

FIGURE 5.29: Spectrogram of the sound clip with a 16.01dB SNR after being filtered in Audacity(R).

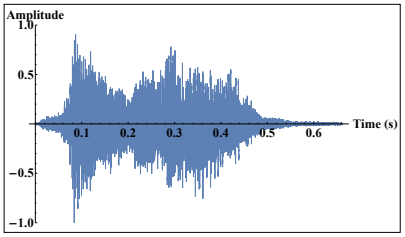

FIGURE 5.30: Temporal plot of the sound clip with a 16.01dB SNR after being filtered in Audacity(R).

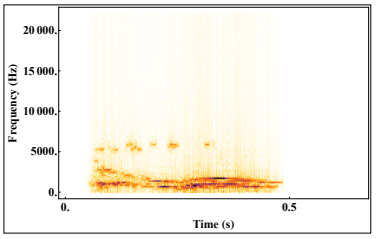

FIGURE 5.31: Spectrogram of the sound clip with a 23.35dB SNR after being filtered by the spectral subtraction model.

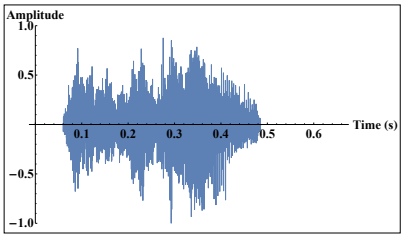

FIGURE 5.32: Temporal plot of the sound clip with a 23.35dB SNR after being filtered by the spectral subtraction model.

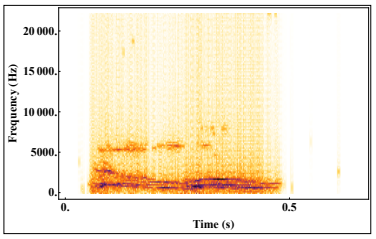

FIGURE 5.33: Spectrogram of the sound clip with a 32.03dB SNR after being filtered by the diffusion of the Ir-Re model.

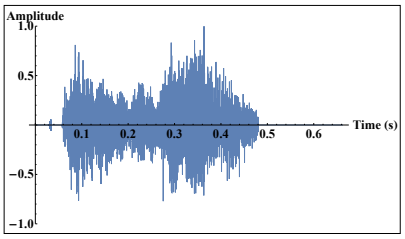

FIGURE 5.34: Temporal plot of the sound clip with a 32.03dB SNR after being filtered by the diffusion of the Ir-Re model.

In figures 5.35, 5.36, 5.37 and 5.38 we can see the spectrograms of the noise itself and the difference between the different filters and the noisy signals. It is clearly evident that each of the filtering processes remove parts of the signal itself, but figure 5.36 clearly shows that less of the signal is removed as described earlier.

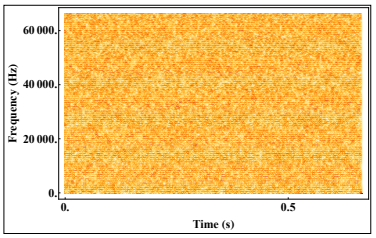

FIGURE 5.35: The spectrogram of the noise of the sound clip which initially had a 20% noise ratio.

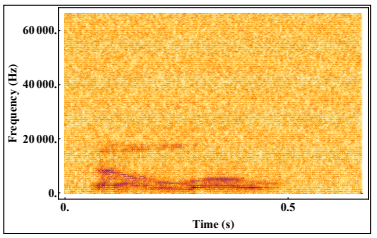

FIGURE 5.36: The difference of the spectrogram of the sound clip, which initially had a 20% noise ratio, after being filtered by the Audacity(R) and the spectrogram of the its respective noisy signal.

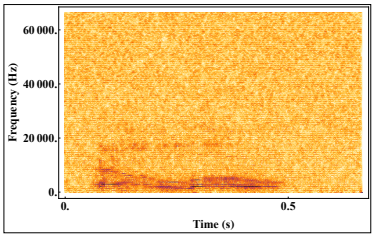

FIGURE 5.37: The difference of the spectrogram of the sound clip, which initially had a 20% noise ratio, after being filtered by the spectral subtraction model and the spectrogram of the its respective noisy signal.

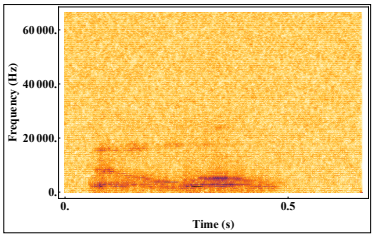

FIGURE 5.38: The difference of the spectrogram of the sound clip, which initially had a 20% noise ratio, after being filtered by the diffusion of the Ir-Re model and the spectrogram of the its respective noisy signal.

5.1.6 Conclusion

In this work I have demonstrated two methods, which use partial differential equations. These two methods obtain results comparable to that of a free audio signal processing software, Audacity(R), with respect to noise removal. They have been presented using a finite difference scheme in order to approximate their solutions. These methods have clear advantages over Audacity(R)'s noise removal method. That is, they both remove more of the noise from a signal contaminated by noise than Audacity(R), however they both lose valuable information of the signal in parts of the signal where the power of the frequencies are comparable to that of the noise's power at the same point. It can be argued that the best of the three methods is the diffusion of the spectrogram model presented in Chapter 4 since it removes more noise than Audacity(R) and preserves majority of the signal. However cases can be made for all three methods and when they are best suited for noise removal. If one wishes to keep a certain amount of noise in order to keep as much of the signal as possible we would find that they would most likely want to use the method implemented by audacity Audacity(R). The method used by Audacity(R) does require that its user gives it a "silent" section of the audio signal where as, while it is preferable to have, the other two methods can work with simple

estimates of the value of η instead of being required to get an exact measurement. If one wishes to have fewer parameters to optimize then it would be recommend that the method of choice, out of the three presented here, would be the spectral subtraction method. The spectral subtraction method also has the advantage that it can be analysed window by window which gives a higher viability of being implemented in real time. If the primary goal is to have a method that removes the most noise, the signal can be processed after its recording, and there is no constraint on processing power for the optimization process required then the recommended method would be the diffusion of the spectrogram model.

Overall, each of these methods has a strength compared to another, and while the numbers do indicate that the diffusion of the spectrogram model does remove the most noise, the choice comes down to which of the methods has the best trade-off. That is, if we want to preserve more of the signal in high noise situations we would most likely use Audacity(R). If we want to remove more noise but have fewer parameters to optimize we would use the spectral subtraction method. Finally, if we want to remove the most noise and have no limitation to processing power for an optimization process we would use the diffusion of the spectrogram model.

Chapter 6

Conclusion and Further Research

6.1 Concluding remarks

Noise is an unwanted phenomenon in audio signal processing. It decreases the intelligibility of speech and often masks important information in an audio signal. In this thesis I have explored various concepts and methods required for the removal of noise in audio signal processing.

In Chapter 2 I presented the various concepts required to proceed with the processing of audio signals and for the approximation of the solution of a partial differential equation given a discontinuous initial condition.

In Chapter 3 I presented a model based solely on a non-linear diffusion equation, that is, there was no sink/source term, designed to filter out the signal in the short-time Fourier transform of a signal in order to obtain a good estimate of the noise and then use spectral subtraction in order to remove the noise from the signal. The diffusion process, with the short-time Fourier transform of a signal used as the initial condition, was carried out via a finite difference scheme, specifically the FTCS scheme, and run to a desired steady state, hence obtaining the estimate of the noise after which the spectral subtraction process was done. This method again required an estimate of the noise profile for the threshold value η required by the coefficient of the diffusion term. This value can be obtained via a "silent" section of the audio or by an estimation by the user of the process. This process was successful in removing noise at extremely low SNR and high SNR, up to 20% of the signal.

In Chapter 4 I presented a model based on a non-linear diffusion equation with a sink/-source term designed to filter out noise in the short-time Fourier transform of a signal and then transform it back to its original form with the noise removed. This model was

presented using a finite difference scheme, specifically the FTCS scheme, as the selected method for approximation of the solutions of the non-linear diffusion equation which used the short-time Fourier transform of a signal as its initial condition and was run to a desired steady state. The only required information for this model is an estimate of the noise profile which can be which yields the value of the threshold value, η, required in the sink/source term, it can be obtained via a "silent" section of the audio signal or can be estimated by the user of the method. The method proved to be successful in removing noise at extremely low SNR and high SNR, up to 20% of the signal.

In Chapter 5 I show the results of the methods presented in Chapters 4 and 3 and compare them to the results of audacity on the same audio corrupted by the same noise. It is important to remember that in this comparison the optimal values of parameters for noise removal were used and not parameters tweaked by a user in order to keep intelligibility of the signal. In this Chapter it can clearly be seen that the method presented in Chapter 4 is the best at removing noise, however Audacity(R), while not removing as much noise, does keep slightly more information of the signal in frequency bands where the signal's power is comparable to that of the power of the noise. That being said, it becomes a trade-off of keeping more signal components with a slight bit more residual noise or remove more of the noise while losing components of the signal in frequency bands where the noise's power is comparable to the signal's. These values can be tweaked by the user of the method so that they might find a good balance of residual noise and preservation of all the signal, all three of the compared methods have the options to do this with correctly chosen parameters.

6.2 Further research

While the results presented in this thesis are the best results I have found there is definitely room for improvement. Different variations of the coefficients of diffusion can be designed to improve the quality of the results. The values of the threshold, η, can be dynamically updated if more knowledge of the noise profile can be obtained with regards to it updating throughout the time of the signal. Such a method would help move toward a real time noise reduction technique.

A different method, which is faster and requires less stability criteria, for the approximation of the solutions of the partial differential equations can be implemented.

6.3 Conclusion

It is possible to remove noise from corrupted audio signals via the application of partial differential equations to the Fourier transforms of the audio signals. The methods presented in this thesis, while only being partially better than that of the method used in Audacity(R), show that partial differential equations are a useful and valuable tool to the field of audio signal processing. In particular, partial differential equations provide a richly dynamic tool that may be engineered to exhibit desirable dynamics and steady states for the application in noise removal of sound in audio band signals. This is a fundamental extension to the application of partial differential equations to that of image processing due to the high variation in audio signals. Thus we have provided valid grounds for further investigation in the applications of partial differential equations in the field of audio signal processing.

Bibliography

[1] H. Moller and C. S. Pedersen, "Hearing at low and infrasonic frequencies," *Noise and health*, vol. 6, no. 23, p. 37, 2004.

[2] S. Geneva, "International organization for standardization; 2005," *International Organization for Standardization ISO Standard*, vol. 4049, pp. 1–4, 2003.

[3] A. N. Salt and T. E. Hullar, "Responses of the ear to low frequency sounds, infrasound and wind turbines," *Hearing research*, vol. 268, no. 1, pp. 12–21, 2010.

[4] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 32, no. 6, pp. 1109–1121, 1984.

[5] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 33, no. 2, pp. 443–445, 1985.

[6] P. J. Wolfe and S. J. Godsill, "Efficient alternatives to the ephraim and malah suppression rule for audio signal enhancement," *EURASIP Journal on Applied Signal Processing*, vol. 2003, pp. 1043–1051, 2003.

[7] S. J. Godsill and P. J. Rayner, "Digital audio restoration. 1998."

[8] S. Godsill and P. Rayner, "Robust noise modelling with application to audio restoration," in *Applications of Signal Processing to Audio and Acoustics, 1995., IEEE ASSP Workshop on*, pp. 143–146, IEEE, 1995.

[9] S. J. Godsill and P. J. Rayner, "Robust treatment of impulsive noise in speech and audio signals," *Lecture Notes-Monograph Series*, pp. 331–342, 1996.

[10] B. P. Carlin, N. G. Polson, and D. S. Stoffer, "A monte carlo approach to non-normal and nonlinear state-space modeling," *Journal of the American Statistical Association*, vol. 87, no. 418, pp. 493–500, 1992.

[11] R. E. McCulloch and R. S. Tsay, "Bayesian inference and prediction for mean and variance shifts in autoregressive time series," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 968–978, 1993.

[12] R. E. McCulloch and R. S. Tsay, "Bayesian analysis of autoregressive time series via the gibbs sampler," *Journal of Time Series Analysis*, vol. 15, no. 2, pp. 235–250, 1994.

[13] J. Ó. Ruanaidh and W. Fitzgerald, "Interpolation of missing samples for audio restoration," *Electronics Letters*, vol. 30, no. 8, pp. 622–623, 1994.

[14] R. Martin, "Speech enhancement based on minimum mean-square error estimation and supergaussian priors," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 5, pp. 845–856, 2005.

[15] T. Lotter and P. Vary, "Speech enhancement by map spectral amplitude estimation using a super-gaussian speech model," *EURASIP journal on applied signal processing*, vol. 2005, pp. 1110–1126, 2005.

[16] I. Andrianakis and P. R. White, "Mmse speech spectral amplitude estimators with chi and gamma speech priors," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 3, pp. III–III, IEEE, 2006.

[17] B. Chen and P. C. Loizou, "A laplacian-based mmse estimator for speech enhancement," *Speech communication*, vol. 49, no. 2, pp. 134–143, 2007.

[18] J. S. Erkelens, R. C. Hendriks, R. Heusdens, and J. Jensen, "Minimum mean-square error estimation of discrete fourier coefficients with generalized gamma priors," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 6, pp. 1741–1752, 2007.

[19] P. J. Wolfe and S. J. Godsill, "Audio signal processing using complex wavelets," in *Audio Engineering Society Convention 114*, Audio Engineering Society, 2003.

[20] G. Yu, E. Bacry, and S. Mallat, "Audio signal denoising with complex wavelets and adaptive block attenuation," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 3, pp. III–869, IEEE, 2007.

[21] L. Breiman, J. Friedman, R. Olshen, and C. Stone, "Classification and regression trees (chapman y hall, eds.)," *Monterey, CA, EE. UU.: Wadsworth International Group*, 1984.

[22] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, pp. 556–562, 2001.

[23] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis," *Neural computation*, vol. 21, no. 3, pp. 793–830, 2009.

[24] M. N. Schmidt, J. Larsen, and F.-T. Hsiao, "Wind noise reduction using non-negative sparse coding," in *Machine Learning for Signal Processing, 2007 IEEE Workshop on*, pp. 431–436, IEEE, 2007.

[25] M. N. Schmidt and J. Larsen, "Reduction of non-stationary noise using a non-negative latent variable decomposition," in *Machine Learning for Signal Processing, 2008. MLSP 2008. IEEE Workshop on*, pp. 486–491, IEEE, 2008.

[26] Z. Duan, G. J. Mysore, and P. Smaragdis, "Speech enhancement by online non-negative spectrogram decomposition in non-stationary noise environments.," in *INTERSPEECH*, 2012.

[27] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Prediction based filtering and smoothing to exploit temporal dependencies in nmf," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 873–877, IEEE, 2013.

[28] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization,"

[29] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 3, pp. 1066–1074, 2007.

[30] G. J. Mysore and P. Smaragdis, "A non-negative approach to semi-supervised separation of speech from noise with the use of temporal dynamics," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 17–20, IEEE, 2011.

[31] K. Han and D. Wang, "An svm based classification approach to speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 4632–4635, IEEE, 2011.

[32] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 7, pp. 1381–1390, 2013.

[33] J. Chen, Y. Wang, and D. Wang, "A feature study for classification-based speech separation at low signal-to-noise ratios," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 12, pp. 1993–2002, 2014.

[34] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 12, pp. 1849–1858, 2014.

[35] R. Jain, R. Kasturi, and B. G. Schunck, *Machine vision*, vol. 5. McGraw-Hill New York, 1995.

[36] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 7, pp. 629–639, 1990.

[37] Y.-L. You, W. Xu, A. Tannenbaum, and M. Kaveh, "Behavioral analysis of anisotropic diffusion in image processing," *Image Processing, IEEE Transactions on*, vol. 5, no. 11, pp. 1539–1553, 1996.

[38] B. Dugnol, C. Fernández, G. Galiano, and J. Velasco, "On pde-based spectrogram image restoration. application to wolf chorus noise reduction and comparison with other algorithms," in *Signal processing for image enhancement and multimedia processing*, pp. 3–12, Springer, 2008.

[39] L. Alvarez, P.-L. Lions, and J.-M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion. ii," *SIAM Journal on numerical analysis*, vol. 29, no. 3, pp. 845–866, 1992.

[40] B. Jacobs and E. Momoniat, "A novel approach to text binarization via a diffusion-based model," *Applied Mathematics and Computation*, vol. 225, pp. 446–460, 2013.

[41] "Audacity(R)." http://audacityteam.org. Accessed: 2017-03-17.

[42] A. Golbabai and M. Javidi, "A numerical solution for non-classical parabolic problem based on chebyshev spectral collocation method," *Applied mathematics and computation*, vol. 190, no. 1, pp. 179–185, 2007.

[43] D.-S. Kim, "On the perceptually irrelevant phase information in sinusoidal representation of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 8, pp. 900–905, 2001.