



Unified Auralization of Room Simulations

Investigation of sampling grids and other parameters

Degree project report in the Department of Architecture and Civil Engineering

Xuanye Liu

Division of Applied Acoustics

CHALMERS UNIVERSITY OF TECHNOLOGY Gothenburg, Sweden 2024 www.chalmers.se

Degree project report 2024

Unified Auralization of Room Simulations

Investigation of sampling grids and other parameters

Xuanye Liu



Department of Architecture and Civil Engineering CHALMERS UNIVERSITY OF TECHNOLOGY Gothenburg, Sweden 2024 Unified Auralization of Room Simulations Investigation of sampling grids and other parameters Xuanye Liu

© Xuanye Liu, 2024.

Supervisor: Jens Ahrens, Department of Architecture and Civil Engineering, Teknisk akustik Examiner: Jens Ahrens, Department of Architecture and Civil Engineering, Teknisk akustik

Degree project report 2024 Department of Architecture and Civil Engineering Chalmers University of Technology SE-412 96 Gothenburg Sweden Telephone +46 31 772 1000

Cover: Create binaura1 signal by PWD-based method(Provided by Jens Ahrens)

Typeset in $\[AT_EX]$ Gothenburg, Sweden 2024 Unified Auralization of Room Simulations Investigation of sampling grids and other parameters Xuanye Liu Department of Architecture and Civil Engineering, Applied Acoustic Chalmers University of Technology

Abstract

With the development of computer science and the increase in computing power, the finite-difference time-domain method has gradually become a viable option for solving wideband, transient problems in acoustics simulation. In this method, the binaural response can be extracted by local sampling of the finite-difference grid using an array of receivers. This thesis focuses on the effects of the grid size, the spacing of grid points, and the distribution of grid points in the case of a volumetric sampling. This thesis also investigates the effect of the change in impulse response length and others. These effects include variations in numerical robustness and the frequency properties of the generated audio signal. In order to achieve this study, this thesis simulates the sound field formed by a single plane wave propagation. By sampling this sound field with a volumetric grid, the plane wave density function is calculated using the plane wave decomposition method and integrated with freefield head-related transfer functions in the spherical harmonics domain to get the binaural impulse response. The results of the valuable work show that the numerical stability of a volumetric grid with uniformly distributed sampling points is greatly affected at high frequencies due to spatial aliasing. In addition, the results also give some valuable conclusions for the selection of the impulse response length and the selection of the number of sampling points.

Keywords: FDTD, Plane wave decomposition, Sampling grid, Spatial audio.

Acknowledgements

I would like to express my deepest appreciation to all those who provided me with the possibility to complete this thesis.

I am very grateful to my supervisor, Jens Ahrens, whose advice helped me clarify my thoughts. He leads my academic research in the direction of spatial audio and gives me a lot of opportunities to think about things far beyond the master's thesis itself.

I would like to thank my parents for their wisdom, unconditional love and support throughout my life; this thesis would not have been possible without them. Thank you.

Xuanye Liu, Gothenburg, March 2024

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

The Finite-Difference Time-Domain
Plane Wave decomposition
Impulse Responce
Head-related transfer function
Fast Fourier Transform
Inverse Fast Fourier Transform

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

n	Index for the spherical harmonic function order
m	Index for the spherical harmonic function degree
d, f, g	Index for position in discrete space
u	Index for position in discrete time
i	Index for grid nodes number

Sets

 \mathcal{Z} Set of Integer

Parameters

С	Speed of sound
λ	Courant number when using FDTD method
C_{1}, C_{2}	Constants when using FDTD method
$a(k, \theta, \phi)$	Plane wave density function at wave number $k, (\theta, \phi)$ direction
$H^l(k,\theta,\phi)$	HRTF of left ear at wave number $k, (\theta, \phi)$ direction
$j_n(kr)$	Spherical Bessel functions of the first kind of n^{th} order
$h_n(kr)$	Spherical Bessel functions of the second kind of n^{th} order
$h_n(kr)$	Spherical Hankel functions of the first kind of n^{th} order
$h_n^2(kr)$	Spherical Hankel functions of the second kind of n^{th} order

$Y_{n}^{m}\left(\theta_{k},\phi_{k}\right)$	Spherical Harmonic at order n, degree m, wave number $k,$ direction (θ_k,ϕ_k)
$b_n(kr)$	Radial function
$\mathbf{a}_{\mathbf{nm}}$	Plane wave density coefficients in frequency domian
$\mathbf{a_{nm}}(\mathbf{t})$	Plane wave density coefficients in time domain(Ambisonics coefficients)
Q	Number of grid nodes
\mathbf{C}	Quadrature Matrix in frequency domain
c _{nm}	Quadrature Matrix in time domain
C_v	Coefficient of variation
N_{IR}	Impulse response length of $\mathbf{c_{nm}}$

Variables

t	Time
r	Vectors in the spherical coordinate system
$p(\mathbf{r},t)$	Pressure of point r at time t
$pert_{d,f,g}^u$	Pressure of $point(d, f, g)$ at time u in discrete space and time
$p^l(k)$	Pressure of left ear at wave number k
$p^l(t)$	Pressure of left ear at time t
р	Pressure matrix in frequency domain
$\mathbf{p}(t)$	Pressure matrix in time domain

Contents

Li	st of	Acronyms	ix
No	omen	clature	xi
Li	st of	Figures	xv
Lis	st of	Tables x	vii
1	Intr	oduction	1
	1.1	Background	1
	1.2	Purpose and Goals	2
2	The	ory	3
	2.1	Wave equation and FDTD method	3
	2.2	Plane wave density function	4
	2.3	Representation sound field in Spherical Harmonics	4
		2.3.1 Spherical Harmonics	4
		2.3.2 Spherical Fourier Transform	5
		2.3.3 Spherical Bessel and Hankel Functions	6
	2.4	2.3.4 Using Spherical Harmonics to express wave	7
	2.4	How to get Plane wave density function by microphone Array	9
3	Met	hods	13
	3.1	General procedure for generating spatial audio signals	13
	3.2	Adjustment of the shape of the sampling grid	14
	3.3	The change in matrix stability expressed through Condition Number	19
	3.4	Compare results generated by different methods	20
4	Res	ults	23
	4.1	Impact on the Condition Number	23
		4.1.1 Influence caused by different grid size	23
		4.1.2 Influence caused by different number of points inside the grid .	24
		4.1.3 Influence caused by different distribution of points inside the	
		grid	25
	4.2	Results generated by different parameters	27
		4.2.1 Difference between ground truth output and the binaural out-	
		put of three Anatomy planes	27

5	Conclusion	n	35
	4.2.5	Influence caused by different grid size	34
	4.2.4	Influence caused by the different impulse response length $\ . \ .$	32
	4.2.3	Influence caused by different grid nodes number \ldots	31
		grid	29
	4.2.2	Influence caused by different distribution of points inside the	

xiv

List of Figures

2.1 2.2	Balloon plot of the spherical harmonics for $n = 0$ (top row) to $n = 2$ (bottom row)	5 9
3.1 3.2	Flowcharts used to show the steps of procedure for generating spatial audio signals	13
	tance between two adjacent points $d = 0.0167 \mathrm{m}$, number of grid points $N = 343$.	15
3.3	Uniform distribution Cubic Array with side length $L = 0.1$ m, number of points on axis $N_{axis} = 7$, number of grid points $N = 343$	16
25	Array with side length $L = 0.1$ m, number of points on axis $N_{axis} = 7$, number of grid points $N = 343$.	19
5.5	An example of the Condition number plot in normalized frequency of a Uniform distribution Cubic Array with side length $L = 0.1$ m, number of points on axis $N_{axis} = 7$, number of grid points $N = 343$.	20
4.1 4.2	Condition number of Different Grid Size Cubic Arrays with number of points on axis $N_{axis} = 7$, number of grid points $N = 343$ Condition number(Normalized Frequency) of Different Grid Size Cu-	23
	bic Arrays with number of points on axis $N_{axis} = 7$, number of grid points $N = 343$.	24
4.3	Condition number of Different number of point on axis Cubic Arrays with grid side length $L = 0.1 \text{ m} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots$	25
4.4	Condition number of Different distribution Cubic Arrays with grid side length $L = 0.1$ m, number of points on axis $N_{axis} = 7$, number of grid points $N = 343$	26
4.5	Average of Condition Number between 1000 Hz and 5000 Hz of Cubic Arrays with grid side length $L = 0.1 \text{ m}$	26
4.6	Human anatomy planes	27
4.7	Deviation of the output depends on the incidence direction (dB)(Transver right ear)	rse, 28
4.8	Deviation of the output depends on the incidence direction (dB)(Frontal, right ear)	28
4.9	Deviation of the output depends on the incidence direction (dB)(Sagittal,	,
	right ear)	29

4.10	Relatively deviation of the output depends on the incidence direction	
	$C_v = -0.5 \dots \dots$	30
4.11	Absolutely deviation of the output depends on the incidence direction	
	$C_v = -0.5 \dots \dots$	30
4.12	Relatively deviation of the output depends on the incidence direction	
	$C_v = 0.5$	31
4.13	Absolutely deviation of the output depends on the incidence direction	
	$C_v = 0.5$	31
4.14	Deviation speed of the output depends on the coefficient of variation	31
4.15	Deviation of the output depends on the number of grid points (dB)	
1.10	with grid side length $L = 0.1 \text{ m}$	32
4.16	Deviation of the output depends on the impulse length Grid info	-
1.10	Side length $L = 0.1$ m Number of points on axis $N_{\text{exis}} = 7$ Number	
	of grid points $N = 343$	33
417	Deviation speed of the output depends on the impulse length Grid	00
1.11	info: Side length $L = 0.1 \mathrm{m}$ Number of points on axis $N = -7$	
	Number of grid points $N = 343$	33
1 18	Number of grid points $N = 545$	00
4.10	Number of points on axis $N_{-} = 7$ Number of grid points $N = 343$	24
4 10	Number of points on axis $N_{axis} = 7$, Number of grid points $N = 545$.	94
4.19	beviation speed of the output depends on the grid size (db). Grid	
	mio: Number of points on axis $N_{axis} = i$, Number of grid points $N_{axis} = 242$	<u>າ</u> ∕
	IV = 545.	54

List of Tables

4.1	Computaion	time t for	$N_{IR} = 2$	2048, 1024,	8				32
-----	------------	--------------	--------------	-------------	---	--	--	--	----

1

Introduction

1.1 Background

Acoustic simulation has a wide range of uses, such as fixing acoustic defects in a room or for virtual reality use. There are many ways to perform acoustic simulations. Diffused field equations (Sabine/Eyring), Ray/particle tracing, and Wave simulation are more commonly used. The diffuse field equations give only a fairly rough result for the sound field. Ray/particle tracing is more accurate than diffuse field equations, but it is a high-frequency approximation of a sound field. Besides, diffraction is not easy to simulate using this method. Wave simulation is more accurate than the previous two, as it takes wave diffraction into account. It validates in the entire frequency range, but its computational cost is enormous, which means it usually takes a lot of time. With the development in processor performance and parallel computing, it has become feasible to use wave simulation.

The Finite Difference Time Domain (FDTD) algorithm is a useful tool for our wave simulations[1]. Since the FDTD method produces results for each time step in the computation process, it enables the real-time auralization of the sound field. In the actual simulation, we need to specify the shape of the house and the location of some of the main reflective surfaces, and for auralization, we should specify the location of the listener. We have many different strategies to achieve auralization. One can choose an audio signal to be used as the source for the sound field simulation so that what is obtained is the result of this audio signal after it has propagated in the sound field. Another alternative is to measure and calculate the binaural impulse response and convolve it at a later stage with the original audio signal, which was measured in the free field.

A human head model can be placed directly at the listener's position during the modeling phase to obtain this binaural response. One detail to consider when embedding a geometric model of a listener in the grid[2][3][4], however, is the amount of detail required to model a mannequin compared to the walls or furniture of a house. A large amount of detail can lead to a significant increase in computing power. Another option to obtain this binaural response is to extract the plane-wave component of the sound field and spatially integrate it with the pre-measured HRTF[5], which is also the approach used in this thesis.

1.2 Purpose and Goals

The main objective of this study is to investigate the effects of different sampling grid sizes and other parameters on the generation of spatial signals using plane wave decomposition. This study also aims to evaluate how parameters change influence the accuracy, efficiency, and overall performance of binaural signal generation. This study aims to discover potential optimization strategies by systematically investigating these parameters.

To achieve the purposes mentioned above, this paper sets forth the following specific goals:

- 1. Systematically vary the sampling grid size and assess its direct impact on the condition number and quality of the generated spatial signals. This involves a comparative analysis of different grid sizes to identify optimal configurations.
- 2. Investigate the role of parameters such as impulse response length and number of grid points. This goal focuses on understanding how these parameters can be tuned to improve signal generation outcomes.
- 3. Propose potential optimization strategies based on the findings of the study.
- 4. Summarize the general principles of signal generation using plane wave decomposition methods. This is to understand the details of signal generation.

2

Theory

In this section, this thesis briefly describes the process of auralization in FDTD using the plane wave decomposition method and summarizes the required formulas. Also, since we are sampling the sound field using a volumetric grid, which is a special case of a spherical grid, it is convenient to use spherical harmonics to express the sampled data and in this way approximate a plane wave density function. Therefore, the knowledge about spherical harmonics will be mainly introduced in this section. The main theoretical basis of this thesis comes from the papers of Jonathan Sheaffer et al.[6] and the book written by Boaz Rafaely[7].

2.1 Wave equation and FDTD method

Before proceeding with a sound field simulation, we should first understand how to properly describe a sound field. In a homogeneous space, a sound field can be formulated by the acoustic wave equation[6]:

$$\nabla^2 p(\mathbf{r}, t) - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} p(\mathbf{r}, t) = 0$$
(2.1)

where ∇ is the Laplace operator, $p(\mathbf{r}, t)$ is the sound pressure and c is the speed of sound. The wave equation can be expressed in a Cartesian coordinate or in a spherical coordinate. When expressed in the spherical coordinate, what \mathbf{r} in the equation represents $\mathbf{r} = (r, \theta, \phi)$, where r denotes radial distance, θ denotes elevation and ϕ is azimuth.

We usually utilize the FDTD method for the simulation of sound propagation. To utilize the FDTD method, we need to mesh the space and specify the time step for the solution. Continuous Cartesian coordinate system is discretized as $(x, y, z, t) \rightarrow$ [dX, fX, gX, uT], u here is index position in discrete time and [d, f, g] are index positions in discrete space. Accordingly, the wave equation can be formulated under the FDTD method as[6]:

$$\begin{bmatrix} \delta_t^2 - \lambda^2 \left(\delta_x^2 + \delta_y^2 + \delta_z^2 + C_1 \delta_x^2 \delta_y^2 + C_1 \delta_x^2 \delta_z^2 + C_1 \delta_y^2 \delta_z^2 + C_2 \delta_x^2 \delta_y^2 \delta_z^2 \right) \end{bmatrix} p \Big|_{d,f,g}^u = 0$$
(2.2)

where δ_D^2 denotes a second-order finite difference operator over the dimension D, λ here is the Courant number and constants C_1 and C_2 , are chosen according to the desired finite difference scheme.

In the above, we have briefly described the basic methods for implementing sound field simulation.

2.2 Plane wave density function

The solution of Eq. (2.1) is the distribution of pressure in space over time; after the wave has traveled through space for a sufficiently long period of time, the sound field can be expressed as the continuum of plane waves. Therefore, we can introduce the Plane wave density function $a(k, \theta, \phi)$ to represent the sound field, where k is the wavenumber. However, the sound field obtained here does not consider the human body's influence on the sound, so we can use the Head-related transfer function (HRTF) to post-process this sound field to introduce the influence of the human body. With the Plane wave density function known, the binaural response considering the HRTF can be obtained by the following equation[6]:

$$p^{l}(k) = \int_{(\theta,\phi)\in S^{2}} a(k,\theta,\phi) H^{l}(k,\theta,\phi) \mathrm{d}\theta \mathrm{d}\phi$$
(2.3)

where $H^{l}(k, \theta, \phi)$ is an HRTF catalog for the left-ear, $p^{l}(k)$ is the sound pressure at the left ear. The essence of this formula is a kind of beamforming synthesis. The question that remains at this point is how to get the Plane wave density function. However, before describing how to get the plane wave density function, I will go into more detail about spherical harmonics, as they are an important tool for us to get the plane wave density function through the microphone array.

2.3 Representation sound field in Spherical Harmonics

2.3.1 Spherical Harmonics

The nature of the spherical harmonics is a set of basis functions, or more precisely, the spherical harmonic functions are a set of complete orthogonal functions, and hence, it is a set of orthogonal basis. Basis function is not an unfamiliar concept; the sine and cosine functions (equivalently, the complex exponential functions) form a set of basis known as Fourier basis functions in the function space. Using Fourier basis functions, we can transform a signal between the time and frequency domains via the Fourier transform, or alternatively, we can use a linear combination of Fourier basis functions to represent any other function. Fourier basis functions serve a very important role in signal processing, and while Fourier basis functions are well suited for working with time-series data and frequency analysis, they do not directly provide a mechanism for working with the three-dimensional spatial distribution of sound. The Fourier transform is primarily concerned with the frequency content of a signal, not its spatial directionality or distribution. Therefore, the spherical harmonic function has a unique advantage when dealing with spatial audio signals. The mathematical expression for the spherical harmonic function is as follows[7]:

$$Y_{n}^{m}(\theta,\phi) \equiv \sqrt{\frac{2n+1}{4\pi} \frac{(n-m)!}{(n+m)!}} P_{n}^{m}(\cos\theta)e^{im\phi},$$
(2.4)

where ! represents the factorial function, $P_n^m(\cos \theta)$ are the associated Legendre functions, $m \in Z$ is an integer denoting the function degree and $n \in Z$ is a natural number denoting the function order. The following figure shows the spherical harmonic functions of orders 0 to 2.



Figure 2.1: Balloon plot of the spherical harmonics for n = 0 (top row) to n = 2 (bottom row)

We introduce a new concept here, the Associated Legendre polynomials. Associated Legendre Polynomials are a special class of functions of particular importance in mathematical physics that are derived from Legendre Polynomials, which were discovered when dealing with the effect of ellipsoids on gravitational potentials.

The Associated Legendre polynomials are derived by differentiating the Legendre polynomials. Its mathematical expression is as follows[7]:

$$P_n^m(x) = (-1)^m \left(1 - x^2\right)^{m/2} \frac{d^m}{dx^m} P_n(x), \quad x \in [-1, 1],$$
(2.5)

where $P_n(x)$ denotes Legendre polynomials, which can be obtained directly by using the following differential formula[7]:

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} \left(x^2 - 1\right)^n,$$
(2.6)

The Legendre polynomials form a complete and orthogonal set of basis functions over the line section $x \in [-1, 1]$.

2.3.2 Spherical Fourier Transform

As mentioned in the previous section, any signal can be expressed in terms of Fourier basis functions, an operation also known as the Fourier transform. The Spherical Fourier Transform is the analog of the Fourier transform on the sphere, which means the Spherical Fourier Transform decomposes a function defined on a sphere into a sum of spherical harmonics, each of which is multiplied by a coefficient that represents the amplitude of the function at a particular frequency. It is important to note here that the spherical harmonic function is actually the set of all square-integrable functions on the unit sphere, which can also be expressed as $Y_n^m(\theta, \phi)$ form a basis in Hilbert space $L_2(S^2)$. Spherical Fourier Transform works in the following way: given a function satisfying $f(\theta, \phi) \in L_2(S^2)$, it can be represented by a set of weighted spherical harmonic functions as[7]:

$$f(\theta,\phi) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} f_{nm} Y_n^m(\theta,\phi), \qquad (2.7)$$

where f_{nm} are the weights function. Similar to the Fourier transform, the spherical Fourier transform has its inverse form, and this inverse form can be used to represent this weighting function[7]:

$$f_{nm} = \int_0^{2\pi} \int_0^{\pi} f(\theta, \phi) \left[Y_n^m(\theta, \phi) \right]^* \sin \theta d\theta d\phi.$$
(2.8)

2.3.3 Spherical Bessel and Hankel Functions

One piece of the mathematical puzzle that is still missing before we use spherical harmonics to represent waves, is Spherical Bessel and Hankel Functions. It is actually a part of the solution of the wave equation in the spherical coordinate system. After solving the wave equation in the spherical coordinate system using the separated variable method, the following solution can be obtained [7]:

$$p(\mathbf{r},t) = j_n(kr)Y_n^m(\theta,\phi)e^{i\omega t}$$
(2.9)

or

$$p(\mathbf{r},t) = h_n(kr)Y_n^m(\theta,\phi)e^{i\omega t},$$
(2.10)

where $j_n(kr)$ denotes spherical Bessel functions of the first kind, $h_n(kr)$ denotes spherical Hankel functions of the first kind. The solutions of these two forms(Eq. (2.9) and Eq. (2.10)) are equivalent.

The spherical Bessel function of the first kind, $j_n(x)$, and of the second kind, $y_n(x)$ can be written as [8]:

$$j_n(k) = (-1)^n x^n \left(\frac{1}{x} \frac{d}{dx}\right)^n \frac{\sin(x)}{x}$$
(2.11)

and

$$y_n(x) = -(-1)^n x^n \left(\frac{1}{x}\frac{d}{dx}\right)^n \frac{\cos(x)}{x}.$$
(2.12)

Similarly, it is possible to write out the spherical Hankel functions of the first kind, $h_n(x)$, and the second kind $h_n^{(2)}(x)$,

$$h_n(x) = -i(-1)^n x^n \left(\frac{1}{x}\frac{d}{dx}\right)^n \frac{e^{ix}}{x}$$
(2.13)

and

$$h_n^{(2)}(x) = i(-1)^n x^n \left(\frac{1}{x}\frac{d}{dx}\right)^n \frac{e^{-ix}}{x}.$$
(2.14)

Spherical Bessel function and Spherical Hankel function are related to each other by

$$h_n(x) = j_n(x) + iy_n(x)$$
 (2.15)

$$h_n^{(2)}(x) = j_n(x) - iy_n(x).$$
(2.16)

Spherical Bessel and Hankel functions have two important properties: Orthogonality and Recurrence Relations. Orthogonality, which means spherical Bessel and Hankel functions are orthogonal with respect to their order under appropriate weighting. Recurrence Relations, which means these functions satisfy specific recurrence relations, which can be used for efficient computation of function values for different orders.

2.3.4 Using Spherical Harmonics to express wave

Consider a plane wave of unit amplitude of single frequency propagating from the direction (θ_k, ϕ_k) , with a wave vector $\tilde{\mathbf{k}} = -\mathbf{k} = (k, \theta_k, \phi_k)$. In a spherical coordinate, this plane wave can be written as[7]:

$$p(k, r, \theta, \phi) = e^{-i\mathbf{k}\cdot\mathbf{r}} = e^{i\mathbf{\tilde{k}}\cdot\mathbf{r}}, \qquad (2.17)$$

where $\mathbf{r} = (r, \theta, \phi)$, with Spherical Fourier Transform, this wave can be written as the sum of the spherical harmonics and the spherical Bessel function[7]:

$$p(k, r, \theta, \phi) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} 4\pi i^{n} j_{n}(kr) \left[Y_{n}^{m}(\theta_{k}, \phi_{k})\right]^{*} Y_{n}^{m}(\theta, \phi).$$
(2.18)

In Eq. (2.18), a set of spherical harmonic functions of infinite order is used to express this plane wave, which is clearly impossible in practice. A set of finite-order spherical harmonics can actually be used to model a plane wave propagating in space[7]:

$$p(k,r,\theta,\phi) \approx \sum_{n=0}^{N} \sum_{m=-n}^{n} 4\pi i^{n} j_{n}(kr) \left[Y_{n}^{m}\left(\theta_{k},\phi_{k}\right)\right]^{*} Y_{n}^{m}(\theta,\phi), \qquad (2.19)$$

which also introduces truncation errors.

It is now possible to express a single plane wave propagating in space using spherical harmonics, but as presented in Section 2.2, after the wave has traveled through space for a sufficiently long period of time, the sound field can be expressed as the continuum of plane waves. Therefore, we can introduce the Plane wave density function $a(k, \theta_k, \phi_k)$ to represent the sound field. With the plane wave density

function, several spherical harmonics can be combined to express the wave formed by combining multiple plane waves[7].

$$p(k, r, \theta, \phi) = \int_0^{2\pi} \int_0^{\pi} a(k, \theta_k, \phi_k) e^{i\tilde{\mathbf{k}}\cdot\mathbf{r}} \sin\theta_k d\theta_k d\phi_k$$

$$= \sum_{n=0}^{\infty} \sum_{m=-n}^n 4\pi i^n a_{nm}(k) j_n(kr) Y_n^m(\theta, \phi)$$
(2.20)

Just as it is possible to express a single plane wave by a set of finite-order spherical harmonics, it is also possible to express the case of multiple plane waves synthesized by a set of finite-order spherical harmonics. The formula is as follows[7]:

$$p(k, r, \theta, \phi) \approx \sum_{n=0}^{N} \sum_{m=-n}^{n} 4\pi i^{n} a_{nm}(k) j_{n}(kr) Y_{n}^{m}(\theta, \phi)$$

= $b_{n}(kr) \sum_{n=0}^{N} \sum_{m=-n}^{n} a_{nm}(k) Y_{n}^{m}(\theta, \phi),$ (2.21)

where radial function is given by $b_n(kr) = 4\pi i^n j_n(kr)$. It is important to note that the plane wave composed by the above equation does not take into account the scattering of the sound wave after encountering an object, i.e., the scattering of the sound wave after encountering a human body is not taken into account in this case of ours. However, we will use HRTF in post-processing to add the effect of the human body to our binaural response, so this absence is reasonable.

As mentioned earlier, finite-order harmonics can simulate plane waves propagating in space with truncation errors, which is mainly caused by the high-frequency decay of the Bessel term in the radial function $b_n(kr)$. A figure (Figure 2.2) from Boaz Rafaely's book [7] is cited here to illustrate this point.



Figure 2.2: Magnitude of the radial function $b_n(kr) = 4\pi i^n j_n(kr)$, for kr = 8, 16

2.4 How to get Plane wave density function by microphone Array

In the previous section, we briefly described the plane wave density function and how to use it to synthesize the binaural response with the HRTF, but we still don't know how to get it. By looking at Eq. (2.21), we can see that we can perform plane wave synthesis given the plane wave density function and spherical harmonics. Knowing that the spherical harmonics are a set of functions that can be pre-calculated if the distribution of pressure in space is measured by a microphone array, the plane wave density function can also be calculated accordingly. This operation is known as plane wave decomposition (PWD). Imagine that Q recievers are arbitrarily placed in a spherical coordinate system, where the positions of these microphones can be represented as $(r_i, \theta_i, \phi_i), i = 1, ..., Q, i \in \mathbb{Z}$, and the pressure at each receiver is captured and transformed to the frequency domain using Fast Fourier Transform(FFT), resulting in a frequencydependent vector[6],

$$\mathbf{p} = [p_1(k), p_2(k), \cdots, p_Q(k)]^T, \qquad (2.22)$$

where $[]^T$ denotes transpose of matrix. According to the previous section, combining a set of spherical harmonics can represent an arbitrary function. In particular, when we consider that the sound in the sound field consists of a series of plane waves, according to Eq. (2.21), this formula can be rewritten in this case as[6]:

$$p(k, r_i, \theta_i, \phi_i) \approx \sum_{n=0}^{N} \sum_{m=-n}^{n} a_{nm}(k) b_n(kr_i) Y_n^m(\theta_i, \phi_i).$$

$$(2.23)$$

Eq. (2.23) can be further written in the form of a matrix as [6]:

$$\mathbf{p} = \mathbf{B}\mathbf{a_{nm}}.\tag{2.24}$$

Since the N th order spherical harmonic has 2N + 1 plane wave coefficients corresponding to it, the first N th order spherical harmonic has a total of $(N + 1)^2$ plane wave density coefficients corresponding to it, $\mathbf{a_{nm}}$ is a $(N + 1)^2$ vector representing the plane wave coefficients of the sound field, which can be written as[6]:

$$\mathbf{a_{nm}} = \left[a_{00}, a_{1(-1)}, a_{10}, a_{11}, \dots, a_{NN}\right]^T.$$
 (2.25)

Similarly, combined with the fact that we have Q sampling points the **B** is a $Q \times (N+1)^2$ matrix, which can be written as[6]:

$$\mathbf{B}^{I} = \begin{bmatrix} b_{0}(kr_{1}) Y_{0}^{0}(\theta_{1}, \phi_{1}) & \cdots & b_{0}(kr_{Q}) Y_{0}^{0}(\theta_{Q}, \phi_{Q}) \\ b_{1}(kr_{1}) Y_{1}^{-1}(\theta_{1}, \phi_{1}) & \cdots & b_{1}(kr_{Q}) Y_{1}^{-1}(\theta_{Q}, \phi_{Q}) \\ \vdots & \ddots & \vdots \\ b_{N}(kr_{1}) Y_{N}^{N}(\theta_{1}, \phi_{1}) & \cdots & b_{N}(kr_{Q}) Y_{N}^{N}(\theta_{Q}, \phi_{Q}) \end{bmatrix} .$$

$$(2.26)$$

From Eq. (2.24) we know that the plane wave density coefficient matrix can be obtained by simply multiplying the inverse of the **B** matrix over the left of **p** matrix. However, in this thesis, the relationship between the number of sampling points Q and the order of the spherical harmonic N is as follows:

$$Q > (N+1)^2 \,. \tag{2.27}$$

In this case, we can only look for the inverse of the **B** matrix in the least squares sense, thus there is [6]:

$$\mathbf{a}_{\rm nm} \approx \mathbf{C} \mathbf{p} = \mathbf{B}^{\dagger} \mathbf{p},\tag{2.28}$$

where **C** denotes the quadrature coefficients matrix, which transforms sound pressure into plane wave density coefficients matrix. The matrix \mathbf{B}^{\dagger} is the Moore-Penrose pseudoinverse of **B**, written as[6]:

$$\mathbf{B}^{\dagger} = \left(\mathbf{B}^{H}\mathbf{B}\right)^{-1}\mathbf{B}^{H},\tag{2.29}$$

where $[]^{H}$ denotes the conjugate transpose. It is worth noting that all equations in this subsection are in the frequency domain. However, this thesis requires the generation of signals in the time domain, so it is necessary to convert **C** into impulse responses of the corresponding order of spherical harmonics by Inverse Fast Fourier Transform(IFFT) and denoted by $\mathbf{c_{nm}}$. In the time domain, Eq. (2.28) can be written as:

$$\mathbf{a}_{\rm nm}(t) \approx \mathbf{c}_{\rm nm} \mathbf{p}(t). \tag{2.30}$$

Where the \mathbf{p} denotes the time-domain signal received by each receiver, the $\mathbf{a}_{nm}(\mathbf{t})$ denotes the ambisonic coefficient matrix.

In addition to this, accordingly, Eq. (2.3) can be written in discretized form as [6]:

$$p^{l}(t) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \tilde{a}_{nm}^{*}(t) * H_{nm}^{l}(t), \qquad (2.31)$$

where

$$\tilde{a}_{nm}(t) = (-1)^m \left[a_{n(-m)}(t) \right]^*.$$
(2.32)

 $a_{n(-m)}(t)$ and $H_{nm}^{l}(t)$ are the ambisonic coefficients and HRTF of the left ear represented in the spherical harmonic domain, respectively. * denotes the convolution operator. The $p^{l}(t)$ is the impulse response at the left ear. An audible binaural signal can be obtained by introducing raw mono signals at a suitable workflow location, which is described in the Methods part.

We now have the complete tool for calculating plane wave density coefficients and a basic understanding of the theoretical foundations needed for this thesis.

2. Theory

Methods

The entire content of this thesis is carried out in Matlab. In this section, I will describe the implementation of my thesis, the various plotting implementations, comparison methods, plotting methods in detail, etc.

3.1 General procedure for generating spatial audio signals

To perform the auralization in this thesis, I mainly use the auralization toolkit written by Jens Ahrens [9]. In this section, I'll explain the workflow in general. There are two methods applied in this paper; one of the simpler methods is to convolve the sound signal directly with the HRTF of the direction from which we preset the sound signal to come from. Through this method, we can generate a signal which is considered as the reference result or so-called ground truth.

Another method is to use the approach expressed in the theory section to process and obtain the signal, which will be used to compare with the reference signal. The rough workflow is shown in the following flowchart(Figure.3.1):



Figure 3.1: Flowcharts used to show the steps of procedure for generating spatial audio signals

The first step in this method is to generate a spatial grid to determine the sampling nodes. In fact, the shape of the sampling grid and the grid nodes' distribution affects the algorithm's stability and the quality of the generated results, but this will be discussed in detail in the subsequent sections.

The second step of this method is to generate the impulse response data at each sampling grid. In real applications, this data should be simulated using the FDTD method. Due to the complex modeling and the large number of computations involved in direct sound field simulation using the FDTD method, apart from this, the primary purpose of this thesis is to investigate the effect of different sampling grids and parameters not related to the FDTD method on the computational speed and the quality of the results; as a result, this thesis will not directly use FDTD method. Without using the FDTD method, this thesis determines the impulse response data at each grid node by simulating a plane wave coming from a specific direction directly.

The third step in this thesis is to compute the quadrature matrix, as the quadrature matrix is only related to the sample grid itself, which can be known from Eq. (2.26); this step can actually be done in parallel with the second step.

The fourth step is calculating ambisonic coefficients, which can be obtained by using the IIFT on plane wave density coefficients. The wave density coefficients can be obtained by multiplying the Moore-Penrose pseudoinverse of the quadrature matrix on the left side of the matrix of the sampled signals (in this case, a set of impulse responses) at each grid node.

The fifth part introduces an audio signal and converts it into an ambisonic signal. For auralization, this thesis introduces an audio sound signal as the original audio. The original signal needs to be convolved with ambisonic coefficients to produce an ambisonic signal.

In step 6, we introduce the HRTF, as described in Eq. (2.31). This step involves re-expressing the HRTF in terms of spherical harmonics; the HRTF measurement is achieved through a process where a loudspeaker is placed in various positions on a sphere, and sound is played, followed by measuring the signal received by the listener's binaural at the center of the sphere. The impulse response is then calculated based on this data. And these points where the speakers are placed are predefined, which follows Lebedev grid. Thus, the weights of the HRTFs obtained by sampling in this way are known conditions when transformed into the spherical harmonic domain.

Above is the general process for generating spatial audio signals. Although generating audio signals using the above process is not complicated, there are still many details worth discussing.

3.2 Adjustment of the shape of the sampling grid

The grid-choosing scheme influences the calculations' stability and the results' quality. The following figure demonstrates one of the main grid types used in this thesis, the cubic grid array(Volumetric grid). The default grid type used in this paper is a uniformly distributed cubic grid.



Figure 3.2: Uniform distribution Cubic Array with side length L = 0.1 m, distance between two adjacent points d = 0.0167 m, number of grid points N = 343.

The different colors of the points in the figure are to show the distance of the different grid points from the origin.

The code to create a uniformly distributed cubic grid is relatively simple:

```
1
2
3
4
5
```

```
L = 0.1; %side length
dx = 0.1/6; % distance between two adjacent points
spatial_axis = -L/2 : dx : L/2; % m
[x, y, z] = meshgrid(spatial_axis, spatial_axis,...
spatial_axis);
```

During subsequent tests, I tried to move the distribution of nodes towards the surface of the cube based on the uniformly distributed cubic array.



Figure 3.3: Uniform distribution Cubic Array with side length L = 0.1 m, number of points on axis $N_{axis} = 7$, number of grid points N = 343.

As you can clearly see in this figure, the gird nodes are distributed closer to the surface. To accomplish this, I wrote a function:

```
1
       function [x, y, z] = creat_grid(Number_of_points_on_the_axis, L ,
\mathbf{2}
           coefficient_of_variation)
3
       switch nargin
4
\mathbf{5}
           case 3
6
7
                   Number_of_gaps_on_half_axis = (Number_of_points_on_the_axis-1)/2;
8
9
                   dx_mean = L/(Number_of_points_on_the_axis-1); % leads to 343
10
                       points
^{11}
                   N_desired = 8; % for prediciting requiered no. of points
12
13
                   % sampling intervall: inverse of aliasing frequency x c (N = omega
14
                       /c r)
                   dx_max = (2*pi * sqrt((L/2)^2 + (L/2)^2) + (L/2)^2) / N_desired;
15
                   dx_min = dx_max ./ 100;% make sure distance is not zero
16
17
               if coefficient_of_variation == 0
18
19
                   dx = L / (Number_of_points_on_the_axis-1);
20
21
                   spatial_axis = -L/2 : dx : L/2; % m
^{22}
                   [x, y, z] = meshgrid(spatial_axis, spatial_axis, spatial_axis);
^{23}
24
                   % reshape
25
                   x = x(:).;
26
                   y = y(:).';
27
                   z = z(:).':
28
               else
29
30
                   %coefficient_of_variation= -1~+1
31
32
                   if mod(Number_of_gaps_on_half_axis,2) == 1
33
34
```

```
variation_of_dx = ((dx_mean-dx_min)./((
35
                          Number_of_gaps_on_half_axis-1)./2)).*
                          coefficient_of_variation;
36
37
                   else
38
                      variation_of_dx = ((dx_mean-dx_min)./((
39
                          Number_of_gaps_on_half_axis)./2)).*coefficient_of_variation
                          ;
40
                   end
41
42
                   variation_of_dx_vector = (dx_mean-dx_min).*
43
                      coefficient_of_variation* (-1) : variation_of_dx : (dx_mean-
                      dx_min) *coefficient_of_variation;
44
                   dx = variation_of_dx_vector + dx_mean;
                   spatial_axis = zeros(size(variation_of_dx_vector));
45
46
                  for i = 1:1:length(variation_of_dx_vector)
47
48
                      spatial_axis(i) = sum(dx(1:i)); % m
49
50
                   end
51
52
                  x = [];
53
                  y = [];
54
                  z = [];
55
56
                  for i = 1:1:length(spatial_axis)
57
58
                      % Define the side length and resolution of the cube
59
                      sideLength = spatial_axis(i);
60
                      resolution = 2*i+1;
61
62
63
                      \% Initialize the coordinates array
64
                      cubeShell = [];
65
66
                      % Generate grid points
67
                      [X, Y, Z] = meshgrid(linspace(-sideLength, sideLength,
68
                          resolution));
69
                      % Add points for each face of the cube
70
                      % Top and bottom faces (excluding the already added boundary
71
                          points)
                      cubeShell = [cubeShell; [reshape(X(:,:,1), [], 1), reshape(Y
72
                          (:,:,1), [], 1), reshape(Z(:,:,1), [], 1)]];
                      cubeShell = [cubeShell; [reshape(X(:,:,end), [], 1), reshape(Y
73
                          (:,:,end), [], 1), reshape(Z(:,:,end), [], 1)]];
74
                      % Left and right faces (excluding the already added boundary
75
                          points)
                      cubeShell = [cubeShell; [reshape(X(1:end,1,2:end-1), [], 1),
76
                          reshape(Y(1:end,1,2:end-1), [], 1), reshape(Z(1:end,1,2:end
                          -1), [], 1)]];
                      cubeShell = [cubeShell; [reshape(X(1:end,end,2:end-1), [], 1),
77
```

```
reshape(Y(1:end,end,2:end-1), [], 1), reshape(Z(1:end,end
                            ,2:end-1), [], 1)]];
78
                        % Front and back faces
79
                        cubeShell = [cubeShell; [reshape(X(1,2:end-1,2:end-1), [], 1),
80
                            reshape(Y(1,2:end-1,2:end-1), [], 1), reshape(Z(1,2:end
                            -1,2:end-1), [], 1)]];
                        cubeShell = [cubeShell; [reshape(X(end,2:end-1,2:end-1), [], 1)
81
                            , reshape(Y(end,2:end-1,2:end-1), [], 1), reshape(Z(end,2:
                            end-1,2:end-1), [], 1)]];
82
                        % reshape
83
                        x = [x cubeShell(:,1).'];
84
                        y = [y cubeShell(:,2).'];
85
                        z = [z cubeShell(:,3).'];
86
87
88
                    end
89
90
                    % reshape
91
                    x = [x 0];
92
                    y = [y 0];
93
                    z = [z 0];
94
95
                end
96
97
            case 2
98
99
                    dx = L / (Number_of_points_on_the_axis-1);
100
101
                    spatial axis = -L/2 : dx : L/2; % m
102
                    [x, y, z] = meshgrid(spatial_axis, spatial_axis, spatial_axis);
103
104
                    % reshape
105
                    x = x(:).';
106
                    y = y(:).';
107
                    z = z(:).';
108
109
110
            otherwise
111
112
                fprintf('Wrong|Input|ParameterNumber')
113
114
        end
115
116
117
118
    end
119
```

A cubic grid can be viewed as a nested set of cubic shells. creat_grid function does not generate the mesh nodes directly but instead generates the cubic shells one layer at a time; by adjusting the distance between the shells, creat_grid can make grid nodes closer to the surface or closer to the center. I use a parameter, the coefficient of variation C_v , in this code to measure grid variation. The coefficient varies between -1 and 1; as the coefficient gets closer to 1, the grid points are closer to the center; as the coefficient gets closer to -1, the grid points are closer to the surface.

3.3 The change in matrix stability expressed through Condition Number

Different gird shape choices directly affect the computational stability of the matrix in Eq. (2.26); this stability is demonstrated by the condition number. The condition number of a matrix is an important concept in numerical linear algebra, used as a measure of the sensitivity of the inverse of a matrix. In brief, the condition number measures the extent to which the output result may change when the change in a matrix is small. A higher condition number indicates that the matrix is closer to singular, and the numerical stability of its inverse is poorer, i.e., a small input error may lead to a large error in the output. When the condition number is low, the matrix is considered well-conditioned, meaning its inverse is numerically stable.

The following figure illustrates the condition number of a uniformly distributed cubic array with edge length L = 0.1 m, with $N_{axis} = 7$ points on the axis and N = 343 grid points.



Figure 3.4: An example of the Condition number of a Uniform distribution Cubic Array with side length L = 0.1 m, number of points on axis $N_{axis} = 7$, number of grid points N = 343.

Another valuable way to show condition numbers is to normalize the frequency axis in Fig. 3.4. This normalization is done in the following way:

$$k_{Normalized} = kr = \frac{\pi f L}{c}.$$
(3.1)

An example figure is given here.



Figure 3.5: An example of the Condition number plot in normalized frequency of a Uniform distribution Cubic Array with side length L = 0.1 m, number of points on axis $N_{axis} = 7$, number of grid points N = 343.

The above two Figures (Fig.3.4 and Fig.3.5) have expanded the frequency range of the frequency axis for demonstration purposes. In the results section, the frequency range of the frequency axis is set based on the following:

- 1. The lowest frequency depends on the size of the sampling grid, which is standardized to be at least half a wavelength that should fit inside the grid, corresponding to my default sampling grid with a 0.1 m side length, which has a minimum frequency of 118 Hz. For simplicity, 100 Hz is used in the result part as the lowest frequency in the plots.
- 2. The highest frequency depends on the order N of the spherical harmonics used to calculate the quadrature matrix. Since the Bessel term decays sharply at kr > N, the sound field reconstruction from the spherical harmonics becomes inaccurate at kr > N. With the geometry of the sampling grid known, we can calculate the corresponding maximum frequency, which in this case is 8734 Hz. For simplicity, 10 000 Hz is used in the result part as the highest frequency in the plots.

3.4 Compare results generated by different methods

As mentioned earlier, this thesis takes two approaches to generate binaural signals; the one obtained by directly convolving the original audio signal with the corresponding directional HRTF is the reference signal, which we consider to be the expected signal, while the signal obtained by the other approach is the experimental signals. In this thesis, I will simultaneously compare the experimental signals of several directions with the corresponding reference signals, and this comparison occurs in the frequency domain. I used a scheme similar to **spectrogram** for this comparison, where I replaced the time domain with an angle-based spatial domain compared to **spectrogram**.

In this comparison, two terms are used in this paper to express differences, with the following meanings:

1. Deviation of output(DO): Used to indicate the difference between two signals under the same conditions. In general, if not otherwise specified(Relatively, Absolutely), the term is calculated as follows:

$$DO = 20 \log_{10} \frac{\operatorname{Sig}_{PWD}(k)}{\operatorname{Sig}_{HRTF}(k)} d\mathbf{B}.$$
(3.2)

2. Deviation speed of output(DSO): Used to indicate the difference between two signals before and after a parameter change when a parameter changes gradually. In general, if not otherwise specified, the term is calculated as follows:

$$DSO = 20 \log_{10} \frac{\mathbf{Sig1_{PWD}}(k)}{\mathbf{Sig2_{PWD}}(k)} \mathbf{dB}.$$
(3.3)

It is easy to see that in both of the above comparisons, when the result is closer to $0 \, dB$, the more similar the two signals involved in the comparison.

3. Methods

Results

This paper tests the effect of changes in various parameters on the algorithm, but there are two leading indicators in it; one is the change in the condition number of the matrix mentioned in Eq. (2.26), which, as the most critical matrix in this thesis, has a crucial impact on the overall stability of the algorithm. The other is the effect of parameter changes on the binaural signals, which is directly related to the quality of the auralization. Therefore, this paper will describe the results from two aspects mentioned above.

4.1 Impact on the Condition Number

4.1.1 Influence caused by different grid size

In this paper, I have tested the condition number for different grid sizes to understand the principle behind it. The size of the grid is gradually increased from 0.1 mto 0.15 m in steps of 0.01 m. The result is shown below:



Figure 4.1: Condition number of Different Grid Size Cubic Arrays with number of points on axis $N_{axis} = 7$, number of grid points N = 343.

It is clear from the Figure.4.1 that the condition number decreases gradually from low to high frequencies at each grid size and becomes unstable from about 6000 Hz onwards and fluctuates but keeps at a low level. The high condition number at low

frequencies can be explained by the fact that the size of the sampling grid is relatively smaller than the wavelength of the low-frequency waves, leading to inaccuracies in the sampling. The fluctuations in the condition number at high-frequency can also be explained in terms of relative wavelength, i.e., at high frequencies, the grid node spacing is significantly larger than the wavelength, leading to the appearance of spatial aliasing effects. As a result, we can observe regular fluctuations at high frequencies, and the spatial aliasing effect is most pronounced when the grid nodes are spaced just enough to accommodate an integer number of high-frequency waves. This is also mentioned in the paper by Gilles Chardon[10] et al.

The plot obtained under regular frequency scaling does not intuitively observe the relation between the behavior of the condition numbers and the various grid sizes, so this Figure 4.1 was replotted at normalized frequencies as follows.



Figure 4.2: Condition number (Normalized Frequency) of Different Grid Size Cubic Arrays with number of points on axis $N_{axis} = 7$, number of grid points N = 343.

At normalized frequencies, it appears that only one line is plotted in the Figure.4.2, but in fact, the condition number curves of different grid sizes overlap almost perfectly. The almost perfect overlap of the multiple curves illustrates that the proportional scaling of the sampling grid has a proportional effect on the condition number. Another piece of information to note here is that multiple curves reach their minimum at the same time when the normalized frequency is around 8, which is a good indication that the Bessel term is only able to reconstruct the sound field accurately at kr < N, as it decays significantly at kr > N.

4.1.2 Influence caused by different number of points inside the grid

In this paper, I test the effect of different numbers of grid points on the condition number; here, I denote the number of nodes located on the center axis of the sampling grid by N_{axis} , and the total number of sampling points for this grid is N^3_{axis} . In performing this test step, the total number of nodes in the grid follows Eq. (2.27) to ensure that the data can support at least a spherical harmonic transform of order 8.

Unlike the effect of different grid sizes on the condition number, the number of grid nodes not only affects the performance of the condition number in the high-frequency part but also greatly influences its behavior in the low-frequency part. The following Figure 4.3 shows the effect of different numbers of grid nodes on the condition number for ordinary frequency scaling (left) and normalized frequency scaling (right).



Figure 4.3: Condition number of Different number of point on axis Cubic Arrays with grid side length L = 0.1 m

From the Figure.4.3, it can be seen that when $N_{axis} = 5$, i.e., N = 125, its corresponding condition number curve is significantly higher than the other curves at all frequencies. At high frequencies, this phenomenon can be interpreted as severe spatial aliasing due to sampling points that are too sparse. The high condition number at $N_{axis} = 5$ suggests that it is unsafe to follow Eq. (2.27) for grid point selection strictly. When $N_{axis} \ge 6$, the curves maintain a similar behavior at low frequencies but vary at high frequencies. However, it can still be conjectured that their lowest point is associated with the Bessel term. Overall, more sampling points ensure the matrix's stability at high frequencies.

Also, under the normalized frequencies scaled axis, it is clear that the conditional number curves for different grid node numbers have different behavioral patterns.

4.1.3 Influence caused by different distribution of points inside the grid

In this thesis, I design a method to change the distribution pattern of the nodes inside the mesh, the specific principle of which has already been stated in the Method section, so I will not go into too much detail here. However, it is still important to emphasize here that the coefficient of variation $C_v \in [-1, 1]$; the closer C_v is to 1, means that the nodes inside the grid are denser in the central part. Vice versa, the closer C_v is to -1, which means that the nodes inside the grid are closer to the grid surface.



Figure 4.4: Condition number of Different distribution Cubic Arrays with grid side length L = 0.1 m, number of points on axis $N_{axis} = 7$, number of grid points N = 343.

It can be seen that when the sampling points are closer to the surface of the grid, there is a slight improvement in the performance of the condition number at low frequencies, but it is tiny. As shown in the right picture of Figure.4.4, it can be seen that at high frequencies, the uneven sampling approach has some improvement on the fluctuations in the condition number, which can be interpreted as an effective reduction of spatial aliasing by the uneven spacing of the grid points. Overall, a sampling approach in which the distribution of grid points is close to the grid's surface is more favorable than a distribution in which the grid points are closer to the center. This is also demonstrated by the following figure, which plots the change in the average of the condition numbers over frequencies from 1000 Hz to 5000 Hz, which N_{axis} goes from 5 to 9 for the three distributions of grid: the grid points closer to the center $C_v = 0.8$.



Figure 4.5: Average of Condition Number between 1000 Hz and 5000 Hz of Cubic Arrays with grid side length L = 0.1 m

Figure 4.5 shows that the distribution of grid points closer to the surface of the cubic grid is somewhat better for the condition number when $N_{axis} > 5$.

4.2 Results generated by different parameters

In this section, I will show the differences between the audio signals. These differences are not limited to the differences between the audio signals generated by the PWD-based method and those generated by the HRTF-based method; they also include the difference between the audio signals generated by the PWD-based method under difference conditions and the rate of change of these differences.

4.2.1 Difference between ground truth output and the binaural output of three Anatomy planes

In this section, this paper first investigates the difference between ground truth output and the binaural output of three Anatomy planes. The three so-called anatomical planes of the human body are the Transverse Plane, the Sagittal Plane, and the Frontal Plane. The study of this subsection is conducted to understand the spatial performance characteristics of the audio signals generated by the PWD method; the three planes were selected based on the fact that they are intuitive as the characteristic planes of the human body. The three planes assumed in this thesis can be seen in the following Figure 4.6.



Figure 4.6: Human anatomy planes

The three sets of comparisons shown in this subsection all take place with the grid size of 0.1 m and grid points number of 343 points uniformly distributed. The signals compared are binaural signals of the right ear. I first compared the signal differences in the transverse plane, i.e., the horizontal direction, as shown below in Figure 4.7.



Figure 4.7: Deviation of the output depends on the incidence direction (dB)(Transverse, right ear)

Figure 4.7 shows that the Difference between ground truth output and the binaural output is not significant in the low frequencies range of about 50 Hz to 2000 Hz and gradually increases at higher frequencies. From about 2000 Hz, this difference demonstrates a clear directional dependence. The figure shows that when the sound comes from the left side of the body, the difference of the signal at the high frequency is that the high-frequency component of the signal generated by the PWD-based method is significantly more than that of the HRTF ground truth signal. When the sound comes from the right side of the body, the signals generated by the two methods are very similar in frequency distribution, which indicates that the signal generated by the PWD-based method is very close to the real situation. In other directions, the PWD-based method's high-frequency components of the signals are significantly less than those of the HRTF ground truth signal. A possible reason is that in the real world, low-frequency waves can propagate well through diffraction to the back of the occlusion, but high-frequency components cannot effectively propagate through diffraction to the back of the occlusion. In my thesis, although HRTF is involved in signal generation based on the PWD method, it still cannot reproduce the diffraction well in the actual situation. Thus, the high-frequency component can only be well simulated if the sound comes from the right and does not travel to the right ear in a diffractive manner. However, this difference may not be audibly significant since the human ear is not as sensitive to high-frequency sounds as around 1000 Hz. It is still important to increase the frequency of the boundaries at which high-frequency differences begin to occur.

Next, I calculated and plotted the difference in the sound signals as they came from each direction in the Frontal plane.



Figure 4.8: Deviation of the output depends on the incidence direction (dB)(Frontal, right ear)

In the Figure.4.8, we can see something similar to the Figure.4.7 occurs at high frequencies when the right ear becomes the occluded ear. At the same time, I can reasonably assume that when sound comes from directions of the Sagittal plane, the high-frequency components of the signal generated by the PWD-based method is less than those of the ground truth signal because the right ear is not completely obscured. When the right ear is not completely occluded (e.g., in direction 225° of Figure.4.7 and direction 45° top of Figure.4.8), the high-frequency components of the signal generated by the PWD-based method is less than those of the ground truth signal is a fixed pattern.

Finally, I plotted the differences in the Sagittal plane of different directions and displayed them in Figure.4.9 The general pattern is consistent with my predictions. At the same time, I observed that the reduction of the high-frequency component is not stable when the sound comes from different directions. A possible reason is that the frequency range we studied here is close to or even beyond the frequency range that can be simulated by our preset 8th-order spherical harmonics.

This subsection gives us a general idea of the distribution of the difference between ground truth output and the binaural output over space. The distribution of this difference in the Transverse plane is highly representative, and therefore, in the subsequent sections, I will choose only this one plane to study.



Figure 4.9: Deviation of the output depends on the incidence direction (dB)(Sagittal, right ear)

4.2.2 Influence caused by different distribution of points inside the grid

In the section on the condition number, we investigated the effect of different grid point distributions on the condition number based on the changing coefficient of variation. The conclusion is that by making the distribution of grid points closer to the surface, we can improve the performance of the condition number. In this subsection, I will investigate the effect of change in the coefficient of variation on the generated binaural signals.

I tested the performance when the coefficient of variation $C_v = -0.5$ and plotted the following figure. Here, it is necessary to explain what the words 'relatively' and 'absolutely' represent in the figure.

1. The term 'relatively' indicates that this figure compares signals generated by the PWD-based method, but one set of signals is generated when the grid points are non-uniformly distributed, and the other set used as a reference is generated when the grid points are uniformly distributed.

2. The term 'absolutely' indicates that the figure compares the difference between the signal generated by the PWD-based method and the HRTF ground truth signals.



Figure 4.10: Relatively deviation of the output depends on the incidence direction $C_v = -0.5$



Figure 4.11: Absolutely deviation of the output depends on the incidence direction $C_v = -0.5$

As shown in the Figure.4.10, changing the coefficient of variation does not significantly affect the results. Notably, there is a visible reduction in the high-frequency component of the sound signal at an angle of sound incidence of 90°, i.e., when the right ear becomes the occluded ear, which is a good phenomenon. This is also shown in the article written by Jens Ahrens[11]: when using spherical surface sampling and cubical surface sampling, the signal of the occluded ear is closer to the ground truth signal at high frequencies. In short, volumetric sampling with internal point uniform distribution is not a good sampling scheme.

As can be seen from the Figure 4.11, even if there is some improvement in the high-frequency part of the signal of the occluded ear, this improvement is not significant.

I also investigated the case when $C_v = 0.5$ and got results similar to those in Figure 4.10 and Figure 4.11. The details are shown in Figure 4.12 and Figure 4.13.



Figure 4.12: Relatively deviation of the output depends on the incidence direction $C_v = 0.5$ Deviation of the output depends on the incidence direction (dB) $C_v = 0.5$ absolutely



Figure 4.13: Absolutely deviation of the output depends on the incidence direction $C_v = 0.5$

In order to further investigate the effect of the changing coefficient of variation on the signal, I chose the 90° as the sound incidence angle, investigated the sound signals with different coefficients of variation, and obtained the rate of change of the signal by dividing the signals. The result is shown below.



Figure 4.14: Deviation speed of the output depends on the coefficient of variation

Figure.4.14 illustrates almost no change in the signal by varying the coefficient of variation when using 90° as the sound incidence angle. However, if one looks closely at the high-frequency part of the figure, one can still see that the variation shows a uniformly increasing or decreasing trend(deviation speed constantly greater than or less than 0) in some very narrow frequency bands. This means that when the grid's points distribution is very extreme (too close to the surface or too concentrated in the center), the change in distribution still significantly affects the high-frequency part of the spectral characteristics.

4.2.3 Influence caused by different grid nodes number

In this section, I will investigate the effect of the number of grid nodes while keeping the mesh size constant. I compared the difference between signals generated by the PWD-based method and the ground truth when the number of grid nodes gradually increased from $5^3 = 125$ to $10^3 = 1000$. Here, 45° is chosen as the sound incident angle. The results are as follows.



Figure 4.15: Deviation of the output depends on the number of grid points (dB) with grid side length L = 0.1 m

Earlier, I investigated its effect on the condition number and came to the conclusion that one should avoid the number of grid nodes being too close to the theoretical lower bound given by the formula $Q > (N+1)^2$, where N stands for the order of the spherical harmonic we choose when try to calculate quadrature matrix. However, in Figure.4.15, from a frequency point of view, the signal is still acceptable when the number of grid nodes is 125, and the high-frequency improvement from increasing the number of grid nodes is very limited.

4.2.4 Influence caused by the different impulse response length

While performing earlier tests, I realized that the quadrature matrix is a set of impulse responses when converted to the time domain. We can preset the length of that impulse response c_{nm} to greatly affect the computation time to generate a signal. A shorter impulse response will make a shorter operation time. However, intuitively, a shorter impulse response will also lead to information loss at low-frequency, which is determined by the sampling principle. In this thesis, N_{IR} denotes the impulse response length of c_{nm} . In the table below, I have listed the impulse response length with its required computation time, which is obtained by averaging the computational time of the program running.

Table 4.1: Computation time t for $N_{IR} = 2048, 1024, ...8$.

N _{IR}	2048	1024	512	256	128	64	32	16	8
t(s)	33.22	17.62	10.17	6.2	4.48	3.39	2.88	2.75	2.73

From Table.4.1, one can know that the computational time decreases as the impulse response length decreases. However, this relationship is not linear; when the impulse response c_{nm} is already short enough, the time used to call other functions is already significantly greater than the time reduction resulting from shortening the impulse response. In This thesis, I choose a very small value($N_{IR} = 16$) as the lowest limit of the impulse response length to explore the extreme cases; the sound incident direction is 45°. The results of the comparison between the signals are shown below.



Figure 4.16: Deviation of the output depends on the impulse length. Grid info: Side length L = 0.1 m, Number of points on axis $N_{axis} = 7$, Number of grid points N = 343.

When looking at the Figure 4.16 from right to left, it is interesting to note that when the impulse response length is less than 128, the audio signal generated by the PWDbased method differs from the ground truth significantly in all frequency bands. This phenomenon is easy to explain; with a known sampling rate of $fs = 48\,000$ Hz, it can be estimated that the lowest frequency that can be sampled with an impulse response length of 128 is about fs = 400 Hz, i.e., the impulse response does not express information below this frequency, so the low-frequency information is missing. In addition, if we consider the problem of spatial aliasing at high frequencies, the signal itself becomes unpredictable in the frequency range where low-frequency loss and spatial aliasing occur simultaneously.

To show at what point the signal stabilizes, I plotted the deviation speed of the output depending on the impulse length.



Figure 4.17: Deviation speed of the output depends on the impulse length. Grid info: Side length L = 0.1 m, Number of points on axis $N_{axis} = 7$, Number of grid points N = 343.

From Figure 4.17, it can be told that the signal is stable only when the impulse response length is greater than 256. This subsection demonstrates that it is feasible to shorten the computational time by decreasing the impulse response length. An excessively long impulse response length does not provide any advantage. Therefore, it is important to choose the appropriate impulse response length.

4.2.5 Influence caused by different grid size

In this section, I will discuss the effect of different grid sizes. To show the effect of grid size on the results, I picked a very aggressive range of mesh sizes, which is from 0.1 m to 4.3 m and sound incident direction is 45 degree.



Figure 4.18: Deviation of the output depends on the grid size (dB). Grid info: Number of points on axis $N_{axis} = 7$, Number of grid points N = 343.

The most conspicuous information in Figure.4.18 is that spatial aliasing occurs at lower frequently as the grid size increases, creating a very distinct boundary in the figure. In order to eliminate the effect of other information, I calculated the deviation rate by dividing the signal produced by the PDW-based method in the frequency domain. The result is shown below.



Figure 4.19: Deviation speed of the output depends on the grid size (dB). Grid info: Number of points on axis $N_{axis} = 7$, Number of grid points N = 343.

In Figure.4.19, we can clearly observe the pattern in which spatial aliasing occurs. Except for the lowest frequency at which spatial aliasing occurs, the other frequencies at which spatial aliasing occurs form several stripes almost parallel to the boundary formed by the lowest frequency at which spatial aliasing occurs.

Conclusion

This thesis performs a series of tests to explore the characteristics of the binaural signals generated by the PWD-based method and explores possible ways to reduce the amount of calculation. This paper has generated the following conclusions through the investigations in the above sections:

- 1. The way the grid is selected affects the results. Volumetric sampling is a simple, easy-to-use grid that can be changed slightly to produce better results. For example, spatial aliasing can be greatly avoided by breaking the uniformity of the mesh nodes. Although this approach may result in a relatively high condition number at high frequencies, it effectively avoids fluctuations in the condition number at high frequencies. At the same time, making the grid nodes closer to the surface of the sampling grid can effectively reduce the condition number and produce signals with better spectral quality.
- 2. Reducing the impulse response length used throughout the process is a very effective way to increase computational speed. Shortening the impulse response time reduces the time required for computation to one-fifth or even less when only the spectral quality of the generated signal is evaluated.
- 3. Although increasing the number of grid nodes while keeping the grid size constant stabilizes the condition number curve and very effectively improves the performance of the condition number at high frequencies, a larger number of grid nodes does not significantly improve the frequency characteristics of the generated signal. Therefore, it is not advisable to try to improve the signal quality by increasing the number of grid nodes while ensuring that the number of grid nodes is higher than the minimum requirement.

However, all the conclusions generated in this paper are based on two criteria: condition number and spectral information. In fact, many audible audio signals were generated while performing the various tests. Listening experiments are bound to give more valuable information.

5. Conclusion

Bibliography

- K. Kowalczyk and M. Van Walstijn, "Room acoustics simulation using 3-d compact explicit fdtd schemes," Audio, Speech, and Language Processing, IEEE Transactions on, vol. 19, pp. 34 – 46, 02 2011.
- [2] T. Xiao and Q. Huo Liu, "Finite difference computation of head-related transfer function for human hearing," *The Journal of the Acoustical Society of America*, vol. 113, no. 5, pp. 2434–2441, 2003.
- [3] J. Sheaffer, C. Webb, and B. M. Fazenda, "Modelling binaural receivers in finite difference simulation of room acoustics," in *Proceedings of Meetings on Acoustics*, vol. 19, AIP Publishing, 2013.
- [4] P. Mokhtari, H. Takemoto, R. Nishimura, and H. Kato, "Computer simulation of hrtfs for personalization of 3d audio," in 2008 Second International Symposium on Universal Communication, pp. 435–440, IEEE, 2008.
- [5] B. Støfringsdal and P. Svensson, "Conversion of discretely sampled sound field data to auralization formats," *Journal of the Audio Engineering Society*, vol. 54, no. 5, pp. 380–400, 2006.
- [6] J. Sheaffer, M. Van Walstijn, B. Rafaely, and K. Kowalczyk, "Binaural reproduction of finite difference simulations using spherical array processing," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2125–2135, 2015.
- [7] B. Rafaely, Fundamentals of spherical array processing, vol. 8. Springer, 2015.
- [8] G. B. Arfken and H. J. Weber, "Mathematical methods for physicists 6th ed.," Mathematical methods for physicists 6th ed. by George B. Arfken and Hans J. Weber. Published: Amsterdam; Boston: Elsevier, 2005.
- [9] J. Ahrens, "auralization-toolbox," 2023. https://github.com/ AppliedAcousticsChalmers/auralization-toolbox, last accessed on February 25, 2024.
- [10] G. Chardon, W. Kreuzer, and M. Noisternig, "Design of spatial microphone arrays for sound field interpolation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 5, pp. 780–790, 2015.
- [11] J. Ahrens, "A software tool for auralization of simulated sound fields," Proc. of the Institute of Acoustics, vol. 45, Sep. 2023.

Department of Architecture and Civil Engineering CHALMERS UNIVERSITY OF TECHNOLOGY Gothenburg, Sweden www.chalmers.se

