Technische Universität Berlin

Fakultät I - Geisteswissenschaften

Institut für Sprache und Kommunikation

Fachgebiet Audiokommunikation

04. Mai 2014

Zur modalen Zerlegung der Richtcharakteristik von Line Array Systemen

Masterarbeit

vorgelegt von: Lutz Jakob Bergner

[redacted]

Erstprüfer: Prof. Dr. Stefan Weinzierl

Zweitprüfer: Frank Schultz, M.Sc.

Erklärung

Hiermit erkläre ich an Eides statt gegenüber der Fakultät I der Technischen Universität Berlin, dass die vorliegende, dieser Erklärung angefügte Arbeit selbstständig und nur unter Zuhilfenahme der im Literaturverzeichnis genannten Quellen und Hilfsmittel angefertigt wurde. Alle Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, sind kenntlich gemacht. Ich reiche die Arbeit erstmals als Prüfungsleistung ein.

Berlin, den 30. April 2014 .....

Lutz Jakob Bergner

Danksagung

Ich möchte Stefan Weinzierl für die fachliche und inhaltliche Unterstützung danken. Großer Dank geht auch an Frank Schultz für das große Engagement bei der Betreuung dieser Arbeit. Anselm Goertz und dem Institut für Akustik und Audiotechnologie sei für die Bereitstellung des Messdatensatzes gedankt.

Ein Besonderer Dank für die Unterstützung während meines gesamten Studiums gilt meiner Familie.

Zusammenfassung

Im Bereich der virtuellen Akustik ist die Schallfeldanalyse mittels *Spherical Harmonic Transform* $\mathcal{SHT}$ eine weit verbreitete und gut dokumentierte Methode. Der Großteil der Publikationen zu diesem Thema bezieht sich dabei auf innere Probleme. Diese Abeit konzentriert sich dagegen auf äußere Probleme, d.h. auf die Beschreibung von (einzelnen) Schallquellen. Konkret wird dabei ein Line Array Lautsprecher betrachtet. Mit Hilfe der inversen $\mathcal{SHT}^{-1}$ soll dieser in sein Modalspektrum zerlegt werden und anschließend auf seine Richtcharakteristik hin untersucht werden. Dabei wird ein Matlab-Code vorgestellt, der die Analyse hinsichtlich des Abstrahlverhaltens durchführt. Anschließend soll dieses mit der analytischen Beschreibung von idealen Linienquellen verglichen werden.

Abstract

The approach of the *spherical harmonic transform* $\mathcal{SHT}$ of a sound field is a well document method in the field of sound field analysis. However most publications to this topic deal with a interior problem. This thesis on the other hand takes a look on the inverse, exterior problem to describe sound sources in general. In doing so a line array loudspeaker is taken under consideration. Its propagated sound field is decomposed to its modal coefficients via the inverse $\mathcal{SHT}^{-1}$, whereupon the directivity characteristic is examined. This is implemented in a practical Matlab source code which is introduced and discussed. Finally the directional behaviour of the loudspeaker is compared to the analytical sound propagation properties of ideal line sources.

Inhaltsverzeichnis

1. Einleitung **1**

2. Schallfeldanalyse und -synthese **3**

2.1. Koordinatensysteme . . . 3
2.2. Fourier-Transformation . . . 4
2.3. Wellengleichung . . . 5
2.3.1. Ebene Wellen in kartesischen Koordinaten . . . 6
2.3.2. Sphärische Wellen in Kugelkoordinaten . . . 7
2.4. Inneres vs. äußeres Problem . . . 10
2.5. Spherical Harmonic Transform $\mathcal{SHT}$ . . . 11
2.6. Plane Wave Decomposition . . . 13
2.7. Darstellung im Wellenzahlenraum . . . 14
2.8. Schallfeldsynthese mittels Single Layer Potential . . . 15

3. Analytische Modelle **18**

3.1. Fernfeldbedingungen . . . 18
3.2. Analytische und numerische Lösungen grundlegender Schallquellen . . . 19
3.2.1. Kugelmonopol . . . 19
3.2.2. Rechteckige Kolbenmembran . . . 20
3.2.3. Linienquelle . . . 24
3.3. Kugelabtastung . . . 25
3.4. Stör- und Fehlerquellen . . . 27
3.4.1. Spatial Aliasing bei der Analyse $\mathcal{SHT}$ . . . 27
3.4.2. Spatial Aliasing bei der Synthese $\mathcal{SHT}^{-1}$ . . . 29
3.4.3. Parametrisierung der spatialen DFT . . . 30

4. Implementierung einer modalen Zerlegung von realen Balloon-Messdaten **32**

4.1. Struktur und Aufbereitung des Datensatzes . . . 33
4.2. Implementierung und Evaluierung von Matlab-Funktionen . . . 35
4.2.1. Koordinatensysteme . . . 35
4.2.2. FFT/IFFT . . . 36
4.2.3. Sphärische Harmonische . . . 37
4.2.4. Einlese- und Aufbereitungsroutine . . . 37
4.2.5. Abtastgrid . . . 37

4.2.6. $\mathcal{SHT}$ 39
4.2.7. Wellenzahlenspektrum aus $\breve{P}_n^m$ 41

5. Ergebnisse und Diskussion **46**
5.1. Gültigkeitsbereich des vorliegenden Datensatzes 46
5.2. Analyse der Ausgangsdaten 47
5.3. Anwendung der modalen Zerlegung 50
5.4. Schallfeldexpansion und k-Spectrum 52
5.5. Zusammenfassung 54
5.6. Ausblick 55

Abbildungsverzeichnis **57**

Tabellenverzeichnis **58**

Literatur **59**

A. Matlab Code **62**
A.1. main functions 63
A.2. tools 71
A.3. plot routines 82

Symbolverzeichnis

Symbol	Beschreibung
FR-BAL	Full-Range Balloon (Messdatensatz)
LF-BAL	Low frequency Balloon (Messdat.)
LM-BAL	Low mid freq. Balloon (Messdat.)
HM-BAL	High mid freq. Balloon (Messdat.)
HF-BAL	High freq. Balloon (Messdat.)
DFT	Diskrete Fouriertransformation
$\mathcal{F}_t$,$\mathcal{F}_t^{-1}$	(inverse) zeitliche Fouriertransformation, Kap. 2.2
$\mathcal{F}_{\mathbf{x}}$,$\mathcal{F}_{\mathbf{x}}^{-1}$	(inverse) örtliche Fouriertransformation, Kap. 2.2
$\mathcal{PWD}$	Plane Wave Decomposition, Kap. 2.6
$\mathcal{SHT}$, $\mathcal{SHT}^{-1}$	(inverse) Spherical Harmonic Transform, Kap. 2.5
c	Schallgeschwindigkeit (wenn nicht anders definiert in dieser Arbeit $c = 343\frac{m}{s}$)
f	(Zeit-) Frequenz in [Hz]
$G\|\|\mathbf{x} - \mathbf{x_0}\|\|, \omega)$	dreidimensionale Green'sche Funktion im Freifeld
$h_n^{(1,2)}(\cdot)$	sphärische Hankelfunktion 1. oder 2. Art
j	imaginäre Einheit $\mathrm{j}^2 = -1$
$j_n(\cdot)$	sphärische Besselfunktion 1.Art
$\mathbf{k}$	Wellenzahlenvektor in kart. Koord. $\mathbf{k} = (k_x, k_y, k_z)$
$k = \|\|\mathbf{k}\|\|$	Wellenzahl in kart. Koord. $k = \sqrt{k_x^2 + k_y^2 + k_z^2}$
L	Anzahl der Abtastpunkte auf der Kugeloberfläche
m	Mode der sphärischen Harmonischen, $m = -n \ldots n$
N	maximale Ordnung der $\mathcal{SHT}$
n	Ordnung der sphärischen Harmonischen $0 \leq n \leq N$
$P_n^m(\cdot)$	zugeordnetes Legendrepolynom
$P(\mathbf{x}, \omega)$	Schallfeld im Ort $\mathbf{x}$
$\tilde{P}(\mathbf{k}, \omega)$	Schallfeld betrachtet im Wellenzahlenraum
$\mathring{P}_n^m(r, \omega)$	Modalspekrtum in Abhängigkeit von r
$\breve{P}_n^m(\omega)$	Modalspektrum nach Eliminierung des Radialterms
r	Radius
$\mathbf{x}$	Ortstvektor in kartesischen $\mathbf{x} = (x, y, z)$ oder sphärischen Koordinaten $\mathbf{x} = (r, \phi, \theta)$
$Y_n^m(\theta, \phi)$	Sphärische Harmonische
$y_n(\cdot)$	sphärische Besselfunktion 2.Art bzw. Neumannfunktion
$\delta_{nn'}$	Kronecker Delta
$\omega = 2\pi f$	Kreisfrequenz [rad/s]
φ	Azimuth im Interval $[0, 2\pi)$
θ	Elevation im Interval $[0, \pi]$

1. Einleitung

Die vorliegende Arbeit beschäftigt sich mit der Messung und Bewertung der Schallabstrahlung von Schallquellen im Allgemeinen und der Beschreibung der Richtcharakteristik von Line Array Systemen im Speziellen. Line Arrays erfreuen sich bei der Publikumsbeschallung im Veranstaltungsbereich großer Beliebtheit. Der Grund dafür ist die Eigenschaft dieser Systeme das Abstrahlverhalten von Linienquellen nachzuahmen. Je nach Größe der zu beschallenden Zuhörerfläche werden Line Array Elemente aneinandergereiht, um so eine lange, durchgehende Linienquelle zu schaffen. Linienquellen haben die Eigenschaft, den Schall mit steigender Frequenz sehr gerichtet abzustrahlen. Mit der entsprechenden Signalverarbeitung kann somit erreicht werden, dass der Schall hauptsächlich auf die relevanten Zuhörerflächen gerichtet wird. Grundlegende analytische Untersuchungen der Richtcharakteristik von Line Array Systemen sind in [HU92], [UHB03] und [Ure04] zu finden. Dort werden geometrische Prädiktionen der Schallabstrahlung vorgestellt, die auf der Summierung von Elementarschallquellen auf einer Linie beruhen. Ein anderer, messtechnischer Ansatz zur Beschreibung von Lautsprechern im Allgemeinen ist das Erstellen von Balloon-Datensätzen. Dazu wird auf einer Kugeloberfläche, die den Lautsprecher im Fernfeld einschließt, an definierten Orten eine Impulsantwort bzw. ein komplexes Spektrum gemessen. Mit geeigneter Software können diese Spektren zusammengesetzt und eine dreidimensionale Darstellung der Richtcharakteristik des Lautsprechers erstellt werden. Durch komplexe Multiplikation dieser Balloons kann so auch die Überlagerung mehrerer Lautsprecher (bzw. hier: Line Array Elementen) vorhergesagt werden. Der Einsatz von Balloon-Datensätzen ist in vielen Softwares zur Akustik-Simulation bereits weit verbreitet. Ein Beispiel dafür ist die *Generic Loudspeaker Library* (GLL), die von der Firma AFMG entwickelt wurde und von vielen Lautsprecher-Herstellern unterstützt wird [FAB05].
Allerdings beschreiben diese Balloon-Daten streng genommen nicht die Schallquelle (Lautsprecher) selbst, sondern lediglich deren produziertes Schallfeld auf eben jener Kugeloberfläche. Zu diesem Zweck existieren im Bereich der virtuellen Akustik Werkzeuge zur Analyse und Synthese dreidimensionaler Schallfelder. Die Grundlagen dieser Werkzeuge sind u.a. in [Wil99], [Bla00], [Ahr12] beschrieben. Mit Hilfe der *Spherical Harmonic Transform* ($\mathcal{SHT}$) ist es nun also möglich, die gemessenen Balloon-Daten modal zu transformieren, sodass quellbeschreibende Koeffizienten entstehen. Mit diesen Koeffizienten kann nun die Schallabstrahlung an (mit Einschränkungen) beliebigen Orten im Raum simuliert werden. Die praktische Anwendung dieser Transformation obliegt allerdings bestimmten Grenzen, die von der numerischen Realisierung der mathematischen

Zusammenhänge abhängen, d.h. konkret von der Diskretisierung in Zeit und Raum sowie von endlichen Rechenkapazitäten. Einen Ansatz für diese Herangehensweise findet man bereits in [Bru07], [FBN+08] und [Faz10].

Das Ziel dieser Arbeit ist somit zweigeteilt. Zum einen soll eine Matlab-Implementierung entstehen, die Balloon-Datensätze mittels der $\mathcal{SHT}$ unter Berücksichtigung der entsprechenden Grenzen in ihr Modalspektrum zerlegt. Zum anderen soll ein Datensatz eines realen Line Array Elements mit dieser Implementierung bearbeitet werden und dessen Richtcharakteristik mit der einer idealen Linienquelle verglichen werden. Dabei sollen Fragen zu Störeinflüssen beantwortet werden und eine sinnvolle grafische Aufarbeitung der Ergebnisse vorgestellt werden. Zunächst werden in Kapitel 2 die verwendeten Grundlagen der Schallfeldanalyse und -synthese vorgestellt. Anschließend werden im Kapitel 3 weiterführende analytische Modelle als Referenz zu den Messungen besprochen. Kapitel 4 beinhaltet die Beschreibung und Evaluierung des erstellten Matlab-Codes. Zum Schluss werden in Kapitel 5 die Ergebnisse des Forschungsziels dargestellt und diskutiert sowie ein Ausblick auf deren praktische Nutzbarkeit gegeben.

2. Schallfeldanalyse und -synthese

Dieses Kapitel enthält die mathematisch-physikalischen Zusammenhänge und Beschreibungen, die für die Analyse und Synthese von Schallfeldern im dreidimensionalen Raum gebräuchlich sind. Dabei werden auch einige Vorzeichen- und Normierungskonventionen von mathematischen Operationen für diese Arbeit festgelegt, die in der einschlägigen Literatur unterschiedlich gehandhabt werden.

2.1. Koordinatensysteme

Zunächst sei darauf hingewiesen, dass die Vektornotation weitesgehend aus [DLMF, §1.6(i)] übernommen wird. Es gilt für einen Vektor in kartesischen Koordinaten $\vec{a} = \mathbf{a} = (a_1, a_2, a_3)$ der Betrag $||\mathbf{a}|| = \sqrt{a_1^2 + a_2^2 + a_3^2}$ und das Skalarprodukt $\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + a_3 b_3$.

In dieser Arbeit werden Koordinaten in kartesischer oder sphärischer Darstellung genutzt. Dabei gelten die Rechenregeln

$$\mathbf{x} = (x, y, z) = r\,(\cos\varphi\ \sin\theta, \sin\varphi\ \sin\theta, \cos\theta) \tag{2.1a}$$

und

$$\mathbf{x} = (r, \varphi, \theta)$$

$$r = \sqrt{x^2 + y^2 + z^2} \qquad r \in [0, \infty) \tag{2.1b}$$

$$\varphi = \arctan\left(\frac{y}{x}\right) \qquad \varphi \in [0, 2\pi) \tag{2.1c}$$

$$\theta = \arccos\left(\frac{z}{r}\right) \qquad \theta \in [0, \pi] \tag{2.1d}$$

Auch wenn es sich bei den sphärischen Koordinaten (r, φ, θ) genau genommen nicht um Vektoren handelt, erhalten sie die dennoch die gleiche Darstellung, damit die Überführung $\mathbf{x} = (x, y, z) = (r, \varphi, \theta)$ übersichtlich gelingt. Analog dazu gilt die Beziehung im Wellenzahlraum $\mathbf{k} = (k_x, k_y, k_z) = (k, \varphi, \theta)$. In Abbildung 2.1 sind die entsprechenden Beziehungen grafisch dargestellt. In einigen dreidimensionalen Plots sind die Achsenbezeichnungen der Übersichtlichkeit wegen weggelassen worden. Stattdessen sind die Koordinatenachen x, y, z der Reihenfolge nach durch die Primärfarben RGB dargestellt. Die blaue, positive z-Achse zeigt dabei stets nach oben, während die anderen Achsen mit

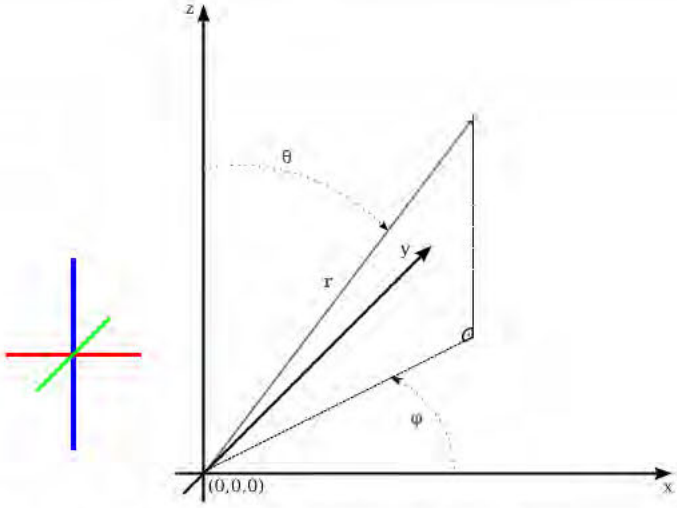


Abbildung 2.1.: Schematische Darstellung des Koordinatensystems

Hilfe des Rechtssystems und der Farbcodierung abgeleitet werden können.

2.2. Fourier-Transformation

Bezüglich der Fourier-Transformation halten wir uns in dieser Arbeit an die Konvention aus [Bla00, Ahr12]. Die kontinuierliche zeitliche (temporale) Fourier-Transformation wird demnach mit

$$\begin{aligned} P(\mathbf{x},\omega) &= \mathcal{F}_t\left\{p(\mathbf{x},t)\right\} \\ P(\mathbf{x},\omega) &= \int_{-\infty}^{+\infty} p(\mathbf{x},t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t \end{aligned} \tag{2.2}$$

definiert und ihre Inversion mit

$$\begin{aligned} p(\mathbf{x},t) &= \mathcal{F}_t^{-1}\left\{P(\mathbf{x},\omega)\right\} \\ p(\mathbf{x},t) &= \frac{1}{2\pi}\int_{-\infty}^{+\infty} P(\mathbf{x},\omega)\mathrm{e}^{+\mathrm{j}\omega t}\mathrm{d}\omega \quad . \end{aligned} \tag{2.3}$$

Die räumliche (spatiale) Fourier-Transformation wird dagegen mit umgekehrtem Vorzeichen im Exponenten definiert. Der Grund dafür liegt in der Richtung der ebenen Wellenausbreitung. Mit der Konvention, wie sie hier benutzt wird, zeigt der Vektor $\mathbf{k}$ in die Richtung, in die sich die Welle ausbreitet. Mit umgekehrten Vorzeichen (also analog zur temporalen Fourier-Transformation) würde $\mathbf{k}$ in die Richtung zeigen, aus der die Welle kommt (vgl. [Ahr12, Kap. 2.2.6]). Es gilt also exemplarisch für die Transformation

in x-Richtung

$$\begin{aligned}\tilde{P}(k_x, y, z, \omega) &= \mathcal{F}_x \left\{P(\mathbf{x}, \omega)\right\} \\ \tilde{P}(k_x, y, z, \omega) &= \int_{-\infty}^{+\infty} P(\mathbf{x}, \omega) \mathrm{e}^{+\mathrm{j}k_x x} \, \mathrm{d}x\end{aligned} \tag{2.4}$$

und

$$\begin{aligned}P(\mathbf{x}, \omega) &= \mathcal{F}_x^{-1} \left\{\tilde{P}(k_x, y, z, \omega)\right\} \\ P(\mathbf{x}, \omega) &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \tilde{P}(k_x, y, z, \omega) \mathrm{e}^{-\mathrm{j}k_x x} \mathrm{d}k_x \quad .\end{aligned} \tag{2.5}$$

2.3. Wellengleichung

Grundlage für die Betrachtung von Schallfeldern ist die homogene, linearisierte Wellengleichung

$$\nabla^2 p(\mathbf{x}, t) - \frac{1}{c^2} \frac{\partial^2 p(\mathbf{x}, t)}{\partial t^2} = 0 \quad , \tag{2.6}$$

wie sie etwa in [MÖ9, Kap. 2.2.1 und 2.3] hergeleitet wird. Sie beschreibt den Schalldruck p am Ort $\mathbf{x} = (x, y, z)$ zum Zeitpunkt t, wobei c die Schallgeschwindigkeit bezeichnet (in dieser Arbeit wird $c = 343\mathrm{m/s}$ angesetzt) und der Laplace-Operator ∇^2 die zweifache Ableitung nach dem Ort bedeutet. Die Wellengleichung in dieser Form ist dann gültig, wenn sämtliche Schallfelder außerhalb von $\mathbf{x}$ liegen (Homogenität).
Mit der temporalen Fourier-Transformation aus (2.2) kann die Beziehung

$$\begin{aligned}\frac{\partial p(\mathbf{x}, t)}{\partial t} &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \mathrm{j}\omega P(\mathbf{x}, \omega) \mathrm{e}^{+\mathrm{j}\omega t} \mathrm{d}\omega \\ \mathcal{F}\left\{\frac{\partial p(\mathbf{x}, t)}{\partial t}\right\} &= \mathrm{j}\omega P(\mathbf{x}, \omega)\end{aligned} \tag{2.7}$$

aufgestellt werden. So kann die Wellengleichung aus (2.6) in den Frequenzbereich transformiert werden, wodurch wir die Helmholtz-Gleichung

$$\nabla^2 P(\mathbf{x}, \omega) + k^2 P(\mathbf{x}, \omega) = 0 \tag{2.8}$$

erhalten, in der

$$k^2 = \left(\frac{\omega}{c}\right)^2 \tag{2.9}$$

die Wellenzahl bezeichnet (vgl. [Wil99, Kap. 2.4]).

2.3.1. Ebene Wellen in kartesischen Koordinaten

In kartesischen Koordinaten entsprechend Abbildung 2.1 wird der Gradient ∇ mit

$$\nabla = \left(\frac{\partial}{\partial x}\mathbf{e}_x, \frac{\partial}{\partial y}\mathbf{e}_y, \frac{\partial}{\partial z}\mathbf{e}_z \right) \tag{2.10}$$

gebildet, wobei $\mathbf{e}_x$, $\mathbf{e}_y$, und $\mathbf{e}_z$ die Einheitsvektoren des Koordinatensystems darstellen. In Anbetracht der Helmholtz-Gleichung (2.8) kann eine ebene Welle in kartesischen Koordinaten mit

$$P(\mathbf{x}, \omega) = \hat{P}(\omega)\mathrm{e}^{-\mathrm{j}(k_x x + k_y y + k_z z)} = \hat{P}(\omega)\mathrm{e}^{-\mathrm{j}\mathbf{k}\mathbf{x}} \tag{2.11}$$

beschrieben werden, wobei $\hat{P}(\omega)$ die Amplitude der entsprechenden Frequenz und $\mathbf{k} = (k_x, k_y, k_z)$ den Wellenvektor, also die Ausbreitungsrichtung, darstellen. Diese Lösung gilt nur, wenn die Dispersions-Gleichung

$$k^2 = ||\mathbf{k}||^2 = \mathbf{k} \cdot \mathbf{k}$$

und konkret in kartesischen Koordinaten

$$k^2 = k_x^2 + k_y^2 + k_z^2 \tag{2.12}$$

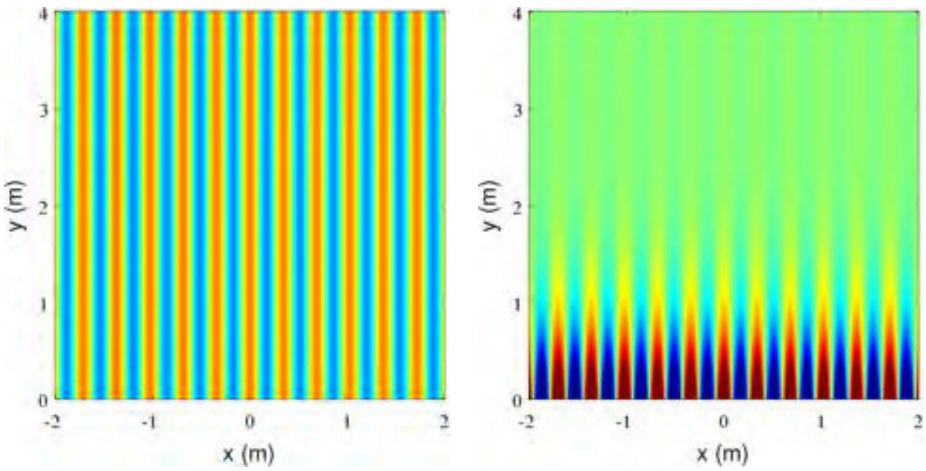
erfüllt ist ([Wil99, Kap. 2.6.2]). Es ist zu beachten, dass k_x, k_y und k_z nicht beschränkt sind. Bei der Umstellung der Dispersions-Gleichung (2.12) zum Beispiel nach k_y behält die Wellengleichung auch für den Fall $k_x^2 + k_z^2 > k^2$ ihre Gültigkeit. Deshalb muss hier eine Fallunterscheidung angewandt werden ([Wil99, Kap. 2.6.3]):

$$k_y = \begin{cases} \pm\sqrt{k^2 - k_x^2 - k_z^2} & \text{für } k^2 \geq k_x^2 + k_z^2 \\ \pm\mathrm{j}\sqrt{k_x^2 + k_z^2 - k^2} & \text{für } k^2 < k_x^2 + k_z^2 \end{cases} \tag{2.13}$$

Im ersten Fall erhalten wir eine propagierende, also ausstrahlende, ebene Welle. Der zweite Fall erzeugt ein evaneszentes, also in y-Richtung rasch abnehmendes Nahfeld. Dies kann nachvollzogen werden, wenn wir $k_y = \pm\mathrm{j}\sqrt{k_x^2 + k_z^2 - k^2}$ in die Helmholtz-Gleichung (2.11) einsetzen und die e-Funktion auftrennen:

$$P(\mathbf{x}, \omega) = \hat{P}(\omega)\underbrace{\mathrm{e}^{\pm\sqrt{k_x^2+k_z^2-k^2}\ y}}_{\mathrm{e}^{-\mathrm{j}k_y y}}\mathrm{e}^{-\mathrm{j}(k_x x + k_z z)} \tag{2.14}$$

Der unterklammerte Term ist in diesem Fall reell. Das heißt bei positivem Vorzeichen vor der Wurzel divergiert dieser für $y \to \infty$ und ist somit physikalisch nicht real. Die Lösung ist an dieser Stelle also auf den abklingenden Fall mit negativem Vorzeichen beschränkt.



(a) (b)

Abbildung 2.2.: Simulation einer ebenen Welle der Frequenz $f = 1000$Hz für den propagierenden Fall (a) und den evaneszenten Fall (b). Quelle: [Ahr12, Abb. 2.1]

Es gilt für den evaneszenten Fall in (2.13) also

$$\lim_{y\to\infty} \mathrm{e}^{-\sqrt{k_x^2+k_z^2-k^2}\ y} = 0 \tag{2.15}$$

Abbildung 2.2 zeigt eine Simulation von propagierender und evaneszenter Schallausbreitung. Nach der Rücktransformation von (2.11) in den Zeitbereich erhalten wir entsprechend den Ausführungen in [Ahr12, App.B], [MÖ9, Kap. 2.2.3] für die Beschreibung einer monochromatischen, ebenen Wellenausbreitung den Ausdruck

$$p(\mathbf{x}, t) = \Re\left\{\hat{P}(\omega)\mathrm{e}^{\mathrm{j}(\omega t - \mathbf{kx})}\right\} \quad . \tag{2.16}$$

2.3.2. Sphärische Wellen in Kugelkoordinaten

In sphärischen Koordinaten wird der Gradient ∇ durch

$$\nabla = \frac{\partial}{\partial r}\mathbf{e}_r + \frac{1}{r}\frac{\partial}{\partial \theta}\mathbf{e}_\theta + \frac{1}{r\sin\theta}\frac{\partial}{\partial \varphi}\mathbf{e}_\varphi \tag{2.17}$$

mit den Einheitsvektoren

$$\mathbf{e}_r = \begin{bmatrix} \cos\varphi\sin\theta \\ \sin\varphi\sin\theta \\ \cos\theta \end{bmatrix}; \quad \mathbf{e}_\theta = \begin{bmatrix} \cos\varphi\cos\theta \\ \sin\varphi\cos\theta \\ -\sin\theta \end{bmatrix}; \quad \mathbf{e}_\varphi = \begin{bmatrix} -\sin\varphi \\ \cos\theta \\ 0 \end{bmatrix} \tag{2.18}$$

bestimmt [Ahr12, (2.12-13)]. Entsprechend (2.3.2) ist der Laplace-Operator ∇^2 gemäß [DLMF, (1.5.17)] für $p(r, \varphi, \theta)$

$$\nabla^2 p = \frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2\frac{\partial p}{\partial r}\right] + \frac{1}{r^2 \sin\theta}\frac{\partial}{\partial \theta}\left[\sin\theta\frac{\partial p}{\partial \theta}\right] + \frac{1}{r^2 \sin^2\theta}\frac{\partial^2 p}{\partial \varphi^2} \quad . \tag{2.19}$$

Die Wellengleichung in sphärischen Koordinaten für $p(r, \varphi, \theta, t)$ ist nach [Bla00, (10.A-1)] also

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left[r^2\frac{\partial p}{\partial r}\right] + \frac{1}{r^2 \sin\theta}\frac{\partial}{\partial \theta}\left[\sin\theta\frac{\partial p}{\partial \theta}\right] + \frac{1}{r^2 \sin^2\theta}\frac{\partial^2 p}{\partial \varphi^2} - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0 \quad . \tag{2.20}$$

Mit Hilfe der Variablenseparation können daraus Eigenlösungen für die einzelnen Dimensionen

$$p(r, \varphi, \theta, t) = f(r) \cdot f(\varphi) \cdot f(\theta) \cdot f(t) \tag{2.21}$$

abgeleitet werden. Die theoretische Herleitung dessen kann etwa in [Bla00, Kap. 10.B.1-5],[Wil99, Kap. 6.2] gefunden werden, weshalb an dieser Stelle lediglich die Lösungen in Tabelle 2.1 aufgeführt sind. Die Lösungen für den zeitlichen Term $f(t)$ sind aus der

Einzelfunktion	**Eigenlösungen**	
$f(r)$	Sphärische Besselfunktion 1. Art	$j_n(\frac{\omega}{c}r)$
	Sphärische Besselfunktion 2. Art	$y_n(\frac{\omega}{c}r)$
	Sphärische Hankelfunktion 1. Art	$h_n^{(1)}(\frac{\omega}{c}r)$
	Sphärische Hankelfunktion 2. Art	$h_n^{(2)}(\frac{\omega}{c}r)$
$f(\varphi)$	Trigonometrische Funktionen	$\sin m\varphi \quad m \in \mathbb{Z}$
		$\cos m\varphi \quad m \in \mathbb{Z}$
	Komplexe Exponentialfunktionen	$\mathrm{e}^{+\mathrm{j}m\varphi} \quad m \in \mathbb{Z}$
		$\mathrm{e}^{-\mathrm{j}m\varphi} \quad m \in \mathbb{Z}$
$f(\theta)$	Zugeordnetes Legendrepolynom	$P_n^m(\cos\theta)$
$f(t)$	Trigonometrische Funktionen	$\sin\omega t$
		$\cos\omega t$
	Komplexe Exponentialfunktionen	$\mathrm{e}^{+\mathrm{j}\omega t}$
		$\mathrm{e}^{-\mathrm{j}\omega t}$

Tabelle 2.1.: Zuordnung der Eigenlösungen für die entsprechenden Teilfunktion der Wellengleichung in Kugelkoordinaten

Signalverarbeitung bekannt. Da in dieser Arbeit hauptsächlich im Frequenzbereich gearbeitet wird, wird dieser Term inhärent mit $\mathrm{e}^{\mathrm{j}\omega t}$ angenommen und im Folgenden stets weggelassen. Für die Lösungen des radialen Terms $f(r)$ bestehen folgende Zusammenhänge:

$$\begin{aligned} h_n^{(1)}\left(\frac{\omega}{c}r\right) &= j_n\left(\frac{\omega}{c}r\right) + \mathrm{j}\ y_n\left(\frac{\omega}{c}r\right) \\ h_n^{(2)}\left(\frac{\omega}{c}r\right) &= j_n\left(\frac{\omega}{c}r\right) - \mathrm{j}\ y_n\left(\frac{\omega}{c}r\right) \end{aligned} \tag{2.22}$$

Die sphärische Hankelfunktion h_n ist die Summe der sphärischen Besselfunktion 1. Art j_n und 2.Art (auch: Neumannfunktion) y_n. Um Verwechslungen vorzubeugen, ist die sphä-

rische Besselfunktion j_n kursiv und die imaginäre Einheit j gerade gesetzt. Abbildung 2.3 zeigt die sphärischen Radialfunktionen für ausgewählte Ordnungen n.

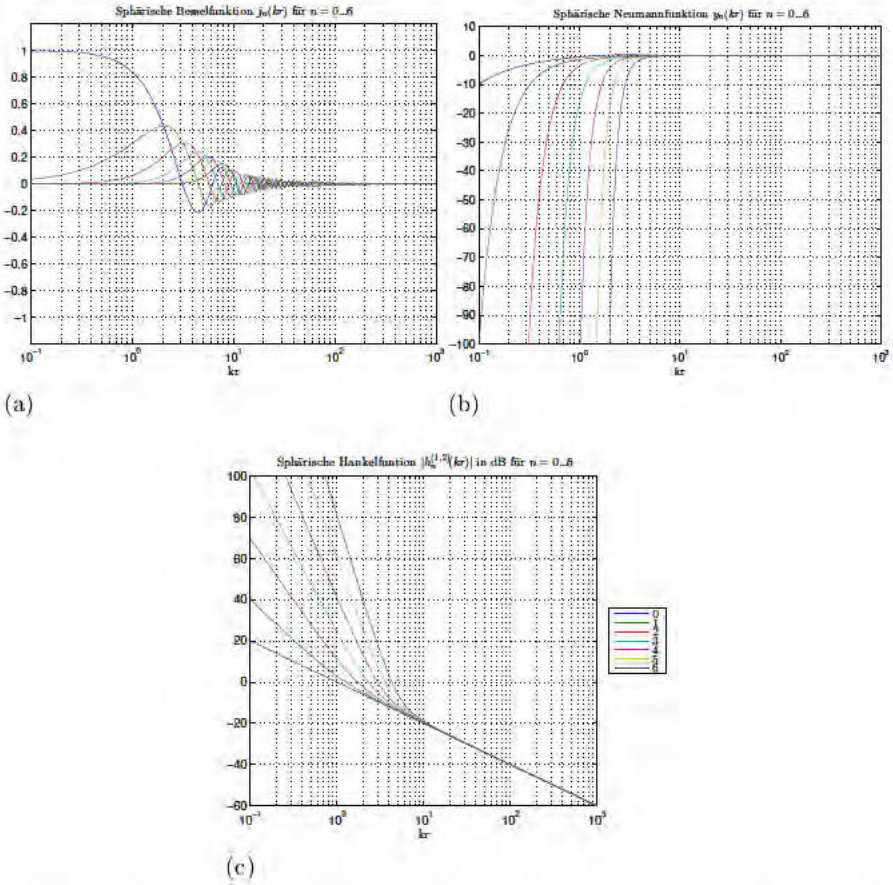


Abbildung 2.3.: Darstellung der sphärischen Bessel- (a), Neumann (b) und Hankelfunktion (c) der Ordnungen $n = 0...6$

Sphärische Harmonische

Üblicherweise werden die winkelabhängigen Teillösungen unter dem Namen *Sphärische Harmonische* zusammengefasst. Dabei gibt es mehrere Konventionen bezüglich Vorzeichen und Normierung N_n^m. Im theoretischen Teil dieser Arbeit wird die Konvention nach [GD04, (2.1.59)] gewählt

$$Y_n^m(\theta,\varphi) = N_n^m \cdot f(\theta) \cdot f(\varphi) \tag{2.23}$$

$$= (-1)^m \sqrt{\frac{2n+1}{4\pi} \cdot \frac{(n-|m|)!}{(n+|m|)!}} \cdot P_n^{|m|}(\cos\theta) \cdot \mathrm{e}^{\mathrm{j}m\varphi} \quad . \tag{2.24}$$

Die praktische Implementierung folgt wegen der Nutzung einer Software-Bibliothek einer anderen Konvention, was im Abschnitt 4.2.3 erläutert wird. Tabelle 2.2 enthält beispiel-

haft die sphärischen Harmonischen bis zur Ordnung $n = 2$. Diese Konvention in (2.24)

$Y_0^0(\theta,\varphi)$	$= \frac{1}{\sqrt{4\pi}}$	$Y_2^{-2}(\theta,\varphi)$	$= \mathrm{e}^{-2\mathrm{j}\varphi}\sqrt{\frac{15}{32\pi}}\sin^2\theta$
$Y_1^{-1}(\theta,\varphi)$	$= \mathrm{e}^{-\mathrm{j}\varphi}\sqrt{\frac{3}{8\pi}}\sin\theta$	$Y_2^{-1}(\theta,\varphi)$	$= \mathrm{e}^{-\mathrm{j}\varphi}\sqrt{\frac{15}{32\pi}}\sin 2\theta$
$Y_1^0(\theta,\varphi)$	$= \sqrt{\frac{3}{4\pi}}\cos\theta$	$Y_2^0(\theta,\varphi)$	$= \sqrt{\frac{5}{16\pi}}(-1+3\cos^2\theta)$
$Y_1^1(\theta,\varphi)$	$= -\mathrm{e}^{\mathrm{j}\varphi}\sqrt{\frac{3}{8\pi}}\sin\theta$	$Y_2^1(\theta,\varphi)$	$= -\mathrm{e}^{\mathrm{j}\varphi}\sqrt{\frac{15}{32\pi}}\sin 2\theta$
		$Y_2^2(\theta,\varphi)$	$= \mathrm{e}^{2\mathrm{j}\varphi}\sqrt{\frac{15}{32\pi}}\sin^2\theta$

Tabelle 2.2.: Lösungen der Sphärischen Harmonischen bis zur Ordnung $n = 2$

erlaubt die einfache Darstellung der konjugiert komplexen sphärischen Harmonischen

$$Y_n^m(\theta,\varphi)^* = Y_n^{-m}(\theta,\varphi) \quad . \tag{2.25}$$

Die sphärischen Harmonischen sind dabei orthonormal. Das heißt, es gilt nach [DLMF, (14.30.8)]

$$\int_{\varphi=0}^{2\pi}\int_{\theta=0}^{\pi} Y_n^m(\theta,\varphi)Y_{n'}^{m'}(\theta,\varphi)^*\sin\theta\ \mathrm{d}\theta\mathrm{d}\varphi = \delta_{nn'}\delta_{mm'} \quad , \tag{2.26}$$

wobei $\delta_{nn'}$ das Kronecker-Delta darstellt

$$\delta_{nn'} = \begin{cases} 1 & \text{für} \quad n = n' \\ 0 & \text{für} \quad n \neq n' \end{cases} \quad . \tag{2.27}$$

In Abbildung 2.4 sind einige sphärische Harmonische für ausgewählte Werte n und m grafisch dargestellt.

2.4. Inneres vs. äußeres Problem

Bei der Beschreibung von Schallfeldern muss, wie wir später sehen werden, in zwei Betrachtungen unterschieden werden: dem *inneren* und dem *äußeren* Problem. Ein inneres Problem liegt vor, wenn der zu betrachtende Bereich $r \leq r_0$ innerhalb einer Kugel mit dem Radius r_0 liegt und sämtliche Schallquellen außerhalb davon. Bei äußeren Problemen wiederum wird ein Bereich betrachtet, der außerhalb jeder Schallquelle liegt. Es existieren auch Probleme, bei denen etwa ein quellfreier Ring betrachtet wird, und die eine Kombination von innerem und äußerem Problem darstellen. Dies wird in dieser Arbeit aber nicht weiter verfolgt. Der Bereich unter Betrachtung muss dabei in jedem Fall frei von Schallquellen sein, um der homogenen Helmholtzgleichung (2.8) zu entsprechen. Eine Illustration der beiden Probleme ist in Abbildung 2.5 dargestellt. Zur Herleitung der Theorie wird in diesem Kapitel stets diese Fallunterscheidung angegeben. Ab Kapitel 3 wird wegen der praktischen Ausrichtung dieser Arbeit dann aber nur noch äußere Probleme behandelt (es sei denn es wird explizit darauf hingewiesen).

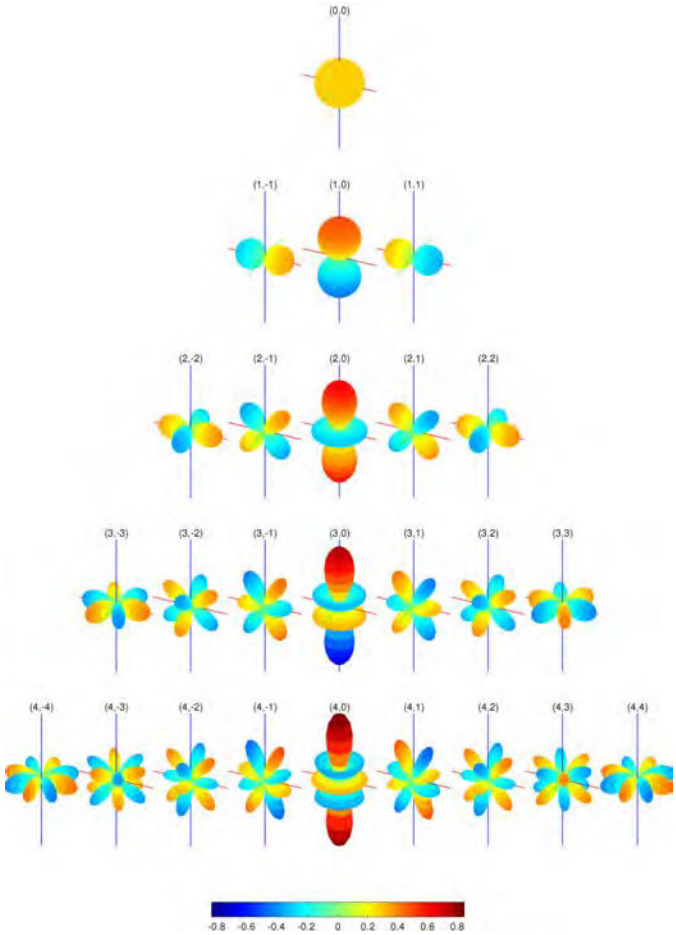


Abbildung 2.4.: Sphärische Harmonische $|\Re\{Y_n^m(\theta,\varphi)\}|$ für $n = 0\ldots4$.

2.5. Spherical Harmonic Transform $\mathcal{SHT}$

Die sphärischen Harmonischen Y_n^m liefern einen kompletten Lösungsansatz für die Helmholtz-Gleichung (2.8) in Kugelkoordinaten. Jedes Schallfeld $P(\mathbf{x},\omega)$ auf einer Kugeloberfläche kann also mit diesem Ansatz und den entsprechenden Koeffizienten beschrieben werden

$$P(\mathbf{x},\omega) = P(r,\theta,\varphi,\omega) = \sum_{n=0}^{\infty}\sum_{m=-n}^{+n} \mathring{P}_n^m(r,\omega)Y_n^m(\theta,\varphi) \quad . \tag{2.28}$$

Die dazugehörige Analyseformel kann dank der Orthonormalitätsrelation aus (2.26) durch Multiplikation mit der komplexen Konjugierten $Y_n^m(\theta,\varphi)^*$ abgeleitet werden ([Zot09, (28)])

$$\mathring{P}_n^m(r,\omega) = \int_{\varphi=0}^{2\pi}\int_{\theta=0}^{\pi} P(r,\theta,\varphi,\omega)Y_n^m(\theta,\varphi)^* \sin\theta \; \mathrm{d}\theta\mathrm{d}\varphi \quad . \tag{2.29}$$

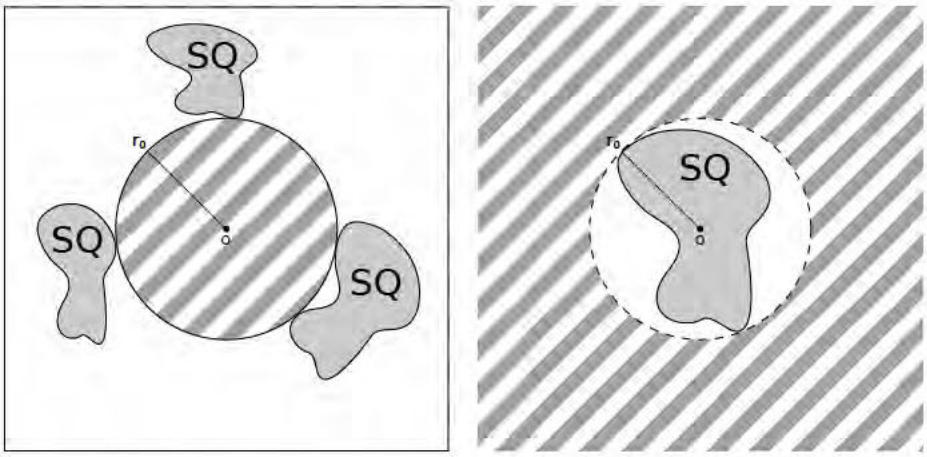


Abbildung 2.5.: Beispiele für innere und äußere Probleme. **SQ** steht für Schallquelle, der schraffierte Bereich für den gültigen Bereich unter Betrachtung.

Diese unitäre Abbildung wird *Spherical Harmonic Transform* ($\mathcal{SHT}$) genannt. Es lässt sich also verlustlos zwischen beiden Signaldarstellungen hin- und hertransformieren

$$\mathring{P}_n^m(r,\omega) = \mathcal{SHT}\left\{P(r,\theta,\varphi,\omega)\right\} \tag{2.30}$$

$$P(r,\theta,\varphi,\omega) = \mathcal{SHT}^{-1}\left\{\mathring{P}_n^m(r,\omega)\right\} \quad . \tag{2.31}$$

Bei den Koeffizienten $\mathring{P}_n^m(r,\omega)$ muss nun zwischen inneren und äußeren Problemen unterschieden werden, was sich durch die Wahl der radialen Eigenfunktion bemerkbar macht (siehe Tabelle 2.1). In [Zot09, Kap. 2.1.1] wird gezeigt, dass jedes der genannten Probleme nur eine einzige Eigenfunktion für den radialen Term zulässt. Diese ist für das innere Problem die sphärische Besselfunktion 1. Art, also

$$P_i(\mathrm{x},\omega) = \sum_{n=0}^{\infty}\sum_{m=-n}^{+n} \underbrace{\breve{P}_{n,i}^m(\omega) j_n\left(\frac{\omega}{c}r\right)}_{=\mathring{P}_{n,i}^m(r,\omega)} Y_n^m(\theta,\varphi) \tag{2.32}$$

und für das äußere Problem und der in dieser Arbeit verwendeten Zeitkonvention die sphärische Hankelfunktion 2. Art, also

$$P_e(\mathrm{x},\omega) = \sum_{n=0}^{\infty}\sum_{m=-n}^{+n} \underbrace{\breve{P}_{n,e}^m(\omega) h_n^{(2)}\left(\frac{\omega}{c}r\right)}_{=\mathring{P}_{n,e}^m(r,\omega)} Y_n^m(\theta,\varphi) \quad . \tag{2.33}$$

Die Koeffizienten $\breve{P}_n^m$ bilden das diskrete Modalspektrum (in der Literatur findet man hierfür u.a. auch die Bezeichnungen b_{nm} und c_{nm} [Wil99, Zot09], die hier verwendete Nomenklatur nach [Ahr12] ist allerdings für spätere Anwendungen sinnvoll). Deren Ana-

lyse erfolgt durch die modale Zerlegung des Schallfeldes mit Hilfe von (2.29) für innere Probleme

$$\breve{P}_{n,i}^{m}(\omega) = \frac{\mathring{P}_{n,i}^{m}(r,\omega)}{j_n(\frac{\omega}{c}r)} \tag{2.34}$$

$$= \frac{1}{j_n(\frac{\omega}{c}r)} \int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\pi} P(r,\theta,\varphi,\omega) Y_n^m(\theta,\varphi)^* \sin\theta \, \mathrm{d}\theta \mathrm{d}\varphi \tag{2.35}$$

und für äußere Probleme

$$\breve{P}_{n,e}^{m}(\omega) = \frac{\mathring{P}_{n,e}^{m}(r,\omega)}{h_n^{(2)}(\frac{\omega}{c}r)} \tag{2.36}$$

$$= \frac{1}{h_n^{(2)}(\frac{\omega}{c}r)} \int_{\varphi=0}^{2\pi} \int_{\theta=0}^{\pi} P(r,\theta,\varphi,\omega) Y_n^m(\theta,\varphi)^* \sin\theta \, \mathrm{d}\theta \mathrm{d}\varphi \quad . \tag{2.37}$$

2.6. Plane Wave Decomposition

Eine weitere Möglichkeit ein Schallfeld zu beschreiben bietet die *plane wave decomposition* ($\mathcal{PWD}$). Der Ansatz dabei ist, ein Schallfeld auf einer abgetasteten Kugeloberfläche als Summe einzelner ebenen Wellen aus unterschiedlichen Richtungen aufzufassen. Zur Berechnung der Koeffizienten der $\mathcal{PWD}$ wird nun das Spektrum $\mathring{P}_n^m$ mit einem Radialfilter d_n multipliziert, um die radialen und Normierungsterme zu eliminieren. Bei Messungen mit einem offenen Mikrofonarray von Druckempfängern (open sphere, pressure transducers) ist dieses Radialfilter

$$d_n(r,\omega) = \frac{1}{4\pi \mathrm{j}^n \cdot j_n(\frac{\omega}{c}r)} \quad . \tag{2.38}$$

Für andere Array-Konfigurationen werden andere Radialfilter eingesetzt (vgl. [BPSW11]). Das Spektrum der $\mathcal{PWD}$ kann für die genannte Array-Konfiguration also mit

$$\mathring{P}_n^m(r,\omega)_{PWD} = \mathring{P}_n^m(r,\omega) \cdot b_n(r,\omega) \tag{2.39}$$

$$= \frac{\mathring{P}_n^m(r,\omega)}{4\pi \mathrm{j}^n \cdot j_n(\frac{\omega}{c}r)} \tag{2.40}$$

beschrieben werden. Dieses Verfahren wird am Beispiel eines Schallfeldes mit einer einzigen ebenen Welle aus der Richtung k_{pw} kommend deutlich. Wir notieren dazu zunächst die sphärische harmonische Darstellung einer ebenen Welle entsprechend [Ahr12, (2.38)]

$$\mathrm{e}^{-\mathrm{j}\mathbf{k}_{pw}\mathbf{x}} = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \underbrace{4\pi \mathrm{j}^{-n} \; Y_n^m(\theta_{pw},\varphi_{pw})^* \; j_n\left(\frac{\omega}{c}r\right)}_{\mathring{P}_n^m(r,\omega)} Y_n^m(\theta,\varphi) \quad . \tag{2.41}$$

Wenn nun das Spektrum der $\mathcal{SHT}$ einer ebenen Welle aus (2.41) in (2.40) eingesetzt wird, erhalten wir das Spektrum der $\mathcal{PWD}$

$$\mathring{P}_n^m(r,\omega)_{PWD} = Y_n^m(\theta_{pw},\varphi_{pw})^* \quad . \tag{2.42}$$

Die Koeffizienten aus (2.42) eingesetzt in die Synthesegleichung (2.28)

$$P(\mathbf{x},\omega) = P(r,\theta,\varphi,\omega) = \sum_{n=0}^{\infty}\sum_{m=-n}^{+n} Y_n^m(\theta_{pw},\varphi_{pw})^* Y_n^m(\theta,\varphi) \tag{2.43}$$

zeigt, dass mit der Abgeschlossenheitsrelation [Ahr12, (2.27)]

$$\sum_{n=0}^{\infty}\sum_{m=0-n}^{n} Y_n^m(\theta,\varphi) Y_n^m(\theta',\varphi')^* = \delta(\varphi-\varphi')\delta(\theta-\theta') \tag{2.44}$$

nur eben die Einfallsrichtung des Wellenvektors k_{pw} ausgetastet wird.

2.7. Darstellung im Wellenzahlenraum

Bei der späteren Analyse der Lautsprecher-Richtcharakteristik werden Schallfelder vorwiegend im Wellenzahlenraum $\tilde{P}(\cdot)$ betrachtet. Der Grund dafür hängt mit der eben vorgestellten Plane Wave Decomposition zusammen, also mit Beschreibung eines Schallfeldes durch die Summation ebener Wellen in sämtliche Richtungen. Um die einzelnen ebenen Wellen nun voneinander zu separieren wird die spatiale Fouriertransformation in (2.4) genutzt. Dabei können die Dimensionen einzeln betrachtet werden, d.h. die Transformation des Schallfeldes exemplarisch in z-Richtung ist mit $\tilde{P}(x,y,k_z)$ gegeben. Die visuelle Darstellung gelingt dabei anschaulich mit einem Plot in Abhängigkeit von Frequenz f und Wellenzahl k_z, der beispielhaft anhand von Abbildung 2.6(a) erläutert werden soll.

Die grauen Linien kennzeichnen $|k_z| = \frac{\omega}{c}$ und somit entsprechend der Dispersionrelation (2.13) die Grenze zwischen propagierender und evaneszenter Schallausbreitung, wobei propagierende, ebene Wellen innerhalb dieses Dreiecks liegen. Eine einzelne, monochromatische ebene Welle ist mit dem Kreuz und $\theta_{pw,1}$ gekennzeichnet. Es wird für die entsprechende Frequenz also nur ein einziger Wert k_z ausgetastet. Die Richtung dieser ebenen Welle lässt sich durch die in (2.1) gegebene Umrechnung von kartesischen in sphärische Koordinaten mit $k_z = \frac{\omega}{c}\cos\theta_{pw}$ ableiten. Eine einzelne, breitbandige ebene Welle ist mit der Geraden $\theta_{pw,2}$ gekennzeichnet. Der Winkelbereich zwischen den Grenzen $|k_z| = \frac{\omega}{c}$ deckt dabei den vollen Elevationswinel $\theta_{pw} = 0...180°$ ab, wobei eine ebene, horizontale Welle mit $\theta_{pw} = 90°$ in der Abbildung auf der Zeit-Frequenzachse liegt. Diese Betrachtung gilt

Codiert man nun die Amplituden aller ebenen Wellen in der zu betrachtenden Dimension farbig, erhält man eine anschauliche Darstellung des zu betrachtenden Schallfeldes. Dies

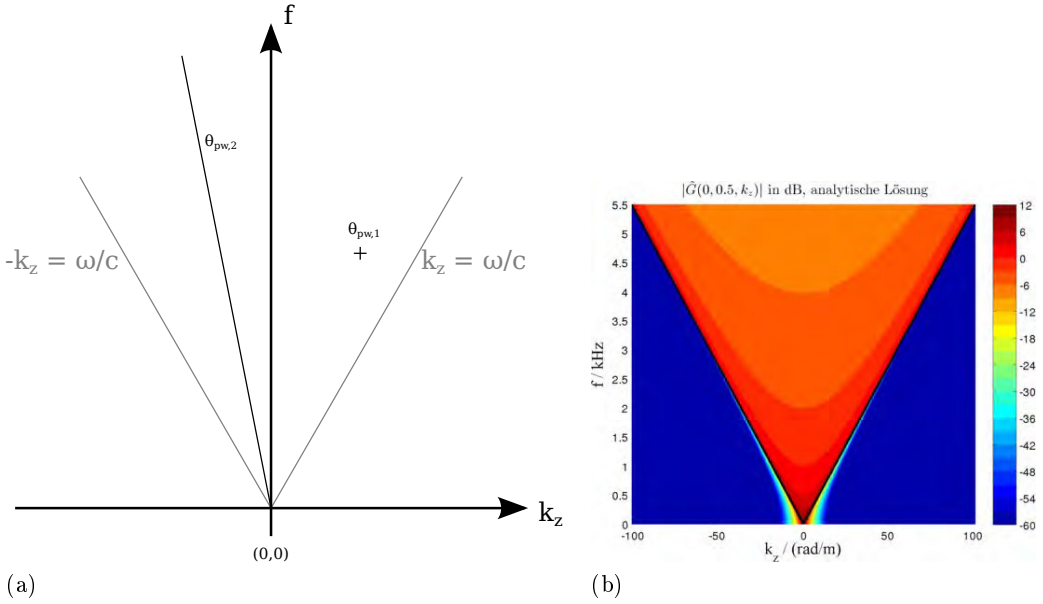


Abbildung 2.6.: (a) Schematische Darstellung von ebenen Wellen in einem k_z-f-Plot, Quelle: [Ahr12, Abb. 2.14]; (b) $|\tilde{G}(0, 0.5, k_z)|$ in dB eines Monopols im Ursprung, normalisiert auf $f = 1000Hz$

sei exemplarisch in Abbildung 2.6(b) für einen Monopol im Ursprung gezeigt. Ein weiterer Vorteil dieser Darstellungsart ist die Visualisierung evaneszenter Anteile, die hier deutlich für Frequenzen $f < 2$kHz außerhalb des Dreiecks $|k_z| = \frac{\omega}{c}$ zu sehen sind.

Für diese Art der Darstellung ist es sicherlich von Nutzen, wenn das Schallfeld auf einer Linie parallel zur entsprechenden Achse vorliegt, wie es wieder exemplarisch für die z-Richtung in Abbildung 2.7 zu sehen ist. Dies ist leicht zu erreichen, wenn das Modalspektrum $\breve{P}_n^m(\omega)$ für die zu untersuchende Schallquelle vorliegt. Mittels der Synthesegleichung der $\mathcal{SHT}^{-1}$ (2.28) kann dieses auf die entsprechende Linie $\mathbf{L_z} = (x_{ref}, y_{ref}, z)$ expandiert werden. Dabei ist lediglich darauf zu achten, dass der Gültigkeitsbereich für äußere Probleme $r > r_0$ eingehalten wird.

2.8. Schallfeldsynthese mittels Single Layer Potential

Im Bereich der virtuellen Akustik gibt es bereits weitreichende Forschungs- und Anwendungsvorhaben für die Schallfeldsynthese mit einer Sekundärquellenverteilung (SSD). Sekundärquellen können reale Schallerzeuger sein, die es ermöglichen virtuelle Primärquellen zu synthetisieren. Die Schallfeldsynthese ist denn auch ein praktischer Anwendungsfall der Ergebnisse dieser Arbeit, da sie nur mit entsprechendem Wissen über die Sekundärquellen (sprich: Lautsprecher) gelingen kann.
Eine sehr grundlegende Methode mit einer kugelförmigen SSD soll an dieser Stelle diskutiert werden. Dabei werden als Sekundärquellen Kugelmonopole angenommen, was

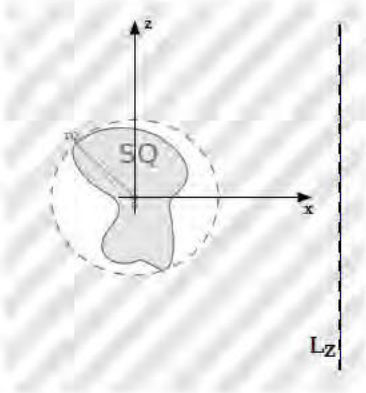


Abbildung 2.7.: Schematische Darstellung der Linie $\mathbf{L_z}$, auf der der Schalldruck bekannt sein muss.

der Methode auch ihren Namen gibt: das *Single Potential Layer* der Form entsprechend [Ahr12, (3.1)]

$$P(\mathbf{x},\omega) = \int\limits_{S_0(\mathbf{x_0})} D(\mathbf{x_0},\omega) \cdot G(||\mathbf{x}-\mathbf{x_0}||,\omega)\mathrm{d}S_0(\mathbf{x_0}) \quad . \tag{2.45}$$

Das Schallfeld $P(\mathbf{x},\omega)$ wird durch Integration von Sekundärquellen $G(\cdot)$ auf der Kugeloberfläche S_0 mit dem Radius r_0 synthetisiert. Die explizite Form des Integrals ist in diesem Fall entsprechend den geometrischen Konventionen aus (2.1)

$$\int\limits_{S_0(\mathbf{x_0})} = \int_0^{2\pi}\int_0^{\pi} r_0^2 \sin\theta_0 \mathrm{d}\theta_0 \mathrm{d}\varphi_0 \tag{2.46}$$

(vgl. [Pol05, (45)]). Jede Sekundärquelle wird dabei von einer Treiberfunktion $D(\cdot)$, also einem komplexen Filterspektrum angesteuert, das frequenzabhängige Amplitude, Phase und Delay enthält. Zunächst sei der einfachste Fall dargestellt, bei dem das Verhalten der Sekundärquellen $G(\cdot)$ bekannt ist und diese translations- und rotationsinvariant sind. Denn dann ließe sich eine explizite Lösung der Treiberfunktion durch entsprechende Parametrierung an das jeweilige Problem anpassen. Für die Analyse der Treiberfunktion wird dazu (2.46) als Faltung entlang der Kugeloberfläche interpretiert und kann dem Faltungs-Theorem entsprechend durch die $\mathcal{SHT}$ mit

$$\mathring{P}_n^m(r,\omega) = r_0^2\sqrt{\frac{4\pi}{2n+1}}\mathring{D}_n^m(\omega) \cdot \mathring{G}_n^0(r,\omega) \tag{2.47}$$

beschrieben werden ([Ahr12, (4.16)]). Es sei bemerkt, dass die angegebene Literaturstelle noch den Faktor 2π führt, der bei einer (noch nicht veröffentlichten) Revision aber wegfällt. Durch Anwendung von (2.34) und nach der Umstellung nach $\mathring{D}_n^m(\omega)$ erhalten

wir

$$\mathring{D}_n^m(\omega) = \frac{1}{r_0^2}\sqrt{\frac{2n+1}{4\pi}}\frac{\breve{P}_{n,i}^m(\omega)j_n(\frac{\omega}{c}r)}{\breve{G}_n^0(\omega)j_n(\frac{\omega}{c}r)} \quad . \tag{2.48}$$

Die Division der sphärischen Besselfunktionen enthält abhängig von der Ordnuing n für $j_n(\frac{\omega}{c}r) = 0$ und $\frac{\omega}{c}r \neq 0$ Unstetigkeiten, die nicht mit der Regel nach l'Hospital behoben werden können und so mehrdeutige Lösungen von (2.48) produzieren. Diese Fälle müssen in der Praxis sensibel behandelt werden. In [FN12, Kap. 4.3] werden für diese Fälle *CHIEF points* eingesetzt. Dies sind zusätzliche Punkte im Raum, an denen der Schalldruck bekannt ist und mit deren Hilfe die Mehrdeutigkeit behoben werden kann. Konkret wird hierzu nun der Punkt $r = 0$ referenziert, womit die Eindeutigkeit wiederhergestellt wird und (2.48) zu

$$\mathring{D}_n^m(\omega) = \frac{1}{r_0^2}\sqrt{\frac{2n+1}{4\pi}}\frac{\breve{P}_{n,i}^m(\omega)}{\breve{G}_n^0(\omega)} \quad . \tag{2.49}$$

vereinfacht werden kann. Nach Anwendung der $\mathcal{SHT}$ aus (2.28) erhält man die Treiberfunktion für sphärische SSD in der Form

$$D(r_0, \varphi_0, \theta_0, \omega) = \frac{1}{r_0^2}\sum_{n=0}^{\infty}\sum_{m=-n}^{+n}\sqrt{\frac{2n+1}{4\pi}}\frac{\breve{P}_{n,i}^m(\omega)}{\breve{G}_n^(\omega)}Y_n^m(\varphi_0, \theta_0) \quad . \tag{2.50}$$

Diese Operation wird in der Literatur Mode-Matching genannt, und stellt die explizite Lösung der unbekannten Treiberfunktion $D(\mathbf{x_0}, \omega)$ des Single Layer Potentials dar([AS08, ZPN10, Faz10]).

3. Analytische Modelle

Nachdem im vorangegangenen Kapitel die mathematischen Grundlagen für die Analyse und Synthese von Schallfeldern gelegt wurden, sollen im folgenden Kapitel konkrete analytische Modelle für die reale modale Zerlegung von Lautsprecher-Messdaten vorgestellt und simuliert werden. Diese Modelle dienen als Referenzen für die spätere Analyse der Messdaten. Dabei werden auch Abweichungen diskutiert, die zwischen den Modellen und der realen Analyse zu erwarten sind. Diese Abweichungen entstehen insbesondere durch Diskretisierung und Begrenzung kontinuierlicher Funktionen. Darunter fallen etwa die Abtastung einer Kugel oder die Begrenzung der Ordnung der sphärischen Harmonischen Transformation.

3.1. Fernfeldbedingungen

Die im Folgenden behandelten Themen sehen oft Näherungen im Fernfeld vor. Diese Näherungen gelten nur, wenn bestimmte Bedingungen bezüglich Quellgröße (3.1a), Phasenlage (3.1b) und Quellimpedanz (3.1c) vorliegen. Diese drei Kriterien stellen die Fernfeldbedingungen in Abhängigkeit zum Abstand r zwischen Mittelpunkt der Schallquelle und zu betrachtenden Aufpunkt

$$r \gg l \tag{3.1a}$$

$$r \gg \frac{l^2}{\lambda} \tag{3.1b}$$

$$r \gg \lambda \quad . \tag{3.1c}$$

Mit l wird die maximale Ausdehnung der Schallquelle beschrieben, während λ die Wellenlänge bezeichnet. Die entsprechenden Herleitungen können aus [MÖ9], [Wei08] entnommen werden. Gln.(3.1b, 3.1c) stellen dabei eine Bandbegrenzung mit oberer und unterer Grenzfrequenz dar, in der bei gegebenem Abstand r Fernfeldbedingungen herrschen.

3.2. Analytische und numerische Lösungen grundlegender Schallquellen

3.2.1. Kugelmonopol

Der ideale Kugelmonopol ist die die grundlegenste Elementar-Schallquelle. Sie besteht aus einer (infinitesimal) kleinen Punktquelle, die Schall frequenz- und richtungsunabhängig abstrahlt. Das durch einen Kugelmonopol an der Stelle $\mathbf{x_0}$ erzeugte Schallfeld P wird mit

$$P(r,\omega) = A \cdot G(||\mathbf{x}-\mathbf{x_0}||,\omega) \quad \text{mit} \quad r = ||\mathbf{x}-\mathbf{x_0}|| \tag{3.2}$$

und der komplexen Amplitude

$$A = \mathrm{j}\omega\rho_0 Q$$

beschrieben. Die Amplitude ist dabei maßgeblich vom Volumenfluss $Q = \int_S V \mathrm{d}S$ abhängig, also der integrierten Schnelle über die emittierende Fläche. Komplexere Schallquellen können immer auch als Summe unendlich vieler Kugelmonopole dargestellt werden, weswegen dieser Elementar-Schallquelle besondere Bedeutung zu kommt. Die verwendete *Green*'sche Funktion im Freifeld, die auch schon im Abschnitt 2.8 benutzt wurde, wird mit

$$G(||\mathbf{x}-\mathbf{x_0}||,\omega) = \frac{\mathrm{e}^{-\mathrm{j}\omega/c||\mathbf{x}-\mathbf{x_0}||}}{4\pi||\mathbf{x}-\mathbf{x_0}||} \tag{3.3}$$

beschrieben. Es existieren für einen Kugelmonopol an der Stelle $\mathbf{x_0}$ bei Betrachtung eines äußeren Problems die analytischen Koeffizienten (vgl. [Wil99, Gl.(8.22)])

$$\breve{G}_n^m(\omega) = -\mathrm{j}\frac{\omega}{c} j_n(\frac{\omega}{c} r_0) Y_n^m(\theta_0,\varphi_0)^* \quad . \tag{3.4}$$

Hieraus ist zu erkennen, dass ein Kugelmonopol im Ursprung gelegen lediglich den Koeffizienten für $n = 0, m = 0$ anregt, da die Besselfunktion $j_n(0) = 0$ für $n \neq 0$ ist. Auch für die räumliche Fouriertransformation der Green'schen Funktion in x-Richtung gibt es mit

$$\tilde{G}(k_x,y,z,\omega) = \begin{cases} -\frac{\mathrm{j}}{4} H_0^{(2)}\left(\sqrt{(\frac{\omega}{c})^2 - k_x^2}\sqrt{|y|^2+z^2}\right) & \text{für } |k_x| < |\frac{\omega}{c}| \\ \frac{1}{2\pi} K_0\left(\sqrt{k_x^2 - (\frac{\omega}{c})^2}\sqrt{|y|^2+z^2}\right) & \text{für } |k_x| > |\frac{\omega}{c}| \end{cases} \tag{3.5}$$

(vgl. [AS10b, Gl.(52)]) einen analytischen Ausdruck. $H_0^{(2)}$ ist die Hankelfunktion 2.Art, nullter Ordnung ([DLMF, (10.2.6)]), und K_0 die modifizierte Besselfunktion 2.Art, nullter Ordnung ([DLMF, (10.25.3)]).

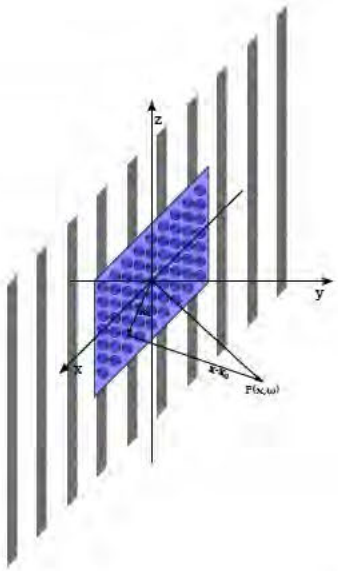


Abbildung 3.1.: Geometrische Zusammenhänge des ersten Rayleigh-Integral mit der Bezugsebene $y_0 = 0$

3.2.2. Rechteckige Kolbenmembran

Zur Beschreibung der Schallabstrahlung von Ebenen in einen quellfreien Halbraum ist das Rayleigh-Integral ein wichtiges Werkzeug der Akustik. Damit besteht die Möglichkeit ein Schallfeld $p(\mathbf{x})$ zu beschreiben, wenn das Schnellefeld $v(\mathbf{x_0})$ einer Bezugsebene gegeben ist. Dabei wird die Normalenschnelle über die Bezugsebene integriert. Abbildung 3.1 zeigt dazu die geometrische Anordnung für die Bezugsebene $y_0 = 0$. Das Schallfeld im Halbraum $|y| > y_0$ kann nun mit dem Rayleigh-Integral

$$P(\mathbf{x}, \omega) = \frac{\mathrm{j}\omega\rho_0}{2\pi} \iint\limits_{-\infty}^{\infty} V(x_0, 0, z_0, \omega) \frac{e^{-\mathrm{j}\omega/c||\mathbf{x}-\mathbf{x_0}||}}{||\mathbf{x}-\mathbf{x_0}||} \, \mathrm{d}x_0 \mathrm{d}z_0 \tag{3.6}$$

([MÖ9, Gl.(3.61)], [Wil99, Gl.(2.75)]) beschrieben werden, wobei $V(\mathbf{x_0})$ die Schnelle in Normalenrichtung zur Bezugsebene ist und der Abstand zwischen dem Aufpunkt des Schallfeldes und des jeweiligen Element $\mathrm{d}x_0\mathrm{d}z_0$ mit

$$||\mathbf{x}-\mathbf{x_0}|| = \sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2}$$

berechnet wird. Streng genommen gilt das Rayleigh-Integral nur für unendliche Ebenen $\mathbf{x_0}$. Bei endlichen Ebenen muss davon ausgegangen werden, dass die Bezugsebene von einer starren, schallharten Wand eingeschlossen ist. Diese Einschränkung kann umgangen werden, indem ausschließlich Frequenzen mit relativ zur Bezugsebene kleinen Wellenlängen betrachtet werden [MÖ9, S.113]. Im Fernfeld kann dieses Integral mit Hilfe der

spatialen Fouriertransformation zu

$$P(\mathbf{x},\omega) = 2\mathrm{j}\omega\rho_0\ G(||\mathbf{x}-\mathbf{x_0}||,\omega)\ \tilde{V}(k_x,k_z) \tag{3.7}$$

$$= \frac{\mathrm{j}\omega\rho_0}{2\pi}\ \frac{\mathrm{e}^{-\mathrm{j}\omega/c||\mathbf{x}-\mathbf{x_0}||}}{||\mathbf{x}-\mathbf{x_0}||}\ \tilde{V}(k_x,k_z) \tag{3.8}$$

umgeformt werden (Herleitung in [Wil99, Kap.2.11]). Normiert auf den Volumenfluss und unabhängig vom Radialterm kann die dimensionslose Richtcharakteristik D im Fernfeld mit der Beziehung

$$P(\mathbf{x},\omega) = 2\mathrm{j}\omega\rho_0 Q\ G(||\mathbf{x}-\mathbf{x_0}||,\omega)\ D(\varphi,\theta)$$

mit

$$D(\varphi,\theta) = \frac{\tilde{V}(k_x,k_z)}{Q} \tag{3.9}$$

ausgedrückt werden ([SRS14, Kap.4]).
Auf Basis des Rayleigh-Integrals können wir nun die Schallausbreitung endlicher Ebenen beschreiben. Dazu soll an dieser Stelle das Schallfeld unter der Annahme betrachtet werden, dass die schallabstrahlende endliche Ebene mit einer darauf ortsunabhängigen Schallschnelle in eine unendliche schallharte Wand eingebettet ist, es gilt also

$$V(\mathbf{x_0},\omega) = \begin{cases} V_0(\omega) & \text{für } -l_x/2 \le x \le l_x/2, \\ & \quad y = 0 \\ & \quad -l_z/2 \le z \le l_z/2 \\ 0 & \text{sonst.} \end{cases} \tag{3.10}$$

Gl.(3.10) stellt eine Rechteckfunktion dar, woraus sich mit $Q = V_0 l_x l_z$ die spatiale Fouriertransformation

$$\tilde{V}(k_x,k_z) = Q\ \mathrm{sinc}(k_x\ l_x/2)\ \mathrm{sinc}(k_z\ l_z/2)$$

$$= Q\ \frac{\sin(\frac{l_x}{2}k\cos\varphi\sin\theta)}{\frac{l_x}{2}k\cos\varphi\sin\theta}\ \frac{\sin(\frac{l_z}{2}k\cos\theta)}{\frac{l_z}{2}k\cos\theta} \tag{3.11}$$

ergibt und eingesetzt in (3.9) die Fernfeld-Directivity

$$D_{rect}(\varphi,\theta) = \mathrm{sinc}(k_x\ l_x/2)\ \mathrm{sinc}(k_z\ l_z/2) \quad . \tag{3.12}$$

Abbildung 3.2 zeigt die Directivity als Balloon-Plot für ausgewählte Frequenzen. Die Abmessungen der schallabstrahlenden Ebene sind mit $l_z = 0.36$m und $l_x = l_z/2$ so gewählt, dass im späteren Vergleich mit realen Messdaten eine geeignete Referenz besteht. Die

erkennbaren Einschnitte liegen in den Nullstellen der sinc-Funktion. Der mathematische Zusammenhang ist allgemein

$$\mathrm{sinc}(n\pi) = \frac{\sin(n\pi)}{\mu\pi} = 0 \quad \text{für} \quad |\mu| \in \{1, 2, 3, \ldots\} \tag{3.13}$$

und im konkreten Fall in (3.12)

$$\mathrm{sinc}(k_{\mathrm{dim}} l_{\mathrm{dim}}/2) = 0 \quad \text{für} \quad k_{\mathrm{dim}} = \mu \frac{2\pi}{l_{\mathrm{dim}}} \quad \text{und} \quad |\mu| \in \{1, 2, 3, \ldots\} \quad . \tag{3.14}$$

Abbildung 3.3 zeigt wiederum die horizontale ($\theta = \pi/2$) und vertikale ($\varphi = -\pi/2$) Richtcharakteristik in Abhängigkeit von der Frequenz mit den in (3.14) beschriebenen Einschnitten (Nullstellen).

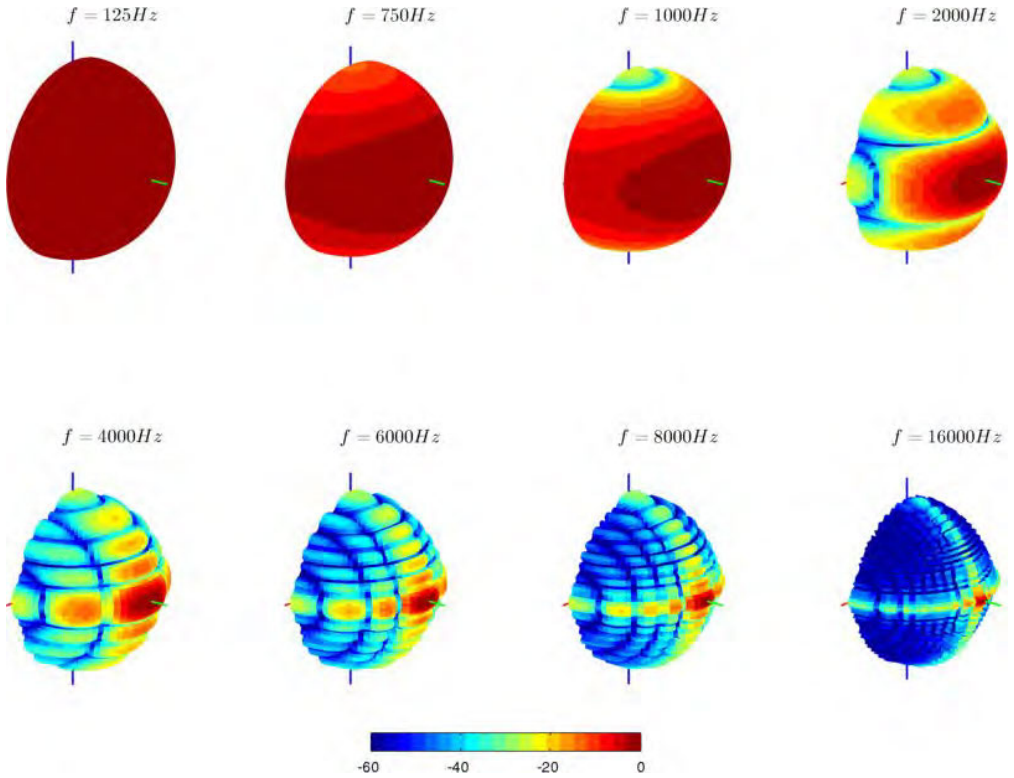
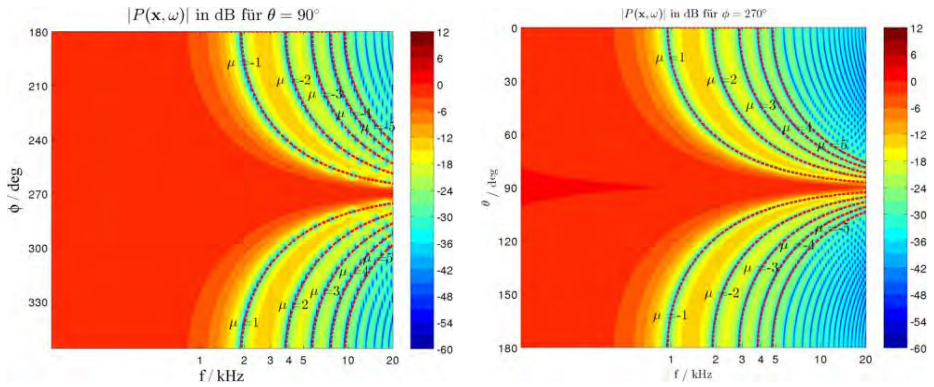


Abbildung 3.2.: Fernfeld-Directivity in dB als Balloon-Plot für eine rechteckige Kolbenmembran, die entsprechend Abb. 3.1 schallhart in der xz-Ebene liegt, mit den Abmessungen $l_z = 0.36$m und $l_x = l_z/2$ im Halbraum $y < y_0$



(a) (b)

Abbildung 3.3.: Horizontale (a) und Vertikale Directivity (b) einer rechteckigen Kolbenmembran mit eingezeichneten Nullstellen (rot)

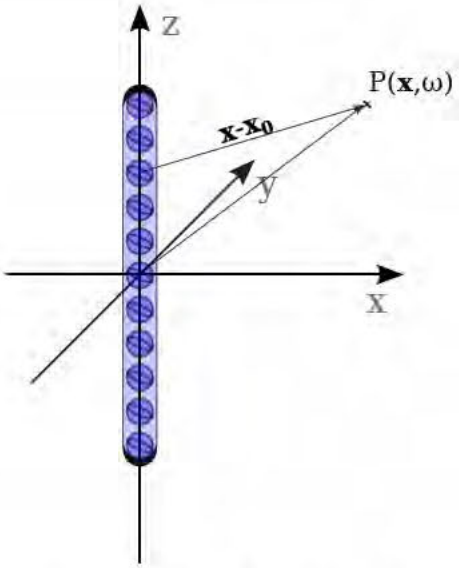


Abbildung 3.4.: Schematische Darstellung einer Linienquelle

3.2.3. Linienquelle

Interpretiert man (3.6) für eine eindimensionale Linienquelle entsprechend Abb. 3.4, also für $l_x \to 0$, kann man ebenfalls entsprechende Aussagen über Schalldruckentwicklung und Directivity im Fernfeld aufstellen. Eine Linienquelle entlang der z-Achse mit konstanter Schnelle (vgl. (3.10)) wird nun mit

$$P(\mathbf{x},\omega) = \frac{\mathrm{j}\omega\rho_0 b}{4\pi} \int_{-l_z/2}^{l_z/2} V_0 \; G(||\mathbf{x}-\mathbf{x_0}||,\omega) dz_0 \tag{3.15}$$

beschrieben. Unter den bekannten Fernfeldbedingungen und entsprechend der Transformation (3.11) ist der Schalldruck

$$P(\mathbf{x},\omega) = \mathrm{j}\omega\rho_0 Q \; G(||\mathbf{x}-\mathbf{x_0}||,\omega) \cdot \frac{\sin(\frac{l_z}{2}k\cos\theta)}{\frac{l_z}{2}k\cos\theta} \tag{3.16}$$

$$= \mathrm{j}\omega\rho_0 Q \; G(||\mathbf{x}-\mathbf{x_0}||,\omega) \cdot \; \mathrm{sinc}(k_z \; l_z/2) \tag{3.17}$$

und die Directivity

$$D_{line}(\theta) = \mathrm{sinc}(k_z \; l_z/2) \quad . \tag{3.18}$$

Da eine Linienquelle ein zweidimensionales, rotationssymmetrisches Schallfeld erzeugt, ist die Directivity in dieser Lage lediglich von θ abhängig, wie in Abbildung 3.5 ersichtlich ist.

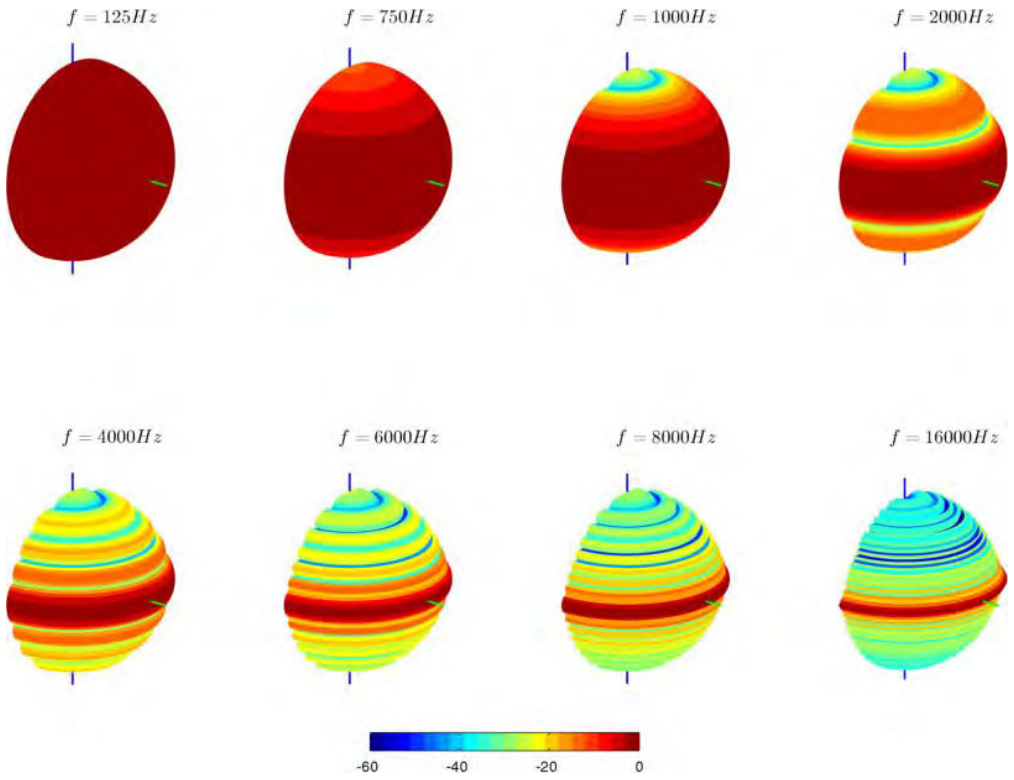


Abbildung 3.5.: Abstrahlcharakteristik einer Linienquelle der Länge $l_z = 0.36$m

3.3. Kugelabtastung

Der theoretische Ansatz der $\mathcal{SHT}$, beschrieben in Abschnitt 2.5, sieht eine kontinuierliche Integraldarstellung für die Beschreibung der Kugeloberfläche vor (vgl. Gl. (2.29)). Dies ist in der Praxis natürlich nicht möglich. Deshalb ist es an dieser Stelle wichtig, sich Gedanken über die Diskretisierung der Kugeloberfläche zu machen. Da die Anzahl der spatialen Stützstellen L direkt mit der maximal beschreibbaren Bandbreite N der $\mathcal{SHT}$ zusammenhängt, bedarf Wahl und Parametrisierung der Diskretisierung besonderer Sorgfalt. Eine ausführliche Diskussion der zur Verfügung stehenden Samplingstrategien kann in [Zot09, Kap. IV] gefunden werden.

Ein häufig genutztes Modell ist demnach *(weighted) quadrature*. Die Vorteile liegen in der algorithmischen Einfachheit, da eine (bei großen Datensätzen) rechenintensive Matrixinversion vermieden werden kann (siehe [Zot09, S.72f]). Zur *weighted quadrature* - Methode gehören u.a. die equiangulare Abtastung sowie Gauß- und Lebedev-Quadratur.

Equiangulare Abtastung

In [DH94] wird eine equiangulare Abtasttheorie für mit $n \leq N$ spatial begrenzte Funktionen vorgestellt. Darin wird die Kugeloberfläche horizontal und vertikal an jeweils

$N_\varphi = N_\theta = 2(N+1)$ Stellen abgetastet

$$\varphi_i = \frac{2\pi i}{2N+2} \quad , \quad i = 0, ..., 2N+1 \tag{3.19}$$

$$\theta_k = \frac{\pi k}{2N+2} \quad , \quad k = 0, ..., 2N+1 \quad . \tag{3.20}$$

Die Gesamtanzahl der Stützstellen beträgt also $L = 4(N+1)^2$. Dieses Layout bedingt, dass die Abtastpunkte zu den Polen hin örtlich dichter beieinander liegen, was durch Gewichtungsfaktoren a_k kompensiert werden muss [DH94, (5),(9)]

$$a_k = \frac{2\sqrt{2}}{N_\theta} \cdot \sin\left(\frac{\pi k}{N_\theta}\right) \cdot \sum_{l=0}^{N_\theta/2-1} \frac{1}{2l+1} \cdot \sin\left((2l+1)\frac{\pi k}{N_\theta}\right), \quad k = 0, ..., N_\theta - 1 \quad . \tag{3.21}$$

Die diskrete $\mathcal{SHT}$ für equiangulare Abtastung kann also wie folgt beschrieben werden ([Raf05, (9)])

$$\mathring{P}_n^m(r,\omega) = \sum_{i=0}^{2N+1} \sum_{k=0}^{2N+1} a_k P(\mathbf{x},\omega) Y_n^m(\theta_k, \varphi_i)^* \quad . \tag{3.22}$$

Die equiangulare Kugelabtastung hat den Nachteil, dass relativ viele Abtastpunkte für eine angestrebte Ordnung N nötig sind. Eine Alternative wird im folgenden Absatz beschrieben. Ebenfalls problematisch ist diese Strategie bei einer algorithmischen $\mathcal{SHT}$ in Matrixnotation bezüglich Kondition, Orthonormalität und Rekonstruktionsfehler, was näher u.a. in [Bru07, Kap. 4.4.1] beschrieben ist. In der konventionellen Lautsprechermesstechnik ist diese Abtaststrategie dennoch weit verbreitet. Das liegt vor allem am relativ übersichtlichen und technisch gut kontrollierbaren Messaufbau sowie an der intuitiven grafischen Darstellung.

Gauß-Quadratur

Auch bei der Gauß-Quadratur wird die Kugeloberfläche horizontal mit $N_\varphi = 2(N+1)$ equiangularen Stützstellen abgetastet. Für die vertikale Abtastung werden nun aber nur $N_\theta = N+1$ Stützstellen benötigt, die in den Nullstellen des Legendre-Polynoms der Ordnung $N+1$ liegen. Es gilt also

$$\varphi_i = \frac{2\pi i}{N_\varphi} \quad , \quad i = 0, ..., N_\varphi - 1 \tag{3.23}$$

$$P_{N+1}(\cos\theta_k) \stackrel{!}{=} 0 \quad , \quad k = 0, ..., N_\theta - 1 \quad . \tag{3.24}$$

Die dazugehörigen Gewichtungsfaktoren können aus den Ansätzen in [DH94] berechnet werden oder Tabellen, etwa in [AW70], entnommen werden. Vorteile der Gauß-Legendre-Quadratur gegenüber equiangularer Abtastung ist die geringere Anzahl von Stützstellen in vertikaler Richtung. Es sind mit $L = 2(N+1)^2$ hier nur halb so viele Abtastpunkte

nötig.

Lebedev-Grid

Das Lebedev-Grid basiert auf dem Ansatz, ein (quasi) gleichförmiges Abtastgitter zu erzeugen, d.h. dass alle Eckpunkte den selben Abstand zu ihren direkten Nachbarn haben. Dieses Ideal gilt allerdings nur für die fünf Platonischen Körper *Tetraeder*, *Hexaeder*, *Oktaeder*, *Dodeeder* und *Ikosaeder*. Die maximale Anzahl der Abtastpunkte wäre beim Dodekaeder mit $L = 20$ sehr begrenzt. In [Leb76, LS92] werden annähernd gleichförmige Abtastgrids für höhere Ordnungen entwickelt. Die Anzahl der Datenpunkte liegt dabei je nach Konfiguration bei $L \approx 1.3(N+1)^2$. Ähnlich wie bei der Gauß-Quadratur ist auch hier der Anspruch an mögliche Messaufbauten vergleichsweise hoch, da die angularen Schrittweiten nicht mehr ganzzahlig sind.
Der Einfluss des verwendeten Abtastgrids auf die Ergebnisse der $\mathcal{SHT}$ wird im Abschnitt 4.2.5 an Hand einiger Beispiele dargestellt.

3.4. Stör- und Fehlerquellen

Bei der praktischen Berechnung der in Kapitel 2 vorgestellten Theorien stößt man unvermeidbar an Grenzen der Berechnungsgenauigkeit. Diese entstehen bei der Überführung der analytischen Modelle in numerische Anwendung, wie sie mit heutiger Computertechnik nötig ist. In diesem Abschnitt sollen diese Grenzen ermittelt und deren quantitativer Einfluss auf die Ergebnisse beschrieben werden.

3.4.1. Spatial Aliasing bei der Analyse $\mathcal{SHT}$

Der theoretische Ansatz für die Synthesegleichung der $\mathcal{SHT}$ geht von einer unendlichen Summation der sphärischen harmonischen Koeffizienten $\mathring{P}_n^m(r,\omega)$ aus (siehe (2.28)). Bei realen Implementierungen muss diese Summation in ihrer modalen Ordnung bandbegrenzt werden. Es soll $\mathring{P}_n^m = 0$ für $n > N_{max}$ gelten. Aus (2.28) wird also

$$P(r,\theta,\varphi,\omega) = \sum_{n=0}^{N_{max}} \sum_{m=-n}^{+n} \mathring{P}_n^m(r,\omega) Y_n^m(\theta,\varphi) \quad . \tag{3.25}$$

Diese maximale Ordnung N_{max} ergibt sich dabei bei der Schallfeldanalyse aus den Eigenschaften der Kugelabtastung. Die Analysegleichung (2.29) über L Abtastpunkte spatial diskretisiert wird mit

$$\mathring{P}_{n,\lim}^m(r,\omega) = \sum_{j=1}^{L} P(\mathbf{x}_j,\omega) Y_n^m(\theta_j,\varphi_j) \sin\theta \mathrm{d}\theta \mathrm{d}\varphi \tag{3.26}$$

approximiert. Für die in Abschnitt 3.3 vorgestellten Kugelabtastungen ergeben sich entsprechend den Ausführungen in [DH94] und [Raf05] die Zusammenhänge in Tabelle 3.1. Besteht ein Schallfeld $P(\mathbf{x}, \omega)$ ausschließlich aus Ordnungen $n \leq N_{max}$, so kann dieses

equiangulare Abtastung:	$L = 4(N_{max}+1)^2$;	$N_{max} = \sqrt{\frac{L}{4}} - 1$
Gauß-Quadratur:	$L = 2(N_{max}+1)^2$;	$N_{max} = \sqrt{\frac{L}{2}} - 1$
Legendre-Quadratur:	$L \approx 1...1.5(N_{max}+1)^2$	

Tabelle 3.1.: Maximale Ordnung N_{max} in Abhänigkeit von der Anzahl der Abtastpunkte L für verschiedene Abtast-Strategien

mit (3.25) und (3.26) exakt transformiert werden. Ein solch definiertes Schallfeld ist allerdings sehr realitätsfern und Schallfelder mit Ordnungen $N' > N_{max}$ die Regel. Der dabei entstehende Fehler

$$\mathring{P}^m_{n,\lim}(r,\omega) = \sum_{n'=0}^{\infty} \sum_{m'=-n'}^{n'} \mathring{P}^{m'}_{n'}(r,\omega) \left[\delta_{nn'}\delta_{mm'} + \epsilon(n,m,n',m')\right] \tag{3.27}$$

([RWB07, (5)]) wird *Spatial Aliasing* genannt. Der Einfluss der nicht angebildeten Ordnungen $N_{max} < n' \leq N'$ ist ausführlich in [RWB07] erläutert. Für strukturierte Kugelabtastungen wie equiangulare oder Gauß-Quadratur ist demnach der entstehende Fehler analytisch beschreibbar ([RWB07, (7)]). An dieser Stelle soll jedoch ein qualitativer Ansatz, in welchem Bereich mit Spatial Aliasing gerechnet werden muss, verfolgt werden. Dazu soll eine Abschätzung gegeben werden, welchen Einfluss die Anteile der Ordnungen $n > N_{max}$, die zweifelsohne bei einer realen Schallquelle vorhanden sind, auf die $\mathcal{SHT}$ haben. Wir betrachten eine Schallquelle wieder gemäß dem Rayleigh-Integral (3.6) als Summation von elementaren Kugelmonopolen. Mit (3.4) wurde dafür bereits ein analytischer Ausdruck für das Wellenmodenspektrum eines Monopols an der Stelle $x_0 = (r_0, \varphi_0, \theta_0)$ gefunden

$$\breve{G}^m_n(\omega) = -\mathrm{j}\frac{\omega}{c} j_n(\frac{\omega}{c} r_0) Y^m_n(\theta_0, \varphi_0)^* \quad .$$

Die konjugiert komplexen sphärischen Harmonischen werden also mit der sphärischen Besselfunktion gewichtet. Befindet sich der Monpol nun im Ursprung d.h. $r_0 = 0$, ist $j_n = 0$ für $n \neq 0$ (siehe Abb. 2.3(a)). Es wird bei der Betrachtung eines äußeren Problems also, wie man es erwarten würde, ausschließlich die sphärische Harmonische $n = 0, m = 0$ angeregt. Sobald sich der Monpol aber vom Ursprung entfernt, tragen auch höhere Ordnungen zum Schallfeld bei.

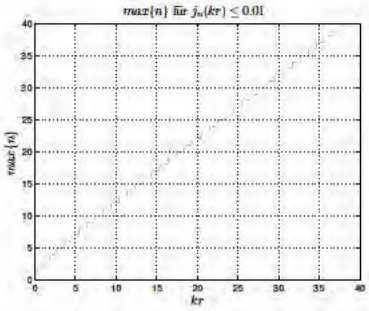


Abbildung 3.6.: Numerische Auswertung $max\{n\}$ für $j_n(kr) \leq 0.01$

Abbildung 3.6 zeigt die maximale Ordnung, für die die sphärische Besselfunktion bei gegebenen kr mindestens 1% annimmt. Aus dieser numerischen Auswertung kann man die qualitative Aussage treffen, dass ein um r_0 vom Ursprung entfernter Monopol bis mindestens zur Ordnung $N = kr_0$ sphärisch harmonisch zerlegt werden muss, damit Aliasing signifikant unterdrückt wird. D.h. übertragen auf das Rayleigh-Integral und die Größe der Schallquelle soll

$$N \geq k\frac{l_{max}}{2} \tag{3.28}$$

bzw.

$$f \leq \frac{N \cdot c}{\pi\, l_{max}} \tag{3.29}$$

gelten, wobei $l_{max}/2$ die größte Abmessung einer im Ursprung zentrierten komplexen Schallquelle bezeichnet (vgl. [Bru07, WA80]). Der Einfluss von Spatial Aliasing auf die praktische Versuchsanwendung wird in Kapitel 5 beschrieben.

3.4.2. Spatial Aliasing bei der Synthese $\mathcal{SHT}^{-1}$

Analog zum temporalen Sampling, das maßgeblich durch die Samplingfrequenz nach Shannon

$$f_s = 2f_{max} \tag{3.30}$$

beeinflusst wird, muss auch bei der räumlichen Fouriertransformation $\mathcal{F}_x$ auf ein entsprechend dicht beieinanderliegendes Netz von Abtastpunkten geachtet werden. Wir stellen uns das später ausgewertete Szenario vor, bei dem das Schallfeld auf einer vertikalen Linie $\mathbf{L_z}$ der Länge L_z, d.h. in den Koordinaten $P(x_{ref}, y_{ref}, z, \omega)$ und $-L_z/2 \leq z \leq L_z/2$, ausgewertet werden soll. Die Diskretisierung von z muss so gewählt werden, dass im worst case (einer ebenen Welle in Richtung $k_x = k_y = 0$, $k_z = k$) Δz nicht größer als die halbe

minimale Wellenlänge ist. Praktisch gesprochen soll also

$$\Delta z \stackrel{!}{=} \frac{1}{2}\lambda_{min} = \frac{c}{2f_{max}} \tag{3.31}$$

gelten. Etwas eleganter gelingt diese Beschränkung im Wellenzahlenraum. Die „spatiale Samplingfrequenz“ kann mit

$$k_s = 2k_{max} = 2 \cdot \frac{2\pi f_{max}}{c} \tag{3.32}$$

beschrieben werden und Δz entsprechend mit

$$\Delta z = \frac{2\pi}{k_s} \quad . \tag{3.33}$$

Wird in (3.32) in (3.33) eingesetzt, erhält man (3.31), womit diese praktische Herangehensweise bestätigt wird (vgl. [AS10a], [AS10b, (38)], [SRS14, (22)]).

3.4.3. Parametrisierung der spatialen DFT

Bei der diskreten Fouriertransformation spielt nicht nur die Samplingfrequenz, sondern auch die Art und Länge des angewandten Fensters eine Rolle. Dies gilt sowohl für die zeitliche wie auch für die örtliche DFT. Die Länge des gewählten Fensters, d.h. die Anzahl der Stützstellen N_{DFT}, hat bei der $\mathcal{F}_t$ direkten Einfluss auf die Frequenzauflösung $\Delta f = \frac{f_s}{N_{DFT}}$. Übertragen auf die spatiale DFT in z-Richtung ist der Zusammenhang

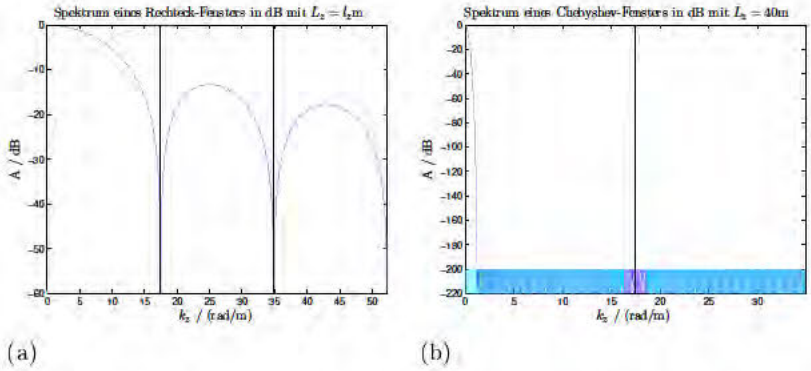
$$\Delta k_z = \frac{k_s}{N_{DFT}} \quad . \tag{3.34}$$

Für das im vorherigen Abschnitt vorgestellte Szenario ergibt sich die Länge des Fensters aus der Länge der auszuwertenden Linie L_z und dessen Diskretisierung

$$N_{DFT} = \frac{L_z}{\Delta z}. \tag{3.35}$$

Da für Δz bereits eine durch Spatial Aliasing bestimmte Grenze bestimmt wurde, bleibt der Freiheitsgrad der Länge der auszuwertenden Linie. Um hierfür einen geeigneten Wert zu bestimmen, ist es sinnvoll sich Gedanken über das später praktisch evaluierte Problem zu machen. Wir haben bei der Beschreibung der zu erwartenden Linienquelle in Abschnitt 3.2.3 festgestellt, dass Nullstellen der sinc-Funktion an den Stellen $k_z = \mu\frac{2\pi}{l_z}$ für $|\mu| = 1, 2, 3, ...$ auftreten, wobei l_z die Länge der Linienquelle bezeichnet. Um diese Einschnitte sauber abzubilden, ist es also sinnvoll dies als obere Grenze zu definieren

$$\Delta k_z \ll \frac{2\pi}{l_z} \quad . \tag{3.36}$$



(a) (b)

Abbildung 3.7.: Fourierspektrum verschiedener Fenster: (a) Rechteckfenster mit $L_z = l_z$ bzw. $N_{DFT} = 42$, (b) Chebyshev-Fenster mit -200dB - Dämpfung und $L_z = 40$ bzw. $N_{DFT} = 4665$

Dass diese Grenze aber noch nicht ausreicht, um ein sauberes Ergebnis zu erzielen ist in Abb. 3.7(a) zu erkennen. Hierzu wurden (3.33), (3.34) und (3.36) in (3.35) eingesetzt, wodurch $L_z = l_z = 0.36$m, die Länge der auszuwertenden Linie gleich der Länge der Linienquelle ist. Es ist nun die Fouriertransformation eines Rechteckfensters zu erkennen, das zwar an den korrekten Stellen k_z austastet. Allerdings werden benachbarte Werte stark miteinbezogen, wodurch das Ergebnis an Eindeutigkeit verliert. Je näher die Fenster-Form der eines Diracs kommt, desto weniger Artefakte entstehen bei der Faltung der DFT. In Abb. 3.7(b) ist nun eine optimierte Fensterform zu sehen. Dabei wurden mit Hilfe eines Chebyshev-Fensters mit -200dB - Stoppdämpfung die Seitenkeulen erheblich gedämpft. Zum anderen wurde mit $L_z = 40m$ die Anzahl der Stützstellen N_{DFT} um den Faktor 110 erhöht, wodurch man eine schmale Hauptkeule erhält. Für die Auswertung der realen Balloon-Daten in Kapitel 5 wird die spatiale DFT also mit den Parametern in Tabelle 3.2 durchgeführt.

f_{max}	k_s	Δz	L_z	N_{DFT}	Δk_z
24 kHz	879 rad/m	0.005m	40m	8001	0.11 rad/m

Tabelle 3.2.: Parameter der spatialen Fouriertransformation für die praktische Anwendung in Kapitel 5

4. Implementierung einer modalen Zerlegung von realen Balloon-Messdaten

In diesem Kapitel wird die Implementierung der modalen Zerlegung und Resynthese der Balloon-Messdaten in Matlab beschrieben. Dabei werden die in Kapitel 2 vorgestellten mathematischen Zusammenhänge genutzt und unter den Bedingungen und Limitierungen aus Kapitel 3 evaluiert. Dazu sei an dieser Stelle das zugrunde liegende Flussdiagramm in Abb. 4.1 gezeigt, an dem sich die einzelnen Schritte ablesen lassen. Im ersten Schritt werden Balloon-Daten in Matlab eingelesen. Dabei alle relevanten Parameter der einzelnen Spektren erfasst und kompakt in einem *Structure Array* (kurz: struct) gespeichert. Anschließend werden diese FFT-Daten mittels $\mathcal{SHT}$ ins Modalspektrum $\breve{P}_n^m(\omega)$ zerlegt, welches ebenfalls als struct auf der Festplatte hinterlegt wird. Im letzten Schritt wird aus dem Modalspektrum mittels $\mathcal{SHT}^{-1}$ das Schallfeld auf eine Linie im Raum (vgl. Abb. 2.7) expandiert und in den Wellenzahlenraum $\tilde{P}(x, y, k_z)$ transformiert. Die Implementierun-

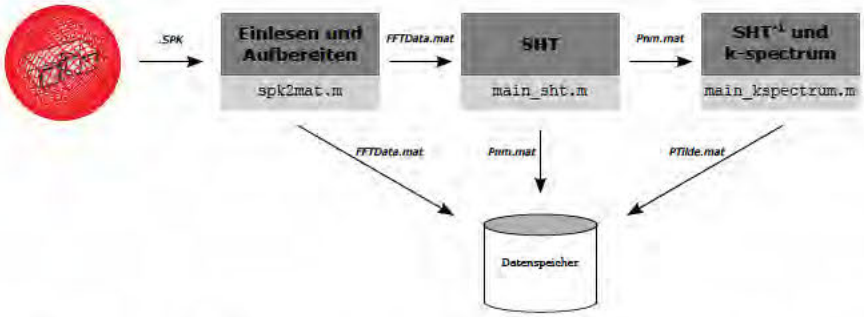


Abbildung 4.1.: Flussdiagramm der Implementierung in Matlab

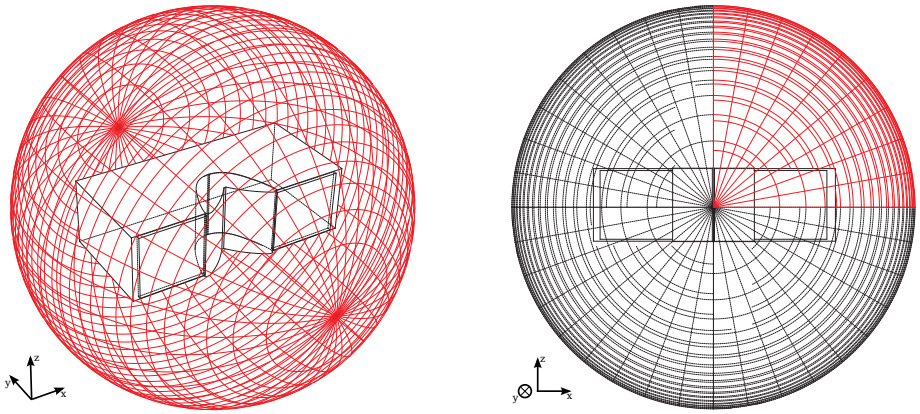
gen sind in diesem Kapitel jeweils kurz mathematisch oder durch Pseudocode beschrieben und verweisen auf die lauffähigen Code-Listings in Anhang A. Alle Berechnungen wurden mit einem Thinkpad T410, 4GB RAM, i5@2.4GHz, 64bit-Linux,32bit-MatlabR2012a durchgeführt. Besonders die Beschränkung des Speicherzugriffs mit 32bit-Programmen setzt der Implementierung deutlich Grenzen. Die maximale Größe einer Matrix für diese Konfiguration ist auf 300e6 `real double` beschränkt und der komplette Worspace auf 3558MB[1]. Hier hätte eine Matlabversion in 64bit Abhilfe geschafft, allerdings ist der Code (auch) entstanden, um später auf möglichst vielen Systemen lauffähig zu sein. An den entsprecheden Stellen wird in diesem Kapitel darauf hingewiesen, wie 64bit-Nutzer den

[1] http://www.mathworks.com/matlabcentral/answers/91711, zuletzt besucht am 29.04.2014

Code zur Performanzsteigerung anpassen können. Dies betrifft insbesondere die Reduzierung von speicherbedingten Iterationen.

4.1. Struktur und Aufbereitung des Datensatzes

Der für die modale Zerlegung verwendete Messdatensatz wurde uns freundlicherweise von Anselm Goertz und Michael Makarski vom *Institut für Akustik und Audiotechnik*[2] zur Verfügung gestellt. Das Messobjekt ist das 4-Wege Line-Array-Element „Flashline“ der Firma *Turbosound*[3], das auch Grundlage für die Messungen in [Goe12] war. Der Radius der Kugelabtastung beträgt 8m und befindet sich somit gemäß den Bedingungen 3.1 im Fernfeld. Der Datensatz besteht aus komplexen Spektren, die durch die inverse, zeitliche Fouriertransformation $\mathcal{F}_t^{-1}$ in Impulsantworten transformiert werden können. Es sei darauf hingewiesen, dass aus Gründen der Symmetrie lediglich die Spektren für Frequenzen im Intervall $[0, \frac{f_s}{2}]$ hinterlegt sind, wobei f_s die Samplingfrequenz bezeichnet. Diese Spektren wurden auf einem equiangularen Grid in verschiedenen Schrittweiten gemessen. Da bei einem Lautsprecher dieser Bauart von zwei Symmetrieachsen ausgegangen werden kann, besteht der Datensatz lediglich für eine Viertelkugel. Mit der Annahme der Symmetrie kann durch entsprechende Spiegelung ein kompletter Messballoon aus den vorhanden Daten erzeugt werden. Die Orientierung der Kugelabtastung ist dabei nicht wie üblich und in 3.3 beschrieben realisiert, sondern um 90° vertikal nach vorne gekippt (siehe Abbildung 4.2). Dies hat den Vorteil, dass in der Hauptabstrahlrichtung des Laut-



(a) (b)

Abbildung 4.2.: Orientierung der Kugelabtastung für den vorliegenden Messdatensatz. In (b) ist die vorliegende Kugelabtastung in rot dargestellt, nach Spiegelung an den Symmetrieachsen ist die gesamte Kugel abgedeckt (a). Grafik: Martin Franck

[2] http://www.ifaa-akustik.de/, zuletzt besucht am 29.04.2014
[3] http://www.turbosound.com/docs/products/TFS-900H.shtml, zuletzt besucht am 29.04.2014

Dataset	**FR-BAL**	**HF-BAL**	**HM-BAL**	**LM-BAL**	**LF-BAL**
$\Delta\theta = \Delta\varphi =$	5	2		5	
$L_{quarter}$	703	2168		703	
L_{full}	2664	16380		2664	
FFT size	8193				
N_{Nodes} (quarter)	5.759.679 `complex double`	34.148.424 `complex double`		5.759.679 `complex double`	
N_{Nodes} (full)	21.826.152 `complex double`	134.201.340 `complex double`		21.826.152 `complex double`	
Size (quarter)	88 MiB	521 MiB		88 MiB	
Size (full)	333 MiB	2048 MiB		333 MiB	

Tabelle 4.1.: Dimensionen der vorliegenden Datensätze

sprechers ein Pol der Kugelabtastung liegt und somit die Dichte der Abtastpunkte am höchsten ist. Nachteil dieser Anordnung ist aber sicherlich die intuitive Übertragung dieser Messanordnung auf die in dieser Arbeit beschriebenen geometrischen Gegebenheiten. Aus diesem Grund wird hier nun die komplette Messanordung virtuell um 90° nach oben gedreht, die Hauptabstrahlrichtung des Lautsprechers ist nun in positiver z-Richtung und die Pole der Kugelabtastung entsprechen wieder dem geographischen Nord- bzw. Südpol. Die geometrische Umwandlung entspricht

$$x_{goertz} = x$$
$$y_{goertz} = z$$
$$z_{goertz} = -y$$

und ist in den Funktionen `cart2goertz.m` (Lst. A.16) und `goertz2cart.m` (Lst. A.17) implementiert. Der Datensatz liegt zum einen als Full-Range-Messung mit der Schrittweite $\Delta\theta = \Delta\varphi = 5°$ vor, bei der alle Lautsprecherwege wie im normalen Einsatz gleichzeitig angestuert werden. Zum anderen liegen die 4 Wege als Einzelmessung unterschiedlichen Schrittweiten vor. Daraus resultieren die Dimensionen der Datensätze in Tabelle 4.1. Anhand dieser Dimensionen ist zu sehen, dass die Datensätze (besonders HF- und HM-BAL) einen großen Bedarf an Speicherplatz haben und die Implementierung in Matlab entsprechend darauf ausgelegt werden muss. Um Ressourcen zu sparen, werden etwa die Datensätze darum nicht zunächst auf die volle Kugel gespiegelt, sondern nur die Daten der Viertelkugel geladen. Die Symmetrieeigenschaften werden nun so ausgenutzt, dass Messpunkte an Winkeln, die sich außerhalb des nativen Datensatzes befinden, durch eine Lookup-Tabelle referenziert werden. Die Funktionalität von `angleLookUpTable.m` in Lst. A.5 ist in einem kurzen Beispiel in Tabelle 4.2 gezeigt. Auch die Wahl des Datensatzes (Full-Range vs. Einzelwege) hat eine wesentliche Bedeutung. Dazu sollte man sich den praktischen Nutzen dieser Arbeit in Erinnerung rufen. Zum einen soll ein praktikabler allgemeiner Workflow für die Berechnung der Koeffizienten $\mathring{P}_n^m(r,\omega)$ bzw. $\breve{P}_n^m(\omega)$ für eine Schallquelle (hier: Lautsprecher) entstehen. Zu diesem Zweck ist es sicherlich zweckmäßig

		φ											
		0	**30**	**60**	**90**	**120**	**150**	**180**	**210**	**240**	**270**	**300**	**330**
	0	1	8	15	22	15	8	1	8	15	22	15	8
	30	2	9	16	23	16	9	2	9	16	23	16	9
	60	3	10	17	24	17	10	3	10	17	24	17	10
θ	**90**	4	11	18	25	18	11	4	11	18	25	18	11
	120	5	12	19	26	19	12	5	12	19	26	19	12
	150	6	13	20	27	20	13	6	13	20	27	20	13
	180	7	14	21	28	21	14	7	14	21	28	21	14

Tabelle 4.2.: Exemplarische Funktionsweise der angularen Lookup-Tabelle, wobei in rot die Indizes der nativen Abtastpunkte gekennzeichnet sind.

einen einheitlichen Datensatz über das gesamte relevante, also hörbare, Frequenzspektrum zu wählen. Zum anderen soll in dieser Arbeit aber auch konkret auf die Frage der Richtwirkung eines Line-Array-Elements eingegangen werden. Dabei spielt das Vorwissen aus Kapitel 3 eine Rolle, dass die Richtwirkung von Lautsprechern mit zunehmender Frequenz zunimmt. Das heißt, dass unter diesem Aspekt die hoch- und mittelhochfrequenten Anteile besonders interessieren. Da diese Anteile entsprechend Tabelle 4.1 bei den Einzelwegs-Messungen in höherer Auflösung vorliegen und dadurch robuster gegenüber spatialen Aliasing sind, eignen sich diese Datensätze hierfür besonders. Entsprechend den Ausführungen in den Abschnitten 3.3, 3.4 gelten folgende Bedingungen für die Datensätze

	FR-BAL	**HF-BAL** **HM-BAL**
$\Delta\varphi$	10°	4°
$\Delta\theta$	5°	2°
L	1296	8100
N_{max}	17	44
f_{alias}	5155Hz	13344Hz

Tabelle 4.3.: Eigenschaften der Datensätze bezüglich Bandbegrenzung und Aliasing

4.2. Implementierung und Evaluierung von Matlab-Funktionen

Da die Implementierung der modalen Zerlegung in Matlab erfolgen und dessen eingebaute Funktionen so weit wie möglich genutzt werden sollen, ist es zunächst nötig diese Funktionen auf die Übereinstimmung mit den in dieser Arbeit übernommenen mathematischen Vorzeichen- und Normierungskonventionen hin zu überprüfen.

4.2.1. Koordinatensysteme

Matlab nutzt eine andere Konvention bezüglich der sphärischen Koordinaten. Dies betrifft den vertikalen Zenith-Winkel bzw. die Elevation. Während in dieser Arbeit der

Elevationswinkel θ im Uhrzeigersinn vom Nordpol (0° bzw. 0 rad) zum Südpol (180° bzw. π rad) gezählt wird, berechnet sich der Winkel in Matlab gegen den Uhrzeigersinn relativ zur positiven x-Achse. Die Definitionsmenge für den Elevations-Winkel vom Süd- zum Nordpol ist in Matlab also $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. Diese beiden Konventionen werden durch die Beziehung

$$\theta_{Matlab} = \frac{\pi}{2} - \theta \tag{4.1}$$

in Verbindung gebracht. Dadurch verändert sich auch die Umrechnung von sphärischen in kartesische Koordinaten. Im Gegensatz zu (2.1a) gelingt in Matlab die Konvertierung mit

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = r \cdot \begin{pmatrix} \cos(\varphi)\cos(\theta_{Matlab}) \\ \sin(\varphi)\cos(\theta_{Matlab}) \\ \sin(\theta_{Matlab}) \end{pmatrix} \tag{4.2}$$

Diese Konvertierung ist in den Skripten `mySph2Cart.m` (Lst. A.19) und `myCart2Sph.m` (Lst. A.18) implementiert.

4.2.2. FFT/IFFT

Um die Vorzeichen- und Normierungskonventionen der zeitlichen und räumlichen Fourier-Transformation in dieser Arbeit (siehe Abschnitt 2.2) korrekt zu implementieren, müssen diese mit den verwendeten Standards in Matlab in Übereinstimmung gebracht werden. Aus der Dokumentation [Mat13] ist dabei die Konvention für die DFT und ihrer Inversion zu entnehmen:

$$\texttt{fft.m:} \qquad X[k] = \sum_{n=0}^{N-1} x[n]\ \mathrm{e}^{(-2\pi \mathrm{j} nk)/N} \tag{4.3}$$

$$\texttt{ifft.m:} \qquad x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\mathrm{e}^{(2\pi \mathrm{j} nk)/N} \tag{4.4}$$

Diese Konvention stimmt als zeitliche DFT der in dieser Arbeit genutzten Konventionen in (2.2), (2.3) überein. Die spatiale Fourier-Transformation (2.4), (2.5) wird in dieser Arbeit mit dazu umgekehrter Vorzeichenkonvention definiert. Die Unterschiede bezüglich Normierung und Vorzeichen zu der in dieser Arbeit verwendeten Konvention sind in Tabelle 4.4 dargestellt. Zur einfachen Verwendung stehen die Skripte `fft_s.m`, `ifft_s.m`, `fft_s.m` und `ifft_s.m` bereit (Lst. A.20 -A.23).

Operation	Vorzeichen	Normierung	Matlab
$\mathcal{F}_t$	$-$	1	`fft(f_t)`
$\mathcal{F}_t^{-1}$	$+$	$\frac{1}{2\pi}$	`ifft(F_t)`
$\mathcal{F}_s$	$+$	1	`ifft(f_s)*length(f_s)`
$\mathcal{F}_s^{-1}$	$-$	$\frac{1}{2\pi}$	`fft(F_s)/length(F_s)`

Tabelle 4.4.: Unterschiede in Normierung und Vorzeichen für Fouriertransformationen

4.2.3. Sphärische Harmonische

In der Literatur gibt es verschiedene Konventionen der sphärischen Harmonischen $Y_n^m(\theta, \varphi)$. Die Unterschiede liegen dabei meistens im Normierungsterm. Im theoretischen Teil dieser Arbeit wird der Konvention nach [GD04] gefolgt (vgl. (2.24)):

$$Y_n^m(\theta, \varphi) = (-1)^m \sqrt{\frac{2n+1}{4\pi} \cdot \frac{(n-|m|)!}{(n+|m|)!}} \cdot P_n^{|m|}(\cos\theta) \cdot \mathrm{e}^{\mathrm{j}m\varphi} \tag{4.5}$$

Eine andere Konvention wird in der SOFiA - Toolbox genutzt ([BPSW11, (1)]), die bestimmte mathematische Funktionen mit Hilfe der *Boost C++ Libraries*[4] implementiert. Aus dessen Dokumentation (Math/Special Functions) geht hervor, dass hier die Konvention implementiert ist, die etwa auch in [Wil99] zu finden ist

$$Y_n^m(\theta, \varphi) = \sqrt{\frac{2n+1}{4\pi} \cdot \frac{(n-m)!}{(n+m)!}} \cdot P_n^m(\cos\theta) \cdot \mathrm{e}^{\mathrm{j}m\varphi} \quad . \tag{4.6}$$

Die beiden Konventionen (4.5) und (4.6) unterscheiden sich lediglich beim Vorzeichen in Fällen von positiven, ungeraden m. Um konsistent zu bleiben, wird in den vorgestellten Matlab-Skripten die Konvention nach (4.6) beibehalten, während in der theoretischen Beschreibung weiterhin (4.5) herangezogen wird. Im Skript `sphHarm_N.m` (Lst. A.6) sind beide Normierungen implementiert, wobei standardmäßig (4.5) berechnet wird.

4.2.4. Einlese- und Aufbereitungsroutine

Der erste Schritt im Flussdiagramm ist das Einlesen und Aufbereiten der komplexen Spektren. In der vorliegenden Implementierung `spk2mat.m` (Lst. A.2) müssen die Rohdaten als Monkey Forest [5] Dateien der Form `VxxxHyyy.SPK` vorliegen, wobei `xxx` den Azimuth und `yyy` die Elevation in Grad bezeichnet. Andere Dateiformate und Konventionen können aber unkompliziert eingefügt werden. Die Funktion benötigt obligatorisch den Pfad zum beeinhaltenden Ordner der Spektren, sowie optional den Messradius und die Orientierung der Kugelabtastung. Es werden alle Spektren eingelesen und zusammen mit allen nötigen Informationen in einem Struct im aktuellen Verzeichnis auf der Festplatte abgelegt. Das Struct wird in der Version -v7.3 abgelegt, die durch das HDF5-Dateiformat komfortable Möglichkeiten zum partiellen Lesen und Schreiben der Datei bereitstellt. Die Felder, die in diesem Struct abgelegt sind, können aus Tabelle 4.5 entnommen werden.

4.2.5. Abtastgrid

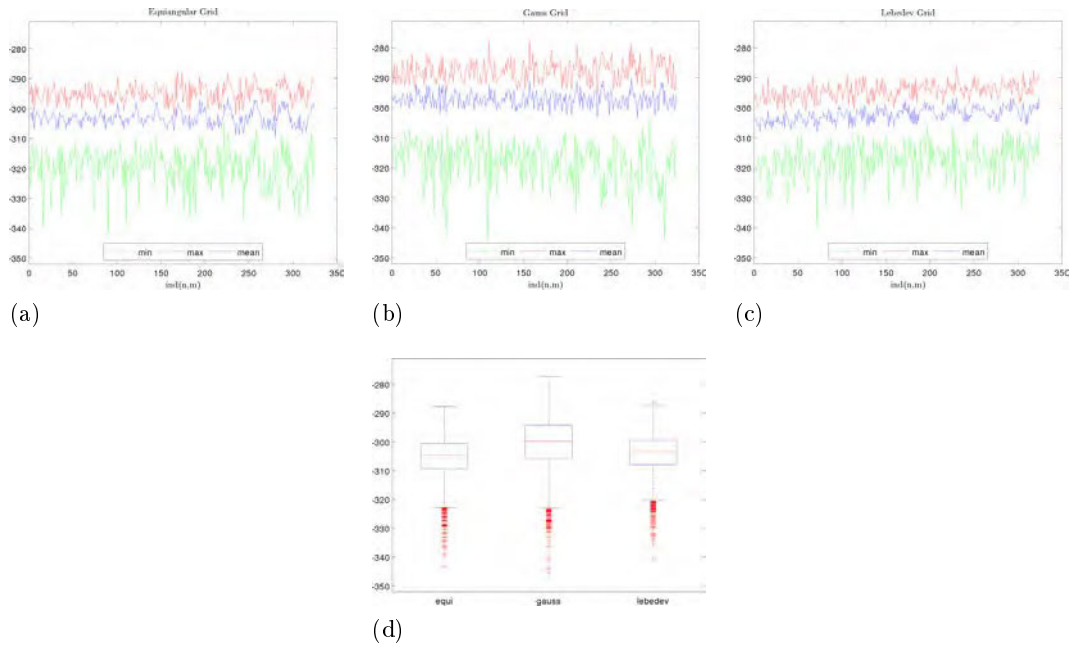
Wie in Abschnitt 3.3 beschrieben, gibt es verschiedene Ansätze die Kugeloberfläche abzutasten. Dass es dabei Strategien gibt, die mehr oder weniger gut geeignet sind, soll

[4] http://www.boost.org/, zuletzt besucht am 29.04.2014
[5] http://www.four-audio.com/de/produkte/monkey-forest.html, zuletzt besucht am 29.04.2014

Name	Format	Beschreibung
dataset	string	Name des Datensatzes
path	string	Pfad zum Ordner der Spektren
equi	logical	equiangulares Grid
fft_size	1 x 1 double	Länge der FFT ($f = 0...fs/2$)
fs	1 x 1 double	Samplingrate
r_meas	1 x 1 double	Messradius
N_max	1 x 1 double	maximale modale Ordnung
sph_orientation	string	Lage der Kugelabtastung (vgl. 4.1)
sph_coverage	1 x 2 double	Abdeckung der Kugelabtastung
resol	1 x 2 double	Schrittweite in °
grid_orig	L_o x 3 double	Winkel der originalen Kugelabtastung [az,el,weight]
grid_full	L_f x 3 double	Winkel der kompletten Kugelabtastung [az,el,weight]
angle_LUT	L_f x 1 double	Lookup-Table der Spektren
XFFT	L_o x fft_size complex double	komplexe Spektren

Tabelle 4.5.: Felder des gespeicherten Structs und ihre Bedeutungen

an dieser Stelle betrachtet werden. Als Testdatensatz wird ein sphärisches Spektrum $P(r, \varphi, \theta, \omega)$ erzeugt, wobei die Winkel den entsprechenden Grid-Konventionen entsprechen. Die Spektren stammen aus der Summe von 30 sphärischen Harmonischen gleicher Amplitude mit zufällig gewählten n und m. Diese Datensätze wurden nun mittels $\mathcal{SHT}$ in seine Koeffizienten $\mathring{P}(r, \omega)$ zerlegt und mit den vorher gewählten n und m verglichen. Abbildung 4.3 zeigt nun die Abweichungen der Koeffizienten für Testdatensätze bis zur Ordnung $n \leq N = 17$ bei 20 Wiederholungen. In den Einzeldarstellungen Abb. 4.3a-c



(a) (b) (c)

(d)

Abbildung 4.3.: Abweichungen der Koeffizienten $\mathring{P}_n^m$ in dB vom Testdatensatz mit verschiedenen Abtastgrids.

Grid	L
Equi	1296
Gauss	648
Lebedev	434

Tabelle 4.6.: Anzahl der Abtastpunkte L für $N = 17$

ist jeweils der mittlere, minimale und maximale Fehler der 20 Wiederholungen für die entsprechenden Kombinationen für n und m aufgetragen. Es stellt sich dabei keine signifikante Abhängigkeit des Fehlers von der Ordnung heraus, weshalb im Boxplot (d) die Gesamtheit aller Fehler je Abtastgrid vereint wurde. Die Mediane liegen dabei nicht mehr als 6dB auseinander, allerdings auf einem mit -300dB allgemein sehr niedrigen Niveau. Die Größenordnung liegt dabei in der Nähe des minimalen Fehlers für double precision floats $eps_{dB} = 20log_{10}(eps) = -313$dB. Es ist hier allerdings darauf hingewiesen, dass es sich hier um best-case Szenarios handelt. Das heißt, dass das Schallfeld der Testdatensätze keine modalen Anteile für $n > N_{max}$ aufweist und somit kein Spatial Aliasing auftritt. Das Ergebnis zeigt also lediglich, dass die Implementierung der $\mathcal{SHT}$ für die gewählten Abtastgrids an sich funktioniert und liefert keinerlei Hinweise auf die Zerlegung echter Balloon-Daten. Die Zerlegung mit einem Gauß-Grid schneidet mit leichtem Vorsprung am besten ab. Wenn man die minimale Anzahl der Abtastpunkte, um diese Ordnung abzubilden, in Tabelle 4.6 betrachtet, kann man aber eine deutliche Aussage über das Aufwand/Nutzen - Verhältnis treffen. Denn um eine ähnliche Genauigkeit zu erreichen sind gegenüber der equidistanten Abtastung für das Gauß-Grid nur halb so viele und für das Lebedev-Grid sogar nur 1/3 mal so viele Messpunkte von Nöten.

4.2.6. $\mathcal{SHT}$

Eine Komponente für die Implementierung der modalen Zerlegung ist die *SOFiA*-Toolbox[6] ([BPSW11]) zur Schallfeldanalyse. Sie wurde und wird von der *ASAR research group*[7] entwickelt und unter GNU General Public License veröfentlicht. Diese Toolbox enthält .m-Skripte sowie als MEX-files kompilierte C/C++ - Funktionen mit höherer Perfomanz. Es werden u.a. Funktionen zur $\mathcal{SHT}$, $\mathcal{PWD}$, Berechnung von Radialfiltern bereitgestellt und ist optimiert für die Nutzung des VariSphear - Systems[8]. In dieser Arbeit wird die Funktion `sofia_stc.m` zur Berechnung der $\mathcal{SHT}$ genutzt

$$\mathring{P}_n^m(r,\omega) = \sum_{s=1}^{L} w_s \; P(\mathbf{x}_s,\omega) \; Y_n^m(\theta,\varphi)^* \quad . \tag{4.7}$$

Dabei ist Y_n^m entsprechend der Konvention (4.6) nach [Wil99] implementiert (siehe Abschnitt 4.2.3). Mit den Gewichtungen w_s wird je nach Abtastgrid die höhere Dichte

[6] http://code.google.com/p/sofia-toolbox/, zuletzt besucht am 29.04.2014
[7] http://www.fh-koeln.de/asar, zuletzt besucht am 29.04.2014
[8] http://code.google.com/p/sofia-toolbox/wiki/VariSphear, zuletzt besucht am 29.04.2014

der Stützstellen zu den Polen hin kompensiert. Aus Gründen der Speicher- und zeitlichen Performanz musste die Berechnung der $\mathcal{SHT}$ im Skript `main_sht.m` (Lst. A.1) abschnittweise implementiert werden. Die Spektren werden dazu frequenzseitig in Blöcke aufgeteilt (Zeile 103-108) und dann in einer Schleife zunächst auf die komplette Kugeloberfläche expandiert (`angle_LUT`) und anschließend im jeweiligen Frequenzblock modal zerlegt (Z.109). Die Anzahl der FFT-Blöcke ist beliebig gewählt, und immer ein Tradeoff zwischen Speicherbelastung und Rechenzeit, die gewählten 8 Iterationen (d.h. `fft_blocksize = 1024`) haben sich bei Tests aber als guter Kompromiss für 32bit-Systeme herausgestellt. Bei 64bit-Systemen mit größerer Speicheradressierung stellt die Berechnung mit einer einzigen Iteration kein Problem dar. Die berechneten Koeffizienten $\mathring{P}_n^m(r,\omega)$ müssen nun von ihrem radialen Term befreit werden (Z.114-128), damit sie später auf beliebige Orte im Raum $\mathbf{x}_{target} = (\varphi, \theta, r)$ expandiert werden können. Da wir ein äußeres Problem betrachten, wird der radiale Term der $\mathcal{SHT}$ mit der sphärischen Hankelfunktion 2. Art gebildet, die von der Ordnung n, der Wellenlänge ω sowie des Radius r abhängig ist. Der Radius des Messdatensatzes ist dabei konstant $r_{meas} = 8$. Die Evaluierung dieses Skriptes erfolgte simultan mit der der Abtastgrids im vorherigen Abschnitt. Mit definiert synthetisierten Schallfeldern konnte eine modale Zerlegung mit sehr guter Genauigkeit erzielt werden ($\epsilon \leq -270\text{dB}$)

4.2.7. Wellenzahlenspektrum aus $\breve{P}_n^m$

Aus den Koeffizienten $\breve{P}_n^m$ kann nun das Schallfeld an jedem beliebigen Ort im Raum berechnet und mit der spatialen DFT in den Wellenzahlenraum transformiert werden. Bei Betrachtung von Lautsprechermessdaten ist es jedoch sinnvoll sich auf eine Dimension x, y oder z zu beschränken, damit eine sinnvolle Darstellung und Evaluierung möglich ist, wie in Abschnitt 2.7 erläutert wurde. In dieser Arbeit steht die vertikale Zunahme der Richtcharakteristik mit steigender Frequenz im Mittelpunkt, weshalb das Wellenzahlenspektrum in Abhängigkeit von k_z dargestellt wird. Die Berechnung im Skript `main_kspectrum.m` (Lst. A.3) wird dabei in zwei Schritten implementiert. Zunächst wird aus den Koeffizienten das Schallfeld mittels $\mathcal{SHT}^{-1}$ auf einer entsprechenden Referenzlinie expandiert (Z.99-133), welches im zweiten Schritt einer räumlicher Fouriertransformation entlang der zu untersuchenden Dimension unterzogen wird (Z.134-157). Als Input-Parameter werden lediglich der Name des Datensatzes sowie die Koordinaten und räumliche Auflösung der Expansionslinie benötigt. Zuvor müssen im aktuellen Pfad die Koeffizienten $\breve{P}_n^m(\omega)$ für den jeweiligen Datensatz abgelegt sein, wie es im vorherigen Abschnitt beschrieben ist. Zur Evaluierung dieses Skriptes wurden verschiedene Testdatensätze erstellt, für die es analytische Lösungen im Wellenzahlenraum gibt. Konkret wurden an dieser Stelle ein Kugelmonopol und eine Linienquelle synthetisiert und das Ergebnis mit den entsprechenden analytischen Lösungen verglichen, die im Abschnitt 3.2 vorgestellt wurden.

Monopol

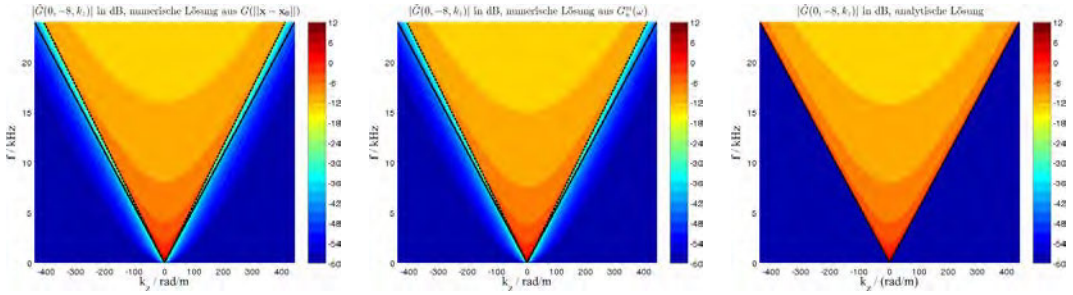
Der Kugelmonopol lässt sich, wie in Abschnitt 3.2.1 vorgestellt, durch die Gleichungen (3.3-3.5) beschreiben

$$G(||\mathbf{x}-\mathbf{x_0}||,\omega) = \frac{\mathrm{e}^{-\mathrm{j}\omega/c||\mathbf{x}-\mathbf{x_0}||}}{4\pi||\mathbf{x}-\mathbf{x_0}||}$$

$$\breve{G}_n^m(\omega) = -\mathrm{j}\frac{\omega}{c} j_n(\frac{\omega}{c} r_0) Y_n^m(\theta_0,\varphi_0)^*$$

$$\tilde{G}(k_x,y,z,\omega) = \begin{cases} -\frac{\mathrm{j}}{4} H_0^{(2)}\left(\sqrt{(\frac{\omega}{c})^2 - k_x^2}\sqrt{|y|^2+z^2}\right) & \text{für } |k_x| \leq |\frac{\omega}{c}| \\ \frac{1}{2\pi} K_0\left(\sqrt{k_x^2 - (\frac{\omega}{c})^2}\sqrt{|y|^2+z^2}\right) & \text{für } |k_x| > |\frac{\omega}{c}| \end{cases} .$$

Mit diesen übersichtlichen analytischen Ausdrücken gelingt eine intuitive Evaluierung der implementierten $\mathcal{SHT}$, $\mathcal{SHT}^{-1}$ und der anschließenden spatialen Fouriertransformation. Abbildung 4.4 zeigt nun drei Spektren $\tilde{G}(k_x,y,z,\omega)$ eines im Ursprung gelegenen Monopols, der auf die Linie $\mathbf{L_z} = (x=0, y=-8, z=-20...20)$ mit $\Delta z = 0.005$ (alle Angaben in m) ausgewertet wurde. Dabei wurden die drei oben genannten Glei-



(a) aus $G(||\mathbf{x}-\mathbf{x_0}||,\omega)$ (b) aus $\breve{G}_n^m(\omega)$ (c) aus $\tilde{G}(x,y,k_z,\omega)$

Abbildung 4.4.: $\tilde{P}(0,-8,k_z,\omega)$ für ein Monopol im Ursprung berechnet mit verschiedenen Ansätzen

chungen implementiert und entsprechend transformiert. Die analytische exakte Lösung in Abb. 4.4(c) zeigt als Referenz sehr deulich den Verlauf der ebenen Wellen. Bei einem Abstand von 8m sind darüber hinaus keine evaneszenten Anteile mit einer Amplitude von mehr als -60dB zu erkennen. Dies unterstützt die Aussage, dass evaneszente Anteile nur im Nahfeld existieren und mit steigender Entfernung rasch abklingen. Die numerischen Lösungen (a) und (b) zeigen grundsätzlich ein ähnliches Bild. Zwei Dinge fallen beim Vergleich zur analytischen Lösung aber auf. Zum einen sind hier evaneszente Anteile zu erkennen. Diese treten durch die Wahl der Parameter der spatialen DFT auf. Eine ausführliche Erklärung und ein Lösungsansatz dies zu vermeiden wird im folgenden Abschnitt zur Betrachtung von Linienquellen gegeben. Der andere Unterschied kann an der Grenze vom propagierenden zum evaneszenten Schallfeld für $|k_z| \to \frac{\omega}{c}$ erkannt werden. Hier wird das durch $|k_z| = \frac{\omega}{c}$ begrenzte Dreieck nicht vollständig ausgefüllt. Der Grund dafür liegt in der geometrischen Konstellation der Schallfeldexpansion. Bei einer endlichen Auswertungslinie $\mathbf{L_z}$ mit $x = 0$ und $max\{|z|\} = 20$ können nur ebene Wellen mit $max\{|\cos\theta_{pw}|\} \approx 0.9284$ abgebildet werden. Dies ist dann auch genau der Faktor, mit dem $k_{z,max} = \frac{\omega}{c} max\{|\cos\theta_{pw}|\}$ je Frequenz angesteuert werden kann. Diese Grenze ist mit der gestrichelten schwarzen Linie markiert.

Linienquelle

Das gleiche Testszenario wird nun mit einer Linienquelle durchgeführt. Dabei kommen die Zusammenhänge aus Abschnitt 3.2.3 zum Einsatz. Konkret werden die Gleichungen (3.17) und (3.5) genutzt herangezogen um das Schallfeld einer Linienquelle zu beschreiben

$$P_{line}(\mathbf{x},\omega) = \mathrm{j}\omega\rho_0 Q\ G(||\mathbf{x}-\mathbf{x_0}||,\omega)\cdot\ \mathrm{sinc}(k_z\ l_z/2)$$

$$\tilde{P}_{line}(x,y,k_z) = \tilde{G}(x,y,k_z)\mathrm{sinc}(k_z\ l_z/2)$$

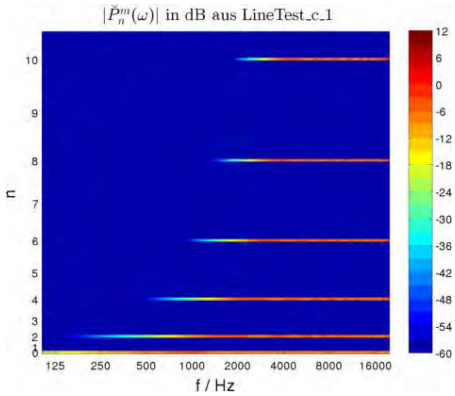


Abbildung 4.5.: $|\breve{P}_n^m(\omega)|$ in dB einer idealen Linienquelle bis zur Ordnung $n = 10$. Die Ticklabel der y-Achse kennzeichnen jeweils die Moden $m = 0$.

mit

$$\tilde{G}(x, y, k_z, \omega) = \begin{cases} -\frac{\mathrm{j}}{4} H_0^{(2)} \left(\sqrt{(\frac{\omega}{c})^2 - k_z^2} \sqrt{|y|^2 + x^2} \right) & \text{für } |k_z| \leq |\frac{\omega}{c}| \\ \frac{1}{2\pi} K_0 \left(\sqrt{k_z^2 - (\frac{\omega}{c})^2} \sqrt{|y|^2 + x^2} \right) & \text{für } |k_z| > |\frac{\omega}{c}| \end{cases}$$

Einen übersichtlichen analytischen Ausdruck für die Koeffizienten $\breve{P}_{n,line}^m(\omega)$ zu finden, ist dagegen nicht so einfach. Man kann allerdings annehmen, dass lediglich Koeffizienten für $m = 0$ vorkommen. Dazu sei daran erinnert, dass das Spektrum einer Linienquelle rotationssymmetrisch um die z-Achse ist, also unabhängig vom Azimuth φ. Dessen Abhängigkeit bei der $\mathcal{SHT}$ in den sphärischen Harmonischen (2.24) soll demnach konstant sein, sodass

$$\mathrm{e}^{\mathrm{j}m\varphi} \stackrel{!}{=} \text{const.} \tag{4.8}$$

mit $m \in \mathbb{Z}$

$$m = 0 \quad . \tag{4.9}$$

Diese Annahme bestätigt sich bei der numerischen $\mathcal{SHT}$ des Schallfeldes einer idealen Linienquelle entsprechend (3.17), wie in Abbildung 4.5 zu sehen ist. Abbildung 4.6 zeigt nun die k-Spektren in z-Richtung mit numerischen und analytischen Ansatz. Die Länge der Linienquelle ist mit $l_z = 0.36$m so gewählt, dass später ein adäquater Vergleich mit dem realen Waveguide gelingt. Das spatial Aliasing in Abhängigkeit von Frequenz, modaler Ordnung und Quellgröße aus (3.29) ist mit den Ordnungen $N_1 = 17$ und $N_2 = 44$ verdeutlicht. Die Frequenz, ab der mit Spatial Aliasing gerechnet werden muss, ist mit der schwarz gestrichelten Linie markiert. Die Nullstellen der sinc-Funktion (3.14) wiederum sind rot markiert. Die beiden numerischen Lösungen liefern bis zur entsprechenden

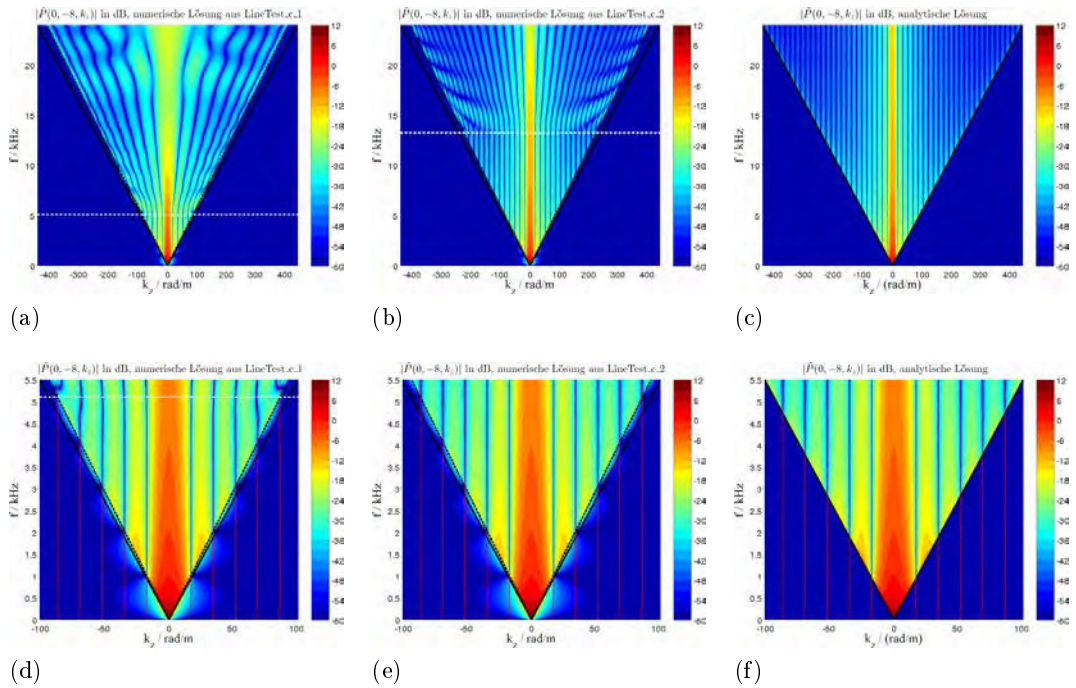


Abbildung 4.6.: $\tilde{P}_{line}(k_x, y, z, \omega)$ berechnet mit verschiedenen Ansätzen. (a) und (b) zeigen numerische Lösungen mit N 17 und N 44, (c) zeigt die analytische Lösung. Normiert auf $k_z = 0$ für $f = 1$kHz,

Aliasing-Frequenz (weiß) eine gute Näherung der analytischen Lösung. Oberhalb dieser Frequenz erkennt man deutlich Spatial Aliasing, das sich in den nicht mehr senkrechten Einschnitte der sinc-Funktion (rot) bemerkbar macht. Es zeigt sich aber, dass die in (3.29) qualitativ bestimmte Alias-Frequenz eine gute Schätzung darstellt. Dies wird für die spätere Betrachtung realer Lautsprecher eine wichtige Rolle spielen. Eine deutliche Abweichung der numerischen zur analytischen Lösung besteht allerdings im evaneszenten Bereich $k_z > \omega/c$ (also außerhalb des farbigen Dreiecks). Da sich die Expansion mit $r \geq 8$m für alle dargestellten Frequenzen im Fernfeld befindet, werden qua Definition keine nennenswerten evaneszenten Anteile erwartet. Ursache dafür ist die Fensterart und -länge der spatialen DFT. Abschnitt 3.4.3) wurden deren theoretische Grundlagen bereits besprochen. Anhand von Abbildung 4.7 kann nun der Einfluss des Fensters betrachtet werden. Alle Plots zeigen das k_z-f-Spektrum einer idealen Linienquelle der Länge $l_z = 0.36$m, das bis zur Ordnung $N_{max} = 17$ sphärisch harmonisch zerlegt und auf die Linie $\mathbf{L_z} = (x_{ref}, y_{ref}, z)$ mit $z = -L_z/2...L_z/2$ und $\Delta z = 0.005$m expandiert wurde. Zur besseren Sichtbarmarmachung der Störartefakte wurden die Spektren auf die Mittelachse $k_z = 0$ normiert. Bei den verschiedenen Plots wurde nun lediglich die Fensterart und die Länge der spatialen DFT variiert. Es sei hier noch einmal an den Zusammenhang aus (3.35) erinnert

$$N_{DFT} = \frac{L_z}{\Delta z}.$$

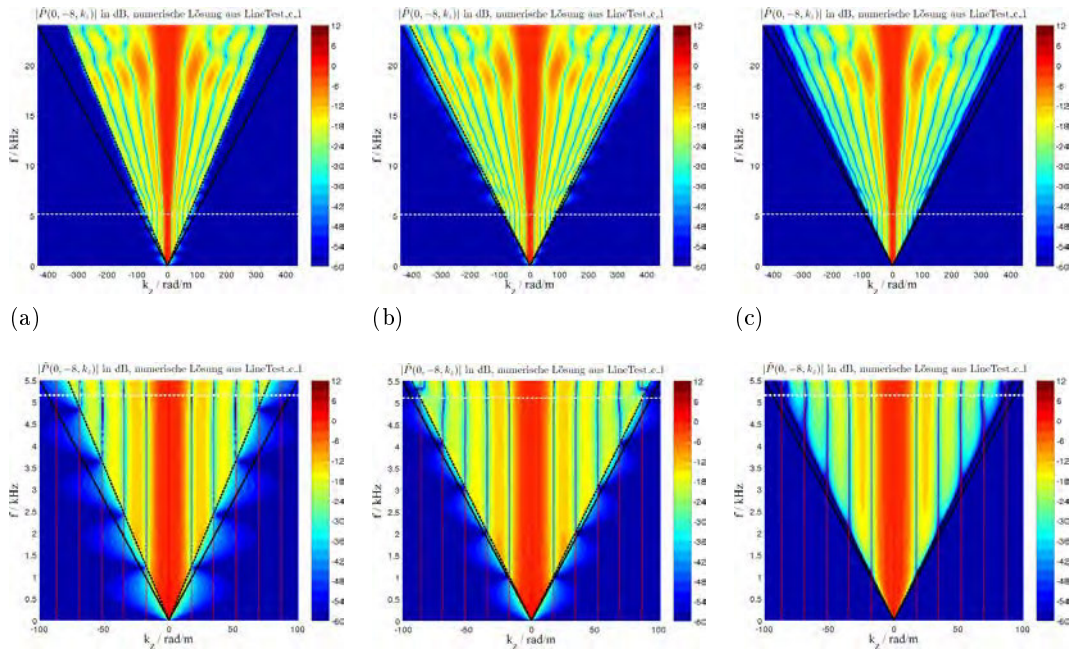


Abbildung 4.7.: $|\tilde{P}_{line}(x, y, k_z)|$ in dB berechnet mit verschiedenen Parametern der spatialen DFT. (a) $L_z = 20$m, $N_{DFT} = 4001$, win rect, (b) $L_z = 40$m, $N_{DFT} = 8001$, win rect, (c) $L_z = 40$m, $N_{DFT} = 8001$, win cheb80,

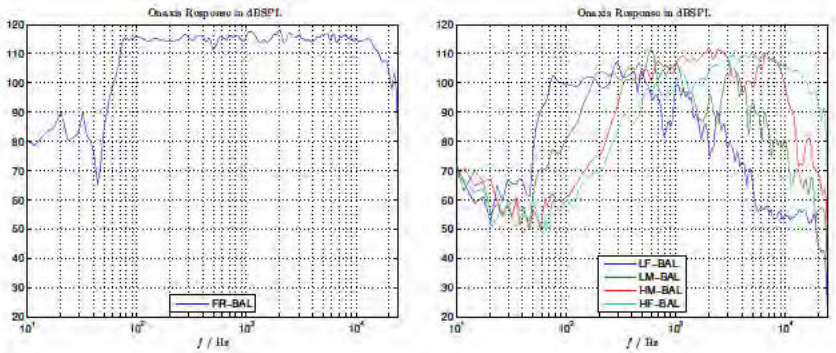
Die Verdoppelung von N_{DFT} von (a) mit $L_z = 20$m zu (b) $L_z = 40$m erzeugt eine schmalere Hauptkeule und zeigt schon wesentlich weniger Störartefakte beim Übergang von propagierendem zu evaneszentem Schallfeld. Bei beiden Plots wurden dabei ein Rechteckfenster verwendet. Außerdem wird erwartungsgemäß der Abstand zwischen $k_{z,max}$ und $k_z = \frac{\omega}{c}$ kleiner. Allerdings führt eine Erhöhung von N_{DFT} natürlich zu höherer Speicher- und Rechenauslastung. In (c) ist deshalb die spatiale DFT mit einem Chebyshev-Fenster mit $d = 80$dB Stoppdämpfung benutzt worden, während die $L_z = 40$m gleich geblieben ist. Dadurch konnte eine weitere Dämpfung der physikalisch nicht vorhandenen evaneszenten Anteile erreicht werden. Die Dämpfung darf hier allerdings nicht zu hoch eingestellt werden, da ansonsten auch propagierende Anteile im Bereich $k_z \to \frac{\omega}{c}$ davon betroffen werden. Die hier vorgestellten Werte $L_z = 40$m und $d = 80$dB stellen einen geeigneten Tradeoff zwischen Reduzierung von Artefakten und Rechenintensität dar und werden später bei der Auswertung der realen Balloon-Daten eingesetzt.

5. Ergebnisse und Diskussion

Nachdem in den vorangegangenen Kapiteln die Theorie für die Schallfeldanalyse und -synthese (Kap. 2) zu Grunde gelegt, die analytischen Referenzmodelle besprochen (Kap. 3) und die nötigen Werkzeuge in Matlab evaluiert (Kap. 4) wurden, kann nun der vorliegende Balloon-Datensatz untersucht werden. Dazu werden zunächst die Grenzen bezüglich Spatial Aliasing sowie Orts- und Frequenzauflösung festgelegt.

5.1. Gültigkeitsbereich des vorliegenden Datensatzes

Für den vorliegenden Lautsprecher liegen insgesamt 5 Datensätze vor (siehe Tabelle 4.1). Der erste FR-BAL liegt in der equiangularen Auflösung von $\Delta\varphi = \Delta\theta = 5°$ vor, der das gesamte Spektrum des Lautsprechers abbildet. Desweiteren liegen Datensätze für die 4 Einzelwege des Lautsprechers vor. Da die Richtcharakteristik mit steigender Frequenz zunimmt, sind hier besonders die Datensätze HM-BAL und HF-BAL mit $\Delta\varphi = \Delta\theta = 2°$ von Interesse. Abbildung 5.1 zeigt den Amplituden-Frequenzgang für die Hauptabstrahlrichtung $\varphi = -\pi/2,\ \theta = \pi/2$ der einzelnen Datensätze normiert auf 1W/1m entsprechend den Ausführungen in [Goe12]. Die Ergebnisse unterhalb der Grenzfrequenz des Messraumes (reflexionsarmer Halbraum, RWTH Aachen) von rund 90Hz sollten vernachlässigt werden. Die Messungen wurden laut Beschreibung in einem Abstand von $r = 8$m durchgeführt, was bei einer Quellgröße (Länge des Waveguides) von $l_z = 0.36$m nach (3.1) Fernfeldbedingungen für Frequenzen zwischen $f_u = 42$Hz und $f_o = 21172$Hz definiert. Mit der oberen Grenzfrequenz, die durch den maximalen Phasenfehler bestimmt ist,



(a) (b)

Abbildung 5.1.: On-Axis-Response der verschiedenen Datensätze normiert auf 1W/1m. (a) Full-Range (b) Einzelwege

	FR – BAL	HF – BAL
N_{max}	17	44
f_u	90 Hz	2000 Hz
f_o	5156 Hz	13344 Hz

Tabelle 5.1.: Grenzfrequenzen für die Ergebnisbewertung der relevanten Datensätze

kann dann auch der starke Einbruch des Spektrums im HF- und FR-Datensatz interpretiert werden. Für die Bewertung der Richtcharakteristik spielen desweiteren besonders Frequenzen oberhalb von 2kHz eine Rolle, die im HF-Datensatz ausreichend gleichmäßig vertreten sind. Für die vorliegende Kugelabtastung entsprechend Abschnitt 3.3 kann der Full-Range-Datensatz bis zur Ordnung $N_{max} = 17$ und der HF-Datensatz bis $N_{max} = 44$ modal zerlegt werden. Dabei kommt es mit (3.29) ab $f_{alias,17} = 5156$Hz bzw. $f_{alias,44} = 13344$Hz zu Spatial Aliasing. Zusammengefasst kann nun also gesagt werden dass die Ergebnisse lediglich in den Frequenzgrenzen in Tabelle 5.1 valide sind und außerhalb dieser Grenzen einzuordnende Störeinflüsse entstehen.

5.2. Analyse der Ausgangsdaten

Es liegt an dieser Stelle sicher nahe, sich zunächst die zugrunde liegenden Datensätze genauer anzusehen, um mögliche Fehlerquellen schon vor der modalen Zerlegung zu erkennen. Der erste Blick sei deshalb auf die Impulsantworten der einzelnen Datensätze (also Full-Range und Einzelwege) in Hauptabstrahlrichtung $\varphi = 270°$, $\theta = 90°$ gerichtet. Diese sind in Abbildung 5.2 als Ausschnitt der ersten 50ms gezeigt. Dabei fällt auf, dass die Datensätze verschiedene Laufzeiten beinhalten. Die Annahme wäre, dass bei einem

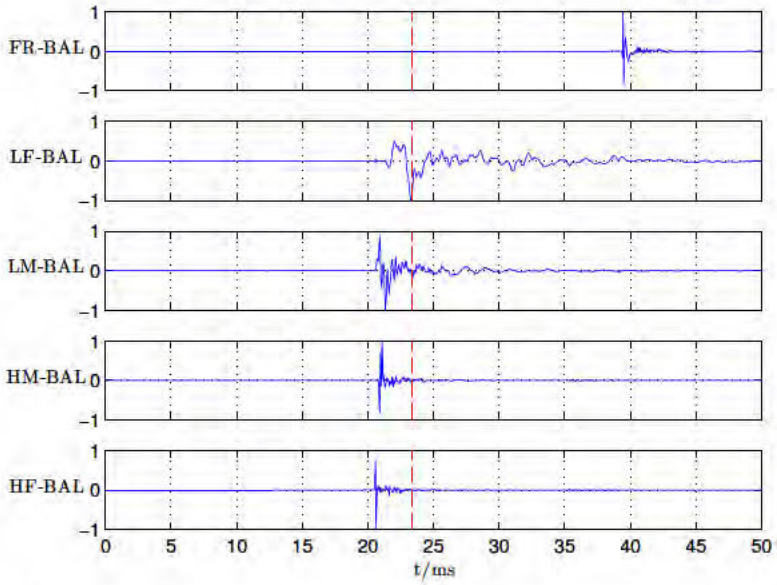


Abbildung 5.2.: Die ersten 50ms der Impulsantworten der 5 Datensätze in Hauptabstrahlrichtung. In rot ist die erwartete Laufzeit $t \approx 23.32$ms eingezeichnet

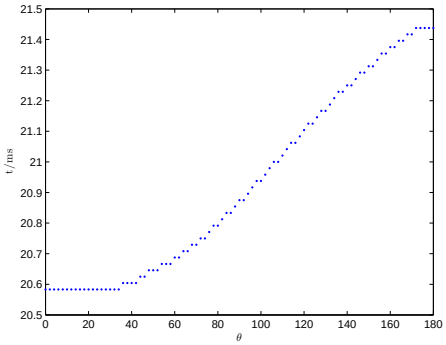


Abbildung 5.3.: Laufzeiten der Impulsantworten des HF-Datensatzes in Hauptabstrahlrichtung ($\varphi = 270°$)

Messabstand von $r_{meas} = 8$m und der Schallgeschwindigkeiten $c = 343\frac{\mathrm{m}}{\mathrm{s}}$ eine Laufzeit von $t \approx 23$ms erricht wird. Während die Einzelwege zumindest in der Größenordnung dieses Wertes liegen, ist der FR-BAL mit einer deulich höheren Laufzeit belegt. Dies weist darauf hin, dass bei der Bespielung des gesamten Lautsprechers noch Verzögerungen durch entsprechende FIR-Filter entstehen. Da uns leider keinerlei Informationen vorliegen, wie die Messkette genau ausgesehen hat, ist es im Nachhinein fast unmöglich Gründe dieser Laufzeit korrekt zu kompensieren. Aus diesem Grund müssen die Messungen des FR-BAL leider für die Analyse durch die $\mathcal{SHT}$ vernachlässigt werden. Doch auch die Einzelwege zeigen Laufzeiten, die nicht den Erwartungen entsprechen. Es ist an dieser Stelle also sinnvoll sich alle Laufzeiten eines Datensatzes anzuschauen. Dies ist exemplarisch für den HF-BAL für $\varphi = 270°$ und $\theta = 0...180°$ in Abbildung 5.3 gezeigt. Die Laufzeiten sind hier nicht konstant sondern zeigen Schwankungen, die sich im übrigen auch für andere Azimuthwinkel wiederholen. Der Grund dafür liegt wiederum im Messaufbau, bei dem das Mikrofon fest steht und der Lautsprecher entsprechend auf einem Dreharm geschwenkt wird. Durch das Abfahren von Azimuth und Elevation entstehen die abgebildeten Schwankungen. Daraus kann abgeleitet werden, dass sich das akustische Zentrum des Lautsprechers nicht in der Mitte der Abtastkugel befindet und sich darüber hinaus während der Messung bewegt. Die abgebildeten Schwankungen entsprechen Abweichungen des Messabstandes von rund ± 0.14m, was sich beim Radialterm der sphärischen Hankelfunktion $h_n(\frac{\omega}{c}r)$ weniger in Amplituden- als in Phasenfehlern niederschlägt. Für die modale Zerlegung ist dies natürlich äußerst problematisch. Wir stellen also fest, dass bei der Ergebnisauswertung der modalen Zerlegung der realen Ballon-Daten mit nicht unerheblichen Fehlern zu rechnen ist. Um trotzdem einige Vergleiche zwischen analystischem Modell und echtem Line Array System anzustellen, werden bei der modalen Zerlegung lediglich die Einzelwege betrachtet. Dabei wird die mittlere Laufzeit $t = 21$ms herangezogen, was mit $c = 343\frac{\mathrm{m}}{\mathrm{s}}$ einen mittleren Messabstand von $r_{meas} = 7.2$m ergibt. Von dieser Problematik weitestgehend unberührt bleibt die Betrachtug der Ausgangsdaten als Balloon-Plot. Im Abschnitt 3.2.3 haben wir ja bereits Prädiktionen für ideale Li-

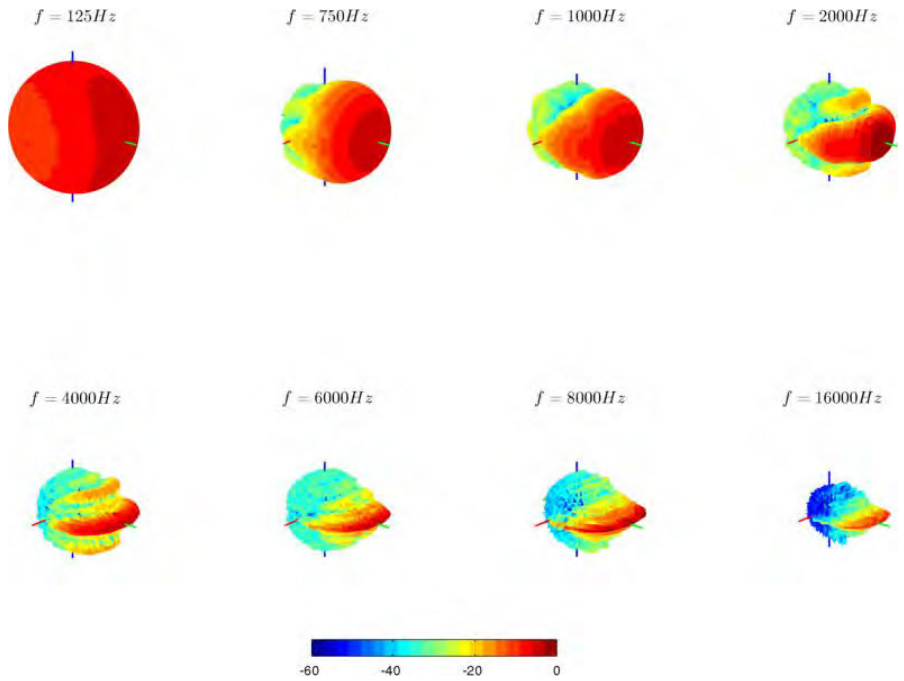


Abbildung 5.4.: Balloon-Plots in dB des Full-Range-Datensatzes für ausgewählte Frequenzen

nienquellen betrachtet. Daher ist es sicherlich interessant zu sehen, dass die Balloon-Plots des realen Lautsprechers in Abb. 5.4 und 5.6 dazu schon einige Ähnlichkeiten zeigen. Desweiteren zeigt diese Visualisierung der Messdaten, dass die Einlese- und Aufbereitungsroutine, beschrieben in Abschnitt 4.1, plausibel ist. Die Nullstellen der sinc-Funktion sind in diesen dreidimensionalen Plots schon zu erkennen, genauso wie die Zunahme der Richtwirkung mit steigender Frequenz. Deutlicher wird es jedoch, wenn man die vertikalen Isobaren in Hauptabstrahlrichtung betrachtet. Dies ist in Abbildung 5.5 dargestellt, zum Vergleich sind noch einmal die Ergebnisse aus Abschnitt 3.2.3 für eine ideale Linienquelle der gleichen Abmessung gezeigt.

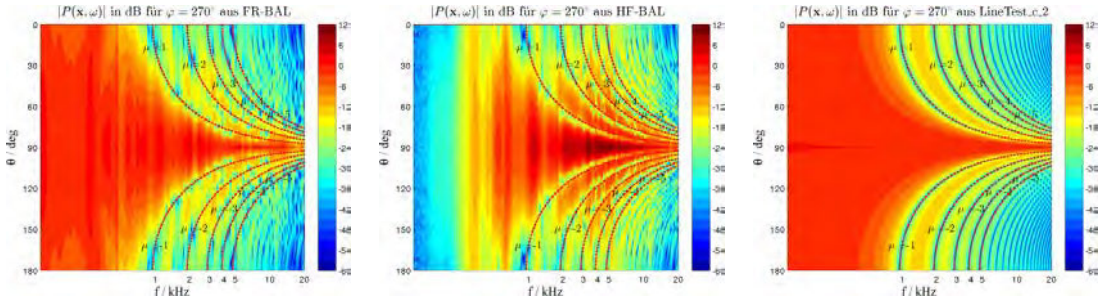


Abbildung 5.5.: Vertikale Isobaren in Hauptabstrahlrichtung der idealen Linienquelle (c), sowie des FR- (a) und HF-Datensatzes. Normiert auf $\theta = 90°$ und $f = 1\text{kHz}$

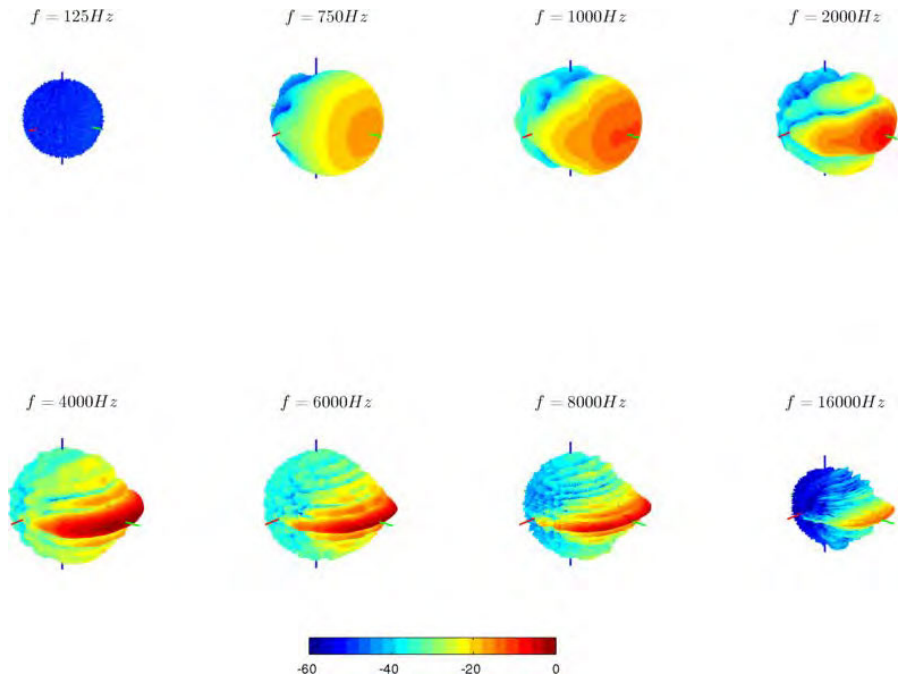


Abbildung 5.6.: Balloon-Plots in dB des High-Frequency-Datensatzes für ausgewählte Frequenzen

In den Isobaren-Plots sind die theoretischen Nullstellen der sinc-Funktion entsprechend(3.14) für

$$k_z = \mu \frac{2 * pi}{l_z} \qquad \text{mit} |\mu| = 1, 2, 3, ...$$

$$\theta = \arccos\left(\mu \frac{c}{f\ l_z}\right)$$

eingezeichnet. Es ist zu erkennen, dass der real gemessene Lautpsrecher diese Einschnitte auch zeigt, wenn auch nicht so sauber ausgetastet, wie im Modell. Hier ist auch noch einmal schön zu sehen, dass der HF-Datensatz erst ab Frequenzen oberhal von 2kHz seine maximale Energie beinhaltet. Diese Analyse der Rohdaten ist ein guter Hinweis darauf, dass die Prädiktion einer Linienquelle dem Lautsprecher nahe kommt.

5.3. Anwendung der modalen Zerlegung

Nachdem die vorliegenden Daten korrekt eingelesen wurden, kann nun die Zerlegung der Spektren ins diskrete Modalspektrum $\breve{P}_n^m(\omega)$ nach (4.7) vorgenommen werden. An dieser Stelle sei auch auf die Zeitkomplexität dieser Operation

$$\breve{P}_n^m = O(L \cdot N^2 \cdot \texttt{fft_size}) \tag{5.1}$$

hingewiesen, wobei L die Anzahl der Abtastpunkte auf der Kugeloberfläche und N die modale Ordnung der Zerlegung bezeichnen. Mit dem hier genutzten System benötigt allein die $\mathcal{SHT}$ für den HF-Datensatz rund 79 Minuten. Die berechneten Koeffizienten sind in den Abbildungen 5.7 dargestellt. Die Plots (a) und (b) zeigen die Koeffizienten des Modalspetrums bis zur Ordnung $N = N_{max}$ als Summe über die Moden $\breve{P}_n = \sum_m \breve{P}_n^m$. Hier erkennt man sofort, dass der reale Lautsprecher ein erheblich komplexeres Schallfeld produziert, als die ideale Linienquelle in Abb. 4.5. Mit steigender Frequenz nehmen auch die Anteile höherer Ordnung zu. Dies stimmt mit der Annahme überein, dass die Richtcharakteristik von Lautsprechern mit steigender Frequenz zunimmt. In (c) und (d) sind nun auch die einzelnen Moden exemplarisch jeweils bis zur Ordnung $N = 4$ dargestellt. Die Indizes der y-Achse bezeichnen jeweils $m = 0$. Man erkennt hier, dass die Verteilung der Koeffizienten in Abhängigketi von n und m einer gewissen Regelmäßigkeit entsprechen. So sind je nach Ordnung die hohen Moden $|m| \to n$ vergleichsweise unterrepräsentiert, während die Amplituden für $m = 0$ am höchsten sind. Letzteres spricht wieder für eine schwache Rotationsvarianz bezüglich der z-Achse, was wir auch schon für die ideale Linienquelle kennen gelernt haben. Sehr auffällig jedoch ist auch die komplette Abwesenheit der ungeraden Moden $|m| = 1, 3, \ldots$. Eine qualitative Erklärung dafür liefern die sphärischen Harmonischen in Abb. 2.4. Hier ist zu erkennen, dass für ungerade Moden keinerlei Rotationssymmetrie bezüglich einer der Achsen vorliegt. Dies liefert einen Hinweis, dass ein Schallfeld mit gewissen rotationssymmetrischen Bestandteilen auch eher die rotationssymmetrischen sphärischen Harmonischen anregt.

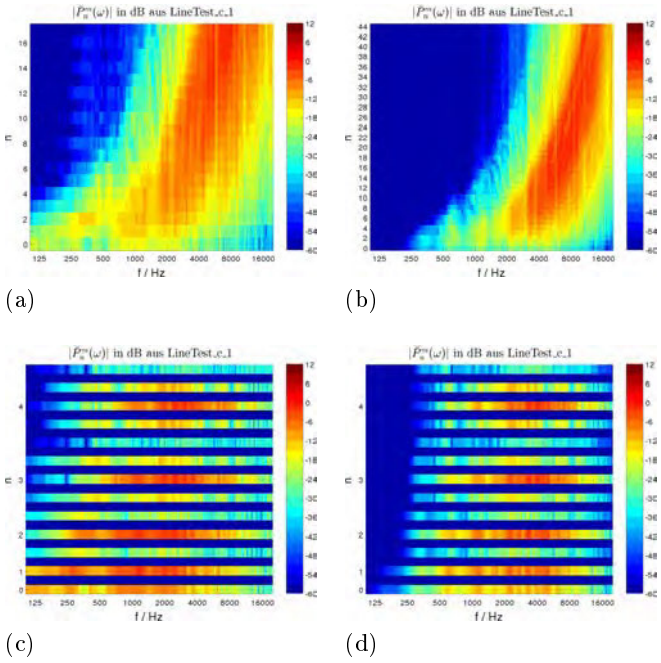


Abbildung 5.7.: $|\breve{P}_n^m(\omega)|$ des FR- (links) und HF-Datensatzes (rechts) als (a) und (b) Summe der Moden $\breve{P}_n = \sum_m \breve{P}_n^m$; (c) und (d) einzelne Moden bis $n = 4$

5.4. Schallfeldexpansion und k-Spectrum

Die Koeffizienten des Modalspektrums $\breve{P}_n^m(\omega)$ können nun mittels der $\mathcal{SHT}^{-1}$ auf beliebige Orte im Raum $\mathbf{x}$ zur Schallfeldexpansion genutzt werden. Wir betrachten, wie zuvor in den Abschnitten 2.7 und 4.2.7, das Schallfeld im Fernfeld auf der Linie $\mathbf{L_z} = (x = 0, y = -8, z = -20 \ldots 20)$ mit $\Delta z = 0.005$ (alle Angaben in m). Im Anschluss der Expansion wird das Schallfeld mittels spatialer Fouriertransformation in z-Richtung in das k_z-Spektrum überführt. Die Betrachtung der Ergebnisse findet wieder in zweierlei Hinsicht statt. Zum einen als Plot ebener Wellen in z-Richtung in Abhängigkeit von k_z und f und zum anderen mit der Umrechnung $k_z = \frac{\omega\{c}{\cos}(\theta)$ als Isobaren-Plot in Abhängigkeit von θ und f. Da bei der Schallfeldexpansion equidistante Δz und somit auch equidistante Δk_z gewählt wurden, entsteht bei der genannten Umformung ungleichmäßig verteilte $\Delta\theta$. Bei der Schallfeld-Expansion werden nun nur die hochaufgelösten HF- und HM-Datensätze betrachtet, da die anderen aus verschiedenen Gründen nicht geeignet sind (siehe oben). Abbildung 5.8 zeigt zunächst die Plots in Abhängigkeit von k_z und f für den HM- und

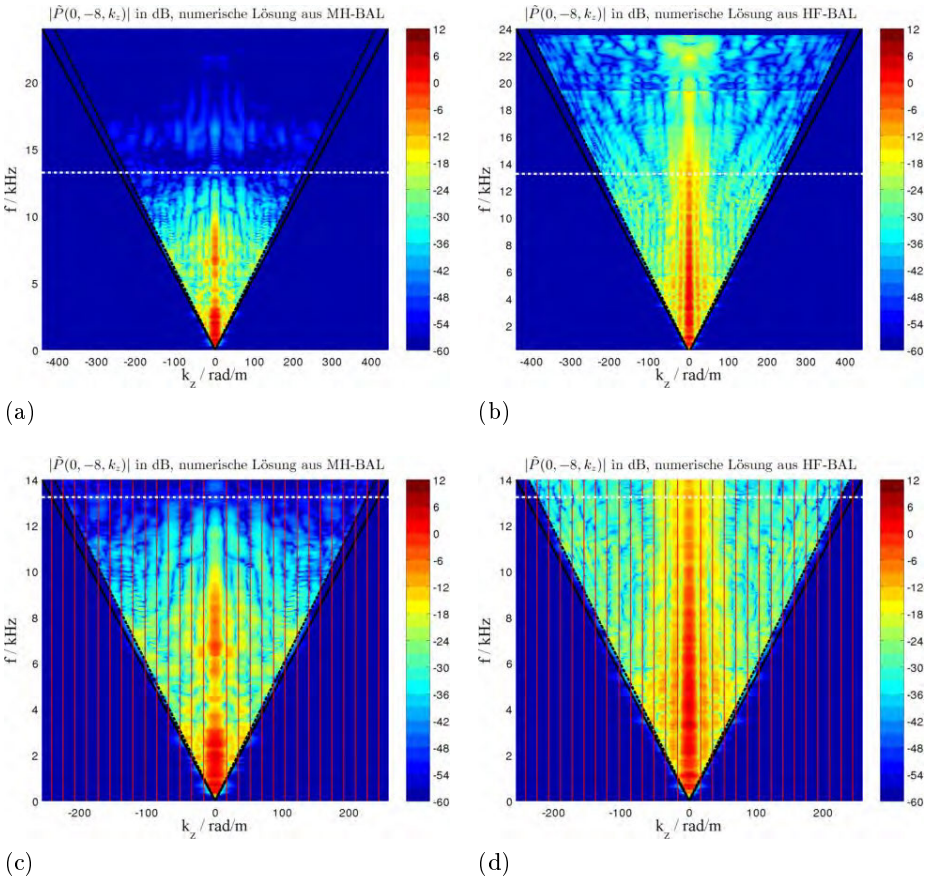
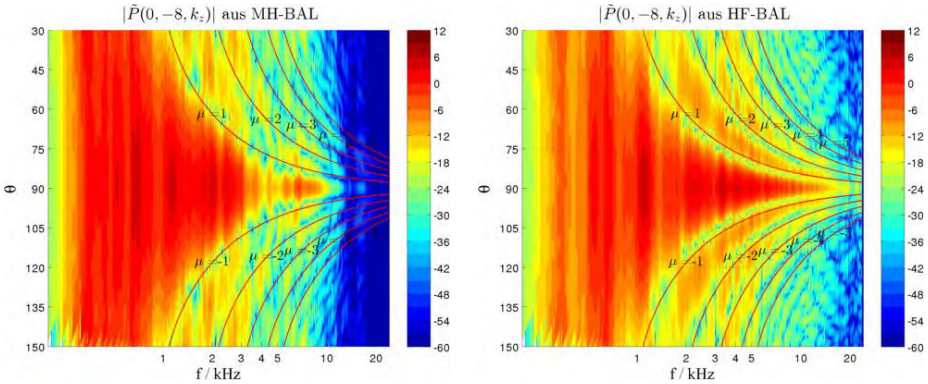


Abbildung 5.8.: Ergebnisse der Schallfeldexpansion und spatialen Fouriertransformation $|P(0,\tilde{-}8,k_z)|$ für den HM- und HF-Datensatz im kompletten Frequenzspektrum (oben) und als Ausschnitt bis zur analyseseitigen AliasFrequenz (unten)



(a) (b)

Abbildung 5.9.: Ergbenisse der Schallfeldexpansion und spatialen Fouriertransformation $|\tilde{P}(0,-8,k_z)|$ in dB als Isobaren-Plot in Abhängigkeit von θ für (a) den HM- und (b) den HF-Datensatz

HF-Datensatz. Dabei ist sowohl der komplette Frequenzbereich abgebildet, als auch der Ausschnitt bis zur analyseseitigen Aliasfrequenz entsprechend 5.1. Es zeigt sich, dass die theoretische Prädiktion der Einschnitte durch die Nullstellen der sinc-Funktion zumindest örtlich gelingt. Allerdings ist deren Verlauf im Vergleich zur idealen Linienquelle in Abb. 4.6 deutlich unsauberer. Es ist außerdem ersichtlich, dass die Vorhersage von $k_{z,max}$ unabhängig vom Datensatz bei der Schallfeldexpansion gelingt, was durch die gestrichelte schwarze Linie markiert ist. In Abbildung 5.9 ist das Ergebnis nun noch mal als Isobaren-Plot in Abhängigkeit von θ aufgetragen. Es zeigt sich zunächst, dass diese Darstellungsform weniger gut geeignet ist, da es wegen der ungleichmäßig verteilten $\Delta\theta$ zu rein optischen Verzerrungen kommt. Dennoch ist auch hier das Abtrahlverhalten des Lautsprecher gut zu sehen. Besonders der High-Frequency-Datensatz in (b) zeigt gut die Nullstellen der sinc-Funktion. Dass dies beim High-Mid-Frequency-Datensatz nicht so stark ausgeprägt ist, liegt hauptsächlich am wiedergegebenen Frequenzgang dieses Weges (siehe Abb. 5.1). Trotzdem ist auch hier besonders Abfall in der ersten Nullstelle gut zu erkennen.

5.5. Zusammenfassung

In dieser Arbeit wurden zwei verschiedene Ansätze der Schallfeldanalyse und -prädiktion kombiniert. Zunächst wurde die *Spherical Harmonic Transform* $\mathcal{SHT}$ zur Beschreibung von Schallfeldern erklärt. Dabei wurden die mathematischen Grundlagen herangezogen und die Grenzen bei einer praktikablen Realisierung diskutiert. Es wurde eine Implementierung in Matlab geschaffen und vorgestellt, die diese Transformationen für äußere Probleme ermöglicht. Damit wurde ein realer Balloon-Datensatz komplexer Spektren eines Line Array Lautsprechers zunächst in sein Modalspektrum $\check{P}_n^m(\omega)$ zerlegt und dieses anschließend auf einen definierten aber beliebig wählbaren Ort im Raum expandiert. Dieses virtuell synthetisierte Schallfeld wurde im Wellenzahlenraum als $\tilde{P}(x_{ref}, y_{ref}, k_z, \omega)$ dargestellt, da dies einige Eigenschaften hervorbringt, die mit dem zweiten Ansatz der Schallfeldanalyse zu tun haben.
Die Wahl eines Line Array Systems beruhte nämlich auf der Annahme, dass diese konstruktionsbedingt eine (ideale) Linienquelle nachahmen sollen. Um diese Annahme auf ihre Richtigkeit hin zu überprüfen, wurde als zweiter Ansatz der Schallfeldanalyse und -prädiktion das wohl definierte Verhalten von Linienquellen erläutert und diskutiert.
Zum Schluss wurden nun diese beiden Ansätze miteinander verglichen. Die Frage der Untersuchung war dabei, ob das emittierte Schallfeld eines Line Array Systems, beschrieben und bearbeitet mit der $\mathcal{SHT}$, und die analytische Prädiktion von Linienquellen Ähnlichkeiten besitzen.
Es zeigte sich zunächst, dass die Zerlegung und Resynthese des Schallfeldes mittels $\mathcal{SHT}$ in den definierten Grenzen mit der vorliegenden Implementierung gut gelingt. Die Grenzen hängen dabei maßgeblich vom vorliegenden Datensatz ab, konkret von der räumlichen Diskretisierung des Schallfeldes und der damit möglichen sphärischen Ordnung N_{max}. Es konnte gezeigt werden, dass mit der vorliegenden equiangularen Kugelabtastung von $\Delta\phi = \Delta\theta = 2°$ und der resultierenden Ordnung $N_{max} = 44$ lediglich der Frequenzbereich bis etwa 13kHz korrekt analysiert werden konnte. Im zweiten Schritt konnte desweiteren gezeigt werden, dass der betrachtete Lautsprecher die Eigenschaften einer idealen Linienquelle in den genannten Grenzen nachvollziehbar abbildet. Dies macht sich insbesondere in den übereinstimmenden Einschnitten der sinc-Funktion bemerkbar. Auch die Art der „Einschnürung“ der Richtcharakteristik mit steigender Frequenz stimmt mit der Prädiktion von Linienquellen überein. Allerdings war die Interpretation der Messdaten nicht immer eindeutig. Bei den auftretenden Artefakten konnte nicht immer genau gesagt werden, ob sie aus der angewandten Methodik oder dem zugrunde liegenden Messdatensatz entstanden sind. Denn es hat sich gezeigt, dass an den Messdatensatz sehr hohe Ansprüche bezüglich Auflösung und Positionierungsgenauigkeit gestellt werden. Zunächst ist eine equiangulares Abtastung nicht besonders effizient bezüglich des Verhältnisses von Anzahl der Abtastpunkte L zur maximal möglichen sphärischen Ordnung N_{max}. Andere Abtaststrategien wie etwa die Lebedev-Quadratur bieten hier höhere Effizienz. Darüber

hinaus muss sich beim Positionieren von Schallquelle und Mikrofon weiterführende Gedanken zum akustischen Mittelpunkt der Quelle und dem resultierenden Radius zwischen Quelle und Mikrofon gemacht werden. Ein Ansatz hierzu ist in [Sha14] zu finden. Wie sich gezeigt, muss auch der Einfluss des Messsystems auf die komplexen Spektren bekannt sein bzw. hinreichend eliminiert werden, um Laufzeit- und Phasenfehler zu vermeiden.

5.6. Ausblick

Aus den vorgestellten Ergebnissen kann nun abgeleitet werden, was dieser Arbeit methodisch folgen könnte. Auf der einen Seite wäre es sinnvoll und interessant, die Matlab-Implementierung auf andere Datensätze anzuwenden, die den Ansprüchen der $\mathcal{SHT}$ besser gerecht werden. Mögliche Messanordnungen wären etwa Mikofon-Arrays wie in [Bru07] oder gegenüber Positionierungsfehler robustere Einzel-Messverfahren. Auf der anderen Seite hat die Matlab-Implementierung noch weiteren Entwicklungsbedarf. So sollte der Code in Zukunft noch auf allgemeinere Problemstellungen hin erweitert werden. Zwar ist die Implementierung schon sehr modular realisiert, dennoch ist etwa die Fokussierung auf äußere Probleme und equiangulare Kugelabtastung erkennbar. Dabei soll auch die Nutzerfreundlichkeit durch ein sinnvolles Front-End erhöht werden, z.B. durch eine globale Konfigurationsdatei oder einem (grafischen) User-Interface. Ein weiterer wichtiger Punkt ist die Optimierung hinsichtlich der Performanz. Der hier vorgestellte Code ist weitesgehend straightforward implementiert, d.h. dass kaum Unterscheidungen des Input und der Problemstellung angewendet werden. Dies hat zunächst den Vorteil, dass die Implementierung gut nachvollziehbar und evaluierbar ist. Ein großer Nachteil liegt allerdings in der notwendigen Rechenkapazität. In einer späteren Weiterentwicklung des Codes ist es demnach sinnvoll, die Anzahl der Rechenschritte zu reduzieren. Mögliche Ansätze zur Reduzierung von Redundanz und Irrelevanz sind etwa

- die Weiterführende Ausnutzung von Symmetrien bei Analyse und Synthese des Schallfeldes,
- die Nutzung der Fernfeld-Näherung der sphärischen Hankelfunktion
- $\breve{P}_n^m(\omega)$ mit geringer Amplitude bei der Synthese zu vernachlässigen.

Zum Schluss nun noch einige Gedanken zur Nutzbarkeit der Ergebnisse. Obwohl die hier vorgestellte Methodik hauptsächlich auf etabliertes Wissen der Schallfeldanalyse und -synthese zurückgreift, kann der Herangehensweise in dieser Arbeit doch einiger praktikabler Nutzen im Bereich Forschung und Entwicklung entnommen werden. Die Ergebnisse können beispielsweise bei Lautsprechermessungen und -simulationen eingesetzt werden. Dort werden momentan oft Balloon-Daten mit je nach Hersteller unterschiedlichen Konfigurationen bereitgestellt. In der Respräsentation dieser Daten durch deren Modalspek-

tren $\breve{P}_n^m$ kann hier eine flexible Handhabung erreicht werden. Natürlich muss dann die Messung an die Eigenschaften des vorgestellten Verfahrens sinnvoll angepasst werden. Auf der anderen Seite liefert die Darstellung des Schallfeldes im Wellenzahlenraum eine gute Möglichkeit, Richtcharakteristiken von Lautsprechern im Sinne der Ausbreitung von ebenen Wellen zu visualisieren.
Es sei noch einmal erwähnt, dass der vorgestellte Programmcode nicht auf die Messung von Line Array Systemen beschränkt ist, sondern zur Analyse jeglicher Lautsprecher und Schallquellen im Allgemeinen bezüglich äußerer Probleme verwendet werden kann. In diesem Sinne kann diese Arbeit einen Beitrag im Bereich der Schallfeldsynthese mit Sekundärquellen wie etwa Wellenfeldsynthese oder Higher-Order Ambisonics leisten. Denn bei diesen Technologien tragen die verwendeten Lautsprecher erheblich zum Ergbenis teil, weshalb es nötig ist, deren Eigenschaften bezüglich ihrer Schallabstrahlung so gut wie möglich zu kennen. Auch hier ist eine Reprßentation des Schallfeldes als Modalspektrum sinnvoll.
Eine artverwandte analytische Untersuchung von Line Array Systemen durch das Single Layer Potential ist desweiteren bereits in [SRS14] zu finden. Nun ist es sicherlich interessant die dort gefundenen analytischen Ergebnisse mit realen Messergebnissen zu vergleichen, die mit der vorgestellten Implementierung analysiert wurden.

Abbildungsverzeichnis

2.1. Koordinatensystem 4
2.2. Propagierende und evaneszente Wellenausbreitung 7
2.3. Sphärische Bessel-, Neumann- und Hankelfunktion 9
2.4. Sphärische Harmonische $|\Re\{Y_n^m(\theta,\varphi)\}|$ für $n = 0 \ldots 4$. 11
2.5. Inneres und äußeres Problem 12
2.6. Ebene Wellen als $k_z - f$-Spektrum 15
2.7. Schematische Darstellung Expansionslinie 16

3.1. Schematische Darstellung eines Rect. Piston 20
3.2. Directivity Balloons Rect. Piston 23
3.3. Horizontale und Vertikale Isobaren RectPiston 23
3.4. Schematische Darstellung einer Linienquelle 24
3.5. Directivity Balloons Linienquelle 25
3.6. Abschätzung $min\{n\}$ der Besselfunktion 29
3.7. Beispiele für Fensterung der DFT 31

4.1. Flussdiagramm der Implementierung in Matlab 32
4.2. Orientierung der Kugelabtastung 33
4.3. Evaluierung verschiedener Abtastgrids 38
4.4. Numerische und analytische Lösungen der Green'schen Funktion im k-space 42
4.5. Modalspektrum Linienquelle 43
4.6. PWD idealer Linienquellen mit verschiedenen Ansätzen 44
4.7. PWD idealer Linienquellen mit verschiedenen DFT-Fenstern 45

5.1. Amplituden-Frequenzgang der verschiedenen Datensätze 46
5.2. Impulsantworten der verschiedenen Datensätze 47
5.3. Laufzeiten der Impulsantworten des HF-Datensatzes 48
5.4. Directivity Balloons FR-BAL 49
5.5. Vergleich der vertikalen Isobaren 49
5.6. Directivity Balloons FR-BAL 50
5.7. Modalspektrum $\check{P}_n^m$ des FR- und HF-Datensatzes 51
5.8. PWD des HM- und HF-Datensatzes 52
5.9. Vertikale Isobaren des HM-und HF-Datensatzes nach der Bearbeitung . . 53

A.1. Funktions-Diagramm 62

Tabellenverzeichnis

2.1. Eigenlösungen der der Wellengleichung in Kugelkoordinaten 8
2.2. Sphärische Harmonische 10

3.1. Vergleich der maximalen Ordnung N_{max} verschiedener Abtastgrids 28
3.2. Parameter spatiale DFT 31

4.1. Dimensionen der vorliegenden Datensätze 34
4.2. Beispiel Lookup-Table 35
4.3. Bandbegreenzung und Aliasing der Datensätze 35
4.4. Implementierung zeitlicher und örtlicher DFT 36
4.5. Felder des structured Arrays 38
4.6. Anzahl der Abtastpunkte L für $N = 17$ 39

5.1. Grenzfrequenzen der Datensätze 47

Literaturverzeichnis

[Ahr12] J. Ahrens. *Analytic Methods of Sound Field Synthesis.* Springer, Berlin, Heidelberg, 2012.

[AS08] J. Ahrens and S. Spors. An Analytical Approach to Sound Field Reproduction Using Circular and Spherical Loudspeaker Distributions. *Acta Acustica united with Acustica*, 94(6):988–999, November 2008.

[AS10a] J. Ahrens and S. Spors. On the anti-aliasing loudspeaker for sound field synthesis employing linear and circular distributions of secondary sources. In *Proc. of the 129th AES Convention*, San Francisco, CA, 2010.

[AS10b] J. Ahrens and S. Spors. Sound Field Reproduction Using Planar and Linear Arrays of Loudspeakers. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):2038–2050, November 2010.

[AW70] G. B. Arfken and H. J. Weber. *Mathematical methods for physicists.* Academic Press, San Diego, CA, 2. ed. edition, 1970.

[Bla00] D. T. Blackstock. *Fundamentals of Physical Acoustics.* John Wiley & Sons, 2000.

[BPSW11] B. Bernschütz, C. Pörschmann, S. Spors, and S. Weinzierl. SOFiA Sound Field Analysis Toolbox. In *Proc. of the International Conference on Spatial Audio*, Detmold, 2011.

[Bru07] V. Brunel. *Measurement and spherical harmonic representation of loudspeaker radiation pattern.* Master Thesis, University of Southampton, 2007.

[DH94] J. R. Driscoll and D. M. Healy. Computing Fourier transforms and convolutions on the 2-sphere. *Advances in applied mathematics*, 15(2):202–250, 1994.

[DLMF] NIST Digital Library of Mathematical Functions. http://dlmf.nist.gov/, Release 1.0.5 of 2012-10-01. Online companion to [OLBC10].

[FAB05] S. Feistel, W. Ahnert, and S. Bock. New Data Format to Describe Complex Sound Sources. In *Audio Engineering Society Convention 119*, 2005.

[Faz10] F. M. Fazi. *Sound Field Reproduction.* PhD thesis, University of Southampton, 2010.

[FBN+08] F. M. Fazi, V. Brunel, P. A. Nelson, L. Hörchens, and J. Seo. Measurement and Fourier-Bessel Analysis of Loudspeakers Radiation Patterns Using a Spherical Array of Microphones. In *Proc. of the 124th AES Convention*, Amsterdam, NL, 2008.

[FN12] F. M. Fazi and P. A. Nelson. Nonuniqueness of the Solution of the Sound Field Reproduction Problem with Boundary Pressure Control. *Acta Acustica united with Acustica*, 98(1):1–14, January 2012.

[GD04] N. A. Gumerov and R. Duraiswami. *Fast multipole methods for the Helmholtz equation in three dimensions.* Elsevier, Oxford, 2004.

[Goe12] A. Goertz. Turbosound Flashline. *Production Partner*, 3:128–139, March 2012.

[HU92] C. Heil and M. Urban. Sound Fields Radiated by Multiple Sound Sources Arrays. In *Proc. of th 92nd AES Convention*, Vienna, 1992.

[Leb76] V. I. Lebedev. Quadratures on a sphere. *USSR Computational Mathematics and Mathematical Physics*, 16(2):10–24, January 1976.

[LS92] V. I. Lebedev and A. L. Skorokhodov. Quadrature-rules for a sphere of 41-order, 47-order and 53-order of accuracy. *Doklady Akademii Nauk*, 324(3):519–524, 1992.

[MÖ9] M. Möser. *Technische Akustik.* Springer, Berlin, Heidelberg, 8. edition, 2009.

[Mat13] MATLAB Function Reference R2013b, 2013.

[OLBC10] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, editors. *NIST Handbook of Mathematical Functions.* Cambridge University Press, New York, NY, 2010. Print companion to [DLMF].

[Pol05] M. A. Poletti. Three-dimensional surround sound systems based on spherical harmonics. *Journal of the Audio Engineering Society*, 53(11):1004–1024, 2005.

[Raf05] B. Rafaely. Analysis and design of spherical microphone arrays. *IEEE Transactions on Speech and Audio Processing*, 13(1):135–143, January 2005.

[RWB07] B. Rafaely, B. Weiss, and E. Bachmat. Spatial Aliasing in Spherical Microphone Arrays. *IEEE Transactions on Signal Processing*, 55(3):1003–1010, March 2007.

[Sha14] N. R. Shabtai. Acoustic Centering for High-Order Source Directivities. In *Fortschritte der Akustik - DAGA*, 2014.

[SRS14] F. Schultz, T. Rettberg, and S. Spors. On Spatial-Aliasing-Free Sound Field Reproduction using Infinite Line Source Arrays. In *Proc. of the 136th AES Convention*, Berlin, 2014.

[UHB03] M. Urban, C. Heil, and P. Bauman. Wavefront sculpture technology. *Journal of the Audio Engineering Society*, 51(10):912–32, 2003.

[Ure04] M. S. Ureda. Analysis of loudspeaker line arrays. *Journal of the Audio Engineering Society*, 52(5):467–95, 2004.

[WA80] G. Weinreich and E. B. Arnold. Method for measuring acoustic radiation fields. *The Journal of the Acoustical Society of America*, 68(2):404–411, 1980.

[Wei08] S. Weinzierl, editor. *Handbuch der Audiotechnik*. Springer, Berlin, Heidelberg, 2008.

[Wil99] E. G. Williams. *Fourier Acoustics: Sound Radiation and Nearfield Acoustic Holography*. Elsevier, London, 1999.

[Zot09] F. Zotter. *Analysis and synthesis of sound-radiation with spherical arrays*. PhD thesis, Universität für Musik und Darstellende Kunst Graz, 2009.

[ZPN10] F. Zotter, H. Pomberger, and M. Noisternig. Ambisonic decoding with and without mode-matching: A case study using the hemisphere. *Proc. of the 2nd International Symposium on Ambisonics and Spherical Acoustics*, 2010.

A. Matlab Code

Im folgenden ist der beschriebene Matlab-Code angehängt. Zunächst sind die ausführbaren Hauptskripte aufgeführt (A.1), anschließend die nötigen Werkzeuge dazu (A.2) und einige Plot-Routinen (A.3). In Abbildung A.1 sind die Abhängigkeiten der Dateien als Funktionsdiagramm dargestellt. Um Verwirrungen vorzubeugen sei darauf hingewiesen, dass die Dateinamen hier im Anhang sowohl in den jeweiligen Datei-Headern als auch unter den Listings zu finden ist.

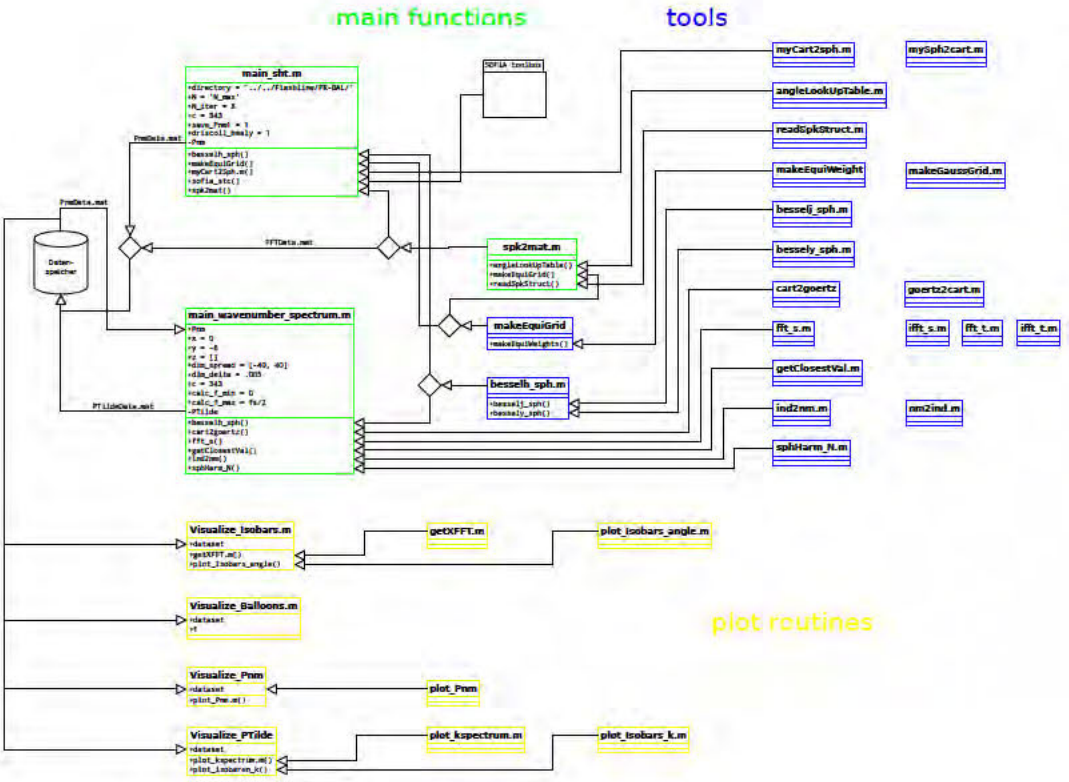


Abbildung A.1.: Funktions-Diagramm

A.1. main functions

```
%%%
```

```
        FFT_Data N_max = N_max;
    else
        warning('Equiangular grid according to Driscoll and Healy (1994) is not executable with this particular dataset ');
    end
end

% maximum order for spherical transformation
if strcmp(N, 'N_max')
    N = N_max;
elseif N<=N_max
    N = N;
else
    error('N must be set either to string ''N_max'' or in the range N<=N_max ')
end

% random stuff
nvector = ind2nm(1 (N+1)^2);
f = linspace(0,fs/2,fft_size);
k = 2*pi*f/c;

tVariables = round(toc(tVariables)*100)/100;
disp(tVariables)
pause(1);
%%  SHT
disp('Spherical Harmonic Transform')
tSHT = tic;
Pnmr = zeros((N+1)^2,fft_size);
for n_iter = 1 N_iter

    fft_start = (n_iter-1)*iter_size + 1;
    if n_iter == N_iter
        fft_end = fft_size;
    else
        fft_end = n_iter*iter_size;
    end
    Pnmr( ,fft_start fft_end) = sofia_stc(N, XFFT(angle_LUT,fft_start fft_end), grid_full);
end
tSHT = round(toc(tSHT)*100)/100;
dis(tSHT);
pause(1);
%% Eliminating radial component
tRadial = tic;
disp('Eliminating radial Component  Pnmr -> Pnm')

[kr_meas n] = meshgrid(k*r_meas, 0 N);
hna = besselh_sph(n, 2, kr_meas);
radial = hna(nvector+1, );

Pnm = Pnmr /radial;

clear kr_meas Pnmr radial hna

tRadial = round(toc(tRadial)*100)/100;
disp(tRadial);
pause(1)
%% Visualization
tVisualization = tic;
disp('Visualization')

hfig = plot_Pnm(Pnm,
    f,
    2,
    'max',
    [100 20000],
    17,
    4,
     36,
    {'$f$ / Hz', 'n'},
    ['$\breve{P}_{n}^m(\omega)$ aus ' dataset, ' f\"ur N_{max}=' num2str(N)],
    false);

tVisualization = round(toc(tVisualization)*100)/100;
disp(tVisualization);
%% Save and Setoff
tTotal = round(toc(tTotal)*100)/100;
```

```
disp(tTotal);

PnmData = struct;
PnmData FFTData          = dataset;
PnmData FFTData_path     = [cd filesep 'FFTData_' dataset ' mat'];
PnmData N                = N;
PnmData N_iter           = N_iter;
PnmData DriscollHealy    = DriscollHealy;
PnmData CPUTime          = {'Variables','SHT','Radial','Visual','Total';
    tVariables,tSHT,tRadial,tVisualization,tTotal};
PnmData Pnm              = Pnm;


if save_Pnm
    save([datestr(now, 'yymmddHH') '_Pnm_' dataset '_N' num2str(N) ' mat'],
        '-struct', 'PnmData', '-v7 3');
end

end_str = datestr(now, 'yy-mm-dd␣HH MM SS');
disp(['End␣time␣is ␣' end_str]);
```

Listing A.1: main_sht.m

```
%%%
```

```
if nargin < 3
    sph_orientation = 'Goertz';
end
%% Folder properties
disp('Folder properties   ')
if filesep=='\'
    path_spk(path_spk=='/')=filesep;
elseif filesep=='/'
    path_spk(path_spk=='\')=filesep;
end

if ~strcmp(path_spk(end), filesep)
    path_spk = [path_spk filesep];
end

dataset = regexp(path_spk, filesep, 'split');

dataset = dataset{end-1};

if exist(path_spk, 'dir')~=7
    error('Path does not exist ')
end

files = dir(fullfile(path_spk));

filecount = 0;
for i =1 length(files)
    [~, file_tmp, ext_tmp] = fileparts(files(i) name);
    v_tmp = regexp(file_tmp, 'V\d{3}', 'match');
    h_tmp = regexp(file_tmp, 'H\d{3}', 'match');

    if ~isempty(v_tmp) && ~isempty(h_tmp)
        filecount = filecount+1;
        filenames{filecount}=file_tmp;
        ext{filecount}=ext_tmp;

        if strcmp(sph_orientation, 'Goertz')
            angles(filecount,1) = str2double(strrep(v_tmp, 'V', ''));
            angles(filecount,2) = str2double(strrep(h_tmp, 'H', ''));
        elseif strcmp(sph_orientation, 'standard')
            angles(filecount,1) = str2double(strrep(h_tmp, 'H', ''));
            angles(filecount,2) = str2double(strrep(v_tmp, 'V', ''));
        end
    end
end

if length(unique(ext))==1
    ext_tmp = unique(ext);
    ext = ext_tmp{1};
else
    error('Extensions must be consistent ')
end

n_h = length(unique(angles( ,1)));
n_v = length(unique(angles( ,2)));

step_h = diff(unique(angles( ,1)));
step_v = diff(unique(angles( ,2)));

if length(unique(step_h))==1 && length(unique(step_v))==1
    equi = true;
    step_h = unique(step_h);
    step_v = unique(step_v);
else
    equi = false;
end

sph_coverage(1) = 360/max(angles( ,1));
sph_coverage(2) = 180/max(angles( ,2));

fname_tmp = [path_spk filenames{1} ext];
tmp = readSPKStruct(fname_tmp);
fs = tmp SamplingRate;
X = tmp F;
% [fs, X] = read_spk(fname_tmp);
fft_size = length(X);
```

```
disp('check')
%% Read SPK
disp('read␣ SPK␣files   ')
tic
XFFT = zeros(length(angles), fft_size);

if exist(['FFTData_' dataset ' mat'],'file')==2   % check if measurement already exists in current folder
    exit_flag = 0;
    while ~exit_flag
        answer = input(['FFTData_' dataset ' mat␣already␣exists ␣Overwrite?␣Y/N ␣'], 's');
        if strcmpi(answer, 'y')
            read_flag = 1;
            exit_flag = 1;
        elseif strcmpi(answer, 'n')
            read_flag = 0;
            exit_flag = 1;
            load(['FFTData_' dataset ' mat']);
        else
            exit_flag = 0;
        end
    end
else
    read_flag = 1;
end

if read_flag
    for n = 1 length(filenames)

        fname_tmp = [path_spk filenames{n} ext];

        %[fs, X] = read_spk(fname_tmp);
        tmp = readSPKStruct(fname_tmp);
        X = tmp F;

        if size(X,1)>size(X,2)
            X = X';
        end

        XFFT(n, ) = X;
    end
end
mymax = max(max(abs(XFFT)));

XFFT = XFFT /mymax;
clear X;
toc
disp('check')
%% Save strut to  mat
disp('save␣to␣struct   ')
if equi
    grid_orig = makeEquiGrid([step_h, step_v], 'plain', max(angles));
    grid_full = makeEquiGrid([step_h, step_v], 'plain');
    resol = [step_h, step_v];
    angle_LUT = angleLookUpTable([n_h, n_v], sph_coverage, [1,1]);
else
    % TODO
   disp('TODO');
end

FFTData = struct();
FFTData dataset = dataset;
FFTData path = [pwd filesep 'FFTData_' dataset ' mat'];
FFTData XFFT = XFFT;
FFTData grid_orig = grid_orig;
FFTData grid_full = grid_full;
FFTData angle_LUT = angle_LUT;
FFTData equi = equi;
FFTData resol = resol;
FFTData r_meas = r_meas;
FFTData fs = fs;
FFTData fft_size = fft_size;
FFTData sph_coverage = sph_coverage;
FFTData sph_orientation = sph_orientation;

save(['FFTData_' dataset ' mat'], '-struct', 'FFTData', '-v7 3');

```

```
disp('check')
end
```

Listing A.2: spk2mat.m

```
%%%
```

```
coord{dimension} = dim_spread(1) dim_delta dim_spread(2);

N_dim = length(coord{dimension}); % number of sampling points
[az, el, r] = cart2sph(coord{1},coord{2},coord{3});
el = pi/2 -el;
if all(size(az)==[1 1])
    az = repmat(az, 1,N_dim);
end
if all(size(el)==[1 1])
    el = repmat(el, 1,N_dim);
end
if all(size(r)==[1 1])
    r = repmat(r, 1,N_dim);
end

% setup for either kx, ky or kz
kdim_s     = 2*pi/dim_delta;
kdim       = linspace(0, kdim_s/2, N_dim/2+1);
kdim       = [fliplr(-kdim(2 end)) kdim];
kdim_shift = getClosestVal(kdim,0)-1;

% kr matrix for expansion
krb = r'*(2 *pi *f /c);

tVariables = round(toc(tVariables)*100)/100;
disp(tVariables);
pause(1);

%% inverse Spherical Harmonic Transform (Sound Field Expansion)
disp('iSHT');
tiSHT = tic;

Y_N = sphHarm_N(N, el, az);

P = zeros(N_dim, length(f));

% progress
fprintf('Please␣wait ␣%3d%%', 0);
Step = floor(length(f)/100);
% loop for every frequency bin
for ctr_w = 1 length(f);

    Pnm_tmp = Pnm( ,ctr_w);
    krb_tmp = krb( ,ctr_w);

    [krb_tmp n_tmp] = meshgrid(krb_tmp, 0 N);

    hnb_tmp =  besselh_sph(n_tmp,2,krb_tmp);
    hnb_tmp = hnb_tmp(nvector+1, );

    P( ,ctr_w) = squeeze(sum(repmat(Pnm_tmp, [1,N_dim]) *hnb_tmp *Y_N, 1));

    if ~mod(ctr_w, Step) || ctr_w==length(f) % Update only whole percents
        fprintf(sprintf('%s%%3d%%%%', repmat('\b', 1, 4)),
            round(ctr_w/length(f)*100));
    end
end

clear krb hnb_tmp Pnm_tmp krb_tmp n_tmp nvector_tmp Pnm Y_N;

tiSHT = round(toc(tiSHT)*100)/100;
disp(tiSHT);
pause(2);
%% Spatial Fourier Transormation along dimension
disp(['Spatial␣Fourier␣Transformation␣->' dim_set_k{dim_orig}])
tFFT = tic;

PTilde = zeros(size(P));

if strcmp(win,'cheb')
    w = chebwin(N_dim,80);
elseif strcmp(win,'rect')
    w = rectwin(N_dim);
end

% loop for every frequency
```

```
for ctr_w = 1 length(f)
    PTilde( ,ctr_w) = fft_s(P( ,ctr_w) *w);
end

clear P;

PTilde = circshift(PTilde,[kdim_shift,0]);

tFFT = round(toc(tFFT)*100)/100;
(disp(tFFT));
pause(2)
%% Visualization
disp('Visualization');
tViusalization = tic;

% kx-f-Plot
hfig = plot_kspectrum(PTilde,
    f,
    kdim,
    1000,
    [0 24000],
    max(abs(cos(el))),
    [4,4],
    [ 36,N],
    {'$k_z$ / rad/m','f / Hz'},
    ['$|\tilde{P}(' coord_str{1} ',' coord_str{2}
        ',' coord_str{3}  ')|$ in dB, aus \verb#' dataset '#'],
    false);

% Ispbars
% hfig2 = plot_Isobaren_k(PTilde,
%     f,
%     kdim,
%     'kdim',
%     [100,20000,0,180],
%     [4,4],
%     [ 36,N],
%     {'$f$ / Hz', '$\theta$'},
%     {'title'},
%     false);

tVisualization = round(toc(tViusalization)*100)/100;
disp(tVisualization);
pause(2);
%% Save and Setoff
tTotal = round(toc(tTotal)*100)/100;
disp(tTotal);

PTildeData = struct;
PTildeData FFTData          = FFTData dataset;
PTildeData FFTData_path     = FFTData path;
PTildeData PnmData          = fname;
PTildeData PnmData_path     = [pname filesep fname];
PTildeData kdim             = dim_set_k{dim_orig};
PTildeData dim_spread       = dim_spread;
PTildeData dim_delta        = dim_delta;
PTildeData x                = x;
PTildeData y                = y;
PTildeData z                = z;
PTildeData flim             = [calc_f_min calc_f_max];
PTildeData win              = win;
PTildeData CPUTime          = {'Variables','iSHT','FFT','Visual','Total';
    tVariables,tiSHT,tFFT,tVisualization,tTotal};
PTildeData PTilde           = PTilde;

save([datestr(now, 'yymmddHH') '_PTilde_' dataset '_'
    coord_str{dim_orig} ' mat'], '-struct', 'PTildeData', '-v7 3');

end_str = datestr(now, 'yy-mm-dd HH MM SS');
disp(['End time is ' end_str]);
```

Listing A.3: main_kspectrum.m

A.2. tools

```
%%%
```

```
end
```

Listing A.4: readSPKStruct.m

```
%%%
```

Listing A.5: angleLookUpTable.m

```
%%%
```

```
        yn_neg = yn(2 end, );
        if strcmp(conv, 'Gumerow')
            yn_neg = conj(yn_neg);
        elseif strcmp(conv, 'Williams')
            yn_neg = diag((-1) ^m(2 end)) * conj(yn_neg);
        end

        yn_neg = flipud(yn_neg);

        yn = [yn_neg; yn];
    end
    Y_N = [Y_N; yn];
end

end
```

Listing A.6: sphHarm_N.m

```
%%%
```

```
    end

    % grid
    az1 = 0 resol(1) 360;
    el1 = 0 resol(2) 180;

    az1(end) = [];
    el1(end) = [];

    [az el] = meshgrid(az1, el1);

    az = reshape(az, [], 1);
    el = reshape(el, [], 1);

    grid = [az, el]/180*pi;

    % weights
    N = length(az1)/2 - 1;
    w_tmp = makeEquiWeights(N);
    w = repmat(w_tmp,length(az1),1);
    w = reshape(w ', [], 1);

    grid( ,3) = w;

    grid(grid( ,1)>max(1)/180*pi, ) = [];
    grid(grid( ,2)>max(2)/180*pi, ) = [];
    az1(az1>max(1)) = [];
    el1(el1>max(2)) = [];


    % outputs
    varargout{1} = grid;
    varargout{2} = az1;
    varargout{3} = el1;
    varargout{4} = N;


elseif strcmp(app, 'plot')

    az = 0 resol(1) max(1);
    el = 0 resol(2) max(2);

    [az el] = meshgrid(az, el);

    varargout{1} = az;
    varargout{2} = el;

elseif strcmp(app, 'plain')

    az1 = 0 resol(1) 360-resol(1);
    el1 = 0 resol(2) 180;

    [az el] = meshgrid(az1, el1);

    az = reshape(az, [], 1);
    el = reshape(el, [], 1);

    grid = [az, el]/180*pi;

    grid(grid( ,1)>max(1)/180*pi, ) = [];
    grid(grid( ,2)>max(2)/180*pi, ) = [];
    az1(az1>max(1)) = [];
    el1(el1>max(2)) = [];

    % outputs
    varargout{1} = grid;
    varargout{2} = az1;
    varargout{3} = el1;

else
    error('app must be either ''SHT'', ''plot'' or ''plain'' ')
end
end
```

Listing A.7: makeEquiGrid.m

```
%%%
```

```
end
```

Listing A.10: besselh_sph.m

```
%%%
```

```
% ind       - index for closest value
% val       - source(ind); closest value to target
%%%
```

```
%  1    0   0
%  2    1  -1
%  3    1   0
%  4    1   1
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [n,m] = ind2nm(ind)

if any(ind<0) || any(round(ind)~=ind)
    error('ind must be a non-negative integer')
end

n = ceil(sqrt(ind)-1);

m = ind + n - (n+1)^2;
end
```

Listing A.14: ind2nm.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (c) 2014 JAKOB BERGNER - Audiocommunication Group, TU Berlin
% bergner lj@gmail com
% developed due to master's thesis
% licensed under Creative Commons by-nc-sa license
%
% in2nm m converts the order and degree of spherical harmonic coefficients
% to an index  The dependency is
% ind   n   m
%  1    0   0
%  2    1  -1
%  3    1   0
%  4    1   1
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ind = nm2ind(n, m)

if any(n<0) || any(round(n)~=n)
    error('n must be a non-negative integer')
end

if any(abs(m)>n) || any(round(m)~=m)
    error('m must be a integer in the range -n <= m <= n')
end

ind = (n+1)^2 - n + m;
end
```

Listing A.15: nm2ind.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (c) 2014 JAKOB BERGNER - Audiocommunication Group, TU Berlin
% bergner lj@gmail com
% developed due to master's thesis
% licensed under Creative Commons by-nc-sa license
%
% function [varargout] = cart2goertz(varargin)
%
% cart2goertz m transforms a set of coordinates (x,y,z) to the equivalent
% coordinates in "Goertz coordinates"  Inputs may be a cell array
% {[x],[y],[z]} or a vector [x,y,z]  Output may be a cell array or vector
% depending on the number of output arguments  The relationships are
% x_goertz = x
% y_goertz = z
% z_goertz = -y
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [varargout] = cart2goertz(varargin)

if nargin==1
    if iscell(varargin{1})
        switch nargout
            case 1
                z_tmp = -varargin{1}{2};
                varargout{1}{2} = varargin{1}{3};
                varargout{1}{3} = z_tmp;
```

```
                varargout{1}{1} = varargin{1}{1};
            case 3
                z_tmp = -varargin{1}{2};
                varargout{2} = varargin{1}{3};
                varargout{3} = z_tmp;
                varargout{1} = varargin{1}{1};
        end
    else
        switch nargout
            case 1
                z_tmp = -varargin{1}(2);
                varargout{1}(2) = varargin{1}(3);
                varargout{1}(3) = z_tmp;
                varargout{1}(1) = varargin{1}(1);
            case 3
                z_tmp = -varargin{1}(2);
                varargout{2} = varargin{1}(3);
                varargout{3} = z_tmp;
                varargout{1} = varargin{1}(1);
        end
    end
elseif nargin == 3
    switch nargout
        case 1
            z_tmp = -varargin{2};
            varargout{1}(2) = varargin{3};
            varargout{1}(3) = z_tmp;
            varargout{1}(1) = varargin{1};
        case 3
            z_tmp = -varargin{2};
            varargout{2} = varargin{3};
            varargout{3} = z_tmp;
            varargout{1} = varargin{1};
    end
end
```

Listing A.16: cart2goertz.m

```
%%%
```

```
                varargout{1}(3) = varargin{1}(2);
                varargout{1}(2) = y_tmp;
                varargout{1}(1) = varargin{1}(1);
            case 3
                y_tmp = -varargin{1}(3);
                varargout{3} = varargin{1}(2);
                varargout{2} = y_tmp;
                varargout{1} = varargin{1}(1);
        end
    end
elseif nargin == 3
    switch nargout
        case 1
            y_tmp = -varargin{3};
            varargout{1}(3) = varargin{2};
            varargout{1}(2) = y_tmp;
            varargout{1}(1) = varargin{1};
        case 3
            y_tmp = -varargin{3};
            varargout{3} = varargin{2};
            varargout{2} = y_tmp;
            varargout{1} = varargin{1};
    end
end
end
```

Listing A.17: goertz2cart.m

```
%%%
```

```
function y = fft_t(x)
    y = fft(x);
end
```

Listing A.20: fft_t.m

```
function y = ifft_t(x)
    y = ifft(x);
end
```

Listing A.21: ifft_t.m

```
function y = fft_s(x)
    y = ifft(x)*length(x);
end
```

Listing A.22: fft_s.m

```
function y = ifft_s(x)
    y = fft(x)/length(x);
end
```

Listing A.23: ifft_s.m

A.3. plot routines

```
%%%
```

```
if ~isempty(lz)

    figure(hfig(1));

    mu = -5 5;
    mu(mu==0) = [];

    args = mu'*(c /(lz*f_hor));
    theta = acos(args);
    theta = (real(theta)+pi)/pi*180;
    hold on;
    grid on;
    plot(f_hor, theta, 'r', 'LineWidth', 2);
    f_text = 10^3 *[16 11 7 4 3 2 2 4 3 7 11 16];
    for mu_ctr = 1 length(mu)
        f_bin = getClosestVal(f_hor, f_text(mu_ctr));
        text(f_text(mu_ctr), theta(mu_ctr,f_bin),
            ['$$\mu␣=␣$$' num2str(mu(mu_ctr))],
            'HorizontalAlignment', 'center',
            'FontSize', 12,
            'Color', 'k');
    end
    hold off;
end
%% Vertical Isobars

% angle dependencies
el_deg = 0 resol(2) 180;
el_rad = el_deg/180*pi;
az = ones(size(el_deg))*+3/2*pi;

% get XFFT data
[fft_ver, az_ver, el_ver, f_ver] = getXFFT(dataset, az, el_rad, f);

bin_1k = getClosestVal(f_ver,1000);
bin_theta90 = getClosestVal(el_rad,pi/2);

% plot isobars
hfig(2) = plot_Isobars_angle(fft_ver /abs(fft_ver(bin_theta90, bin_1k)),
    unique(az),
    el_deg,
    f_ver,
    ['$|P(\mathbf{x},␣\omega)|$␣in␣dB␣f\"ur␣$\varphi␣=270^\circ$␣aus␣LineTest\_c\_2']);

% optional sinc notches
if ~isempty(lz)

    figure(hfig(2));

    mu = -5 5;
    mu(mu==0) = [];

    args = mu'*(-343 /( 36 *f_ver));
    theta_sinc = acos(args);
    theta_sinc = real(theta_sinc)/pi*180;
    hold on;
    grid on;
    plot(f_ver, theta_sinc, 'r', 'LineWidth', 1);
    f_text = 10^3 *[12 7 4 2 2 1 1 2 2 4 7 12];
    for mu_ctr = 1 length(mu)
        f_bin = getClosestVal(f_ver, f_text(mu_ctr));
        text(f_text(mu_ctr), theta_sinc(mu_ctr,f_bin),
            ['$$\mu␣=␣$$' num2str(-mu(mu_ctr))],
            'HorizontalAlignment', 'center',
            'FontSize', 12,
            'Color', 'k');
    end
end
```

Listing A.24: Visualize_Isobars.m

```
%%%
```

```
% licensed under Creative Commons by-nc-sa license
%
% function [FFT_out, az_out, el_out, f_out] = getXFFT(source, az_in, el_in, f_in)
%
% getXFFT m returns the fourier spectra of the spatial points determined by
% the vectors az_in and el_in which have to be the same size  These angles
% are transposed to the "Goertz coordinates" if necesseray  If designated
% angles or frequencies are not part of the dataset the closest available
% data points are chosen  The corresponding angles and frequencies are
% returned in az_out, el_out and f_out
%%%
```

```
% function plot_Isobars_angle(fft_data, az, el, f, tit)
%
% plot_Isobars_angle m plots two-dimensional isobars  To be used with
% Visualize_Isobars m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function hfig = plot_Isobars_angle(fft_data, az, el, f, tit)

MaxBar=12; MinBar=-60; StepdB=3;

if nargin < 5
    tit = 'Plot';
end

n_az = length(az);
n_el = length(el);

if n_az==1
    % plot_data
    fft_data = 20*log10(abs(fft_data( ,f>0)));
    v = [min(fft_data( )) MinBar StepdB MaxBar];

    % figure
    hfig = figure('name', 'Vertical Isobars');
    contourf(f(f>0),el,fft_data,v, 'LineColor', 'none');
    axis ij;
    grid on;

    % limits
    axis([100 20000 min(el) max(el)]);

    % labels
    my_YTicks = (0 12)*30;
    my_YTicks = my_YTicks(my_YTicks>=min(el)&my_YTicks<=max(el));
    my_XTicks = [1 2 3 4 5 10 20];
    my_XTickLabels = num2cell(my_XTicks);
    my_XTickLabels =  cellfun(@num2str, my_XTickLabels, 'UniformOutput', false);

    xlabel('f / kHz', 'interpreter', 'tex', 'fontname', 'times new roman', 'fontsize', 14);
    ylabel('\theta / deg', 'interpreter', 'tex', 'fontname', 'times new roman', 'fontsize', 14);
    title(tit, 'interpreter', 'latex', 'fontsize', 14);

    % colorbar
    caxis([MinBar, MaxBar]);
    cbr = colorbar;
    cm = jet(abs((MaxBar-MinBar)/StepdB));
    colormap(cm);
    set(cbr, 'YTick', MinBar 6 MaxBar);
    set(gca, 'XScale','log',
        'YTick', my_YTicks,
        'XTick', my_XTicks *1e3,
        'XTickLabel', my_XTickLabels,
        'layer', 'top');

elseif n_el==1
    % plot_data
    fft_data = 20*log10(abs(fft_data( ,f>0)));
    v = [min(fft_data( )) MinBar StepdB MaxBar];

    % figure
    hfig = figure('name', 'Horizontal Isobars');
    contourf(f(f>0),az,fft_data,v, 'LineColor', 'none');
    axis ij;
    grid on;

    % limits
    axis([100 20000 min(az) max(az)]);

    % labels
    my_YTicks = (0 12)*30;
    my_YTicks = my_YTicks(my_YTicks>=min(az)&my_YTicks<=max(az));
    my_XTicks = [1 2 3 4 5 10 20];
    my_XTickLabels = num2cell(my_XTicks);
    my_XTickLabels =  cellfun(@num2str, my_XTickLabels, 'UniformOutput', false);

    xlabel('f / kHz', 'interpreter', 'tex', 'fontname', 'times new roman', 'fontsize', 14);
```

```
    ylabel('\phi / deg', 'interpreter', 'tex', 'fontname', 'times new roman', 'fontsize', 14);
    title(tit, 'interpreter', 'latex', 'fontsize', 14);

    % clorbar
    caxis([MinBar, MaxBar]);
    cbr = colorbar;
    cm = jet(abs((MaxBar-MinBar)/StepdB));
    colormap(cm);
    set(cbr, 'YTick', MinBar 6 MaxBar);
    set(gca, 'XScale','log',
        'YTick', my_YTicks,
        'XTick', my_XTicks *1e3,
        'XTickLabel', my_XTickLabels,
        'layer', 'top');
else
    error('At least one input (az or el) must be scalar ')
end
%set(htit, 'Interpreter', 'latex');
end
```

Listing A.26: plot_Isobars_angle.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (c) 2014 JAKOB BERGNER - Audiocommunication Group, TU Berlin
% bergner lj@gmail com
% developed due to master's thesis
% licensed under Creative Commons by-nc-sa license
%
% Visualize_Balloons m plots balloon plots for several frequencies defined
% in cell 'Setup'  Number of frequnecies shold be a multiple of 2 for
% symmetry purposes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; clc;
set(0, 'defaultTextInterpreter', 'latex');

%% Setup
dataset = 'HF-BAL';
% f_plot = [125, 250, 500, 750, 1000, 2000, 8000, 16000];
% f_plot = [1000 1250 2500 5000 8000 10000 16000 20000];
f_plot  = [125, 750, 1000, 2000, 4000, 6000, 8000, 16000];

log     = true;    % logarithmic
MinBar  = -60;
MaxBar  = 0;
%% Variables
FFT_Data = matfile(['FFTData_' dataset ' mat'], 'writable', true);

fft_data = FFT_Data XFFT;
grid_full = FFT_Data grid_full;
angle_LUT = FFT_Data angle_LUT;

f = linspace(0, FFT_Data fs/2, FFT_Data fft_size);
f_bin = getClosestVal(f, f_plot);

fft_data = fft_data(angle_LUT,f_bin);

az = unique(grid_full( ,1));
el = unique(grid_full( ,2));

az_length = length(az);
el_length = length(el);

az(end+1) = 2*pi;
[az, el] = meshgrid(az,el);

max_fft_data = max(abs(fft_data( )));
min_fft_data = min(abs(fft_data( )));

%% Visualization
fig = figure;
set(fig, 'Units', 'normalized', 'Position', [0, 0,  4,  7]);

for n_plot = 1 length(f_plot)

    plot_data = reshape(fft_data( ,n_plot), el_length, az_length);
```

```
    plot_data = [plot_data, plot_data( ,1)];
    plot_data = abs(plot_data);

    if log
        max_p_data = 20*log10(max_fft_data);
        min_p_data = 20*log10(min_fft_data);
        plot_data = 20*log10(plot_data);
        plot_data = plot_data - min_p_data;
    end

    [x,y,z] = sph2cart(az, pi/2-el, plot_data);

    if strcmp(FFT_Data sph_orientation, 'Goertz')
        [x,y,z] = goertz2cart(x,y,z);
    end

    lim(n_plot, ) = [min(x( )) max(x( )) min(y( )) max(y( )) min(z( )) max(z( ))];

    h(n_plot) = subplot(2, length(f_plot)/2, n_plot);

    hold on;
    % balloon
    p(n_plot) = surf(x,y,z);
    % axes
    line(1 2*[min(x( )) max(x( ))], [0 0], [0,0], 'Color', 'r', 'LineWidth', 1 5);
    line([0 0], 1 2*[min(y( )) max(y( ))], [0,0], 'Color', 'g', 'LineWidth', 1 5);
    line([0 0], [0,0], 1 2*[min(z( )) max(z( ))], 'Color', 'b', 'LineWidth', 1 5);
    hold off;

    title(['$$f=' num2str(f_plot(n_plot)) 'Hz$$'], 'interpreter', 'latex', 'fontsize', 12);
    view(-50,20);
    axis equal
    grid off;
    axis off;

    if log
        clim = [max( MinBar, min_p_data) max(MaxBar, max_p_data)];
        set(p(n_plot), 'FaceColor', 'texturemap', 'CData', plot_data + min_p_data, 'FaceAlpha', 1, 'EdgeColor', 'none');
        set(gca, 'CLim', clim);
    else
        set(p(n_plot), 'FaceColor', 'texturemap', 'CData', plot_data, 'FaceAlpha', 1, 'EdgeColor', 'none');
        set(gca, 'CLim', [min_p_data max_p_data]);
    end
end
set(0, 'defaultTextInterpreter', 'none');
axis(h, [min(lim( ,1)), max(lim( ,2)), min(lim( ,3)), max(lim( ,4)), min(lim( ,5)), max(lim( ,6))]);
cbar = colorbar('horiz');
set(cbar, 'Position', [ 4  1  2  02])
set(cbar, 'XTick', MinBar 6 MaxBar);
cmap = jet(round(abs(clim(2)-clim(1))/3));
colormap(cmap);
```

Listing A.27: Visualize_Balloons.m

```
%%%
```

```
%% Setup

dataset          = 'HF-BAL';
mode             = 1;
norm             = 'max';
undersampling    = [1,8];
flim             = [100 20000];
nlim             = 4;
%% Variables
[fname, pname] = uigetfile(['*_Pnm_*' dataset '* mat']); % PTildeData

PnmData = matfile([pname,fname]); % PnmData
FFTData = matfile(PnmData FFTData_path); % corresponding FFTData


f = linspace(0, FFTData fs/2, FFTData fft_size);
Pnm = PnmData Pnm;
%% Visualization
hfig = plot_Pnm(Pnm,
    f,
    mode,
    norm,
    flim,
    nlim,
    4,
    {'f / Hz', 'n'},
    ['$|\breve{P}_n^m(\omega)|$ in dB aus LineTest\_c\_1' ],
    true);
```

Listing A.28: Visualize_Pnm.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% (c) 2014 JAKOB BERGNER - Audiocommunication Group, TU Berlin
% bergner lj@gmail com
% developed due to master's thesis
% licensed under Creative Commons by-nc-sa license
%
% plot_Pnm m plots the coefficients Pnm  To be used with Viualize_Pnm m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function hfig = plot_Pnm(Pnm,f,mode,norm,flim,nlim, undersampling,
    axlabels,tit, save)

MaxBar = 12; MinBar = -60; StepdB = 3;

% variables
N        = ind2nm(size(Pnm,1));
f_plot   = f(f>=flim(1) & f<=flim(2));
f_plot   = f_plot(1 undersampling end);
fbins    = getClosestVal(f,f_plot);
indlim   = min(nm2ind(nlim,nlim),nm2ind(N,N));
fTicks   = flog(125,24000,1);
fTicks   = fTicks(fTicks>=flim(1) & fTicks<=flim(2));

% data mining
if mode == 1
    ind = 1 indlim;
    plot_data = Pnm(1 indlim,fbins);

    NTickLabel = num2cell(0 indlim);
    NTickLabel =  cellfun(@num2str, NTickLabel, 'UniformOutput', false);
    %NTickLabel(mod(0 indlim,2)==1) = {''};
    % NTickLabel{1} = '';
    NTicks = nm2ind(0 indlim,0);
elseif mode == 2
    for n = 0 min(N,nlim)
        plot_data(n+1, ) = sum(abs(Pnm(nm2ind(n,-n) nm2ind(n,n),fbins)),1);
    end

    ind = 1 n+1;

    NTicks = ind;
    NTickLabel = num2cell(ind-1);
    NTickLabel =  cellfun(@num2str, NTickLabel, 'UniformOutput', false);
    NTickLabel(mod(0 indlim,2)==1) = {''};
```

```
end

% normalisation
if isnumeric(norm)
    norm_fac = max(abs(plot_data( ,getClosestVal(f_plot,norm))));
elseif strcmp(norm,'max')
    norm_fac = max(abs(plot_data( )));
elseif isempty(norm)
    norm_fac = 1;
end
plot_data = plot_data /norm_fac;
plot_data = 20*log10(abs(plot_data));

% figure
hfig = figure;
hold on;
h = bar3(f_plot, plot_data ',1);
shading flat
axis xy
view(90,-90);
set(gca,'YScale','log')


% colorbar
for k = 1 length(h)
    zdata = get(h(k),'ZData');
    set(h(k),'CData',zdata,'FaceColor','interp');
end

cbr = colorbar;
cm = jet(abs((MaxBar-MinBar)/StepdB));

colormap(cm);
set(gca,'CLim',[MinBar MaxBar]);
set(cbr, 'YTick', MinBar 6 MaxBar);

% labels
ylabel(axlabels{1}, 'interpreter', 'tex', 'FontSize', 14);
xlabel(axlabels{2}, 'interpreter', 'tex', 'FontSize', 14);
title(tit, 'interpreter', 'latex', 'FontSize', 14)
set(gca, 'YTick', fTicks, 'XTick', NTicks, 'XTickLabel', NTickLabel);


grid on;
set(gca, 'layer', 'top');

% limits
ylim([f_plot(1) f_plot(end)])
xlim([ind(1)- 5 ind(end)+ 5]);

if save
    set(gcf,'color','w');
    export_fig([' /05_graphics/' datestr(now, 'yymmddHHMM') '_Pnm png']);
end
end
```

Listing A.29: plot_Pnm.m

```
%%%
```

```
% lz            - length of virtual line source for aliasing simulation
%%%
```

```
if(size(f,1)>size(f,2))
    f = f ';
end

% axes values and grids
fbins_plot = getClosestVal(f,lim(1 2));
fbins_plot = fbins_plot(1) undersampling(2) fbins_plot(2);
f_plot = f(fbins_plot);

kdimbins_plot = 1 undersampling(1) length(kdim);
kdim_plot = kdim(kdimbins_plot);

% assign plot_data
plot_data = PTilde(kdimbins_plot,fbins_plot);

if isnumeric(norm)
    norm_fac = abs(plot_data(getClosestVal(kdim_plot,0), getClosestVal(f_plot,norm)));
elseif strcmp(norm,'kdim')
    norm_fac = abs(plot_data(getClosestVal(kdim_plot,0), ));
elseif strcmp(norm,'max')
    norm_fac = max(abs(plot_data(getClosestVal(kdim_plot,0), )));
end

plot_data = bsxfun(@rdivide, plot_data, norm_fac);

% figure
hfig = figure;
hold on;
imagesc(kdim_plot,f_plot/1000,20*log10(abs(plot_data ')));
axis xy
shading flat;

% kdim = Omega/c
plot(kdim_plot, abs(kdim_plot) *c /(2*pi*1000), 'k', 'LineWidth', 2);
if ~isempty(max_angular)
    plot(kdim_plot, abs(kdim_plot) *c /(2*pi*max_angular*1000), 'k--', 'LineWidth', 2);
end

% alias
if ~isempty(alias)
    alias_x = 2*pi/alias(1) 2*pi/alias(1) max(kdim_plot);
    %alias_x = alias_x -3;   % opt
    alias_x = [-fliplr(alias_x) alias_x];
    alias_y = f_plot/1000;
    alias_mesh = meshgrid(alias_x, alias_y);
    line(alias_mesh, alias_y, 'Color','r', 'LineWidth',1);
    if length(alias)==2
        alias_f = alias(2)*c/(pi*alias(1)*1000);
        plot([kdim(1),kdim(end)], [alias_f,alias_f], 'w- ', 'Linewidth', 2)
    end
end

% colorbar
cbr = colorbar;
MaxBar=12; MinBar=-60; StepdB=3;
cm = jet(abs((MaxBar-MinBar)/StepdB));
%cm(length(MinBar StepdB 0), ) = [1 1 1]; %Max_dB in weiss um Clipping besser zu sehen
colormap(cm);
set(gca,'CLim',[MinBar MaxBar]);
set(cbr, 'YTick', MinBar 6 MaxBar);

% limits
ylim(lim(1 2)/1000);
xlim([-2*pi/c*f_plot(end) 2*pi/c*f_plot(end)]);

% labels
xlabel(axlabels{1}, 'interpreter', 'tex', 'fontname', 'times new roman', 'fontsize', 14);
ylabel(axlabels{2}, 'interpreter', 'tex', 'fontname', 'times new roman', 'fontsize', 14);
title(tit, 'interpreter', 'latex', 'fontsize', 12);

if save
    set(gcf,'color','w');
    %set(gca, 'Fontsize', 12);
    export_fig([' /05_graphics/' datestr(now, 'yymmddHHMM') '_kspectrum png']);
end
end
```

Listing A.31: plot_kspectrum.m

```
%%%
```

```

% sinc-notches
plot(f_plot, theta_sinc, 'r', 'LineWidth', 1);
f_text = 10^3 *[16 11 7 4 3 2 2 4 3 7 11 16];
for mu_ctr = 1 length(mu)
    f_bin = getClosestVal(f_plot, f_text(mu_ctr));
    text(f_text(mu_ctr), theta_sinc(mu_ctr,f_bin),
        ['$$\mu␣=␣$$' num2str(-mu(mu_ctr))],
        'HorizontalAlignment', 'center',
        'FontSize', 12,
        'Color', 'k');
end

% limits
axis(lim);

% labels
my_YTicks = 0 15 180;
my_YTickLabels = num2cell(my_YTicks);
my_YTickLabels =  cellfun(@num2str, my_YTickLabels, 'UniformOutput', false);
my_XTicks = [1 2 3 4 5 10 20];
my_XTickLabels = num2cell(my_XTicks);
my_XTickLabels =  cellfun(@num2str, my_XTickLabels, 'UniformOutput', false);

xlabel(axlabels{1}, 'interpreter', 'tex', 'fontname', 'times␣new␣roman', 'fontsize', 14);
ylabel(axlabels{2}, 'interpreter', 'tex', 'fontname', 'times␣new␣roman', 'fontsize', 14);
title(tit, 'interpreter', 'latex', 'fontsize', 14)

% colorbar
cbr = colorbar;
cm = jet(abs((MaxBar-MinBar)/StepdB));
%cm(length(MinBar StepdB 0), ) = [1 1 1]; %Max_dB in weiss um Clipping besser zu sehen
colormap(cm);
set(cbr, 'YTick', MinBar 6 MaxBar);
set(gca, 'XScale', 'log',
    'YTick', my_YTicks,
    'YTickLabel', my_YTickLabels,
    'XTick', my_XTicks *1e3,
    'XTickLabel', my_XTickLabels,
    'layer', 'top');

if save
    set(gcf,'color','w');
    export_fig([' /05_graphics/' datestr(now, 'yymmddHHMM') '_IsoVer png']);
end
end
```

Listing A.32: plot_Isobars_k.m