universität wien

DIPLOMARBEIT

Titel der Diplomarbeit

Quality of Gabor Multipliers
for Operator Approximation

angestrebter akademischer Grad

Magister der Naturwissenschaften (Mag. rer. nat.)

Wien, am May 23, 2012

Verfasser:	Kirian Anselm Döpfner
Studienkennzahl:	A405
Matrikel-Nummer:	a0607964
Studienrichtung:	Mathematik
Betreuer:	Univ.-Prof. Dr. Hans Georg Feichtinger

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Wien, am May 23, 2012

Unterschrift
(Kirian Anselm Döpfner)

Danksagung

Ich möchte hier ein großes Dankeschön an meinen Diplomarbeitsbetreuer Hans Georg Feichtinger sagen. Danke für die Hilfe und und weitgefächerte Unterstützung sowie Anstöße und Vorschläge über die Zeit der Arbeit. Auch dafür, dass er sich immer Zeit nahm, wenn ich mit Fragen zu ihm kam. Ebenfalls Danke an Radu Corneliu Frunza für seine Hilfe mit ein paar MATLAB™ Routinen.

Ein großes Dankeschön geht auch an meine Familie und vor allem meine Partnerin Elisabeth dafür, dass sie mich immer wieder aufbauten wenn ich zu zweifeln anfing.

Contents

List of Figures

List of Tables

Abstract

In many applications, and thus also in signal processing, there are real-time calculations to be done. For example, if one tries to invert the changes done to a signal coming from a mobile phone that travels on a train or car. There is the Doppler effect, as well as changes due to weather, buildings, hills and so on. The changes can be well measured by measuring the changes done to *pilot tones*. But if one wants to invert the changes in real time, the exact computations (if possible) can be very costly ones. This is why we want to approximate the operators needed with operators that act on the signal at low computational cost, so we do not have a bad lag when talking on the phone. In the same way we should also get an approximation of the operator that is close enough to the original one. Here we use *Gabor multipliers* as approximating operators.These are based on transforming a signal in to a discretized time-frequency representation, then do some modification (filtering) by pointwise multiplying with a weight function and going back to a time representation.

The aim of this paper is to find criteria on the building blocks of such a Gabor multiplier, that tell us how viable the so-called *Gabor system* is for approximation. For this, we will introduce some condition numbers, especially the *compound condition number*, for a Gabor system. It should give us an idea of how good the tested Gabor system is, hopefully in a uniform way, such that the range of good condition numbers are the same independent of the sampling size.

The first chapter is an introduction to time-frequency analysis and some tools needed in that manner. It will introduce some basic operations, particularly the *short-time Fourier transform*. It also introduces our "playground", the space $\boldsymbol{S}_0(\mathbb{R}^d)$ and its corresponding *Banach Gefand triple* $\big(\boldsymbol{S}_0(\mathbb{R}^d), \boldsymbol{L}^2(\mathbb{R}^d), \boldsymbol{S}_0'(\mathbb{R}^d)\big)$.

Chapter two gives insight to *frames*, a generalization of *bases* and a concept needed a lot in Gabor analysis. With bases being a very tight concept, as one needs all coefficients of a basis expansion to reconstuct an element. Frames come into play, when one needs more flexibility.

After that we can view some achievements of Gabor analysis in chapter three. We also derive some very important tools to use for approximation of elements of a Hilbert space by orthogonal projection onto a so-called *spline-type space* $\mathbf{V}_{g,\Lambda}$. These consist of the closed linear span of an atom function g shifted on a discrete subgroup Λ. And there are some considerations concerning realization of the calculations on computers as well.

Finally Gabor multipliers are introduced as filters in the time-frequency plane in chapter four. Also a representation of them as infinite linear combinations of rank-one operators is given. Then we develop the theory needed to know how we can approximate $\mathcal{HS}$-operators by Gabor multipliers via orthogonal projection onto a subspace. We also introduce some representations of operators like the Kohn-Nirenberg symbol

or the spreading representation, which are quite similar to the (distributional) kernel of an operator.

Chapter five is dedicated to formulating quality criteria for Gabor systems to be well suited for our use. This is where the condition numbers come into play. The aim here was, to first give a theoretical background and then try to simulate and ratify this via tests. There are also tests regarding the kind of operator we approximate, or better which ones can not be well approximated with certain building blocks for the Gabor multiplier. One conclusion is the when the upper symbol of a STFT multiplier contains high frequencies, we get stronger constraints on the building blocks of the approximating Gabor multiplier.

As the last chapter is quite applied and some of the theory might not be all clear to the reader at first, I went with the saying: "A picture is worth a thousand words". I hope to not have overdone it, but I think it is always good to see whats happening rather than try to read it in a bunch of numbers.

There should be a digital version of this paper available on the webpage of the NuHAG http://www.univie.ac.at/nuhag-php/bibtex/.

Chapter 1.

Introduction to Time-Frequency Analysis

This chapter deals with some of the basics in classical Fourier analysis and leads us to time-frequency analysis. As some general results, e.g. concerning the Fourier transform, are well-known and can be found in the standard books, there will be no proofs for them and references left out. As this is the starting and introductory chapter, there will be some statements and definitions from functional analysis included, to complete the picture.

1.1. Banach and Hilbert Spaces

As a starting point, we consider a *vectorspace* over the scalar field $\mathbb{K}$, which is in our case always $\mathbb{R}$ or $\mathbb{C}$.

A vector space V together with a norm $\|.\| : V \to \mathbb{R}_{\geq 0}$ for the vector space is called *normed space* and written as $(V, \|.\|_V)$.

Definition 1.1.1. *For a subset $W \subseteq V$ of a vector space we call*

(i) W *a* subspace *of V, if for all $x, y \in W, \alpha \in \mathbb{K} \Rightarrow x + y \in W, \alpha x \in W$.*

(ii) W *a* dense subset *of V, if for any $x \in V$ there is a sequence $a_n \in W$ with $\lim_n a_n = x$.*

(iii) the linear span of W*, span(W), the smallest subspace of V which containes W. This coincides with the set of all finite linear combinations of elements in W. The* closed linear span *of W is the closure $\overline{span}(w)$.*

Definition 1.1.2. *Let a_n be a sequence in V.*

(i) a_n converges *to a in V, if $\forall \varepsilon > 0, \exists N > 0 : \|a_n - a\| < \varepsilon, \forall n > N$.*

(ii) a_n *is a* Cauchy sequence *in V, if $\forall \varepsilon > 0, \exists N > 0 : \|a_n - a_m\| < \varepsilon, \forall n, m > N$*

Definition 1.1.3 (Banach space)**.** *Let $(B, \|.\|_V)$ be a normed space. B is said to be* complete *if every Cauchy sequence in B converges in B. A* Banach space *B is a complete normed space. It is called* seperable *if it contains a countable dense subset.*

Example 1. *A few examples of Banach spaces used in this paper, $1 \leq p \leq \infty$.*

(i) $\boldsymbol{\ell}^p(I) := \{a = \{a_i\}_{i\in I} : a_i \in \mathbb{C}, \|a\|_p < \infty\}$

with $\|a\|_p := \left(\sum_{i\in I} |a_i|^p\right)^{1/p}$. *For* $p = \infty$ *we get the space of all bounded sequences* $\boldsymbol{\ell}^\infty(I)$ *with respect to the infinity norm* $\|a\|_\infty := \sup_{i\in I} |a_i|$.

(ii) $\boldsymbol{L}^p(\mathbb{R}^d) := \{f : \|f\|_p < \infty\}$

with $\|f\|_p := \left(\int_{\mathbb{R}^d} |f(x)|^p dx\right)^{1/p}$. *These are the Banach spaces of equivalence classes of functions over* $\mathbb{R}^d$, *whose* p^{th} *power is Lebesgue integrable. Here two functions are equal if they are equal almost everywhere. For* $p = \infty$ *we get the essentially bounded functions with the norm* $\|f\|_{\boldsymbol{L}^\infty} := ess\, sup_{x\in\mathbb{R}^d} |f(x)|$.

Some spaces that appear as we go on are, e.g. $\boldsymbol{C}_0(\mathbb{R}^d)$, which is the space of continuous functions that vanish at infinity ($\lim_{|x|\to\infty} f(x) = 0$). The *Schwartz class* $\boldsymbol{\mathcal{S}}(\mathbb{R}^d)$ is the space of functions over $\mathbb{R}^d$, which are infinitely differentiable and rapidly decreasing at infinity (to give an illustrative idea). Its dual $\boldsymbol{\mathcal{S}}'(\mathbb{R}^d)$ is the space of *tempered distributions*. As the Schwartz class is a Frechet space and therefore hard to deal with and its dual contains some "nasty" distributions, we later introduce another space of test functions denoted by $\boldsymbol{S}_0(\mathbb{R}^d)$ which is a Banach space (as well as its dual) and has much nicer properties.

A vector space V together with an *inner product* $\langle .,. \rangle : V \times V \to \mathbb{K}$ is called an *inner product space*. It is a normed space with the so-called *induced norm* $\|x\|_V := \sqrt{\langle x, x\rangle}$.

Definition 1.1.4 (Hilbert space)**.** *A complete inner product space is called a* Hilbert space*.*

Example 2. *Some Hilbert spaces used in this paper.*

(i) $\boldsymbol{\ell}^2(I)$ *together with the inner product* $\langle x, y\rangle := \sum_{i\in I} x_i \overline{y_i}$.

(ii) $\boldsymbol{L}^2(\mathbb{R}^d)$ *together with the inner product* $\langle f, g\rangle := \int_{\mathbb{R}^d} f(x)\overline{g(x)}dx$.

For the inner product in $\mathbb{C}^d$ we write $x \cdot y := \sum_{i=1}^d x_i \overline{y_i}$ and in the right context $x^2 := x \cdot x$.

Definition 1.1.5 (Dual space)**.** *The* dual space V' *of a normed space* V *is defined as the set of continuous linear functionals* $\phi : V \to \mathbb{K}$ *on* V. *It is a normed space with respect to the dual norm* $\|\phi\|_{V'} := \sup_{\|x\|\le 1} |\phi(x)|$.

Definition 1.1.6 (Bounded linear operators)**.** *The set of* bounded linear operators*, i.e. the set of linear operators* $T : V \to W$, *with* V, W *normed spaces and* finite *operator norm*

$$|||T||| := \sup_{\|x\|_V \le 1} \|Tx\|_W = \sup_{\|x\|_V = 1} \|Tx\|_W,$$

is denoted by $\mathfrak{L}(V, W)$.

Lemma 1.1.1. *For a linear operator* T *the following are equivalent*

(i) $T \in \mathfrak{L}(V, W)$, *i.e. is bounded on* V.

(ii) T *is continuous in every* $x \in V$.

(iii) T *is contiunous in one* $x \in V$.

As a direct consequence of this, one can write the dual space V' of a normed space V as $V' = \mathfrak{L}(V, \mathbb{K})$.

Definition 1.1.7 (Self-adjoint operators)**.** *An operator* $T \in \mathfrak{L}(\mathcal{H})$ *is called* self-adjoint *if* $T = T^*$*, where* T^* *denotes the adjoint operator of* T *if*

$$\langle Tx, y\rangle_{\mathcal{H}} = \langle x, T^*y\rangle_{\mathcal{H}}.$$

Theorem 1.1.1. *For a self-adjoint operator* $T = T^* \in \mathfrak{L}(\mathcal{H})$ *the operatornorm is given by*

$$|||T||| = \sup_{\|f\|\leq 1} |\langle Tf, f\rangle|$$

Proof. See [Wer97, V.5.7]. □

Definition 1.1.8 (Positive operator)**.** *Let* $(\mathcal{H}, \langle ., .\rangle)$ *be a Hilbert space. A self-adjoint operator* $T \in \mathfrak{L}(\mathcal{H})$ *is called a* positive operator*, if* $\langle Tf, f\rangle \geq 0$ *for all* $f \in \mathcal{H}$.

For a positive operator T, there always exists a self-adjoint operator $A : \mathcal{H} \to \mathcal{H}$ with $T = A^2$. We denote $A = T^{\frac{1}{2}}$.

Theorem 1.1.2 (Neumann series)**.** *Let* V *be a normed space and* $T \in \mathfrak{L}(V)$*. If the series* $\sum_{k=0}^{\infty} T^k$ *converges in* $\mathfrak{L}(V)$*, then* $\mathrm{Id} - T$ *is invertible and*

$$(\mathrm{Id} - T)^{-1} = \sum_{k=0}^{\infty} T^k. \tag{1.1}$$

If further V *is a Banach space and* $|||T||| < 1$*, then the series converges and we get the norm estimate* $|||(\mathrm{Id} - T)^{-1}||| \leq (1 - |||T|||)^{-1}$.

Proof. For a proof, see [Wer97, Theorem II.1.11] □

1.1.1. Singular Value Decomposition and Pseudo-inverses

As it is well know from linear algebra, not all matrices have an inverse. Therefore it is desirable to have a generalization for the important features of the inverse. Given an arbitrary $m \times n$-matrix A seen as linear mapping $A : \mathbb{C}^n \to \mathbb{C}^m$, we denote by $\mathcal{N}_A$ the nullspace or kernel of A and by $\mathcal{R}_A$ its range. By restricting A to the orthogonal complement of its nullspace, the restriction $\tilde{A}$ becomes an injective linear mapping

$$\tilde{A} : \mathcal{N}_A^{\perp} \to \mathcal{R}_A$$

with the same range, $\mathcal{R}_{\tilde{A}} = \mathcal{R}_A$. Therefore $\tilde{A}$ has an inverse

$$(\tilde{A})^{-1} : \mathcal{R}_A \to \mathcal{N}_A^{\perp}.$$

This inverse can now be extended to an operator $A^{\dagger} : \mathbb{C}^m \to \mathbb{C}^n$ by simply defining

$$A^{\dagger}(x + y) = (\tilde{A})^{-1}x \text{ if } x \in \mathcal{R}_A, y \in \mathcal{R}_A^{\perp} \tag{1.2}$$

We get $AA^\dagger x = x,\ \forall x \in \mathcal{R}_A$ and hence $A^\dagger$ will be called the *pseudo-inverse* or *Moore-Penrose inverse* of A. If A is in fact invertible, then $A^\dagger$ coincides with $(\tilde{A})^{-1}$ and $\tilde{A}$ coincides with A, therefore we get for invertible A that $A^\dagger = A^{-1}$. The pseudo-inverse can be characterized as follows [SB93, 4.8.5.1].

Proposition 1.1.1. *Let A be an $n \times m$-matrix. Then the pseudo-inverse $A^\dagger$ is an $m \times n$-matrix satisfying:*

1. *$A^\dagger A = P$ is the orthogonal projection $P : \mathbb{C}^m \to \mathcal{N}_A^\perp \subseteq \mathbb{C}^n$.*
2. *$AA^\dagger = \tilde{P}$ is the orthogonal projection $\tilde{P} : \mathbb{C}^n \to \mathcal{R}_A \subseteq \mathbb{C}^m$.*

One important property about the pseudo-inverse is its minimization property.

Theorem 1.1.3. *Let A be an $n \times m$-matrix. Given $y \in \mathcal{R}_A$, the minimization problem*

$$\min_x \|Ax - y\|_2$$

has a unique solution, namely $\hat{x} = A^\dagger y$.

Proof. As $AA^\dagger x = x,\ \forall x \in \mathcal{R}_A$, we know that $\hat{x} := A^\dagger y$ is a solution to the equation $Ax = y$. All solutions have the form $x = A^\dagger y + z$, where $z \in \mathcal{N}_A$. Since $A^\dagger y \in \mathcal{N}_A^\perp$, the 2-norm of the general solution is given by

$$\|x\|_2^2 = \|A^\dagger y + z\|_2^2 = \|A^\dagger y\|_2^2 + \|z\|_2^2$$

which is minimal when z=0. □

A good way, for computational reasons, to find the pseudo-inverse is via the *singular value decomposition* of A.

Theorem 1.1.4 (Singular Value Decomposition)**.** *For every $n \times m$-matrix A with rank $r \geq 1$ there is a decomposition, called the* singular value decomposition *(SVD), i.e.*

$$A = U \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} V^* \tag{1.3}$$

where U is a unitary $n \times n$-matrix, V is a unitary $m \times m$-matrix, and $\begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix}$ is an $n \times m$ block matrix in which D is an $r \times r$ diagonal matrix with positive entries $\sigma_1, \ldots, \sigma_r$ in the diagonal.

The numbers $\sigma_1, \ldots, \sigma_r$ are called the singular values of A and are the square roots of the positive eigenvalues of A^*A.

Corollary 1.1.1. *With the notation above, the pseudo-inverse of A is given by*

$$A^\dagger = V \begin{pmatrix} D^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^*, \tag{1.4}$$

where D^{-1} is the $r \times r$ diagonal matrix with the entries $1/\sigma_1, \ldots, 1/\sigma_r$ in the diagonal.

For a proof of the last theorem and corollary look at [Chr03, Chapter 1].

1.2. The Fourier Transform

A function f on $\mathbb{R}^d$ is, mathematically speaking, a rule that associates a complex value to each point $x \in \mathbb{R}^d$. For applications one can view a function on $\mathbb{R}$ as a function of "time" which assigns an amplitude $f(t)$ to each point in time t. In this case we call the function a *signal*. In dimension $d = 2$, we call $f(x_1, x_2)$, viewed as an assignment of a color to each point in the plane, an *image*.

The *Fourier transform* as defined below can be looked at as a transform, that goes from the "time domain" to the "frequency domain". This means, that for a function $f(t)$ (t in seconds), the Fourier transform of f is a function $\hat{f}(\omega)$ that assigns an amplitude to each frequency ω (in hertz). So the Fourier transform decomposes a function into its frequencies. This is a basic tool for signal and image processing.

Definition 1.2.1 (The Fourier transform)**.** *For $f \in \boldsymbol{L}^1(\mathbb{R}^d)$ and $\omega \in \mathbb{R}^d$, we choose the following normalization for the* Fourier transform *of f.*

$$(\mathcal{F}f)(\omega) = \hat{f}(\omega) = \int_{\mathbb{R}^d} f(x) e^{-2\pi i x \cdot \omega} dx. \tag{1.5}$$

Classically, the Fourier transform is first defined on $\boldsymbol{L}^1$ and later extended to $\boldsymbol{L}^2$ (or $\boldsymbol{L}^p$) or even bigger classes of functions (e.g. the *Schwartz class* and its dual), via density arguments or duality properties. Of course, then one might have to omit the pointwise aspect of (1.5) (cf. [Grö01, Appendix A.1]).

Lemma 1.2.1 (Riemann-Lebesgue)**.** *For $f \in \boldsymbol{L}^1(\mathbb{R}^d)$, its Fourier transform $\hat{f}$ is uniformly continuous and $\lim_{|\omega| \to \infty} |\hat{f}(\omega)| = 0$.*

As direct consequence we get the following property of the Fourier transform

$$\mathcal{F} : \boldsymbol{L}^1(\mathbb{R}^d) \to \boldsymbol{C}_0(\mathbb{R}^d)$$

The extension of $\mathcal{F}$ to a unitary operator $\mathcal{F} : \boldsymbol{L}^2(\mathbb{R}^d) \to \boldsymbol{L}^2(\mathbb{R}^d)$ is justified via Plancherel's theorem.

Theorem 1.2.1 (Plancherel)**.** *If $f \in \boldsymbol{L}^1 \cap \boldsymbol{L}^2(\mathbb{R}^d)$, then*

$$\|f\|_{\boldsymbol{L}^2} = \|\hat{f}\|_{\boldsymbol{L}^2} \tag{1.6}$$

As a consequence $\mathcal{F}$ extends in a unique way to a unitary operator on $\boldsymbol{L}^2(\mathbb{R}^d)$ and satisfies Parseval's formula

$$\langle f, g \rangle_{\boldsymbol{L}^2} = \langle \hat{f}, \hat{g} \rangle_{\boldsymbol{L}^2} \tag{1.7}$$

The Fourier transform is therefore an energy preserving operation on signals, in this interpretation for signal analysis.

1.3. Fundamental Operations

This will introduce some fundamental operations and their operator symbols, which will appear more often in this text.

Translation and Modulation

Definition 1.3.1. *The* translation *with the operator symbol* T_x *for* $x \in \mathbb{R}^d$ *is defined as*

$$(\mathrm{T}_x f)(t) := f(t-x) \tag{1.8}$$

and the modulation *with operator symbol* M_ω *for* $\omega \in \mathbb{R}^d$ *as*

$$(\mathrm{M}_\omega f)(t) := e^{2\pi i \omega \cdot t} f(t). \tag{1.9}$$

These are the two main operators used in time-frequency analysis, where T_x is called a *time shift* and M_ω a *frequency shift*. They have the inverse operations $\mathrm{T}_x^{-1} = \mathrm{T}_{-x}$ for the translation and $\mathrm{M}_\omega^{-1} = \mathrm{M}_{-\omega}$ for the modulation. The compositions of the two $\mathrm{M}_\omega \mathrm{T}_x$ are called *time-frequency shifts* (TF-shifts), and denoted by $\pi(\lambda) = \mathrm{M}_\omega \mathrm{T}_x$ for $\lambda = (x, \omega) \in \mathbb{R}^d \times \widehat{\mathbb{R}}^d$. Here $\widehat{\mathbb{R}}^d$ is just a copy of $\mathbb{R}^d$ but denoted differently so that we can clearly see when we are in the frequency domain.

By an easy calculation one can see that the time shifts and frequency shifts satisfy the following *commutation relation*

$$\mathrm{T}_x \mathrm{M}_\omega = e^{-2\pi i x \cdot \omega} \mathrm{M}_\omega \mathrm{T}_x \,. \tag{1.10}$$

Note that the inverse of $\pi(\lambda)$ is $\mathrm{T}_{-x} \mathrm{M}_{-\omega}$, as $\mathrm{T}_{-x} \mathrm{M}_{-\omega} \mathrm{M}_\omega \mathrm{T}_x = \mathrm{M}_\omega \mathrm{T}_x \mathrm{T}_{-x} \mathrm{M}_{-\omega} = \mathrm{Id}$ and therefore

$$\pi(\lambda)^{-1} = e^{-2\pi i x \cdot \omega} \pi(-\lambda). \tag{1.11}$$

Thus the adjoint time-frequency shift is the inverse time-frequency shift

$$\pi^*(\lambda) = \mathrm{T}_x^* \mathrm{M}_\omega^* = \mathrm{T}_{-x} \mathrm{M}_{-\omega} = e^{-2\pi i x \cdot \omega} \pi(-\lambda) = \pi(\lambda)^{-1} \tag{1.12}$$

which is equivalent to the fact that time-frequency shifts are unitary. They also satisfy

$$\pi(\lambda)\pi(\lambda') = e^{-2\pi i x \cdot \omega'} \pi(\lambda + \lambda'). \tag{1.13}$$

The tensor product $(\pi \otimes \pi^*)(\lambda)$, as it is defined in [FK98] via its action on an operator T, is

$$(\pi \otimes \pi^*)(\lambda) T = \pi(\lambda) \circ T \circ \pi^*(\lambda)$$

and sometimes denoted by $\pi_2(\lambda)$.

It can be shown by an easy calculation, that $\pi_2(\lambda)$ corresponds to a time-frequency shift $\mathrm{M}_{(\omega,-\omega)} \mathrm{T}_{(x,x)}$ of the distributional kernel $\kappa(T)$ of the operator. See [Sch04, 1.1.3 (b)] for the steps.

There are relations between translation and modulation under the Fourier transform

$$\mathcal{F}(\mathrm{T}_x f) = \mathrm{M}_{-x} \mathcal{F} f$$

and

$$\mathcal{F}(\mathrm{M}_\omega f) = \mathrm{T}_{-\omega} \mathcal{F} f.$$

Another property of the TF-shifts is that they are isometries on $\boldsymbol{L}^p$ for $1 \leq p \leq \infty$. This means that

$$\|\pi(\lambda) f\|_p = \|f\|_p.$$

Convolution, Involution and Reflection

Definition 1.3.2. *The* convolution *of two functions* $f, g \in \boldsymbol{L}^1(\mathbb{R}^d)$ *is the function* $f * g$ *defined by*

$$(f * g)(x) := \int_{\mathbb{R}^d} f(y)g(x-y)dy. \tag{1.14}$$

The convolution satisfies

$$\|f * g\|_1 \leq \|f\|_1 \|g\|_1$$

and

$$\mathcal{F}(f * g) = \mathcal{F}f \cdot \mathcal{F}g.$$

Definition 1.3.3. *The* involution * *is defined by*

$$f^*(x) := \overline{f(-x)} \tag{1.15}$$

and the reflection $\mathcal{I}$ *by*

$$\mathcal{I}f(x) := f(-x) \tag{1.16}$$

For these we have

$$\mathcal{F}(f^*) = \overline{\mathcal{F}f} \text{ and } \mathcal{F}(\mathcal{I}f) = \mathcal{I}(\mathcal{F}f).$$

With the notation of these operators, one can write the convolution as an "inner product" with a translation and an involution, that is,

$$(f * g)(x) = \langle f, \mathrm{T}_x\, g^* \rangle_{\boldsymbol{L}^2}, \tag{1.17}$$

assuming both sides are defined.

The Inversion Formula

If one does a Fourier transform of a function f, one may want to go back to the time domain (maybe after some alternation of $\hat{f}$). So to say we need an inverse Fourier transform $\mathcal{F}^{-1}$.

Theorem 1.3.1 (Inverse Fourier transform)**.** *If* $f \in \boldsymbol{L}^1(\mathbb{R}^d)$ *and* $\hat{f} \in \boldsymbol{L}^1(\mathbb{R}^d)$*, then*

$$f(x) = \int_{\mathbb{R}^d} \hat{f}(\omega) e^{2\pi i x \cdot \omega} d\omega \text{ for all } x \in \mathbb{R}^d. \tag{1.18}$$

In other words

$$\mathcal{F}^{-1} = \mathcal{I}\mathcal{F}.$$

Similar to the Fourier transform, the inversion formula can be extended to other function spaces ($\boldsymbol{L}^p, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{S}}', ...$).
Remark: Let $f \in \boldsymbol{L}^1 \cap \boldsymbol{L}^2(\mathbb{R}^d)$ with $\hat{f} \in \boldsymbol{L}^1 \cap \boldsymbol{L}^2(\mathbb{R}^d)$, then $\mathcal{F}^{-1}\hat{f} \in \boldsymbol{C}_0(\mathbb{R}^d) \subseteq \boldsymbol{L}^2(\mathbb{R}^d)$. This means that $g = \mathcal{F}^{-1}\mathcal{F}f$ for a function $f \in \boldsymbol{L}^1 \cap \boldsymbol{L}^2(\mathbb{R}^d)$ gives back a continuous representative of the equivalence class $[f]$ in $\boldsymbol{L}^2(\mathbb{R}^d)$.

1.4. The Short-Time Fourier Transfrom

As we have seen in the previous section, the Fourier transform gives a complete representation of a signal in the frequency domain. But one sees only which amplitudes appear in the whole signal, not which ones at what time. So we can now look at a signals amplitudes eigther depending on time or depending on frequency. The problem now is, that if we want to find out the melody of a musical signal, we need to know which *frequencies* appear at what *time*. So to speak, we need a representation of both f and $\hat{f}$ at once. This should then be a function of two variables $v(x, \omega)$, so we want a representation of f in the *time-frequency plane*, also called the *phase space* $\mathbb{R}^d \times \widehat{\mathbb{R}}^d$.

As for a given signal, it is (intuitively) not possible to know the frequencies at a certain time x, if one knows the signal only at the point x. This means that we need to know the signal at least for a short interval to determine the frequencies over that time. For an interval $[x-\delta, x]$ we do the Fourier transform of $f_{[x-\delta,x]} = f \cdot g$ where g is a so-called *window function* with support in $[x-\delta, x]$, e.g. $g = \chi_{[x-\delta,x]}$. In this manner we can see the frequencies of f that appear in $[x-\delta, x]$ as we look at $\hat{f}_{[x-\delta,x]}$.

Now by shifting g along time, we get

$$\begin{aligned} v(x,\omega) :&= \mathcal{F}(f \cdot \mathrm{T}_x\, \overline{g})(\omega) && (1.19)\\ &= \int_{\mathbb{R}^d} f(t)\overline{g(t-x)}e^{-2\pi i t\cdot\omega}dt && (1.20)\\ &= \langle f, \mathrm{M}_\omega\, \mathrm{T}_x\, g\rangle && (1.21) \end{aligned}$$

and by this we can make the following definition

Definition 1.4.1 (Short-Time Fourier Transform)**.** *For a fixed window function $g \neq 0$, the* short-time Fourier transform *(STFT) of a function f with respect to g is defined by*

$$\begin{aligned} \mathcal{V}_g f(x,\omega) :&= \int_{\mathbb{R}^d} f(t)\overline{g(t-x)}e^{-2\pi i t\cdot\omega}dt && (1.22)\\ &= \int_{\mathbb{R}^d} f(t)\overline{\pi(\lambda)g}\; dt, && (1.23) \end{aligned}$$

for $x, \omega \in \mathbb{R}^d$.

With this transform, we can look at the "local frequency spectrum". But for sharp cut-off functions like $g = \chi_{[a,b]}$ we get discontinuities of f that can create some problems like unwanted noise in the frequency spectrum. We usually use smooth (well localized) cut-off windows g like the *Gaussians*

Definition 1.4.2 (Gaussian)**.** *Let*

$$\phi_a(x) = e^{-\frac{\pi x^2}{a}}, \tag{1.24}$$

denote the non-normalized Gaussian function with width $a > 0, a \in \mathbb{R}$.

Recall that $x^2 = x \cdot x = \sum_{i=1}^d |x_i|^2$ in this context for $x \in \mathbb{R}^d$. One nice property of the Gaussian ϕ_1 is that it is normalized and invariant under the Fourier transform, $\mathcal{F}\phi_1 = \phi_1$ ([Grö01, Lem. 1.5.1.]).

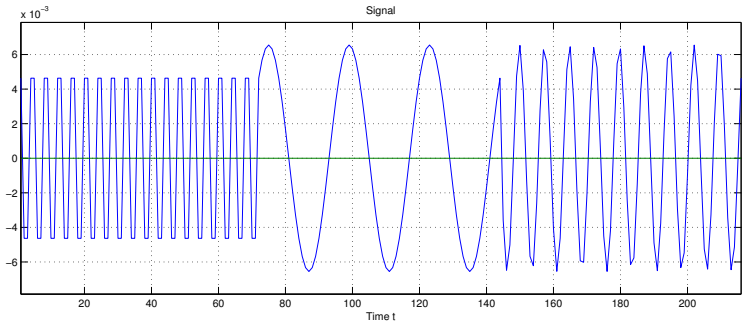


Figure 1.1.: A signal with 3 different frequencies

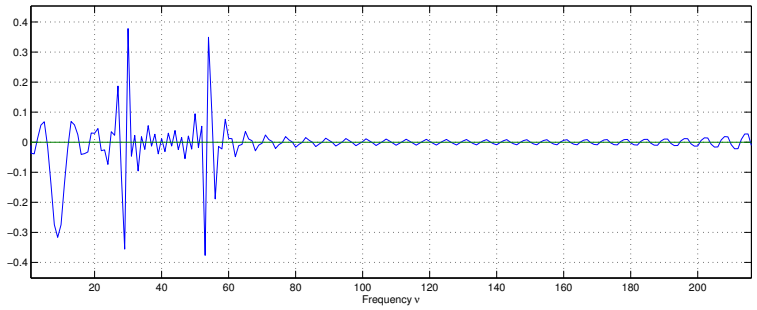


Figure 1.2.: Fourier transform of the signal

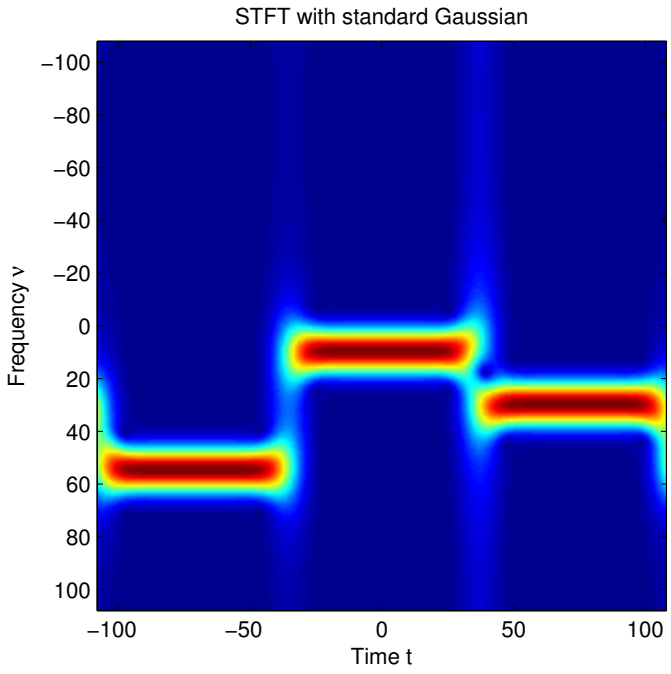


Figure 1.3.: STFT of the signal with standard Gaussian window ϕ_1

In Figure 1.1 we see a signal with 3 different frequencies which do not overlap in time. In Figure 1.2 we see its Fourier transform and in Figure 1.3 its short-time Fourier transform. Observe that, from the Fourier transform one cannot distinguish temporal properties of the signal, but with the STFT one can see the change of frequency in the signal.

The definition of the short-time Fourier transform can be extended to a greater class of functions/distributions, where the integral in (1.22) is no longer defined. This is done via the inner product representation (1.21). It follows that for test functions $g \in \boldsymbol{\mathcal{S}}$, the STFT $\mathcal{V}_g\sigma$ of σ is well defined for all $\sigma \in \boldsymbol{\mathcal{S}}'$, i.e.

$$(\mathcal{V}_g\sigma)(\lambda) = \sigma\left(\overline{\pi(\lambda)g}\right). \tag{1.25}$$

Equiped with the STFT tool, one might think we can determine the exact frequency at an exact time. But it is not all that nice. There is the fundamental observation called the *uncertainty principle*, which states that no function g can be well-concentrated in both time and frequency. This means, that the essetial support of g and $\hat{g}$ always cover some minimal area in the time-frequency plane $\mathbb{R}^d \times \widehat{\mathbb{R}}^d$. Formally (for $d = 1$) this is know as the Heisenberg-Pauli-Weyl inequality.

Theorem 1.4.1 (Uncertainty principle)**.** *If $f \in \boldsymbol{L}^2(\mathbb{R})$ and $a, b \in \mathbb{R}$ are arbitrary, then*

$$\left(\int_{-\infty}^{\infty}(x-a)^2|f(x)|^2dx\right)^{1/2}\left(\int_{-\infty}^{\infty}(\omega-b)^2|\hat{f}(\omega)|^2d\omega\right)^{1/2} \geq \frac{1}{4\pi}\|f\|_2^2. \tag{1.26}$$

Equality holds if and only if f is a multiple of $\mathrm{T}_a\,\mathrm{M}_b\,\phi_c(x) = e^{2\pi ib(x-a)}\cdot e^{-\pi(x-a)^2/c}$ for some $a, b \in \mathbb{R}$ and $c > 0$.

Proof. See [Grö01, 2.2]. □

If we want better frequency resolution, we need a wider window g, but then again, we get less time resolution. Just as taking a short window results in a blurred frequency representation. In Figure 1.4, we see the short-time Fourier transform with a wide and a short window. The windows are plotted in Figure 1.5 in comparison to the standard Gaussian. The Uncertainty principle in Theorem 1.4.1 tells us, that the STFT representation can not tell us both, exact time of this frequency change and the frequency itself.

There are some similar properties for the STFT, as for the Fourier transform. The equation in the next theorem is the analogon to Parseval's formula and as a corollary we get the inversion formula.

Theorem 1.4.2 (Orthogonality relations for the STFT)**.** *Let $f_1, f_2, g_1, g_2 \in \boldsymbol{L}^2(\mathbb{R}^d)$, then $\mathcal{V}_{g_j}f_j \in \boldsymbol{L}^2(\mathbb{R}^{2d})$ for $j = 1, 2$, and*

$$\langle \mathcal{V}_{g_1}f_1, \mathcal{V}_{g_2}f_2\rangle_{\boldsymbol{L}^2(\mathbb{R}^{2d})} = \langle f_1, f_2\rangle\overline{\langle g_1, g_2\rangle} \tag{1.27}$$

Proof. See [Grö01, Theorem 3.2.1.] □

There is also a similar result to Plancherel, for g with $\|g\|_2 = 1$.

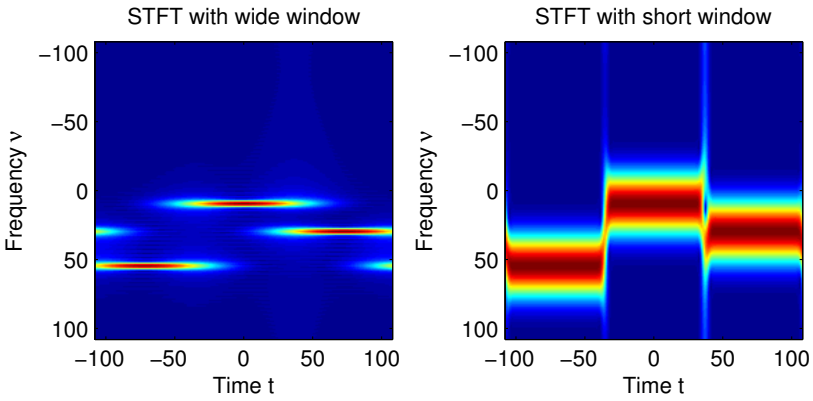


Figure 1.4.: STFT with wide window (left) and short window (right)

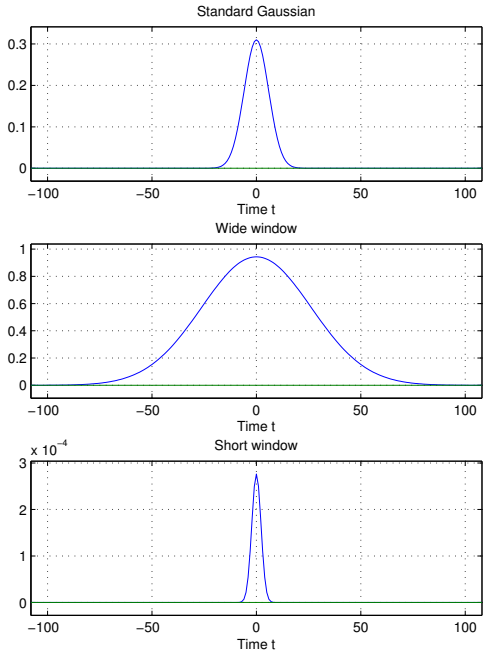


Figure 1.5.: 3 Gaussians with different width

Corollary 1.4.1. *If $f, g \in \boldsymbol{L}^2(\mathbb{R}^d)$, then*

$$\|\mathcal{V}_g f\|_2 = \|f\|_2 \|g\|_2.$$

In particular, if $\|g\|_2 = 1$ then

$$\|f\|_2 = \|\mathcal{V}_g f\|_2 \tag{1.28}$$

for all $f \in \boldsymbol{L}^2(\mathbb{R}^d)$.
Thus, in this case the STFT is an isometry from $\boldsymbol{L}^2(\mathbb{R}^d)$ into $\boldsymbol{L}^2(\mathbb{R}^{2d})$.

Corollary 1.4.2 (Inversion formula for the STFT)**.** *Suppose that $g, \gamma \in \boldsymbol{L}^2(\mathbb{R}^d)$ and $\langle g, \gamma \rangle \neq 0$. Then for all $f \in \boldsymbol{L}^2(\mathbb{R}^d)$*

$$f = \frac{1}{\langle g, \gamma \rangle} \iint_{\mathbb{R}^{2d}} \mathcal{V}_g f(x, \omega)\, \mathrm{M}_\omega\, \mathrm{T}_x\, \gamma d\omega dx. \tag{1.29}$$

Proof. See [Grö01, Cor. 3.2.3.] □

Remark: Equation (1.29) is understood in the vector-valued weak sense, i.e

$$\langle f, h \rangle = \frac{1}{\langle g, \gamma \rangle} \iint_{\mathbb{R}^{2d}} \mathcal{V}_g f(x, \omega) \overline{\langle h, \mathrm{M}_\omega\, \mathrm{T}_x\, \gamma \rangle} d\omega dx \tag{1.30}$$

for all $h \in \boldsymbol{L}^2(\mathbb{R}^d)$.

Processing a signal (or an image/video) for compression or modifying the spectrum, etc., usually includes these three steps:

1) *Analysis*: Here the decomposition of the signal f into a time-frequency representation is done, e.g. the short-time Fourier transform.

2) *Processing*: In this TF-representation, we can now do some modifications, like truncation to an area of interest (a certain frequency spectrum in a certain time window). Or we might multiply pointwise with some function on the TF-plane, maybe to amplify certain areas and reduce others.

3) *Synthesis*: After that one can resynthesis the modified function with the inversion formula to get back a processed function.

1.5. The Space $\boldsymbol{S}_0(\mathbb{R}^d)$

As suggested before, we are going to introduce the space $\boldsymbol{S}_0(\mathbb{R}^d)$ which is much easier to handle than the Schwartz space.

Definition 1.5.1 (The space $\boldsymbol{S}_0(\mathbb{R}^d)$)**.** *Let g be the normalized Gaussian $g(x) = \phi_1(x) = e^{-\pi x^2}$ (or equvivalently any arbitrary fixed Schwartz class window $g \neq 0$), then the so-called* Feichtinger's algebra $\boldsymbol{S}_0(\mathbb{R}^d)$ *is defined by*

$$\boldsymbol{S}_0(\mathbb{R}^d) := \{f \in \boldsymbol{L}^1(\mathbb{R}^d), \|\mathcal{V}_g f\|_{\boldsymbol{L}^1(\mathbb{R}^d \times \widehat{\mathbb{R}}^d)} < \infty\} \tag{1.31}$$

It is one of a class of function spaces called modulation spaces $\boldsymbol{M}_m^{p,q}$ and $\boldsymbol{S}_0(\mathbb{R}^d)$ coincides with the space $\boldsymbol{M}^1 := \boldsymbol{M}_0^{1,1}$ which is the smallest Banach space that is isometrically invariant under translations and modulations. $\boldsymbol{S}_0(\mathbb{R}^d)$ is a Banach space w.r.t. the norm

$$\|f\|_{\boldsymbol{S}_0} := \|\mathcal{V}_g f\|_{\boldsymbol{L}^1(\mathbb{R}^d \times \widehat{\mathbb{R}}^d)}.$$

More about these modulation spaces can be found in [Grö01].

Another class of functions/distributions are the so-called *Wiener Amalgam spaces* which consist of elements with certain global behavior of local properties. An exact description can be found in [Fei92], but as we dont need a description that general, we only introduce a case for generally well-known spaces.

Definition 1.5.2 (The Wiener Amalgam Spaces $\boldsymbol{W}(\boldsymbol{C}, \boldsymbol{\ell}^p)$)**.** *The Wiener Amalgam Spaces over the continuous functions $f \in \boldsymbol{C}(\mathbb{R}^d)$ with respect to $\boldsymbol{\ell}^p$ are defined as*

$$\boldsymbol{W}(\boldsymbol{C}, \boldsymbol{\ell}^p) := \left\{ f \in \boldsymbol{C}(\mathbb{R}^d), \|f\|_{\boldsymbol{W}(\boldsymbol{C},\boldsymbol{\ell}^p)} = \left(\sum_{j \in \mathbb{Z}^d} \left(\sup_{x \in j+Q} |f(x)| \right)^p \right)^{1/p} < \infty \right\} \tag{1.32}$$

where $Q = [0,1]^d$.

This means that we consider continuous functions whose local suprema form a sequence in $\boldsymbol{\ell}^p(\mathbb{Z}^d)$.

We now introduce the notion of a (Banach) Gelfand triple, which consists of three spaces in some manner embeded in each other.

Definition 1.5.3 (Banach Gelfand Triple)**.** *A combination of the three spaces $(\boldsymbol{B}, \mathcal{H}, \boldsymbol{B}')$ is called a* Banach Gelfand triple*, if $\mathcal{H}$ is a Hilbert space and $\boldsymbol{B}$ a Banach space which is continuously and densely embedded into $\mathcal{H}$. Also $\mathcal{H}$ is weak*-continuously and densely embedded into the dual space $\boldsymbol{B}'$.*

$$\boldsymbol{B} \underset{densely}{\overset{cont.}{\hookrightarrow}} \mathcal{H} \underset{densely}{\overset{\omega^*\text{-}cont.}{\hookrightarrow}} \boldsymbol{B}'$$

Here, as opposed to the more general Gelfand triple, the two outer spaces $\boldsymbol{B}$ and $\boldsymbol{B}'$ are both Banach spaces.

A *unitary (Banach-)Gelfand triple isomorphism* from one (Banach) Gelfand triple $(\boldsymbol{B}_1, \mathcal{H}_1, \boldsymbol{B}_1')$ to another one $(\boldsymbol{B}_2, \mathcal{H}_2, \boldsymbol{B}_2')$ is an operator, which is on all three levels isomorphic and between $\mathcal{H}_1$ and $\mathcal{H}_2$ unitary. One can denote that an operator U is an unitary Gelfand triple isomorphism by the so-called *Gelfand bracket*:

$$\langle f_1, f_2 \rangle_{(\boldsymbol{B}_1, \mathcal{H}_1, \boldsymbol{B}_1')} = \langle U f_1, U f_2 \rangle_{(\boldsymbol{B}_2, \mathcal{H}_2, \boldsymbol{B}_2')}$$

Here we have the inner product at the Hilbert space level and linear functionals at the Banach space levels, i.e. $\langle f_1, f_2 \rangle = f_1(f_2) = f_2(f_1)$ for $f_1 \in \boldsymbol{B}, f_2 \in \boldsymbol{B}'$.

Any unitary mapping $U : \mathcal{H}_1 \to \mathcal{H}_2$ extends to a unitary Gelfand triple isomorphism if and only if the restrictions of U and U^* are bounded linear operators from $\boldsymbol{B}_1$ to $\boldsymbol{B}_2$. More details about Banach Gelfand triples and isomorphisms on them can be found in [FK98] and [Ban10].

An example for a *Gelfand triple* (with no Banach spaces $\boldsymbol{B}$) is $(\boldsymbol{\mathcal{S}}, \boldsymbol{L}^2, \boldsymbol{\mathcal{S}}')$.

Proposition 1.5.1. *Some fundamental properties of* $\boldsymbol{S}_0(\mathbb{R}^d)$

(i) $\boldsymbol{S}_0(\mathbb{R}^d)$ *is a Banach space with dual space*

$$\boldsymbol{S}_0'(\mathbb{R}^d) = \{f \in \boldsymbol{\mathcal{S}}', \|\mathcal{V}_g f\|_{\boldsymbol{L}^\infty(\mathbb{R}^d \times \widehat{\mathbb{R}}^d)} < \infty\}.$$

Together with $\boldsymbol{L}^2(\mathbb{R}^d)$*, they form a Banach Gelfand triple* $\big(\boldsymbol{S}_0(\mathbb{R}^d), \boldsymbol{L}^2(\mathbb{R}^d), \boldsymbol{S}_0'(\mathbb{R}^d)\big)$.

(ii) For a function $f \in \boldsymbol{S}_0$*,* f *and its Fourier transform* $\hat{f}$ *are in* $\boldsymbol{W}(\boldsymbol{C}, \boldsymbol{\ell}^1)$ *and therefore continuous.*

(iii) If $f, g \in \boldsymbol{S}_0(\mathbb{R}^d)$*, then the STFT* $\mathcal{V}_g f \in \boldsymbol{W}(\boldsymbol{C}, \boldsymbol{\ell}^1)(\mathbb{R}^{2d})$ *(in fact in* $\boldsymbol{S}_0(\mathbb{R}^{2d})$*).*

(iv) All time-frequency shifts $\pi(\lambda)$ *and the Fourier transform* $\mathcal{F}$ *are unitary Gelfand triple isomorphisms on* $\big(\boldsymbol{S}_0(\mathbb{R}^d), \boldsymbol{L}^2(\mathbb{R}^d), \boldsymbol{S}_0'(\mathbb{R}^d)\big)$*, especially* $\boldsymbol{S}_0(\mathbb{R}^d)$ *is isometrically invariant under TF-shifts and the (partial) Fourier transform.*

(v) $\boldsymbol{S}_0(\mathbb{R}^d) \otimes \boldsymbol{S}_0(\mathbb{R}^d) \subseteq \boldsymbol{S}_0(\mathbb{R}^{2d})$

Proof. (i)-(iii) A proof for the claims here can be found in [Grö01, Chapter 11/12]. (iv) We only show the straight foreward calculations for the isometric invariances under $\mathcal{F}$ and $\pi(\lambda)$, the rest can befound in the book cited above.

$$\begin{aligned}\|\mathcal{F}f\|_{\boldsymbol{S}_0} &= \int_{\mathbb{R}^{2d}} |\langle \mathcal{F}f, \mathrm{M}_\omega \mathrm{T}_x\, g\rangle| d\omega dx = \int_{\mathbb{R}^{2d}} |\langle \mathcal{F}f, \mathrm{M}_\omega \mathrm{T}_x\, \mathcal{F}g\rangle| d\omega dx \\ &= \int_{\mathbb{R}^{2d}} |\langle \mathcal{F}f, \mathcal{F}(\mathrm{T}_{-\omega} \mathrm{M}_{-x}\, g)\rangle| d\omega dx = \int_{\mathbb{R}^{2d}} |\langle f, \mathrm{T}_{-\omega} \mathrm{M}_{-x}\, g\rangle| d\omega dx = \|f\|_{\boldsymbol{S}_0}\end{aligned}$$

$$\begin{aligned}\|\pi(\lambda)f\|_{\boldsymbol{S}_0} &= \int_{\mathbb{R}^{2d}} |\langle \pi(\lambda)f, \pi(\lambda')g\rangle| d\omega' dx' = \int_{\mathbb{R}^{2d}} |\langle f, \pi^*(\lambda)\pi(\lambda')g\rangle| d\omega' dx' \\ &= \int_{\mathbb{R}^{2d}} |\langle f, e^{2\pi i x(\omega'-\omega)}\pi(\lambda'-\lambda)g\rangle| d\omega' dx' = \int_{\mathbb{R}^{2d}} |\langle f, \pi(\lambda'-\lambda)g\rangle| d\omega' dx' = \|f\|_{\boldsymbol{S}_0}\end{aligned}$$

(v) Note that the tensor product of two d-dimensional Gaussians is the 2d-dimensional Gaussian $\phi_1^{2d} = \phi_1^d \otimes \phi_1^d$. Now by writting $\lambda = (\lambda_1, \lambda_2)$ with $\omega = (\omega_1, \omega_2)$ and $x = (x_1, x_2)$, we also have $\pi(\lambda)g = \pi(\lambda_1)g \otimes \pi(\lambda_2)g$ and therefore

$$\begin{aligned}\|f_1 \otimes f_2\|_{\boldsymbol{S}_0} &= \int_{\mathbb{R}^{4d}} |\langle f_1 \otimes f_2, \pi(\lambda_1)g \otimes \pi(\lambda_2)g\rangle| d\lambda_1 d\lambda_2 \\ &= \int_{\mathbb{R}^{4d}} |\langle f_1, \pi(\lambda_1)g\rangle| \cdot |\langle f_2, \pi(\lambda_2)g\rangle| d\lambda_1 d\lambda_2 = \|f_1\|_{\boldsymbol{S}_0} \|f_2\|_{\boldsymbol{S}_0}\end{aligned}$$

□

.

Remark: We have $\boldsymbol{\mathcal{S}} \subseteq \boldsymbol{S}_0$ and $\boldsymbol{S}_0' \subseteq \boldsymbol{\mathcal{S}}'$ (with continuous embeddings). In [Wei07] it

is shown that for $f \in \boldsymbol{L}^2(\mathbb{R}^d)$ and $g, \gamma \in \boldsymbol{S}_0(\mathbb{R}^d)$, the Riemannian sum for the inversion formula of the short-time Fourier transform [(1.29)]

$$S_m f(t) := \frac{1}{\langle g, \gamma \rangle} \frac{1}{m^{2d}} \sum_{k \in \mathbb{Z}^d} \sum_{n \in \mathbb{Z}^d} \mathcal{V}_g f(k/m, n/m)\, \mathrm{M}_{n/m}\, \mathrm{T}_{k/m}\, \gamma(t)$$

converges unconditionally in $\boldsymbol{L}^2(\mathbb{R}^d)$ norm and $\lim_{m\to\infty} S_m f = f$. Also there exist $g = \gamma \in \boldsymbol{L}^2(\mathbb{R}^d)$, such that the convergence in $\boldsymbol{L}^2(\mathbb{R}^d)$ norm cannot be guaranteed for all $f \in \boldsymbol{L}^2(\mathbb{R}^d)$.

We see that we have some nice properties for $\boldsymbol{S}_0(\mathbb{R}^d)$, but with it we have some nice properties for its dual $\boldsymbol{S}_0'(\mathbb{R}^d)$. It is as well invariant under the Fourier transform and all TF-shifts. And one can show that distributions in $\boldsymbol{S}_0'(\mathbb{R}^d)$ are not too "wild", but the space still contains important functions like the pure frequencies $e^{2\pi i x\cdot\omega}$ and, because it is Fourier invariant as well, also their Fourier transforms the δ-distributions.

The action of an integral operators T on $f \in \boldsymbol{L}^2(\mathbb{R}^d)$ is defined by integrating f over the product with the distributional kernel $\kappa(T)$ defined on $\mathbb{R}^{2d}$.

$$Tf(x) = \int_{\mathbb{R}^d} \kappa(T)(x, y) f(y) dy \tag{1.33}$$

The well known class of Hilbert-Schmidt operators $\mathcal{HS}$ consists of all T with kernel $\kappa(T) \in \boldsymbol{L}^2(\mathbb{R}^d \times \mathbb{R}^d)$. Together with the inner product

$$\langle T, S \rangle_{\mathcal{HS}} := \langle \kappa(T), \kappa(S) \rangle_{\boldsymbol{L}^2(\mathbb{R}^d \times \mathbb{R}^d)}$$

$(\mathcal{HS}, \|.\|_{\mathcal{HS}})$ is a Hilbert space of operators with the induced norm $\|T\|_{\mathcal{HS}} := \sqrt{\langle T, T \rangle_{\mathcal{HS}}} = tr(T * T)$.

Theorem 1.5.1. *Let $\mathcal{B} \subseteq \mathcal{L}(\boldsymbol{S}_0', \boldsymbol{S}_0)$ be the Banach space of bounded linear operators T from $\boldsymbol{S}_0'$ to $\boldsymbol{S}_0$ that additionally map weak*-convergent sequences $(\langle f, g_n \rangle_{\boldsymbol{S}_0(\mathbb{R}^d)} \to \langle f, g_0 \rangle_{\boldsymbol{S}_0(\mathbb{R}^d)}, \quad \forall f \in \boldsymbol{S}_0(\mathbb{R}^d))$ into $\boldsymbol{S}_0$-norm convergent sequences $(\|Tg_n - Tg_0\|_{\boldsymbol{S}_0(\mathbb{R}^d)} \to 0)$. Then the dual $\mathcal{B}'$ can be identified with $\mathcal{L}(\boldsymbol{S}_0, \boldsymbol{S}_0')$ with the duality pairing being the natural extension of the inner product for $\mathcal{HS}$-operators.*
Together with the Hilbert space of $\mathcal{HS}$-operators on $\boldsymbol{L}^2(\mathbb{R}^d)$, $(\mathcal{B}, \mathcal{HS}, \mathcal{B}')$ forms a Banach Gelfand triple that is isomorphic to $(\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')(\mathbb{R}^{2d})$ via the correspondence

$$(\mathcal{B}, \mathcal{HS}, \mathcal{B}') \ni T \longleftrightarrow \kappa(T) \in (\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')(\mathbb{R}^{2d}) \tag{1.34}$$

with $\kappa(T)$ being the distributional kernel of the operator T.

Proof. A proof can be found in [FK98, Theorems 7.4.1/2] □

This correspondence of operators T and their kernels $\kappa(T)$ implies validity of the extended inner product:

$$\langle T, S \rangle_{(\mathcal{B}, \mathcal{HS}, \mathcal{B}')} := tr(T * S) = \langle \kappa(T), \kappa(S) \rangle_{(\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')(\mathbb{R}^{2d})}$$

Similar results like the one in Theorem 1.5.1 above will be shown later for different representations of operators in $(\mathcal{B}, \mathcal{HS}, \mathcal{B}')$.

Chapter 2.

Frames and Riesz Bases

When one works with finite dimensional vector spaces V, a basis is a set of linear independent elements $v_i, i = 1, \ldots, n$, which span the the vector space V. Important is, that each element in V can then be expressed as a linear combination with v_i in a unique way, i.e. the coefficiants are uniquely determined. For infinite dimensional spaces the notion of a basis has the same concept. One wants a set of elements that allow an expansion with unique coefficiants.

It is a nice mathematical way to approach decompositon of elements, but it is also a tight one, as one has not much freedom. The idea of a frame is one that allows more elements than needed for an expansion, i.e. two bases combined. In that way the coefficiants are not unique anymore, but on the other hand one gets more options for the choice of them.

2.1. Bases in Banach and Hilbert Spaces

Definition 2.1.1 (Schauder Basis)**.** *Let $\boldsymbol{B}$ be a Banach space. A sequence of vectors $\{e_k\}_{k=1}^{\infty}$ belonging to $\boldsymbol{B}$ is a (Schauder) basis for $\boldsymbol{B}$ if, for each $f \in \boldsymbol{B}$, there exist unique scalar coefficiants $\{c_k(f)\}_{k=1}^{\infty}$ such that*

$$f = \sum_{k=1}^{\infty} c_k(f) e_k. \tag{2.1}$$

Here the sum converges for the given order $k = 1, \ldots, \infty$, it converges unconditionally if and only if it converges for all permutations of the basis elements e_k. In this case we call it an *unconditional basis*.

In the Definition 2.1.1, the coefficiants are given by linear functionals $\{c_k\}_{k=1}^{\infty}$ from the dual space $\boldsymbol{B}'$. $\{c_k\}_{k=1}^{\infty}$ and $\{e_k\}_{k=1}^{\infty}$ are *biorthogonal* by means of the following definition.

Definition 2.1.2. *Two sequences $\{f_k\}_{k=1}^{\infty}$ in a Banach space B and $\{g_k\}_{k=1}^{\infty}$ in its dual $\boldsymbol{B}'$ are* biorthogonal*, if*

$$g_i(f_j) = \delta_{i,j}$$

for all i, j.

If we consider a Hilbert space $\mathcal{H}$, we can write this with an inner product, due to Riesz' Representation Theorem. This means that we get a basis expansion of the form

$$f = \sum_{k=1}^{\infty} \langle f, c_k \rangle_{\mathcal{H}} e_k. \tag{2.2}$$

For an orthonormal basis in $\mathcal{H}$, we have $\langle e_i, e_j\rangle_{\mathcal{H}} = \delta_{i,j}$, so it is biorthogonal to itself and we get

$$f = \sum_{k=1}^{\infty} \langle f, e_k\rangle_{\mathcal{H}} e_k.$$

If one has a basis for a Hilbert space $\mathcal{H}$, it can be shown [Chr03, Thm. 3.3.2] that there exists a unique biorthogonal basis for which (2.2) holds.

2.1.1. Riesz Bases

In a Hilbert space, all orthonormal bases are precisely the sets $\{Ue_k\}_{k=1}^{\infty}$ for an arbitrary orthonormal basis $\{e_k\}_{k=1}^{\infty}$ and a unitary operator $U : \mathcal{H} \to \mathcal{H}$ ([Chr03, Thm. 3.4.7]).

For the introduction of a Riesz basis, we weaken the condition on the operator U to get other bases for $\mathcal{H}$.

Definition 2.1.3 (Riesz bases)**.** *A Riesz basis for $\mathcal{H}$ is a family of the form $\{f_k\}_{k=1}^{\infty} = \{Ve_k\}_{k=1}^{\infty}$, where $\{e_k\}_{k=1}^{\infty}$ is an orthonormal basis for $\mathcal{H}$ and $V : \mathcal{H} \to \mathcal{H}$ is a bounded bijective operator.*

A Riesz basis is always a (Schauder) basis and its dual (biorthogonal) basis is again a Riesz basis. Moreover the expansion (2.2) for $\{f_k\}_{k=1}^{\infty}$ and its dual converges unconditionally.

$\{f_k\}_{k=1}^{\infty}$ is a *complete* sequence in $\mathcal{H}$, if it holds that $\overline{span}\{f_k\}_{k=1}^{\infty} = \mathcal{H}$. The next theorem gives equivalent formulations for a Riesz basis.

Theorem 2.1.1. *For a sequence $\{f_k\}_{k=1}^{\infty}$ in $\mathcal{H}$, the following conditions are equivalent:*

(i) *$\{f_k\}_{k=1}^{\infty}$ is a Riesz basis for $\mathcal{H}$.*

(ii) *$\{f_k\}_{k=1}^{\infty}$ is complete in $\mathcal{H}$, and there exist constants $A, B > 0$ such that for every finite scalar sequence $\{c_k\}$ one has*

$$A \sum |c_k|^2 \leq \| \sum c_k f_k \|^2 \leq B \sum |c_k|^2 \tag{2.3}$$

(iii) *$\{f_k\}_{k=1}^{\infty}$ is complete, and its Gram matrix $\{\langle f_i, f_j\rangle\}_{i,j=1}^{\infty}$ defines a bounded, invertible operator on $\boldsymbol{\ell}^2(\mathbb{N})$.*

Proof. A complete proof can be found in [Chr03]. □

If the conditions in (ii) or (iii) are given with the exception that $\{f_k\}_{k=1}^{\infty}$ is not a complete sequence in $\mathcal{H}$, then it is a Riesz basis for $\overline{span}\{f_k\}_{k=1}^{\infty}$.

The bounds A and B in (2.3) are called *upper* resp. *lower Riesz bound*. They are not unique, as a greater B or a smaller A is just as sufficient. Therefore we call the infimum of all B resp. the supremum of all A the *optimal Riesz bounds*.

2.2. From Bases to Frames

As stated above, we want to extend our tools away from bases to get higher flexibility. One might not always find a basis with the properties wanted or if one does slight modifications to a basis, it might cease to be a basis at all. This is why we introduce the concept of *frames*. Intuitively it is like a basis but with more elements than needed. This means, that we loose the uniqueness of the coefficiants in (2.1), but we still have the expansion property.

First we define a property that sequences in a Hibert space can have.

Definition 2.2.1 (Bessel sequences)**.** *A sequence* $\{g_k\}_{k=1}^{\infty}$ *in* $\mathcal{H}$ *is called a* Bessel sequence *with* Bessel bound B *if there exists such a constant* $B > 0$ *with*

$$\sum_{k=1}^{\infty} |\langle f, g_k\rangle|^2 \leq B\|f\|^2, \quad \forall f \in \mathcal{H}. \tag{2.4}$$

Now, in the same spirit, we define sequences which are Bessel and in some way have an "opposite inequality" as well. Which means nothing more or less than that we have a bound from below as B is from above.

Definition 2.2.2 (Frames)**.** *A sequence* $\{g_k\}_{k=1}^{\infty}$ *of elements in* $\mathcal{H}$ *is called a* frame *for* $\mathcal{H}$ *if there exist constants* $A, B > 0$ *such that*

$$A\|f\|^2 \leq \sum_{k=1}^{\infty} |\langle f, g_k\rangle|^2 \leq B\|f\|^2, \quad \forall f \in \mathcal{H}. \tag{2.5}$$

The numbers A and B in (2.5) are called frame bounds. As for Riesz bases, they are not unique and we call the infimum over all B the *optimal upper frame bound* B_{opt} and the supremum over all A the *optimal lower frame bound* A_{opt}. If we can choose frame bounds with $A = B$, then we call the frame *tight*.

The frame bounds are in a way bounds that tell us where the frame is concentrated. It can be understood in the way that if A is small, then one "direction" in $\mathcal{H}$ is underrepresented. In the same way if B is large, this means that one "direction" in $\mathcal{H}$ is overrepresented.

Now we want to shortly recap the notion of the Fourier series to get motivated for the frame expansion. The Fourier series coefficiants of a signal in $\boldsymbol{L}^2(0,1)$ are given by

$$\{c_k\}_{k=1}^{\infty} := \{\langle f, e_k\rangle\}_{k=1}^{\infty}$$

where $\{e_k\}_{k=1}^{\infty}$ is the orthonormal basis $\{e^{2\pi ikx}\}_{k=1}^{\infty}$ and the inner product is

$$\langle f, e_k\rangle = \int_0^1 f(x)e^{-2\pi ikx}dx.$$

With that the Fourier series is given by

$$f = \sum_{k=1}^{\infty} c_k e_k = \sum_{k=1}^{\infty} \langle f, e_k\rangle e_k.$$

For the expansion of a signal with a frame we define the *analysis* and the *synthesis operator*.

Definition 2.2.3. *For a frame* $\{g_k\}_{k=1}^{\infty}$ *in* $\mathcal{H}$ *the* analysis *or* coefficient operator *is defined as*

$$C : \mathcal{H} \to \ell^2(\mathbb{N}), \quad Cf := \{\langle f, g_k\rangle_{\mathcal{H}}\}_{k=1}^{\infty} \tag{2.6}$$

and the adjoint operator called synthesis *or* reconstruction operator *as*

$$D : \ell^2(\mathbb{N}) \to \mathcal{H}, \quad D\{c_k\}_{k=1}^{\infty} := \sum_{k=1}^{\infty} c_k g_k. \tag{2.7}$$

One can show that D *defines a bounded operator, [Chr03, 3.2.3]. By composing* C *and* $D = C^*$ *we get the* frame operator

$$S : \mathcal{H} \to \mathcal{H}, \quad Sf = DCf = \sum_{k=1}^{\infty} \langle f, g_k\rangle_{\mathcal{H}} g_k. \tag{2.8}$$

These operators can be defined for sequences instead of frames, but for a frame, the frame operator has some important properties.

Lemma 2.2.1. *Let* $\{g_k\}_{k=1}^{\infty}$ *be a frame with frame operator* S *and frame bounds* A *and* B*. Then the following holds:*

(i) S *is bounded, invertible, self-adjoint and positive.*

(ii) $\{S^{-1}g_k\}_{k=1}^{\infty}$ *is a frame with bounds* B^{-1}, A^{-1} *and frame operator* S^{-1}*.* B_{opt}^{-1} *and* A_{opt}^{-1} *are the optimal frame bounds for* $\{S^{-1}g_k\}_{k=1}^{\infty}$*.*

Proof. This should only give an idea for the proof (compare [Chr03]).

(i) S is bounded as a composition of bounded operators and self-adjoint as $S^* = (DC)^* = C^*D^* = DC = S$.
The inequality (2.5) reads as $A\|f\|^2 \leq \langle Sf, f\rangle \leq B\|f\|^2$ for all $f \in \mathcal{H}$, so S is positive. By notating $U \leq V$ for operators which satisfy $\langle Uf, f\rangle \leq \langle Vf, f\rangle$ for all $f \in \mathcal{H}$, it has the form $A\,\mathrm{Id} \leq S \leq B\,\mathrm{Id}$ and furthermore $0 \leq \mathrm{Id} - B^{-1}S \leq \frac{B-A}{B}\,\mathrm{Id}$.

$$|||\mathrm{Id} - B^{-1}S||| = \sup_{\|f\|=1} |\langle(\mathrm{Id} - B^{-1}S)f, f\rangle| \leq \frac{B-A}{B} < 1,$$

so by Theorem 1.1.2 (Neumann series) we get invertibility.

(ii) Let S^{-1} be the inverse of S. Multiplying $A\,\mathrm{Id} \leq S \leq B\,\mathrm{Id}$ with S^{-1} yields

$$B^{-1}\,\mathrm{Id} \leq S^{-1} \leq A^{-1}\,\mathrm{Id}$$

and therefore we have a frame $\{S^{-1}g_k\}_{k=1}^{\infty}$ with frame bounds B^{-1}, A^{-1}

□

Remark: We see that for a tight frame ($A = B$) we have $S = A\,\mathrm{Id}$ and conversely.

We obtain that every $f \in \mathcal{H}$ has the expansion

$$f = S^{-1}Sf = \sum_{k=1}^{\infty} \langle f, g_k\rangle_{\mathcal{H}} S^{-1}g_k \tag{2.9}$$

and

$$f = SS^{-1}f = \sum_{k=1}^{\infty} \langle f, S^{-1}g_k\rangle_{\mathcal{H}} g_k \tag{2.10}$$

and thus we found something like a biorthogonal sequence for our frame. We call $\{S^{-1}g_k\}_{k=1}^{\infty}$ the *canonical dual frame* and denote it by $\{\tilde{g}_k\}_{k=1}^{\infty}$. It is one of maybe many dual frames with an expansion

$$f = \sum_{k=1}^{\infty} \langle f, \tilde{g}_k\rangle_{\mathcal{H}} g_k = \sum_{k=1}^{\infty} \langle f, g_k\rangle_{\mathcal{H}} \tilde{g}_k$$

and the one whose frame expansion coefficiants $\{\langle f, \tilde{g}_k\rangle_{\mathcal{H}}\}_{k=1}^{\infty}$ have minimal $\boldsymbol{\ell}^2$-norm. If a frame is a basis, then it is a Riesz basis. But if it is no basis, then there always exist seweral other dual frames, than the canonical dual, as shown in [Chr03, 5.6.1].

The dual frame is our concept for reconstructing the signal. The problem is that one can not be sure that a dual frame of a frame in $\boldsymbol{L}^2$ resp. $\boldsymbol{S}_0$ is also in $\boldsymbol{L}^2$ resp. $\boldsymbol{S}_0$. For tight frames this is given, as the canonical dual is just $\{\frac{1}{A}g_k\}_{k=1}^{\infty}$ so it has the same structure. It is known that the tight window is rather expensive to compute for applications. Therefore we want simpe criteria for a nice dual frame.

One kind of quality criteria for a frame are its optimal bounds A_{opt} and B_{opt}. One can see from $A\|f\|^2 \leq \langle Sf, f\rangle \leq B\|f\|^2$ and the form of the operator norm for self-adjoint operators that

$$B_{opt} = \sup_{\|f\|=1} \langle Sf, f\rangle = |||S|||$$

and similarly from $\frac{1}{B}\|f\|^2 \leq \langle S^{-1}f, f\rangle \leq \frac{1}{A}\|f\|^2$ that

$$A_{opt} = \frac{1}{|||S^{-1}|||}$$

via $|||S^{-1}||| = \sup_{\|f\|=1}\langle S^{-1}f, f\rangle = \frac{1}{A_{opt}}$. These respective values for optimal frame bounds correspond to the largest and smallest singular values of S.

As the frame operators S and S^{-1} are positive and self-adjoint, they can be written as $S = T^2 = S^{\frac{1}{2}}S^{\frac{1}{2}}$ and $S^{-1} = S^{-\frac{1}{2}}S^{-\frac{1}{2}}$ with $S^{\frac{1}{2}}$ and $S^{-\frac{1}{2}}$ being again self-adjoint.

$$f = S^{-\frac{1}{2}}S(S^{-\frac{1}{2}}f) = \sum_{k=1}^{\infty} \langle f, S^{-\frac{1}{2}}g_k\rangle_{\mathcal{H}} S^{-\frac{1}{2}}g_k$$

As $S^{-\frac{1}{2}}SS^{-\frac{1}{2}} = \mathrm{Id}$ and

$$\begin{aligned}\sum_{k=1}^{\infty} |\langle f, S^{-\frac{1}{2}}g_k\rangle|^2 &= \sum_{k=1}^{\infty} |\langle S^{-\frac{1}{2}}f, g_k\rangle|^2 = \sum_{k=1}^{\infty} \langle S^{-\frac{1}{2}}f, g_k\rangle\langle g_k, S^{-\frac{1}{2}}f\rangle \\ &= \langle SS^{-\frac{1}{2}}f, S^{-\frac{1}{2}}f\rangle = \langle S^{-\frac{1}{2}}SS^{-\frac{1}{2}}f, f\rangle = \|f\|^2\end{aligned}$$

one can see that $\{S^{-\frac{1}{2}}g_k\}_{k=1}^{\infty}$ is a tight frame with bounds $A = B = 1$ from now on called the *canonical tight frame*. For this frame we have the nice property, that the dual for reconstruction is again the same frame. This seems to be quite similar to the Fourier series with an orthonormal basis. For any tight frame, the redundancy of the frame is equal to the (optimal) frame bound. But the canonical tight (redundancy 1) is an orthonormal basis if and only if it is normalized.

Chapter 3.

Gabor Analysis

3.1. Discrete Gabor Analysis

One major point of Gabor analysis is to find a *discretized* version of the STFT inversion formula in (1.29), as this is a highly redundant system of TF-shifts over the whole TF-plane. What we want is to reduce this redundancy by choosing discrete points in the TF-plane where we sample $\mathcal{V}_g f$. But we still want an expansion for our function f. This means finding functions $g, \gamma \in \boldsymbol{L}^2(\mathbb{R}^d)$ with $\langle g, \gamma \rangle = 1$ (w.l.o.g., as they can be accordingly normalized) and a discrete subset of the TF-plane $\Lambda \subseteq \mathbb{R}^d \times \widehat{\mathbb{R}}^d$, such that

$$f = \sum_{\lambda \in \Lambda} \mathcal{V}_g f(\lambda) \pi(\lambda) \gamma, \quad f \in \boldsymbol{L}^2(\mathbb{R}^d), \tag{3.1}$$

converges in $\boldsymbol{L}^2(\mathbb{R}^d)$. To do this we need to introduce *lattices* as the appropriate setting for such *sampling grids*.

Definition 3.1.1 (Lattice)**.** *A* discrete subgroup $\Lambda \subseteq \mathbb{R}^d \times \widehat{\mathbb{R}}^d$ *of the TF-plane is called a* lattice *if it has the form* $\Lambda = A\mathbb{Z}^{2d}$*. Here* A *is an invertible* $2d \times 2d$*-Matrix over* $\mathbb{R}$ *called the generator of the lattice. This matrix is not unique for a lattice. For all matrices* U *with* $det(U) = 1$*,* $U \cdot A$ *generates the same lattice.*
For $a, b > 0$ *a lattice of the form* $\Lambda = a\mathbb{Z}^d \times b\mathbb{Z}^d$ *is called a* seperable lattice *or* product lattice*.*

For a lattice we define the *size* as $s(\Lambda) = |det(A)|$ and the *redundancy* $\text{red}_\Lambda = s(\Lambda)^{-1}$. By doing a singular value decomposition of A, one gets $A = U\Sigma V$ with unitary U and V and $\Sigma = diag(\sigma_1, \ldots, \sigma_{2d})$. U and V do not contribute to the determinant (they are unitary). We get

$$s(\Lambda) = |\sigma_1 \ldots \sigma_{2d}|$$

and therefore

$$\text{red}_\Lambda = \frac{1}{|\sigma_1 \ldots \sigma_{2d}|}.$$

The redundancy represents the average number of lattice points in large balls.

For a suitable window g, e.g. $g \in \boldsymbol{S}_0(\mathbb{R}^d)$, and a lattice $\Lambda \subseteq \mathbb{R}^d \times \widehat{\mathbb{R}}^d$, we call

$$f \mapsto c(\lambda) = \mathcal{V}_g f(\lambda) = \langle f, \pi(\lambda) g \rangle, \quad \lambda \in \Lambda \tag{3.2}$$

the *Gabor transform* of f with respect to g and Λ. Its inverse is then given by

$$c \mapsto \sum_{\lambda \in \Lambda} c(\lambda) \pi(\lambda) \gamma \tag{3.3}$$

for suitable γ. From this we get the Gabor expansion

$$f = \sum_{\lambda\in\Lambda} \langle f, \pi(\lambda)g\rangle \pi(\lambda)\gamma = \sum_{\lambda\in\Lambda} \langle f, \pi(\lambda)\gamma\rangle \pi(\lambda)g \tag{3.4}$$

Now the question arises, for which choices of g and Λ we get a frame $\{g_\lambda\}_{\lambda\in\Lambda} := \{\pi(\lambda)g\}_{\lambda\in\Lambda}$. We call $\{g_\lambda\}_{\lambda\in\Lambda}$ a *Gabor system* and denote it by $\mathcal{G}(\mathbf{g}, \Lambda)$. If $\mathcal{G}(\mathbf{g}, \Lambda)$ constitutes a frame, we call it a *Gabor frame*. This is the case if and only if the *Gabor frame operator*

$$S_{g,\Lambda}(f) = \sum_{\lambda\in\Lambda} \langle f, \pi(\lambda)g\rangle \pi(\lambda)g$$

is bounded and invertible on $\boldsymbol{L}^2(\mathbb{R}^d)$. It holds that if $\mathcal{G}(\mathbf{g}, \Lambda)$ is a Gabor frame for $\boldsymbol{L}^2(\mathbb{R}^d)$, then $s(\Lambda) \leq 1$, [Grö01, 9.4.8.]. So it makes sense to only investigate lattices with $s(\Lambda) \leq 1$ or equivalently $\mathrm{red}_\Lambda \geq 1$.

The special thing about a Gabor frame $\{g_\lambda\}_{\lambda\in\Lambda}$, in comparison with a general frame, is that it consists of a single function g, the so-called *atom* or *window*, shifted over a lattice in the TF-plane.

The Gabor frame operator $S_{g,\Lambda}$ commutes with all TF-shifts $\pi(\lambda')$ for $\lambda' \in \Lambda$, as

$$\begin{aligned}
\pi(\lambda')^{-1} S_{g,\Lambda} \pi(\lambda') f &= \sum_{\lambda\in\Lambda} \langle \pi(\lambda')f, \pi(\lambda)g\rangle \pi(\lambda')^{-1}\pi(\lambda)g \\
&= \sum_{\lambda\in\Lambda} \langle f, \pi^*(\lambda')\pi(\lambda)g\rangle e^{-2\pi i x'\omega'} \pi(-\lambda')\pi(\lambda)g \\
&= \sum_{\lambda\in\Lambda} \langle f, e^{-2\pi i x'\omega'}\pi(-\lambda')\pi(\lambda)g\rangle e^{-2\pi i x'\omega'} \pi(-\lambda')\pi(\lambda)g \\
&= \sum_{\lambda\in\Lambda} \langle f, e^{2\pi i x'(\omega-\omega')}\pi(\lambda-\lambda')g\rangle e^{2\pi i x'(\omega-\omega')} \pi(\lambda-\lambda')g \\
&= \sum_{\lambda\in\Lambda} \langle f, \pi(\lambda-\lambda')g\rangle \pi(\lambda-\lambda')g \\
&= \sum_{\eta\in\Lambda} \langle f, \pi(\eta)g\rangle \pi(\eta)g = S_{g,\Lambda} f
\end{aligned}$$

for $\lambda = (x, \omega)$ and $\lambda' = (x', \omega')$, see [Chr93, Lemma 6.6.]. Note that we have used the group structure of the lattice Λ in the last step.

If one takes a look at the canonical dual frame $\{S_{g,\Lambda}^{-1} g_\lambda\}_{\lambda\in\Lambda}$, it has again a Gabor frame with respect to the same lattice, as the frame operator $S_{g,\Lambda}$ commutes with all TF-shifts $\pi(\lambda)$, i.e.

$$S_{g,\Lambda}^{-1}(\pi(\lambda)g) = \pi(\lambda)(S_{g,\Lambda}^{-1} g).$$

The canonical dual frame is a suitable γ (dual frame) in (3.4), as

$$f = S_{g,\Lambda}^{-1} S_{g,\Lambda} f = \sum_{\lambda\in\Lambda} \langle f, \pi(\lambda)g\rangle \pi(\lambda)(S_{g,\Lambda}^{-1} g).$$

The same argument can be applied to the canonical tight frame $\{S_{g,\Lambda}^{-1/2} g_\lambda\}_{\lambda\in\Lambda}$ and we denote the canonical dual window with $\widetilde{g} := S_{g,\Lambda}^{-1} g$ and the canonical tight window with $g_t := S_{g,\Lambda}^{-1/2} g$. We get

$$f = \sum_{\lambda\in\Lambda} \langle f, \pi(\lambda)\widetilde{g}\rangle \pi(\lambda)g \tag{3.5}$$

and

$$f = \sum_{\lambda \in \Lambda} \langle f, \pi(\lambda) g_t \rangle \pi(\lambda) g_t. \tag{3.6}$$

With the notion of a lattice, we can define the *convolution over a lattice* Λ as

Definition 3.1.2. *For sequences $(c(\lambda)_{\lambda\in\Lambda})$ and $(d(\lambda)_{\lambda\in\Lambda})$ over the lattice Λ, the convolution over Λ is defined as*

$$(c *_\Lambda d)(\mu) := \sum_{\lambda \in \Lambda} c(\lambda) d(\mu - \lambda)$$

for $\mu \in \Lambda$.

Also the discrete Fourier transform can be extended for sequences c over any discrete group like Λ.

Definition 3.1.3 (Λ-Fourier transform)**.** *For a sequence $c \in \boldsymbol{\ell}^2(\Lambda)$, $\Lambda = A\mathbb{Z}^d$, we define the* Λ-Fourier transform $\mathcal{F}_\Lambda$ *as*

$$\mathcal{F}_\Lambda c(x) := \hat{c}(x) := \det(A) \sum_{\lambda \in \Lambda} c(\lambda) e^{2\pi i x \cdot \lambda} \tag{3.7}$$

For $\Lambda = \mathbb{Z}^d$, this is just the reversed discrete Fourier transform. So $\mathcal{F}_\Lambda$ is just a generalization for other subgroups than $\mathbb{Z}^d$.

Definition 3.1.4 (Dual lattice)**.** *For a lattice $\Lambda = A\mathbb{Z}^{2d}$, the so-called* dual *or* orthogonal lattice $\Lambda^\perp$ *is defined as*

$$\Lambda^\perp := \{\lambda^\perp \in \mathbb{R}^d \times \widehat{\mathbb{R}}^d : e^{2\pi i \lambda \lambda^\perp} = 1,\ \forall \lambda \in \Lambda\}.$$

A generator of the dual lattice is $A^\perp := (A^T)^{-1}$

Definition 3.1.5 (Adjoint lattice)**.** *For a lattice $\Lambda = A\mathbb{Z}^{2d}$, the* adjoint lattice Λ° *is defined as*

$$\Lambda^\circ = \{\lambda^\circ \in \mathbb{R}^d \times \widehat{\mathbb{R}}^d : \pi(\lambda)\pi(\lambda^\circ) = \pi(\lambda^\circ)\pi(\lambda),\ \forall \lambda \in \Lambda\}.$$

This is actually the annihilator group with respect to the symplectic Fourier transform (more on that can be found in [FK98]).

The adjont lattice is given by a rotation of the dual lattice $\Lambda^\perp$. To be more exact, it is given by

$$\Lambda^\circ = J\Lambda^\perp \text{ with } J = \begin{pmatrix} 0 & \mathrm{Id} \\ -\mathrm{Id} & 0 \end{pmatrix}$$

where Id is the identity matrix. As the generator of a lattice is not unique, the generator of the adjoint lattice is chosen to be defined by

$$A^\circ := JA^\perp J^T = J(A^{-1})^T J^T.$$

For the adjoint lattice we get $A^\circ = J(A^{-1})^T J^T = JV^{-T}\Sigma^{-T}U^{-T}J^T$ where $\Sigma^{-T} = diag(\frac{1}{\sigma_1} \ldots, \frac{1}{\sigma_{2d}})$. By that we get $\mathrm{red}_{\Lambda^\circ} = |\sigma_1 \ldots \sigma_{2d}|$ as J is unitary as well. Its easy to see now, that we have

$$\mathrm{red}_\Lambda \cdot \mathrm{red}_{\Lambda^\circ} = \frac{\sigma_1 \ldots \sigma_{2d}}{\sigma_1 \ldots \sigma_{2d}} = 1.$$

Example 3. *For a separable lattice* $\Lambda = a\mathbb{Z}\times b\mathbb{Z}$, *we have* $\Lambda = A\mathbb{Z}^2$ *with* $A = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}$. *As*

$$(A^T)^{-1} = \begin{pmatrix} 1/a & 0 \\ 0 & 1/b \end{pmatrix},$$

we get $\Lambda^\perp = A^\perp\mathbb{Z}^2 = \frac{1}{a}\mathbb{Z} \times \frac{1}{b}\mathbb{Z}$ *for the orthogonal lattice. The adjoint lattice is given by* $\Lambda^\circ = A^\circ\mathbb{Z}^2 = \frac{1}{b}\mathbb{Z} \times \frac{1}{a}\mathbb{Z}$ *as*

$$A^\circ = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1/a & 0 \\ 0 & 1/b \end{pmatrix} \cdot \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1/b & 0 \\ 0 & 1/a \end{pmatrix}.$$

The next theorem will introduce some fundamental results in Gabor analysis.

Theorem 3.1.1. *Let* Λ *be a lattice in* $\mathbb{R}^d \times \widehat{\mathbb{R}}^d$ *with adjoint lattice* Λ°. *Then, for* $g, \gamma \in \boldsymbol{S}_0(\mathbb{R}^d)$, *the following hold.*

1. *(Fundamental Identity of Gabor Analysis)*

$$\sum_{\lambda\in\Lambda} \langle f, \pi(\lambda)\gamma\rangle\langle\pi(\lambda)g, h\rangle = s(\Lambda) \sum_{\lambda^\circ\in\Lambda^\circ} \langle g, \pi(\lambda^\circ)\gamma\rangle\langle\pi(\lambda^\circ)f, h\rangle \tag{3.8}$$

   for all $f, h \in \boldsymbol{L}^2(\mathbb{R}^d)$, *where both sides converge absolutely.*

2. *(Wexler-Raz Identity)*

$$S_{g,\gamma,\Lambda} f = s(\Lambda) \cdot S_{f,\gamma,\Lambda^\circ} g \tag{3.9}$$

   for all $f \in \boldsymbol{L}^2(\mathbb{R}^d)$.

3. *(Janssen Representation)*

$$S_{g,\gamma,\Lambda} = s(\Lambda) \sum_{\lambda^\circ\in\Lambda^\circ} \mathcal{V}_g\gamma(\lambda^\circ)\pi(\lambda^\circ) \tag{3.10}$$

   where the series converges unconditionally in the strong operator sense.

Proof. To proof this would take quite a while and need advanced concepts in harmonic analysis. Therefore we refer the interested reader to [FLW07]. □

3.2. Dual Windows

The canonical dual window is usually denoted by $\widetilde{g}$ but sometimes we denote it by γ°.

For a window g, any function γ satisfying (3.4) is a dual window, i.e. yields a Gabor frame under the TF-shifts. This means that there might be more than just the canonical dual $\gamma^\circ = S_{g,\Lambda}^{-1} g$. In [Grö01, 7.6.1] it is shown (only for seperable lattices), that all dual windows γ are within a subspace of $\boldsymbol{L}^2(\mathbb{R}^d)$. More exact , if γ is a dual window for $\mathcal{G}(g, a, b)$, then $\gamma \in \gamma^\circ + \boldsymbol{\mathcal{K}}^\perp$ with $\boldsymbol{\mathcal{K}}$ being the closed linear span of $\mathcal{G}(g, \frac{1}{a}, \frac{1}{b})$. This means that for al dual windows γ we have $\gamma = \gamma^\circ + h$ for $h \in \boldsymbol{\mathcal{K}}^\perp$. In the same setting one can show that the canonical dual window is the one with minimal $\boldsymbol{L}^2$-norm.

3.2.1. The Canonical Dual Window $\widetilde{g}$ for Increasing red_Λ

In order to have a nice dual window for a given window $g \in \boldsymbol{S}_0(\mathbb{R}^d)$, we need it to be in $\boldsymbol{S}_0(\mathbb{R})$ as well. This has been investigated and proven, see [GL04]. There are conditions on Λ that ensure that we can get that.

Here we want to take a look at some properties of $\widetilde{g}$ and how it behaves if the *redundancy* of the lattice Λ gets higher.

Theorem 3.2.1. *Let $g \in \boldsymbol{S}_0(\mathbb{R}^d)$ be a Gabor window and $\Lambda = A\mathbb{Z}^{2d}$. If we have a sufficiently fine lattice λ, then the pair $(\mathbf{g}, \Lambda)$ generates a Gabor frame with $\widetilde{g} \in \boldsymbol{S}_0(\mathbb{R}^d)$.*

Proof. For $g \in \boldsymbol{S}_0(\mathbb{R}^d)$ we get $\mathcal{V}_g g \in \boldsymbol{W}(\boldsymbol{C}, \boldsymbol{\ell}^1)$ (see [FZ98, Lemma 3.2.15]). Looking at the Jannsen Representation (3.10) of the frame operator

$$S_{g,\Lambda} = \mathrm{red}_\Lambda \cdot \sum_{\lambda^\circ \in \Lambda^\circ} \mathcal{V}_g g(\lambda^\circ)\pi(\lambda^\circ) = \mathrm{red}_\Lambda \cdot \left(\mathrm{Id} - \sum_{\lambda^\circ \neq 0} \mathcal{V}_g g(\lambda^\circ)\pi(\lambda^\circ) \right)$$

and rewriting this identity we get

$$\frac{1}{\mathrm{red}_\Lambda} S_{g,\Lambda} - \mathrm{Id} = \sum_{\lambda^\circ \neq 0} \mathcal{V}_g g(\lambda^\circ)\pi(\lambda^\circ). \tag{3.11}$$

We define $H := -\sum_{\lambda^\circ \neq 0} \mathcal{V}_g g(\lambda^\circ)\pi(\lambda^\circ)$. Now we do an estimation of the norm

$$\|H\|_{\mathcal{L}(\boldsymbol{S}_0)} = \|\frac{1}{\mathrm{red}_\Lambda} S_{g,\Lambda} - \mathrm{Id}\|_{\mathcal{L}(\boldsymbol{S}_0)} \leq \sum_{\lambda^\circ \neq 0} |\mathcal{V}_g g(\lambda^\circ)| \tag{3.12}$$

and call $\gamma_1 := \sum_{\lambda^\circ \neq 0} |\mathcal{V}_g g(\lambda^\circ)|$. By definition of H and some reformulation of (3.11) we get

$$\mathrm{red}_\Lambda \, S_{g,\Lambda}^{-1} = (\mathrm{Id} - H)^{-1} = \sum_{k=0}^{\infty} H^k$$

if the Neumann series $\sum_{k=0}^{\infty} H^k$ converges in the operator norm. For the operatornorm of the inverse frame operator this means

$$\|\mathrm{red}_\Lambda \, S_{g,\Lambda}^{-1}\|_{\mathcal{L}(\boldsymbol{S}_0)} = \|\sum_{k=0}^{\infty} H^k\|_{\mathcal{L}(\boldsymbol{S}_0)} \leq \sum_{k=0}^{\infty} \|H\|_{\mathcal{L}(\boldsymbol{S}_0)}^k \leq \sum_{k=0}^{\infty} \gamma_1^k.$$

The last sum converges if and only if $\gamma_1 < 1$ (γ_1 is positiv and real!). We can achieve this, as for sufficiently fine lattices, the redundancy red_Λ is large and the redundancy of the adjoit lattice $\mathrm{red}_{\Lambda^\circ}$ is low and therefore $\sum_{\lambda^\circ \neq 0} |\mathcal{V}_g g(\lambda^\circ)| < 1$. This implies invertibility of $S_{g,\Lambda}$ on $\boldsymbol{S}_0(\mathbb{R}^d)$ and $\widetilde{g} = S_{g,\Lambda}^{-1} g \in \boldsymbol{S}_0(\mathbb{R}^d)$.

The frame property follows from [FZ98, Thm. 3.6.4] □

Note that the condition on the lattice, i.e. Λ is sufficiently fine, is meant in the following way. The points of the lattice Λ are nicely spread over the TF-plane, i.e.

$$\forall g \; \exists \delta = \delta(g) : \Lambda \cap B_\delta(0) \text{ generates } \Lambda$$

and

$$\bigcup_{\lambda} B_\delta(\lambda) = \mathbb{R}^d \times \widehat{\mathbb{R}}^d.$$

Corollary 3.2.1. *Let $g \in \boldsymbol{S}_0(\mathbb{R}^d)$ be a Gabor window and $(\alpha_1, \dots, \alpha_{2d})$ be sufficiently small. Consider a sequence $\Lambda_n = A_n\mathbb{Z}^{2d}$ with singular values $\sigma_i^n, i = 1, \dots, 2d$, of the corresponding A_n in $[0, \alpha_1) \times \cdots \times [0, \alpha_{2d})$ and $\max_{i\in\{1,\dots,2d\}}(\sigma_i^n) \to 0$ for $n \to \infty$. Then*

$$\operatorname{red}_{\Lambda_n} \widetilde{g}_{\Lambda_n} \to g$$

in $\boldsymbol{S}_0(\mathbb{R}^d)$. Where $\widetilde{g}_{\Lambda_n}$ is the canonical dual of g with respect to the lattice Λ_n and $\operatorname{red}_{\Lambda_n} = \frac{1}{|\sigma_1^n \cdots \sigma_{2d}^n|}$.

Proof. In the proof of Theorem 3.2.1, Equation (3.12) we saw that

$$\|\frac{1}{\operatorname{red}_\Lambda} S_{g,\Lambda} - \mathrm{Id}\|_{\mathcal{L}(\boldsymbol{S}_0)} \leq \sum_{\lambda^\circ \neq 0} |\mathcal{V}_g g(\lambda^\circ)|.$$

If $\max_{i\in\{1,\dots,2d\}}(\sigma_i^n) \to 0$, the singular values of the adjoint lattice go to ∞. This means that $\operatorname{red}_{\Lambda_n^\circ} \to 0$ and the adjoint lattice without the 0 has less and less points. Therefore the right side in (3.12) goes to 0. We get

$$\frac{1}{\operatorname{red}_{\Lambda_n}} S_{g,\Lambda_n} \to \mathrm{Id}$$

in the operator norm and this implies

$$\operatorname{red}_{\Lambda_n} S_{g,\Lambda_n}^{-1} \to \mathrm{Id}$$

in the operator norm, and thus

$$\operatorname{red}_{\Lambda_n} \widetilde{g}_{\Lambda_n} = \operatorname{red}_{\Lambda_n} S_{g,\Lambda_n}^{-1} g \to g$$

in the $\boldsymbol{S}_0$-norm. □

3.3. Orthogonal Projection onto Spline-Type Spaces

For some operators which are in their exact form quite complex, it might be good to approximate them by more simple ones. This could be due to computational realization, e.g. for realtime computations used in mobile commumication. When approximating an operator of some kind, one can find a subspace of the space it belongs to and do an orthogonal projection onto it. Here the subspace should be some space with properties that suit our problem better, but still yields only an acceptable small error to the original operator. Orthogonal projection gives in a natural way the minimal difference in the 2-norm. This means that it is the least-squares difference from the original operator and its approximation in the subspace.

Now for best approximation in a special subspace spanned by a *shifted* atom g, we need a solution of coefficiants c_λ to the following problem.

$$\|f - \sum_{\lambda\in\Lambda} c_\lambda \, \mathrm{T}_\lambda \, g\|_{\boldsymbol{L}^2(\mathbb{R}^d)} = min$$

A solution to the problem is the orthogonal projection $c_\lambda = P_{g,\Lambda} f$ onto the space spaned by $\{\mathrm{T}_\lambda \, g\}_{\lambda\in\Lambda}$. The only problem is, when we do not have linear independence of the family $\{\mathrm{T}_\lambda \, g\}_{\lambda\in\Lambda}$, because then the coefficients are not unique anymore. For that purpose we want it to be a Riesz basis as they then are linearly independent.

Definition 3.3.1 (Spline-type Space). *Let Λ be a discrete subgroup (lattice) of $\mathbb{R}^d$ and g an element in some translation-invariant Banach space, e.g. $\left(\boldsymbol{S}_0(\mathbb{R}^d), \|\cdot\|_{\boldsymbol{S}_0}\right)$. We denote the subspace $\overline{span}\{\mathrm{T}_\lambda\, g\}_{\lambda\in\Lambda}$ by $\mathbf{V}_{g,\Lambda}$ and call it a* spline-type space *if the family $\{\mathrm{T}_\lambda\, g\}_{\lambda\in\Lambda}$ is a Riesz basis for $\mathbf{V}_{g,\Lambda}$.*

So to say a spline-type space is not a space of splines, but the closure of all finite linear combinations of a fixed atom g, shifted over the lattice Λ. As this family is a Riesz basis, there is a biorthogonal (dual) family $\widetilde{\{\mathrm{T}_\lambda\, g\}}_{\lambda\in\Lambda}$ and one can show that it is of the form $\{\mathrm{T}_\lambda\, \widetilde{g}\}_{\lambda\in\Lambda}$ for some dual atom $\widetilde{g}$, [Fei02, Lemma 3.5].

Theorem 3.3.1 (Spline-type spaces with $\boldsymbol{S}_0(\mathbb{R}^d)$-atom). *Let $g \in \boldsymbol{S}_0(\mathbb{R}^d)$ be given, and let $\Lambda \lhd \mathbb{R}^d$ be a lattice.*

1. *The family $\{\mathrm{T}_\lambda\, g\}_{\lambda\in\Lambda}$ is a Riesz basis (for its closed linear span $\mathbf{V}_{g,\Lambda}$) if and only if the $\Lambda^\perp$-periodized version of $\hat{g}$, i.e., $H := \sum_{\lambda^\perp\in\Lambda^\perp} |\,\mathrm{T}_{\lambda^\perp}(\hat{g})|^2$, is free of zeros.*

2. *In this case the Fourier transform of the function $\widetilde{g}$ generating the biorthogonal Riesz basis $\{\mathrm{T}_\lambda\, \widetilde{g}\}_{\lambda\in\Lambda}$ is given by $\widehat{\widetilde{g}} = \hat{g}/H$, and $\widetilde{g}$ belongs to $\boldsymbol{S}_0(\mathbb{R}^d)$.*

Proof. See [Fei02, Theorem 3.6]. □

As $\boldsymbol{L}^2(\mathbb{R}^d)$ is a Hilbert space, orthogonal projection makes sense, and in this manner, the orthogonal projection from $\boldsymbol{L}^2(\mathbb{R}^d)$ to $\mathbf{V}_{g,\Lambda}$, for $(\mathbf{g}, \Lambda)$ as above, can be described as

$$f \mapsto P_{g,\Lambda}(f) = \sum_{\lambda\in\Lambda} \langle f, \mathrm{T}_\lambda\, g\rangle\, \mathrm{T}_\lambda\, \widetilde{g} = \sum_{\lambda\in\Lambda} \langle f, \mathrm{T}_\lambda\, \widetilde{g}\rangle\, \mathrm{T}_\lambda\, g \tag{3.13}$$

where the dual is obtained by

$$\widetilde{g} = \mathcal{F}^{-1}\left(\frac{1}{H}\hat{g}\right) \tag{3.14}$$

with $H(s) = \sum_{\lambda^\perp\in\Lambda^\perp} |\hat{g}(s-\lambda^\perp)|^2$. This means, that the coefficients for the best approximation are given by

$$c_\lambda = \langle f, \mathrm{T}_\lambda\, \widetilde{g}\rangle. \tag{3.15}$$

Orthogonal projection onto spline-type spaces will be needed later in Chapter 4 and is used for the calculations done with MATLAB™.

3.4. Finite Discrete Gabor Analysis

This section deals with realization of calculations in Gabor analysis with a computer. A computer can only calculate with finite and discrete objects, oposed to continuous signals or even infinite discretizations. This means we need a finite theory, such that it represents our results. This is well known for many applications in mathmatics, so only an outlook will be given here.

3.4.1. Finite Signals

With the constraint, that we can only have finite objects, a singal, e.g. a $\boldsymbol{L}^2$-function, is realized as a vector $f \in \mathbb{C}^L$. This means, we look at the function f as a discretized version cut to finite length L. We write $f = (f(0), \ldots, f(L-1))$ with $f : [0, \ldots, L-1] \to \mathbb{C}^L$. With that we can again define the operator for modulation as

$$\mathrm{M}_l\, f(j) := e^{2\pi i j l/L} f(j) \tag{3.16}$$

but for the translation it is not as easy, because f might not be defined for $f(j-k)$. However, the natural solution is to periodize f, such that

$$f(j + nL) := f(j) \quad \forall n \in \mathbb{Z}, j \in [0, \ldots, L-1].$$

With this we now have functions defined on the finite Abelian group $\mathbb{Z}_L := \mathbb{Z}/L\mathbb{Z}$ and therefore $f \in \boldsymbol{\ell}^2(\mathbb{Z}_L) \cong \mathbb{C}^L$. With the periodization we can define translation for finite signals.

$$\mathrm{T}_k\, f(j) := f(j-k) \tag{3.17}$$

The realization of these two operations as matrices are easy to see. The modulation matrix has the exponential entries in the main diagonal

$$\mathrm{M}_l = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & e^{2\pi i 1 l/L} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{2\pi i (L-1) l/L} \end{pmatrix} \tag{3.18}$$

and the translation has ones in the k-th (periodized) subdiagonal.

$$\mathrm{T}_k = \begin{pmatrix} 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \\ 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \end{pmatrix}. \tag{3.19}$$

Here the one in the first row is on the k-th position. An straight foreward result is again that they do not commute.

$$\mathrm{T}_k\, \mathrm{M}_l = e^{2\pi i k l/L}\, \mathrm{M}_l\, \mathrm{T}_k \quad k, l \in \mathbb{Z}_L.$$

For a TF-shift we write

$$\pi(\lambda) := \pi(k,l) := \mathrm{M}_l\, \mathrm{T}_k \quad \text{with} \quad \lambda = (k,l) \in \mathbb{Z}_L \times \widehat{\mathbb{Z}_L}$$

which is, e.g. for $L = 6$

$$\pi(3,2) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{2\pi i 1/3} & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{2\pi i 2/3} & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{2\pi i 3/3} \\ e^{2\pi i 4/3} & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{2\pi i 5/3} & 0 & 0 & 0 & 0 \end{pmatrix}.$$

3.4.2. Gabor Frames in $\mathbb{C}^L$

The results we have given for continuous Gabor analysis can be directly adapted. But we can reformulate some of them in terms of linear algebra, as this is our setting now. For a finite set of vectors $\{g_i\}_{i=0,\dots,n-1}$ in $\mathbb{C}^L$ to be a frame for $\mathbb{C}^L$ it suffices that there exist $A, B > 0$ with

$$A\sum_{j=0}^{L-1}|f(j)|^2 \leq \sum_{i=0}^{n-1}|\langle f, g_i\rangle|^2 \leq B\sum_{j=0}^{L-1}|f(j)|^2 \quad \forall f \in \mathbb{C}^L. \tag{3.20}$$

By writing

$$C = \begin{pmatrix} g_0^* \\ \vdots \\ g_{n-1}^* \end{pmatrix}$$

with $g^* = \overline{g}^T$ we get

$$A\|f\|_2^2 \leq \|Cf\|_2^2 \leq B\|f\|_2^2 \quad \forall f \in \mathbb{C}^L. \tag{3.21}$$

C is the analysis operator of the set $\{g_i\}_{i=0,\dots,n-1}$ and has size $n \times L$. This means that $L \leq n$ is necessary for $\{g_i\}_{i=0,\dots,n-1}$ to be a frame, as otherwise one can find $f \neq 0$ with $Cf = 0$, which stands in contradiction with (3.21). We actually get the equivalance of C having full rank ($rk(C) = L$) for $\{g_i\}_{i=0,\dots,n-1}$ to be a frame.

The frame operator $S = C^*C$ is then an invertible $L \times L$-matrix, with C as the analysis operator and its adjoint operator $C^* = D$ being the synthesis operator. As C is an operator with $C : \mathbb{C}^L \to \mathbb{C}^n$, its adjoint is a mapping $D : \mathbb{C}^n \to \mathbb{C}^L$. Which means that we do not have a unique solution to the equation

$$f = Dc$$

for the coefficients c as $L \leq n$. By writing $f = SS^{-1}f = C^*C(C^*C)^{-1}f$ we see that one solution is given by

$$c = C(C^*C)^{-1}f = (C^*)^\dagger f = D^\dagger f$$

the pseudoinverse of D, by the well known formula for an operator with linear independent rows $D^\dagger = D^*(DD^*)^{-1}$.

Theorem 3.4.1. *Let $\{g_i\}_{i=0,\dots,n-1}$ be a frame for $\mathbb{C}^L$, its frame operator S and synthesis operator D. Than it holds that*

$$D^\dagger f = \{\langle f, S^{-1}g_i\rangle\}_{i=0,\dots,n-1}, \quad \forall f \in \mathbb{C}^L$$

Proof. Let $f \in \mathbb{C}^L$, then $Dc = f$ can be expressed as $f = \sum_{i=0}^{n-1} c_i g_i$ and the solution with the minimal 2-norm is given by $c_i = \langle f, S^{-1}g_i\rangle$ for $i = 0, \dots, n-1$. □

Now $D^\dagger$ is equal to $D^*(DD^*)^{-1} = C(DC)^{-1}$. As D is a surjective $L \times n$-matrix, because it has full row rank, its singular value decomposition is

$$D = U\,(\Sigma\ 0)\,V^*$$

with $(\Sigma\ 0)$ being $L \times n$ just as D, U is $L \times L$ and V is $n \times n$. For an arbitray $(n-L) \times L$-matrix F, we get

$$DV \begin{pmatrix} \Sigma^{-1} \\ F \end{pmatrix} U^* f = U\,(\Sigma\ 0)\, V^* V \begin{pmatrix} \Sigma^{-1} \\ F \end{pmatrix} U^* f = UU^* f = f.$$

and therefore

$$V \begin{pmatrix} \Sigma^{\dagger} \\ F \end{pmatrix} U^*$$

is usually called a *generalized inverse* for D.

Just as in the continuous setting, a Gabor frame is a frame that is given by TF-shifted versions of a window function $g \in \mathbb{C}^L$ over $\Lambda \subseteq \mathbb{Z}_L \times \widehat{\mathbb{Z}_L}$, i.e. $\{g_\lambda\}_{\lambda\in\Lambda} = \{\pi(\lambda)g\}_{\lambda\in\Lambda}$. If Λ is a lattice and therefore a subgroup of $\mathbb{Z}_L \times \widehat{\mathbb{Z}_L}$, we call it a regular Gabor frame, otherwise an irregular Gabor frame. Alltough in this paper we only consider regular ones.

In the finite setting the redundancy of the lattice Λ is given by

$$\mathrm{red}_\Lambda := \frac{n}{L}$$

which tells us how many more elements than needed are given to span all of $\mathbb{C}^L$.

3.4.3. Discrete Fourier Transform

As a lot of the algorithms needed in signal processing use the Fourier transform in some way. We are going to take a closer look at the Fourier transform in the finite discrete setting. There are two reasons the discrete Fourier transform (DFT) is such an important tool.

1. The DFT can be calculated precisely up to numerical round-off errors and it can be used to approximate the continuous version.

2. There is a very fast implementation for it, called the fast Fourier transform (FFT).

Let $f \in \mathbb{C}^L$ be a finite signal as $f := (f(0), \ldots, f(L-1))^T$, i.e. a column vector. And we put

$$v_k := \frac{1}{\sqrt{L}} \begin{pmatrix} 1 \\ e^{2\pi i k \frac{1}{L}} \\ \vdots \\ e^{2\pi i k \frac{L-1}{L}} \end{pmatrix} \in \mathbb{C}^L,$$

then the discrete Fourier transform can be defined as.

Definition 3.4.1 (Discrete Fourier Transform)**.** *For $f \in \mathbb{C}^L$ and $k \in \{0, \ldots, L-1\}$, the* discrete Fourier transform *(DFT) of length L is defined as*

$$\mathcal{F}_L(f)(k) := \langle f, v_k \rangle = \frac{1}{\sqrt{L}} \sum_{j=0}^{L-1} f(j) e^{-2\pi i k \frac{j}{L}}. \tag{3.22}$$

$\mathcal{F}_L : \mathbb{C}^L \to \mathbb{C}^L$ is a unitary mapping with inverse

$$\mathcal{F}_L^{-1}(\hat{f})(j) = \sum_{k=0}^{L-1} \hat{f}(k) v_k = \frac{1}{\sqrt{L}} \sum_{k=0}^{L-1} \hat{f}(k) e^{2\pi i k \frac{j}{L}}. \tag{3.23}$$

In this definition we have the normalization factor of $\frac{1}{\sqrt{L}}$ which is needed for it to be unitary. There are different ways to define it, e.g. the DFT with no prefactor and therefore the inverse with $\frac{1}{L}$. In MATLAB™, like in most computation programs, it is implemented with no prefactor on both, the DFT and the inverse. This means on has to input this normalization factor manually to have a unitary mapping. Usually one calls up the DFT with the command `fxx = fft(xx)/sqrt(L)` and its inverse in the same manner.

There is also a way to implement the DFT as a matrix vector product. For this we put $\omega_L := e^{2\pi i 1 L}$. Then ω_L is the primitive L-th root of unity and we have $v_k(j) = \frac{1}{L}\omega_L^{kj}$ for all $k, j \in \{0, \dots, L-1\}$. Then we can put

$$\mathbf{F}_L := (v_0| \cdots |v_{L-1}) = \frac{1}{\sqrt{L}} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega_L^1 & \cdots & \omega_L^{L-1} \\ 1 & \omega_L^2 & \cdots & \omega_L^{2(L-1)} \\ \vdots & \vdots & & \vdots \\ 1 & \omega_L^{L-1} & \cdots & \omega_L^{(L-1)(L-1)} \end{pmatrix}.$$

This Vandermonde matrix is unitary and it holds

$$\mathcal{F}_L(f) = \begin{pmatrix} \langle f, v_0 \rangle \\ \langle f, v_1 \rangle \\ \vdots \\ \langle f, v_{L-1} \rangle \end{pmatrix} = \begin{pmatrix} v_0^* \cdot f \\ v_1^* \cdot f \\ \vdots \\ v_{L-1}^* \cdot f \end{pmatrix} = \begin{pmatrix} v_0^* \\ v_1^* \\ \vdots \\ v_{L-1}^* \end{pmatrix} \cdot f = \mathbf{F}_L^* \cdot f = \mathbf{F}_L^{-1} \cdot f.$$

And in the same way one can find that $\mathcal{F}_L^{-1}(\hat{f}) = \mathbf{F}_L \cdot \hat{f}$.

Remark: The DFT as defined in (3.22) can be directly calculated in $\mathcal{O}(L^2)$ computer operations. The FFT, which is based on splitting the transform in terms of devisors of L and other refinements, can be done in $\mathcal{O}(L \log_2 L)$ computer operations, if L is rich of divisors. In fact, if L is rich in divisors, then there are many lattices Λ in $\mathcal{G} \times \widehat{\mathcal{G}}$. Therefore it is always good to choose such a L for calculations.

Chapter 4.

Gabor Multipliers

4.1. Gabor Multipliers as Filters

In many applications there is need to process a signal in some kind of way. As an example one can look at mobile communication. When person A calls person B from a mobile phone, while travelling on a train, the signal B receives is modified by various factors. Among them are the noise from the train or other travellers, as well as signal modifications resulting from the Doppler effect of objects in relative motion to each other or other environment related factors. When B wants to get a signal with essentially only the relevant parts, such as the voice of A, he needs to process the signal in a way to filter the relevant information.

When processing a signal, it is quite popular to use so-called *time-invariant filters*. These are filters used to modify a signal in the frequency domain (via Fourier transform) to get a filtered signal. It gets more and more popular to do filtering with *time-variant filters*. These are similar, with the main difference being that the change depends on the point of time in the signal. This is helpfull if for example the speed of the train is changing, because then one needs a different filter.

Another application of time-variant filters would be encoding MP3. Here the storage size of a musical signal is reduced by filtering parts that are irrelevant for the perception of the human ear. These filters are adapted to research in psychoacoustics in a way that minimizes the loss of quality of the musical signal.But this will not be the point of investigation in this work.

The kind of time-variant filters we take a look at are using the sampled version of the short-time Fourier transform over a lattice as it gives a time-frequency representation of a signal. In this manner, we do the analysis of a signal using the short-time Fourier transform and get a time-frequency representation. Before resynthesizing we multiply the coefficiants with a general function in two variables, e.g. a function which is 1 in a certain area and 0 elsewhere to filter certain frequencies at certain points of time.

In terms of mathematical notation, processing with a time-variant filter of that kind

would look like the following.

$$\begin{aligned} f & \qquad \text{(signal)} \\ \downarrow & \\ \mathcal{V}_{g_1} f(\lambda) & \qquad \text{(analysis)} \\ \downarrow & \\ m(\lambda) \cdot \mathcal{V}_{g_1} f(\lambda) & \qquad \text{(filtering)} \\ \downarrow & \\ \sum_{\lambda \in \Lambda} [m(\lambda) \cdot \mathcal{V}_{g_1} f(\lambda)] \pi(\lambda) g_2 & \qquad \text{(synthesis)} \end{aligned}$$

With this in mind we go on to the formal definition of Gabor multipliers.

Definition 4.1.1 (Gabor Multipliers)**.** *Let $g_1, g_2 \in \boldsymbol{L}^2(\mathbb{R}^d)$ be two windows, Λ a time-frequency lattice for $\mathbb{R}^d$, i.e. a discrete subgroup of the phase space $\Lambda \lhd \mathbb{R}^d \times \widehat{\mathbb{R}}^d$. Let $\mathbf{m} = (m_\lambda)_{\lambda \in \Lambda}$ be a complex valued sequence on Λ. The* Gabor multiplier *associated to the triple (g_1, g_2, Λ) with* upper symbol *(multiplier) $\mathbf{m}$ is defined by*

$$G_{\mathbf{m}}(f) := G_{g_1, g_2, \Lambda, \mathbf{m}}(f) = \sum_{\lambda \in \Lambda} [m_\lambda \cdot \langle f, \pi(\lambda) g_1 \rangle] \pi(\lambda) g_2. \tag{4.1}$$

With this definition we see that if we want to remove a certain frequency band $[\omega_1, \omega_2]$ during the time interval $[t_1, t_2]$, we just put $m_\lambda = 0$ for $\lambda = (x, \omega) \in [t_1, t_2] \times [\omega_1, \omega_2] \cap \Lambda$.

It is also easy to see that a Gabor multiplier is an infinite linear combination of rank-one operators $f \mapsto \langle f, \pi(\lambda) g_1 \rangle \pi(\lambda) g_2$ with coefficiants m_λ. We will denote these rank-one operators by P_λ.

$$P_\lambda(f) := \langle f, \pi(\lambda) g_1 \rangle \pi(\lambda) g_2 = [(\pi(\lambda) g_1) \otimes (\pi(\lambda) g_2)^*] f$$

In this notations a Gabor multiplier takes the form

$$G_{g_1, g_2, \Lambda, \mathbf{m}} = \sum_{\lambda \in \Lambda} m_\lambda P_\lambda \tag{4.2}$$

and for $P_0 = g_1 \otimes g_2^*$ we get $P_\lambda = (\pi \otimes \pi^*)(\lambda) P_0$.

In Figure 4.1 we have 3 different upper symbols $M1, M2, M3$ and their corresponding Gabor multipliers beneath. The lattice used is separable with lattice parameters $a = 20, b = 16$ and as a window we used a Gaussian. $M1$ is a filter that is circular in the time frequency domain with radius 100. $M2$ removes all frequencies except 10 Hz to 30 Hz over the whole length of the signal. And $M3$ is just a random smooth symbol. In Figures 4.2 and 4.3 we see the (random band-limited) signal of length $n = 480$ and its modified versions for each Gabor multiplier, for the time domain and its STFT's.

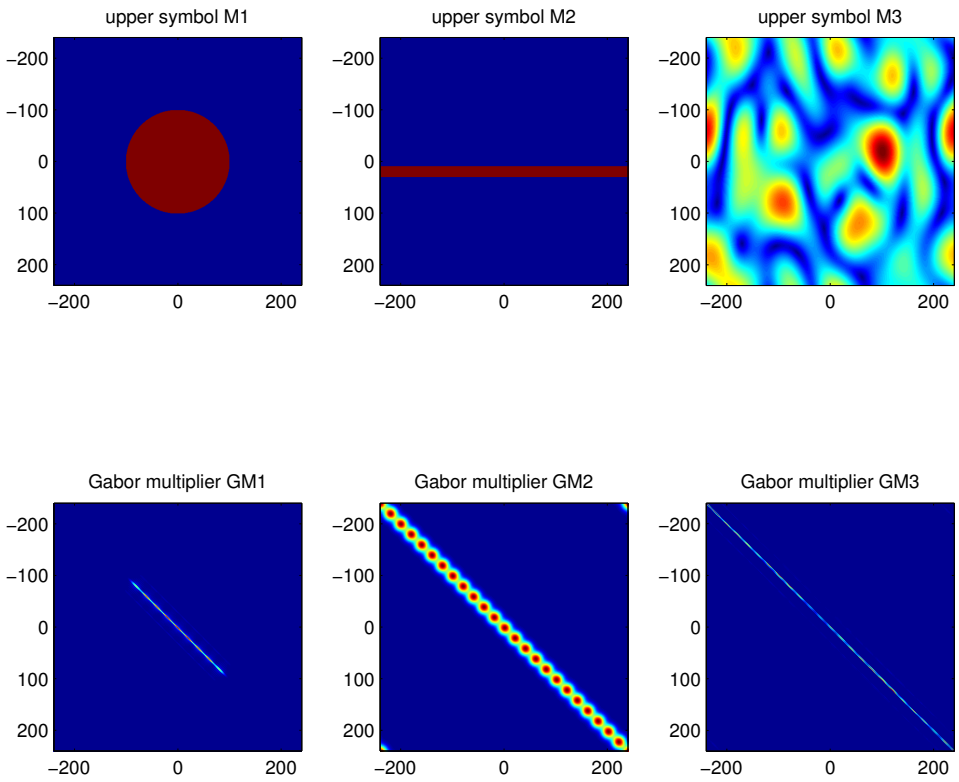


Figure 4.1.: 3 upper symbols with corresponding Gabor multipliers

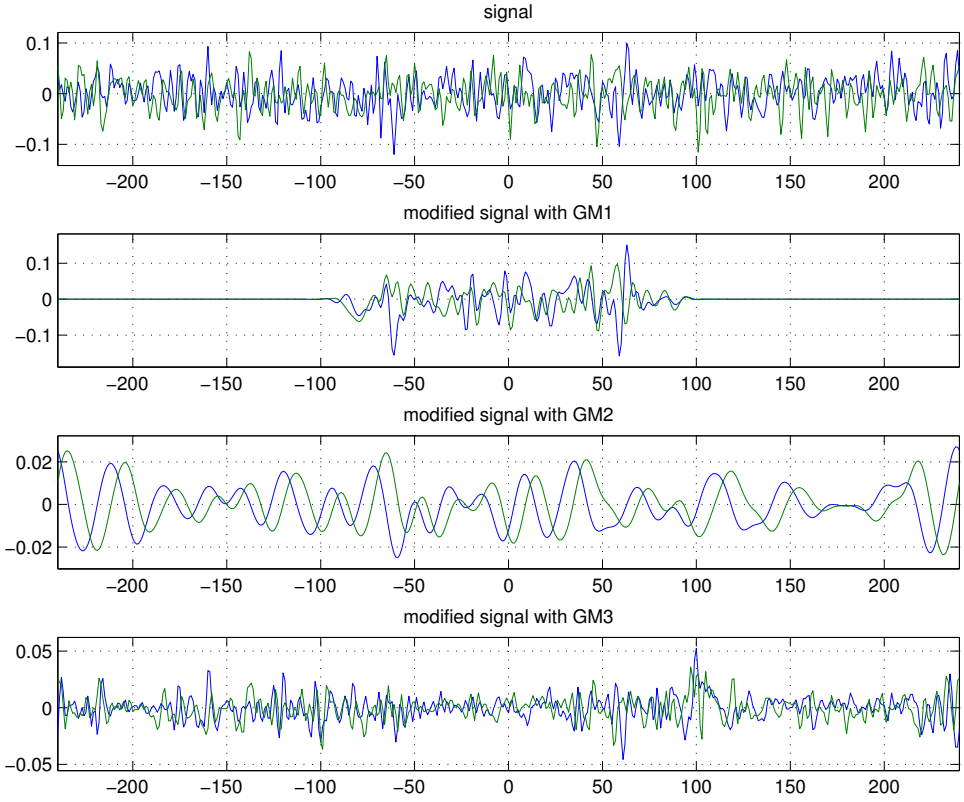


Figure 4.2.: A band-limited signal and the versions modified with the Gabor multipliers

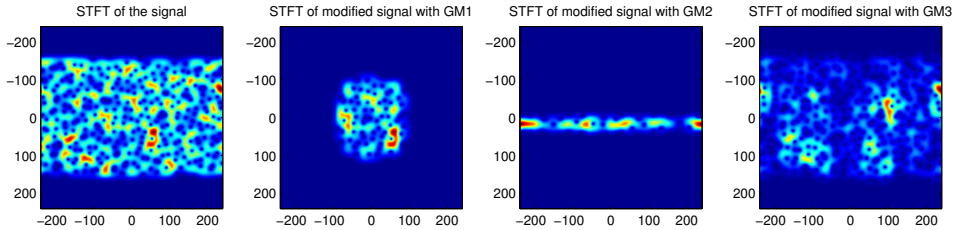


Figure 4.3.: STFT's of the signal and the modified versions

There have been surveys about Gabor multipliers that have shown various properties, like the next theorem as well as the continuous dependence of a Gabor multiplier on its boulding blocks $\mathbf{m}, g_1, g_2$ and in a way also on the lattice Λ. For more on the continuity see [Fei03].

Theorem 4.1.1. *Let the windows g_1, g_2 be in $\boldsymbol{S}_0(\mathbb{R}^d)$ and Λ be an arbitrary TF-lattice. Then the mapping from $(m_\lambda)_{\lambda\in\Lambda}$ to the associated Gabor multiplier $G_{g_1,g_2,\Lambda,\mathbf{m}}$ maps the Banach Gelfand triple $(\boldsymbol{\ell}^1, \boldsymbol{\ell}^2, \boldsymbol{\ell}^\infty)(\Lambda)$ into the bounded linear operators with kernel in $(\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')(\mathbb{R}^d \times \widehat{\mathbb{R}}^d)$, i.e., into $(\mathcal{B}, \mathcal{HS}, \mathcal{B}')$.*

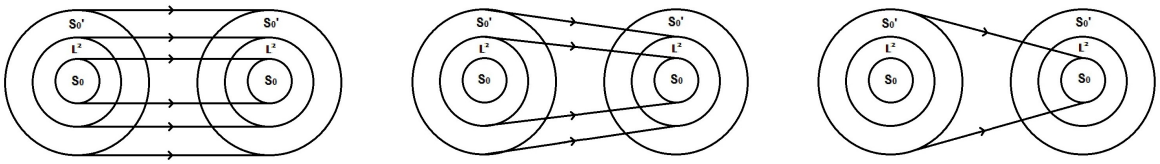
(i) If $\mathbf{m} \in \boldsymbol{\ell}^\infty(\Lambda)$ then $G_\mathbf{m}$ is a bounded operator on $(\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')(\mathbb{R}^d)$ with $|||G_\mathbf{m}||| \leq C\|\mathbf{m}\|_\infty$.

(ii) If $\mathbf{m} \in \boldsymbol{\ell}^2(\Lambda)$ then $G_\mathbf{m} : \boldsymbol{S}_0'(\mathbb{R}^d) \to \boldsymbol{L}^2(\mathbb{R}^d)$ and $G_\mathbf{m} : \boldsymbol{L}^2(\mathbb{R}^d) \to \boldsymbol{S}_0(\mathbb{R}^d)$ and $G_\mathbf{m} \in \mathcal{HS}$.

(iii) If $\mathbf{m} \in \boldsymbol{\ell}^1(\Lambda)$ then $G_\mathbf{m} : \boldsymbol{S}_0'(\mathbb{R}^d) \to \boldsymbol{S}_0(\mathbb{R}^d)$ and $G_\mathbf{m} \in \mathcal{B}$.

Proof. Can be found in [Sch04, Chapter 2]. □

These pictures fit to Theorem 4.1.1. The Banach Gelfand triple $(\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')$ is here viewed as circles included in each other. This means that the inner circle represents $\boldsymbol{S}_0$, the second circle going out is $\boldsymbol{L}^2$ and the outer circle is $\boldsymbol{S}_0'$. The arrows in the pictures are the mappings of the Gabor multipliers in Theorem 4.1.1.



(a) $G_\mathbf{m}$ with $\mathbf{m} \in \boldsymbol{\ell}^\infty(\Lambda)$ (b) $G_\mathbf{m}$ with $\mathbf{m} \in \boldsymbol{\ell}^2(\Lambda)$ (c) $G_\mathbf{m}$ with $\mathbf{m} \in \boldsymbol{\ell}^1(\Lambda)$

Figure 4.4.: Visualization of Theorem 4.1.1

Now we can define the space of Gabor multipliers as

$$\mathcal{GM}_p := \left\{ \sum_{\lambda\in\Lambda} m_\lambda P_\lambda, \text{ with } m_\lambda \in \boldsymbol{\ell}^p(\Lambda) \right\} \quad \text{for } p \in [1, \infty] \tag{4.3}$$

and the preceding theorem shows that for the Gelfand triple $(\mathcal{GM}_1, \mathcal{GM}_2, \mathcal{GM}_\infty)$ we have $(\mathcal{GM}_1, \mathcal{GM}_2, \mathcal{GM}_\infty) \subseteq (\mathcal{B}, \mathcal{HS}, \mathcal{B}')$.

4.2. Approximation by Gabor Multipliers

The aim of this paper is not to investigate applications for Gabor multipliers, but rather how to approximate other operators by them. This is desireable for simple computational reasons, as there are fast ways to evaluate discretized Gabor multipliers because of their structure. We will see later that exact representation by Gabor multipliers is a goal that is quite restrictive. Therefore we approximate with Gabor multipliers and try to get the minimal error possible, i.e. find c_λ such that the error

$$\varepsilon(c) = \|T_{app} - T\|_{\mathcal{HS}}$$

is minimal for

$$T_{app} = \sum_{\lambda\in\Lambda} c_\lambda P_\lambda.$$

We come to one important result for approximation by Gabor mutipliers which can be found in [FN03].

Theorem 4.2.1. *Suppose that $(\mathbf{g}, \Lambda)$ generates a $\boldsymbol{S}_0$-Gabor frame for $\boldsymbol{L}^2(\mathbb{R}^d)$, with $\|g\|_2 = 1$, and write P_λ for the orthogonal projection $f \mapsto \langle f, \pi(\lambda)g\rangle\pi(\lambda)g$.*

(i) The family $(P_\lambda)_{\lambda\in\Lambda}$ is a Riesz basis for its closed linear span within the Hilbert space $\mathcal{HS}$ of all Hilbert-Schmidt operators on $\boldsymbol{L}^2(\mathbb{R}^d)$ if and only if the function $H(s)$, defined as the Λ-Fourier transform of $(|\mathcal{V}_g g(\lambda)|^2)_{\lambda\in\Lambda}$ is bounded away from zero.

(ii) An operator T belongs to the closed linear span of this Riesz basis if and only if it belongs to $\mathcal{GM}_2$, the space of Gabor multipliers with symbol in $\boldsymbol{\ell}^2(\Lambda)$.

(iii) The canonical biorthogonal family to $(P_\lambda)_{\lambda\in\Lambda}$ is of the form $(Q_\lambda)_{\lambda\in\Lambda}$,

$$Q_\lambda = (\pi\otimes\pi^*)(\lambda)Q_0 \text{ for } \lambda\in\Lambda,$$

for a uniquely determined Gabor multiplier $Q_0 \in \boldsymbol{B}$

(iv) The best approximation of $T \in \mathcal{HS}$ by Gabor multpliers based on the pair $(\mathbf{g}, \Lambda)$ is of the form

$$PG(T) := \sum_{\lambda\in\Lambda} \langle T, Q_\lambda\rangle_{\mathcal{HS}} P_\lambda, \tag{4.4}$$

thus PG describes the orthogonal projection from $\mathcal{HS}$ onto $\mathcal{GM}_2$.

Proof. For a proof one can look up [Dör02, Section 5.4.1] as in [FN03] only a sketch is given. □

Remark: The orthogonal projection in 4.2.1 (iv) can actually be extended to a bounded mapping from $(\mathcal{B}, \mathcal{HS}, \mathcal{B}')$ into $(\mathcal{GM}_1, \mathcal{GM}_2, \mathcal{GM}_\infty)$, if one takes "orthogonal projection" liberally. In this case the inner product in (4.4) is replaced by the Gelfand bracket. We see that the mapping $\mathbf{m} \mapsto G_\mathbf{m}$ from upper symbol in $(\boldsymbol{\ell}^1, \boldsymbol{\ell}^2, \boldsymbol{\ell}^\infty)(\Lambda)$ to Gabor multipliers $(\mathcal{GM}_1, \mathcal{GM}_2, \mathcal{GM}_\infty)$ is invertibel and the inverse mapping is given by

$$m_\lambda = \langle G_\mathbf{m}, Q_\lambda\rangle.$$

Now we can define the upper symbol for more general operators than only Gabor multipliers. The upper symbol for an operator $T \in (\mathcal{B}, \mathcal{HS}, \mathcal{B}')$ is

$$\sigma_U(T) := \langle T, Q_\lambda \rangle.$$

As the families $(P_\lambda)_{\lambda\in\Lambda}$ and $(Q_\lambda)_{\lambda\in\Lambda}$ are biorthogonal families (if $(P_\lambda)_{\lambda\in\Lambda}$ is a Riesz basis), one can write

$$PG(T) = \sum_{\lambda\in\Lambda} \langle T, Q_\lambda \rangle P_\lambda = \sum_{\lambda\in\Lambda} \langle T, P_\lambda \rangle Q_\lambda$$

and the coefficient in the second sum can be rewritten as

$$\langle T, P_\lambda \rangle = \int_{\mathbb{R}^{2d}} \kappa(T)(x,y) g_\lambda(x) \overline{g_\lambda(y)} dxdy = \langle T g_\lambda, g_\lambda \rangle.$$

Now looking at this with operators from $(\mathcal{B}, \mathcal{HS}, \mathcal{B}')$ gives us the following definition.

Definition 4.2.1 (Lower symbol)**.** *The* lower symbol *of an operator $T \in (\mathcal{B}, \mathcal{HS}, \mathcal{B}')$ with respect to the window g and lattice Λ is given by*

$$\sigma_L(T)(\lambda) := \langle T g_\lambda, g_\lambda \rangle. \tag{4.5}$$

Lemma 4.2.1. *The lower symbol $\sigma_L(G_\mathbf{m})$ for a Gabor multiplier $G_\mathbf{m} \in (\mathcal{GM}_1, \mathcal{GM}_2, \mathcal{GM}_\infty)$ is related to the upper symbol $\mathbf{m}$ via a bounded, invertible linear mapping on $(\boldsymbol{\ell}^1, \boldsymbol{\ell}^2, \boldsymbol{\ell}^\infty)(\Lambda)$,*

$$\sigma_L(G_\mathbf{m}) = |\mathcal{V}_g g|^2 *_\Lambda \mathbf{m}.$$

Proof. Let $G_\mathbf{m} = \sum_{\lambda\in\Lambda} m(\lambda) P_\lambda$, then

$$\begin{aligned}
\sigma_L(G_\mathbf{m})(\lambda') &= \langle \sum_{\lambda\in\Lambda} m(\lambda) P_\lambda g_{\lambda'}, g_{\lambda'} \rangle \\
&= \sum_{\lambda\in\Lambda} m(\lambda) \langle P_\lambda g_{\lambda'}, g_{\lambda'} \rangle \\
&= \sum_{\lambda\in\Lambda} m(\lambda) \langle P_\lambda, P_{\lambda'} \rangle \\
&= \sum_{\lambda\in\Lambda} |\mathcal{V}_g g(\lambda' - \lambda)|^2 m(\lambda) \\
&= (|\mathcal{V}_g g|^2 *_\Lambda \mathbf{m})(\lambda').
\end{aligned}$$

Where $\langle P_\lambda, P_{\lambda'} \rangle = |\mathcal{V}_g g(\lambda' - \lambda)|^2$ is shown in (5.5). □

For better understanding of these relationships, one can look at the commutative diagram in Figure 4.5. Now the question arrises, if we can have similar results when $(P_\lambda)_{\lambda\in\Lambda}$ is no Riesz basis, e.g. only a frame. But the next theorem stats that this is not possible.

Theorem 4.2.2. *Let $g \in \boldsymbol{L}^2(\mathbb{R}^d)$ and $\lambda = A\mathbb{Z}^{2d}$ be a lattice. Then $P_\lambda = (\pi \otimes \pi^*)(\lambda) g \otimes g^*$ is eigther a Riesz basis or not a frame for its closed linear span in $\mathcal{HS}$.*

This theorem tells us that if $(P_\lambda)_{\lambda\in\Lambda}$ is a frame for $\mathcal{GM}_2$, then it already is a Riesz basis. So we want by all means that $(P_\lambda)_{\lambda\in\Lambda}$ is a Riesz basis, as only then our calculations can be stable. A proof for this result can be found in [BP06, section 3].

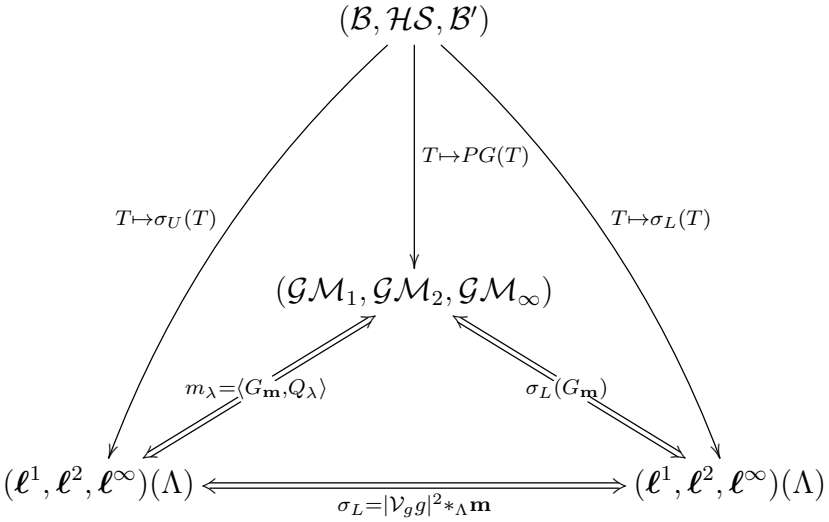


Figure 4.5.: Commutative diagram for the Gelfand triple of Gabor multipliers (middle), the upper symbols (left), the lower symbols (right) and projection from the operator Gelfand triple (top).

4.3. Operator Representations

We are now giving two more representations of operators in $(\mathcal{B}, \mathcal{HS}, \mathcal{B}')$ in the spirit of Theorem 1.5.1. To know where it comes from, we introduce so-called *pseudo-differential operators*.

Definition 4.3.1. *A* pseudo-differential operators T *with symbol* $\mu(T) \in \boldsymbol{S}_0'(\mathbb{R}^d \times \widehat{\mathbb{R}}^d)$ *is defined as*

$$Tf(x) = \int_{\mathbb{R}^d} \mu(T)(x, \omega) \hat{f}(\omega) e^{2\pi i x \cdot \omega} d\omega. \tag{4.6}$$

The function $\mu(T)$ is then called the *Kohn-Nirenberg symbol* of T. These operators are sort of multiplication operators in the frequency domain. They appear as generalizations of differential operators and are used in the field of partial differential equations and have proven usefull in physics, especially quantum field theory. In signal processing, one can interpret the function σ as *time-varying transfer function*.

4.3.1. The Kohn-Nirenberg Symbol

The generalized Kohn-Nirenberg correspondance is a mapping from the distibutional kernel $\kappa(T)$ defined on $\mathbb{R}^{2d}$ to the symbol $\sigma(T)$ defined on $\mathbb{R}^d \times \widehat{\mathbb{R}}^d$. It is given by the following definition.

Definition 4.3.2. *The* Kohn-Nirenberg symbol *(KNS) of an operator* T *with distributional kernel* $\kappa(T)$ *is defined as*

$$\sigma(T)(x, \omega) := \int_{\mathbb{R}^d} \kappa(T)(x, x - y) e^{-2\pi i \omega y} dy. \tag{4.7}$$

It is a function (or distribution) over the TF-plane $\mathbb{R}^d \times \widehat{\mathbb{R}}^d$.

The mapping $\kappa(T) \to \sigma(T)$ is an invertible mapping between kernels in $\boldsymbol{S}_0(\mathbb{R}^{2d})$ and their KNS in $\boldsymbol{S}_0(\mathbb{R}^d \times \widehat{\mathbb{R}}^d)$ and its inversion is given by

$$\kappa(T)(x,y) := \int_{\widehat{\mathbb{R}}^d} \sigma(T)(x,\omega) e^{-2\pi i \omega(x-y)} d\omega. \tag{4.8}$$

For $\mathcal{HS}$-operators (kernel in $\boldsymbol{L}^2(\mathbb{R}^{2d})$) we have the unitary isomorphism

$$\boldsymbol{L}^2(\mathbb{R}^{2d}) \ni \kappa(T) \longleftrightarrow \sigma(T) \in \boldsymbol{L}^2(\mathbb{R}^d \times \widehat{\mathbb{R}}^d)$$

which extends to a unitary Banach-Gelfand triple isomorphism

$$(\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')(\mathbb{R}^{2d}) \ni \kappa(T) \longleftrightarrow \sigma(T) \in (\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')(\mathbb{R}^d \times \widehat{\mathbb{R}}^d).$$

We immediately get the associated Gelfand-bracket identity:

$$\langle \kappa(T), \kappa(S) \rangle_{(\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')(\mathbb{R}^{2d})} = \langle \sigma(T), \sigma(S) \rangle_{(\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')(\mathbb{R}^d \times \widehat{\mathbb{R}}^d)}. \tag{4.9}$$

The reason we need the KNS representation here is the property that it can be used to write the KNS of a time-frequency shifted operator T as a translated KNS of T.

Lemma 4.3.1 (Shift-Covariance)**.** *Let $T \in (\mathcal{B}, \mathcal{HS}, \mathcal{B}')$, then the action of $(\pi \otimes \pi^*)(\lambda)$ on T corresponds to a λ-translation of the Kohn-Nirenberg symbol $\sigma(T)$, i.e.*

$$\sigma((\pi \otimes \pi^*)(\lambda) T) = \mathrm{T}_\lambda \, \sigma(T). \tag{4.10}$$

Proof. Let us first show that the action of $(\pi \otimes \pi^*)(\lambda)$ on T corresponds to a TF-shift $\mathrm{M}_{(\omega,-\omega)} \, \mathrm{T}\,(x,x)$ of the kernel $\kappa(T)$ with $\lambda = (x,\omega)$:

$$\begin{aligned}
((\pi \otimes \pi^*)(\lambda) T) f(t) &= \pi(\lambda) \int_{\mathbb{R}^d} \kappa(T)(s,t) (\pi^*(\lambda) f)(t) dt \\
&= \pi(\lambda) \int_{\mathbb{R}^d} \kappa(T)(s,t) f(t+x) e^{-2\pi i \omega(t+x)} dt \\
&\overset{[r=t+x]}{=} \int_{\mathbb{R}^d} \kappa(T)(s-x, r-x) e^{2\pi i \omega s} e^{-2\pi i \omega r} f(r) dr \\
&= \int_{\mathbb{R}^d} \left[\mathrm{M}_{(\omega,-\omega)} \, \mathrm{T}\,(x,x) \kappa(T) \right](s,r) f(r) dr
\end{aligned}$$

With this we get

$$\begin{aligned}
\sigma((\pi \otimes \pi^*)(\lambda) T)(y,\eta) &= \int_{\mathbb{R}^d} \kappa((\pi \otimes \pi^*)(\lambda) T)(y, y-t) e^{-2\pi i \eta t} dt \\
&= \int_{\mathbb{R}^d} \kappa(T)(y-x, y-t-x) e^{2\pi i \omega((y-x)-(y-t-x))} e^{-2\pi i \eta t} dt \\
&= \int_{\mathbb{R}^d} \kappa(T)((y-x), (y-x)-t) e^{-2\pi i t(\eta-\omega)} dt \\
&= \sigma(T)(y-x, \eta-\omega) = \mathrm{T}_\lambda \, \sigma(T)(y,\eta)
\end{aligned}$$

(cf.[Sch04]) □

Example 4. *For the use with Gabor multipliers, we will directly calculate the KNS of the rank-one operator* $P_0 = g_1 \otimes g_2^*$*:*

$$\begin{aligned}\sigma(P_0)(x,\omega) &= \int_{\mathbb{R}^d} g_1(x)\overline{g_2(x-t)}e^{-2\pi i\omega t}dt \\ &= g_1(x) \int_{\mathbb{R}^d}^{[r=x-t]} \overline{g_2(r)}e^{2\pi i\omega(r-x)}dr \\ &= g_1(x)e^{-2\pi i\omega x}\int_{\mathbb{R}^d} \overline{g_2(r)e^{-2\pi i\omega r}}dr \\ &= e^{-2\pi i\omega x}g_1(x)\overline{\widehat{g_2}(\omega)} = e^{-2\pi i\omega x}[g_1 \otimes \widehat{g_2}^*](x,\omega).\end{aligned}$$

One very nice consequence of the Shift-Covariance Lemma 4.3.1 is that if we look at the KNS of the family $(P_\lambda)_{\lambda\in\Lambda}$, we see that they are of the form

$$\sigma(P_\lambda) = \sigma((\pi\otimes\pi^*)(\lambda)P_0) = \mathrm{T}_\lambda\,\sigma(P_0). \tag{4.11}$$

And by straight foreward calculation one gets

$$\sigma(G_{g_1,g_2,\Lambda,\mathbf{m}}) = \sum_{\lambda\in\Lambda} m_\lambda\ \sigma(P_\lambda) = \sum_{\lambda\in\Lambda} m_\lambda\,\mathrm{T}_\lambda\,\sigma(P_0). \tag{4.12}$$

If Λ is a lattice and $(P_\lambda)_{\lambda\in\Lambda}$ forms a Riesz basis for its closed linear span, then

$$\mathbf{V}_{\sigma(P_0),\Lambda} = \overline{span}\{\mathrm{T}_\lambda\,\sigma(P_0), \lambda\in\Lambda\} \tag{4.13}$$

is a spline-type space within the space $\{\sigma(T), T\in\mathcal{HS}\}$, because as shown in Example 4, $\sigma(P_0) = e^{-2\pi i\omega x}[g_1\otimes\widehat{g_2}^*](x,\omega)$ and therefore is in $\boldsymbol{S}_0(\mathbb{R}^{2d})$ for $g_1, g_2 \in \boldsymbol{S}_0(\mathbb{R}^d)$, as $\boldsymbol{S}_0$ is invariant under the Fourier transform $\mathcal{F}$, the tensor product $\otimes$ and TF-shifts [Proposition 1.5.1].

This means, that $\mathcal{GM}_2$ is a closed subspace of $\mathcal{HS}$ if the functions

$$\mathrm{T}_\lambda\,\sigma(P_0) = \mathrm{T}_\lambda(e^{-2\pi i\omega x}[g_1\otimes\widehat{g_2}^*](x,\omega)), \quad \lambda\in\Lambda$$

form a Riesz basis for their closed linear span. This can be extended to $(\mathcal{GM}_1, \mathcal{GM}_2, \mathcal{GM}_\infty)$ being closed subspaces of $(\mathcal{B}, \mathcal{HS}, \mathcal{B}')$.

Therefore we can use Equations (3.15) and (3.14) for the best approximation with Gabor multpliers. Given an operator T from $(\mathcal{B}, \mathcal{HS}, \mathcal{B}')$, we go to the KNS $\sigma(T)$. Then we do a best approximation in $\mathbf{V}_{\sigma(P_0),\Lambda}$ for a lattice Λ and an atom $P_0 \in \boldsymbol{S}_0(\mathbb{R}^{2d})$ and go back from KNS to operators. This way we get T_{app}, the best approximation by a Gabor multiplier.

4.3.2. The Spreading Function

Definition 4.3.3 (Spreading Representation)**.** *The* spreading representation *of an operator* T *is an expansion with unitary time-frequency shift operators* $\pi(\lambda) = \mathrm{M}_\omega\,\mathrm{T}_t$ *as*

$$T = \int_{\mathbb{R}^d\times\widehat{\mathbb{R}}^d} \eta(T)(t,\omega)\,\mathrm{M}_\omega\,\mathrm{T}_t\,d\omega dt \tag{4.14}$$

with the so-called spreading function *of* T

$$\eta(T)(t,\omega) = \int_{\mathbb{R}^d} \kappa(T)(x, x-t) e^{-2\pi i \omega x} dx. \tag{4.15}$$

The mapping from operator T to its spreading function is, same as for the Kohn-Nirenberg symbol, invertible for operators in $\mathcal{B}$ and spreading function in $\boldsymbol{S}_0(\mathbb{R}^d \times \widehat{\mathbb{R}}^d)$.

$$\kappa(T)(x,t) = \int_{\widehat{\mathbb{R}}^d} \eta(T)(x-t, \omega) e^{2\pi i x \omega} d\omega. \tag{4.16}$$

It even extends to a unitary Gelfand triple isomorphism form $(\mathcal{B}, \mathcal{HS}, \mathcal{B}')$ to $(\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')(\mathbb{R}^d \times \widehat{\mathbb{R}}^d)$ and therefore it holds that

$$\langle T, S \rangle_{(\mathcal{B}, \mathcal{HS}, \mathcal{B}')} = \langle \eta(T), \eta(S) \rangle_{(\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')(\mathbb{R}^d \times \widehat{\mathbb{R}}^d)}. \tag{4.17}$$

The spreading function relates to the Kohn-Nirenberg symbol vie the *symplectic Fourier transform* which arises in a natural way as a slightly modified Fourier transform on the TF-plane.

Definition 4.3.4 (Symplectic Fourier Transform)**.** *The* symplectic Fourier transform *on* $\mathbb{R}^d \times \widehat{\mathbb{R}}^d$ *is formally defined by*

$$\mathcal{F}_s f(x,\omega) := \int_{\mathbb{R}^d \times \widehat{\mathbb{R}}^d} f(t,\xi) e^{2\pi i (x\xi - t\omega)} dt d\eta. \tag{4.18}$$

The symplectic Fourier transform $\mathcal{F}_s$ is self-inverse, i.e. $\mathcal{F}_s^{-1} = \mathcal{F}_s$, and defines a unitary Gelfand triple isomorphism on $(\boldsymbol{S}_0, \boldsymbol{L}^2, \boldsymbol{S}_0')(\mathbb{R}^d \times \widehat{\mathbb{R}}^d)$. For this and where the symplectic Fourier transform comes from, see [FK98] or [FLW07].

Lemma 4.3.2. *For an operator* T*, its spreading function* $\eta(T)$ *and its Kohn-Nirenberg symbol* $\sigma(T)$*, the following identity holds*

$$\sigma(T) = \mathcal{F}_s[\eta(T)]. \tag{4.19}$$

Proof. Let $k_t(x) := \kappa(T)(x, x-t)$ then

$$\begin{aligned} k_t(x) &= \mathcal{F}^{-1}\mathcal{F}(k_t)(x) \\ &= \mathcal{F}^{-1}\left(\int_{\mathbb{R}^d} k_t(y) e^{-2\pi i y \xi} dy \right)(x) \\ &= \int_{\widehat{\mathbb{R}}^d} \int_{\mathbb{R}^d} k_t(y) e^{-2\pi i y \xi} dy \; e^{2\pi i \xi x} d\xi \\ &= \int_{\mathbb{R}^d \times \widehat{\mathbb{R}}^d} \kappa(T)(y, y-t) e^{2\pi i \xi (x-y)} dy d\xi \end{aligned} \tag{4.20}$$

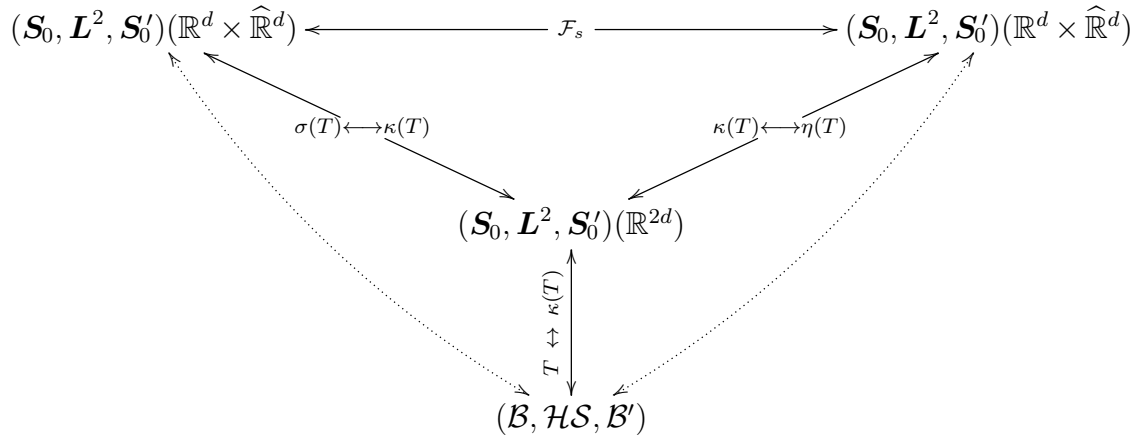


Figure 4.6.: Commutative diagram for the Gelfand triple of operators (bottom), their kernels (middle), their Kohn-Nirenberg symbols (top left) and their spreading functions (top right).

and with that we get

$$\begin{aligned}
\mathcal{F}_s[\eta(T)](x,\omega) &= \int_{\mathbb{R}^d \times \widehat{\mathbb{R}}^d} \eta(T)(t,\xi) e^{2\pi i(x\xi - t\omega)} dt d\xi \\
&= \int_{\mathbb{R}^d \times \widehat{\mathbb{R}}^d} \int_{\mathbb{R}^d} \kappa(T)(y, y-t) e^{-2\pi i \xi y} dy \; e^{2\pi i(x\xi - t\omega)} dt d\xi \\
&= \int_{\mathbb{R}^d} \underbrace{\int_{\mathbb{R}^d \times \widehat{\mathbb{R}}^d} \kappa(T)(y, y-t) e^{2\pi i \xi (x-y)} dy d\xi}_{=\kappa(T)(x,x-t)\text{[see (4.20)]}} \; e^{-2\pi i \omega t} dt \\
&= \int_{\mathbb{R}^d} \kappa(T)(x, x-t) e^{-2\pi i \omega t} dt \\
&= \sigma(T)(x,\omega).
\end{aligned}$$

□

As the symplectic Fourier transform is self-inverse, we also get

$$\eta(T) = \mathcal{F}_s \mathcal{F}_s[\eta(T)] = \mathcal{F}_s[\sigma(T)]. \tag{4.21}$$

In Figure 4.6 we see the interrelationship of the Kohn-Nirenberg symbols, the spreading functions and the (distributional) kernels of operators.

To find out which operators can be well approximated with Gabor multipliers, we first look at the KNS of a general Gabor multiplier. Because of (4.12), we see that the KNS of a Gabor multiplier is given by a convolution of the upper symbol $\mathbf{m}$ with the KNS of P_0

$$\sigma(G_{\mathbf{m}}) = \sum_{\lambda \in \Lambda} m_\lambda \, \mathrm{T}_\lambda \, \sigma(P_0) = \mathbf{m} *_\Lambda \sigma(P_0). \tag{4.22}$$

As a cosequence we get the following corollary.

Corollary 4.3.1. *A given operator T can be exactly represented as a Gabor multiplier with respect to $\mathcal{G}(\mathbf{g}, \Lambda)$, if and only if $(\mathcal{V}_g g(\lambda))_{\lambda\in\Lambda}$ is bounded away from 0 on the support of the spreading function $\eta(T)$.*

Proof. By applying the symplectic Fourier transform to (4.22) we get

$$\eta(T) = \mathcal{F}_s(\sigma(T)) = \mathcal{F}_s\left(\sigma_U(T) *_\Lambda \sigma(P_0)\right) = \mathcal{F}_s\sigma_U(T) \cdot \eta(P_0).$$

Here we used, that the symplectic Fourier transform turns convolution into pointwise multiplication just as the common Fourier transform. The speading function of P_0 is

$$\begin{aligned}\eta(P_0)(t,\omega) &= \int_{\mathbb{R}^d} \kappa(P_0)(x, x-t)e^{-2\pi i x\omega}dx \\ &= \int_{\mathbb{R}^d} g_1(x)\overline{g_2(x-t)}e^{-2\pi i x\omega}dx = \mathcal{V}_g g(t,\omega).\end{aligned}$$

Therefore the upper symbol of T is well defined, if $\mathcal{V}_g g$ satisfies the condition above, as

$$\sigma_U(T) = \mathcal{F}_s\left(\frac{\eta(T)}{\mathcal{V}_g g}\right).$$

(cf.[Dör02]) □

4.4. Classes of Operators

We will take a look at two classes of operators which can be well approximated by Gabor multipliers. Experiments for approximation of them will be done in Chapter 5.

4.4.1. Underspread Operators

In mobile communication, the channel we receive a signal through is usually a *slowly time-varying channel*. We therefore try to identify the channel with pilot tones, then try to estimate it and invert it to get the best possible approximation of the original signal. These slowly time-varying channels usually correspond to *underspread operators*.

Definition 4.4.1 (Underspread Operators)**.** *Let $t_0, \nu_0 \in \mathbb{R}_+$. An operator T is called* underspread operator *if the (essetial) support of its spreading function $\eta(T)$ suffices the following condition*

$$\operatorname{supp}(\eta(T)) \subseteq Q(t_0,\nu_0) := [-t_0, t_0] \times [-\nu_0, \nu_0] \tag{4.23}$$

with

$$s := 4t_0\nu_0 < 1. \tag{4.24}$$

If $s \geq 1$, then T is called overspread*.*

Condition (4.23) tells us, that t_0 corresponds to the maximum time-delay and ν_0 to the maximum Doppler shift of the channel.

4.4.2. STFT Multipliers

As another kind of operator, we introduce the so-called *STFT-multipliers* $SM_{\mathbf{m}}$, which are the continuous version of a Gabor multiplier. Approximating a STFT-multiplier with a Gabor multiplier is in a way similar to approximating an integral with Riemannian sums. There are certain results for the limiting case (done only for seperable lattices, cf. [FN03, Theorem 5.9.8.]), when we let $(a_k, b_k) \to (0,0)$ for $k \to \infty$ such that the lattices Λ_k get denser and denser. Then the series of Gabor multipliers $G_{\mathbf{m}_k}$, with upper symbol being just the symbol $\mathbf{m}$ sampled on the lattice $a_k\mathbb{Z} \times b_k\mathbb{Z}$, converges to $SM_{\mathbf{m}}$ in the trace-class operator norm over $\boldsymbol{L}^2(\mathbb{R}^d)$. Let us define these operators.

Definition 4.4.2 (STFT-Multipliers)**.** *For a window $g \in \boldsymbol{S}_0(\mathbb{R}^d)$ the* STFT-multiplier *with (upper) symbol $\mathbf{m}$ is given by*

$$SM_{\mathbf{m}}(f) = \mathcal{V}_g^*(\mathbf{m} \cdot \mathcal{V}_g(f)) \tag{4.25}$$

which is (in the weak sense on $\boldsymbol{L}^2(\mathbb{R}^d)$)

$$SM_{\mathbf{m}}(f) = \int\limits_{\mathbb{R}^d \times \widehat{\mathbb{R}}^d} m(x,\omega)\langle f, \pi(x,\omega)g\rangle \pi(x,\omega)g \; dxd\omega. \tag{4.26}$$

From this definition we directly get

Proposition 4.4.1. *If $g \in \boldsymbol{S}_0(\mathbb{R}^d)$ and $\mathbf{m} \in \boldsymbol{L}^2(\mathbb{R}^{2d})$, then the resulting operator $SM_{\mathbf{m}}$ is in $\mathcal{HS}$.*

Proof. Let g and $\mathbf{m}$ be as above, then we get $\kappa(P_{(x,\omega)}) = \pi(x,\omega)[g \otimes g^*]\pi^*(x,\omega)$ is in $\boldsymbol{S}_0(\mathbb{R}^{2d}) \subseteq \boldsymbol{L}^2(\mathbb{R}^{2d})$ and therefore $\kappa(SM_{\mathbf{m}}) = m(x,\omega)\kappa(P_{(x,\omega)}) \in \boldsymbol{L}^2(\mathbb{R}^{2d})$. This is equivalent to $SM_{\mathbf{m}} \in \mathcal{HS}$, as it has the form

$$SM_{\mathbf{m}}f(y) = \int\limits_{\mathbb{R}^d \times \widehat{\mathbb{R}}^d} [m(x,\omega)\kappa(P_{(x,\omega)})](y,z)f(z)dxd\omega dz.$$

□

4.5. Gabor Multipliers in the Discrete Setting

When doing computations, linear operators are approximately represented by matrices. This means, that when we approximate an operator T by a Gabor multiplier $G_{\mathbf{m}}$, we actually approximate a $n \times n$-matrix $T \in \mathcal{M}_n$ by the matrix representation of a Gabor multiplier. This means, that we want to find $T_{app} = \sum_{\lambda\in\Lambda} c_\lambda P_\lambda$ satisfying

$$\min = \|T - T_{app}\|_{\text{Fro}}.$$

Here we assume that $(P_\lambda)_{\lambda\in\Lambda}$ form a Riesz basis for their closed linear span. We introduce the canonical isomorphism $\Phi : \mathcal{M}_n \to \mathbb{C}^{n^2}$ as

$$\Phi(T) = ((Te_1)^T, (Te_2)^T, \ldots, (Te_n)^T) =: \tilde{T}.$$

This means that Φ transforms a matrix to a raw vector by putting its consecutive columns on top of each other and then transpose it. In this way we can view the collection of projection operators $(P_\lambda)_{\lambda\in\Lambda}$ (matrices) as vectors and put them together into one single matrix, i.e.

$$A := \begin{pmatrix} \tilde{P}_{\lambda_1} \\ \tilde{P}_{\lambda_2} \\ \vdots \\ \tilde{P}_{\lambda_r} \end{pmatrix} \in \mathcal{M}_{r\times n^2}$$

with $r = \#\Lambda$ being the size of the (finite) lattice. Our problem becomes the following. Find $\vec{\mathbf{c}} = (c_{\lambda_1}, c_{\lambda_2}, \ldots, c_{\lambda_r})$ which solves

$$min = \left\| T - \sum_{\lambda\in\Lambda} c_\lambda P_\lambda \right\|_{\text{Fro}} = \|\tilde{T} - \vec{\mathbf{c}} \cdot A\|_2.$$

The solution for this problem is given by

$$\vec{\mathbf{c}}_A = \tilde{T} \cdot A^\dagger$$

with the pseudo-inverse $A^\dagger$ as of Theorem 1.1.3. As P_λ are assumed to be a Riesz basis, they are linearly independent and therefore the pseudo-inverse is given by $A^\dagger = A^*(AA^*)^{-1}$. We denote $G_A = AA^*$ and get as coefficiants for the best approximation

$$\vec{\mathbf{c}}_A = \tilde{T} \cdot A^* \cdot G_A^{-1} \text{ and } \tilde{T}_{app} = \vec{\mathbf{c}}_A \cdot A.$$

In this setting, we get the following.

(i) $f : \mathbb{C}^{n^2} \to \mathbb{C}^r; \tilde{T} \mapsto \tilde{T} \cdot A^*$, maps a matrix T to its lower symbol s (via Φ).

(ii) $g : \mathbb{C}^r \to \mathbb{C}^r; s \mapsto s \cdot G_A^{-1}$, maps the lower symbol s to the upper symbol m.

(iii) $h : \mathbb{C}^r \to \mathbb{C}^{n^2}; m \mapsto m \cdot A$, maps the upper symbol m to the Gabor multiplier matrix T_{app}.

The straight forward calculations are not the best approach here, but there are quite efficient ways to compute the mappings above. The interested reader can find explanations for how efficient calculations based on the FFT work in [FHK04].

Chapter 5.

Applications and Experiments

5.1. Condition of a Gabor System $(\mathbf{g}, \Lambda)$

The aim in this chapter is to find criteria which tell us how good a Gabor system $(\mathbf{g}, \Lambda)$ is for calculations. So we want to find a Gabor system, where the atom g and dual atom $\widetilde{g}$ are both well localized, e.g. a good system would be a well localized atom g and a lattice Λ such that the dual $\widetilde{g}$ is a nice atom as well. In the same way we need the family of projection operators P_λ to form a Riesz basis for their linear span in the space of Hilbert Schmidt operators, as this certifies that our calculations are stable.

Over all we are looking for good Gabor systems to compute Gabor multipliers. With this in mind we try to find a good criteria for both, the system $(\mathbf{g}, \Lambda)$ and the projection operators P_λ in one which leads us to the *compound condition number*.

In numerical mathematics, condition numbers are a way of declaring a measure for speed of numerical algorithems (in terms of convergence) and computational stability. For that we can use them as a measure of quality of a Gabor system in the computational sense.

5.1.1. Condition Number of the Frame Operator $S_{g,\Lambda}$

The condition number of an operator A is given by

$$cond(A) = |||A||| \cdot |||A^{-1}|||$$

As this is a quite straigth foreward way and calculating the inverse of the frame operator is sometimes quite costly, we want to find an estimate for the condition number. For this we need two aspects. First there should be a faster way to calculate it and second it should not be an estimate too high above the true value.

Looking at the Janssen Representation we can estimate the norm of the frame operator by

$$|||S_{g,\Lambda}||| \leq \mathrm{red}_\Lambda \sum_{\lambda^\circ \in \Lambda^\circ} |\mathcal{V}_g g(\lambda^\circ)| \underbrace{|||\pi(\lambda^\circ)|||}_{=1} = \mathrm{red}_\Lambda \, \gamma_2. \tag{5.1}$$

with $\gamma_2 := \sum_{\lambda^\circ \in \Lambda^\circ} |\mathcal{V}_g g(\lambda^\circ)|$.

Now we know from the proof of Theorem 3.2.1 that

$$|||\mathrm{red}_\Lambda \, S_{g,\Lambda}^{-1}||| \leq \sum_{k=0}^{\infty} \gamma_1^k$$

with $\gamma_1 := \sum_{\lambda^\circ \neq 0} |\mathcal{V}_g g(\lambda^\circ)|$. We see that $\gamma_2 = \gamma_1 + 1$ assuming $\|g\| = 1$ which yields $\mathcal{V}_g g(0) = 1$. If $\gamma_1 < 1$ then $\sum_{k=0}^{\infty} \gamma_1^k = \frac{1}{1-\gamma_1}$ and we get an upper bound for the operatornorm of the inverse frame operator.

$$\|\|\|S_{g,\Lambda}^{-1}\|\|\| \leq \frac{1}{\mathrm{red}_\Lambda(1-\gamma_1)} \tag{5.2}$$

With this estimate we get an upper bound for the condition number of the frame operator.

$$cond(S_{g,\Lambda}) \leq \frac{1+\gamma_1}{1-\gamma_1} = \frac{\gamma_2}{2-\gamma_2} \tag{5.3}$$

This upper bound is easier to calculate than the condition number of the frame operator itself, because one does not have to explicitely calculate the inverse of the frame operator. This saves a lot of computational work for higher dimensions. γ_2 can be calculated by only one STFT and one sum. There remains the question of how close this upper bound is to the condition number. We denote it from here on as the *Janssen condition number*.

Definition 5.1.1. *The condition number of the Gabor system* $(\mathbf{g}, \Lambda)$ *is defined by the condition number of the analysis* C *or the synthesis* D *mapping or equivalently the square root of the condition number of the frame operator* $S_{g,\Lambda}$*, e.g.*

$$g(\mathbf{g}, \Lambda) := \sqrt{\kappa(S_{g,\Lambda})} \tag{5.4}$$

5.1.2. Condition Number of the Family of Projection Operators P_λ

We want to have a unique representation of Gabor multipliers, which means we need the projection operators P_λ to be a Riesz basis in $\mathcal{HS}$.

For testing the stability of the family of projection operators P_λ, we look at the Gram matrix. It always has 1 in the main diagonal (for normalized g) and decreases quickly in the sidediagonals.

Here we assume that the atom g is normalized, as $\|g\| = 1$ implies $\mathcal{V}_g g(0) = \langle g, g\rangle = \|g\|^2 = 1$. Then

$$\sum_{\lambda \in \Lambda} |\mathcal{V}_g g(\lambda)|^2 - 1$$

is actually the sum over a row/column of the Gram matrix of $\{P_\lambda\}_{\lambda\in\Lambda}$ without the entry in the main diagonal, because the entries of the Gram matrix are

$$\begin{aligned}
\langle P_\lambda, P_{\lambda'}\rangle_{\mathcal{HS}} &= \langle g_\lambda \otimes g_\lambda^*, g_{\lambda'} \otimes g_{\lambda'}^*\rangle_{\boldsymbol{L}^2(\mathbb{R}^{2d})} = \langle g_\lambda, g_{\lambda'}\rangle_{\boldsymbol{L}^2(\mathbb{R}^d)} \langle g_\lambda^*, g_{\lambda'}^*\rangle_{\boldsymbol{L}^2(\mathbb{R}^d)} = \\
&= |\langle \pi(\lambda) g, \pi(\lambda') g\rangle|^2 = |\langle g, \pi^*(\lambda)\pi(\lambda') g\rangle|^2 = \\
&= |\langle g, e^{2\pi i x(\omega'-\omega)} \pi(\lambda'-\lambda) g\rangle|^2 = |\langle g, \pi(\lambda'-\lambda) g\rangle|^2 = \\
&= |\mathcal{V}_g g(\lambda'-\lambda)|^2.
\end{aligned} \tag{5.5}$$

What we need is that the Gram matrix is invertible on $\boldsymbol{\ell}^2(\Lambda)$, as invertibility of the Gram matrix is equivalent to P_λ forming a Riesz basis for their linear span in $\mathcal{HS}$. This means that approximation of $\mathcal{HS}$-operators as Gabor multipliers with $\boldsymbol{\ell}^2$

coefficiants is stable. Invertibility is equivalent to $\sum_{\lambda\in\Lambda} |\mathcal{V}_g g(\lambda)|^2 - 1 < 1$. But we want a better measure of how good it is or how "far" from singular.

Now as the Gram matrix is circulant, it can be diagonalized by the Fourier transform which then yields the eigenvalues. For general lattices Λ, we need to take the Λ-Fourier transform. In this manner, we define the condition number of the projection operators P_λ. We take the eigenvalues and build the square root of the quotient of its maximum over its minimum.

Definition 5.1.2. *The P-condition number of the projection operators P_λ for the lattice Λ is defined by*

$$p(\mathbf{g}, \Lambda) := \sqrt{\frac{\max_{\lambda\in\Lambda} \mathcal{F}_\Lambda(|\mathcal{V}_g g(\lambda)|)}{\min_{\lambda\in\Lambda} \mathcal{F}_\Lambda(|\mathcal{V}_g g(\lambda)|)}} \tag{5.6}$$

As we want both, the condition number of the frame operator and the projection operators to be nice (i.e. small), we will introduce the *compound condition number* which is just the product of the two. We can see that if red_Λ gets higher, we get higher $p(\mathbf{g}, \Lambda)$ and for $\text{red}_\Lambda \to 1$, the critical case, we get higher $g(\mathbf{g}, \Lambda)$. So optimal would be both as small as possible.

Definition 5.1.3 (Compound condition number)**.** *The* compound condition number *is defined as the product of the condition number of the Gabor system $(\mathbf{g}, \Lambda)$ and the condition number of the projection operators P_λ.*

$$c(\mathbf{g}, \Lambda) := g(\mathbf{g}, \Lambda) \cdot p(\mathbf{g}, \Lambda) \tag{5.7}$$

A small compound condition number indicates small $g(\mathbf{g}, \Lambda)$ and small $p(\mathbf{g}, \Lambda)$, a balance between the two condition numbers.

5.1.3. Finding good Gabor Systems within a given family

In MATLAB™ a lattice is represented by a $(0, 1)$-Matrix of size $n \times n$ with the ones being the lattice points. The MATLAB™ function `sidedigm.m`, developed by NuHAG, produces a lattice from a given lattice by putting the sidediagonals of the lattice to rows of the new lattice. This is always a lattice as it produces another subgroup (Figure 5.1).

When given a certain size n and redundancy *red* one can find all a and b for this redundancy and build the seperable lattices $\Lambda = a\mathbb{Z} \times b\mathbb{Z}$ for them (`lattp.m` by NuHAG). Then applying `sidedigm.m` repeatedly gives a certain number of lattices until it comes back to the original. It can only be finitely many times, as a finite group has only finitely many subgroups of a certain order. We store all the different ones we get, which are not all there are, but a set which we can use for our tests. Now we call this set of lattices with a fixed redundancy the set $\{\Lambda_i^{red}\}_{i\in I}$. It can contain seperable and non-seperable lattices.

We want to test such a set of lattices $\{\Lambda_i^{red}\}_{i\in I}$ with a Gabor atom g for a good Gabor system. This is done by calculating the compound condition number of the pair (g, Λ_i^{red}) for all $i \in I$ and then finding the lowest $c(g, \Lambda_i^{red})$ for the set.

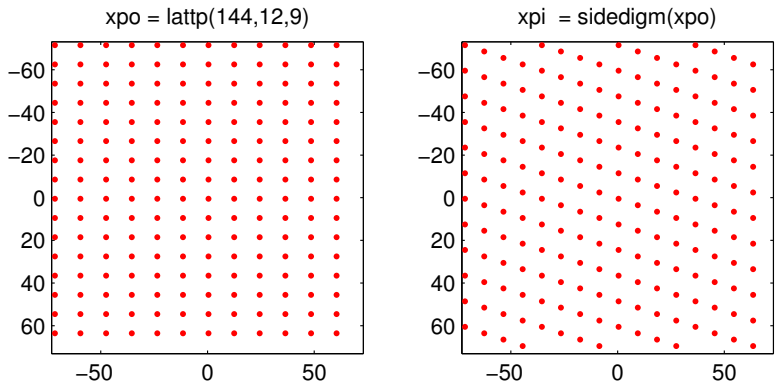


Figure 5.1.: Seperable lattice xpo and `sidedigm.m` applied to it once

Experiment 1. *To find the good lattices in a set* $\{\Lambda_i^{red}\}_{i\in I}$ *for a fixed window (in these cases a standard Gaussian window is used), we go through the index set and calculate for each set* (g, Λ_i^{red}) *the compound condition number. Plotting the lattice with the lowest resp. highest condition number gives an idea of the form of a "good" resp. "bad" lattice. This is done for various redundancies and the condition numbers is ploted in the Figures 5.2 - 5.7. The black stars mark good lattices, with compound condition number lower than 2 times the minimum. Beneath the graph one can see the best and worst lattice of the set. In Figure 5.3 we see the same test results with the canonical tight window. What we can easily observe is that in this context the same lattices are good resp. bad, but we get a lower compound condition number for all of them. One might wonder why in Figure 5.2 the minimum lattice is* #5 *where as in Figure 5.3 it is* #10*. They actually both have (in each setting) the same compound condition number up to numerical precession.*

In the rest of the figures, one can see a variation of several n *and* red *to see what happens. We see that for fixed* red *and variable* n*, the minimum does not change a lot. This is good, because it indicates that the sampling size does not affect our computations.*

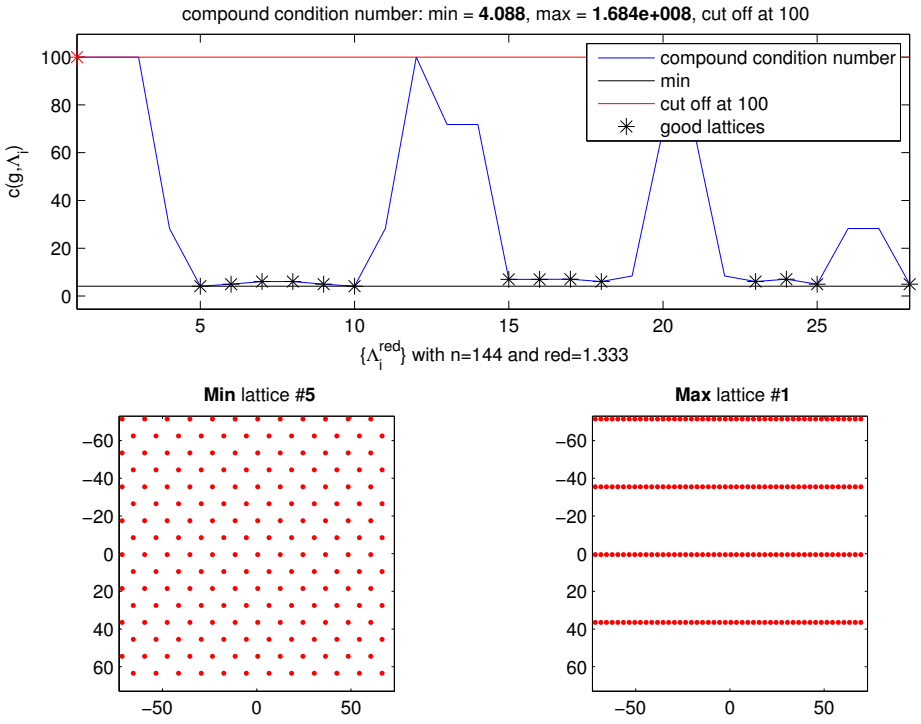


Figure 5.2.: Compound condition number, n = 144, red = 1.333

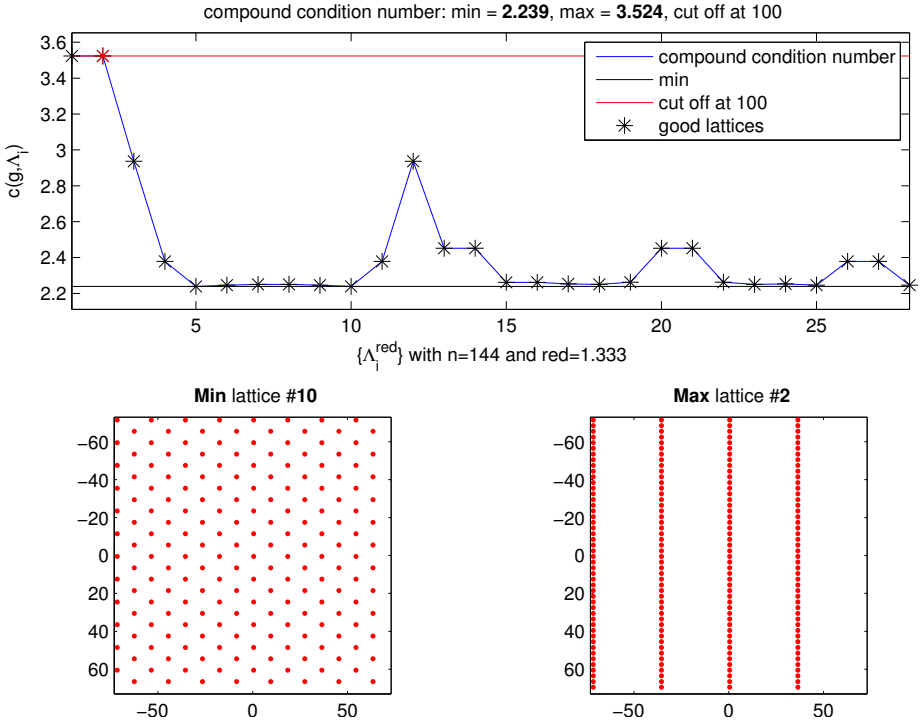


Figure 5.3.: Compound condition number, n = 144, red = 1.333, tight window

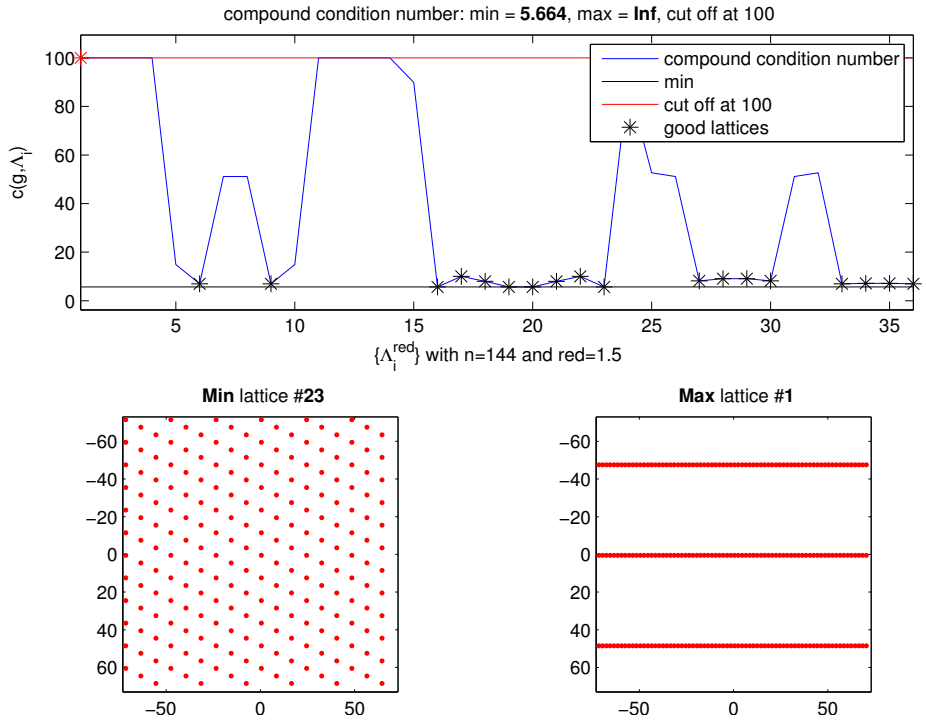


Figure 5.4.: Compound condition number, n = 144, red = 1.5

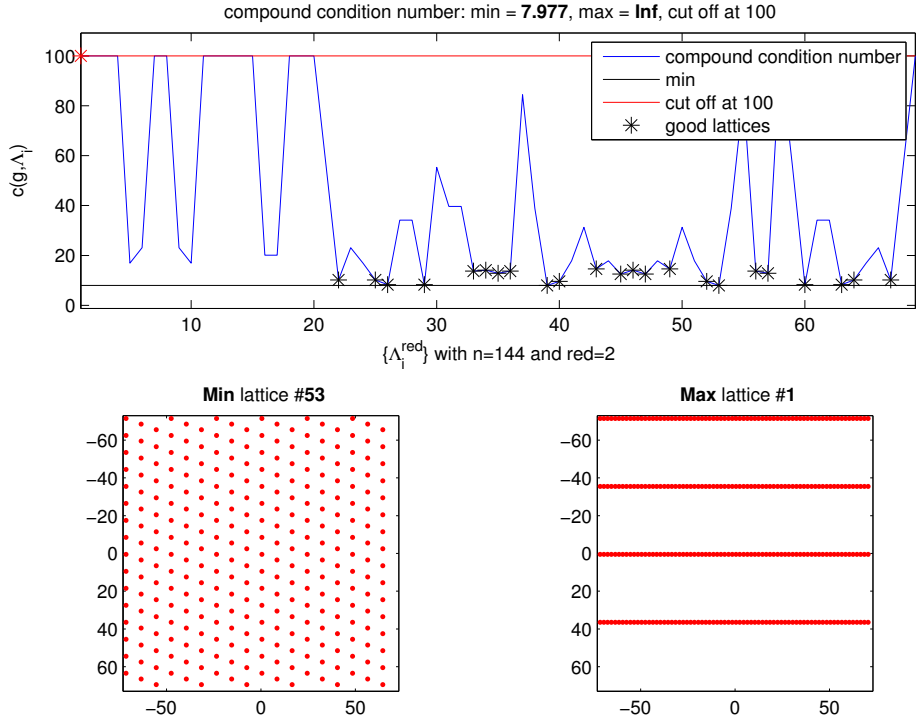


Figure 5.5.: Compound condition number, n = 144, red = 2

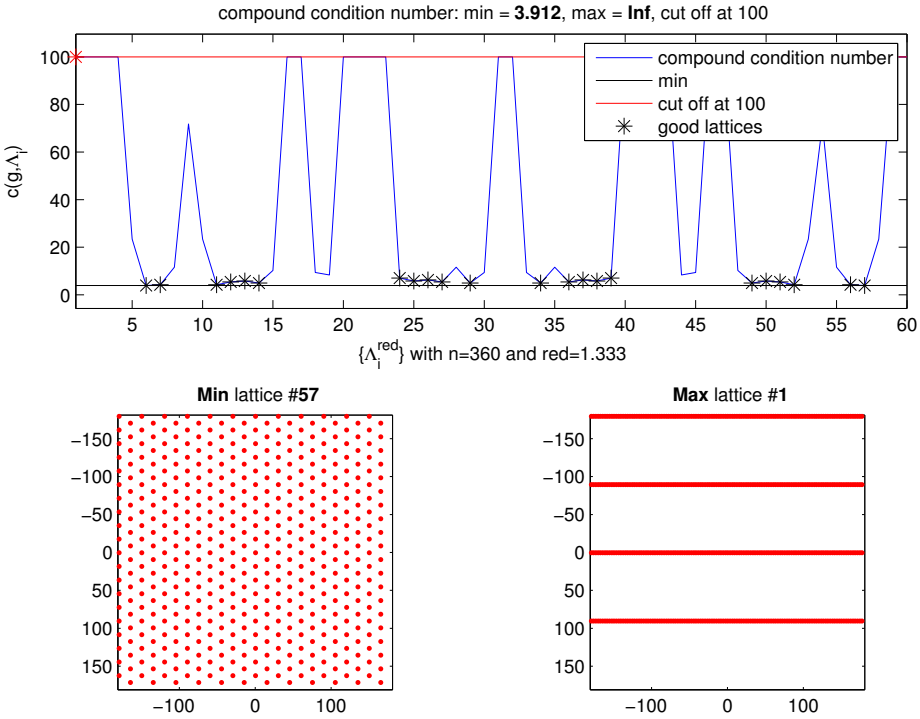


Figure 5.6.: Compound condition number, n = 360, red = 1.333

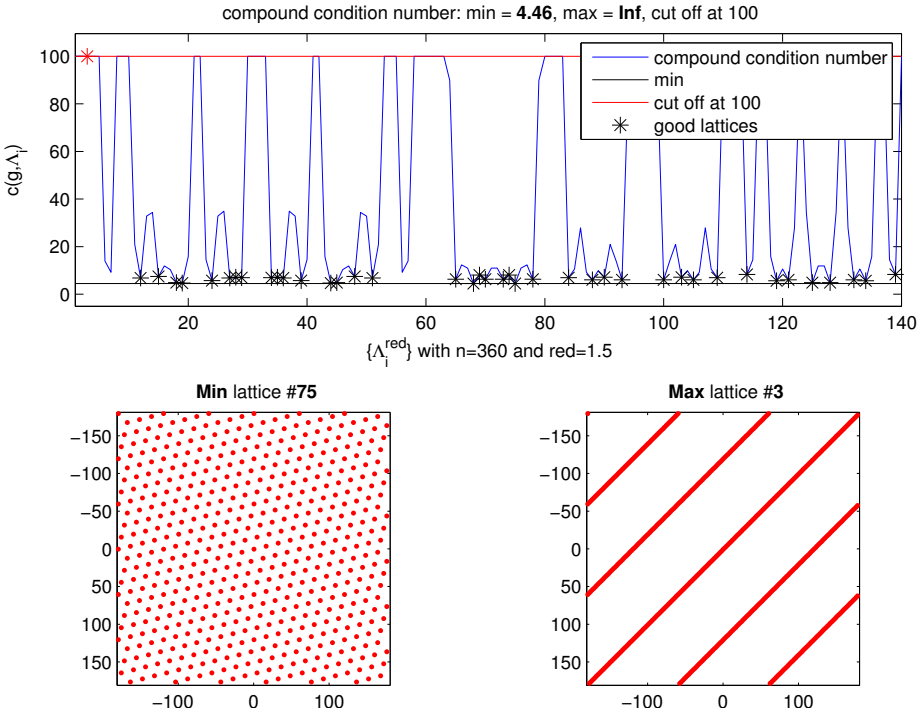


Figure 5.7.: Compound condition number, n = 360, red = 1.5

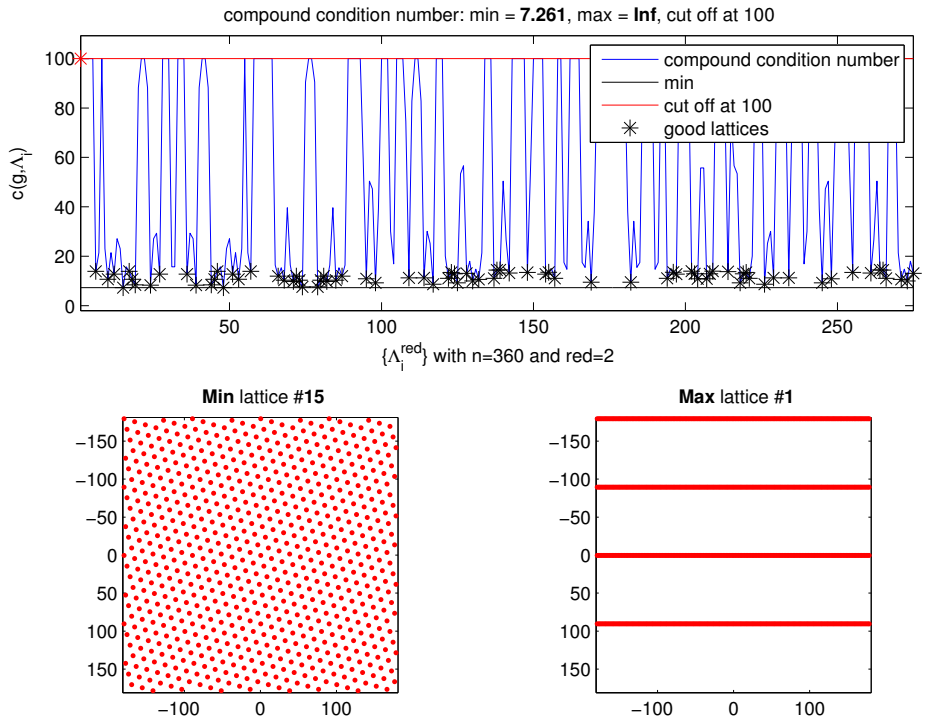


Figure 5.8.: Compound condition number, n = 360, red = 2

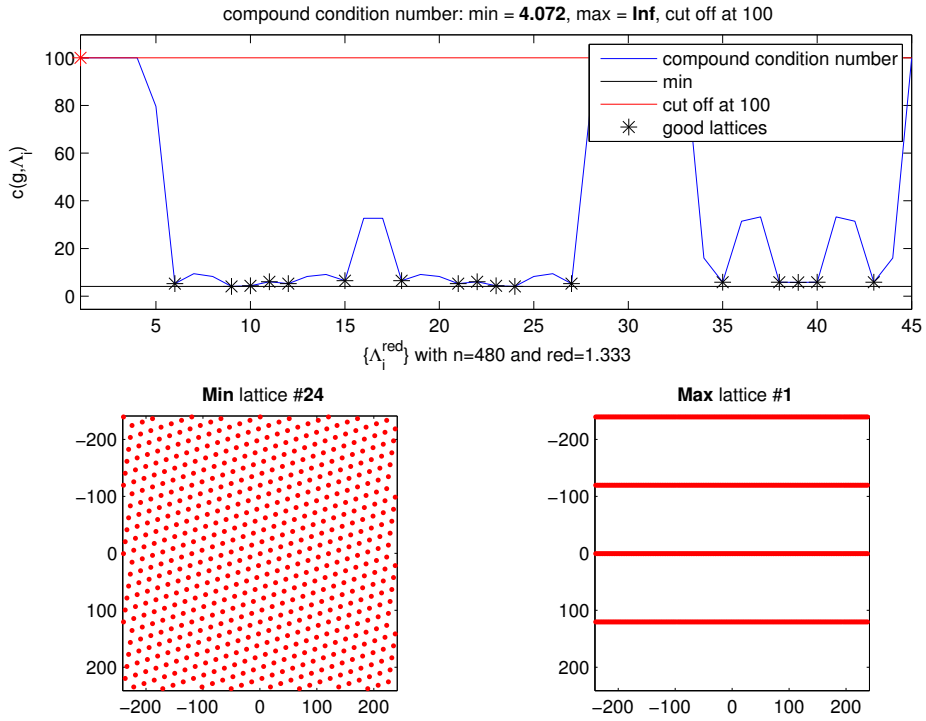


Figure 5.9.: Compound condition number, n = 480, red = 1.333

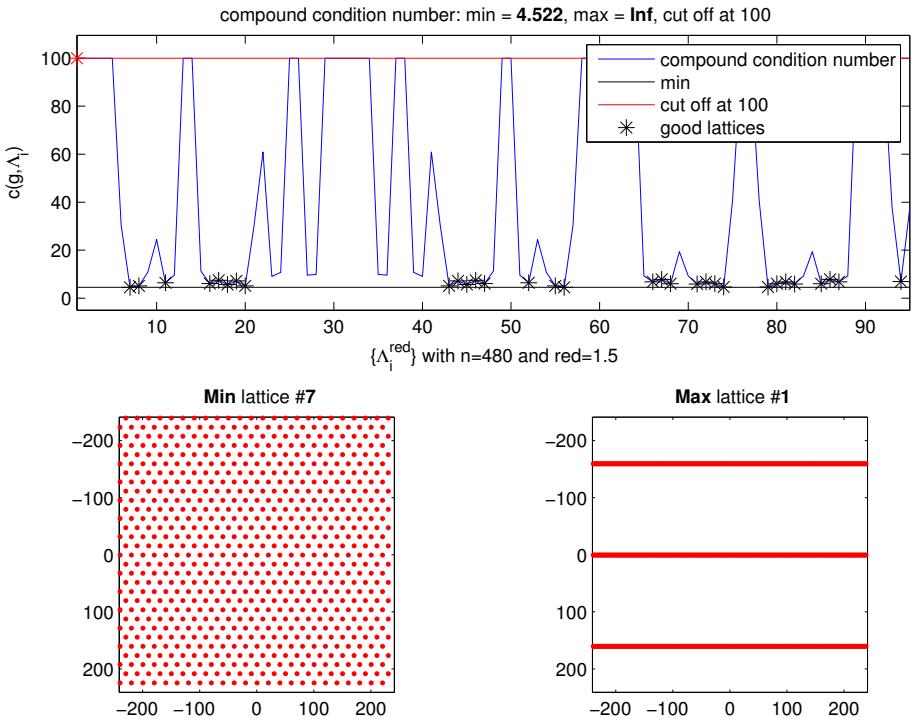


Figure 5.10.: Compound condition number, n = 480, red = 1.5

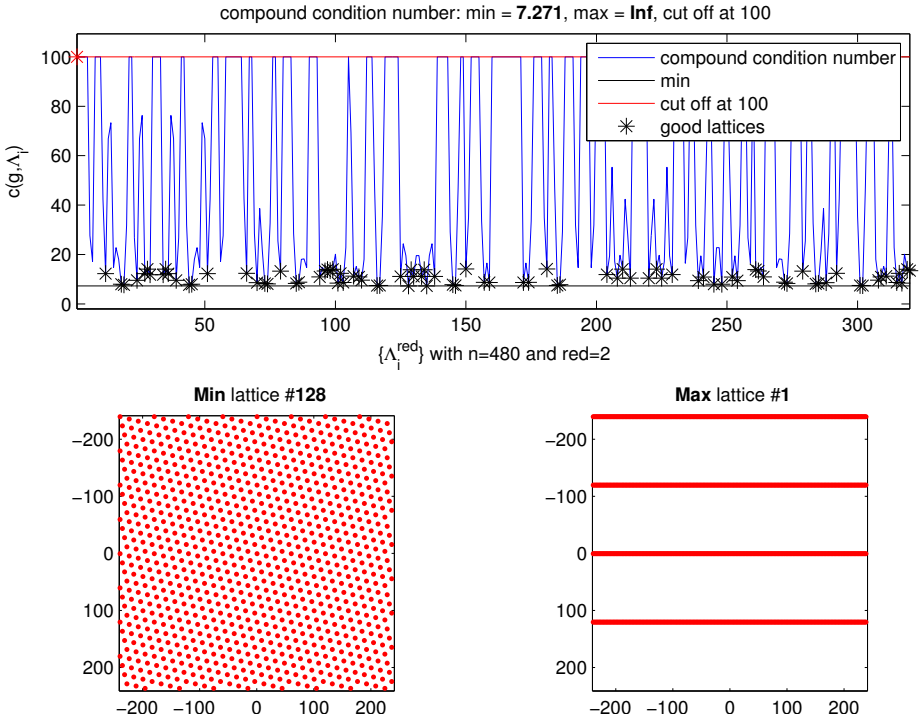


Figure 5.11.: Compound condition number, n = 480, red = 2

Looking at the different figures created, one can see that (for a standard Gaussian) we always get a lattice, that is a slightly tilted hexagonal lattice, as the best lattice. One obsevation was, that the lattices with minimal compound condition number all are of that form. So we can see that for a Gaussian window it is a good choice to take a hexagonal lattice.

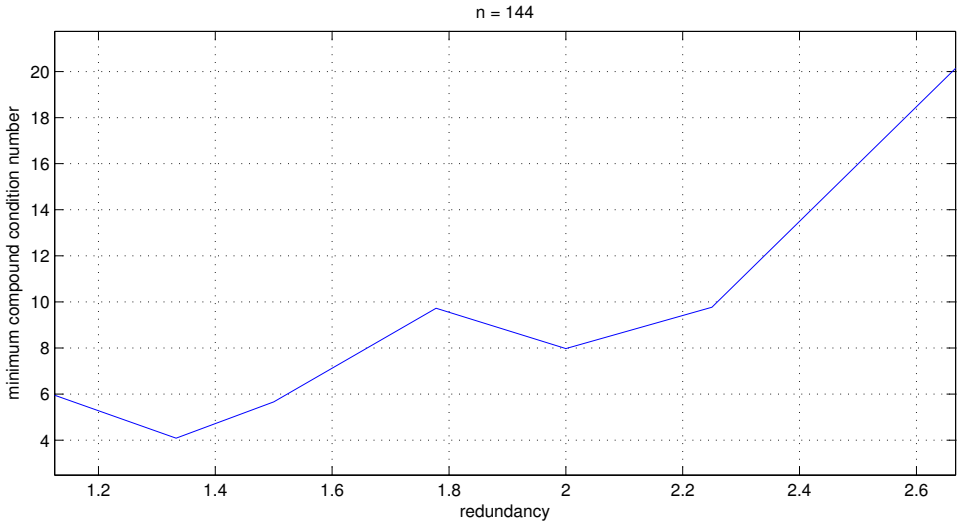


Figure 5.12.: Minimum compound condition number for different redundancies

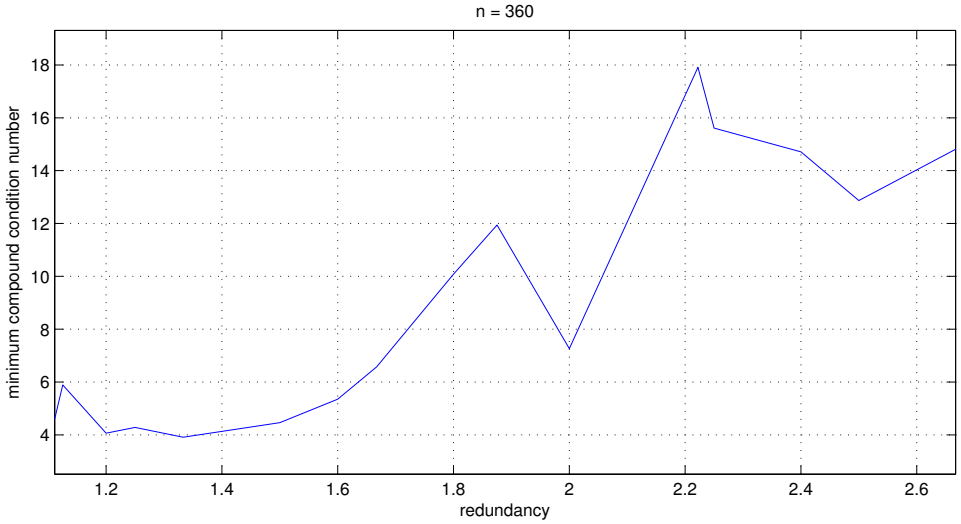


Figure 5.13.: Minimum compound condition number for different redundancies

In Figures 5.12 - 5.14 the minimum compound condition number for various redundancies and fixed $n = 144, n = 360$ and $n = 480$ is plotted. The observation here is, that as the redundancy rises, so does the compound condition number. We get the best condition numbers somewhere between redundancy 1.2 and 1.6. Again this is valid independent of the sampling size n.

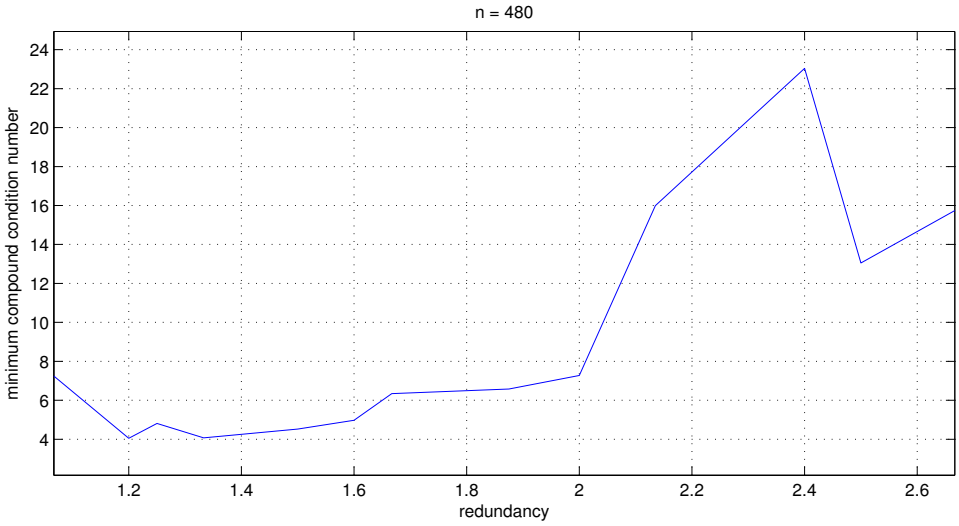


Figure 5.14.: Minimum compound condition number for different redundancies

5.2. Best Operator Approximation by Gabor Multipliers

In this section we will do some tests of the quality of an approximation of Hilbert-Schmidt operators by Gabor multipliers for different Gabor systems. For computational reasons we approximate via the Shift-Covariance Lemma 4.3.1 in the Kohn-Nirenberg representation. Note: The Kohn-Nirenberg symbol is an unitary equivalent representation for $\mathcal{HS}$-operators.

5.2.1. Best Approximation by Linear Combination of Translates

We want to do a best approximation (in the Hilbert-Schmidt sense) of a $\mathcal{HS}$-operator by Gabor multipliers with analysis/synthesis atoms g_1 and g_2 and the lattice Λ.

It is done via the shift covariance of the Kohn-Nirenberg symbol (KNS) for P_0 with $\kappa(P_0) = g_1 \otimes g_2^*$. Thus the KNS $\sigma(T)$ of an $\mathcal{HS}$-operator T has to be approximated by orthogonal projection onto the spline-type space

$$\mathbf{V}_{\sigma(P_0),\Lambda} := \overline{span}\{\mathrm{T}_\lambda \, \sigma(P_0), \lambda \in \Lambda\}.$$

This space is unitary equivalent to the space of KNS of all Gabor multipliers with g_1, g_2, Λ and an arbitrary sequence m_λ over Λ, because of the shift covariance and the fact that the mapping between an operator and its KNS is unitary. If we let m_λ be in $\big(\boldsymbol{\ell}^1(\Lambda), \boldsymbol{\ell}^2(\Lambda), \boldsymbol{\ell}^\infty(\Lambda)\big)$, then we get to the corresponding space in $(\mathcal{GM}_1, \mathcal{GM}_2, \mathcal{GM}_\infty)$.

Now as $\mathcal{GM}_2$ is a subspace of $\mathcal{HS}$, we can approximate $\mathcal{HS}$-operators by orthogonal projection of its KNS onto the spline-type space $\mathbf{V}_{\sigma(P_0),\Lambda}$.

When doing this, we apply the MATLAB™ routine `gmapr2hf.m`, which does our approximation as follows. First we build the KNS of both, the operator $T \in \mathcal{HS}$ and the atom $P_0 \in \boldsymbol{S}_0(\mathbb{R}^{2d})$. These two are both in $\boldsymbol{L}^2(\mathbb{R}^{2d})$, so the best approximation is the orthogonal projection onto $\mathbf{V}_{\sigma(P_0),\Lambda}$. This is what `trl2aphf.m` does. For a given function $\sigma(T)$ and an atom $\sigma(P_0)$ in $\boldsymbol{L}^2(\mathbb{R}^{2d})$ and a lattice Λ it gives back the orthogonal projetion of $\sigma(T)$ onto $\mathbf{V}_{\sigma(P_0),\Lambda}$.

So what we do is for for an operator T, the analysis/synthesis atoms g_1 and g_2 and the lattice Λ, we go to the KNS of T and $P_0 = g_1 \otimes g_2^*$ and apply `trl2aphf.m` to it. This gives us the best approximation of $\sigma(T)$ in $\sigma(\mathcal{GM}_2)$. So isomorphically going back from the KNS to the operator, we get $PG(T)$, the best approximation.

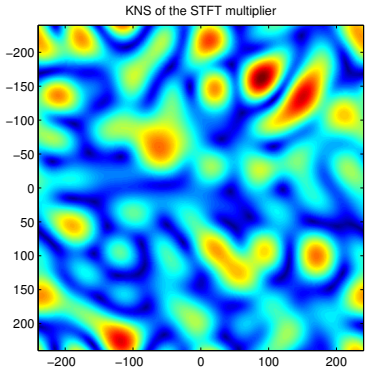


Figure 5.15.: Kohn-Nirenberg symbol of a smooth STFT multiplier

Experiment 2 (Relative error of approximation of a smooth STFT multiplier)**.** *In this experiment, a normalized STFT multiplier T with random smooth symbol of bandwidth 4 is computed. Its KNS can be seen in Figure 5.15. Then it is approximated using the* `gmapr2hf.m` *routine and the relative error of approximation is calculated. In Table 5.1 we see the results of this for various windows and seperable lattices $\Lambda = a\mathbb{Z}^d \times b\mathbb{Z}^d$. The sampling size used is $n = 480$. First the analysis and synthesis window were both standard Gaussians (Figure 5.16). We can clearly see that for increasing redundancy $(a, b \to 0)$, the approximating Gabor multiplier gets closer and closer to the STFT multiplier, but at the same time the dimension for the calculations increases. This is actually theoretical verified [FN03, Theorem 5.9.8.].*

We also see that using the canonical dual window for the re-synthesis or even canonical tight Gabor frame makes the approximation significantly better. At least for lower redundancies, like red $= 1.5$ $(a/b = 16/20)$*. For low redundancies, the advantage of using the tight window is about half the error. But for higher redundancies, like* red $= 7.5$ $(a/b = 8/8)$*, this effect almost vanishes. This seems to be due to the fact that for these high redundancies, the covering is already quite good and we dont need the frame to be tight.*

Approximation of a smooth STFT multiplier with bandwidth 4 symbol

analysis/synthesis	a/b red	16/20 1.5	16/16 1.875	12/16 2.5	10/12 4	8/8 7.5	4/3 40
g/g		26.09%	13.31%	9.51%	1.14%	3.4e-003%	1.3e-010%
g/gd		15.79%	8.44%	6.02%	0.86%	3.1e-003%	4.1e-012%
gt/gt		12.29%	6.76%	4.84%	0.73%	2.9e-003%	3.7e-012%

Table 5.1.: Relative error of approximation for a standard Gaussian (g) as well as for the dual (gd) and tight (gt) window with different lattice constants (a/b), n = 480.

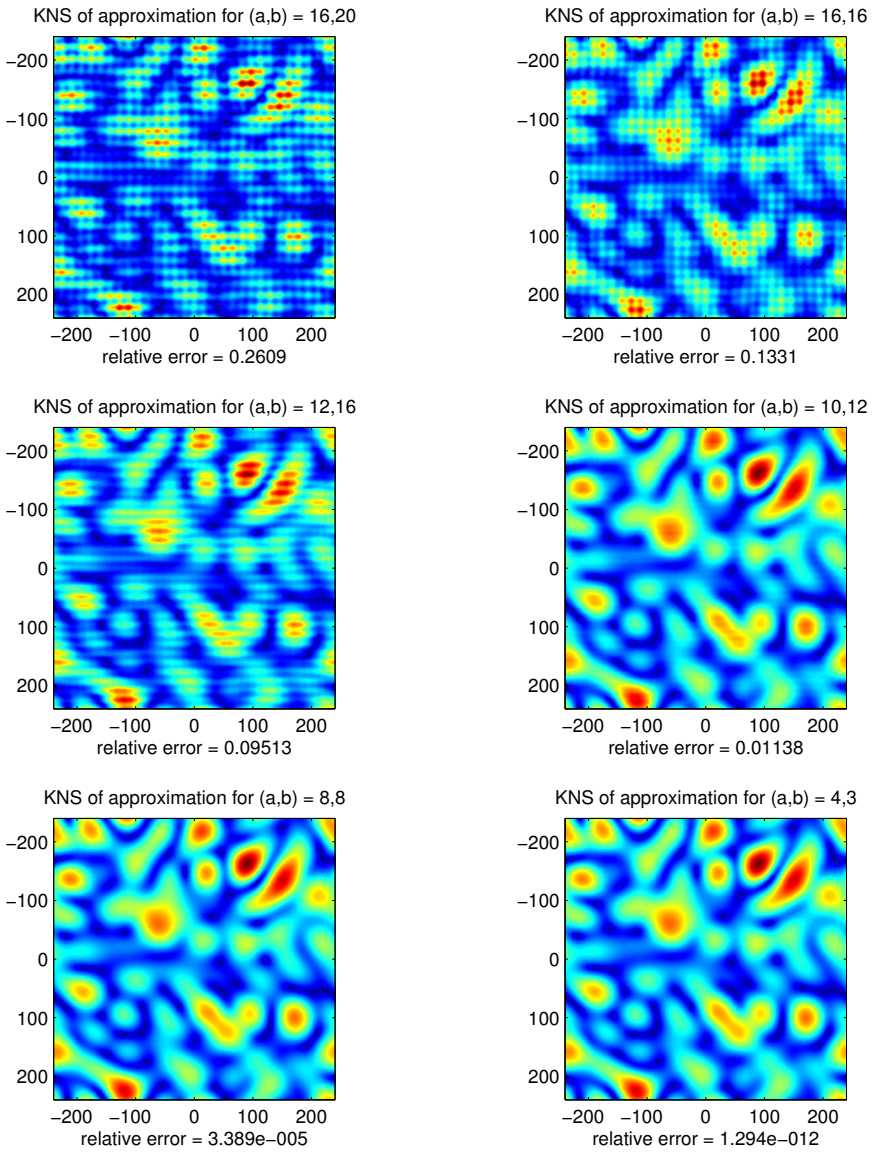


Figure 5.16.: Kohn-Nirenberg symbols of the best approximations with standard Gaussian and different lattice constants (a/b), n = 480.

Experiment 3 (Relative error of approximation of a projection operator P_λ)**.** *Now Experiment 2 is repeated approximating a projection operator P_λ which is not sitting on a lattice point, i.e $\lambda = (5, 5)$. In Table 5.2 we see that here it does not help a lot to take the canonical dual or tight window for the approximation, as for all it gives back a very bad error. But we still get a good approximation if we take low enough lattice constants a and b, i.e. for higher redundancies. The problem here is that the projection operator is not sitting on a lattice point, so it can not be well represented by a Gabor multiplier, as the family of projection operators over Λ reach there badly. Nevertheless, if we try this with a projection operator sitting in $(0, 0)$ (this is a lattice*

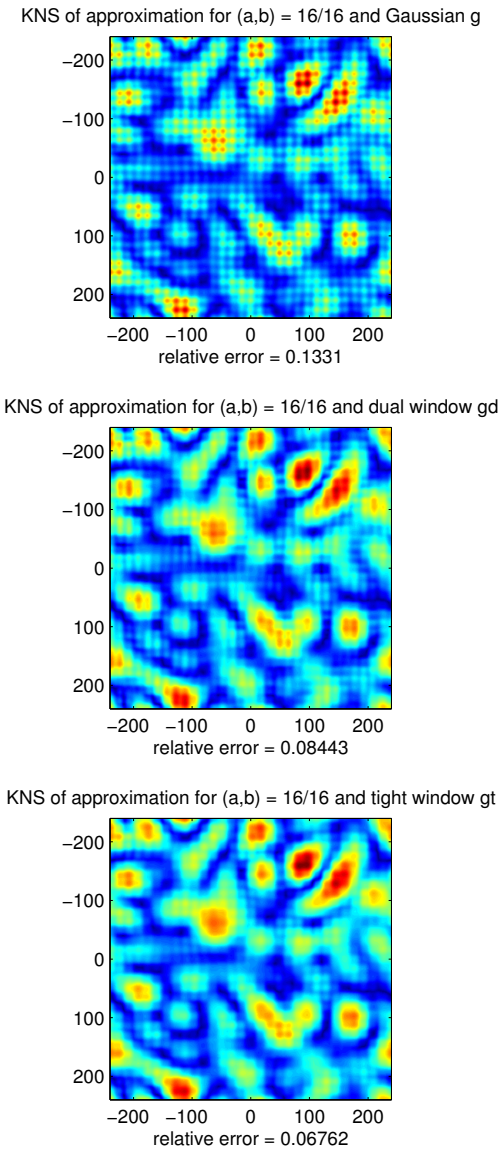


Figure 5.17.: Kohn-Nirenberg symbols of the best approximations with standard Gaussian as well as the dual and tight window in comparison.

point for all these lattices), then our approximation is almost perfect up to numerical precession even for high lattice constants like $a/b = 16/20$. *This is as expected, as we can take the Gabor multiplier with* m_λ *being* 0 *everywhere except for* $\lambda = (0, 0)$, *where it is equal to* 1. *In Figure 5.20 we see the relative approximation error for* P_λ *sitting in each point of the area between* $(1, 1)$ *and* $(60, 30)$. *The error is plotted as a color representation in that point. So we see the regions where the approximation is bad as the red regions.*

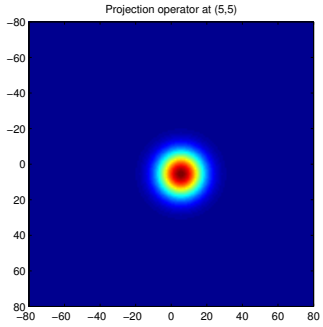


Figure 5.18.: Kohn-Nirenberg symbol of P_λ, displayed with 3x zoom

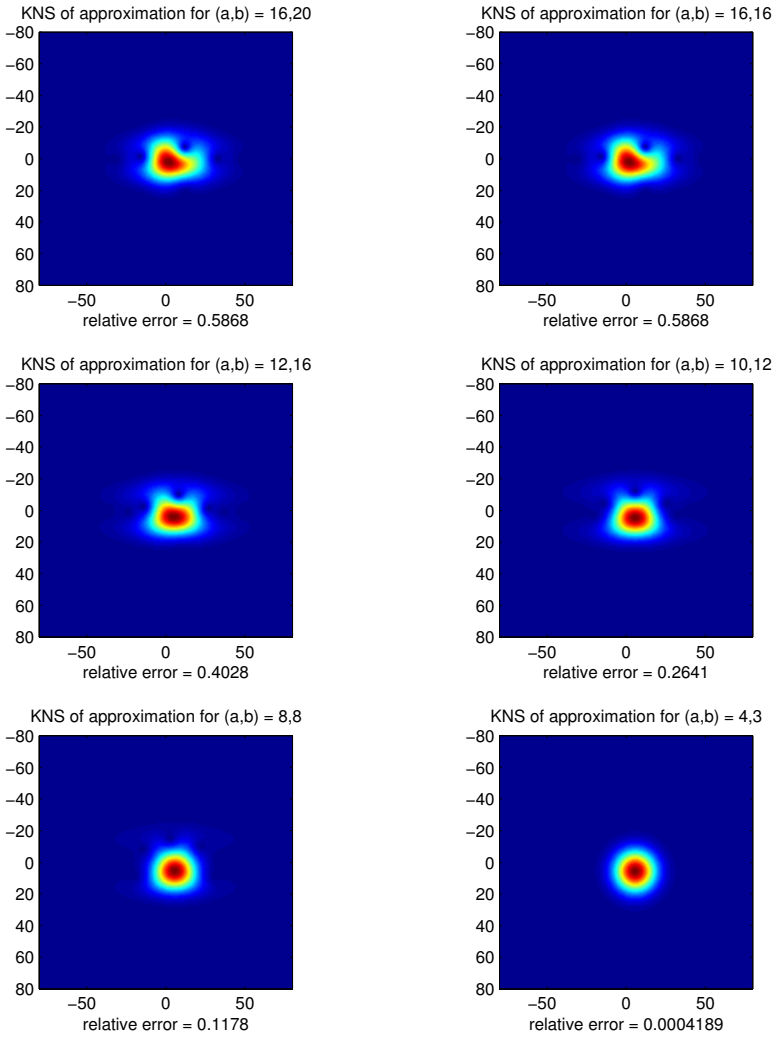


Figure 5.19.: Kohn-Nirenberg symbols of the best approximations of P_λ with standard Gaussian and different lattice constants (a/b), n = 480, displayed with 3x zoom.

Approximation of a projection operator P_λ

analysis/synthesis	a/b red	16/20 1.5	16/16 1.875	12/16 2.5	10/12 4	8/8 7.5	4/3 40
g/g		58.68%	58.68%	40.28%	26.41%	11.78%	0.04%
g/gd		61.57%	58.93%	40.7%	26.42%	11.78%	0.87%
gt/gt		60.2%	58.66%	40.4%	26.41%	11.78%	0.87%

Table 5.2.: Relative error of approximation for a standard Gaussian (g) as well as for the dual (gd) and tight (gt) window with different lattice constants (a/b), n = 480.

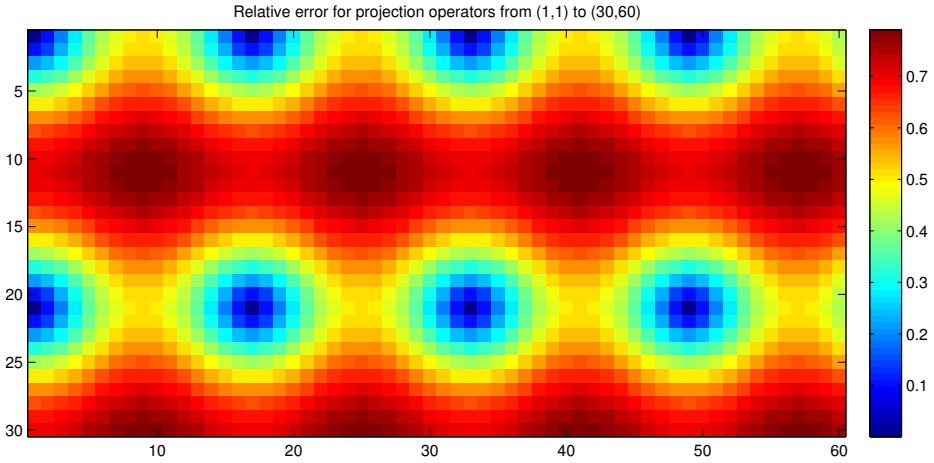


Figure 5.20.: Relative approximation error for P_λ sitting on a point $\lambda = (x, \omega)$ with $x = 1, \ldots, 60$ and $\omega = 1, \ldots, 30$. n = 480, seperable lattice with a = 16, b = 20

We can learn form the two previous experiments that regions near the lattice points are better represented by the approximation. This is also the reason why higher redundancy produces a lower error, as they cover the area more densely. Using the tight window will compensate some problems in the regions between the lattice points. For calculation of the canonical dual or tight window the functions from the LTFAT (Linear Time-Frequency Analysis Toolbox) by Peter L. Søndergaard were used. They can be found on http://ltfat.sourceforge.net/. Using them did not take more time for the calculations, which means we can always use the tight window and do not need to be concerned about the algorithm taking longer. At least not for dimensions used for the tests.

Approximation of STFT multipliers

Bandwidth of the symbol: 15

analysis/synthesis	a/b red	16/20 1.5	16/16 1.875	12/16 2.5	10/12 4	8/8 7.5	4/3 40
g/g		50.75%	38.49%	28.02%	7.47%	0.09%	1.2e-010%
gt/gt		49.61%	37.63%	27.45%	7.48%	0.09%	3e-012%

Bandwidth of the symbol: 20

analysis/synthesis	a/b red	16/20 1.5	16/16 1.875	12/16 2.5	10/12 4	8/8 7.5	4/3 40
g/g		57.5%	46.24%	36.16%	14.05%	0.35%	1.1e-010%
gt/gt		56.72%	45.75%	35.88%	14.05%	0.35%	2.9e-012%

Table 5.3.: Relative error of approximation of STFT multipliers with different bandwidth for a standard Gaussian (g) and tight (gt) window with different lattice constants (a/b), n = 480.

When doing an approximation of a STFT multiplier with random symbol of higher bandwidth the error gets worse. As one can see in Table 5.3 with bandwidth 15 resp. 20, we get high relative errors and even taking the tight window does not help a lot in most cases. Increasing the redundancy does help, but for STFT multipliers with high bandwidth, we need something like red = 4 or red = 7.5 to get an acceptable approximation error lower than 10%. We see that for higher bandwidth of the symbol, the approximation gets worse. This indicates that STFT multipliers can be better approximated by Gabor multipliers if the symbol is smooth, when using a low redundancy lattice.

5.2.2. Approximation Error in Dependence of the Bandwidth

We now take the seperable lattice $\Lambda = a\mathbb{Z} \times b\mathbb{Z}$ with $a = 16, b = 20$ and $n = 480$. We produce a *pure frequency* for our symbol of the STFT multiplier, see Figure 5.21 for a pure frequency with $(\nu_x, \nu_y) = (3, 5)$. Then we build the STFT multiplier and approximate it with a Gabor multiplier using the seperable lattice $a\mathbb{Z} \times b\mathbb{Z}$ (red = 1.5) and a Gaussian window. Doing this for all frequencies (ν_x, ν_y) between $(1, 1)$ and $(60, 30)$ and plotting for each the relative error of approximation, we get Figure 5.22. What we see is that the approximation is good for frequencies with about $\|(\nu_x, \nu_y)\|_2 \leq 5$ but gets high after that. This routine can be used to find out how high the frequencies in the upper symbol can be chosen, such that the error is still under control. In this very case we see that the bandwidth 4 we chose for the STFT multiplier in Figure 5.15 is suitable to be well approximated. If we took lower lattice constants a and b, then we could go to higher frequencies and still have a reasonable approximation, i.e. we need higher redundancies for higher frequencies in the upper symbol.

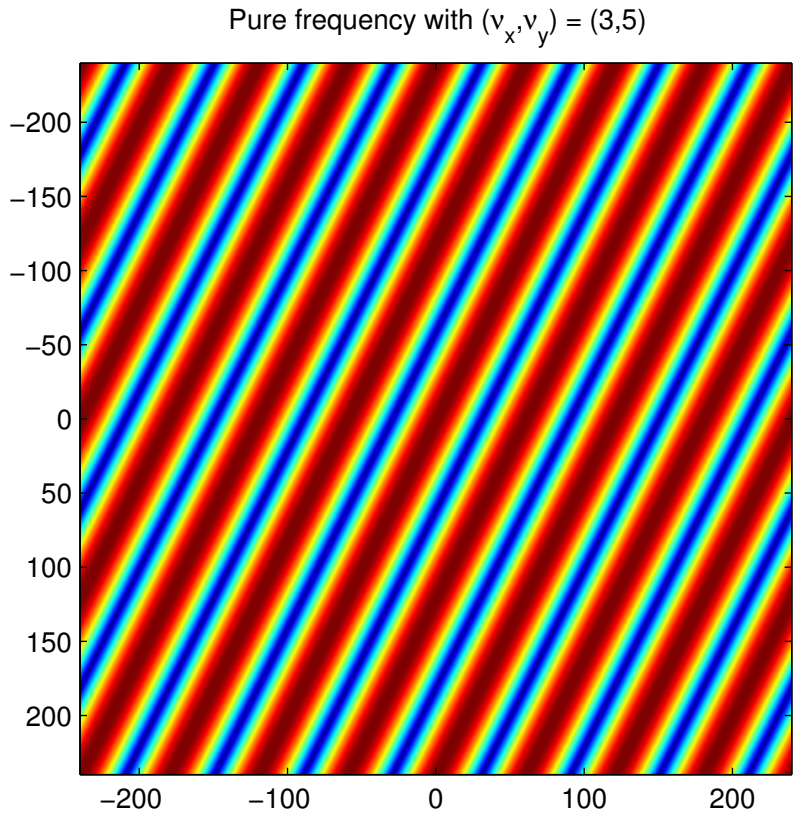


Figure 5.21.: 2-dimensional pure frequency with $(\nu_x, \nu_y) = (3, 5)$

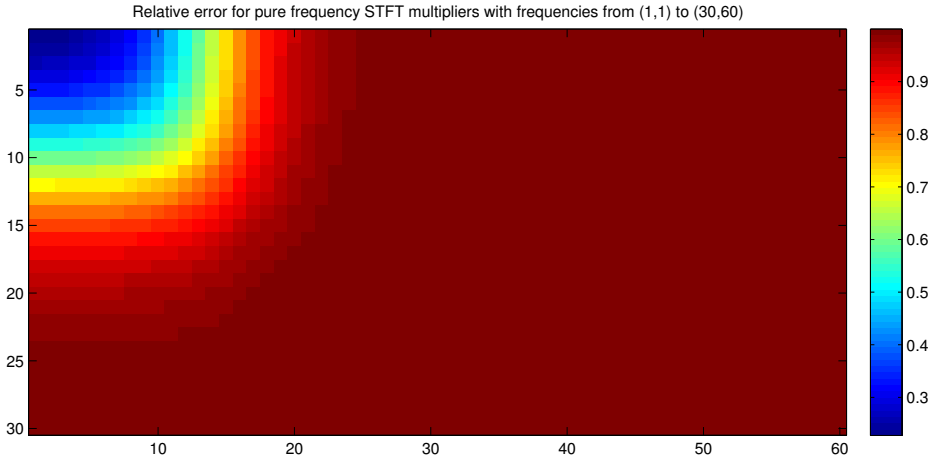


Figure 5.22.: Relative approximation error for pure frequency STFT multipliers $\nu_x = 1, \ldots, 60$ and $\nu_y = 1, \ldots, 30$. n = 480, seperable lattice with a = 16, b = 20

5.2.3. Finding the Best Lattice for Approximation

In this section we take a look at how approximation by Gabor multipliers is depending on the compound condition numbers from Section 5.1.3. For this we take $\{\Lambda_i^{red}\}_{i\in I}$ and approximate a STFT multiplier by a Gabor multiplier with each Λ_i and a Gaussian. Then we look at the relative approximation error and find the lattice Λ_j for which the error is lowest. In Figures 5.23 and 5.24 we can see the approximation error plotted for all lattices in $\{\Lambda_i^{red}\}_{i\in I}$, as well as the approximation for the lattice that produces the lowest error.

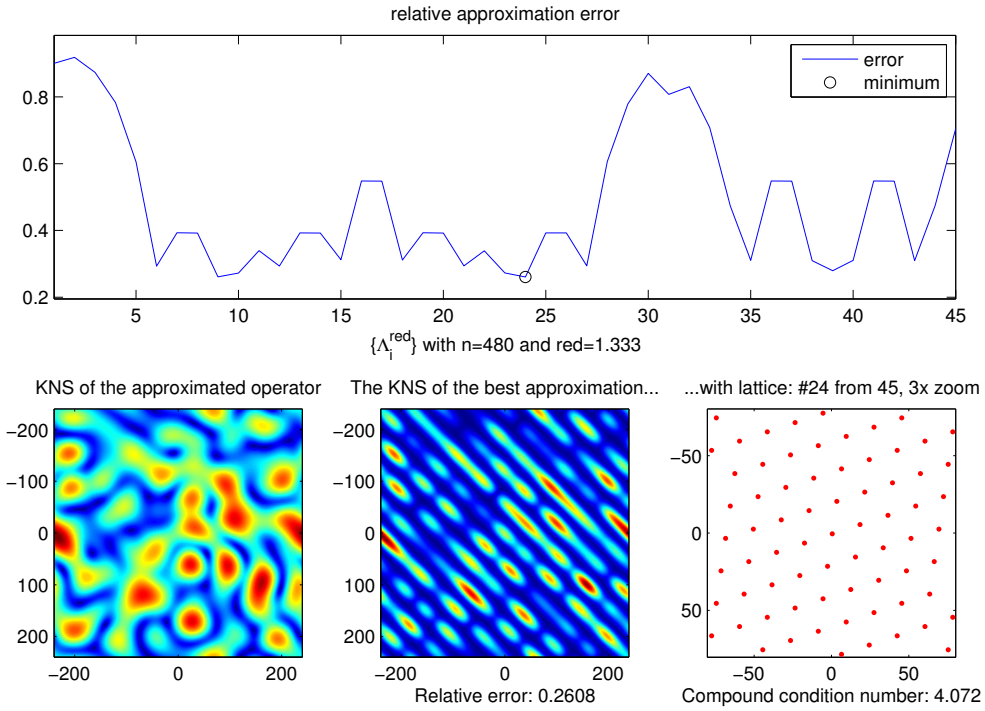


Figure 5.23.: Relative approximation error of a STFT multiplier approximated by a Gabor multiplier with Gaussian window and $\Lambda \in \{\Lambda_i^{red}\}_{i\in I}$, n = 480, red = 1.333

When comparing Figures 5.23 and 5.24 with Figures 5.11 and 5.10 we can see that the graphs are shaped similar, i.e. lattices with low compound condition number were good for the approximation and lattices with high compound condition number were bad. This is a nice result, as we see that the Gabor systems with a low compound condition number are the same as the ones best suited for approximation by Gabor multipliers. If we take a look at the best fitting lattice, which is #24 for red = 1.333, #7 for red = 1.5 and #135 for red = 2, the are either the same as in the tests for the compound condition number or it is one with the same condition number as the minimum from these tests. Results can be seen in Tables 5.4 and 5.5.

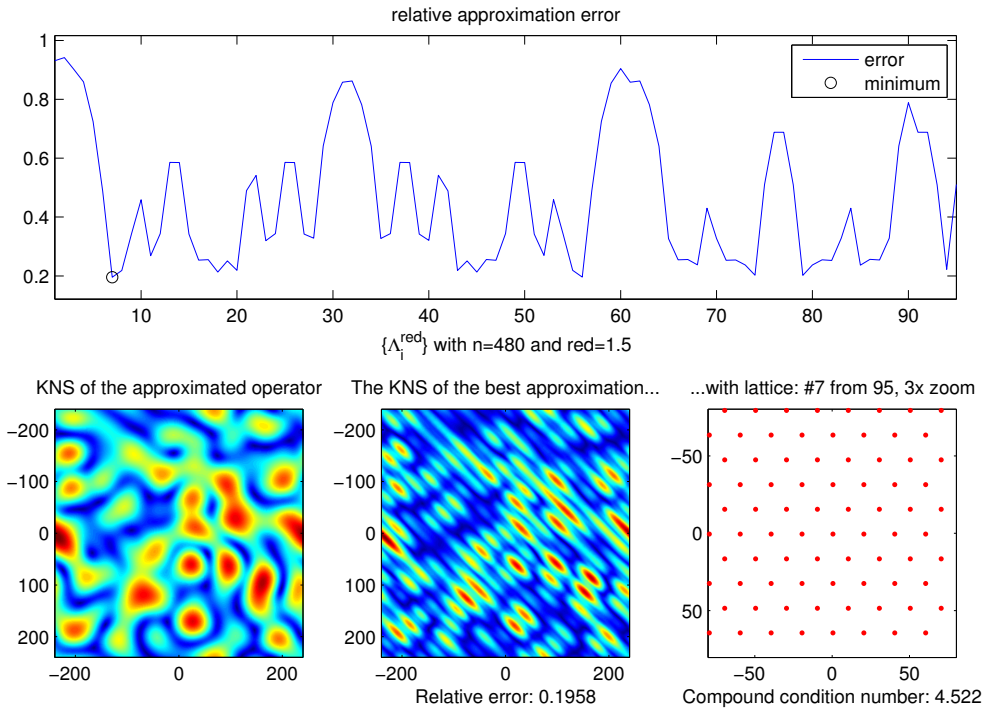


Figure 5.24.: Relative approximation error of a STFT multiplier approximated by a Gabor multiplier with Gaussian window and $\Lambda \in \{\Lambda_i^{red}\}_{i \in I}$, n = 480, red = 1.5

n \ red	1.333	1.5	2
144	#10 / 4.088	#23 / 5.664	#53 / 7.977
360	#57 / 3.912	#68 / 4.46	#48 / 7.261
480	#24 / 4.072	#7 / 4.522	#135 / 7.271

Table 5.4.: Lattice # in $\{\Lambda_i^{red}\}_{i \in I}$ for the best approximation of a STFT multiplier and its corresponding compound condition number

n \ red	1.333	1.5	2
144	#5 / 4.088	#23 / 5.664	#53 / 7.977
360	#57 / 3.912	#75 / 4.46	#15 / 7.261
480	#24 / 4.072	#7 / 4.522	#128 / 7.271

Table 5.5.: Lattice # in $\{\Lambda_i^{red}\}_{i \in I}$ with lowest compound condition number

If we take a look at the shapes of the lattices that form a good Gabor system together with a gaussian window, we find that they all are close to being hexagonal lattices. In Figures 5.2 to 5.11 we see that the "bad" lattices are the ones that have big regions without lattice points and other regions where there are many in a concentrated

form. Also the fact that regions between lattice points give a bad approximation error, imply that we want a lattice where the points are equally distributed over the TF-plane. Hexagonal lattices seem to be nicely spread out over the TF-plane and very well fitting for the circular shape of the STFT of a Gaussian window.

5.2.4. Testing the Janssen Condition Number

We ask ourself the question when the Janssen condition number as defined in Equation 5.3 is a good estimate for the condition number of the frame operator $cond(S_{g,\Lambda})$. What we want to find out is, for which Gabor systems the Janssen condition number is a bad estimate for the condition number of the frame operator. For this we go through $\{\Lambda_i^{red}\}_{i\in I}$ and calculate both. Then we get rid of the ones for which the estimate does not work, as $\gamma_1 > 1$. After this we plot the two condition numbers and mark the ones where the Janssen condition number is more than 1.5 times the actual condition number of the frame operator. By looking at the compound condition number of the Gabor systems where the estimate is bad, we see that they all are greater than something around 30. The ones we took out because $\gamma_1 > 1$ where actually the ones with quite high compound condition number greater than 100 most of the time. This means that for Gabor systems with low compound condition number, the estimate (Janssen condition number) is alright.

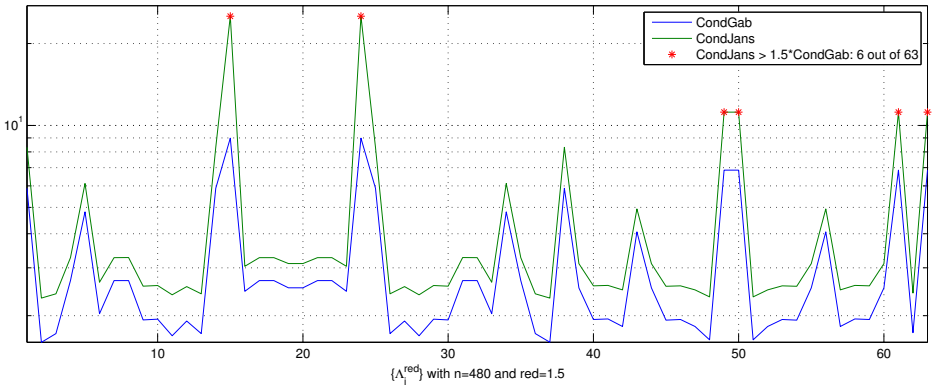


Figure 5.25.: Comparing the Janssen condition number and the condition number of the frame operator for a Gaussian window and $\Lambda \in \{\Lambda_i^{red}\}_{i\in I}$, n = 480, red = 1.5

5.2.5. Approximating Underspread Operators

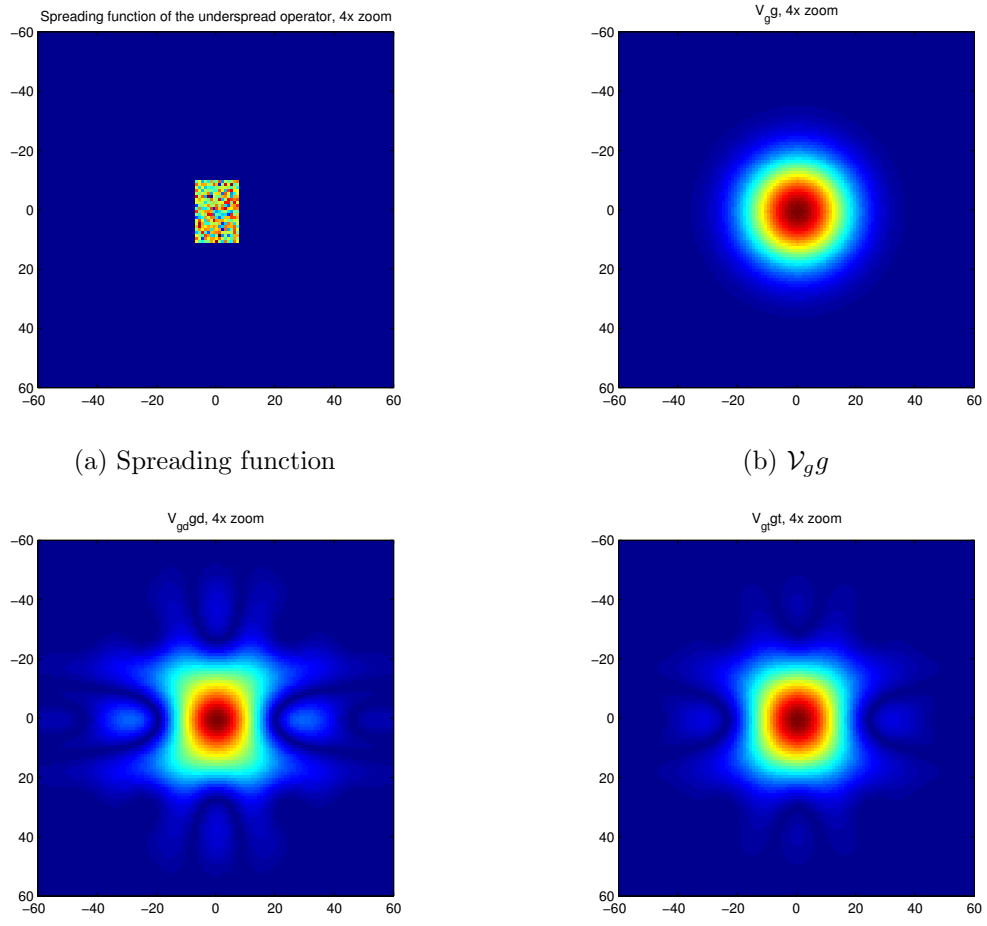
In the finite discrete setting, condition (4.24) for an operator T to be underspread translates to

$$s := 4t_0\nu_0 < n \tag{5.8}$$

where the support of the spreading function $\eta(T)$ lies within $[-t_0, t_0] \times [-\nu_0, \nu_0]$. And as of Corollary 4.3.1, an operator T can be well approximated by a Gabor multiplier, if and only if $(\mathcal{V}_g g(\lambda))_{\lambda\in\Lambda}$ is non-zero on the support of the spreading function $\eta(T)$. Now given an underspread operator, this gives us a condition on the Gabor system for the approximation. It also tells us, that the STFT of the window g should be shaped

about the same as the spreading function. So if we have an operator with very high but narrow spreading function, e.g. $\text{supp}(\eta) = [-3, 3] \times [-20, 20]$, then the STFT of the window g should be shaped as well high and narrow.

Experiment 4. *For this experiment, I built a random underspread operator T with support of the spreading function $\eta(T)$ in $[-7, 7] \times [-10, 10]$. In Figure 5.26 we can clearly see that the above mentioned condition on $(\mathcal{V}_g g(\lambda))_{\lambda \in \Lambda}$ is sattisfied. The same applies to the dual window $\widetilde{g}$ and the tight window g_t. Now $4t_0\nu_0 = 4 * 10 * 7 = 280 < 480$, i.e. we do have an underspread operator as the inequality (5.8) is sattisfied. The results of approximating this underspread operator can be seen in Table 5.6. What we see is that the approximation quality rises with increasing redundancy, but using the dual window for resynthesis or even the tight window does not give a much better error. This might be due to the fact, that even both, the STFT of the dual and the tight window, are shaped very much like the spreading function, the standard Gaussian has just as nice a shape.*



(a) Spreading function

(b) $\mathcal{V}_g g$

(c) $\mathcal{V}_{\widetilde{g}} \widetilde{g}$ for $a = 16, b = 20$

(d) $\mathcal{V}_{g_t} g_t$ for $a = 16, b = 20$

Figure 5.26.: Spreading function of the underspread operator with $\text{supp}(\eta(T)) = [-7, 7] \times [-10, 10]$ and $\mathcal{V}_g g$ for comparison, displayed with 4x zoom

Approximation of an underspread operator

analysis/synthesis	a/b red	16/20 1.5	16/16 1.875	12/16 2.5	10/12 4	8/8 7.5	4/3 40
g/g		35.44%	23.27%	13.73%	2.01%	1.7e-002%	1.4e-010%
g/gd		33.92%	21.59%	13.23%	2.07%	1.7e-002%	3.3e-012%
gt/gt		30.53%	20.27%	11.94%	1.99%	1.7e-002%	3.2e-012%

Table 5.6.: Relative error of approximation for a standard Gaussian (g) as well as for the dual (gd) and tight (gt) window with different lattice constants (a/b), n = 480.

5.3. Conclusion and Outlook

The aim of this thesis was to investigate the quality of approximation of an operator in terms of the ingredients of a Gabor multiplier, i.e. the window function g and the time-frequency lattice Λ. As in theory this can be done for any lattice and any suitable window function, we have only done tests with a standard Gaussian resp. the canonical dual or tight window. But one could as well try to use other functions, like *streched/compressed* and *rotated Gaussians* (oval STFT shapes) as they might fit different lattices better. It seems quite obvious that it is good if the STFT of the window function is adapted to the lattice in a way that the whole frame covers the time-frequency plane as good as possible. Another option for windows that can form a frame are the Hermite functions. Doing further testing with different frames that originate from other windows might give some more insight to approximation quality.

- Can lattices that give a very bad approximaiton error with a Gaussian window become a viable choice with another window function?
- Is this independent of the operator we want to approximate?

One concept we did not look into is the concept of *multi-window Gabor frames*, i.e window functions $g_1, \dots, g_r \in \boldsymbol{L}^2(\mathbb{R}^d)$ with corresponding lattices $\Lambda_1, \dots, \Lambda_r \trianglelefteq \mathbb{Z}^{2d}$, such that the union $\bigcap_{i=1}^{r} \mathcal{G}(g_i, \Lambda_i)$ forms a frame. This might be worth looking at for specific operators.

We can also try to approximate operators with so-called *multiple Gabor multipliers* (MGM) as definded in [DT07]. These are operators of the form

$$\mathbb{M} = \sum_j G_{g,h_j,\Lambda,\mathbf{m}_j}$$

for upper symbols $\mathbf{m}_j$ with $g \in \boldsymbol{S}_0(\mathbb{R}^d)$ and a family of resynthesis windows $h_j \in \boldsymbol{S}_0(\mathbb{R}^d)$. Not to be confused with the above mentioned multi-window Gabor frames, as here we have only one window function and one lattice. Now we can again ask the question:

- What is the quality of approximating an operator with a multiple Gabor multiplier $\mathbb{M}$ and how does it depend on the ingredients Λ, g, h_j?

Going even further one could try to take a look at *irregular Gabor frames*, which consist of a window function g and a subset Λ (not necessarily a subgroup) of $\mathbb{R} \times \widehat{\mathbb{R}}$. This setting is a bit harder, as the group structure of the lattice is lost and with it some very strong mathematical tools. But one can see that for some cases it might be usefull to have a good concentration of lattice points in areas where higher frequencies should be well represented and widen the gaps between lattice points in less important areas or areas with low frequencies. More on irregular Gabor frames/multipliers can be found in [Bal05].

- Can one get a better approximation quality for operators that give a large approximation error when using irregular Gabor frames?
- And is the complexity one buys using an irregular Gabor frame worth it?

In this paper we only really tested approximation of STFT multipliers with band-limited upper symbol functions (which are actually underspread if the limitation for the bandwidth is small enough). But then again one can ask the question of how good the approximation can be for various other operators in $\mathcal{HS}$. We know that not all operators in $\mathcal{HS}$ can be well approximated, but there are many more that can be.

In general, we only tried to find good lattices in the set $\{\Lambda_i^{red}\}_{i \in I}$ obtained by the `sidedigm` routine. This routine gives us a collection of lattices with fixed redundancy. In the special case of $n = 480$ and $red = 1.5$, we get 95 different lattices ($|I| = 95$). The total number of subgroups of $\mathbb{Z}^n \times \mathbb{Z}^n$ with order $k = n \times red$ would be 186. This means, that there are a lot more lattices one could test. If we had a routine that produced all subgroups/lattices, we would find many more good, as well as bad, lattices. So with our tests we can not claim that we found the best or the worst lattices over all, but only out of the collection $\{\Lambda_i^{red}\}_{i \in I}$.

- Would it give us a big advantage to test the set of all subgroups/lattices for finding sufficiently good ones or is $\{\Lambda_i^{red}\}_{i \in I}$ a good enough sample?

Appendix A.

MATLAB Files

All MATLAB™ files are based on code from the NuHAG database. I do not claim this code being my own work, but I did some contributions to some of it and some of the files are done mainly by myself. The code is based on work of Hans G. Feichtinger, Radu Corneliu Frunza and Stevan Kanjo. Nevertheless, the computations for the experiments in this paper are done by myself with the encouraging help of Hans G. Feichtinger and Radu Corneliu Frunza. All functions that are not listed here but appear in the code are eigther standard MATLAB™ functions or functions that can be found in the NuHAG database (http://www.nuhag.eu/) or the LTFAT (http://ltfat.sourceforge.net/). Note that in these codes, a signal, function or window is always a *row* vector.

For simplicity reasons, the copyright agreement is taken out of the `.m` files and put seperatly.

A.1. condfei.m

```
1  % CONDFEI.M,  hgfei Nov. 2011
2  %
3  % Description : This function allows to determine/assess the overall
4  %               quality of a Gabor family.
5  %
6  % Input :       g = the Gabor atom of the Gabor system
7  %               xp1 = a 0/1 matrix describing the 2-D grid (lattice)
8  %
9  % Output :      condf  = the compound condition number.
10 %               cond1  = the condition number square root of the ...
       frame
11 %                        operator.
12 %               cond2  = the P-condition number of the projection ...
       operators.
13 %               faVgg1 = the diagonalized Gram matrix of the ...
       projection
```

```
14 %                          operators.
15 %
16 % Usage :       [condfei,cond1,cond1,faVgg1] = condfei(g,xp1);
17
18 function [condf,cond1,cond2,faVgg1] = condfei(g,xp1);
19 G1 = gabbas(g,xp1);
20 S1 = G1'*G1;
21 cond1 = sqrt(cond(S1));
22
23 Vgg = stft(g,g);
24 aVgg1 = abs(Vgg).*xp1;
25 faVgg1 = fft2(aVgg1);
26 cond2 = sqrt(max(abs(faVgg1(:)))/min(abs(faVgg1(:))));
27
28 condf = cond1 * cond2;
```

A.2. trl2aphf.m

```
1  % TRL2APHF.M
2  %
3  % Description : This function finds the best approximation of the ...
     2-D signal xx
4  %               in the space spanned by translations of the 2-D ...
     atom over a 2-D grid xpo.
5  %
6  % Input :       xx = a matrix, describing the 2-D signal
7  %               atom = a matrix, describing the 2-D atom on the ...
     fourier side
8  %               third argument xpo  = a 0/1 matrix describing the ...
     2-D grid
9  %               third argument gapx = a number, describing the ...
     gap in x-direction
10 %                                        (used to calculate 2-D grid)
11 %               fourth argument gapy = a number, describing the ...
     gap in y-direction
12 %                                         (used to calculate 2-D grid)
13 %
14 % Output :      xa = a matrix, describing the best approximation ...
     of xx using atom
15 %               relerr = a number, the relative error of the ...
     approximation
16 %               cof = a matrix, the coefficients of xx in the ...
     translational space of atom.
17 %
18 % Usage :      [xa,relerr,cof] = trl2aphf(xx,atom,xpo);
19 %                                = trl2aphf(xx,atom,gapx,gapy);
20 %                                = trlaphf; starts a demonstration
21 %
22 % Author(s) : S. Kanjo 1994, A. Hilfinger 08/1999
23 %
24 % Subfunctions : ONEOVER; PUREGRID
25
26 function [xa,relerr,lam] = trl2aphf(xx,atom,gapx,gapy)
27
```

```
[hig,wid]=size(atom);
% Constructing the 2-D grid
if nargin == 3
    xpo=gapx;
    xp0=puregrid(xpo);
    [mm,nn]=size(xp0);
else
    xpo=zeros(hig,wid);
        xp0=ones(hig/gapy,wid/gapx);
    mm=hig/gapy;
    nn=wid/gapx;
    xpo(1:gapy:hig,1:gapx:wid)=ones(mm,nn);
end;
xpo=logical(xpo);xp0=logical(xp0);
c=zeros(mm,nn);
coef=zeros(mm,nn);
xes=zeros(hig,wid);

fatom=fft2(atom);
b2=ifft2(abs(fatom).^2);
c(xp0)=b2(xpo);

l1=ifft2(fft2(xx).*conj(fatom));
coef(xp0)=l1(xpo);
lam=ifft2(fft2(coef).*oneover(fft2(c)));
cof=lam(xp0).';
xes(xpo)=cof;

xa=ifft2(fatom.*fft2(xes));

if nargout > 1
    relerr=norm(xx-xa,'fro')/norm(xx,'fro');
end
end

% SUBFUNCTION
% Function    y = oneover(x);
% Descripton: Given a matrix x, oneover(x) inverts the
%             values above 10^(-10) and sets the rest to zero.

function  y = oneover(x)
ndx = find(abs(x)>0.0000000001);
y = zeros(size(x));
y(ndx)=ones(size(ndx))./x(ndx);
end

% SUBFUNCTION
% Function   xpopur = puregrid(xpo);
% Description: Given a gridmatrix xpo, puregrid (xpo) removes
%              null rows and columns (null must be exact)

function  xpopur = puregrid(xpo)
scol=sum(xpo);
f0col=find(scol);
xpopur=xpo(:,f0col);

```

```
srow=sum(xpopur');
f0row=find(srow);
xpopur=xpopur(f0row,:);
end
```

A.3. gmapr2hf.m

```
% GMAPR2HF.M, hgfei Oct. 2011
%
% Description: This routine calculates the best approximation (in the
%              Hilbert-Schmidt sense) of a general matrix by Gabor
%              multipliers with analysis/synthesis atoms g1 and ...
    g2 and the
%              lattice given by LAM. It is done by switching to ...
    the Kohn-
%              Nirenberg symbol of Pg=g1xg2 and T. There KN(T)is
%              approximated by orthogonal projection onto the ...
    space of
%              shifted KN(Pg) on the lattice. This space spans by ...
    shift
%              covariance the space of KN of all Gabor ...
    multipliers with g1,
%              g2 and LAM.
%
% Input:  T = matrix/operator
%         g1 = Gabor analysis atom
%         g2 = Gabor synthesis atom
%         LAM = TF-lattice
%
% Output:  Tap = T-approximation by Gabor multipliers
%          KNTap = KN-symbol of best approximation
%
% Usage: [Tap,KNTap,SPTap,KNT,KNPg] = gmapr2hf(T,g,LAM);
%        [Tap,KNTap,SPTap,KNT,KNPg] = gmapr2hf(T,g1,g2,LAM);

function  [Tap,KNTap,SPTap,KNT,KNPg] = gmapr2hf(T,g1,g2,LAM)

if nargin < 4
    LAM = g2;
    g2 = g1;
end

KNPg =  mat2kns(g2(:)*g1); % KN-symbol of the rank-one operator P_0
KNT    =  mat2kns(T);  % KN-symbol of T

% Best approximation of KNT by linear combinations of translates ...
    of the
% atom KNTg (shift covariance of the kns)
KNTap  =  trl2aphf(KNT,KNPg,LAM);

Tap    =  kns2mat(KNTap);  % Back from KN-symbol to matrix
SPTap  =  mat2spr(Tap);  % Calculate the spreading function of Tap
return
```

A.4. test_knsappr.m

```
% TEST_KNSAPPR.M, Feb. 2012, DOEPFNER K.
%
% Description : This function does an approximation of an ... operator using
%               different seperable lattices for the Gabor ... system. One can
%               do those calculations using the canonical dual ... frame for
%               resynthesizing or using the canonical tight frame.
%
% Usage :       Plots the KNS of the approximations for the lattices
%               declaired by aa and bb. For approximating different
%               operators, uncomment and comment beneath. Then ... evaluate
%               cells.
%
%


n = 480;
g = gaussnk(n);
aa = [16,16,12,10,8,4];
bb = [20,16,16,12,8,3];

% STFTm %%%%%%%%%%%%%%%%%%%
W = lowsign(n,n,4,4)*10^2;
% W = lowsign(n,n,15,15)*10^2;
STLOW = gabmulmh(W,g);  STLOW = STLOW/norm(STLOW,'fro');

% Proj op %%%%%%%%%%%%%%%%%
% ge = rotmod(g,5,-5);
% STLOW = ge(:)*ge;  STLOW = STLOW/norm(STLOW,'fro');

% STLOW = g(:)*g;  STLOW = STLOW/norm(STLOW,'fro');

% underspread op %%%%%%%%%%%%%%%%%%%
% Tu = lowspr(n,10,7);
% STLOW = Tu/norm(Tu,'fro');

KNT = mat2kns(STLOW);
%% Calculations for all 6 lattices
figure(1)
imgc(KNT); %zoom(3);
title('KNS of the STFT multiplier');
% title('Projection operator at (5,5)');

err = zeros(6,1);
t = zeros(1,6);
figure(2)
for jj = 1 : 6
    tic
%     gd = gabdual(g,aa(jj),n/bb(jj));
%     gt = gabtight(g,aa(jj),n/bb(jj));
    LAM = lattp(n,aa(jj),bb(jj));

```

```
51      [Tap,KNTap,SPTap,KNT,KNPg] = gmapr2hf(STLOW,g,LAM);
52      t(jj) = toc;
53      subplot(3,2,jj);
54      imgc(KNTap); %zoom(3);
55      title([ 'KNS of approximation for (a,b) = '    ...
            num2str(aa(jj)),',',  num2str(bb(jj)) ] );
56      err(jj) = norm(STLOW-Tap,'fro');
57      xlabel(['relative error = ' num2str(err(jj))]);
58  end
59  t
60  clear t
61
62  %% Normal, dual and tight calculations for one lattice
63  figure(3)
64  subplot(311)
65  [Tap,KNTap,SPTap,KNT,KNPg] = gmapr2hf(STLOW,g,lattp(n,aa(2),bb(2)));
66  imgc(KNTap);%zoom(3);
67  title(['KNS of approximation for (a,b) = ' num2str(aa(2)) '/' ...
        num2str(bb(2)) ' and Gaussian g']);
68  xlabel(['relative error = ' num2str(norm(STLOW-Tap,'fro'))]);
69
70
71  subplot(312)
72  gd = gabdual(g,aa(2),n/bb(2));
73  [Tap,KNTap,SPTap,KNT,KNPg] = ...
        gmapr2hf(STLOW,g,gd,lattp(n,aa(2),bb(2)));
74  imgc(KNTap);%zoom(3);
75  title(['KNS of approximation for (a,b) = ' num2str(aa(2)) '/' ...
        num2str(bb(2)) ' and dual window gd']);
76  xlabel(['relative error = ' num2str(norm(STLOW-Tap,'fro'))]);
77
78
79  subplot(313)
80  gt = gabtight(g,aa(2),n/bb(2));
81  [Tap,KNTap,SPTap,KNT,KNPg] = gmapr2hf(STLOW,gt,lattp(n,aa(2),bb(2)));
82  imgc(KNTap);%zoom(3);
83  title(['KNS of approximation for (a,b) = ' num2str(aa(2)) '/' ...
        num2str(bb(2)) ' and tight window gt']);
84  xlabel(['relative error = ' num2str(norm(STLOW-Tap,'fro'))]);
```

A.5. best_appr_lattice.m

```
1  % BEST_APPR_LATTICE.m : Nov. 2011, DOEPFNER K.
2  %
3  % Description : Tests a set of lattices generated by genlatredT.m ...
       and finds
4  %               the one with which an operator is best ...
       approximated by
5  %               Gabor multipliers using the gmapr2hf.m routine ...
       for best
6  %               approximation. The Gabor system used for the Gabor
7  %               multipliers is a standard Gaussian window with ...
       the lattices.
8  %
```

```
% Input :   T   = normalized ('fro') operator to be approximated
%           n    = dimension
%           red  = redundancy for our lattices (frames)
%           win  = 0/1, 1 for tight window, 0 default.
%
% Output :  Plots the relative approximation error, the KNT, KNT
%           approximation and the best fitting lattice.
%           Tap = best approximation of the operator
%           L = lattice used for best approximation
%           y = relative error of approximation
%
% Usage :   [Tap,L,y] = best_appr_lattice(T,n,red,win);
%           [Tap,L,y] = best_appr_lattice(T,n,red);

function [Tap,L,y] = best_appr_lattice(T,n,red,win)
if nargin<4
    win = 0;
end
% generating set of lattices
TAB = genlatredT(n,red);
LAM = ndx2stck(n,TAB);

g = gaussnk(n); % standard gaussian window
ll = size(LAM,3); % ll is the number of lattices in the set
aprx = zeros(ll,1);

for jj = 1:ll
    if win == 1
        gtf = g*sqrtm(inv(gabfrmat(g,LAM(:,:,jj))));% tight ...
            window for the lattice
        gtf=gtf/norm(gtf); % normalization
    else
        gtf=g;
    end
    [Tap,KNTap,SPTap,KNT,KNPg] = gmapr2hf(T,gtf,LAM(:,:,jj));

    % The approximation error
    aprx(jj) =  norm(T-Tap,'fro');
end

[y,i] = min(aprx);

ccn = condfei(gtf,LAM(:,:,i));
L = LAM(:,:,i);

% Plots the relative approximation error
subplot(2,3,[1 3])
plot(1:ll,aprx,i,y,'ko'); ax20; %grid;
title('relative approximation error');
xlabel(['\{\Lambda_i^{red}\} with n=' num2str(n) ' and red=' ...
    num2str(red)]);
legend('error','minimum')

% Plots KNT, KNTap and the best fitting lattice
subplot(234); imgc(KNT); %colorbar; plotax;
title('KNS of the approximated operator');
```

```
63
64 subplot(235); imgc(KNTap); %colorbar; plotax;
65 title('The KNS of the best approximation...');
66 xlabel(['Relative error: ' num2str(y)]);
67
68 subplot(236); spyc(L); zoom(3);
69 title(['...with lattice: #' num2str(i) ' from ' num2str(ll) ', 3x ...
       zoom']);
70 xlabel(['Compound condition number: ' num2str(ccn)]);
71 end
```

A.6. test_unitmat.m

```
1  % TEST_UNITMAT.M,  DOEPFNER K. Feb. 2012
2  %
3  % Description : Does a best approximation of a projection ...
       operator Plam
4  %               sitting in the points from (0,0) to (spx,spy).
5  %
6  % Input :   g       = the Gabor atom of the Gabor system
7  %           gapx    = a number, describing the gap in x-direction
8  %                     (used to calculate 2-D grid)
9  %           gapy    = a number, describing the gap in y-direction
10 %                     (used to calculate 2-D grid)
11 %           spx,spy =  last point where Plam sits in.
12 %
13 % Output :     err  = error-matrix of size spy x spx
14 %
15 % Usage :       err = test_unitmat(g,gapx,gapy,spx,spy);
16
17 function err = test_unitmat(g,gapx,gapy,spx,spy)
18 n = length(g);
19 LAM = lattp(n,gapx,gapy);
20 err = zeros(spy,spx);
21 % tic
22 for ii = 1:spy
23     for jj = 1:spx
24         W = unitmat(n,n,ii,jj);
25
26         STFTm = gabmulmh(W,g);  STFTm = STFTm/norm(STFTm,'fro');
27
28         [Tap,KNTap,SPTap,KNT,KNPg] = gmapr2hf(STFTm,g,LAM);
29
30         err(ii,jj) = norm(STFTm-Tap,'fro');
31     end
32 end
33 % t = toc
34 imagesc(err); colorbar;
35 title(['Relative error for projection operators from (1,1) to (' ...
       num2str(spy) ','  num2str(spx) ')']);
36 end
```

A.7. compcdgabjanskd.m

```
% COMPCDGABJANSKD.M,  DOEPFNER K. Feb. 2012
%
% Description : Calculates the condition number of the frame operator and
%               the Janssen condition number which estimates it. It goes
%               through all lattices in genlatredT(n,red), where n is the
%               length of g. One can then compare the two condition
%               numbers.
%
% Input :       g = the Gabor atom of the Gabor system
%               red = redundancy
%
% Output :      CDGgood = condition number of the frame operator for which
%                         the estimate works.
%               CDJgood = Janssen condition number.
%               LAMgood = lattices for which the estimate works.
%               LAMbad  = lattices for which the estimate does not work.
%          Displays an error, if the Janssen cond. number is not an upper
%          estimate.
%
% Usage :      [CDGgood,CDJgood,LAMgood,LAMbad] = compcdgabjanskd(g,red);

function [CDGgood,CDJgood,LAMgood,LAMbad] = compcdgabjanskd(g,red)
n = length(g);
TAB = genlatredT(n,red);
LAM = ndx2stck(n,TAB);

ll = size(LAM,3);
CDG = zeros(ll,1);
CDJ = zeros(ll,1);

for ii= 1:ll
    S = gabfrmat(g,LAM(:,:,ii));
    CDG(ii) = cond(S);
    CDJ(ii) = condjans(g,LAM(:,:,ii));
end

ndgood = find(CDJ > 1);
ndbad = find(CDJ <= 1);

CDGgood = CDG(ndgood); CDGgood = CDGgood(:);
CDJgood = CDJ(ndgood); CDJgood = CDJgood(:);

if length(find(CDGgood > CDJgood)) > 0;
    disp('there is a problem for');
    find(CDGgood > CDJgood);
end
LAMgood = LAM(:,:,ndgood);
LAMbad = LAM(:,:,ndbad);
```

```
49 end
```

A.8. test_cdgabjans.m

```
1  % TEST_CDGABJANS.M,  DOEPFNER K. Feb. 2012
2  %
3  % Description : Tests the difference between the actuall ...
       condition number
4  %               of the Gabor system and the Janssen condition ...
       number and
5  %               finds out for which the difference is more than a ...
       certain
6  %               given multiple.
7  %
8  % Input :   g    = the Gabor atom of the Gabor system
9  %           red = redundancy
10 %           maxmul = maxumim multiple for the estimate
11 %
12 % Output : LAMgood  = lattices for which the estimate is less ...
       than maxmul
13 %                     times.
14 %          LAMBAD   = lattices for which the estimate is more ...
       than maxmul
15 %                     times.
16 %
17 % Usage :      [LAMgood,LAMBAD] = test_cdgabjans(g,red,maxmul);
18
19 function  [LAMgood,LAMBAD] = test_cdgabjans(g,red,maxmul)
20 n = length(g);
21 [CDGgood,CDJgood,LAMgood,LAMbad] = compcdgabjanskd(g,red);
22
23 ll = length(CDGgood);
24
25 nbad = find(maxmul*CDGgood < CDJgood);
26
27 figure(1)
28 semilogy(1:ll,CDGgood,1:ll,CDJgood,nbad,CDJgood(nbad),'r*'); ...
       grid; axis tight; ax20;
29 legend('CondGab','CondJans',['CondJans > ' num2str(maxmul) ...
       '*CondGab: ' num2str(length(nbad)) ' out of ' num2str(ll)])
30 xlabel(['\{\Lambda_i^{red}\} with n=' num2str(n) ' and red=' ...
       num2str(red)]);
31
32 LAMBAD = LAMgood(:,:,nbad);
33
34 lb = length(nbad);
35 ccng = zeros(lb,1);
36 for jj = 1:lb
37     ccng(jj) = condfei(g,LAMBAD(:,:,jj));
38 end
39
40 l = size(LAMbad,3);
41 ccnb = zeros(l,1);
42 for jj = 1:l
```

```
43      ccnb(jj) = condfei(g,LAMbad(:,:,jj));
44  end
45
46  LAMgood(:,:,nbad) = [];
47
48  lg = size(LAMgood,3);
49  ccngood = zeros(lg,1);
50  for jj = 1:ll
51      ccngood(jj) = condfei(g,LAMgood(:,:,jj));
52  end
53
54  if any(ccnb < 100)
55      disp('there is gamma_1 > 1 ones less then 100')
56      ccnb = ccnb(find(ccnb < 100))
57  end
58
59  if any(ccng < 30)
60      disp('there is bad ones less then 30')
61      ccng = ccng(find(ccng < 30))
62  end
63
64  if any(ccngood > 30)
65      disp('there is good ones higher then 30')
66      ccngood = ccngood(find(ccngood > 30))
67  end
68  end
```

A.9. purfrapp.m

```
1  % PURFRAPP.M : Doepfner K., Mar. 2012
2  %
3  % Description : Goes through all pure frequencies from (1,1) to ...
     (spx,spy)
4  %               and does a best approximation of the STFT ...
     mutiplier with a
5  %               Gabor multiplier using the seperable lattice aZ x ...
     bZ and a
6  %               standard Gaussian window. The symbol of the STFT ...
     multiplier
7  %               consists of a single frequency at a time.
8  %               One can set
9  %                                    'flag' to 1
10 %               to do the computations using a tight window.
11 %               Finally one gets a plot with the relative errors of
12 %               approximation.
13 %
14 % Input :   n    = dimension
15 %           gapx = a number, describing the gap in x-direction
16 %                  (used to calculate 2-D grid)
17 %           gapy = a number, describing the gap in y-direction
18 %                  (used to calculate 2-D grid)
19 %           spx  = maximum wavelength in x-direction
20 %           spy  = maximum wavelength in y-direction
21 %
```

```
% Output : stsmdiff = Difference between the approximation and the
%                     Gabor multiplier of the sampled symbol.
%          aprerr   = Relative error for pure frequency STFT ...
    multipliers.
%    Plots the 2 outputs in color code.
%
% Usage :      [stsmdiff,aprerr] = ...
    purfrapp(n,gapx,gapy,spx,spy,flag);

function [stsmdiff,aprerr] = purfrapp(n,gapx,gapy,spx,spy,flag)
if exist('flag','var') == 0
    flag = 0;
end

LAM = lattp(n,gapx,gapy); % lattice
g = gaussnk(n); % Gaussian window

if flag == 1
    gt = gabtgtps(g,gapx,gapy); % tight window
end

stsmdiff = zeros(spy,spx);
aprerr = zeros(spy,spx);
for ii = 1:spy
    for jj = 1:spx
        % STFT multiplier
        W = purfr2(n,n,ii,jj);
        STFTm = gabmulhf(W,g); STFTm = STFTm/norm(STFTm,'fro');
        % approximation
        if flag == 1
            Tap = gmapr2hf(STFTm,gt,LAM);
        else
            Tap = gmapr2hf(STFTm,g,LAM);
        end
        % sampling
        GMsmp = gabmulhf(W(1:gapx:n,1:gapy:n),g);
        % difference and error
        stsmdiff(ii,jj) =  norm(Tap-GMsmp,'fro');
        aprerr(ii,jj) =  norm(Tap-STFTm,'fro');
    end
end
% Plot the difference between the approximation and the Gabor ...
    multiplier of
% the sampled symbol.
figure(1)
imagesc(stsmdiff); colorbar;
title('Difference between the approximation and the Gabor ...
    multiplier of the sampled symbol');

% Plot the relative errors of the approximation
figure(2)
imagesc(aprerr); colorbar;
title(['Relative error for pure frequency STFT multipliers with ...
    frequencies from (1,1) to (' num2str(spy) ','  num2str(spx) ')']);
end
```

Bibliography

[Bal05] Peter Balazs. *Regular and Irregular Gabor Multiplier With Application To Psychoacoustic Masking*. PhD thesis, 2005.

[Ban10] Severin Bannert. Banach-Gelfand Triples and Applications in Time-Frequency Analysis. Master's thesis, 2010.

[Ber01] Johanna Berndorfer. Spline-Type Spaces. Master's thesis, 2001.

[BP06] John J. Benedetto and Götz E. Pfander. Frame expansions for Gabor multipliers. *Appl. Comput. Harmon. Anal.*, 20(1):26–40, 2006.

[Chr93] Ole Christensen. *Frame Decompositions in Hilbert Spaces*. PhD thesis, 1993.

[Chr03] Ole Christensen. *An Introduction to Frames and Riesz Bases.* Applied and Numerical Harmonic Analysis. Birkhäuser, 2003.

[Dör02] Monika Dörfler. *Gabor Analysis for a Class of Signals called Music*. PhD thesis, 2002.

[DT07] Monika Dörfler and Bruno Torrésani. Spreading function representation of operators and Gabor multiplier approximation. In *Proceedings of SAMPTA07*. NuHAG;MOHAWI, June 2007.

[Fei92] Hans G. Feichtinger. Wiener amalgams over Euclidean spaces and some of their applications. 136:123–137, 1992.

[Fei02] Hans G. Feichtinger. Spline-type spaces in Gabor analysis. In D. X. Zhou, editor, *Wavelet Analysis: Twenty Years Developments Proceedings of the International Conference of Computational Harmonic Analysis, Hong Kong, China, June 4–8, 2001*, volume 1 of *Ser. Anal.*, pages 100–122. World Sci.Pub., 2002.

[Fei03] Hans G. Feichtinger. Gabor multipliers with varying lattices. In *Proc. SPIE Conf.*, page 14. NuHAG, August 2003.

[FHK04] Hans G. Feichtinger, M. Hampejs, and Guenther Kracher. Approximation of matrices by Gabor multipliers. *IEEE Signal Proc. Letters*, 11(11):883–886, November 2004.

[FK95] Hans G. Feichtinger and St. Kanjo. Best approximation of smooth signals by linear combinations of translates. In F. Solina et al., editors, *Proc. of OEAGM-95*, pages 205–212. NuHAG, May 1995.

[FK98] Hans G. Feichtinger and Werner Kozek. Quantization of TF lattice-invariant operators on elementary LCA groups. In Hans G. Feichtinger and T. Strohmer, editors, *Gabor Analysis and Algorithms. Theory and Applications.*, Applied and Numerical Harmonic Analysis, pages 233–266, 452–488. Gabor;NuHAG, Birkhäuser Boston, 1998.

[FK04] Hans G. Feichtinger and Norbert Kaiblinger. Varying the time-frequency lattice of Gabor frames. *Trans. Amer. Math. Soc.*, 356(5):2001–2023, 2004.

[FLW07] Hans G. Feichtinger, Franz Luef, and Tobias Werther. A guided tour from linear algebra to the foundations of Gabor analysis. In *Gabor and Wavelet Frames*, volume 10 of *Lect. Notes Ser. Inst. Math. Sci. Natl. Univ. Singap.*, pages 1–49. World Sci. Publ., Hackensack, 2007.

[FN03] Hans G. Feichtinger and Krzysztof Nowak. A first survey of Gabor multipliers. In Hans G. Feichtinger and T. Strohmer, editors, *Advances in Gabor Analysis*, Appl. Numer. Harmon. Anal., pages 99–128. Birkhäuser, 2003.

[FZ98] Hans G. Feichtinger and Georg Zimmermann. A Banach space of test functions for Gabor analysis. In Hans G. Feichtinger and T. Strohmer, editors, *Gabor Analysis and Algorithms: Theory and Applications*, Applied and Numerical Harmonic Analysis, pages 123–170. Gabor;NuHAG, Birkhäuser Boston, 1998.

[GL04] Karlheinz Gröchenig and Michael Leinert. Wiener's lemma for twisted convolution and Gabor frames. *J. Amer. Math. Soc.*, 17:1–18, 2004.

[Grö01] Karlheinz Gröchenig. *Foundations of Time-Frequency Analysis.* Appl. Numer. Harmon. Anal. Birkhäuser Boston, 2001.

[JS02] A. J. E. M. Janssen and Thomas Strohmer. Characterization and computation of canonical tight windows for Gabor frames. *J. Fourier Anal. Appl.*, 8(1):1–28, January 2002.

[Pau07] Stephan Paukner. Foundations of Gabor Analysis for Image Processing. Master's thesis, 2007.

[Pri96] Peter Prinz. Theory and Algorithms for Discrete 1-Dimensional Gabor Frames. Master's thesis, 1996.

[SB93] Josef Stoer and Roland Bulirsch. *Introduction to Numerical Analysis.* Springer-Verlag, 2 edition, 1993.

[Sch04] K. Schnass. Gabor Multipliers. A self-contained survey. Master's thesis, 2004.

[STB11] Peter Sondergaard, Bruno Torrésani, and Peter Balazs. The Linear Time Frequency Analysis Toolbox. *International Journal of Wavelets, Multiresolution and Information Processing*, to appear:accepted, 2011.

[Wei07] Ferenc Weisz. Inversion of the short-time Fourier transform using Riemannian sums. *J. Fourier Anal. Appl.*, 13(3):357–368, 2007.

[Wer97] Dirk Werner. *Funktionalanalysis. (Functional Analysis) 2., Überarb. Aufl.* Springer-Verlag, 1997.

Curriculum Vitae

Name:	Kirian Anselm Döpfner
Born:	September 30, 1985 Lambrechten, Austria
Nationality:	Austria
Parents:	Sieglinde and Horst (†) Döpfner
E-mail:	kirian.doepfner@gmail.com

1992 - 2001:	Primary and secondary school (VS and HS) at Bildungswerkstatt - Ried im Innkreis
2001 - 2003:	5th and 6th grade of grammar school at BORG - Ried im Innkreis
2003 - 2004:	7th grade of grammar school at James Hargest High School - Invercargill, New Zealand
2004 - 2005:	8th grade of grammar school at BORG - Ried im Innkreis, school leaving examination (Matura) with distinction
2005 - 2006:	Community service at Kinderfreunde - Wels
2007 - 2012:	Study of mathematics at the University of Vienna